Master MLDM - First year
Introduction to Artificial Intelligence - Exam on Prolog

Maximum time allocated: 2h00 - No documents allowed. TAKE CARE: any cheating will be severely punished and will lead to a formal complaint to the disciplinary council of the university.

1 Proof tree (5 points)

Consider the Prolog program below:

```
p1(X,Y) :- p2(X), p3(Y).
p1(X,Y) :- p4(X,_), p5(_,Y).

p2(X) :- s(X),t(X), u(X).

p3(aa).  s(a).  s(b).  t(1).  t(2).  u(99).  w(98).

p4(X,Z) :- u(X), t(Z).
p5(X,Y) :- w(X),t(Y).
```

1. Draw the proof tree of the resolution of the goal: `?- p1(A,B).` and give all the solutions for this goal.
2. Suppose we put a cut between `p2(X)` and `p3(Y)` in the first clause of the program and we replace the fact `t(1)` by `t(1):- !.` Show, on the tree you built at the previous question, which branches are pruned during the resolution of the goal: `?- p1(A,B).` and give again all the solutions for this goal.

2 Lists (5 points)

Define the following predicates that specify some relationships between lists.

1. `mylast/2` where `mylast(X,L)` is true if X is the last element of the list L.
2. `element_k/2` where `element_k(X,L,K)` is true if the value X is in position K in the list L.
3. `duplicate/2` where `duplicate(L1,L2)` is true if the elements of L1 are duplicated twice in the list L2.
4. `compress/2` where `compress(L1,L2)` is true if L2 is equal to L1 without any consecutive duplicated value.
5. `myreverse/2` where `myreverse(L1,L2)` is true if L2 is the list L1 reversed.

3 DCG (5 points)

Consider the formal grammar of arithmetic expressions :

```
exp => exp ’+’ exp1
exp => exp ’-’ exp1
exp => exp1
exp1 => exp1 ’*’ exp2
exp1 => exp1 ’/’ exp2
exp1 => exp2
exp2 => ’(’ exp ’)’
exp2 => ’0’
exp2 => ’1’
exp2 => ’2’
exp2 => ’3’
```

1. Convert this grammar to a DCG that can be used to prove if an arithmetic expression is syntactically correct and to evaluate it.
2. Write the Prolog goal you have to run to prove that the arithmetic expression 2 * (2 + 1) - 1 is syntactically correct and to evaluate it.
3. More generally, give the Prolog clause generated from the following DCG rule after loading it into the Prolog workspace: `p1(X,Y) --> p2(X,Z),[a,b],p3(Z),[c],p4(Z,Y).`

4 Knowledge base modeling and querying (5 points)

Convert the following information into a Prolog program (to choose the appropriate Prolog representation, you should look at the goals we then want to prove):

When a person P passes the exam of the theoretical part of a course C and the project part of the same course, then he passes the global exam for this course. When a person P1 is a friend of a person P2 that passes the global exam for a course C, that person P1 also passes the global exam for the course C. To prove that a person P passes the theoretical part of a course C we have to prove that a teacher T delivers this course and P attends this course and he gets a mark M greater than 10 at the theoretical part of the exam of this course. Another way to prove that a person P passes the theoretical part of a course C is that either that person got an equivalence for this course or this person has a PhD. If a course C requires knowledge of a programming language of type T (functionnal programming, logic programming, etc) and a person P masters a programming language L of type T and writes some code Co using this language L and gets a mark M for this code Co greater than 10, then this person passes the project part of the course C. To prove that a person P masters a language L of type T we have to prove that the language L is of type T and that this person has learnt this language L. Another way to prove that a person P masters a language L of type T is to prove that this person is a friend of ironman and ironman masters this language L of type T. To prove that a person P has learnt a programming language L we have to prove that a teacher T delivers a course C, and that the language L is part of course C and P attends to the course C. Prolog is a programming language of type logic programming. Lisp is a programming language of type functional programming. Caml is a programming language of type functional programming. Python is a programming language of type object oriented programming. The Machine Learning course requires a knowledge of an object oriented programming language. Mary has a PhD. John is a friend of Mary. John has learnt the Python language. Paul passes his global exam on Machine Learning. Emily delivers a course on Machine Learning. John attends the course on Machine Learning. John gets the mark 15 to the course Machine Learning.

Considering the Prolog program you just wrote, convert the following english queries into Prolog goals (do not try to solve those goals, this is not the question):

1. Who passes which exam?
2. Who passes the project part of which exam?
3. who masters which language of type object oriented programming?
4. Who delivers a course on Machine Learning?