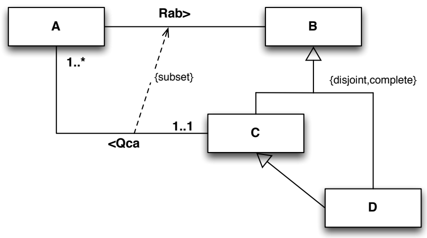
SAPIENZA Università di Roma – MSc. in Engineering in Computer Science

Formal Methods – July 9, 2014

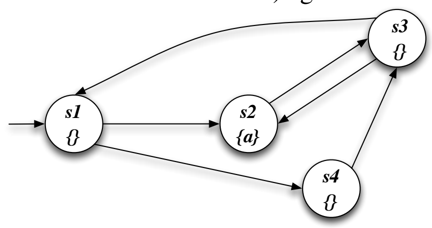
(Time to complete the test: 2 hours)

Exercise 1. Consider the following UML class diagram.



1. Express it in FOL.
2. Express it in *DL-Lite*$_{\mathcal{A}}$, highlighting the parts that are not expressible.
3. Given the ABox $\mathcal{A} = \{A(c)\}$, check formally if the diagram, as expressed in *DL-Lite*$_{\mathcal{A}}$, is consistent with $\mathcal{A}$.

Exercise 2. Model check the Mu-Calculus formula $\nu X.\mu Y.((a \wedge \langle next\rangle X) \vee \langle next\rangle Y)$ and the CTL formula $AG(EX(EXa \vee EXEXa))$ (showing its translation in Mu-Calculus) against the following transition system:



Exercise 3. Consider the following predicates: $Person(x)$, saying that x is a person, $Appetizer(x)$, saying that x is a gourmet appetizer, $MainCourse(x)$, saying that x is a gourmet main course, and $likes(x, y)$, saying that x likes y. Express in FOL the following boolean queries, stating which ones are CQs (do not use abbreviations for cardinalities):

1. *There exists a person who likes an appetizer and a main course.*
2. *There exists a person who likes two appetizers and a main course.*
3. *There exists a person who likes exactly one main course.*
4. *There exists a person who likes all appetizers.*
5. *Return the persons who like no main courses.*
6. *Return the pairs of persons such that the first likes all appetizers that the second likes, but not viceversa.*

Exercise 4. Compute the certain answers to the CQ $q(x) \leftarrow Person(x), Person(y), Likes(x, z), Likes(z, y)$ over the following incomplete database (naive tables), and discuss how you obtained the result:

$Person$

$name$
Smith
$null_1$
Brown
$null_2$
Green

$Likes$

ls	ld
Smith	$null_1$
$null_1$	Brown
Brown	$null_2$
Green	White
White	$null_2$
$null_2$	White

Exercise 5. Compute the weakest precondition for getting $\{x = 100\}$ by executing the following program:

```
x := 90 - y;
if (x = 0) then {
  if (y > 10) then
    x := y - x;
  else x := 10 - x
}
x := x + y;
y := 10 + y
```