UNIVERSITY OF OTTAWA FACULTY OF ENGINEERING SCHOOL OF IT AND ENGINEERING

CSI 2165 B

Midterm February 28, 2006 8-10 am

Examiner:

Dr. Diana Inkpen

Name	
Student Number	

Total marks:	45
Duration:	110 minutes
Total Number of pages:	10

Important Regulations:

- 1. No calculators are allowed.
- 2. A student identification card (or another photo ID) 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.

Α	В	С	D	Ε	F	G	Total
5	10	12	4	7	3	4	45

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	Туре
1.	ab13	Atom
2.	_const3	Variable
3.	A(B)	No
4.	[1, [2], f(a)]	Structure (list)
5.	country(name('Canada'))	Structure

B. [10 marks]

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

	Term1	Term2	Instantiati	ons
Example	[1, 2, 3]	[X Y]	X = 1	Y = [2, 3]
1.	[1 [2,3]]	[X, Y]	X = No	Y = No
2.	[[a, c], g(Y)]	[X, g(b)]	$\mathbf{X} = [\mathbf{a}, \mathbf{c}]$	$\mathbf{Y} = \mathbf{b}$
3.	[f(a), g(b)]	[X Y]	X = f(a)	$\mathbf{Y} = [\mathbf{g}(\mathbf{b})]$
4.	[[a, b], c, d]	[X, Y]	X = No	Y = No
5.	f([1,2], [a,b,c])	g(X,Y)	X = No	Y = No
6.	[X [Y, Z [a]]]	[b, X [Y, _]]	X = b $Y = b$	Z = b
7.	[1, 2 [3]]	[X Y]	X = 1	Y = [2, 3]
8.	[1, 2]	[X [Y []]]	X = 1	Y = 2
9.	[1, [2, 3]]	[_ [X Y]]	X = [2, 3]	Y = []
10.	a([X, [Y, a(X)]])	a([[a] [[X Z]]])	X = [a] Y = [a]	Z = [a([a])]

C. [12 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([],[]). prog_1([X T], [X,X T1]):- prog_1(T, T1).
Query	?- prog_1([a, b],R).
Answer	R = [a, a, b, b]

2. Program	<pre>prog_2(L,R) :- acc(L, 1, R). acc([],A,A). acc([H T],A,R):- A1 is A*H, acc(T,A1,R).</pre>
Query Answer	<pre>?- prog_2([1,2,3,4],R). R = 24 multiplies all elements</pre>

3. Program	prog_3([], Y, [Y]). prog_3([H T], Y, [H T1]):- prog_3(T, Y, T1).
Query	prog_3([a,b,c], 5 , R).
Answer	R = [a, b, c, 5] adds element at end of list

4. Program	prog_4(X, [X]). prog_4(X, [_ L]) :- prog_4(X, L).
Query	?- prog_4(X, [a, b, c, d]).
Answer	X = d, find the last element

5. Program	<pre>prog_5([X Xs], 1, Xs). prog_5([Y Xs], K,[Y Ys]) :-</pre>
Query	?- prog_5([a, b, c, d], 2, R).
Answer	R = [a,c,d], remove element at given position

6. Program	prog_6(L, X): - append(L1, [X L2], L), length(L1, Y), length(L2, Y).
Query	?- prog_6([a, b, c, d, e], X).
Answer	X = c, middle element

D. [4 marks]

Given the two predicates dell and del2, fill the table below with the SWI-Prolog's response to the corresponding query. The predicates are similar, but the clause order is different. The first argument is an element, the second is a list, and the third is a list where the result is stored.

Note that you have to include all answers by assuming you hit ";" instead of "ENTER".

```
del1(_, [], []).
del1(X, [X | Tail], Tail_1):- del1(X, Tail, Tail_1).
del1(X, [Y | Tail], [Y | Tail_1]):- del1(X, Tail, Tail_1).
del2(_, [], []).
del2(X, [Y | Tail], [Y | Tail_1]):- del2(X, Tail, Tail_1).
del2(X, [X | Tail], Tail_1):- del2(X, Tail, Tail_1).
```

Query	del1(1, [2,1],P).	del2(1, [2,1],P).
All answers	P = [2] ; P = [2, 1] ; No	P = [2, 1]; P = [2]; No

E. [7 marks] Facts and rules in Prolog

In the kingdom *Logiquogne* the penal code contains the following laws:

All the thieves are criminals. All the persons who help a criminal are also criminals. The adult criminals go to prison. The minor criminals go to a correction house.

1. The kind asks you to write a Prolog program to replace the judge in the courthouse. Translate the laws into Prolog, using the following symbols for predicates:

```
thief(X): X is a thief
criminal(X): X is a criminal
helps(X,Y): X helps Y
adult(X): X is an adult
minor(X): X is a minor
prison(X): X goes to prison
correction(X): X goes to the correction house
```

2. Using the following database:

```
thief(thomas).
thief(arsene).
minor(thomas).
minor(philibert).
minor(paul).
minor(sara).
adult(arsene).
adult(mireille).
adult(caroline).
adult(charles).
helps(philibert,mireille).
helps(paul, arsene).
helps(paul, charles).
helps(arsene,philibert).
helps(mireille, sara).
helps(charles, caroline).
helps(caroline,thomas).
```

- 2.1. What Prolog query do you need to ask for finding all the persons who go to prison?
- 2.2. What are the Prolog answers if you press ; after each solution ?
- 2.3. What Prolog query do you need to ask for finding all the persons who go to the correction house?
- 2.4. What are the Prolog answers if you press ; after each solution ?

Solution

1. Rules:

criminal

criminal(X) :- thief(X). criminal(X) :- helps(X,Y), criminal(Y).

prison

```
prison(X) :- criminal(X), adult(X).
```

correction

```
correction(X) :- criminal(X), minor(X).
```

2. Queries and answers :

```
?- prison(X).
X = arsene ;
X = charles ;
X = caroline ;
No
?- correction(X).
X = thomas ;
X = paul ;
X = paul ;
No
```

F. [3 marks]

Write a predicate named rem_first that has three arguments: a list, a number, and an output list. The predicate removes a given number of elements from the front of the given list. Assume the number is less or equal to the length of the list.

```
Examples:
?- rem_first([a,b,c,d,e,f,g], 4, L).
L = [e,f,g] ;
No
?- rem_first([a,b,c,d,e], 1, L).
L = [b,c,d,e] ;
No
```

```
rem_first(L, 0, L).
rem_first([_ | T], N, T1):- N1 is N-1, rem_first(T, N1, T1).
```

G. [4 marks]

Write a predicate named pairs that has three arguments: a first list, a second list, and a third argument. The predicate returns in the third argument a list of all the possible pairs of elements from the two given lists.

```
Example:

?- pairs([1, 2, 3],[a, b],L).

L = [[1, a], [1, b], [2, a], [2, b], [3, a], [3, b]];

No
```

```
pairs([], _, []).
pairs([H|T], L2, R):- p_helper(H,L2,R1), pairs(T, L2, R2),
append(R1, R2, R).
p_helper(_, [], []).
p_helper(E, [H|T], [[E,H]|T1]) :- p_helper(E,T,T1).
```