SAARLAND UNIVERSITY ARTIFICIAL INTELLIGENCE
Dr. Álvaro Torralba, Prof. Wolfgang Wahlster
Dr. Cosmina Croitoru, Daniel Gnad, Marcel Steinmetz
Yannick Körber, Michael Barz
Christian Bohnenberger, Sören Bund-Becker, Sophian Guidara,
Alexander Rath, Khansa Rekik, Julia Wichlacz, Anna Wilhelm

Exercise Sheet 10.
Solutions due Tuesday, **July 10**, 16:00 – 16:15, in the lecture hall.[1]

Exercise 36 - OWL. (4.5 Points)

(a) Consider the provided ontology (RDF/XML) `persons.xml`.
Extract all triples defined in this ontology and list them in the following format:

```
(ai:Person rdf:type  owl:Class)
(ai:related rdfs:range  ai:Person)
(ai:related rdfs:domain  ai:Person)
(ai:related rdf:type  owl:ReflexiveProperty)
...
```

You might notice that there is a triple indicating Susanne is related to Georg and Georg is related to Peter. But no triple indicating Susanne is related to Peter. Think about reasoning: ai:related is a reflexive, transitive, symmetric relation. Your task is to write a production rule for a reasoner that would create the missing **transitive** triples. You can assume that only new triples are added by your production rule. You can use A-Z and X1-Xn as variables. Hint: Don't forget to check range and domain against their types!

Example syntax for (a senseless!) production rule:

```
R1: (X1 rdf:type ai:Person) &
(X1 rdf:type any:Human) ->
    ADD (X1 rdf:type any:HumanPerson)
```

(b) What is the difference between `owl:ObjectProperty` and `owl:DatatypeProperty`? Explain the difference and give an example for each type of property (use triples).

[1] Solutions in paper form only, and solution submission only at the stated time at the stated place. At most 3 authors per solution. All authors must be in the same tutorial group. All sheets of your solution must be stapled together. At the top of the first sheet, you must write the names of the authors and the name of your tutor. Your solution must be placed into the correct box for your tutorial group. Also, you should write the solutions of the exercises in order, in particular, do not interleave parts of different exercises otherwise we may oversee part of your solution. Please, don't use red ink, preferably use a black or blue pen instead. If you don't comply with these rules, 3 points will be subtracted from your score for this sheet.

Exercise 37 - Production Rules. (5.5 Points)

The following production system is designed to calculate the first seven Fibonacci-Numbers:

```
WM: (START) (P 0) (Q 1) (END 5)

R1: (START) & (P X1) & (Q X2) →
    DELETE((START)), ADD((CALC)),
    ADD((COUNT 0)), ADD((TEMP 0)),
    PRINT(X1), PRINT(X2)

R2: (END X1) & (COUNT X1) →
    STOP

R3: (CALC) & (P X1) & (Q X2) & (TEMP X3) →
    DELETE((CALC)), ADD((PRINTOUT)),
    REPLACE(X3, PLUS(X1,X2), (TEMP X3))

R4: (PRINTOUT) & (TEMP X3) & (COUNT X4) →
    DELETE((PRINTOUT)), ADD((SHIFT)),
    PRINT(X3), REPLACE(X4, SUCC(X4), (COUNT X4))

R5: (SHIFT) & (P X1) & (Q X2) & (TEMP X3) →
    DELETE((SHIFT)),  REPLACE(X1, X2, (P X1)),
    REPLACE(X2, X3, (Q X2))
```

Here, `PLUS(X,Y)` calculates the sum of X and Y, `SUCC(X)` calculates the successor of X (in other words: $X + 1$) and `PRINT(X)` prints out the value of X on the console.

(a) Calculate the output of this production system until no more rules are left to fire. For each cycle, write down the content of the WM, the firing rule, the instantiation of the firing rule, the output (if there is any) and the content of the resulting WM.

Example:
WM: (START) (P 0) (Q 1) (END 5)
Firing rule: R1
Instantiation: [R1 ((START) (P 0) (Q 1))]
Resulting WM: (TEMP 0) (COUNT 0) (CALC) (P 0) (Q 1) (END 5)

(b) Obviously there is an error in the rules above: Rule R2 is never going to fire. Change rule R5 in such a way that the cycle R3, R4, R5 is repeated until R2 fires.

(c) The rules above are independent of the chosen conflict resolution strategy. Why?