Master in Computational Logic & Mestrado Integrado em Engenharia Informática
Universidade Nova de Lisboa

Knowledge Representation and Reasoning Systems

First Test – Closed Book – 2h00m

23rd October 2017

Group 1 [3 val.]

1) Transform the following sentences into clausal form:

$$\begin{array}{ll} \textbf{S1} & \exists x \forall y \exists z (R(x,y) \supset P(y,z)) \\ \textbf{S2} & \neg \forall x \forall y (Q(x,y) \vee \exists z Q(y,z)) \\ \textbf{S3} & \forall x \forall y ((P(x) \vee Q(y)) \supset (P(y) \wedge Q(x))) \end{array}$$

2) Show by resolution that clauses **C1-C3** below entail $\neg R(f(1), 1)$.

$$\begin{array}{ll} \textbf{C1} & [R(1, f(1))] \\ \textbf{C2} & [\neg R(y,x), P(f(x))] \\ \textbf{C3} & [\neg P(x), \neg R(y,x)] \end{array}$$

Answer:

1)

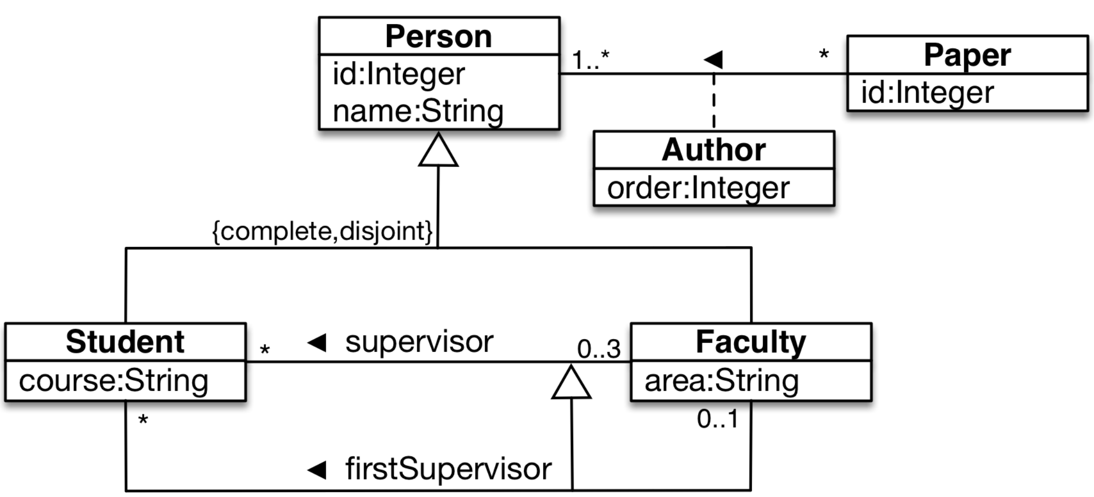
$$\begin{array}{ll} \textbf{S1} & [\neg R(a,y), P(y, f(y))] \\ \textbf{S2} & [\neg Q(a,b)], [\neg Q(b,z)] \\ \textbf{S3} & [\neg P(x), P(y)], [\neg P(x), Q(x)], [\neg Q(y), P(y)], [\neg Q(y), Q(x)] \end{array}$$

2) We have **C4** as $[R(f(1), 1)]$.

1. $[\neg P(f(1))]$ from $C1$ and $C3$, $x/f(1), y/1$
2. $[P(f(1))]$ from $C2$ and $C4$, $x/1, y/f(1)$
3. $[]$ from 1. and 2.

Group 2 [7 val.]

Consider the following UML class diagram representing information about authors of scientific papers in the university:



1) Translate the UML class diagram into an appropriate Description Logic.

2) Express in Description Logic (in the fragment you think is more appropriate) the following concepts:

   i. Faculty members that supervise at least one student
   ii. Unsupervised students
   iii. Faculty members that are not first supervisors
   iv. Students that have co-authored only one paper
   v. Papers whose student co-authors are supervised
   vi. Supervisors whose students have all written more than one paper

3) Indicate which of these concepts can be expressed in $\mathcal{ALC}$.

Answer:

1)

$$
\begin{array}{rcl}
\exists id & \sqsubseteq & Person \\
\exists id^- & \sqsubseteq & Integer \\
Person & \sqsubseteq & \exists id \sqcap (\leq 1 id) \\
\exists name & \sqsubseteq & Person \\
\exists name^- & \sqsubseteq & String \\
Person & \sqsubseteq & \exists name \sqcap (\leq 1 name) \\
\exists pid & \sqsubseteq & Paper \\
\exists pid^- & \sqsubseteq & Integer \\
Paper & \sqsubseteq & \exists pid \sqcap (\leq 1 pid) \\
\exists order & \sqsubseteq & Author \\
\exists order^- & \sqsubseteq & Integer \\
Author & \sqsubseteq & \exists order \sqcap (\leq 1 order) \\
\exists course & \sqsubseteq & Student \\
\exists course^- & \sqsubseteq & String \\
Student & \sqsubseteq & \exists course \sqcap (\leq 1 course) \\
\exists area & \sqsubseteq & Faculty \\
\exists area^- & \sqsubseteq & String \\
Faculty & \sqsubseteq & \exists area \sqcap (\leq 1 area)
\end{array}
$$

$$
\begin{array}{rcl}
\exists supervisor & \sqsubseteq & Faculty \\
\exists supervisor^- & \sqsubseteq & Student \\
Student & \sqsubseteq & (\leq 3 supervisor^-) \\
\exists firstSupervisor & \sqsubseteq & Faculty \\
\exists firstSupervisor^- & \sqsubseteq & Student \\
Student & \sqsubseteq & (\leq 1 firstSupervisor^-) \\
firstSupervisor & \sqsubseteq & supervisor \\
\exists authorPerson & \sqsubseteq & Author \\
\exists authorPerson^- & \sqsubseteq & Person \\
\exists authorOfPaper & \sqsubseteq & Author \\
\exists authorOfPaper^- & \sqsubseteq & Paper \\
Author & \sqsubseteq & \exists authorPerson \sqcap \exists authorOfPaper \sqcap \\
 & & (\leq 1 authorPerson) \sqcap (\leq 1 authorOfPaper) \\
Paper & \sqsubseteq & (\geq 1 authorOfPaper^-) \\
Student & \sqsubseteq & Person \\
Faculty & \sqsubseteq & Person \\
Student & \sqsubseteq & \neg Faculty \\
Person & \sqsubseteq & Student \sqcup Faculty
\end{array}
$$

2)

   i. $Faculty \sqcap \exists supervisor$
   ii. $Student \sqcap \neg \exists supervisor^-$
   iii. $Faculty \sqcap \neg \exists firstSupervisor$
   iv. $Student \sqcap \exists authorPerson^- \sqcap (\leq 1 authorPerson^-)$
   v. $Paper \sqcap \forall paperOfAuthor^-.\forall authorPerson.(\neg Student \sqcup \exists supervisor^-)$
   vi. $Faculty \sqcap \forall supervisor.(\geq 2 authorPerson^-)$

3) i. and iii.

Group 3 [7 val.]

The tableau algorithm for $\mathcal{ALC}$ shown in the class can be extended to deal with transitive roles by adding the rule: $\rightarrow_{tr}$ $(\forall r.D)(x), r(x,y) \in \mathcal{A}$ and r is transitive, then $\mathcal{A} := \mathcal{A} \cup \{\forall r.D(y)\}$.

1) Using tableau, showing every step, determine the satisfiability of

$$\exists r.A \sqcap \forall r.B \sqcap \neg(\exists s.A \sqcup \forall s.(\forall s.B \sqcap \forall s.\neg A))$$

where s is a transitive role.

2) If the concept is satisfiable, construct a model for it in which $a \in A^I$.

Answer:
1) We have to transform into negation normal form first:

$$C_0 = \exists r.A \sqcap \forall r.B \sqcap \forall s.\neg A \sqcap \exists s.(\exists s.\neg B \sqcup \exists s.A)$$

We apply the tableaux algorithm starting with C_0:

$$\begin{array}{lll}
 & \mathcal{A}_0 = \{C_0(x_0)\} & \\
\rightarrow_{\sqcap}^{*} & \mathcal{A}_1 = \mathcal{A}_0 \cup \{\exists r.A(x_0), \forall r.B(x_0), \forall s.\neg A(x_0), \exists s.(\exists s.\neg B \sqcup \exists s.A)(x_0)\} & \\
\rightarrow_{\exists} & \mathcal{A}_2 = \mathcal{A}_1 \cup \{r(x_0,x_1), A(x_1)\} & \\
\rightarrow_{\forall} & \mathcal{A}_3 = \mathcal{A}_2 \cup \{B(x_1)\} & \\
\rightarrow_{\exists} & \mathcal{A}_4 = \mathcal{A}_3 \cup \{s(x_0,x_2), (\exists s.\neg B \sqcup \exists s.A)(x_2)\} & \\
\rightarrow_{\forall} & \mathcal{A}_5 = \mathcal{A}_4 \cup \{\neg A(x_2)\} & \\
\rightarrow_{tr} & \mathcal{A}_6 = \mathcal{A}_5 \cup \{\forall s.\neg A(x_2)\} & \\
\rightarrow_{\sqcup} & \mathcal{A}_7 = \mathcal{A}_6 \cup \{\exists s.\neg B(x_2)\} & \qquad \mathcal{A}_{7'} = \mathcal{A}_6 \cup \{\exists s.A(x_2)\} \\
\rightarrow_{\exists} & \mathcal{A}_8 = \mathcal{A}_7 \cup \{s(x_2,x_3), \neg B(x_3)\} & \rightarrow_{\exists} \quad \mathcal{A}_{8'} = \mathcal{A}_{7'} \cup \{s(x_2,x_4), A(x_4)\} \\
\rightarrow_{\forall} & \mathcal{A}_9 = \mathcal{A}_8 \cup \{\neg A(x_3)\} & \rightarrow_{\forall} \quad \mathcal{A}_{9'} = \mathcal{A}_{8'} \cup \{\neg A(x_4)\} \times \\
\rightarrow_{tr} & \mathcal{A}_{10} = \mathcal{A}_9 \cup \{\forall s.\neg A(x_3)\} \checkmark &
\end{array}$$

Since $\mathcal{A}_{10}$ is complete and clash-free, C_0 is satisfiable.
2) Consider $\Delta = \{a,b,c,d\}$. Then, $A^I = \{a\}$, $B^I = \{a\}$, $r^I = \{(b,a)\}$ and $s^I = \{(b,c),(c,d)\}$.

Group 4 [3 val.]

Answer the following questions in a ***short*** and ***concise*** way.

1) What is the point of *Description Logics* and other ontology languages? Why not simply use *First-Order Logic*?

2) What are the benefits of using an ontology at runtime?

3) Description Logics only allow the usage of unary and binary predicates. But sometimes we want to model ternary relationships. How can we overcome this problem?

Answer:
1) First-Order Logic in only semi-decidable. Description Logics and decidable fragments of First-Order Logic specifically tailored to represent ontologies. Many in the family of Description Logics are even tractable fragments of First-Order Logic.
2) The benefits of using an Ontology at runtime are twofold: on the one hand it can be used to check for consistency as the database is modified; on the other hand, it can be used for reasoning at query time, e.g. to complete the answers to queries e.g. when the data in the database is incomplete. Additionally, ontologies can also be used as a means to integrate knowledge from different sources.
3) This can be overcome by use of reification, which is achieved through the creation of a new concept that represents the elements in the ternary relationship, and three binary predicates (roles) each relating the new concept and each of the concepts participating in the original ternary relationship.

Group 5 [Bonus: up to 2 val.]

Sketch a proof that $\mathcal{ALC}$ is less expressive than FOL_{bin}.

Answer: $\mathcal{ALC}$ has the tree model property. The FOL_{bin} formula $\forall x.P(x,x)$ cannot be translated into $\mathcal{ALC}$ i.e. there is no $\mathcal{ALC}$ TBox $\mathcal{T}$ such that

$$\mathcal{I} \models_{\mathcal{ALC}} \mathcal{T} \text{ if and only if } \mathcal{I} \models_{FOL} \forall x.P(x,x)$$

because models of $\forall x.P(x,x)$ are not tree shaped.
A consequence of the above fact, and of the fact that $\mathcal{ALC}$ can be expressed in FOL_{bin} (as shown in class) is that $\mathcal{ALC}$ is strictly less expressive than FOL_{bin}.