CSC236, Summer 2005, Assignment 4 Draft Solutions

Joe Whitney

1. Let x, y, and z be propositional variables and let n be a natural number. Using only the connectives \lor, \land , and \rightarrow and the given variables, how many different propositional formulae are there with exactly n connectives?

For n = 0 there are 3 formulae, namely x, y, and z.

For n = 1 there are 27 formulae.

For n = 2 there are 486 formulae...

Derive a general formula for the number of such propositional formulae having n connectives. Prove that your formula is correct.

SOLUTION: I claim that the number of formulae with n connectives is

$$f(n) = \begin{cases} 3 & \text{if } n = 0\\ \sum_{i=0}^{n-1} 3f(i)f(n-i-1) & \text{if } n > 0 \end{cases}$$

for all natural numbers n.

PROOF (By complete induction). When n = 0, there are only 3 formulae with 0 connectives (namely "x", "y", and "z"). So the base case holds.

For the inductive step, let n > 0 be an arbitrary natural number and assume (IH) that for all natural numbers k such that k < n, the number of formulae with k connectives is given by f(k). Now consider an arbitrary propositional formula P_n (of the prescribed kind) with exactly n connectives. Since n > 0, P_n has some connectives. There is then a unique innermost connective (by the inductive construction of P_n), which I'll generically call \odot , so that P_n can be rewritten $(P_l \odot P_r)$ where P_l and P_r are arbitrary formulae having a total of n-1 connectives between them. Thus, the number of ways of forming such a P_n from allowed choices of P_l, P_r , and \odot will give the number of formulae with n connectives.

Let *i* be a natural number between 0 and n-1. Let S_i be the set of formulae *P* having exactly *n* connectives such that, when *P* is decomposed into P_i , P_r , and \odot , P_l has exactly *i* connectives. It follows that P_r must have exactly n-i-1 connectives. Each distinct triple P_l, P_r , and \odot gives a distinct propositional formula (and every formula *P* in S_i corresponds to such a choice). There are always exactly three choices of \odot from the fixed set $\{\wedge, \lor, \rightarrow\}$. And, by the IH, there are f(i) choices of P_l and f(n-i-1) choices of P_r . Therefore the number of formulae in S_i is given by 3f(i)f(n-i-1).

Each choice of *i* induces a *distinct* S_i , since only formulae in S_i can be uniquely decomposed so that P_i has *i* connectives. Further, every formula with *n* connectives is in some S_i . Therefore the sets $S_0, S_1, \ldots, S_{n-1}$ partition the formulae with *n* connectives, and the total number of such formulae is given by the sum

$$\sum_{i=0}^{n-1} |S_i|$$

which as argued above is

$$\sum_{i=0}^{n-1} 3f(i)f(n-i-1)$$

which is f(n).

I conclude that the number of propositional formulae (of the prescribed kind) with n connectives is given by f(n), for all natural numbers n.

- 2. In the propositional formulae below I use the rules of precedence from chapter 5 of the Course Notes to reduce the number of parentheses. In your solution you are welcome to re-introduce parentheses if it makes things clearer.
 - (a) Use the logical equivalences in section 5.6 (no truth tables) to prove that $(P \to (Q \leftrightarrow R))$ LEQV $\neg (P \land ((Q \to R) \lor (R \to Q)) \land \neg (\neg Q \land \neg R) \land \neg (Q \land R))$

SOLUTION: Starting with the right hand side:		
$ eg (P \land ((Q ightarrow R) \lor (R ightarrow Q)) \land eg (\neg Q \land \neg R) \land eg (Q \land R))$	LEQV	$(ightarrow ext{law twice})$
$ eg (P \land ((\neg Q \lor R) \lor (\neg R \lor Q)) \land \neg (\neg Q \land \neg R) \land \neg (Q \land R))$	LEQV	(commutative and associative laws)
$ eg (P \land ((\neg Q \lor Q) \lor (\neg R \lor R)) \land \neg (\neg Q \land \neg R) \land \neg (Q \land R))$	LEQV	(distributive laws)
$ egin{aligned} egi$	LEQV	(identity law twice, idempotence)
$ eg (P \land \neg (\neg Q \land \neg R) \land \neg (Q \land R))$	LEQV	(De Morgan's, double negation)
$ eg (P \land ((\neg Q \land \neg R) \lor (Q \land R)))$	LEQV	$(\leftrightarrow \texttt{law})$
$ eg (P \land (Q \leftrightarrow R))$	LEQV	(De Morgan's, double negation)
$ eg P \lor (Q \leftrightarrow R)$	LEQV	ightarrow law
$P ightarrow (Q \leftrightarrow R)$		

(b) Write a CNF formula equivalent to both formulae in part (a). Prove that your CNF formula is correct using the logical equivalences from Section 5.6. SOLUTION: An equivalent CNF formula is $(\neg P \lor \neg Q \lor R) \land (\neg P \lor Q \lor \neg R)$, as shown below: $(\neg P \lor \neg Q \lor R) \land (\neg P \lor Q \lor \neg R)$ $\neg P \lor ((\neg Q \lor R) \land (Q \lor \neg R))$ $\neg P \lor ((\neg Q \land Q) \lor (\neg Q \neg R) \lor (R \land Q) \lor (R \lor \neg R))$ $\neg P \lor ((\neg Q \neg R) \lor (R \land Q)) \lor (R \lor Q))$ $\neg P \lor ((\neg Q \neg R) \lor (R \land Q))$ $P \lor ((\neg Q \neg R) \lor (R \land Q))$ $P \lor ((\neg Q \neg R) \lor (R \land Q))$ $P \lor ((\neg Q \neg R) \lor (R \land Q))$ $P \lor ((\neg Q \lor R) \lor (R \land Q))$ $P \lor ((\neg Q \lor R) \lor (R \land Q))$ $P \lor ((\neg Q \lor R) \lor (R \land Q))$

- (c) The Sheffer's stroke (or nand) operator, |, is a binary connective defined on page 138 of the Course Notes. Is | associative? Prove your claim.
 SOLUTION: The | operator is not associative. Consider the truth assignment where x ← 1, y ← 0 and z ← 0. Then ((x | y) | z) evaluates to 1 but ((x | (y | z)) evaluates to 0.
- (d) Let $P_1, P_2, \ldots P_n$ and Q be arbitrary propositional formulae. Prove that for any $n \geq 2$,

$$P_1 \vee P_2 \vee \ldots \vee P_n \to Q$$

is logically equivalent to

$$\neg Q \to \neg P_1 \land \neg P_2 \land \ldots \land \neg P_n$$

by induction on n.

SOLUTION:

For the base case where n = 2, $P_1 \vee P_2 \rightarrow Q$ can be shown equivalent to $\neg Q \rightarrow (\neg P_1 \land \neg P_2)$ directly using logical equivalences from section 5.6 of the course notes.

 $\begin{array}{ll} P_1 \lor P_2 \to Q & \text{LEQV} & (\to \text{law}) \\ \neg (P_1 \lor P_2) \lor Q & \text{LEQV} & (\to \text{law}, \text{ double negation, associativity}) \\ \neg Q \to \neg (P_1 \lor P_2) & \text{LEQV} & (\text{De Morgan's, double negation}) \\ \neg Q \to (\neg P_1 \land \neg P_2) & \text{LeqV} & (\neg P_1 \land \neg P_2) \end{array}$

For the inductive step we assume that for any arbitrary $n \ge 2$, $P_1 \lor P_2 \lor \ldots \lor P_n \to Q$ LEQV $\neg Q \to \neg P_1 \land \neg P_2 \land \ldots \neg P_n$ holds for any $P_1 \ldots P_n$.

Consider a formula $F = P_1 \vee P_2 \vee \ldots \vee P_n \vee P_{n+1} \to Q$ where $P_1 \ldots P_{n+1}$ are arbitrary. We want to argue that F is equivalent to $F' = \neg Q \to \neg P_1 \land \neg P_2 \ldots \land \neg P_{n+1}$. For convenience I'll rewrite F as $(\bigvee_{i=1}^n P_i) \vee P_{n+1} \to Q$ and F' as $\neg Q \to (\bigwedge_{i=1}^n P_i) \land \neg P_{n+1}$. By the induction hypothesis $(\bigvee_{i=1}^n P_i) \to Q$ is equivalent to $\neg Q \to (\bigwedge_{i=1}^n P_i)$. Now I'll argue by a sequence of logical equivalences:

 $\begin{array}{ll} (\bigvee_{i=1}^{n}P_{i})\vee P_{n+1}\rightarrow Q & \text{LEQV} \\ ((\bigvee_{i=1}^{n}P_{i})\rightarrow Q)\wedge (P_{n+1}\rightarrow Q) & \text{LEQV} & \text{apply the IH} \\ (\neg Q\rightarrow (\bigwedge_{i=1}^{n}P_{i}))\wedge (P_{n+1}\rightarrow Q) & \text{LEQV} & \text{contrapositive} \\ (\neg Q\rightarrow (\bigwedge_{i=1}^{n}P_{i}))\wedge (\neg Q\rightarrow \neg P_{n+1}) & \text{LEQV} \\ (\neg Q\rightarrow (\bigwedge_{i=1}^{n}P_{i})\wedge \neg P_{n+1}) & \text{LEQV} \end{array}$ The last formula above is F', so I have shown that F=F' as desired.

I conclude that for all natural $n \geq 2, P_1 \lor P_2 \lor \ldots \lor P_n \to Q$ LEQV $\neg Q \to \neg P_1 \land \neg P_2 \land \ldots \neg P_n$.

- 3. The Course Notes mention that the | (nand) and ↓ (nor) operators form complete sets of connectives by themselves. We've also seen the binary operators →, ∨, ∧, ↔ and ⊕ (exclusive or), each of which cannot form a complete set of connectives by itself. The 7 boolean functions represented by these connectives are frequently used in mathematics, logic, and computer science. There are other boolean operators, but they don't seem to get as much attention.
 - (a) Two binary boolean operators (or functions) \odot and \diamond are *distinct* if there exist propositional formulae P and Q such that $P \odot Q$ is *not* logically equivalent to $P \diamond Q$. The 7 operators listed above are all distinct. How many distinct boolean operators are there in total? Explain your answer.

SOLUTION

If two binary operators are distinct then they must have different truth tables. Conversely, operators \odot and \oplus with different truth tables must be distinct: since there is a row on the truth table for \odot that differs from the corresponding row on the truth table for \oplus then we can always find formulae P and Q such that $P \odot Q$ has a different truth value than $P \oplus Q$. (Remember that we have tautologies and contradictions like $(x \lor \neg x)$ and $(x \land \neg x)$ available to use as P and Q). A truth table for a binary operator has 4 rows, each of which can contain a 0 or a 1 as the "return value" for the operator¹. Therefore there are $2^4 = 16$ different such truth tables, so there are 16 distinct binary boolean operators.

(b) A binary boolean operator ⊙ is trivial if for all propositional formulae P, Q, R, and S, P ⊙ Q LEQV R ⊙ S. How many distinct trivial boolean operators are there? Explain your answer. SOLUTION

If $P \odot Q$ LEQV $R \odot S$ for any choice of P, Q, R, and S, then the truth value of $P \odot Q$ must always be the same as the truth value for $Q \odot R$. Since P, Q, R, and S could have any two pairs of truth values, it must be that $P \odot Q$ takes on the same truth value regardless of its arguments. There are only two truth functions (truth tables) satisfying this requirement: the truth function that maps every pair of arguments to 0 (false) and the function that maps every pair of arguments to 1 (true).

(c) A binary boolean operator ⊙ is one-sided if for all propositional formulae P, Q and S, either P⊙QLEQVP⊙S or P⊙QLEQVS⊙Q. How many distinct one-sided boolean operators are there? Explain your answer.

 $^{^{1}}$ I assume of course that the rows are identified by the truth values of the two operands; changing the order of the rows does not change the table.

SOLUTION

A one-sided operator essentially ignores one of its arguments, since leaving one of the arguments unchanged guarantees an unchanged truth value. We can view a one-sided operator which ignores its lefthand argument as a function of its righthand operator only. Such an operator can be represented by a truth table having only one "input" variable, which has only two rows. There are $2^2 = 4$ of these. Similarly there are 4 one-sided operators which ignore their righthand argument, but two of these (the trivial "always true" and "always false" operators) also ignore their lefthand argument. Thus there are 6 distinct one-sided binary boolean operators.

(d) How many non-trivial, non-one-sided operators are missing from the above list of seven? Could we use the missing operators as a complete set of connectives? Prove your claims. SOLUTION

We have been given seven operators, and we've found six one-sided operators, two of which are trivial. That accounts for 13 out of 16 possible operators. The remaining 3 are all non-trivial and non-one-sided, which can be verified by inspecting their truth tables. (Specifically, since each table has a single row with a different truth value from all the others, none of them can be one-sided since changing the value of either argument from the values on that row will change the result). I've chosen symbols which seem a reasonable reflection of the "semantics" of these operators: $\leftarrow, \neq, \text{and } \not>$.

The truth table for \leftarrow :

It is easily verified by comparing truth tables that $P \to Q$ is logically equivalent to $Q \leftarrow P$, and that $\neg P$ is equivalent to $(P \not\rightarrow P) \leftarrow P$. Therefore any formula using only connectives from $\{\rightarrow, \neg\}$ can be rewritten using connectives from $\{\leftarrow, \not\leftarrow, \not\rightarrow\}$. Since $\{\rightarrow, \neg\}$ is known to be a complete set of connectives, $\{\leftarrow, \not\leftarrow, \not\rightarrow\}$ is also a complete set.

- 4. Let LG be a first-order language having an infinite set of variables including a, b, and c, predicate P of arity 3, and predicate ≈ (the equality predicate). Consider an interpretation where the domain D is the set of all vertices of some (simple) graph G, and P(a, b, c) is true if and only if b lies on a shortest simple path from a to c in G².
 - (a) Give a formula expressing the claim: "there are exactly three vertices in G". Explain what your formula says in precise English.

²Recall that a path from a to b is a sequence of vertices $v_1, v_2, \ldots v_k$ such that each two adjacent vertices are connected by an edge, $a = v_1$ and $b = v_k$. A simple path is a path such that all k vertices in the path are distinct (not equal). A shortest simple path from a to b is a simple path such that no other path from a to b has fewer vertices (smaller k). Note that a path from a to b includes both a and b, so P(a, a, b) and P(a, b, b) are always true if there is a path from a to b. Also note that every path runs in both directions (on a simple graph) so $P(a, b, c) \rightarrow P(c, b, a)$.

SOLUTION:

A formula expressing this claim is:

 $\exists x \exists y \exists z (\neg \approx (x,y) \land \neg \approx (y,z) \land \neg \approx (z,x) \land \forall w (\approx (w,x) \lor \approx (w,y) \lor \approx (w,z)))$

An English reading of the formula says:

"There exists vertices x, y, and z such that the following is true: x is not equal to y, y is not equal to z and z is not equal to x, and for all vertices w either w is x, or w is y, or w is z."

(b) Give a formula expressing the claim "there exists an edge from vertex a to vertex b in G". Explain what your formula says in precise English.

SOLUTION: A formula expressing the claim is:

$$orall x(P(a,x,b)
ightarrow (pprox (x,a) ee pprox (x,b))) \wedge \exists x P(a,x,b)$$

. An English reading of the formula says: "For any vertex x, if x is on shortest path from a to b then either x is a or x is b. Furthermore, there exists an x such that x is on a shortest path from a to b."

(c) Give a formula expressing the claim "G is a tree"³. Explain what your formula says in precise English.

SOLUTION:

This one is a bit more complicated so I'll break it into parts (conjuncts). The first part expresses the fact that G is connected (i.e. there is a shortest path between every pair of distinct vertices):

$$orall x orall y \exists z (\neg pprox (x,y)
ightarrow P(x,z,y))$$

A literal English reading of this part says: "For any vertices x and y, if x and y are distinct then there exists a vertex z such that z is on shortest a path from x to y".

The second part expresses the fact that there is at most one shortest path between any pair of distinct vertices. I've already shown how to express the fact that there is an edge between two vertices (a and b in part (b)). Let E(x,y) be shorthand for the formula expressing the fact that there is an edge between x and y. Then I can express the uniqueness of shortest paths like this:

 $\forall x \forall y \forall z \forall z' ((P(x, z, y) \land P(x, z', y) \land E(x, z') \land E(x, z')) \rightarrow \approx (z, z'))$

Translation: "For all vertices a, b, z, and z', if both z and z' are

- connected to x by an edge; and
- on a shortest path from x to y

then z and z' are identical.

Having expressed that all shortest paths must be unique, I've ruled out the possibility of a shortest cycle in G with even degree. I must also rule out the possibility of a shortest cycle with odd degree. This is equivalent to saying that for any three distinct vertices x, y, and z, at least one of them must lie on a shortest path between the other two⁴. We already saw in part (a) how to express

³A tree is a graph which is connected (every vertex has at least one path to every other vertex) and acyclic (every vertex has no more than one path to any other vertex).

⁴This is violated by an odd shortest cycle because such a cycle will have an edge (a, b) and vertex c such that c is equidistant from a and b, and none of a, b, or c is then on a shortest path between the other two. Since the cycle is shortest we know that all the shortest paths between its vertices are on the cycle (this was given as a hint).

the distinctness of three vertices, so let D(x,y,z) be shorthand for the formula expressing that x, y and z are distinct. Then our final conjunct is simply:

$$\forall x \forall y \forall z (D(x,y,z) \rightarrow (P(x,y,z) \lor P(y,x,z) \lor P(x,z,y))$$

Translation: "If any three vertices x, y and z are distinct, then at least one is on a shortest path between the other two."

Putting the three parts together we have:

$$\begin{split} &\forall x \forall y \exists z (\neg \approx (x, y) \rightarrow P(x, z, y)) \land \\ &\forall x \forall y \forall z \forall z' ((P(x, z, y) \land P(x, z', y) \land E(x, z') \land E(x, z')) \rightarrow \approx (z, z')) \land \\ &\forall x \forall y \forall z (D(x, y, z) \rightarrow (P(x, y, z) \lor P(y, x, z) \lor P(x, z, y)) \end{split}$$