CS486 Problem Set 2 DUE: 2/11/03

1. Solve the exercise on p. 24, items 1, 4, 5, and 8.

(1)

(1)	F	$q \supset (p \supset q)$	
(2)	T	q	(1)
(3)	F	$p \supset q$	(1)
(4)	T	p	(3)
(5)	F	q	(3)
		X(2)	

(4)

(1)	F	$[((p \supset r) \wedge (q \supset r)) \wedge (p \vee q)] \supset r$	
(2)	T	$((p \supset r) \wedge (q \supset r)) \wedge (p \vee q)$	(1)
(3)	F	r	(1)
(4)	T	$(p \supset r) \wedge (q \supset r)$	(2)
(5)	F	$p \vee q$	(2)
(6)	T	$p \supset r$	(4)
(7)	T	$q \supset r$	(4)

(8) T p (5)		(9) T q (5)	
(10) F p (6)	(11) T r (6)	(12) F q (7)	(13) T r (7)
X(8)	X(3)	X(9)	X(3)

(5)

(1)	F	$\neg(p \wedge q) \supset (\neg p \vee \neg q)$	
(2)	T	$\neg(p \wedge q)$	(1)
(3)	F	$\neg p \vee \neg q$	(1)
(4)	F	$p \wedge q$	(2)
(5)	F	$\neg p$	(3)
(6)	F	$\neg q$	(3)

(7) F p (4)	(8) F q (4)
(9) T p (5)	(10) T q (6)
X(7)	X(8)

(8)

(1)	F	$(p \vee (q \wedge r)) \supset ((p \vee q) \wedge (p \vee r))$	
(2)	T	$p \vee (q \wedge r)$	(1)
(3)	F	$(p \vee q) \wedge (p \vee r)$	(1)

(4) T p (2)		(5) T $q \wedge r$ (2)
(6) F $p \vee q$ (4)	(7) F $p \vee r$ (4)	*
(8) F p (6)	(9) F p (7)	
X(4)	X(4)	

*

(10)	T	q	(5)
(11)	T	r	(5)

(12) F $p \vee q$ (4)	(13) F $p \vee r$ (4)
(14) F p (12)	(16) F p (13)
(15) F q (12)	(17) F r (13)
X(10)	X(11)

2. Recall that a tableaux $\mathcal{T}$ is complete if all its branches are either closed or complete, where a branch θ of a tableaux $\mathcal{T}$ is complete if for every α on θ both α_1 and α_2 occur on θ and if for every β on θ at least one off β_1, β_2 occur on θ.

   We prove that the tableaux method terminates.

   We generate an analytic tableaux for the unsigned formula X in the following manner. At each stage, we have an ordered dyadic tree $\mathcal{T}_i$, In the first stage, we construct the tree $\mathcal{T}_1$, a one-point tree whose origin is $\mathsf{F}\ X$. In the n^{th} stage, we select from $\mathcal{T}_{n-1}$ the left-most branch θ that is unclosed and incomplete. From this branch, we consider all α on θ such that either α_1 or α_2 does not occur on θ and all β on θ such that neither β_1 nor β_2 occur on θ. We select the α or β node that dominates all others. If we select an α node such that α_1 does not occur on θ, then extend θ by α_1 to form $\mathcal{T}_n$. If we select an α node such that α_2 does not occur on θ, then extend θ by α_2 to form $\mathcal{T}_n$. If we select β node, then extend θ by β_1, β_2 to form $\mathcal{T}_n$.

   We show that the above method either closes or completes any branch θ by induction on the "size" of the branch. Define $size(\theta)$ as follows:

$$size(\theta) = \begin{cases} 0 & \text{if } \theta \text{ closed or completed} \\ \sum_{x \text{ dominates } y} deg(x) & \text{where } x \text{ is the node selected} \end{cases}$$

   Note that an incomplete branch necessarily has an α or β node selected by the above procedure; furthermore, the degree of an α or β node is greater than zero. Hence $size(\theta) > 0$ for any unclosed, incomplete branch.

   Suppose $size(\theta) = 0$. Then θ is closed or complete.

   Suppose $size(\theta) > 0$. Suppose an α node is selected. In either one or two steps, we must close or complete $\theta : \alpha_1 : \alpha_2$. Note $size(\theta : \alpha_1 : \alpha_2) < size(\theta)$ because $deg(\alpha_1) + deg(\alpha_2) < deg(\alpha)$. Hence, by the induction hypothesis, the method closes or completes the branch. Suppose a β node is selected. Then we must close or complete $\theta : \beta_1$ and $\theta : \beta_2$. Note $size(\theta : \beta_1) < size(\theta)$ because $deg(\beta_1) < deg(\beta)$; likewise, $size(\theta : \beta_2) < size(\theta)$ because $deg(\beta_2) < deg(\beta)$. Hence, by the induction hypothesis, the method closes or completes both branches.

   Hence, the method closes or completes any branch. In particular, the method closes or completes the one-point tree whose origin is $\mathsf{F}\ X$. Thus, the tableaux method terminates.

3. We prove that any downward closed set S satisfying *for all signed formulas* X*,* $X \in S$ *iff* $\overline{X} \notin S$ is a truth set.

   Let S be a downward closed set satisfying *for all signed formulas* X*,* $X \in S$ *iff* $\overline{X} \notin S$. We show that S satisfies the laws of a truth set:

   (0) Show that for any X, exactly on of X, $\overline{X}$ belongs to S.
   Let X be a signed formula. If $X \in S$, then $\overline{X} \notin S$, by the definition of S. If $X \notin S$, then $\overline{X} \in S$, by the definition of S. Hence, exactly one of X, $\overline{X}$ belongs to S.

   (a) Show $\alpha \in S$ iff $\alpha_1 \in S$ and $\alpha_2 \in S$.
   Consider α. If $\alpha \in S$, then $\alpha_1 \in S$ and $\alpha_2 \in S$, by the definition of downward closed. Suppose $\alpha_1 \in S$ and $\alpha_2 \in S$. By way of contradiction, assume $\alpha \notin S$. By the definition of S, $\overline{\alpha} \in S$. By $(J_1(a))$, $\overline{\alpha}$ is some β. Hence $\beta_1 \in S$ or $\beta_2 \in S$, by the definition of downward closed. Furthermore, by $(J_2(b))$, $\overline{\alpha_1}$ is β_1 and $\overline{\alpha_2}$ is β_2. Thus, $\overline{\alpha_1} \in S$ or $\overline{\alpha_2} \in S$. If $\overline{\alpha_1} \in S$, then $\alpha_1 \notin S$, by definition of S, but contrary to the assumption. If $\overline{\alpha_2} \in S$, then $\alpha_2 \notin S$, by the definition of S, but contrary to the assumption. Hence $\alpha \in S$.

   (b) Show $\beta \in S$ iff $\beta_1 \in S$ or $\beta_2 \in S$.
   Consider β. If $\beta \in S$, then $\beta_1 \in S$ or $\beta_2 \in S$, by the definition of downward closed. Suppose $\beta_1 \in S$ or $\beta_2 \in S$. By way of contradiction, assume $\beta \notin S$. By the definition of S, $\overline{\beta} \in S$. By $(J_1(b))$, $\overline{\beta}$ is some α. Hence $\alpha_1 \in S$ and $\alpha_2 \in S$, by the defintion of downward closed. Furthermore, by $(J_2(a))$, $\overline{\beta_1}$ is α_1 and $\overline{\beta_2}$ is α_2. Thus, $\overline{\beta_1} \in S$ and $\overline{\beta_2} \in S$. Hence $\beta_1 \notin S$ and $\beta_2 \notin S$, by the defintion of S, but contrary to the assumption. Hence, $\beta \in S$.

4. We give a recursive datatype definition for an analytic tableau.

```
Form = var: Var +
       neg: Form +
       and: Form * Form +
       or: Form * Form +
       imp: Form * Form
Sign = t: Unit + f: Unit
Tableaux = index: Nat *
           sign: Sign *
           form: Form *
           next: (closed: Int +
                  complete: Unit +
                  alpha: Int * Tableaux +
                  beta: Int * Tableaux * Tableaux)
```