P. van Beek

CMPUT 325, Final Examination B1

Tuesday, April 25, 2000

Name: ______________________ ID: ______________________

You have 2 hours to complete the exam.
Be sure to put your name and ID on this page.
The pages are numbered from 1 to 11; make sure you have them all.
This exam is worth 40% of your final mark.
Write your answers on the exam sheet.
This exam is a closed book exam.

Question	Worth	Mark
1	(12)	______
2	(12)	______
3	(24)	______
4	(12)	______
5	(12)	______
6	(22)	______
7	(6)	______
Total	(100)	______

1. (6 + 6 = 12 points) Lambda calculus.

For each of the following lambda expressions, reduce them to normal form using the given reduction technique. If an expression cannot be reduced to normal form using the reduction technique then state: not reducible. You must show each step of the reduction. Your mark will be reduced if you miss showing a step or include additional unnecessary steps (such as any unnecessary renamings).

(a) Applicative order reduction of:
$((\lambda x \mid ((\lambda y \mid (y\ x))(\lambda z \mid (z\ x))))((\lambda y \mid (y\ y))(\lambda x \mid x)))$

(b) Normal order reduction of:
$((\lambda x \mid ((\lambda y \mid (y\ x))(\lambda z \mid (z\ x))))((\lambda y \mid (y\ y))(\lambda x \mid x)))$

2. (12 points) Lisp interpreters.

In Assignment 3, we wrote an interpreter which evaluated S-expressions. The implementation of eval (actually we called it xeval) used contexts and closures. A context was implemented using a name and value list which were initially both nil. Show how the interpreter would reduce the following expression using contexts and closures:

eval[((lambda (x) ((lambda (y) (* x y)) x)) ((lambda (x y) (+ x y)) 2 1)), nil, nil]

As a reminder, the S-expression above is equivalent to the following lambda calculus expression,

$$((\lambda\ x \mid ((\lambda\ y \mid (*\ x\ y))\ x))(((\lambda\ x \mid (\lambda\ y \mid (+\ x\ y)))\ 2)\ 1)),$$

and to eval a function application: evaluate the function, evaluate the arguments, bind the parameters, and call eval. **Show all of your work.**

3. (9 + 15 = 24 points) Prolog programming.

(a) Consider the following Prolog program. For each query, show what Prolog would return. You only need to show the first answer in the case of "yes". Put your final answer on the line provided.

```
p([], X, X).
p([Head|Tail], X, [Head, Y]) :- p(Tail, X, Y).
q([], []).
q([Head|Tail], Y) :- q(Tail, X), p(X, [Head], Y).
```

(i) `?- q([a,b,c], W).`

Answer = ______________________________

(ii) `?- q([a], [W]).`

Answer = ______________________________

(iii) `?- p([1,2], [a,b,c], W).`

Answer = ______________________________

(b) Write the Prolog predicate:

```
append( L1, L2, L3 )
```

where L1, L2, and L3 are lists. The predicate is true if and only if L3 is the result of appending L1 and L2. Here are some examples:

```
?- append([a,b,c], [d,e,f], [a,b,c,d,e,f]).
yes

?- append([a,b,c], [d,e,f], X).
X = [a,b,c,d,e,f]

?- append([a,b,c], X, [a,b,c,d,e,f]).
X = [d,e,f]

?- append(X, [d,e,f], [a,b,c,d,e,f]).
X = [a,b,c]
```

You may not use "cuts" in your program.

4. $(4 + 4 + 4 = 12$ points) Clausal form.

Translate the following well-formed formulas into clausal form. Put your final clause(s) on the line(s) provided.

(a) $((P \Rightarrow Q) \land \neg P) \Rightarrow \neg Q$

Clause ____________________________________

Clause ____________________________________

Clause ____________________________________

(b) $\forall x\ \forall y\ [P(x, y) \Rightarrow (Q(x, y) \lor R(x, y))]$

Clause ____________________________________

Clause ____________________________________

Clause ____________________________________

(c) $[(\exists x\ (\neg P(x) \land Q(x))) \Rightarrow (\forall x\ (\neg P(x) \lor Q(x)))] \land [(\forall x\ P(x)) \Rightarrow (\exists x\ Q(x))]$

Clause ____________________________________

Clause ____________________________________

Clause ____________________________________

5. $(3 + (3 \times 3) = 12$ points) Unification.

(a) Why is the unification algorithm needed for predicate logic proof procedures?

(b) What is the output of the unification algorithm for the following inputs? The symbols W, X, Y, and Z are variables; everything else is either a constant, a functor symbol, or a predicate symbol.

(i) unify(q(a, b), q(W, Z))

$\alpha =$ ______________________________

(ii) unify(p(f(X), Z), p(a, b))

$\alpha =$ ______________________________

(iii) unify(p(Z, h(Z, W), f(W)), p(f(X), h(Y, f(a)), Y))

$\alpha =$ ______________________________

6. ($3 + 3 + 8 + 8 = 22$ points) Logical consequence and resolution refutation proofs.

Consider the blocks world. Suppose we have the predicates On, Above, Clear, Green, and OnTable with the intended meanings as,

On(x, y)	x is on y,
Above(x, y)	x is above (i.e., in the same tower but not necessarily directly on) y,
Clear(x)	x is clear,
Green(x)	x is colored green,
OnTable(x)	x is on the table.

and suppose that Γ is the conjunction of the following statements,

On(A, B),	On(C, D),	Green(B),	
On(B, C),	On(D, E),	¬Green(D),	OnTable(E),

$\forall x \forall y\, [\text{On}(x, y) \Rightarrow \text{Above}(x, y)]$,
$\forall x \forall y \forall z\, [\text{Above}(x, y) \wedge \text{Above}(y, z) \Rightarrow \text{Above}(x, z)]$,
$\forall x \neg \exists y\, [\text{Above}(x, y) \wedge \text{Clear}(y)]$,
$\forall x \neg \exists y\, [\text{Above}(x, y) \wedge \text{OnTable}(x)]$,
$\forall x [(\exists y\, \text{On}(y, x)) \vee \text{Clear}(x)]$.

(a) Give an interpretation in which Γ is satisfied.

(b) Give an interpretation in which Γ is not satisfied.

(c) For the following query, ϕ, determine whether the query is a logical consequence of Γ (i.e., does $\Gamma \models \phi$ hold?). If the query is not a logical consequence, state why it is not. If the query is a logical consequence, give a resolution refutation proof and say why this allows us to conclude logical consequence.

Is there a block that is clear that has a green block directly below it?
$\exists x \exists y\ [\text{Clear}(x) \wedge \text{On}(x,\ y) \wedge \text{Green}(y)]$

(d) For the following query, ϕ, determine whether the query is a logical consequence of Γ (i.e., does $\Gamma \models \phi$ hold?). If the query is not a logical consequence, state why it is not. If the query is a logical consequence, give a resolution refutation proof and say why this allows us to conclude logical consequence.

Is there a green block that has a block below it that is on the table?
$\exists x \exists y\ [\text{Green}(x) \wedge \text{Above}(x, y) \wedge \text{OnTable}(y)]$

7. (6 points) Prolog interpreters.

   State what it means for a proof procedure to be **complete** and explain whether Prolog is complete or incomplete. (Guesses without any explanation will not be given any marks.)