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**

Last Name:		**First Name:**	
Student Number:			

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. **<u>Please answer all questions on this paper, in the indicated spaces.</u>**
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 <u>all</u> 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	Atom
2.	_x	Variable
3.	50.1	Number
4.	F([1, a, 3])	No
5.	likes(john, mary)	Structure

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 = a	Y = [[c]]
2.	[[a]]	[X , Y]	X = No	Y = No
3.	[f(a)]	[X | Y]	X = f(a)	Y = []
4.	[f(a), f(b), c]	[X | Y]	X = f(a)	Y = [f(b), c]
5.	[[a], a]	[X | Y]	X = [a]	Y = [a]
6.	[f(Y), 1]	[X | Y]	X = f([1])	Y = [1]
7.	f([1, 2, 3])	f([X | Y])	X = 1	Y = [2,3]
8.	f(X, g(a))	g(X, Y)	X = No	Y = No
9.	f(X, g(a))	f(g(Y),Y)	X = g(g(a))	Y = g(a)
10.	f(X, g([1, 2]))	f(g(a),Y)	X = g(a)	Y = g([1, 2])

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

a) No

b) R = []
Yes

c) R = [red, book]
Yes

d) None of the above

Answer Sheet: Part C

Question	Answer
1.	**D**
2.	**C**
3.	**B**
4.	**D**
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	`prog_1(X,[X,_]).` `prog_1(X,[_|L]):-prog_1(X,L).`
Query	`?- prog_1(X,[a,b,c,d]).`
Answer	`X = c`

2.
Program
	`prog_2(L,R) :- acc(L, 0, R).`
	`acc([_],A,A).`
	`acc([H|T],A,R):- A1 is A+H, acc(T,A1,R).`

Query	`prog_2([1,2,3,4,5],R).`
Answer	`R = 10`

3. Program	prog_3([X|_], 1, X). prog_3([_|Xs], K, Y):- K > 1, K1 is K-1, prog_3(Xs,K1,Y).
Query	?- prog_3([a,b,c,d],3,R).
Answer	R = c

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

5. Program	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).
Query	?- prog_5(t(1, t(2, t(4, nil, nil), t(5, nil, nil)),t(3, nil, nil)), Z).
Answer	Z = [1, 2, 4, 5, 3]

6. Program	p1(1). p2(X) :- p1(X),!,fail. p2(_). p3(X) :- p1(X),fail,!. p3(_).
Query	?- p2(1),p3(2).
Answer	No

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

8. Program	prog_8(S) :- pp(S,[]). pp --> d. pp --> d,[.],pp. d --> [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]).
Query	?- prog_8([1,1,'.',5]).
Answer	No

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

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

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)))
```

```
add_one(nil, nil).

add_one(t(X, L, R), t(X1, NL, NR)):-
      number(X), !, X1 is X + 1, add_one(L, NL), add_one(R, NR).

add_one(t(X, L, R), t(X, NL, NR)):-
     add_one(L, NL), add_one(R, NR).
```

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
```

```
count([],0).

count([H|T], N) :- is_list(H), !, count(H,N1), count(T,N2), N is N1 + N2.

count([_|T], N) :- count(T,N1), N is N1 + 1.
```