



uOttawa

L'Université canadienne  
Canada's university

**Faculty of Engineering  
School of Information Technology and Engineering**

**CSI 2165: Prolog Concepts Laboratory**

**Final Exam**

**Examiner:**  
Diana Inkpen

**December 20, 2005, 9:30**  
Total marks: **48**  
Duration: **3** hours  
Total number of pages: **12**

<b>Last Name:</b>		<b>First Name:</b>	
<b>Student Number:</b>			

**Important Regulations:**

1. No calculators are allowed.
2. A student identification cards (or another photo ID and signature) is required.
3. An attendance sheet shall be circulated and should be signed by each student.
4. Please answer all questions on this paper, in the indicated spaces.
5. Use both sides of these sheets if you need some extra space for rough work.
6. At the end of the exam, when time is up:
  - a. Stop working and turn your exam upside down.
  - b. Remain silent.
  - c. Do not move or speak until all exams have been picked up, and a TA or the Professor gives the go-ahead to leave.

A	B	C	D	E	F	Total
5	10	5	20	4	4	48

**A. [5 marks]**

Which of the following are syntactically correct Prolog objects? If yes, identify the types of object they are (atom, number, variable, structure). If not, use a “No” as an answer.

	Term	Type
1.	xY	
2.	_x	
3.	50.1	
4.	F([1, a, 3])	
5.	likes(john, mary)	

**B. [10 marks]**

Show the variable instantiations (the values of X and Y) if the matching between the term in the first column and the term in the second column succeeds. Use “No” otherwise.

	Term1	Term2	Instantiations	
Example	[1, 2, 3]	[X   Y]	X = 1	Y = [2, 3]
1.	[a, [c]]	[X   Y]	X =	Y =
2.	[[a]]	[X , Y]	X =	Y =
3.	[f(a)]	[X   Y]	X =	Y =
4.	[f(a), f(b), c]	[X   Y]	X =	Y =
5.	[[a], a]	[X   Y]	X =	Y =
6.	[f(Y), 1]	[X   Y]	X =	Y =
7.	f([1, 2, 3])	f([X   Y])	X =	Y =
8.	f(X, g(a))	g(X, Y)	X =	Y =
9.	f(X, g(a))	f(g(Y),Y)	X =	Y =
10.	f(X, g([1, 2]))	f(g(a),Y)	X =	Y =

C. [5marks]

Indicate which one of the following choices is correct. (Note: Only **one** of the answers is correct.) **Use “Answer Sheet: Part C” to indicate your answer.**

What is the SWI-Prolog answer to the following queries?

1. `?- f(a,b) =.. [X,Y].`

- a) `X=a, Y=b`
- b) `X=f, Y=a`
- c) `X=f, Y=b`
- d) `No`

2. `?- assert(a(2)),assert(a(3)),retract(a(X)),write(X),write(' '),fail.`

- a) `2`  
`No`
- b) `2 3`  
`Yes`
- c) `2 3`  
`No`
- d) `3 2`  
`Yes`

3. `?- X is 5, fail.`

- a) `X=5`  
`Yes`
- b) `No`
- c) `Yes`
- d) `none of the above`

4. `assert(a(1)),assert(a(2)),retract(a(X)),!,write(X), write(' '),fail.`

a) 1 2  
Yes

b) 1 2  
No

c) 2 1  
No

d) 1  
No

5. Given the following code (DCG):

```
np --> adj, noun.  
adj --> [red].  
adj --> [blue].  
noun --> [book].
```

The Prolog translation of these 4 DCG clauses is:

```
np(A, B):- adj(A, C), noun(C, B).  
adj([red|A], A).  
adj([blue|A], A).  
noun([book|A], A).
```

What is the answer to the following query?

```
?- np([red, book],R).
```

a) No

b) `R = []`  
Yes

c) `R = [red, book]`  
Yes

d) None of the above

**Answer Sheet: Part C**

<b>Question</b>	<b>Answer</b>
<b>1.</b>	
<b>2.</b>	
<b>3.</b>	
<b>4.</b>	
<b>5.</b>	

**D. [20 marks]**

Read the following programs and provide the answer to the query (only the first answer that SWI-Prolog would produce).

1. Program	<pre>prog_1(X, [X, _]). prog_1(X, [_ L]) :- prog_1(X, L).</pre>
Query	<pre>?- prog_1(X, [a,b,c,d]).</pre>
Answer	<pre>X =</pre>

2. Program	<pre>prog_2(L,R) :- acc(L, 0, R). acc([], A, A). acc([H T], A, R) :- A1 is A+H, acc(T, A1, R).</pre>
Query	<pre>prog_2([1,2,3,4,5], R).</pre>
Answer	<pre>R =</pre>

<p>3.</p> <p>Program</p>	<pre>prog_3([X _], 1, X). prog_3([_ Xs], K, Y):- K &gt; 1,                       K1 is K-1,                       prog_3(Xs,K1,Y).</pre>
<p>Query</p>	<pre>?- prog_3([a,b,c,d],3,R).</pre>
<p>Answer</p>	<p>R =</p>

<p>4.</p> <p>Program</p>	<pre>prog_4([H _], H):- number(H), !. prog_4([_ T], S):- prog_4(T,S).</pre>
<p>Query</p>	<pre>?- prog_4([a,b,4,d,3,a(b,c)],S).</pre>
<p>Answer</p>	<p>S =</p>

<p>5.</p> <p>Program</p>	<pre>prog_5(nil, []). prog_5(t(X,nil,nil), [X]):- !. prog_5(t(X,L,R), Y):- prog_5(L,NewL),                       prog_5(R,NewR),                       append([X NewL],NewR,Y).</pre>
<p>Query</p>	<pre>?- prog_5( t(1, t(2, t(4, nil, nil),               t(5, nil, nil)),t(3, nil, nil)), Z).</pre>
<p>Answer</p>	<p>Z =</p>

<p>6.</p> <p>Program</p>	<pre>p1(1). p2(X) :- p1(X),!,fail. p2(_). p3(X) :- p1(X),fail,!. p3(_).</pre>
<p>Query</p>	<pre>?- p2(1),p3(2).</pre>
<p>Answer</p>	



7. Program	<pre>prog_7(t(Node,_,nil), Node). prog_7(t(_,LT,_), M):- LT \== nil, prog_7(LT, M).</pre>
Query	<pre>?- prog_7(t(5, t(1, t(2, nil, nil), t(4, nil, nil)),            t(9, nil, nil)), M).</pre>
Answer	M =

8. Program	<pre>prog_8(S) :- pp(S, []). pp --&gt; d. pp --&gt; d, [ ], pp. d --&gt; [X], {member(X, [0,1,2,3,4,5,6,7,8,9])}.  The 3 DCG clauses above are translated in Prolog as: pp(A, B):- d(A, B). pp(A, B):- d(A, ['.' D]), pp(D, B). d(A, B):- A = [C B],            member(C, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]).</pre>
Query	<pre>?- prog_8([1,1, '.',5]).</pre>
Answer	

9. Program	<pre> prog_9(L, NewL) :- prog(L, [], NewL). prog([], A, A). prog([X T], A, R) :- prog(T, [X A], R). </pre>
Query	<pre> prog_9([a,b,c], R). </pre>
Answer	R =

10. Program	<pre> new_stack(stack([])). push(Elem, stack(Stack), stack([Elem   Stack])). pop(stack([Top   Stack]), Top, stack(Stack)). </pre>
Query	<pre> ?- new_stack(S1), push(5, S1, S2), push(7, S2, S3). </pre>
Answer	<pre> S1 = S2 = S3 = </pre>

**E.** [4 marks]

Write a predicate named `add_one` that increments by 1 all the numbers in a binary tree. The first argument is the initial tree and the second argument is the result. Assume that the initial tree is given and that it can contain numbers or atoms. If a key is not a number, it remains unchanged in the result. Example :

```
?- add_one(t(1,t(a,nil,nil),t(3,nil,t(b,nil,nil))), NewT).
```

```
NewT = t(2, t(a, nil, nil), t(4, nil, t(b, nil, nil)))
```



**F.** [4 marks]

Write a predicate named `count` that counts how many elements are in a nested list, at all levels. The first argument is the nested list and the second is the number of elements.

Example:

```
?- count([a,[[b, c], d], e],R).
```

```
R = 5
```

