

(Attempt 2 questions out of 3)

Question 1.

- (a) Suppose a Prolog program contains the definition of a predicate `enrolled` such that `enrolled(S, N)` means that student `S` is enrolled in course `N`. Formulate queries that correspond to the following questions:

- (i) Is Sophia enrolled in course `CS_125`?
- (ii) Who is enrolled in course `CS_125`?
- (iii) In which courses are Sophia and Julian enrolled?
- (iv) Is it true that neither Sophia nor Julian are enrolled in course `CS_125`?
- (v) Is it true that nobody is enrolled in course `CS_125`?

[10 Marks]

- (b) Referring to the predicate `enrolled` in (a), determine the list of students enrolled in both course `CS_125` and course `MAM111`.

You may use Prolog's built-in predicate `findall/3`.

[8 Marks]

- (c) Which of the following terms are unifiable? In case the terms are unifiable, compute their most general unifier, otherwise explain why they are not unifiable.

- (i) `f(X, g(X, Y), Y)` `f(h(a, U), g(Z, b), a)`
- (ii) `f(X, g(X, Y), Y)` `f(h(a, U), g(Z, Z), h(U, a))`

[7 Marks]

Question 2.

- (a) Say for the following queries whether they will fail or succeed or cause an error. If the query fails or causes an error, explain why this is the case. If the query succeeds, give the answers that Prolog will return.

- (i) `?- X = 2 * 3.`
- (ii) `?- X == 2 * 3.`
- (iii) `?- X is 2 * 3.`
- (iv) `?- 6 = 2 * X.`
- (v) `?- 6 is 2 * X.`
- (vi) `?- [H,H|T] = [a,a,b]`
- (vii) `?- [L|L] = [A,b,b]`

[7 Marks]

- (b) Define a predicate `first_elements(LL,L)` which given a list of lists `LL` and a list `L` succeeds if all lists in `LL` are non-empty and `L` contains the first elements of the lists in `LL`.

For example, the queries

```
?- first_elements([[1,2,3],[a,b],[4,5]],L).
```

and

```
?- first_elements([[1,2,3],[],[4,5]],L).
```

should return the answers `L = [1,a,4]` and `No` respectively.

[6 Marks]

- (c) Consider the following program:

```
s(L, [H|L]).  
s(L, [H|T]) :- s(L,T).
```

- (i) Introduce anonymous variables at all appropriate places such that the behaviour of the program is not changed.
- (ii) Describe informally the meaning of the predicate defined in the above program.
- (iii) Draw the derivation trees for the following queries and give all answers which would be obtained by repeatedly pressing “;”:

```
?- s([a], [A]).  
?- s([a], [A,B,C]).  
?- s(L, [a,b]).
```

[12 Marks]

Question 3.

- (a) (i) Explain the general effect of the cut predicate on Prolog's search trees.
(ii) How is the `not` predicate (negation as failure) in Prolog defined? Give a definition without using “;”.
[8 Marks]
- (b) Draw the resolution trees for the following three queries, and for each query predict the subsequent answers that Prolog will return.
(i) `?- not(x=1).`
(ii) `?- not(X=1).`
(iii) `?- X=1, not(not(X=1)).`
[9 Marks]
- (c) Define a predicate `count_diff_a(L,N)` which holds if `L` is a list and `N` is the number of elements in `L` different from `a`.
For example, the queries `?- count_diff_a([a,b,c,b,a], N).` and `?- count_diff_a([a,a,a], N).` should yield the answers `N = 3` and `N = 0` respectively.
[8 Marks]