UNIVERSITY OF MALTA

FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & A. I.
B.Sc. (Hons.) YEAR I

January 2001 Assessment Session

CSI 107 Introduction to Logic and Logic Programming **27th January 2001**

09:15 – 11:45

This paper contains six questions. You are to attempt questions **ONE** *and* **FOUR** *and any other* **two** *questions.*

Section A

1. Explain what a Horn sentence is, giving suitable examples of Horn and non-Horn sentences.
[5 marks]

Give a detailed overview of the resolution method. Make sure to include an explanation of what resolvents are and give a definition of R^*.
[10 marks]

Explain what NNF, DNF and CNF sentences are, giving suitable examples.
[4 marks]

Explain the underlying concepts behind satisfiability on a single sentence and a set of sentences. Why are efficient methods of determining satisfiability needed?
[6 marks]

Assuming that p, q, r are atomic statements, T_0 a tautology and F_0 a contradiction, give definitions of:

(a) De Morgan's Law and its quantified equivalent.

(b) Identity, inverse and double negation laws.

[5 marks]

2. Explain the difference between bound and unbound variables, giving a suitable example.
[5 marks]

Define logical equivalence. Give a chain of equivalences showing that the negation of $(\neg\exists x\ (P(x) \wedge Q(x)))$ is equivalent to $(\forall x\ (P(x) \rightarrow \neg Q(x)))$.

[7 marks]

(Continues on next page)
Using truth tables show whether the sentences $\neg((A \vee B) \wedge \neg C)$ and $(\neg A \wedge \neg B) \vee C$ are equivalent or not.
[3 marks]

Define what a Formal Deductive System (FDS) is. Make sure to explain the architecture and the three sub-components of FD architecture S. Give two examples of FDS.

[10 marks]

3. Explain the difference between a proof and a theorem. Additionally give a definition of what a deduction is.

[5 marks]

Valid inference steps are used in proofs. Give a definition of Double Negation, Conjunction Introduction, Disjunction Introduction, and Conjunction Elimination. Give a brief explanation of the basic steps involved in producing a proof by contradiction.

[10 marks]

Assume the following names and predicates.

	English	**FOL**
Names	Charles	charles
	Abigail	abigail
	Max	max
Predicates	X is a student	Student(x)
	X is a pet	Pet(x)
	X is Y's father	Father(x, y)
	X is taller than Y	Taller(x, y)
	X is shorter than Y	Shorter(x, y)

Translate the following into FOL.

(a) Max is a student, not a pet.
(b) Charles is Abigail's father.
(c) Charles is taller than Abigail.
(d) Max is the same height as Abigail.

Translate the following into English.

(e) $\forall x$ Student(x) $\rightarrow \neg$Pet(x).
(f) $\forall x \exists y$ Father(x, y) $\wedge$ Taller(y, x).

(Hint: You do not need to introduce any new names or predicates).

[10 marks]

Section B

4. Consider the program below:

```
goal1(alf).
goal1(bert).
goal1(bob).
nl,
goal1(dick).
goal2(ann).
goal2(dick).
goal2(cliff).
goal2(doris).
('end'),

run(1):-    goal1(A),
            goal2(B),
            A=/=B,

            write(A),
            write(B), nl,
            fail.

run(2) :-   goal1(A),

            !,
            goal2(B),
            write(A),
            write(B), nl,

run(X) :-   X > 2,
            goal1(A),
            write(A),

            Y is X+1,
            Y < 5,
            run(Y).

run( _ ) :- write

            nl.
```

Determine what output is produced by the following goals.

(a) **run(1)**

[7 marks]

(b) **run(2)**

[8 marks]

(c) **run(3)**

[10 marks]

5. The **Cartesian Product** of two sets is the set containing all the tuples produced from one element of one set and second element from the other set. Consider the sets A = {a, b, c} and B = {1, 2, 3}

The cartesian product of A and B = A × B

= { (a,1), (a,2), (a,3), (b,1), (b,2), (b,3), (c,1), (c,2), (c,3) }

Write a Prolog program in which it is possible to issue goals to display all the tuples produced from the Cartesian Product of any two sets, including the empty set.

e.g. cartesian_product ([a,b,c],[1,2,3]) would display:

```
a 1   a 2   a 3
b 1   b 2   b 3
c 1   c 2   c 3
```

[25 marks]

6. Write recursive clauses for the following predicates:

(a) **deduct(List_of_Integers, Number, Result)** which binds **Result** to the value of the **Number** after that each element within the **List_of_Integers** are deducted from it.

e.g. deduct([3,2,5], 20, Ans) binds Ans to 10.

[9 marks]

(Continues on next page)

(b) **display_items(List_of_Items)** which displays a sequence of the items within the **List_of_Items** from left to right.

e.g. display_items([max, $, 2, lca648]) displays max $ 2 lca648

[8 marks]

(c) **push(Item, List, Stack)** which binds **Stack** to **List** with the **Item** entered at its head.
e.g. push(a, [b,c], NewStack) binds NewStack to [a,b,c].

[8 marks]