CS 81 Practice Problems for the Midterm

Note: This is not a sample exam. The set of problems here is more than would be on an exam.

The midterm will be in two parts, in this order:

1. Part 1 will ask about concepts and definitions. The problems to be solved, if any, will be relatively straightforward. This part is to be done without reference to any materials.

2. Part 2 can be started as soon as you have turned in Part 1. It will consist of problems and perhaps deeper questions. For this part, you may use as your only resource a crib sheet: one sheet of paper, double-sided, with contents of your own construction. Turn in your sheet with the exam.

The approximate division of points will be 25% for part 1 and 75% for part 2.

Terms and concepts to review:

Propositional connectives	Soundness
Sequents	Completeness
Natural deduction proof rules for propositional logic	Quantifiers
Intuitionistic rules	Equality rules
Reductio ad absurdum, modus ponens, modus tollens	Structural induction
Law of the Excluded Middle	Interpretation
Syntax vs. semantics	Model
Mathematical induction	Validity
Strong form of mathematical induction	Counter-model
Tautology	Number theory axioms
Satisfiability for propositional logic	Group axioms
Truth table	Weaker vs. stronger formulas
Quine's method (or Boole/Shannon method)	Satisfiability for predicate logic
Tree method for propositional logic	Tree method for predicate logic
Natural deduction proof rules for predicate logic	Hoare triples method
	Invariant

Review problems:

1. Classify these formulas as one of
   {tautology, satisfiable but not a tautology, not satisfiable}:

   a. $((p \rightarrow q) \rightarrow (q \rightarrow r)) \rightarrow (p \rightarrow r)$
      Tautology check:
      $\neg(((p \rightarrow q) \rightarrow (q \rightarrow r)) \rightarrow (p \rightarrow r))$ 1.

$((p \to q) \to (q \to r))$ 3.

$\neg(p \to r)$ 2.

p

$\neg r$

$\neg(p \to q)$ 4. $(q \to r)$

p

$\neg q$

There is an open path: p = T, q = F, r = F, which makes the original formula false. So it is **not a tautology**. However, it is **satisfiable**, by making r = T.

b. $(p \to q) \to (q \to p)$

Not a tautology, by taking p = F, q = T. However, **satisfiable** by taking p = T, q = F.

c. $\neg(\neg q \to p) \to \neg(p \wedge \neg q)$

Testing the negation for satisfiability:

$\neg(\neg(\neg q \to p) \to \neg(p \wedge \neg q))$ 1.

$\neg(\neg q \to p))$ 4.

$\neg\neg(p \wedge \neg q))$ 2.

$(p \wedge \neg q))$ 3.

p

$\neg q$

$\neg q$

$\neg p$

X

So the original formula **is a tautology**.

Proof:

1.	$\neg(\neg q \to p)$	Assumption
2.	$(p \wedge \neg q)$	Assumption
3.	$\neg q$	Assumption
4.	p	$\wedge$e 2
5.	$\neg q \to p$	$\to$i 3-4
6.	$(p \wedge \neg q) \to (\neg q \to p)$	$\to$i 2-5
7.	$(p \wedge \neg q)$	Assumption
8.	$(\neg q \to p)$	$\to$e 7, 6
9.	$\bot$	$\neg$e 1, 8
10.	$\neg(p \wedge \neg q)$	$\neg$i 7-9
11.	$\neg(\neg q \to p) \to \neg(p \wedge \neg q)$	$\to$i 1-10

d. $(p \wedge (q \vee r)) \equiv ((p \wedge q) \vee (p \wedge r))$

This is a tautology. Proof:

1.	$p \wedge (q \vee r)$	Assumption
2.	p	$\wedge e_1$ 1
3.	$q \vee r$	$\wedge e_2$ 1
4.	q	Assumption
5.	$p \wedge q$	$\wedge i$ 2, 4
6.	$(p \wedge q) \vee (p \wedge r)$	$\vee i_1$ 5
7.	r	Assumption
8.	$p \wedge r$	$\wedge i$ 2, 7
9.	$(p \wedge q) \vee (p \wedge r)$	$\vee\ i_2$ 8
10.	$(p \wedge q) \vee (p \wedge r)$	$\vee e$ 3, 4-6, 7-9
11.	$(p \wedge q) \vee (p \wedge r)$	Assumption
12.	$p \wedge q$	Assumption
13.	p	$\wedge e_1$ 12
14.	q	$\wedge e_2$ 12
15.	$q \vee r$	$\vee i$ 14
16.	$p \wedge (q \vee r)$	$\wedge i$ 13, 25
17.	$p \wedge r$	Assumption
18.	p	$\wedge e_1$ 17
19.	r	$\wedge e_2$ 17
20.	$q \vee r$	$\vee i_1$ 17
21.	$p \wedge (q \vee r)$	$\wedge i$ 18, 20
22.	$p \wedge (q \vee r)$	$\vee e$ 11, 12-16, 17-21
23.	$(p \wedge (q \vee r)) \equiv ((p \wedge q) \vee (p \wedge r))$	$\equiv i$ 1-10, 11-22

2. For each formula above:

   a. If the formula is a tautology, give a natural deduction proof of it.
   b. If the formula is unsatisfiable, give a natural deduction proof of its negation.
   c. If the formula is satisfiable, but not a tautology, give an assignment that satisfies it and one that does not.

   I have included the answers with those for #1.

3. Prove that in any group, (x y)' = y'x', where x' represents the inverse of the element x and juxtaposition represents multiplication.

   Proof: (Liberally using equality rules without stating all of them.)

   1. $(xy)\,(x\,y)' = u$ G3
   2. $x\,(y\,(x\,y)') = u$ G1 1

3. x'(x (y (x y)'))= x'u = axiom 2
4. x'(x (y (x y)'))= x' G2 3
5. (x'x) (y (x y)') = x' G1 4
6. u (y (x y)') = x' G3 5
7. (y (x y)') = x' G2 6
8. y' (y (x y)') = y'x' = axiom 7
9. (y' y) (x y)' = y'x' G1 8
10. u (x y)' = y'x' G3 9
11. (x y)' = y'x' G2 10

4. For the natural numbers, using axioms and theorems we have developed so far, prove that:

$\forall x \forall y \forall z\ (x + y)*z = (x*z) + (y*z)$

Proof by induction on z (freely using ∀e and = axioms implicitly):

1.	x, y	
2.	(x + y)*0 = 0	N5
3.	x*0 = 0	N5
4.	y*0 = 0	N5
5.	x*0 + y*0 = 0	N3
6.	(x + y)*0 = x*0 + y*0	= 1, 5
7.	∀x∀y (x + y)*0 = x*0 + y*0	∀i twice, 6

8.	x, y	
9.	(x + y)*z = (x*z) + (y*z)	Assumption
10.	(x + y)*S(z) = (x+y)*z + (x+y)	N6
11.	(x + y)*S(z) = ((x*z) + (y*z)) + (x+y)	=subst 9 into rhs of 10
12.	(x + y)*S(z) = (((x*z) + (y*z)) + x)+y	Assoc. of +, 11
13.	(x + y)*S(z) = (x + ((x*z) + (y*z)))+y	Comm. of +, 12
14.	(x + y)*S(z) = ((x + (x*z)) + (y*z))+y	Assoc. of +, 13
15.	(x + y)*S(z) = (((x*z)+x) + (y*z))+y	Comm. of +, 14
16.	(x + y)*S(z) = (x*S(z) + (y*z))+y	N6 15
17.	(x + y)*S(z) = x*S(z) + ((y*z) +y)	Assoc. of +, 16
18.	(x + y)*S(z) = x*S(z) + y*S(z)	N6 17
19.	∀x∀y (x + y)*S(z) = x*S(z) + y*S(z)	∀i twice, 18

20.	∀z ∀x ∀y (x + y)*z = (x*z) + (y*z)	induction rule 7, 8-19
21.	∀x ∀y∀z (x + y)*z = (x*z) + (y*z)	rearrange ∀ 20

5. Classify these formulas as one of
{valid, satisfiable but not valid, not satisfiable}:

1. (∃x P(x) → ∃xQ(x)) → ∃x (P(x) → Q(x))

Valid, proof:

1.	(∃x P(x) → ∃xQ(x))	Assumption
2.	∃xQ(x) ∨ ¬∃xQ(x)	LEM
3.	∃xQ(x)	Assumption
4.	x_0 Q(x_0)	Assumption
5.	Q(x_0)	
6.	P(x_0)	Assumption
7.	P(x_0) → Q(x_0)	→i 5-6
8.	∃x (P(x) → Q(x))	∃i 7
9.	∃x (P(x) → Q(x))	∃e 3, 4-8
10.	¬∃xQ(x)	Assumption
11.	∃xP(x) ∨ ¬∃xP(x)	LEM
12.	∃x P(x)	Assumption
13.	∃xQ(x)	→e 12, 1
14.	⊥	¬e 13, 10
15.	∃x (P(x) → Q(x))	⊥e 14
16.	¬∃xP(x)	Assumption
17.	P(z)	Assumption
18.	∃x P(x)	∃i 17
19.	⊥	¬e 18, 16
20.	Q(z)	⊥e 19
21.	P(z) → Q(z)	→i 17-20
22.	∃x (P(x) → Q(x))	∃i 21
23.	∃x (P(x) → Q(x))	∨e 11, 12-15, 16-22
24.	∃x (P(x) → Q(x))	∨e 2, 3-9, 10-23
25.	(∃x P(x) → ∃xQ(x)) → ∃x (P(x) → Q(x))	→i 1-24

2. (∀x P(x) → ∀x Q(x)) → ∀x (P(x) → Q(x))

This formula is not valid, but is satisfiable.
Model: Δ = {1}, P = {}, Q = {(1)} since (F → T) → T.
Countermodel: Δ = {1, 2}, P = {(1)}, Q = {} since not (F → F) → F.

3. $(\exists y\ \forall x\ R(x, y)) \rightarrow (\forall x\ \exists y\ R(x, y))$
This formula is valid. Proof

1.	$\exists y\ \forall x\ R(x, y)$	Assumption
2.	$y_0\ \forall x\ R(x, y_0)$	Assumption
3.	x_0	
4.	$R(x_0, y_0)$	$\forall$e 2
5.	$\exists y\ R(x, y_0)$	$\exists$i 4
6.	$\forall x\ \exists y\ R(x, y)$	$\forall$i 3-5
7.	$\forall x\ \exists y\ R(x, y)$	$\exists$e 2-6

6. For each formula above:

   a. If the formula is a valid, give a natural deduction proof of it.
   b. If the formula is unsatisfiable, give a tree refutation of it.
   c. If the formula is satisfiable, but not valid, give both a model and a counter model for it.

   Already done above.

7. For each program below, derive the weakest assumption that will cause the indicated expectation to be met. Assume that all variables are integer.

1. {y > 3}   ans. y > 3
   {(y+1)*5 > 20}
   x := y+1;
   {x*5 > 20}
   y = x*5;
   {y > 20}
   {y+1 > 21}
   x = y+1;
   {x > 21}

1. { true }   ans. true
   { $(x > 0) \rightarrow (x \geq 0) \wedge \neg(x > 0) \rightarrow (-x \geq 0)$ }
   if(x > 0)
      {$x \geq 0$}
      x := x;
      {$x \geq 0$}
   else
      {$-x \geq 0$}
      x := -x;
      {$x \geq 0$}
   {$x \geq 0$}

2. $\{x > y \land x > z\}$ ans. $\{x > y \land x > z\}$

$\{ x > y \rightarrow x > z \land \neg\, x > y \rightarrow \text{false}\}$

```
if( x > y )
    { x > z }
    { x > z → 2 = 2 ∧ ¬x > z → 1 = 2}
    if( x > z )
            r := 2;
    else
            r := 1;
else
    {false}
    { x > z → 1 = 2 ∧ ¬ x > z → 0 = 2}
    if( x > z )
            r := 1;
        else
            r := 0;
        {r = 2}
```

8. For each program below, indicate the invariant for the while loops that will cause the expectation to be met, given that the assumption is met. Then prove that the program meets the specification.

   a. $\{ n \geq 0 \}$

```
{0 ≤ n ∧ 1 = 0 + 1 ∧ 0 = 0²}
s := 0;
{0 ≤ n ∧ 1 = 0 + 1 ∧ s = 0²}
k := 1;
{0 ≤ n ∧ k = 0 + 1 ∧ s = 0²}
i := 0;
{i ≤ n ∧ k = 2i + 1 ∧ s = i²}                    invariant
while( i < n )
    { i < n ∧ i ≤ n ∧ k = 2i + 1 ∧ s = i²}
    { i + 1 ≤ n ∧ k+2 = 2(i + 1) + 1 ∧ s + k = (i + 1)²}
    s := s + k;
    { i + 1 ≤ n ∧ k+2 = 2(i + 1) + 1 ∧ s = (i + 1)²}
    k := k + 2;
    { i + 1 ≤ n ∧ k = 2(i + 1) + 1 ∧ s = (i + 1)²}
    i := i + 1;
    {i ≤ n ∧ k = 2i + 1 ∧ s = i²}
{¬ i < n ∧ i ≤ n ∧ k = 2i + 1 ∧ s = i²}
{ i = n ∧ s = i²}
{ s = n² }
```

b. $\{ m \geq 0 \wedge n \geq 0 \wedge m = m_0 \}$

$\{ m \geq 0 \wedge 0 = 0\}$

$\{ m \geq 0 \wedge 0 = n*(m_0 - m)\}$

$\{ m \geq 0 \wedge 0 = n*(m_0 - m)\}$

s := 0;

$\{ m \geq 0 \wedge s = n*(m_0 - m)\}$ invariant

while(m > 0)

$\{ m > 0 \wedge m \geq 0 \wedge s = n*(m_0 – m))\}$

$\{ m > 0 \wedge s = n*(m_0 – m))\}$

$\{ m > 0 \wedge s+n = n*(m_0 – m+1))\}$

$\{ m-1 \geq 0 \wedge (s+n) = n*(m_0 – (m-1))\}$

s := s + n;

$\{ m-1 \geq 0 \wedge s = n*(m_0 – (m-1))\}$

m := m-1;

$\{ m \geq 0 \wedge s = n*(m_0 - m)\}$

$\{ \neg(m > 0) \wedge m \geq 0 \wedge s = n*(m_0 - m)\}$

$\{ (m = 0)\ s = n*(m_0 - m)\}$

$\{ s = m_0*n \}$