UE07: OWL Reasoning

Semantic AI 2023, JKU
Course Instructor: Bernd Neumayr

Your tasks:

- Read and understand the given OWL ontologies.
- Do the OWL reasoning in your head and indicate the entailments in the given tables.
- You may use Protégé to check your reasoning (see 06-Intro-OWL).

Part I: Instance Checking (8 points)

Given are the following OWL ontologies.

For each individual derive

- of which classes it is definitely a member, i.e., that individual is a member of that class in every possible world (mark with "+"),
- of which classes it is definitely not a member, i.e., there is no possible world in which that individual is a member of that class (mark with "-"),
- and of which classes it is possibly a member, i.e., there is at least one possible world in which that individual is a member of that class and at least one possible world in which that individual is not a member of that class (mark with "?" or leave empty).

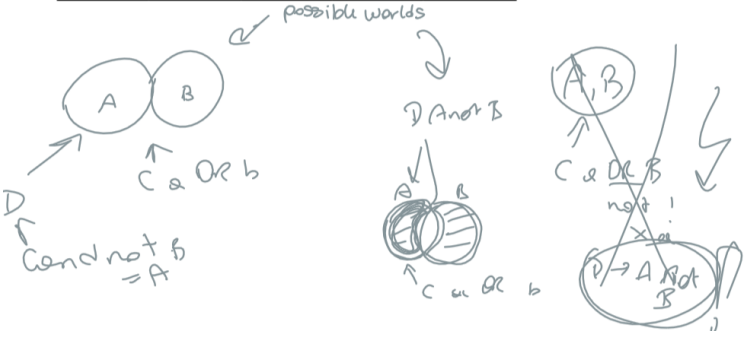
Task 1

```
Class: A
Class: B
Class: C
    EquivalentTo: A or B
Class: D
    EquivalentTo: C and (not (B))
Individual: o1
    Types: A
Individual: o2
    Types: B
Individual: o3
    Types: C
Individual: o4
    Types: D
```

edit:

	A	B	C	D
o1	+	?	+	?
o2	?	+	+	-
o3	?	?	+	?
o4	+	-	+	+

kann sein, dass sich A&B überschneiden



Task 2

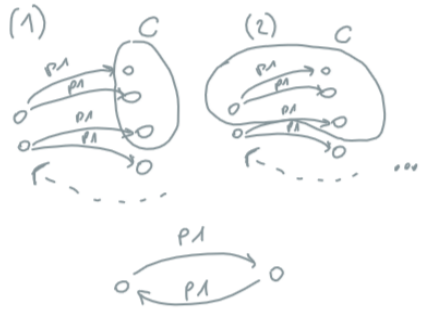
```
Individual: o1
  Facts: p1 o2
  Types: C
Individual: o2
  Types: C
ObjectProperty: p1
  Characteristics: Symmetric
Class:  C1
  EquivalentTo: p1 some C
Class:  C2
  SubclassOf: p1 some C
Class:  C3
  EquivalentTo: p1 only C
```

edit:

OPEN WORLD ASS.

→ KANN sein

	C1	C2	C3
o1	+	?	?
o2	+	?	?

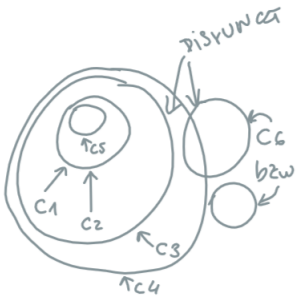


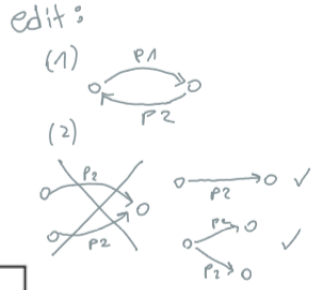
→ es KANN sein, dass o1 p1 some other obj.
→ wir wissen NICHT, dass C1 & C disjunct sind → würden wir das wissen, wäre C3 bei o1 & o2 jeweils "-"

Task 3

```
Individual: o1
  Types: C1
Class:  C1
Class:  C2
  EquivalentTo: C1
  SubclassOf: C3
Class: C3
  SubclassOf: C4
Class: C4
Class: C5
  SubclassOf: C1
Class: C6
  DisjointWith: C3
```

	C1	C2	C3	C4	C5	C6
o1	+	+	+	+	?	-





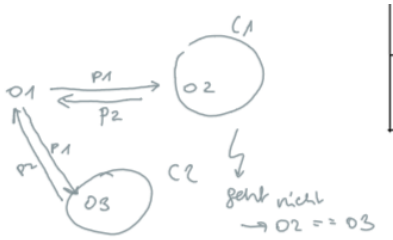
Task 4

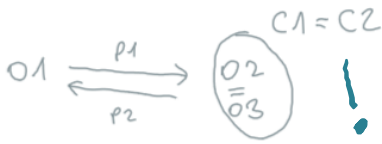
```
Individual: o1
  Facts: p1 o2, p1 o3
Individual: o2
  Types: C1
Individual: o3
  Types: C2

ObjectProperty: p1
  InverseOf: p2
ObjectProperty: p2
  Characteristics: InverseFunctional
```

AXIOM → ALWAYS TRUE!

	C1	C2
o1	?	?
o2	+	+
o3	+	+





Task 5

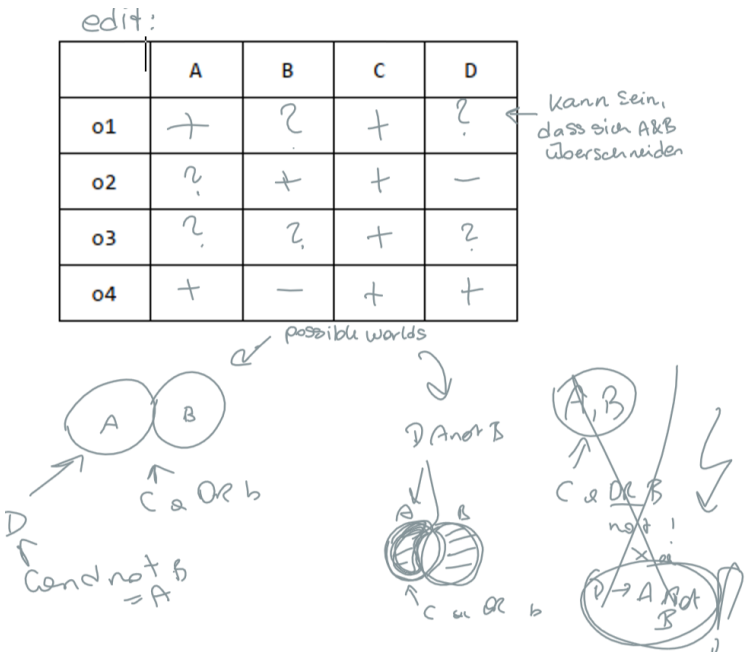
(selbes bsp wie task 1)

```
Class: A
Class: B
Class: C
    EquivalentTo: A or B
Class: D
    EquivalentTo: C and (not (B))
Individual: o1
    Types: A
Individual: o2
    Types: B
Individual: o3
    Types: C
Individual: o4
    Types: D
```

edit:

	A	B	C	D
o1	+	?	+	?
o2	?	+	+	-
o3	?	?	+	?
o4	+	-	+	+

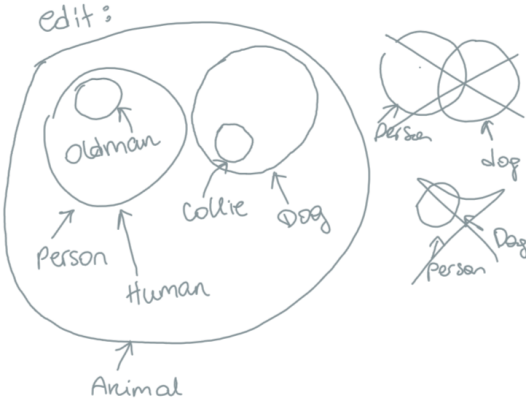
kann sein, dass sich A&B überschneiden



Task 6

```
Class: PERSON
    EquivalentTo: HUMAN
Class: ANIMAL
Class: DOG
    SubClassOf:ANIMAL
    DisjointWith: OLDMAN
Class: HUMAN
    EquivalentTo: PERSON
    SubClassOf: ANIMAL
Class: OLDMAN
    SubClassOf: PERSON
    DisjointWith: DOG
Class: COLLIE
    SubClassOf: DOG
Individual: flipper
    Types: ANIMAL
Individual: rex
    Types: DOG
Individual: mary
    Types: HUMAN
Individual: john
    Types: PERSON
Individual: jim
    Types: OLDMAN
Individual: lassie
    Types: COLLIE
```

	PERSON	ANIMAL	DOG	HUMAN	OLD-MAN	COL-LIE
FLIPPER	?	+	?	?	?	?
REX	-	+	+	-	-	?
MARY	+	+	-	+	?	-
JOHN	+	+	-	+	?	-
JIM	+	+	-	+	+	-
LASSIE	-	+	+	-	-	+



Task 7

```
Individual: o1
  Types: C
Individual: o2
  Facts: p o1
  Types: C
Individual: o3
  Types: C1
Individual: o4
  Facts: p o1, p o5
Individual: o5
  Types: not C, p exactly 0
Class: C
Class: C1
  SubclassOf: p some C
Class: C2
  EquivalentTo: p some C
Class: C3
  EquivalentTo: p only C
```

psome C ponly C

	C	C1	C2	C3
o1	+	?	?	?
o2	+	?	+	?
o3	?	+	+	?
o4	?	?	+	-
o5	-	-	-	+

weil leere Menge ✓ gilt

Task 8

```
Class: Woman
Class: Teacher
Class: Artist
Class: JohnsFriends
    EquivalentTo: MarysFriends, {Bob , Mary}
Class: MarysFriends
    EquivalentTo: JohnsFriends, {MsKeller , Mueller}
Individual: Bob
    Types: Teacher, not (Woman)
Individual: Mary
    Types: Artist, Woman
Individual: MsKeller
    Types: Woman
Individual: Mueller
```

{ Bob , Mary } ⇒ 2 Personen
= { MsK. , Mueller }

Mary → Woman (not Man)
MsK. → Woman (not Man)
} 1 Pers.

Bob → not Woman
Mueller → ?
} 1 Pers.

	JohnsFriends	MarysFriends	Teacher	Artist	Woman
Bob	+	+	+	?	-
Mary	+	+	?	+	+
MsKeller	+	+	?	?	+
Mueller	+	+	?	?	?

Part II: Subsumption Checking (2 points)

Task 9

Given is an OWL ontology in Manchester Syntax.

```
ObjectProperty: eats
Class: Cereals
Class: Egg
Class: Fish
Class: Fruits
Class: Meat
Class: Milk
Class: Vegetables
DisjointClasses:
    Cereals,Egg,Fish,Fruits,Meat,Milk,Vegetables
Class: Person
    SubClassOf: eats some Food
    DisjointWith: Food
Class: Food
    EquivalentTo: Cereals or Egg or Fish or Fruits or Meat or Milk or Vegetables
    DisjointWith: Person
Class: Carnivor
    EquivalentTo: Person and (eats only (Fish or Meat))
Class: Frutarier
    EquivalentTo: Person and (eats only Fruits)
Class: Vegan
    EquivalentTo: Person and (eats only (Cereals or Fruits or Vegetables))
Class: Vegetarian
    EquivalentTo: Person and (eats only (not (Fish or Meat)))
Class: xPerson
    SubClassOf: Person and (eats some Fruits)
Class: yPerson
    EquivalentTo: Person and (eats some Fruits)
Class: zPerson
    EquivalentTo: (eats some Cereals) and (eats some Fruits) and (eats some Vegetables)
    SubClassOf: Person
```

Your task is to find the pairwise semantic relationships of classes `Carnivor, Frutarier, Vegan, Vegetarian, zPerson, xPerson, yPerson`. Indicate the relationships in the cells in the table below using the following symbols:

- subclass of ⊑
- equivalent to ≡
- superclass of ⊒
- disjoint with ⊑¬
- none of the above leave empty

As an example, the semantic relationship *xPerson is subclass of yPerson* is already indicated in the table.

	yPerson	xPerson	zPerson	Vegetarian	Vegan	Frutarier	Carnivor
Carnivor	⊑¬	⊑¬	⊑¬	⊑¬	⊑¬	⊑¬	≡
Frutarier	⊑ f sub y	⊑	⊑ f sub z	⊑ (f sub v)	⊑ (f sub v)	≡	
Vegan	⊒ v super y	⊑	⊑ v sub z	⊒ vegan s v	≡		
Vegetar-ian	⊒ v super y	⊑	⊑ v sub z	≡			
zPerson	⊒ z super y	⊒ z super x	≡				
xPerson	⊑ x sub y	≡					
yPerson	≡						

Edit:

	yPerson	xPerson	zPerson	Vegetarian	Vegan	Frutarier	Carnivor
Carnivor							
Frutarier							
Vegan							
Vegetarian							
zPerson							
xPerson							
yPerson