CT331 Assignment 3

Declarative Programming with Prolog

Andrew Reid East

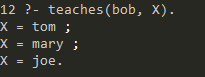
16280042

C:\Users\deskase\AppData\Local\Microsoft\Windows\INetCache\Content.Word\fa-github-square.png <https://github.com/reideast/ct331_assignment3>

# Question 1

|  |  |  |
| --- | --- | --- |
| 1. | sunny AND warm | True |
| 2. | sunny AND cold | False |
| 3. | sunny OR cold | True |
| 4. | (sunny OR cold) AND warm | True |
| 5. | happy XOR sunny | False |
| 6. | warm XOR (NOT happy) | True |
| 7. | early NAND happy | False |
| 8. | (late NOR (NOT early)) AND (windy OR (NOT warm)) | True |
| 9. | (cloudy AND windy) AND (warm AND early) | False |
| 10. | (cloudy AND windy) XOR (warm OR early) | True |

# Question 2

1. teaches(INSTRUCTOR, STUDENT) :- instructs(INSTRUCTOR, CLASS),  
    takes(STUDENT, CLASS).
2. 
3. 
4. False  
     
   Prolog begins matching teaches(ann, joe). line-by-line through the database. It will not match anything until it finds the line which defines the rule teaches, which has two un-unified variables, INSTRUCTOR and STUDENT, which it unifies to ann and joe. It then examines the remainder of the rule, which begins with the relation instructs. One argument to instructs is already bound, but the variable CLASS is not, and it remains ununified. The antecedent proceeds with a conjunction to a relation, takes. All the arguments to takes are already unified. It encounters the full stop character, and has finished processing this line.  
     
   Prolog will then proceed to backtrack through the database, attempting to match lines that will satisfy the unbound variables. Looking for instructs(ann, CLASS), it finds the fact instructs(ann, ct345), and binds ct345 to CLASS. It must then backtrack again to find a fact that makes the entire AND clause true. With all the variables unified, it is looking for the precise fact: takes(joe, ct345). It will loop through the database, but will not find this fact. Therefore, it will undo its last unification, ct345 to class. Then, it continues looping to look for new information that matches instructs(ann, CLASS). It reaches the end of the database without finding anything, and because of negation-as-failure, the query is false.
5. classmates(A, B) :- takes(A, CLASS), takes(B, CLASS).  
   classmates(mary, joe). True  
   classmates(tom, mary). True  
   classmates(mary, tom). True  
   classmates(joe, bob). False  
   classmates(bob, joe). False

# Question 3