Mathematical Logic for Computer Science

Second revised edition, Springer-Verlag London, 2001

Answers to Exercises

Mordechai Ben-Ari

Department of Science Teaching

Weizmann Institute of Science

Rehovot 76100 Israel

Version 1.0. 5 January 2001.

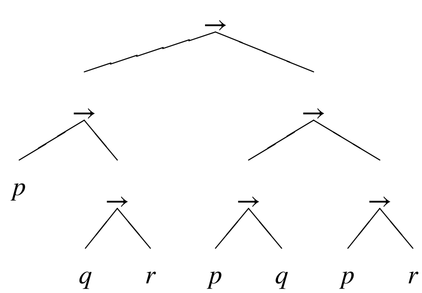
Please send comments and corrections to moti.ben-ari@weizmann.ac.il.

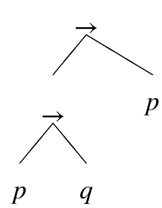
1 Introduction

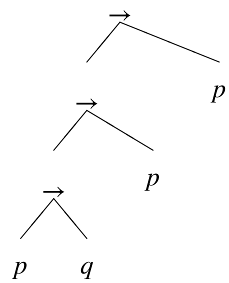
1. 'Some' is being used both as an existential quantifer and as an instantiation. Symbolically, the first statement is $\exists x(car(x) \wedge rattles(x))$, and the second is $car(mycar)$. Clearly, $rattles(mycar)$ does not follow.

2 Propositional Calculus: Formulas, Models, Tableaux

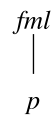
1. The truth tables can be obtained from the Prolog program. Here are the formation trees.



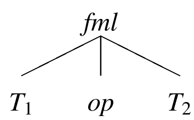




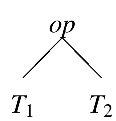
2. The proof is by an inductive construction that creates a formation tree from a derivation tree. Let *fml* be a nonterminal node with no occurrences of of *fml* below it. If the node is



for some atom p, then the only formation tree is p itself. Otherwise, suppose that the non-terminal is



where T_1 and T_2 are formation trees and *op* is a binary operator. The only fomration tree is



3. The proof is by induction on the structure of an arbitrary formula A. If A is an atom, there is no difference between an assignment and an interpretation. If $A = A_1\, op\, A_2$ is a formula with a binary operator, then by the inductive hypothesis $v(A_1)$ and $v(A_1)$ are uniquely defined, so there is a single value that can be assigned to $v(A)$ according to the table. The case for negation is similar.

   Induction is also used to prove that assignments that agree on the atoms of a formula A agree on the formula. For an atom of A, the claim is trivial, and the inductive step is straightforward.

4. Construct the truth tables for the formulas and compare that they are the same. For example, the table for the formulas in the fourth equivalence is:

A	B	$v(A \to B)$	$v(A \wedge \neg B)$	$v(\neg(A \wedge \neg B))$
T	T	T	F	T
T	F	F	T	F
F	T	T	F	T
F	F	T	F	T

where we have added an extra column for the subformula $A \wedge \neg B$.

5. By associativity and idempotence, $((p \oplus q) \oplus q) \equiv (p \oplus (q \oplus q)) \equiv p \oplus \mathit{false}$. Using the definition of $\oplus$, we find that $p \oplus \mathit{false} \equiv p$. Similarly, $((p \leftrightarrow q) \leftrightarrow q) \equiv (p \leftrightarrow (q \leftrightarrow q)) \equiv p \leftrightarrow \mathit{true} \equiv p$.

6. We prove
$$A_1 \; op \; A_2 \equiv B_1 \circ \cdots \circ B_n \equiv \neg \cdots \neg B_i$$
by induction on n. If $n = 1$, clearly B_i is either A_1 or A_2. If $n = 2$ and the definition of $\circ$ is

A_1	A_2	$A_1 \circ A_2$
T	T	F
T	F	T
F	T	F
F	F	T

then $A_1 \circ A_2 \equiv \neg A_2$, $A_2 \circ A_1 \equiv \neg A_1$, $A_1 \circ A_1 \equiv \neg A_1$, $A_2 \circ A_2 \equiv \neg A_2$, and symmetrically for

A_1	A_2	$A_1 \circ A_2$
T	T	F
T	F	F
F	T	T
F	F	T

Suppose now that
$$A_1 \; op \; A_2 \equiv (B_1 \circ \cdots \circ B_k) \circ (B_{k+1} \circ \cdots B_n).$$
By the inductive hypothesis, $B_1 \circ \cdots \circ B_k \equiv \neg \cdots \neg B_i$ and $B_{k+1} \circ \cdots \circ B_n \equiv \neg \cdots \neg B_{i'}$, where $\neg \cdots \neg B_i$ and $\neg \cdots \neg B_{i'}$ are each logically equivalent to A_1, $\neg A_1$, A_2, or $\neg A_2$. By an argument similar to that used for $n = 2$, the claim follows.

7. Let A be a formula constructed only from only p and $\wedge$ or $\vee$. We prove by induction that $A \equiv p$. Clearly, if A is an atom, A is p. Suppose that A is $A_1 \wedge A_2$ and that $A_1 \equiv A_2 \equiv p$. Then $A \equiv p \wedge p \equiv p$. Similarly, for $\vee$.

8. If $U = \{p\}$ and B is $\neg p$, then U is satisfiable, but $U \cup \{B\} = \{p, \neg p\}$ is not.

9. We prove Theorem 2.35; the others are similar. Suppose that $U - \{A_i\}$ is satisfiable and let $\mathcal{I}$ be a model. But a valid formula is true in all interpretations, so A is true in $\mathcal{I}$. Thus, $\mathcal{I}$ is a model for U, contradicting the assumption.

10.

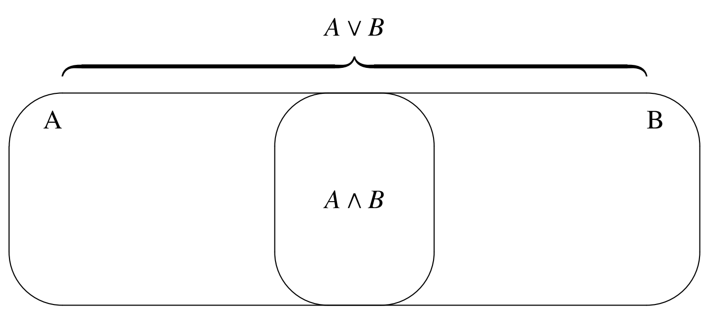
    Theorem 2.38: Any interpretation which falsifies $U = \{A_1, \ldots, A_n\}$ assigns true to $A_1 \wedge \cdots \wedge A_n \rightarrow A$ by the definition of $\wedge$ and $\rightarrow$. Any model for $U = \{A_1, \ldots, A_n\}$, assigns true to A by assumption.

    Theorem 2.39: Adding an additional assumption can only *reduce* the number of interpretations that have to satisfy A.

    Theorem 2.40: Since any interpretation satisfies a valid formulas, the set of models for U is exactly the set of models for $U - \{B\}$.

11. For $\mathcal{T}(U)$ to be closed under logical consequence means that if $\{A_1, \ldots, A_n\} \models A$ where $A_i \in \mathcal{T}(U)$ then $A \in \mathcal{T}(U)$. Let $\mathcal{I}$ be an arbitrary model for U. If $A_i \in \mathcal{T}(U)$, then $U \models A_i$, so $\mathcal{I}$ is a model for A_i for all i and $\mathcal{I}$ is a model for A. Thus $U \models A$ so $A \in \mathcal{T}(U)$.

12. A logical equivalence can be proven by replacing $\equiv$ by $\leftrightarrow$ and constructing a truth table or semantic tableau for the negation. Alternatively, truth tables can be constructed for both sides and checked for equality. The use of Venn diagrams is interesting in that it shows an equivalence between propositional logical and set theory or Boolean algebra. The Venn diagram for a proposition represents the set of interpretations for which it is true as demonstrated in the following diagram:

    How is $A \rightarrow B$ represented in a Venn diagram? $A \rightarrow B$ is logically equivalent to $\neg A \vee B$, so the diagram for $A \rightarrow B$ consists of the area *outside* A together with the area for B. $A \leftrightarrow B$ is represented by the intersection of the areas for A and B (where both are true), together with the area outside both (where both are false).

    Here are the proofs of the equivalences in terms of Venn diagrams:

    $A \rightarrow B \equiv A \leftrightarrow (A \wedge B)$: If a point is in the area for $A \rightarrow B$, it is either in the area for $A \wedge B$ or outside both. So if it is in A, it must be within $A \wedge B$.

$A \rightarrow B \equiv B \leftrightarrow (A \vee B)$: Similarly, if it is in the union of the areas, it must be within B.

$A \wedge B \equiv (A \leftrightarrow B) \leftrightarrow (A \vee B)$: Points are in both of the areas for $A \leftrightarrow B$ and $A \vee B$ iff they are within the area for $A \wedge B$.

$A \leftrightarrow B \equiv (A \vee B) \rightarrow (A \wedge B)$: If a point is in the union of the areas for A and B, it must be within the area for $A \wedge B$ if it is to be within the area for $A \leftrightarrow B$.

13. Let $W(l) = 4^{e(l)+1} \cdot (3b(l) + n(l) + 3)$, where $e(l)$ is the number of equivalence and non-equivalence operators. If $e(l)$ is decreased by one, the new $b(l)$ will be $2b(l) + 2$ and the new $n(l)$ will be at most $2n(l) + 2$. A computation will show that:
$$4^{e(l)+1} \cdot (3b(l) + n(l) + 3) > 4^{e(l)} \cdot (6b(l) + 2n(l) + 8).$$

14. We have to show that if the label of a node contains a complementary pair of formulas, then any tableau starting from that node will close (atomically). The proof is by induction. The base case is trivial. Suppose that $\{\alpha, \neg\alpha\} \subseteq U(n)$, and that we use the α-rule on α, resulting in $\{\alpha_1, \alpha_2, \neg\alpha\} \subseteq U(n')$, and then the β-rule on $\neg\alpha$, resulting in $\{\alpha_1, \alpha_2, \neg\alpha_1\} \subseteq U(n''_1)$ and $\{\alpha_1, \alpha_2, \neg\alpha_2\} \subseteq U(n''_2)$. The result follows by the inductive hypothesis. The case for $\{\beta, \neg\beta\} \subseteq U(n)$ is similar.

15. Add facts to the `alpha` and `beta` databases for the decompositions on page 32.

16. A node can become closed only if the addition of a new subformula to the label contradicts an existing one. Rather than check all elements of the label against all others, include the check for contradiction in the predicates `alpha_rule` and `beta_rule`.

3 Propositional Calculus: Deductive Systems

1.

1.	$A, B, \neg A$	Axiom
2.	$\neg B, B, \neg A$	Axiom
3.	$\neg(A \rightarrow B), B, \neg A$	$\beta\!\rightarrow$ 1, 2
4.	$\neg(A \rightarrow B), \neg\neg B, \neg A$	$\alpha\neg$ 3
5.	$\neg(A \rightarrow B), (\neg B \rightarrow \neg A)$	$\alpha\!\rightarrow$ 4
6.	$(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$	$\alpha\!\rightarrow$ 5

1.	$A, \neg A, B$	Axiom
2.	$A, \neg B, B$	Axiom
3.	$A, \neg(\neg A \rightarrow B), B$	$\beta\rightarrow$ 1, 2
4.	$\neg B, \neg A, B$	Axiom
5.	$\neg B, \neg B, B$	Axiom
6.	$\neg B, \neg(\neg A \rightarrow B), B$	$\beta\rightarrow$ 4, 5
7.	$\neg(A \rightarrow B), \neg(\neg A \rightarrow B), B$	$\beta\rightarrow$ 3, 6
8.	$\neg(A \rightarrow B), (\neg A \rightarrow B) \rightarrow B$	$\alpha\rightarrow$ 7
9.	$(A \rightarrow B) \rightarrow ((\neg A \rightarrow B) \rightarrow B)$	$\alpha\rightarrow$ 8

1.	$\neg A, B, A$	Axiom
2.	$A \rightarrow B, A$	$\alpha\rightarrow$ 1
3.	$\neg A, A$	Axiom
4.	$\neg((A \rightarrow B) \rightarrow A), A$	$\beta\rightarrow$ 2, 3
5.	$((A \rightarrow B) \rightarrow A) \rightarrow A$	$\alpha\rightarrow$ 4

2. The proof is by induction on the structure of the proof. If $\vdash U$ where U is an axiom, then U is a set of literals containing a complementary pair $\{p, \neg p\}$, that is, $U = U_0 \cup \{p, \neg p\}$. Obviously, there is a closed tableau for $\bar{U} = \bar{U}_0 \cup \{\neg p, p\}$.

   Let the last step of the proof of U be an application an α- or β-rule to obtain a formula $A \in U$; we can write $U = U_0 \cup \{A\}$. In the following, we use $\vee$ and $\wedge$ as examples for α- and β-formulas.

   Case 1: An α-rule was used on $U' = U_0 \cup \{A_1, A_2\}$ to prove $U = U_0 \cup \{A_1 \vee A_2\}$. By the inductive hypothesis, there is a closed tableau for $\bar{U}' = \bar{U}_0 \cup \{\neg A_1, \neg A_2\}$. Using the tableau α-rule, there is a closed tableau for $\bar{U} = \bar{U}_0 \cup \{\neg(A_1 \vee A_2)\}$.

   Case 2: An β-rule was used on $U' = U_0 \cup \{A_1\}$ and $U'' = U_0 \cup \{A_2\}$ to prove $U = U_0 \cup \{A_1 \wedge A_2\}$. By the inductive hypothesis, there are closed tableaux for $\bar{U}' = \bar{U}_0 \cup \{\neg A_1\}$ and $\bar{U}'' = \bar{U}_0 \cup \{\neg A_2\}$. Using the tableau β-rule, there is a closed tableau for $\bar{U} = \bar{U}_0 \cup \{\neg(A_1 \wedge A_2)\}$.

3.

1.	$\vdash (A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$	Theorem 3.24
2.	$\vdash (A \rightarrow B)$	Assumption
3.	$\vdash \neg B \rightarrow \neg A$	MP 1, 2
4.	$\vdash \neg B$	Assumption
5.	$\vdash A$	MP 3, 4

4.

1.	$\neg A, \neg B, A$	Axiom
2.	$\neg A, B \rightarrow A$	$\alpha\rightarrow$ 1
3.	$A \rightarrow (B \rightarrow A)$	$\alpha\rightarrow$ 2

   For Axiom 2 we will use a shortcut by taking as an axiom any set of *formulas* containing a complementary pair of literals.

1.	$B, A, \neg A, C$	Axiom
2.	$B, \neg B, \neg A, C$	Axiom
3.	$B, \neg(A \rightarrow B), \neg A, C$	$\beta\!\rightarrow$ 1, 2
4.	$\neg C, \neg(A \rightarrow B), \neg A, C$	Axiom
5.	$\neg(B \rightarrow C), \neg(A \rightarrow B), \neg A, C$	$\beta\!\rightarrow$ 3, 4
6.	$A, \neg(A \rightarrow B), \neg A, C$	Axiom
7.	$\neg(A \rightarrow (B \rightarrow C)), \neg(A \rightarrow B), \neg A, C$	$\beta\!\rightarrow$ 5, 6
8.	$\neg(A \rightarrow (B \rightarrow C)), \neg(A \rightarrow B), A \rightarrow C$	$\alpha\!\rightarrow$ 7
9.	$\neg(A \rightarrow (B \rightarrow C)), (A \rightarrow B) \rightarrow (A \rightarrow C)$	$\alpha\!\rightarrow$ 8
10.	$(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$	$\alpha\!\rightarrow$ 9

The proof of Axiom 3 is similar to the proof of $(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$ from exercise 1.

5.

1.	$\{\neg A \rightarrow A\} \vdash \neg A \rightarrow A$	Assumption
2.	$\{\neg A \rightarrow A\} \vdash \neg A \rightarrow \neg\neg A$	Contrapositive
3.	$\{\neg A \rightarrow A\} \vdash (\neg A \rightarrow \neg\neg A) \rightarrow \neg\neg A$	Theorem 3.28
4.	$\{\neg A \rightarrow A\} \vdash \neg\neg A$	MP 2, 3
5.	$\{\neg A \rightarrow A\} \vdash A$	Double negation
6.	$\vdash (\neg A \rightarrow A) \rightarrow A$	Deduction

6.

1.	$\vdash A \rightarrow (\neg A \rightarrow B)$	Theorem 3.21
2.	$\vdash A \rightarrow (A \vee B)$	Definition of $\vee$

1.	$\vdash B \rightarrow (B \vee A)$	Just proved
2.	$\vdash B \rightarrow (A \vee B)$	Theorem 3.32

The proof of Theorem 3.32 does not use this theorem so it can be used here.

1.	$\vdash (\neg C \rightarrow (A \rightarrow B)) \rightarrow ((\neg C \rightarrow A) \rightarrow (\neg C \rightarrow B))$	Axiom 2
2.	$\vdash (C \vee (A \rightarrow B)) \rightarrow ((C \vee A) \rightarrow (C \vee B))$	Definition of $\vee$
3.	$\vdash (A \rightarrow B) \rightarrow (C \vee (A \rightarrow B))$	Just proved
4.	$\vdash (A \rightarrow B) \rightarrow ((C \vee A) \rightarrow (C \vee B))$	Transitivity

7. Of course, $\leftrightarrow$ should be $\rightarrow$.

1.	$\{(A \vee B) \vee C\} \vdash (A \vee B) \vee C$	Assumption
2.	$\{(A \vee B) \vee C\} \vdash \neg(\neg A \rightarrow B) \rightarrow C$	Definition of $\vee$
3.	$\{(A \vee B) \vee C\} \vdash \neg C \rightarrow (\neg A \rightarrow B)$	Contrapositive
4.	$\{(A \vee B) \vee C\} \vdash \neg A \rightarrow (\neg C \rightarrow B)$	Exchange
5.	$\{(A \vee B) \vee C\} \vdash (\neg C \rightarrow B) \rightarrow (\neg B \rightarrow C)$	Contrapos., double neg.
6.	$\{(A \vee B) \vee C\} \vdash \neg A \rightarrow (\neg B \rightarrow C)$	Transitivity
7.	$\{(A \vee B) \vee C\} \vdash A \vee (B \vee C)$	Definition of $\vee$
8.	$\vdash (A \vee B) \vee C \rightarrow A \vee (B \vee C)$	Deduction

8. The proofs are trivial.

9. The second node below is obtained by applying the α-rule for $\rightarrow$ three times.

$$
\begin{array}{c}
\neg((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))) \\
\downarrow \\
A \rightarrow (B \rightarrow C), A \rightarrow B, A, \neg C \\
\swarrow \qquad\qquad \searrow \\
A \rightarrow (B \rightarrow C), \neg A, A, \neg C \qquad A \rightarrow (B \rightarrow C), B, A, \neg C \\
\times \qquad\qquad \swarrow \qquad \searrow \\
\neg A, B, A, \neg C \qquad B \rightarrow C, B, A, \neg C \\
\times \qquad\qquad \swarrow \qquad \searrow \\
\neg B, B, A, \neg C \qquad C, B, A, \neg C \\
\times \qquad\qquad \times
\end{array}
$$

10. Let $(A_1, \ldots, A_n)$ be the elements of $U - U'$ in some order.

1.	$\vdash \bigvee U'$	Assumption
2.	$\vdash \bigvee U' \vee A_1$	Theorem 3.31
	$\cdots$	
n+1.	$\vdash \bigvee U' \vee A_1 \vee \cdots \vee A_n$	Theorem 3.31

So we have to prove that if U' is a permutation of U and $\vdash \bigvee U$ then $\vdash \bigvee U'$. The proof is by induction on n the number of elements in U. If $n = 1$, there is nothing to prove, and if $n = 2$, the result follows immediately from Theorem 3.32. Let $\bigvee U = \bigvee U_1 \vee \bigvee U_2$ and $\bigvee U' = \bigvee U'_1 \vee \bigvee U'_2$ have n elements. If U'_1 and U'_2 are permutations of U_1 and U_2, respectively, then the result follows by the inductive hypothesis and Theorem 3.31. Otherwise, without loss of generality, suppose that there is an element A of U'_2 which is in U_1. Suppose that $\bigvee U'_2 = A \vee \bigvee U''_2$, so that $\bigvee U' = \bigvee U'_1 \vee (A \vee \bigvee U''_2)$. Then by Theorem 3.33, $\bigvee U' = (\bigvee U'_1 \vee A) \vee \bigvee U''_2$. Thus all we have to prove is that $A_1 \vee \cdots \vee A_i \vee \cdots \vee A_k$ can be written $A_i \vee A_1 \vee \cdots \vee A_{i-1} \vee A_{i+1} \vee \cdots \vee A_k$ for arbitrary i. This is proved by a simply induction using Theorem 3.33.

11. The first formula was proved in Theorem 3.24.

1.	$\{A \rightarrow B, \neg A \rightarrow B\} \vdash \neg A \rightarrow B$	Assumption
2.	$\{A \rightarrow B, \neg A \rightarrow B\} \vdash \neg B \rightarrow A$	Contrapositive
3.	$\{A \rightarrow B, \neg A \rightarrow B\} \vdash A \rightarrow B$	Assumption
4.	$\{A \rightarrow B, \neg A \rightarrow B\} \vdash \neg B \rightarrow B$	Transitivity
5.	$\{A \rightarrow B, \neg A \rightarrow B\} \vdash (\neg B \rightarrow B) \rightarrow B$	Theorem 3.29
6.	$\{A \rightarrow B, \neg A \rightarrow B\} \vdash B$	MP 4, 5
7.	$\{A \rightarrow B\} \vdash (\neg A \rightarrow B) \rightarrow B$	Deduction
8.	$\vdash (A \rightarrow B) \rightarrow ((\neg A \rightarrow B) \rightarrow B)$	Deduction

1.	$\{(A \to B) \to A\} \vdash (A \to B) \to A$	Assumption
2.	$\{(A \to B) \to A\} \vdash \neg A \to (A \to B)$	Theorem 3.20
3.	$\{(A \to B) \to A\} \vdash \neg A \to A$	Transitivity
4.	$\{(A \to B) \to A\} \vdash (\neg A \to A) \to A$	Theorem 3.29
5.	$\{(A \to B) \to A\} \vdash A$	MP 3, 4
6.	$\vdash (A \to B) \to A) \to A$	Deduction

12. The deduction theorem can be used because its proof only uses Axioms 1 and 2.

1.	$\{\neg B \to \neg A, A\} \vdash \neg B \to \neg A$	Assumption
2.	$\{\neg B \to \neg A, A\} \vdash (\neg B \to \neg A) \to ((\neg B \to A) \to B)$	Axiom 3'
3.	$\{\neg B \to \neg A, A\} \vdash (\neg B \to A) \to B$	MP 1,2
4.	$\{\neg B \to \neg A, A\} \vdash A \to (\neg B \to A)$	Axiom 1
5.	$\{\neg B \to \neg A, A\} \vdash A$	Assumption
6.	$\{\neg B \to \neg A, A\} \vdash \neg B \to A$	MP 4,5
7.	$\{\neg B \to \neg A, A\} \vdash B$	MP 6,3
8.	$\{\neg B \to \neg A\} \vdash A \to B$	Deduction
9.	$\vdash (\neg B \to \neg A) \to (A \to B)$	Deduction

13. It follows from Definition 3.47 that a sequent $\{U_1, \ldots, U_n\} \Rightarrow \{V_1, \ldots, V_m\}$ is true iff $\neg U_1 \vee \cdots \vee \neg U_n \vee V_1 \vee \cdots \vee V_m$ is true. The completeness of $\mathcal{S}$ follows from the completeness of $\mathcal{G}$ by showing that the rules of the two are the same. For example,

$$\frac{U \cup \{A\} \Rightarrow V \cup \{B\}}{U \Rightarrow V \cup \{A \to B\}}$$

is

$$\frac{\neg U_1 \vee \cdots \neg U_m \vee \neg A \vee V_1 \vee \cdots \vee V_n \vee B}{\neg U_1 \vee \cdots \neg U_m \vee V_1 \vee \cdots \vee V_n \vee (A \to B)}$$

which is the α-rule for $\to$, and

$$\frac{U \Rightarrow V \cup \{A\} \qquad U \cup \{B\} \Rightarrow V}{U \cup \{A \to B\} \Rightarrow V}$$

is

$$\frac{\neg U_1 \vee \cdots \neg U_m \vee V_1 \vee \cdots \vee V_n \vee A \qquad \neg U_1 \vee \cdots \neg U_m \vee \neg B \vee V_1 \vee \cdots \vee V_n}{\neg U_1 \vee \cdots \neg U_m \vee \neg(A \to B) \vee V_1 \vee \cdots \vee V_n}$$

which is the β-rule for $\to$. We leave the check of the other rules to the reader.

14. If $\vdash \neg A_1 \vee \cdots \vee \neg A_n$ then clearly $U \vdash \neg A_1 \vee \cdots \vee \neg A_n$, since we do not need to use the assumptions. But that is the same as $U \vdash A_1 \to (A_2 \to \cdots \to \neg A_n)$. Now $U \vdash A_i$ is trivial, so by $n-1$ applications of modus ponens, $U \vdash \neg A_n$, which together with $U \vdash A_n$, prove that U is inconsistent.

Conversely, if U is inconsistent, then $U \vdash A$ and $U \vdash \neg A$ for some A. But only a finite number of formulas $\{A_1, \ldots, A_n\} \subseteq U$ are used in either one of the proofs, so $\{A_1, \ldots, A_n\} \vdash A$ and $\{A_1, \ldots, A_n\} \vdash \neg A$. By n applications of the deduction theorem, $\vdash A_1 \rightarrow \cdots \rightarrow A_n \rightarrow \neg A$ and $\vdash A_1 \rightarrow \cdots \rightarrow A_n \rightarrow A$. From propositional reasoning, $\vdash A_1 \rightarrow \cdots \rightarrow A_n \rightarrow \mathit{false}$, and $\vdash \neg A_1 \vee \cdots \vee \neg A_n$.

15.

(a) Assume that $U \subseteq S$ is finite and unsatisfiable. Then $\neg \bigwedge U$ is valid, so $\vdash \neg \bigwedge U$ by completeness (Theorem 3.35). By repeated application of $\vdash A \rightarrow (B \rightarrow (A \wedge B))$ (Theorem 3.30), $U \vdash \bigwedge U$. But $S \cup U = S$ and you can always add unused assumptions to a proof so $S \vdash \neg \bigwedge U$ and $S \vdash \bigwedge U$, contradicting the assumption that S is consistent.

(b) Assume that $S \cup \{A\}$ and $S \cup \{\neg A\}$ are both inconsistent. By Theorem 3.41, both $S \vdash A$ and $S \vdash \neg A$, so S is inconsistent, contradicting the assumption.

(c) S can be extended to a maximally consistent set.

Consider an enumeration $\{A_1, \ldots, \}$ of all propositional formulas. Let $S_0 = S$ and for each i let $S_{i+1} = S_i \cup \{A_i\}$ or $S_{i+1} = S_i \cup \{\neg A_i\}$, whichever is consistent. (We have just proved that one of them must be.) Let $S' = \bigcup_{i=0}^{\infty} S_i$. First we prove that S' is consistent. If S' is inconsistent, then $S' \vdash p \wedge \neg p$ (for example) by Theorem 3.39. But only a finite number of elements of S' are used in the proof, so there must be some large enough i such that S_i includes them all. Then $S_i \vdash p \wedge \neg p$ contracting the consistency of S_i. To prove that S' is maximally consistent, suppose that $B \notin S'$. By construction, $\neg B \in S_i \subset S'$ for some i, so $S' \cup \{B\} \vdash \neg B$. But trivially, $S' \cup \{B\} \vdash B$, so $S' \cup \{B\}$ is inconsistent.

16. I would be pleased if someone would contribute a program!

4 Propositional Calculus: Resolution and BDDs

1. This theorem can be proved both syntactically and semantically. The syntactic proof uses the same construction as the one for CNF except that the distributive laws used are: $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$ and $(A \vee B) \wedge C \equiv (A \wedge C) \vee (B \wedge C)$. For the semantic proof, start by constructing a truth table for the formula A. For each line in the truth table that evaluates to T, construct a conjunction with the literal p if p is assigned T in that row and $\bar{p}$ if p is assigned F. Let A' be the disjunction of all these conjunctions. Then $A \equiv A'$. Let v be an arbitrary model for A, that is, $v(A) = T$. Then the assignments in v are such that the row of the truth table contains T, and by construction $v(C) = T$ for the conjunction C built from that row. Since A' is a disjunction of such conjunctions, it is sufficient for one of them to be true for A' to be true. Hence $v(A') = T$. For the converse, if v is an arbitrary interpretation so that $v(A') = T$, then by the structure of A', $v(C) = T$ for at least one conjunction C in A' (in fact, for exactly one such conjunction). But for this assignment, $v(A) = T$ by construction.

2. The formula constructed in the previous exercise is in complete DNF.

3. I would be pleased if someone would contribute a program!

4. This exercise is rather trivial because the sets of clauses are satisfiable and for S a set of satisfiable clauses, $S \approx \{\}$ the valid empty set of clauses. For each of the sets, we give a sequence of sets obtained by using the various lemmas.

$$\{p\bar{q},\ q\bar{r},\ rs,\ p\bar{s}\} \approx \{q\bar{r},\ rs\} \approx \{q\bar{r}\} \approx \{\,\},$$
$$\{pqr,\ \bar{q},\ p\bar{r}s,\ qs,\ p\bar{s}\} \approx \{pr,\ p\bar{r}s,\ s,\ p\bar{s}\} \approx \{pr,\ p\} \approx \{p\} \approx\approx \{\,\},$$
$$\{pqrs,\ \bar{q}rs,\ \bar{p}rs,\ qs,\ \bar{p}s\} \approx \{\bar{q}rs,\ \bar{p}rs,\ qs,\ \bar{p}s\} \approx \{\,\},$$
$$\{\bar{p}q,\ qrs,\ \bar{p}\bar{q}rs,\ \bar{r},\ q\} \approx \{\bar{p}rs,\ \bar{r}\} \approx \{\bar{r}\} \approx \{\,\}.$$

5.

Refutation 1:

5.	$\bar{q}r$	1, 2
6.	r	3, 5
7.	$\Box$	4, 6

Refutation 2:

5.	$\bar{p}\bar{q}$	1, 4
6.	p	2, 4
7.	$\bar{q}$	5, 6
8.	q	3, 4
9.	$\Box$	7, 8

6. The clausal form of the set is:

$$(1)\,p,\ (2)\,\bar{p}qr,\ (3)\,\bar{p}\bar{q}\bar{r},\ (4)\,\bar{p}st,\ (5)\,\bar{p}\bar{s}\bar{t},\ (6)\,\bar{s}q,\ (7)\,rt,\ (8)\,\bar{t}s.$$

A refutation is:

9.	$\bar{p}\bar{s}r$	5, 7
10.	$\bar{p}s$	4, 8
11.	$\bar{p}\bar{q}\bar{s}$	3, 9
12.	$\bar{p}\bar{s}$	11, 6
13.	$\bar{p}$	12, 10
14.	$\Box$	13, 1

7. The clausal form of the formulas is:

$$\{(1)\,\bar{s}\bar{b}1\bar{b}2,\ (2)\,\bar{s}b1b2,\ (3)\,s\bar{b}1b2,\ (4)\,sb1\bar{b}2,\ (5)\,\bar{c}b1,\ (6)\,\bar{c}b2,\ (7)\,c\bar{b}1\bar{b}2\}.$$

The addition of the set of clauses: $\{(8)\,b1,\ (9)\,b2,\ (10)\,\bar{s},\ (11)\,\bar{c}\}$, enables a refutation to be done by resolving clauses $11, 7, 8, 9$. The addition of the clauses $\{(8)\,b1,\ (9)\,b2,\ (10)\,\bar{s},\ (11)\,c\}$ gives a satisfiable set by assigning F to s and T to all other atoms (check!). The meaning of the satisfiable set is that $1 \oplus 1 = 0$ carry 1, by identifying 1 with T and 0 with F. The unsatisfiable set shows that it is not true that $1 \oplus 1 = 0$ carry 0.

8. The statement of the claim should say: Prove adding a unit clause to a set of clauses *such that the atom of the unit clause does not already appear in the set* and ....

   Let S be the original set of clauses and S' the new set of clauses obtained by adding $\{l\}$ to S and l^c to every other clause in S, and let v be a model for S. Extend v to v' by defining $v'(p) = T$ if $l = p$, $v'(p) = F$ if $l^c = p$, and v' is the same as v on all other atoms. (Here is where we need the proviso on the new clause.) By construction, $v(l) = T$ so the additional clause in S' is satisfied. For every other clause C, $v(C) = v'(C) = T$ since the addition of a literal to a clause (which is a disjunction) cannot falsify it.

9. By induction on the depth of the resolution tree. If the depth of the tree is 1, the result is immediate from Theorem 4.24. If the depth of the tree is n, then the children of the root are satisfiable by the inductive hypothesis, so the root is satisfiable by Theorem 4.24.

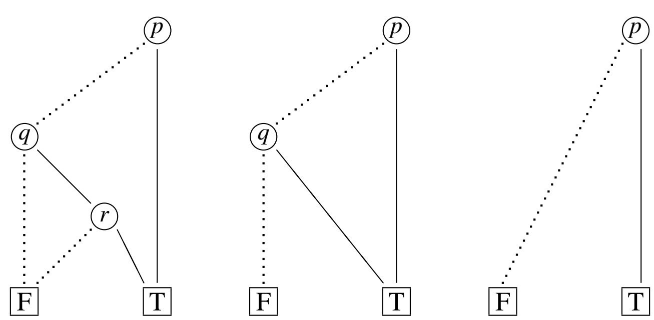
10. First prove a lemma: for any v, $v(A_1|_{p=T}\ op\ A_2|_{p=T}) = v(A_1\ op\ A_2)$ if $v(p) = T$ and $v(A_1|_{p=F}\ op\ A_2|_{p=F}) = v(A_1\ op\ A_2)$ if $v(p) = F$. The proof is by structural induction. Clearly, $v(p|_{p=T}) = T = v(p)$ and $q|_{p=T}$ is the same formula as q for $q \neq p$. Suppose now that $v(p) = T$. By the inductive hypothesis, $v(A_1|_{p=T}) = v(A_1)$ and $v(A_2|_{p=T}) = v(A_2)$ so by the semantic definitions of the operators, $v(A_1|_{p=T}\ op\ A_2|_{p=T}) = v(A_1\ op\ A_2)$. A similar argument holds for F.

    We can now prove the Shannon expansion. Let v be an arbitrary interpretation. If $v(p) = T$,

    $$v(\,(p \wedge (A_1|_{p=T}\ op\ A_2|_{p=T})) \vee (\neg p \wedge (A_1|_{p=F}\ op\ A_2|_{p=F}))\,) = v(A_1|_{p=T}\ op\ A_2|_{p=T}),$$

    which equals $v(A_1\ op\ A_2)$ by the lemma, and similarly if $v(p) = F$. Since v was arbitrary, the formulas are logically equivalent.

11. From Example 4.62, the BDDs for $p \vee (q \wedge r)$ and for $A|_{r=T}$ and $A|_{r=F}$ are:



    Using the algorithm `apply` with $\vee$ gives the middle BDD above for $p \vee q$: recursing on the left subBDD gives $q \vee F$ which is q and recursing on the right subBDD is clearly T. Using the algorithm `apply` with $\wedge$ gives the right BDD above for p: recursing on the left subBDD gives the controlling operand F for $\wedge$ and recursing on the right subBDD is clearly T.

12. The programs in the software archive implement the optimizations.

13. Let us number the clauses as follows:

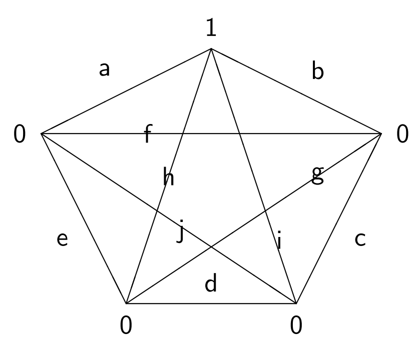
$$(1)\,\bar{p}q,\ (2)\,p\bar{q},\ (3)\,prs,\ (4)\,\bar{p}\bar{r}s,\ (5)\,\bar{p}r\bar{s},\ (6)\,p\bar{r}\bar{s},$$
$$(7)\,\bar{s}t,\ (8)\,s\bar{t},\ (9)\,\bar{q}rt,\ (10)\,q\bar{r}t,\ (11)\,qr\bar{t},\ (12)\,\bar{q}\bar{r}\bar{t}.$$

The refutation is not for the faint-hearted....

13.	qrs	1, 3
14.	qst	13, 10
15.	$\bar{q}\bar{r}s$	2, 4
16.	$\bar{q}st$	15, 9
17.	st	14, 16
18.	s	17, 8
19.	$q\bar{r}\bar{s}$	1, 6
20.	$q\bar{s}\bar{t}$	19, 11
21.	$\bar{q}r\bar{s}$	2, 5
22.	$\bar{q}\bar{s}\bar{t}$	21, 12
23.	$\bar{s}\bar{t}$	20, 22
24.	$\bar{s}$	23, 7
25.	$\square$	18, 24

14. This is trivial as all the leaves are labeled false.

15. Here is the complete graph with the edges and vertices labeled.



The clauses are the eight even-parity clauses on the atoms *abhi* and the eight odd-parity clauses on each of *aefj*, *degh*, *cdij* and *bcfg*, forty clauses in all. We leave the construction of a resolution refutation to the reader.

16. Suppose $\Pi(v_n) = b_n$ and let $C = l_1 \cdots l_k$ be an arbitrary clause associated with n. Then C can is only falsified by the assignment

$$v(p_i) = F \text{ if } l_i = p_i \;\text{ and }\; v(p_i) = T \text{ if } l_i = \bar{p}_i.$$

Then

$\Pi(C)$	=	(by definition)
parity of negated atoms of C	=	(by construction)
parity of literals assigned T	=	(by definition)
$\Pi(v_n)$	=	(by assumption)
b_n,		

which contradicts the assumption that $C \in C(n)$. Thus if $\Pi(v_n) = b_n$, v must satisfy all clauses in $C(n)$.

17. Decision procedure for satisfiability of *sets* of formulas whose only operators are $\neg$, $\leftrightarrow$ and $\oplus$:

    - Use $p \oplus q \equiv \neg(p \leftrightarrow q) \equiv \neg\, p \leftrightarrow q$ and $\neg\,\neg\, p \equiv p$ to reduce the formulas to the form $q_1 \leftrightarrow \cdots \leftrightarrow q_n$.
    - Use commutativity, associativity and:
      $p \leftrightarrow p \equiv \mathit{true}$, $p \leftrightarrow \neg\, p \equiv \mathit{false}$, $p \leftrightarrow \mathit{true} \equiv p$, $p \leftrightarrow \mathit{false} \equiv \neg\, p$,
      to reduce the formulas to equivalences on distinct atoms.
    - If all formulas reduce to *true*, the set is valid. If some formula reduces to *false*, the set is unsatisfiable.
    - Otherwise, delete all formulas which reduce to *true*. Transform the formulas as in the lemma. Let $\{p_1, \ldots, p_m\}$ be the new atoms. Assign *true* to p_1 and each q_j^1 to which it is equivalent. By induction, assign *true* to p_i, unless some q_j^i has already been assigned to; if so, assign its value to p_i. If p_i has already been assigned a clashing value, the set of formulas is unsatisfiable.

    Each of the three steps increases the size of the formula by a small polynomial, so the procedure is efficient.

5 Predicate Calculus: Formulas, Models, Tableaux

1. A falsifying interpretation is $(\{1, 2\}, \{\{2\}\}, \{1\}\}$. Then $v(p(2)) = T$ so $v(\exists\, xp(x)) = T$, but $v(p(a)) = v(p(1)) = F$.

2. We will prove the validity by constructing a closed semantic tableau for the negation of each formula. To simply formatting, a linear representation will be

used for tableaux: 1 will number the first or only child and 2 the second child. Copying of the universal formulas will be omitted as it is not necessary to prove these formulas.

(a) For this formula we prove the implication for each direction separately.

1.	$\neg[\exists x(A(x) \rightarrow B(x)) \rightarrow (\forall x A(x) \rightarrow \exists x B(x))]$	$\alpha \rightarrow$
11.	$\exists x(A(x) \rightarrow B(x)), \neg(\forall x A(x) \rightarrow \exists x B(x))$	$\alpha \rightarrow$
111.	$\exists x(A(x) \rightarrow B(x)), \forall x A(x), \neg \exists x B(x))$	δ
1111.	$A(a) \rightarrow B(a), \forall x A(x), \neg \exists x B(x))$	$\beta \rightarrow$
11111.	$\neg A(a), \forall x A(x), \neg \exists x B(x))$	γ
111111.	$\neg A(a), A(a), \neg B(a)$	$\times$
11112.	$B(a), \forall x A(x), \neg \exists x B(x)$	γ
111121.	$B(a), A(a), \neg B(a)$	$\times$

1.	$\neg[(\forall x A(x) \rightarrow \exists x B(x)) \rightarrow \exists x(A(x) \rightarrow B(x))]$	$\alpha \rightarrow$
11.	$\forall x A(x) \rightarrow \exists x B(x), \neg \exists x(A(x) \rightarrow B(x))$	$\beta \rightarrow$
111.	$\neg \forall x A(x), \neg \exists x(A(x) \rightarrow B(x))$	δ
1111.	$\neg A(a), \neg \exists x(A(x) \rightarrow B(x))$	γ
11111.	$\neg A(a), \neg(A(a) \rightarrow B(a))$	$\alpha \rightarrow$
111111.	$\neg A(a), A(a), \neg B(a)$	$\times$
112.	$\exists x B(x), \neg \exists x(A(x) \rightarrow B(x))$	δ
1121.	$B(a), \neg \exists x(A(x) \rightarrow B(x))$	γ
11211.	$B(a), \neg(A(a) \rightarrow B(a))$	$\alpha \rightarrow$
112111.	$B(a), A(a), \neg B(a)$	$\alpha \rightarrow$

(b)

1.	$\neg[(\exists x A(x) \rightarrow \forall x B(x)) \rightarrow \forall x(A(x) \rightarrow B(x))]$	$\alpha \rightarrow$
11.	$\exists x A(x) \rightarrow \forall x B(x), \neg \forall x(A(x) \rightarrow B(x))$	$\beta \rightarrow$
111.	$\neg \exists x A(x), \neg \forall x(A(x) \rightarrow B(x))$	δ
1111.	$\neg \exists x A(x), \neg(A(a) \rightarrow B(a))$	$\alpha \rightarrow$
11111.	$\neg \exists x A(x), A(a), \neg B(a)$	γ
111111.	$\neg A(a), A(a), \neg B(a)$	$\times$
112.	$\forall x B(x), \neg \forall x(A(x) \rightarrow B(x))$	δ
1121.	$\forall x B(x), \neg(A(a) \rightarrow B(a))$	$\alpha \rightarrow$
11211.	$\forall x B(x), A(a), \neg B(a)$	γ
112111.	$B(a), A(a), \neg B(a)$	$\times$

(c)

1.	$\neg[\forall x(A(x) \vee B(x)) \rightarrow (\forall x A(x) \vee \exists x B(x))]$	$\alpha \rightarrow$
11.	$\forall x(A(x) \vee B(x)), \neg(\forall x A(x) \vee \exists x B(x))$	$\alpha \vee$
111.	$\forall x(A(x) \vee B(x)), \neg \forall x A(x), \neg \exists x B(x))$	δ
1111.	$\forall x(A(x) \vee B(x)), \neg A(a), \neg \exists x B(x))$	γ
11111.	$A(a) \vee B(a), \neg A(a), \neg B(a)$	$\beta \vee$
111111.	$A(a), \neg A(a), \neg B(a)$	$\times$
111112.	$B(a), \neg A(a), \neg B(a))$	$\times$

(d)

1.	$\neg[\forall x(A(x) \rightarrow B(x)) \rightarrow (\exists x A(x) \rightarrow \exists x B(x))]$	$\alpha \rightarrow$
11.	$\forall x(A(x) \rightarrow B(x)), \neg(\exists x A(x) \rightarrow \exists x B(x))$	$\alpha \rightarrow$
111.	$\forall x(A(x) \rightarrow B(x)), \exists x A(x), \neg \exists x B(x))$	δ
1111.	$\forall x(A(x) \rightarrow B(x)), A(a), \neg \exists x B(x))$	γ
11111.	$A(a) \rightarrow B(a), A(a), \neg B(a)$	$\beta \rightarrow$
111111.	$\neg A(a), A(a), \neg B(a)$	$\times$
111112.	$B(a), A(a), \neg B(a)$	$\times$

3. We will show this for (b), (c) and (d) of the previous exercise.

   (b) $(\{1,2\}, \{\{1\},\{1\}\}, \{\})$. Since A and B are interpreted by the same relation, $\forall x(A(x) \rightarrow B(x))$ is true, as is $\exists x A(x)$, but $\forall x B(x))$ is not when x is assigned 2.

   (c) For the same interpretation, $\forall x A(x) \vee \exists x B(x)$ is true, because $\exists x B(x)$ is true, but $\forall x(A(x) \vee B(x))$ is false if x is assigned 2.

   (d) $(\{1,2\}, \{\{1\},\{2\}\}, \{\})$. $\exists x A(x) \rightarrow \exists x B(x)$ is true, because there are assignments making each of $A(x)$ and $B(x)$ true, but assigning 1 to x makes $A(x) \rightarrow B(x)$ false.

4.

   (a) A falsifying interpretation must cause the negation

$$\forall x \exists y((p(x,y) \wedge \neg p(y,x)) \wedge (p(x,x) \oplus p(y,y))$$

to be true. Let the domain be the positive natural numbers and the predicate $p(x,y)$ be interpreted by $\lfloor\sqrt{x}\rfloor^2 < y$, where $\lfloor a \rfloor$ is the largest integer less than a. For every x, we have to produce a y such that the formula is true. If x is the square of a natural number, let $y = x+1$. Otherwise, let $y = \lceil\sqrt{x}\,\rceil^2$, the square of the smallest number larger than x. For example, pairs for which p is true include:

$$(1,2), (2,4), (3,4), (4,5), (5,9), (6,9), \ldots.$$

Case 1: x is a square (hence $x+1$ is not) and $y = x+1$.

$\lfloor\sqrt{x}\rfloor^2 = x = y-1 < y$, so $p(x,y)$ is true.
$\lfloor\sqrt{y}\rfloor^2 = \lfloor\sqrt{x+1}\rfloor^2 = x \not< x$, so $p(y,x)$ is false.
$\lfloor\sqrt{x}\rfloor^2 = x \not< x$, so $p(x,x)$ is false.
$\lfloor\sqrt{y}\rfloor^2 = \lfloor\sqrt{x+1}\rfloor^2 < x+1 = y$, so $p(y,y)$ is true.

Case 2: x is not a square and $y = \lceil\sqrt{x}\,\rceil^2$.

$\lfloor\sqrt{x}\rfloor^2 < \lceil\sqrt{x}\,\rceil^2 = y$, so $p(x,y)$ is true.
$\lfloor\sqrt{\lceil\sqrt{x}\,\rceil^2}\rfloor^2 = \lceil\sqrt{x}\,\rceil^2 = \not< x$, so $p(y,x)$ is false.
$\lfloor\sqrt{x}\rfloor^2 < x$, so $p(x,x)$ is true.
$\lfloor\sqrt{\lceil\sqrt{x}\,\rceil^2}\rfloor^2 = \lceil\sqrt{x}\,\rceil^2 = y \not< y$, so $p(y,y)$ is false.

The truth of the matrix of the above formula follows from these truth values.

(b) In the domain of the integers with $\leq$ assigned to p, the consequent states that there is a lower bound which is false.

5. Suppose that $v_{\mathcal{I}}(\forall x_1 \cdots \forall x_n A') = T$ and let $\sigma_{\mathcal{I}}$ be an arbitrary assignment which assigns $(d_1, \ldots, d_n)$ to $(x_1, \ldots, x_n)$, for $d_i \in D$. But $v_{\sigma_{\mathcal{I}}[x_1 \leftarrow d_1, \ldots, x_1 \leftarrow d_1]}(A') = T$ by assumption and $\sigma_{\mathcal{I}}[x_1 \leftarrow d_1, \ldots, x_n \leftarrow d_n] = \sigma_{\mathcal{I}}$, so $v_{\sigma_{\mathcal{I}}}(A') = T$. The converse is trivial and the proof for the existential case is similar.

6.

1.	$\neg[\forall x(p(x) \vee q(x)) \rightarrow (\forall x p(x) \vee \forall x q(x))]$	$\alpha \rightarrow$
11.	$\forall x(p(x) \vee q(x)), \neg(\forall x p(x) \vee \forall x q(x))$	$\alpha \vee$
111.	$\forall x(p(x) \vee q(x)), \neg \forall x p(x), \neg \forall x q(x)$	δ
1111.	$\forall x(p(x) \vee q(x)), \neg p(b), \neg \forall x q(x)$	δ
11111.	$\forall x(p(x) \vee q(x)), \neg p(b), \neg q(a)$	γ
111111.	$p(a) \vee q(a), p(b) \vee q(b), \neg p(b), \neg q(a)$	$\beta \vee$
1111111.	$p(a), p(b) \vee q(b), \neg p(b), \neg q(a)$	$\beta \vee$
1111112.	$q(a), p(b) \vee q(b), \neg p(b), \neg q(a)$	$\times$
11111111.	$p(a), p(b), \neg p(b), \neg q(a)$	$\times$
11111112.	$p(a), q(b), \neg p(b), \neg q(a)$	$\odot$

7.

1.	$\neg[(\forall x p(x) \rightarrow \forall x q(x)) \rightarrow \forall x(p(x) \rightarrow q(x))]$	$\alpha \rightarrow$
11.	$\forall x p(x) \rightarrow \forall x q(x),\ \neg \forall x(p(x) \rightarrow q(x))$	$\alpha \rightarrow$
111.	$\forall x p(x),\ \neg \forall x q(x),\ \neg \forall x(p(x) \rightarrow q(x))$	δ
1111.	$\forall x p(x),\ \neg \forall x q(x),\ \neg(p(a) \rightarrow q(a)$	$\alpha \rightarrow$
11111.	$\forall x p(x),\ \neg \forall x q(x),\ p(a),\ \neg q(a)$	δ
111111.	$\forall x p(x),\ \neg q(b),\ p(a),\ \neg q(a)$	γ
1111111.	$p(a),\ p(b),\ \neg q(b),\ p(a),\ \neg q(a)$	$\odot$

8. Note that in the text the label Hintikka's Lemma is attached to Theorm 5.32 instead of to Lemma 5.33.

   The proof is by structural induction. The base case and the cases for the Boolean operators are the same as for the propositional calculus. Suppose that $A = \forall x A'(x)$ is a γ formula in U. By condition 4 of the Hintikka set, $A'(a) \in U$ for all constants a. By the inductive hypothesis, there is an interpretation $\mathcal{I}$ in which all formulas $A'(a)$ are true. So $v_{\sigma_{\mathcal{I}}[x \leftarrow a]}(A'(x)) = T$ for all a, so $v_{\sigma_{\mathcal{I}}}(\forall x A'(x)) = T$. The case for the existential quantifier is similar.

9. Here are some of the changes that need to be made to `utility.pl`.

```
instance(all(v(N1), A), all(v(N2), A1), Y, C) :-
  var(C), N1 N2, !, instance(A, A1, Y, C).
instance(all(v(N), A), all(v(N), A1), Y, C) :-
  atom(C), !, instance(A, A1, Y, C).
instance(ex(v(N1), A), ex(v(N2), A1), Y, C) :-
  var(C), N1 N2, !, instance(A, A1, Y, C).
instance(ex(v(N), A), ex(v(N), A1), Y, C) :-
  atom(C), !, instance(A, A1, Y, C).

subst_constant(X, C, [X | Tail], [C | Tail1]) :- !,
  subst_constant(X, C, Tail, Tail1).
subst_constant(X, C, [v(N) | Tail], [v(N) | Tail1]) :-
  var(C), !, subst_constant(X, C, Tail, Tail1).
subst_constant(X, C, [Y | Tail], [Y | Tail1]) :-
  atom(C), !, subst_constant(X, C, Tail, Tail1).
subst_constant(_, _, [], []).
```

10. Let A_n be an arbitrary γ formula appearing on the open branch; then $A_n \in U(i)$ for all $i \geq n$. Let a_m be an arbitrary constant appearing on the open branch; a_m was introduced by a δ-rule at some node m', and $a_m \in C(i)$ for all $i \geq m'$. Let $k = \max(n, m')$, so that $A_n \in U(k)$ and $a_m \in C(k)$. By the systematic construction, eventually the γ-rule is applied to A_n so $A_n(a_m) \in U$. Since A_n and a_m were arbitrary this proves condition 4 of the Hintikka set.

    Thus the set of formulas on an open branch form a Hintikka set, and the same proof holds for Hintikka's Lemma. The number of constant symbols is countable because they were introduced at the nodes where the δ-rule was applied: a subset of the countable number of nodes on the path. Thus every satisfiable countable set of formulas has a countable model.

11.

    (a) Let $M = M_1 \wedge \cdots \wedge M_m$ be an unsatisfiable conjunction of instances of the matrix. M itself is an unsatisfiable conjunction of literals, so there must be two clashing literals $l_i \in M_i, l_j \in M_j$. If $i = j$, then M_i itself is unsatisfiable, otherwise $M_i \wedge M_j$ is unsatisfiable. The converse is trivial.

    (b) Let $M = M_1 \wedge \cdots \wedge M_m$ be an unsatisfiable conjunction of instances of the matrix, which can be written as $p_1 \wedge \cdots \wedge p_k \wedge C_1 \wedge \cdots \wedge C_l$, where the p_i are atoms and the C_i are disjunctions of at most n negative literals. $C_1 \wedge \cdots \wedge C_l$ is clearly satisfiable, so there must be some C_i which becomes false when $p_1 \wedge \cdots \wedge p_k$ are conjoined. Let $p_{i_1}, \ldots, p_{i_n}$ be atoms that clash with the n literals of C_i. The conjunction $C_i \wedge p_{i_1} \wedge \cdots \wedge p_{i_n}$ is unsatisfiable, and at worst each conjunct comes from a different instance of the matrix. The converse is trivial.

12. The proof can be found in Section 2-1.6 of Zohar Manna. *Mathematical Theory of Computation*. McGraw-Hill, 1974.

6 Predicate Calculus: Deductive Systems

1.

1.	$p(a), \neg p(a), q(a), \neg \forall x(p(x) \rightarrow q(x)), \exists xq(x)$	Axiom
2.	$\neg q(a), \neg p(a), q(a), \neg \forall x(p(x) \rightarrow q(x)), \exists xq(x)$	Axiom
3.	$\neg(p(a) \rightarrow q(a)), \neg p(a), q(a), \neg \forall x(p(x) \rightarrow q(x)), \exists xq(x)$	$\beta\rightarrow$, 1, 2
4.	$\neg p(a), q(a), \neg \forall x(p(x) \rightarrow q(x)), \exists xq(x)$	γ
5.	$\neg p(a), \neg \forall x(p(x) \rightarrow q(x)), \exists xq(x)$	γ
6.	$\neg \exists p(x), \neg \forall x(p(x) \rightarrow q(x)), \exists xq(x)$	δ
7.	$\neg \forall x(p(x) \rightarrow q(x)), \exists p(x) \rightarrow \exists xq(x)$	$\alpha\rightarrow$
8.	$\forall x(p(x) \rightarrow q(x)) \rightarrow (\exists p(x) \rightarrow \exists xq(x))$	$\alpha\rightarrow$

1.	$\neg \forall xp(x), \neg p(a), \exists xq(x), q(a), p(a)$	Axiom
2.	$\neg \forall xp(x), \neg p(a), \exists xq(x), q(a), \neg q(a)$	Axiom
3.	$\neg \forall xp(x), \neg p(a), \exists xq(x), q(a), \neg(p(a) \rightarrow q(a))$	$\beta\rightarrow$, 1, 2
4.	$\neg \forall xp(x), \exists xq(x), q(a), \neg(p(a) \rightarrow q(a))$	γ
5.	$\neg \forall xp(x), \exists xq(x), \neg(p(a) \rightarrow q(a))$	γ
6.	$\neg \forall xp(x), \exists xq(x), \neg \exists(p(x) \rightarrow q(x))$	δ
7.	$\forall xp(x) \rightarrow \exists xq(x), \neg \exists(p(x) \rightarrow q(x))$	$\alpha\rightarrow$
8.	$\exists(p(x) \rightarrow q(x)) \rightarrow (\forall xp(x) \rightarrow \exists xq(x))$	$\alpha\rightarrow$

1.	$p(a), \neg p(a), q(a), \exists x(p(x) \rightarrow q(x))$	Axiom
2.	$p(a), p(a) \rightarrow q(a), \exists x(p(x) \rightarrow q(x))$	$\alpha\rightarrow$
3.	$p(a), \exists x(p(x) \rightarrow q(x))$	γ
4.	$\forall xp(x), \exists x(p(x) \rightarrow q(x))$	δ
5.	$\neg q(a), \neg p(a), q(a), \exists x(p(x) \rightarrow q(x))$	Axiom
6.	$\neg q(a), p(a) \rightarrow q(a), \exists x(p(x) \rightarrow q(x))$	$\alpha\rightarrow$
7.	$\neg q(a), \exists x(p(x) \rightarrow q(x))$	γ
8.	$\neg \exists q(x), \exists x(p(x) \rightarrow q(x))$	δ
9.	$\neg(\forall xp(x) \rightarrow \exists q(x)), \exists x(p(x) \rightarrow q(x))$	$\beta\rightarrow$, 4, 8
10.	$(\forall xp(x) \rightarrow \exists q(x)) \rightarrow \exists x(p(x) \rightarrow q(x))$	$\alpha\rightarrow$

2. Soundness: The axioms are clearly valid and we have proven that the α- and β-rules preserve validity. Let $\mathcal{I}$ be an arbitrary interpretation; then $\mathcal{I} \models U$ where $\{\exists xA(x), A(a)\} \in U$. By Theorem 2.32, $\mathcal{I} \models U - \{A(a)\}$, so the γ-rule is valid. Let $\mathcal{I}$ be an arbitrary interpretation; then $\mathcal{I} \models U$, where $\{A(a)\} \in U$, and a does not occur in any other formula in U. By definition, $v_{\mathcal{I}}(A(a)) = T$ where a is assigned $d \in D$ in $\mathcal{I}$. However, U is valid and a does not occur otherwise in U, so $v_{\mathcal{I}_{d'}}(A(a)) = T$ where $\mathcal{I}_{d'}$ is any interpretation that is the same as $\mathcal{I}$ except that $d' \in D$ is assigned to a. Clearly, $v_{\sigma_{\mathcal{I}}[x \leftarrow d']}(A(x)) = T$ for all $d' \in D$ and

therefore $v_{\mathcal{I}}(\forall x A(x)) = T$. Completeness is immediate from the completeness of semantic tableaux and the relation of tableaux to $\mathcal{G}$.

3. Let $\mathcal{I}$ be an interpretation for Axiom 4, and suppose that $v(\forall x A(x)) = T$, but $v(A(a)) = F$. But $v(\forall x A(x)) = T$ iff $v_{[x\leftarrow d]}(A(x)) = T$ for all $d \in D$, in particular for the d' that is assigned to a by $\mathcal{I}$. Since x is not free in $A(a)$, $v(A(a)) = v_{[x\leftarrow d]}(A(a)) = T$ contradicting the assumption.

   Let $\mathcal{I}$ be an interpretation for Axiom 5 and suppose that $v(\forall x(A \to B(x))) = T$, $v(A) = T$ but $v(\forall x B(x)) = F$. $v_{[x\leftarrow d]}(A \to B(x)) = T$ for all d and $v_{[x\leftarrow d]}(A) = v(A) = T$ since x is not free in A. By MP, $v_{[x\leftarrow d]}(B(x)) = T$ for all d and $v(\forall B(x)) = T$ by definition.

4.

1.	$\neg \forall x A(x), A(a), \neg A(a)$	Axiom
2.	$\neg \forall x A(x), A(a)$	γ
3.	$\forall x A(x) \to A(a)$	$\alpha\to$

1.	$\neg \forall x(A \to B(x)), A, \neg A, B(a)$	Axiom
2.	$\neg \forall x(A \to B(x)), \neg B(a), \neg A, B(a)$	Axiom
3.	$\neg \forall x(A \to B(x)), \neg(A \to B(a)), \neg A, B(a)$	$\beta\to$, 1, 2
4.	$\neg \forall x(A \to B(x)), \neg A, B(a)$	γ
5.	$\neg \forall x(A \to B(x)), \neg A, \forall x B(x)$	δ
6.	$\neg \forall x(A \to B(x)), A \to \forall x B(x)$	$\alpha\to$
7.	$\forall x(A \to B(x)) \to (A \to \forall x B(x))$	$\alpha\to$

5.

1.	$\{\forall x(p(x) \to q)\} \vdash \forall x(p(x) \to q)$	Assumption
2.	$\{\forall x(p(x) \to q)\} \vdash p(a) \to q$	Axiom 4
3.	$\{\forall x(p(x) \to q)\} \vdash \neg q \to \neg p(a)$	Contrapositive
4.	$\{\forall x(p(x) \to q)\} \vdash \forall x(\neg q \to \neg p(x))$	Generalization
5.	$\vdash \forall x(p(x) \to q) \to \forall x(\neg q \to \neg p(x))$	Deduction

1.	$\{\forall x(\neg q \to \neg p(x))\} \vdash \forall x(\neg q \to \neg p(x))$	Assumption
2.	$\{\forall x(\neg q \to \neg p(x))\} \vdash \neg q \to \neg p(a)$	Axiom 4
3.	$\{\forall x(\neg q \to \neg p(x))\} \vdash p(a) \to q$	Contrapositive
4.	$\{\forall x(\neg q \to \neg p(x))\} \vdash \forall x(p(x) \to q)$	Generalization
5.	$\forall x(\neg q \to \neg p(x)) \to \forall x(p(x) \to q)$	Deduction

6.

1. $\{\forall x(p(x) \rightarrow q(x)), \exists xp(x)\} \vdash \exists xp(x)$ Assumption
2. $\{\forall x(p(x) \rightarrow q(x)), \exists xp(x)\} \vdash p(a)$ C-Rule
3. $\{\forall x(p(x) \rightarrow q(x)), \exists xp(x)\} \vdash \forall x(p(x) \rightarrow q(x))$ Assumption
4. $\{\forall x(p(x) \rightarrow q(x)), \exists xp(x)\} \vdash p(a) \rightarrow q(a)$ Axiom 4
5. $\{\forall x(p(x) \rightarrow q(x)), \exists xp(x)\} \vdash q(a)$ MP 2, 4
6. $\{\forall x(p(x) \rightarrow q(x)), \exists xp(x)\} \vdash \exists xq(x)$ Theorem 6.11
7. $\{\forall x(p(x) \rightarrow q(x))\} \vdash \exists xp(x) \rightarrow \exists xq(x)$ Deduction
8. $\vdash \forall x(p(x) \rightarrow q(x)) \rightarrow (\exists xp(x) \rightarrow \exists xq(x))$ Deduction

1. $\{\exists x(p(x) \rightarrow q(x)), \forall xp(x)\} \vdash \exists x(p(x) \rightarrow q(x))$ Assumption
2. $\{\exists x(p(x) \rightarrow q(x)), \forall xp(x)\} \vdash p(a) \rightarrow q(a)$ C-Rule
3. $\{\exists x(p(x) \rightarrow q(x)), \forall xp(x)\} \vdash \forall xp(x)$ Assumption
4. $\{\exists x(p(x) \rightarrow q(x)), \forall xp(x)\} \vdash p(a)$ Axiom 4
5. $\{\exists x(p(x) \rightarrow q(x)), \forall xp(x)\} \vdash q(a)$ MP 2, 4
6. $\{\exists x(p(x) \rightarrow q(x)), \forall xp(x)\} \vdash \exists xq(x)$ Theorem 6.11
7. $\{\exists x(p(x) \rightarrow q(x))\} \vdash \forall xp(x) \rightarrow \exists xq(x)$ Deduction
8. $\vdash \exists x(p(x) \rightarrow q(x)) \rightarrow (\forall xp(x) \rightarrow \exists xq(x))$ Deduction

For the converse, we will prove the contrapositive.

1. $\{\forall(\neg q(x) \rightarrow \neg p(x)), \forall \neg q(x)\} \vdash \forall \neg q(x)$ Assumption
2. $\{\forall(\neg q(x) \rightarrow \neg p(x)), \forall \neg q(x)\} \vdash \neg q(a)$ Axiom 4
3. $\{\forall(\neg q(x) \rightarrow \neg p(x)), \forall \neg q(x)\} \vdash \forall(\neg q(x) \rightarrow \neg p(x))$ Assumption
4. $\{\forall(\neg q(x) \rightarrow \neg p(x)), \forall \neg q(x)\} \vdash \neg q(a) \rightarrow \neg p(a)$ Axiom 4
5. $\{\forall(\neg q(x) \rightarrow \neg p(x)), \forall \neg q(x)\} \vdash \neg p(a)$ MP 2, 4
6. $\{\forall(\neg q(x) \rightarrow \neg p(x)), \forall \neg q(x)\} \vdash \exists \neg p(x)$ Theorem 6.11
7. $\{\forall(\neg q(x) \rightarrow \neg p(x))\} \vdash \forall \neg q(x) \rightarrow \exists \neg p(x)$ Deduction
8. $\vdash \forall(\neg q(x) \rightarrow \neg p(x)) \rightarrow (\forall \neg q(x) \rightarrow \exists \neg p(x))$ Deduction

7.

1. $\{\forall x \exists yp(x, y)\} \vdash \forall x \exists yp(x, y)$ Assumption
2. $\{\forall x \exists yp(x, y)\} \vdash \exists yp(a, y)$ Axiom 4
3. $\{\forall x \exists yp(x, y)\} \vdash p(a, b)$ C-Rule
4. $\{\forall x \exists yp(x, y)\} \vdash \forall xp(a, b)$ Generalization (illegal)
5. $\{\forall x \exists yp(x, y)\} \vdash \exists y \forall xp(x, y)$ Theorem 6.11
6. $\vdash \forall x \exists yp(x, y) \rightarrow \exists y \forall xp(a, y)$ Deduction

8. $\vdash A'$ should be $\vdash \neg A'$.

Suppose that $\vdash A \vee B$, so that by soundness $\models A \vee B$ The first step in a tableau construction is to apply the α-rule to $\neg(A \vee B)$ to obtain the set $\{\neg A, \neg B\}$. The tableau will eventually close, so by the proof of the tableau construction, we know that this set is unsatisfiable, that is, for *arbitrary* v, $v(\neg A) \neq v(\neg B)$, so $v(A) \neq (B)$, $v(A \wedge B) = F$ and $v(\neg(A \wedge B)) = T$. Since v was arbitrary, $\models \neg(A \wedge B)$ and by completeness $\vdash \neg(A \wedge B)$.

Let $A = \forall x B(x)$. Then $\neg A' = \neg \exists x B'(x) \equiv \forall x \neg B'(x)$. In a tableau construction for A, eventually we reach ground instances of $B(x)$ whose tableaux close. By induction, the tableaux for the corresponding instances of $\neg B'(x)$ must also close.

9. I would be pleased if someone would contribute a program!

10. Suppose that A is satisfied in a finite model $\mathcal{I}$, that is, a model with a finite domain D. By definition D is non-empty, so there is at least one element $d \in D$. Define an infinite interpretation $\mathcal{I}'$ from $\mathcal{I}$ by adding to D an infinite sequence of new elements $e_1, e_2, \ldots$ and for every n-ary relation R_i in $\mathcal{I}$ and tuple $(x_1, \ldots, x_n)$, $(x_1, \ldots, x_n) \in R_I$ iff $(x'_1, \ldots, x'_n) \in R_I$ where $x'_j = d$ if $x_j = e_k$ for some k, otherwise, $x'_j = x_j$. It is easy to show by induction that $\mathcal{I}'$ is also a model for A. The converse is trivial.

11. Let $\{A_1, A_2, \ldots\}$ be a list of all formulas in $\mathcal{T}$. Construct $\mathcal{T}_i$ by induction as follows. $\mathcal{T}_0 = \mathcal{T}$. If $\mathcal{T}_i \not\vdash \neg A_{i+1}$ then set $\mathcal{T}_{i+1} = \mathcal{T}_i \cup \{A_{i+1}\}$ otherwise $\mathcal{T}_{i+1} = \mathcal{T}_i$. Finally, let $\mathcal{T}' = \bigcup_i \mathcal{T}_i$ Show that $\mathcal{T}'$ is both consistent and complete. Note that the proof is not constructive since in general we cannot show if a formula in provable in a theory.

7 Predicate Calculus: Resolution

1.

Original formula	$\forall x(p(x) \rightarrow \exists y q(y))$
Rename bound variables	(no change)
Eliminate Boolean operators	$\forall x(\neg p(x) \vee \exists y q(y))$
Push negation inwards	(no change)
Extract quantifiers	$\forall x \exists y(\neg p(x) \vee q(y))$
Distribute matrix	(no change)
Replace existential quantifiers	$\neg p(x) \vee q(f(x))$.

Original formula	$\forall x \forall y(\exists z p(z) \wedge \exists u(q(x, u) \rightarrow \exists v q(y, v)))$
Rename bound variables	(no change)
Eliminate Boolean operators	$\forall x \forall y(\exists z p(z) \wedge \exists u(\neg q(x, u) \vee \exists v q(y, v)))$
Push negation inwards	(no change)
Extract quantifiers	$\forall x \forall y \exists z \exists u \exists v(p(z) \wedge (\neg q(x, u) \vee q(y, v)))$
Distribute matrix	(no change)
Replace existential quantifiers	$p(f(x, y)),\ \neg q(x, g(x, y)) \vee q(y, h(x, y))$.

Original formula	$\exists x(\neg \exists y p(y) \rightarrow \exists z(q(z) \rightarrow r(x)))$
Rename bound variables	(no change)
Eliminate Boolean operators	$\exists x(\neg \neg \exists y p(y) \vee \exists z(\neg q(z) \vee r(x)))$
Push negation inwards	$\exists x(\exists y p(y) \vee \exists z(\neg q(z) \vee r(x)))$
Extract quantifiers	$\exists x \exists y \exists z(p(y) \vee \neg q(z) \vee r(x))$
Distribute matrix	(no change)
Replace existential quantifiers	$p(b) \vee \neg q(c) \vee r(a)$

2.

$$
\begin{aligned}
H_1 &= \{a, f(a), f(f(a)), \ldots\} \\
B_1 &= \{pa, pf(a), pf(f(a)), \ldots, qf(f(a)), qf(f(f(a))), \ldots\}
\end{aligned}
$$

$$
\begin{aligned}
H_2 = \;& \{a, b, \\
& f(a,a), f(a,b), f(b,a), f(b,b), \\
& g(a,a), g(a,b), g(b,a), g(b,b), \\
& h(a,a), h(a,b), h(b,a), h(b,b), \\
& f(a, f(a,a)), f(f(a,a),), f(a, g(a,a), f(g(a,a), a), \ldots, \\
& g(a, f(a,a)), g(f(a,a),), g(a, g(a,a), g(g(a,a), a), \ldots, \\
& h(a, f(a,a)), h(f(a,a),), h(a, g(a,a), h(g(a,a), a), \ldots, \\
& \ldots\} \\
B_2 = \;& \{pa, pb, qaa, qab, qba, qbb, \\
& pf(a,a), pf(a,b), pf(b,a), pf(b,b), \ldots \\
& qaf(a,a), qaf(a,b), qaf(b,a), qaf(b,b), \ldots \\
& qbf(a,a), qbf(a,b), qbf(b,a), qbf(b,b), \ldots \\
& \ldots\} \\
H_3 = \;& \{a, b, c\} \\
B_3 = \;& \{pa, pb, pc, qa, qb, qc, ra, rb, rc\}
\end{aligned}
$$

3. Suppose that $\mathcal{I} \models \forall y_1 \cdots \forall y_n p(y_1, \ldots, y_n, f(y_1, \ldots, y_n))$. We will show that $\mathcal{I} \models \forall y_1 \cdots \forall y_n \exists x p(y_1, \ldots, y_n, x)$. Let $(d_1, \ldots, d_n)$ be arbitrary elements of the domain, and let d' be the domain element $f(d_1, \ldots, d_n)$. By assumption, $(d_1, \ldots, d_n, f(d_1, \ldots, d_n)) \in R_p$ where R_p is the relation assigned to p. Since $(d_1, \ldots, d_n)$ were arbitrary, this proves that $\mathcal{I} \models \forall y_1 \cdots \forall y_n \exists x p(y_1, \ldots, y_n, x)$.

4. Suppose that $\forall x_1 \cdots \forall x_n A(x_1, \ldots, x_n)$ is satisfiable in an interpretation

$$\mathcal{I} = (D, \{R_1, \ldots, R_m\}, \{d_1, \ldots, d_k\}).$$

Define the interpretation

$$\mathcal{I}' = (D', \{R'_1, \ldots, R'_m\}, \{d'_1, \ldots, d'_k\}),$$

where $D' = \{e\}$ for some $e \in D$, $d'_i = e$ for all i and $(e, \ldots, e) \in R'_i$ iff $(e, \ldots, e) \in R_i$. We claim that $\mathcal{I}'$ is a model for the formula. We have to show that for any choice of $\{e_1, \ldots, e_n\}$, $A(e_1, \ldots, e_n)$ is true. But there is only one domain element so the only formula that has to be checked is $A(e, \ldots, e)$. By assumption, this formula contains no quantifiers or function symbols, so a simple induction on the Boolean structure of the formula will suffice. Since $\forall x_1 \cdots \forall x_n A(x_1, \ldots, x_n)$ is true in $\mathcal{I}$, it follows that $A(e, \ldots, e)$ is true. For an atomic formula, the formula is true by construction, and for Boolean operators, the result follows by propositional reasoning. The converse is trivial.

5. Soundness follows immediately from Theorem 7.29.

   Let A be an arbitrary formula. By Theorem 7.11, there is a formula A' in clausal form such that $A' \approx A$. Clearly, $\neg A' \approx \neg A$ by just complementing the relations in an interpretation. Suppose now that A' is not *valid*; then there is a model for $\neg A'$, so there is a model for $\neg A$, so A is not valid. We have shown that if A is valid, then the formula A' in clausal form is also valid. Therefore, it is sufficient to show that a valid formula in clausal form is provable.

   By the semantic form of Herbrand's Theorem (7.23), there is a finite set S of ground instances of clauses in A' which is unsatisfiable. Assign a distinct new propositional letter to each ground atom in S, obtaining a set S' of propositional clauses. It is trivial that S' is unsatisfiable. By the completeness of resolution for the propositional calculus, there is a refutation of S' and the refutation can be mapped into a ground refutation of S.

6.

$$\begin{aligned}
E\theta &= p(f(g(y)), f(u), g(u), f(y)) \\
(E\theta)\sigma &= p(f(g(f(a))), f(y), g(y), f(f(a))) \\
\theta\sigma &= \{x \leftarrow f(g(f(a))),\ z \leftarrow f(f(a)),\ u \leftarrow y\} \\
E(\theta\sigma) &= p(f(g(f(a))), f(y), g(y), f(f(a)))
\end{aligned}$$

7. We prove the theorem for an expression that is a single variable u; the theorem follows from a simple induction on the structure of the expression.

$$\begin{aligned}
u\theta &= t_i, \text{ if } u \text{ is } x_i \text{ for some } i \\
u\theta &= u, \text{ if } u \text{ is not } x_i \text{ for all } i
\end{aligned}$$

$$
\begin{aligned}
(u\theta)\sigma &= t_i\sigma, \text{ if } u \text{ is } x_i \text{ for some } i\\
(u\theta)\sigma &= s_j, \text{ if } u \text{ is not } x_i \text{ for all } i \text{ and } u \text{ is } y_j \text{ for some } j
\end{aligned}
$$

$$
\begin{aligned}
u(\theta\sigma) &= t_i\sigma, \text{ if } u \text{ is } x_i \text{ for some } i\\
u(\theta\sigma) &= s_j, \text{ if } u \text{ is } y_j \text{ for some } j \text{ and } y_j \text{ is not } x_i \text{ for all } i
\end{aligned}
$$

Clearly, the two definitions are the same.

8.

$p(a, x, f(g(y)))$ and $p(y, f(z), f(z))$ can be unified by

$$\{y \leftarrow a,\ x \leftarrow f(g(y)),\ z \leftarrow g(y)\}.$$

$p(x, g(f(a)), f(x))$ and $p(f(a), y, y)$ cannot be unified because of the function symbol clash between $g(f(a))$ and $f(x)$ after unifying each with y.

$p(x, g(f(a)), f(x))$ and $p(f(y), z, y)$ cannot be unified because of the occur check after substituting for x in the third equation.

$p(a, x, f(g(y)))$ and $p(z, h(z, u), f(u))$ can be unified by

$$\{x \leftarrow h(a, g(y)),\ z \leftarrow a,\ y \leftarrow g(y)\}.$$

9.

$$
\begin{aligned}
\theta\theta &= \{x_i \leftarrow t_i\theta \mid x_i \in X, x_i \neq t_i\theta\} \cup \{x_i \leftarrow t_i \mid x_i \in X, x_i \neq X\}\\
&= \{x_i \leftarrow t_i\theta \mid x_i \in X, x_i \neq t_i\theta\}
\end{aligned}
$$

If $\theta\theta = \theta$, then $t_i\theta = t_i$ so $V \cap X = \emptyset$, and conversely, if $V \cap X = \emptyset$ then $t_i\theta = t_i$, so $\theta\theta = \theta$.

Let θ be the mgu produced by the unification algorithm, and suppose that $x \leftarrow t_i \in \theta$ is such that $x_j \in X$ is in t_i. Then x_j appears on the left-hand side of some substitution in θ so the fourth rule would be applicable, contradicting the assumption.

10. The clausal form of the negation of $\forall x(A(x) \rightarrow B(x)) \rightarrow (\forall x A(x) \rightarrow \exists x B(x))$ is $\{\neg A(x)B(x),\ A(y),\ \neg B(z)\}$. Using the substitution $y \leftarrow x$, the first two clauses resolve to give $B(x)$ and using the substitution $z \leftarrow x$, this clause resolves with the third one to give $\Box$.

The clausal form of the negation of $\forall x(A(x) \rightarrow B(x)) \rightarrow (\exists xA(x) \rightarrow \exists xB(x))$ is $\{\neg A(x)B(y),\ A(f(x)),\ \neg B(z)\}$. Using the substitution $z \leftarrow y$, the first and third clauses resolve to give $\neg A(x)$. After *standardizing apart* the second clause to $A(f(x'))$, the substitution $x \leftarrow f(x')$ can be used to resolve it with $\neg A(x)$ to give $\square$.

11. I would be pleased if someone would contribute a program!

12. I would be pleased if someone would contribute a program!

13. (From: Harry Lewis, *Renaming a set of clauses as a Horn set*, Journal of the ACM, 25(1), 1978, 134-135.) Suppose that $R_U(S)$ is Horn. Let I be the interpretation obtained by assigning *true* to all atoms in U and *false* to all other atoms. Let $C = l^i_j \vee l^i_k \in S^*$ be *arbitrary*. If both literals are positive, one must have been renamed to make the clause Horn, so it is assigned *true*, satisfying C. If both are negative, one must *not* have been renamed, so its atom is assigned *false*, satisfying C. If one is positive and the other is negative, the positive one is renamed if the negative one is also renamed. Either the atom for negative literal is assigned *false*, or the positive literal is assigned *true*, satisfying C.

    Conversely, suppose that S^* is satisfied by an interpretation M, and let U be the set of atoms assigned true in M. Let $C \in R_U(S)$ be arbitrary and suppose that C had two positive literals p and q after renaming. If they were derived by renaming both $\neg p$ and $\neg q$, then p and q are assigned *true* in M, falsifying $\neg p \vee \neg q \in S^*$ which is a contradiction. If p were renamed and q not, p is assigned *true* and q is assigned *false*, falsifying $\neg p \vee q \in S^*$. If neither were renamed, they are both assigned *false*, falsifying $p \vee q \in S^*$.

8 Logic Programming

1. The identity substitution is a correct answer substitution iff $P \models \forall(\neg G)$ which is $p(a) \models \forall xp(x)$. But as shown on page 129–130, $(\mathcal{Z}, \{even(x)\}, \{2\})$ is a model for $p(a)$ but not for $\forall xp(x)$.

2. (Omitted)

3. See Figure 1.

4. We give an Hebrand model for each set of clauses obtained by omitting a clause.

    (a) $\{p(c, b),\ p(b, c),\ p(b, b),\ p(c, c)\}$.

    (b) $\{p(a, b),\ p(b, a),\ p(a, a),\ p(b, b)\}$.

    (c) $\{p(a, b),\ p(b, a),\ p(c, b),\ p(b, c)\}$.

    (d) $\{p(a, b),\ p(c, b)\}$.

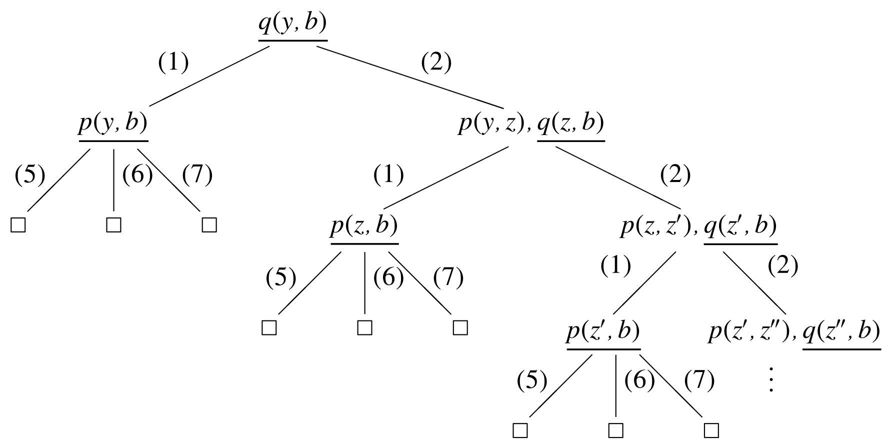


Figure 1: SLD-tree for selection of rightmost literal

For the goal clause $\leftarrow p(a,c)$, neither of the first two rules apply, so either the third or fourth rules must be applied. Whichever one comes first, in a depth-first search it will be repeatedly applied, and the other one will never be applied. Similarly, examination of the rules shows that applications of rules three or four will never produce a clause to which rules one or two can be applied, even if they come before rules three or four in the ordering. However, we have just shown that all rules are necessary to produce a refutation.

5. With the occurs check, $q(x, f(x))$ and $\leftarrow q(x,x)$ do not unify so there is no refutation. Without the check, they unify and resolve to give the empty clause. The "correct answer substitution" gives $\forall x q(x, f(x)) \models \forall x q(x,x)$ which is not true in the interpretation $\{\mathcal{Z}, \{<\}, \{increment\}, \{\}\}$.

6. The resolution procedure would be infinite as all the clauses $q(f^{n-1}(x), f^n(x))$ would be produced.

7.
```
slow_sort(L1, L2) :- permutation(L1,L2), ordered(L2).

ordered([_]).
ordered([H1,H2|Tail]) :- H1 =< H2, ordered([H2|Tail]).

permutation([],[]).
permutation(List,[X|Perm]) :-
    select(X,List,Rest),
    permutation(Rest,Perm).

select(X,[X|Rest],Rest).
select(X,[Head|List],[Head|Rest]) :- select(X,List,Rest).
```

8. As long as both lists are non-empty, the or-parallelism is allowed to select either of the first two clauses. The result would be an arbitrary interleaving of the two lists.

9. No guards are need since there is only pattern matching done on the tree input to `flatten` and the list into to `sum_list`. The output assignments, of course, must be in the bodies of the clauses.

```
flatten(null, List) :- true | List = [].
flatten(tree(Left, Val, Right), List) :- true |
  flatten(Left, List1),
  flatten(Right, List2),
  append(List1, [Val], List3),
  append(List3, List2, List4),
  List = List4.

sum_list([], Sum) :- true | Sum = 0.
sum_list([Head|Tail], Sum) :- true |
  sum_list(Tail, Sum1),
  Sum is Head + Sum1.
```

10. This program solves the problem after generating $3,496,459$ permutations.

```
send_more_money( S, E, N, D, M, O, R, Y ) :-
  permutation(
      [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ],
      [ S, E, N, D, M, O, R, Y, _, _ ] ),
  S =\= 0, M =\= 0,
  full_adder( D, E, 0, Y, C1 ),
  full_adder( N, R, C1, E, C2 ),
  full_adder( E, O, C2, N, C3 ),
  full_adder( S, M, C3, O, C4 ),
  full_adder( 0, 0, C4, M, 0 ).

full_adder( A, B, CarryIn, Sum, CarryOut ) :-
  S is A + B + CarryIn,
  ( S =< 9 -> Sum = S, CarryOut = 0 ;
      Sum is S - 10, CarryOut = 1).
```

11. In the following program, `select` is called only $6,045$ times to select a digit.

```
send_more_money( S, E, N, D, M, O, R, Y ) :-
  R0 = [ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ],
  select(D, R0, R1), select(E, R1, R2), select(Y, R2, R3),
  full_adder( D, E, 0, Y, C1 ),
  select(N, R3, R4), select(R, R4, R5),
  full_adder( N, R, C1, E, C2 ),
  select(O, R5, R6),
  full_adder( E, O, C2, N, C3 ),
  select(S, R6, R7), S =\= 0,
  select(M, R7, _), M =\= 0,
  full_adder( S, M, C3, O, C4 ),
  full_adder( 0, 0, C4, M, 0 ).
```

9 Programs: Semantics and Verification

1. $\{p\}$ `S` $\{true\}$ is true iff when S is started in a state in which p is true and S terminates then *true* is true. Of course, this holds for any precondition and statement so $wp(\mathtt{S},\ true) = true$.

2.

$$
\begin{aligned}
& wp(\mathtt{x{:}{=}x{+}y;y{:}{=}x{*}y},\ x < y) \\
&\quad = \quad wp(\mathtt{x{:}{=}x{+}y},\ wp(\mathtt{y{:}{=}x{*}y},\ x < y)) \\
&\quad = \quad wp(\mathtt{x{:}{=}x{+}y},\ x < y[y \leftarrow x * y]) \\
&\quad = \quad wp(\mathtt{x{:}{=}x{+}y},\ x < x * y) \\
&\quad = \quad x < x * y[x \leftarrow x + y] \\
&\quad = \quad x + y < (x + y) * y \\
&\quad \equiv \quad ((x > -y) \rightarrow (y > 1)) \wedge ((x < -y) \rightarrow (y < 1)).
\end{aligned}
$$

3. Let s be an arbitrary state in which $wp(\mathtt{S},\ p \wedge q)$ is true. Executing `S` leads to a state s' such that $p \wedge q$ is true, so p and q are both true in s'. Since s was arbitrary, we have proved that

$$\{s \mid\ \models wp(\mathtt{S},\ p \wedge q)\} \subseteq \{s \mid\ \models wp(\mathtt{S},\ p)\},$$

and similarly for q, so $\models wp(\mathtt{S},\ p \wedge q) \rightarrow wp(\mathtt{S},\ q) \wedge wp(\mathtt{S},\ q)$.

4.

$$
\begin{aligned}
& wp(\mathtt{if\ B\ then\ begin\ S1;S3\ end\ else\ begin\ S2;S3\ end},\ q) \\
&\quad = \quad (B \rightarrow wp(\mathtt{S1;S3},\ q)) \wedge (\neg B \rightarrow wp(\mathtt{S2;S3},\ q))
\end{aligned}
$$

$$\begin{aligned}
&= (B \rightarrow wp(\texttt{S1},\, wp(\texttt{S3},\, q))) \wedge (\neg B \rightarrow wp(\texttt{S2},\, wp(\texttt{S3},\, q))) \\
&= wp(\texttt{if B then S1 else S2},\, wp(\texttt{S3},\, q)) \\
&= wp(\texttt{if B then S1 else S2; S3},\, q).
\end{aligned}$$

5. Informally, this can be seen from the figure on page 208. If there no states leading to ∞, then the two weakest preconditions are complements of each other. Formally, let s be an arbitrary state in $wp(\texttt{S},\, p)$. Executing S results in a state s' such that p is true, so $\neg p$ is false in s'. Thus,

$$\{s \mid \models wp(\texttt{S},\, p)\} \subseteq \overline{\{s \mid \models wp(\texttt{S},\, \neg p)\}},$$

and $\models wp(\texttt{S},\, p) \rightarrow \neg\, wp(\texttt{S},\, \neg p)$, the converse of Theorem 9.27.

Using this result and Theorem 9.25 we have:

$$\begin{aligned}
& wp(\texttt{S},\, p \vee q) \\
& \quad \equiv \; wp(\texttt{S},\, \neg(\neg p \wedge \neg q)) \\
& \quad \equiv \; \neg\, wp(\texttt{S},\, \neg p \wedge \neg q) \\
& \quad \equiv \; \neg(wp(\texttt{S},\, \neg p) \wedge wp(\texttt{S},\, \neg q)) \\
& \quad \equiv \; \neg\, wp(\texttt{S},\, \neg p) \vee \neg\, wp(\texttt{S},\, \neg q) \\
& \quad \equiv \; wp(\texttt{S},\, p) \vee wp(\texttt{S},\, q).
\end{aligned}$$

6.

Soundness: Suppose that $\{p\}$ `if B then S1 else S2` $\{q\}$ is deduced from $\{p \wedge B\}$ `S1` $\{q\}$ and $\{p \wedge \neg B\}$ `S2` $\{q\}$ using the alternative rule. By the inductive hypothesis, $\models (p \wedge B) \rightarrow wp(\texttt{S1},\, q)$ and $\models (p \wedge \neg B) \rightarrow wp(\texttt{S2},\, q)$. By propositional reasoning, $\models p \rightarrow ((B \rightarrow wp(\texttt{S1},\, q)) \wedge (\neg B \rightarrow wp(\texttt{S2},\, q)))$ so by Definition 9.21, $\models p \rightarrow wp(\texttt{if B then S1 else S2},\, q)$.

Completeness: By assumption $\models p \rightarrow wp(\texttt{if B then S1 else S2},\, q)$ which is equivalent to $\models p \rightarrow ((B \rightarrow wp(\texttt{S1},\, q)) \wedge (\neg B \rightarrow wp(\texttt{S2},\, q)))$, which by propositional reasoning implies $\models (p \wedge B) \rightarrow wp(\texttt{S1},\, q)$ and $\models (p \wedge \neg B) \rightarrow wp(\texttt{S2},\, q)$. By the inductive hypothesis, $\vdash \{p \wedge B\}$ `S1` $\{q\}$ and $\vdash \{p \wedge \neg B\}$ `S2` $\{q\}$ so an application of the alternative rule yields $\vdash \{p\}$ `if B then S1 else S2` $\{q\}$.

7. The program is based on the formula $\sum_{k=0}^{n}(2k+1) = (n+1)^2$. The invariant $(0 \leq x^2 \leq a) \wedge y = (x+1)^2$ is trivially established by the precondition and initialization, and trivially establishes the postcondition. The invariance of $0 \leq x^2 \leq a$ under the loop condition $y \leq a$ is also trivial since $x'^2 = (x+1)^2 = y$. The only non-trivial part is the invariance of $y = (x+1)^2$:

$$\begin{aligned}
x' &= x + 1 \\
y' &= y + 2x' + 1
\end{aligned}$$

$$
\begin{aligned}
y' &= (x+1)^2 + 2(x+1) + 1 \\
y' &= x^2 + 4x + 4 \\
y' &= (x+2)^2 \\
y' &= (x'+1)^2.
\end{aligned}
$$

8. The program is based on the theorem:

$$
\begin{aligned}
&(x > y) \rightarrow (\gcd(x, y) = \gcd(x - y, y)) \\
&(x < y) \rightarrow (\gcd(x, y) = \gcd(x, y - x)) \\
&(x = y) \rightarrow (\gcd(x, y) = x = y)
\end{aligned}
$$

which is proved as follows. Let $g = \gcd(x, y)$. By definition, there exist m, n such that $x = gm$ and $y = gn$. Then $x - y = g(m - n)$ so g is also *a* common divisor of $x - y$ and y. If g is not a *greatest* common divisor, there is some other common divisor $g' > g$ such that $x - y = g'm'$ and $y = g'n'$. But

$$x = y + g'm' = g'n' + g'm' = g'(m' + n')$$

so g' is also a common divisor of x and y contrary to the assumption that g was the greatest common divisor. The second formula is symmetrical and the third is trivial.

By the theorem, it is trivial to show that $\gcd(x, y) = \gcd(a, b)$ is an invariant of the `while`-loop; when the loop terminates, $x = y$ and by the third formula, $x = y = \gcd(a, b)$.

9. The proof is almost the same as the proof of the previous exercise.

10. Take as the loop invariant $z \cdot x^y = a^b$. Clearly it holds initially. If y is non-zero and even, $y = 2 \cdot k$ for some k and $z \cdot x^{(2 \cdot k)} = z \cdot (x^2)^k$, so the invariant is preserved by the execution of the loop body. Otherwise, $z \cdot x^y = z \cdot (x \cdot x^{y-1}) = (z \cdot x) \cdot x^{y-1}$, and again the invariant is preserved. When the loop terminates, $y = 0$ and $z \cdot x^0 = z = a^b$.

10 Programs: Formal Specification with Z

1. This exercise refers to the material in the first edition of the book.

2. The definitions are taken from Section 5.6.2 of Potter, Sinclair & Till (1996). See also Section 4.3.3 for the notation $\{Decl \mid Pred \bullet Expr\}$ which means the set of *Expr* such that for some *Decl*, *Pred* holds.

The *head* is the first element of the sequence. If there are n elements, the *last* element is the element at position n. *front* is the domain restriction of the sequence to the first $\#s$ elements. For *tail*, the expression gives all mappings for n

from $2-1$ to $\#s-1$ into s n. Concatenation is obtained by taking the maps in s and adding the maps in t, offset by n. Note that $n \in 2..\#t$. should be $n \in 1..\#t$.

11 Temporal Logic: Formulas, Models, Tableaux

1. Let A be a valid propositional formula with atoms $\{p_1, \ldots, p_n\}$ and let A' be a PTL formula obtained from A by the substitution $[p_1 \leftarrow A_1, \ldots, p_n \leftarrow A_n]$, where each A_i is a PTL formula. Let $\mathcal{I}' = (\mathcal{S}, \rho)$ be an arbitrary PTL interpretation for A' and let s be an arbitrary state in $\mathcal{S}$. Let $\mathcal{I}$ be an interpretation for A defined as follows: $v_{\mathcal{I}}(p_i) = T$ iff $v_{\mathcal{I}',s}(A_i) = T$.

   We prove by structural induction that $v_{\mathcal{I}}(A) = v_{\mathcal{I}',s}(A')$, so that if A is valid so is A'. The base case follows by definition. Since A' is a substitution instance of A, its principal operator is Boolean and the result follows because the definition of interpretation on Boolean operators in PTR is the same as it is for the propositional calculus.

2. Let $\mathcal{I} = (\mathcal{S}, \rho)$ be an arbitrary interpretation for the formula and let $s \in S$ be an arbitrary state. Assume that $s \models \neg \Diamond \neg p$, so that $s \not\models \Diamond \neg p$ and for all states $s' \in \rho(s)$, $s' \models p$, proving $s \models \Box p$.

3. Let $\mathcal{F}_i$ be a linear frame, let $\mathcal{I}$ be an arbitrary interpretation based on $\mathcal{F}_i$, and suppose that $\mathcal{I} \not\models \bigcirc A \leftrightarrow \neg \bigcirc \neg A$. Assume that $\mathcal{I} \not\models \bigcirc A \rightarrow \neg \bigcirc \neg A$ (the proof for the converse direction is similar). Then $\mathcal{I} \models \bigcirc A$ and $\mathcal{I} \not\models \neg \bigcirc \neg A$ so that $\mathcal{I} \models \bigcirc \neg A$. But this is impossible in a linear interpretation where there is only one successor to any state.

   Conversely, suppose that $\mathcal{F}_i$ is not linear, and let $s \in \mathcal{S}$ be a state such that $s', s'' \in \rho(s)$ for distinct states s', s''. Let $\mathcal{I}$ be an interpretation based on $\mathcal{F}_i$ such that $v_{s'}(p) = T$ and $v_{s''}(p) = F$. From $\mathcal{I}, s \models \bigcirc p$ and $\mathcal{I}, s \models \bigcirc \neg p$, we have $\mathcal{I}, s \not\models \bigcirc p \rightarrow \neg \bigcirc \neg p$.

4. We must assume reflexivity.

   The first formula is called the B axiom and the second the E axiom. The B axiom characterizes *symmetrical* frames, that is, frames in which $s' \in \rho(s)$ iff $s \in \rho(s')$. If the E axiom is satisfied, the frame is both transitive and symmetrical.

   Let $\mathcal{F}_i$ be a symmetrical frame, let $\mathcal{I}$ be an arbitrary interpretation based on $\mathcal{F}_i$, and let s be an arbitrary state in $\mathcal{S}$. Suppose that $s \models A$ and let s' be an arbitrary state in $\rho(s)$. By the assumption of a symmetrical frame, $s \in \rho(s')$ so $s' \models \Diamond A$. Since s' was arbitrary, $s \models \Box \Diamond A$.

   Conversely, suppose that $\mathcal{F}_i$ is not symmetrical, and let $s \in \mathcal{S}$ be a state such that $s' \in \rho(s)$ but $s \notin \rho(s')$. Let $\mathcal{I}$ be an interpretation based on $\mathcal{F}_i$ such that

$v_s(p) = T$ and $v_t(p) = F$ for all other states t. Clearly, $s' \not\models \Diamond p$, so $s \not\models \Box\Diamond p$ even though $s \models p$.

Let $\mathcal{F}_i$ be a symmetrical and transitive frame, let $\mathcal{I}$ be an arbitrary interpretation based on $\mathcal{F}_i$, and let s be an arbitrary state in $\mathcal{S}$. Suppose that $s \models \Diamond A$, so that $s' \models A$ for some $s' \in \rho(s)$. Let s'' be an arbitrary state in $\rho(s)$. By the assumption of a symmetrical frame, $s \in \rho(s'')$ and by the assumption of a transitive frame, $s'' \in \rho(s')$ so $s'' \models \Diamond A$. Since s'' was arbitrary, $s \models \Box\Diamond A$.

Conversely, suppose that $\mathcal{F}_i$ is not symmetrical and let $s \in \mathcal{S}$ be a state such that $s' \in \rho(s)$ but $s \notin \rho(s')$. Let $\mathcal{I}$ be an interpretation based on $\mathcal{F}_i$ such that $v_{s'}(p) = T$ and $v_t(p) = F$ for all other states t. Clearly, $s' \not\models \Diamond p$, so $s \not\models \Box\Diamond p$ even though $s \models \Diamond p$. Consider now the instance of E obtained by subsituting $\Box A$: $\Diamond\Box A \rightarrow \Box\Diamond\Box A$. By reflexivity $\Box A \rightarrow \Diamond\Box A$, so $\Box A \rightarrow \Box\Diamond\Box A$. But the construction used in the proof of Theorem 11.12 shows that this can only hold in a transitive frame.

5. We have shown that in a reflexive transitive interpretation, $\Box\Box A \leftrightarrow A$. We show also that $\Box\Diamond\Box\Diamond A \leftrightarrow \Box\Diamond A$. These formulas, together with their duals, are sufficient to show that all sequences of temporal operators can be collapsed as claimed.

   Suppose that there is an interpretation and state in which $s \models \Box\Diamond\Box\Diamond p$ but $s \not\models \Box\Diamond p$, that is $s \models \Diamond\Box\neg p$. Let $s' \in \rho(s)$ be such that $s' \models \Box\neg p$. By the assumption $s \models \Box\Diamond\Box\Diamond p$, we have $s' \models \Diamond\Box\Diamond p$, so there is an $s'' \in \rho(s')$ such that $s'' \models \Box\Diamond p$, and by reflexivity, $s'' \models \Diamond p$. Then for some $s''' \in \rho(s'')$, $s''' \models p$. By transitivity $s''' \in \rho(s')$, so $s' \models \Diamond p$ which is a contradiction.

   Conversely, suppose that $s \models \Box\Diamond p$, but $s \not\models \Box\Diamond\Box\Diamond p$ that is $s \models \Diamond\Box\Diamond\Box\neg p$. As before, there is an s' such that $s' \models \Box\Diamond\Box\neg p$ and $s' \models \Diamond\Box\neg p$, then an s'' such that $s'' \models \Box\neg p$, which is $s'' \models \neg\Diamond p$, which contradicts $s \models \Box\Diamond p$.

6. A *very* long tableau is created by my program. Can anyone shorten it?

7. The elements in the labels of the nodes of a tableau are subformulas of the formula, or their negations, nexttimes or negated nexttimes. As in the propositional calculus, we can show that the α- and β-rules must eventually produce states. The number of states is limited by the number of subsets of nexttime formulas, so eventually the construction can only construct X-nodes that already exist.

8. By Lemma 11.32, a path in a Hintikka structure is a linear Hintikka structure. Suppose that A is a future formula in some node. If A is fulfilled in its own component, then by construction there will be a node further along the path in which it is fulfilled: (a) for a terminal component by the (infinite) repetition of all the nodes, or (b) for a non-terminal component, we have ensure that every node appears after every other one in the path. If A is not fulfilled in its own

component, then it is fulfilled in some other component that is accessible in the component graph. By construction, the path includes all nodes of all accessible components.

9. The notation refers to the first edition of the book. The intent here is to give a list of the formulas in each of the four nodes of the structure. Abbreviating $\Diamond(p \wedge q)$, $\Diamond(\neg p \wedge q)$ and $\Diamond(p \wedge \neg q)$ by A, B, C, respectively, all nodes include

$$\Box(A \wedge B \wedge C),\ A \wedge B \wedge C,\ \bigcirc\Box(A \wedge B \wedge C),\ A,\ B \wedge C,\ B,\ C,$$

and in addition

$$\begin{aligned} s_0 &= \{p \wedge q, p, q, \bigcirc B, \bigcirc C\} \\ s_1 &= \{\neg p \wedge q, \neg p, q, \bigcirc A, \bigcirc C\} \\ s_2 &= \{p \wedge \neg q, p, \neg q, \bigcirc A, \bigcirc B\} \\ s_3 &= \{\bigcirc A, \bigcirc B, \bigcirc C\}. \end{aligned}$$

10. The negation of $\Box\Diamond p \rightarrow \Diamond\Box p$ is equivalent to $\Box\Diamond p \wedge \Box\Diamond\neg p$, which is satisfied by the model with states $s_0 \models p$ and $s_1 \models \neg p$ and transitions $s_0 \rightsquigarrow s_1$ and $s_1 \rightsquigarrow s_0$.

11. I would be pleased if someone would contribute a program!

12 Temporal Logic: Deduction and Applications

1.

1.	$\vdash \Box(p \wedge q) \rightarrow p \wedge q$	Expansion
2.	$\vdash \Box(p \wedge q) \rightarrow p$	PC
3.	$\vdash \Box\Box(p \wedge q) \rightarrow \Box p$	Generalization
4.	$\vdash \Box(p \wedge q) \rightarrow \Box p$	Transitivity
5.	$\vdash \Box(p \wedge q) \rightarrow \Box q$	(similarly)
6.	$\vdash \Box(p \wedge q) \rightarrow \Box p \wedge \Box q$	PC

1.	$\vdash \bigcirc(\neg p \wedge \neg q) \leftrightarrow (\bigcirc\neg p \wedge \bigcirc\neg q)$	Theorem 12.3
2.	$\vdash \neg\bigcirc(\neg p \wedge \neg q) \leftrightarrow \neg(\bigcirc\neg p \wedge \bigcirc\neg q)$	PC
3.	$\vdash \neg\bigcirc(\neg p \wedge \neg q) \leftrightarrow \neg\bigcirc\neg p \vee \neg\bigcirc\neg q)$	PC
4.	$\vdash \bigcirc\neg(\neg p \wedge \neg q) \leftrightarrow (\bigcirc p \vee \bigcirc q)$	Linearity
5.	$\vdash \bigcirc(p \vee q) \leftrightarrow (\bigcirc p \vee \bigcirc q)$	PC

2. Except for (e) all are trivial consequences of linearity.

1.	$\vdash (\Box(p \wedge q) \wedge \Diamond p \wedge \neg\Diamond q) \rightarrow (\Box(p \wedge q) \wedge \Diamond p \wedge \Box\neg q)$	Duality
2.	$\vdash (\Box(p \wedge q) \wedge \Diamond p \wedge \neg\Diamond q) \rightarrow ((p \wedge q) \wedge \neg q)$	Expansion
3.	$\vdash (\Box(p \wedge q) \wedge \Diamond p \wedge \neg\Diamond q) \rightarrow \mathit{false}$	Expansion
4.	$\vdash \Box(p \wedge q) \rightarrow (\Diamond p \rightarrow \neg\Box\neg q)$	PC
5.	$\vdash \Box(p \wedge q) \rightarrow (\Diamond p \rightarrow \Diamond q)$	Duality

3.

Axiom 2: Suppose that $s \models \bigcirc(A \to B)$ and $s \models \bigcirc A$. Let $s' = \tau(s)$; by linearity, there is only one such s'. $s' \models A \to B$ and $s' \models A$ so $s' \models B$ by MP. Therefore $s \models \bigcirc B$.

Axiom 3: Suppose that $s \models \Box A$. By reflexiveness, $s \in \rho(s)$, so $s \models A$. By definition, $s' = \tau(s)$ implies $s' \in rho(s)$, so $s \models \bigcirc A$. Now let $s' = \tau(s)$ and let $s'' \in \rho(s')$ be arbitrary. By definition of ρ as τ^*, $s'' \in \rho(s)$. By assumption $s \models \Box A$, so $s'' \models A$. Then $s' \models \Box A$ since s'' was arbitrary, and $s \models \bigcirc\Box A$ follows.

Axiom 6: Suppose that $s_0 \models A \mathcal{U} B$. By definition, there exists some n such that $s_n \models B$ *and* for all $0 \leq i < n$, $s_i \models A$. If $n = 0$, then $s_0 \models B$, otherwise $s_0 \models A$. Furthermore, if $n > 0$, then $s_0 \not\models B$, so for n (which is ≥ 1), $s_n \models B$ and for all $1 \leq i < n$, $s_i \models A$. Thus $s_1 \models A \mathcal{U} B$ and $s_0 \models \bigcirc(A \mathcal{U} B)$.

4. $\vdash \Diamond\Box\Diamond p \leftrightarrow \Box\Diamond p$ follows immediately by duality.

To prove $\vdash \Diamond\Box p \to \Box\Diamond p$, let $r = \Diamond\Box p \wedge \neg\Box\Diamond p$ and prove that r is inductive ($\vdash r \to \Box r$); the inductive formula will be easy to prove if you first prove the lemmas $\vdash \Box p \to \bigcirc\Box\Diamond p$ and $\vdash \neg\Diamond p \to \bigcirc\neg\Diamond\Box p$. Continue the proof as follows:

1.	$\vdash (\Diamond\Box p \wedge \neg\Box\Diamond p) \to \Box\Diamond\Box p$	$r \to \Box r$, distribution, PC
2.	$\vdash (\Diamond\Box p \wedge \neg\Box\Diamond p) \to \Box\Diamond p$	Expansion
3.	$\vdash \Diamond\Box p \to \Box\Diamond p$	PC

5.

1.	$\vdash \Box(\Box\Diamond p \to \Diamond q) \to (\Box\Box\Diamond p \to \Box\Diamond q)$	Generalization
2.	$\vdash \Box(\Box\Diamond p \to \Diamond q) \to (\Box\Diamond p \to \Box\Diamond q)$	Transitivity
3.	$\vdash \Box(\Box\Diamond p \to \Diamond q) \to (\neg\Box\Diamond p \vee \Box\Diamond q)$	PC
4.	$\vdash \Box(\Box\Diamond p \to \Diamond q) \to (\Diamond\Box\neg p \vee \Box\Diamond q)$	Duality
5.	$\vdash \Box\Diamond q \to \Diamond q$	Expansion
6.	$\vdash \Box\Diamond q \to (\neg\Box\Diamond p \vee \Diamond q)$	Dilution
7.	$\vdash \Box\Diamond q \to (\Box\Diamond p \to \Diamond q)$	PC
8.	$\vdash \Box\Box\Diamond q \to \Box(\Box\Diamond p \to \Diamond q)$	Generalization
9.	$\vdash \Box\Diamond q \to \Box(\Box\Diamond p \to \Diamond q)$	Transitivity
10.	$\vdash \Diamond\Box\neg p \to \Box(\Box\Diamond p \to \Diamond q)$	(similarly)
11.	$\vdash \Box(\Box\Diamond p \to \Diamond q) \leftrightarrow (\Box\Diamond q \vee \Diamond\Box\neg p)$	PC

6. The converse is easy:

1.	$\vdash \Box p \to (p \vee \Box q)$	Expansion, dilution
2.	$\vdash \Box p \to (\Box p \vee q)$	Dilution
3.	$\vdash \Box p \to \Box((p \vee \Box q) \wedge (\Box p \vee q))$	PC, generalization, transitivity
4.	$\vdash \Box q \to \Box((p \vee \Box q) \wedge (\Box p \vee q))$	(similarly)
5.	$\vdash (\Box p \vee \Box q) \to \Box((p \vee \Box q) \wedge (\Box p \vee q))$	PC

For the forward direction:

1.	$\vdash s \rightarrow r \wedge \neg \Box p \wedge \neg \Box q$	Expansion
2.	$\vdash s \rightarrow p \wedge q$	PC
3.	$\vdash s \rightarrow (p \wedge q) \wedge \neg \Box p \wedge \neg \Box q$	1, 2, PC
4.	$\vdash s \rightarrow (p \wedge q) \wedge (\neg p \vee \bigcirc \neg \Box p) \wedge (\neg q \vee \bigcirc \neg \Box q)$	Expansion
5.	$\vdash s \rightarrow \bigcirc \neg \Box p \wedge \bigcirc \neg \Box q$	PC
6.	$\vdash s \rightarrow \bigcirc r \wedge \bigcirc \neg \Box p \wedge \bigcirc \neg \Box q$	Expansion
7.	$\vdash s \rightarrow \bigcirc s$	Distribution
8.	$\vdash s \rightarrow \Box s$	Induction
9.	$\vdash \Box s \rightarrow \Box (p \wedge q)$	2, Generalization
10.	$\vdash s \rightarrow \Box (p \wedge q)$	8, 9, PC
11.	$\vdash s \rightarrow \Box p \wedge \Box q$	Distribution
12.	$\vdash \Box r \rightarrow \Box p \vee \Box q$	PC

7. In Theorem 12.20, the q's should be B's.

   Let $s_0 \models A \,\mathcal{U} B$. Then for some n, $s_n \models B$, so $s_0 \models \Diamond B$. For $0 \leq i < n$, $s_i \models A$ so $s_0 \models A \,\mathcal{W} B$. Conversely, $s_0 \models \Diamond B$ so for some n, $s_n \models B$ and the other condition follows immediately. The proof of the second formula is analogous.

   For the third formula, both $\Diamond A$ and $true \,\mathcal{U} A$ assert that for some n, $s_n \models A$, while *true* is trivially true in all states.

   Suppose that $A \,\mathcal{W}\, false$ is true. Since *false* is always false, for all i, $s_i \models A$, so $s_0 \models \Box A$. The converse is similar.

8. If $\exists \Diamond p$ then there exists a path π and state $s \in \pi$ such that $s \models p$. Suppose that $s \models \forall \Box \neg p$. The for all paths, in particular for π and for all states in π, in particular for s, $s \models \neg p$ which is a contradiction.

   If $\forall \Diamond p$ then for all paths π_i there is an $s_{ij} \in \pi_i$ such that $s_{ij} \models p$. Suppose that $s \models \exists \Box \neg p$. Then for some path π', $\Box \neg p$. But π' is π_i for some i, so $s_{ij} \models \neg p$ which is a contradiction.

9. If $s_i \models \diamondminus \Diamond p$ then for some $s_j, j \leq i$, $s_j \models \Diamond p$, so for some $s_k, j \geq k$, $s_k \models p$. Suppose that $k \geq i$. Since $s_k \models p$ reflexively implies $s_k \models \diamondminus p$, we have $s_i \models \Diamond \diamondminus p$. Otherwise, $k < i$, so $s_i \models \diamondminus p$ and reflexively $s_i \models \Diamond \diamondminus p$. Conversely, suppose that $s_i \models \Diamond \diamondminus p$. Then for $j \geq i$, $s_j \models \diamondminus p$, so for some $k \geq 0$, $s_k \models p$. Clearly, $s_0 \models \Diamond p$, so $s_i \models \diamondminus \Diamond p$.

   The forward direction of the second formula is not true in general, only at the origin. To prove the converse, suppose that $s_i \models \neg(\Box p \vee \Box q)$, which is $s_i \models \Diamond \neg p \wedge \Diamond \neg q$. Let $i' \geq i, i'' \geq i$ be such that $s_{i'} \models \neg p$ and $s_{i''} \models \neg q$. Clearly, for all $j \geq \max(i', i'')$, $s_j \not\models \boxminus p$ and $s_j \not\models \boxminus q$, so $s_j \not\models \Box(\boxminus p \vee \boxminus q)$ and hence $s_i \not\models \Box(\boxminus p \vee \boxminus q)$.

10. (a) is trivial since *Last* is initially assigned 1 and subsequent statements can only assign 1 or 2. (b) should be $\vdash \Box(C1 \leftrightarrow (Test1 \vee CS1 \vee Reset1))$ and

similarly for (c). Consider first $C1 \rightarrow (Test1 \vee CS1 \vee Reset1)$. The implication is trivially true initially because *C1* is assigned *false*. Now the only way that an implication can be falsified is if (i) both the antecedent and the consequent are true and the consequent become false while the antecedent remains true, or (ii) both are false and the antecedent becomes true which the consequent remains false. (i) The consequent becomes false only on the transition from *Reset1* to *NC1*, but the assignent assigns *false* to *C1*. (ii) The antecedent becomes true only on the transition from *Set1* to *Test1*, but the consequent also becomes true. The converse is similar and the same proof holds for (c) with 1 replaced by 2 in the names.

11. We prove the first conjunct as the second is symmetric. (i) Suppose that both the antecedent and the consequent are true and that the consequent becomes false. Falsifying *C2* when `P2` is at *CS2* is impossible, as is falsifying $Last = 1$ when `P1` is at *Test1*. (ii) Suppose that both the antecedent and the consequent are false and that the antecedent becomes true. By the semantics of conjunction and the interleaving of transitions, the antecedent can become true only if `P1` moves from *Set1* to *Test1 while* `P2` is at *CS2*, or if `P2` moves from *Test2* to *CS2 while* `P1` is at *Test1*. For the first transition, *C2* is and remains true (Lemma 12.16(c)), while the transition makes $Last = 1$ true; for the second transition, since *C2* is and remains true, the only way that the consequent could be false is if $\neg(Last = 1)$, which by Lemma 12.26(a) means that $Last = 2$ is true. Furthermore, *Test1* implies that *C1* is true. But if $C1 \wedge Last = 2$ is true then the transition to *CS2* is not taken.

12. The algorithm and its proof are given in Section 3.6 of Ben-Ari(1990).

13. Add $P \wedge C$ as a conjunct in the antecedent of the existing formulas formulas, and add the formulas $P \vee C$, $(\neg P \wedge C) \rightarrow \bigcirc(Q = 1)$ and $(P \wedge \neg C) \rightarrow \bigcirc(Q = 0)$.

14. Note that the state 12 is inconsistent and should be deleted both from the text and Figure 12.6. In the following we denote $\bigcirc\Diamond(S1 \wedge \Box\neg R1)$ by N (for *next*). From state 13 the only consistent state is $16.((\mathtt{T1}, \mathtt{T2}, 2), N)$. Executing `P2` leads back to itself, and execution `P1` leads to $17.((\mathtt{R1}, \mathtt{T2}, 2), N)$. From state 17, the execution goes to state 14 or back to itself. From state 14, the `P2` transition is consistent and leads to state 7. From state 15, there are three consistent transitions: to 3, to itself and to $18.((\mathtt{T1}, \mathtt{T2}, 1), N)$. From state 18, executing `P1` leads back to itself, and executing `P2` leads to $19.((\mathtt{T1}, \mathtt{R2}, 1), N)$ which leads to itself and to state 13.