C SC 322 Midterm
Thursday, October 25th, 2008 @ 2:30pm in UVic ECS 108

Name: Aaron Williams

Student ID:

Each page is worth 10 marks.

1. Let \( A = p \lor q \rightarrow q \land \neg p \).
   a) Draw the formation tree for \( A \).
   b) Give the inorder traversal of the formation tree.
   c) Use semantic tableau to show that \( A \) is satisfiable.
   d) Show that \( A \) is falsifiable.
   e) Write a formula that is unfalsifiable (a tautology) and whose formation tree has \( A \) as its inorder traversal.

```
\[ A = (p \lor q) \rightarrow (q \land \neg p) \] by precedence rules

\[ \begin{align*}
\text{a)} & \quad \rightarrow \\
\text{b)} & \quad p \lor q \rightarrow q \land \neg p
\end{align*} \]
```

Note: some additional comments are included in these solutions but were not necessary for full marks.

---

```
\[ A = (p \lor q) \rightarrow (q \land \neg p) \] by precedence rules

\[ \begin{align*}
\text{c)} & \quad p \lor q \rightarrow q \land \neg p \\
\text{d)} & \quad (T \lor T) \rightarrow (T \land F) \equiv T \rightarrow F \equiv F \equiv F
\end{align*} \]
```

\[ \text{d)} \quad \text{If } v(p) = T \text{ and } v(q) = T \text{ then } v(A) = F \text{ since } (T \lor T) \rightarrow (T \land F) \equiv T \rightarrow F \equiv F \equiv F \]

\[ \text{e)} \quad p \lor ((q \lor q) \land \neg p) \text{ is a tautology since } p \lor ((q \lor q) \land \neg p) \]

\[ \equiv p \lor (T \land \neg p) \equiv p \lor \neg p \equiv T \]

\[ \text{and its formation tree} \]

\[ \begin{align*}
\text{has an inorder traversal} \\
p \lor q \rightarrow q \land \neg p
\end{align*} \]

Since inorder traversals just "remove the brackets"
2. Use the three reduction rules to reduce the Binary Decision Diagram (BDD) that appears below. Show your work.
3. If $B$ is a BDD then $\text{maxTrueSAT}(B)$ is the maximum number of propositions that can be set to True in a model for the formula it represents. For example, if $B$ is the BDD from the previous page then $\text{maxTrueSAT}(B) = 2$ since at most two propositions can be set to True on a path from the root to a True leaf. If $B$ has no True leaves then we write $\text{maxTrueSAT}(B) = \times$.

a) Fill in the blank values for the leaves in an unreduced BDD.

\[
\text{maxTrueSAT} = \begin{cases} 
0 & \text{since satisfying assignment sets zero propositions to True} \\
1 & \text{since no satisfying assignments}
\end{cases}
\]

b) Fill in the blank value for this parent in an unreduced BDD.

\[
\text{maxTrueSAT} = \max(6, 8+1) = 9 \checkmark
\]

Since the first proposition is set to True.

\[
\text{maxTrueSAT} = 6 \\
\text{maxTrueSAT} = 8
\]

c) Fill in the blank value for this parent in an unreduced BDD.

\[
\text{maxTrueSAT} = \max(3, x+1) = 3 \checkmark
\]

Since the first proposition is set to True.

\[
\text{maxTrueSAT} = 3 \\
\text{maxTrueSAT} = \times
\]

d) Fill in the blank value for this parent in a reduced BDD.

\[
\text{maxTrueSAT} = \max(5+4, 5+2) = 9 \checkmark
\]

Since the first proposition is set to True and the second proposition can be set to True.

\[
\text{maxTrueSAT} = 5 \\
\text{maxTrueSAT} = 5
\]

e) Suppose $B$ is an independent set BDD for a graph $G$ and $\text{maxTrueSAT}(B) = k$. What can we say about the independent sets in $G$?

Then $G$ has an independent set of size $k$, and there are no larger independent sets. That is, the maximum independent set size in $G$ is $k$. \checkmark
4. The Fibonacci numbers are defined as follows: \( F(0) = 0, F(1) = 1, \) and \( F(n) = F(n-1) + F(n-2) \) for \( n \geq 2. \) Write Prolog code for `fibonacci(N,F)` so that \( F = F(N) \) whenever \( N \) is given as a constant. For example, entering `fibonacci(7,F)` into the interpreter should give the binding \( F = 13. \)

```prolog
fibonacci(0, 0).
fibonacci(1, 1).
fibonacci(N, F) :-
    N > 1,
    N1 is N-1,
    N2 is N-2,
    fibonacci(N1, F1),
    fibonacci(N2, F2),
    F is F1 + F2.
```

5. The decision problem NOT-SAT(\( \mathcal{A} \)) returns true iff \( \mathcal{A} \) is unsatisfiable. The decision problem EQUIV(\( \mathcal{B}, \mathcal{C} \)) returns true iff \( \mathcal{B} \equiv \mathcal{C} \). Explain how EQUIV can be used to solve NOT-SAT.

\[ \text{NOT-SAT}(\mathcal{A}) \text{ is equivalent to } \text{EQUIV}(\mathcal{A}, \text{False}). \]
Alternate answer: \( \text{NOT-SAT}(\mathcal{A}) \text{ is equivalent to } \text{EQUIV}(\mathcal{A}, \text{\_17p}). \)