CS 512, Spring 2014

Assignment 2

Shan Sikdar

Due Monday January 27th

1 Problem 1

Algebraic Structure: $\tau = (T, Lt, \mathbb{N}, Rt, height)$. Define the size operation $||$: $\tau \rightarrow \mathbb{N}$ and the height operation height: $\tau \rightarrow \mathbb{N}$ in the style of the definitions for the operations Node Lt, and Rt. If $t \in T$ is represented graphically by a finite binary tree, then $|t|$ should return should return the total number of leaf nodes in t + the total number of internal nodes in t, and the hieght(t) should return the length of the longest path in t.

(1)

$$|t| = \begin{cases} 1 + |t_1| + |t_2|, & \text{if } t =< t_1 \;\; t_2 > . \\ 1 + |t_1|, & \text{if Rt does not exists (there is no right subtree)} \\ 1 + |t_2|, & \text{if Lt does not exists (there is no left subtree)} \\ 1, & \text{otherwise} \end{cases}$$

(2) $$height(x) = \begin{cases} max(|t_1|, |t_2|) + 1, & \text{if } t =< t_1 \;\; t_2 > . \\ 0, & \text{otherwise} \end{cases}$$

2 Problem 2

Algebraic Structure: $A = (\mathbb{N} - 0, lcm, gcd, \trianglelefteq)$ where for all $m, n \in \mathbb{N} - 0$, we have that: $m \trianglelefteq n$ iff "m divides n".
(a) Show that A is a lattice, where lcm playes the role of $\vee$ and gcd plays the role of $\wedge$.
(b) Is A a distributice lattice? Justify your answer carefully, based on the distributivity axioms that A must satisfy.

(a)
From the lecture notes, to prove lattice, Need to Show: (1)Poset, (2)Least upper bound, (3) and greatest lower bound
(1) Show $(A, \trianglelefteq)$ is a poset.

For all $a, b, c \in \mathbb{N} - 0$
reflexive: "a divides a" therefore $a \trianglelefteq a$
anit-symmetric:
Assume $a \trianglelefteq b$ and $b \trianglelefteq a$. Then "a divides b" and "b divides a".
So then this is only true if $a = b$, assuming otherwise will lead to a contradiction. So anti-symettric.
transitive:
Assume $a \trianglelefteq b$ and $b \trianglelefteq c$. Then "a divides b" and 'b divdes c".
If "a divides b" then b must be some multiple of a ($b = m \cdot a, m \in \mathbb{N} - 0$).
If "b divides c" then c must be some multiple of b ($c = n \cdot b, m \in \mathbb{N} - 0$).
So $c = n \cdot b = n \cdot (m \cdot a)$ where $m, n \in \mathbb{N} - 0$
But then c is some multiple of a. So "a divides c".
Therefore $a \trianglelefteq c$.

(2) For all $m, n \in A$ the least upper bound of a and b in the ordering $\trianglelefteq$ exists, is unique, and is the result of the operation $m \vee n$

Existance: For any two numbers m and n we know $m * n \in N$ exists so there must be some number r such that "m divdes r " and "n divides r" $\leq m * n$. Uniqueness since we know the least upperbound exists in the $\trianglelefteq$, then this least upperbound must be a natural number and therefore is unique. Lcm by definition and proof finds the lowest common multiple between any two numbers using prime factorization. so we can use lcm to find this least upper bound so it is an operation of $m \vee n$

(3) For all $m, n \in A$ the greatest lower bound of m and n in the ordering $\trianglelefteq$ exists, is unique m and is the result of the operation $m \wedge n$
Existance: Since the number 1 divides all numbers: $1 \trianglelefteq n, n \in N$, So the greatest lower bound exists since in the worst case it will be 1. So there exists r , a greatest lowerbound $1 \leq r$. Uniqueness since we know the greatest lower bound exists in the ordering, it must be a specific number, therefore it must be unique. The Greatest common demonitor function by definition and proof finds the largest natural number n that divides any to numbers so it will produce a largest number that divides m and n. So the greatest lower bound will be the result of the operation $m \wedge n$
(b)
Yes it is a distributive lattice.
proof:
The two conditions listed above can be rewritten equivelently as show lcm distributes over gcd, Show gcd distributes over lcm. So show GCD(a, LCM(b,c)) = LCM(GCD(a,b),GCD(b,c)) and LCM(a,GCD(b,c)) = GCD(LCM(a,b),LCM(a,c)).
Number thoery proof from Coppel : Using the fact that LCM(a,b) * GCD(a,b) = a,b:
Let B = LCM(a,GCD(b,c)) and A = GCD(LCM(a,b),LCM(a,c)) .
Then: $A = gcd(\frac{ab}{gcd(a,b)}, \frac{ac}{gcd(a,c)})$
B = $\frac{a*gcd(b,c)}{gcd(a,gcd(b,c))}, \frac{ac}{gcd(a,c)}) = gcd(\frac{ab}{gcd(a,(b,c)}), \frac{ac}{gcd(a,(b,c))})$

Since any common divisor of ab/gcd(a,b) and ac/gcd(a,c) is also a common divisor of ab/gcd(a,(b,c)) and ac/gcd(a,(b,c) then A divides B.
Also if a divides A and since a will divide LCM(a,b) and LCM(b,c) , then B divides A. Therefore if A divides B and B divides A, A = B.
So LCM distirbutes over GCD. The proof for GCD distributing over LCM can be obtained by switching LCM and GCD in the proof above.

3 Problem 3: Excercise 1.5.3 parts (a),(b),(c)

(a) Show that $\{\neg, \wedge\}, \{\neg, \rightarrow\}, \{\rightarrow, \perp\}$ are adequate sets of connectives for propositional logic.

(a1) $\{\neg, \wedge\}$:
If I can show that there is an expression that is equivalent to $p \vee q$, then this will also show that there exists an equivelent for $\rightarrow$ Since from the example we know $p \rightarrow q \equiv \neg p \vee q$. In order to do that I need to find an expression that has same truth values as $p \vee q$. Using a truth table we can find an equivelent expression for $p \vee q$:

p	q	$Want : p \vee q$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$	$\neg(\neg p \wedge \neg q)$
T	**T**	**T**	**F**	**F**	**F**	**T**
T	**F**	**T**	**F**	**T**	**F**	**T**
F	**T**	**T**	**T**	**F**	**F**	**T**
F	**F**	**F**	**T**	**T**	**T**	**F**

(a2) $\{\neg, \rightarrow\}$:
If I can show that there is an expression that is equivalent to $p \vee q$, then this will also show that there exists an equivelent for $\wedge$. This is beacuse from the example we know: $p \wedge q \equiv \neg(\neg p \vee \neg q)$. (So every $\wedge$ statement can be reduced into $\vee$ statements). In order to do that I need to find an expression that has same truth values as $p \vee q$. Using a truth table we can find an equivelent expression for $p \vee q$:

p	q	$Want : p \vee q$	$\neg p$	$\neg q$	$\neg q \rightarrow p$	
T	**T**	**T**	**F**	**F**	**T**	
T	**F**	**T**	**F**	**T**	**T**	
F	**T**	**T**	**T**	**F**	**T**	
F	**F**	**F**	**T**	**T**	**F**	

(a3) $\{\neg, \perp\}$:
Page 38 in the book shows that the nullary connective turns all values into

False. Looking at the results from the example and previous questions, I need to find an expression equivelent to $\neg$. (Since that was the only one we havent found an equivelent expression for yet.) If I can do that then equivelent expressions for $\vee$ and $\wedge$ can be found by using the logic used in the previous questions. Looking at the truth Tables:

p	$Want : \neg p$	$\perp$	$p \rightarrow \perp$
T	**F**	**F**	**T**
T	**F**	**F**	**T**
F	**T**	**F**	**F**
F	**T**	**F**	**F**

(b) Show that if $C \subseteq \{\neg, \wedge, \vee, \rightarrow, \perp\}$ is adequate for propositonal logic, then $\neg \in C$ or $\perp \in C$
Assume $\neg \notin C$ and $\neg \notin C$. Then look at $C \subseteq \{\wedge, \vee, \rightarrow, \}$. For any formula of any length, ϕ formed by using the connectives in C, if you have a valuation in which every atom is assigned T show that this formula always has to evaluate to True. If this is the case and there is no way to reach an expression that is false, there does not exist a expression to represent the negation of a formula when all atoms valuations are true and thus we need to have $\neg$ or $\perp$ in our set.
To do this let n = the number of connectives used in the formula ϕ, do induction on n:

Base Case: n= 1
If there is only one connective in the expression than we have three possible formulas for ϕ: (1) $p \wedge q$, (2) $p \vee q$, and (3) $p \rightarrow q$. Since we assume p, q to have valuations of True, using the truth tables from the textbook we know that all these expressions will evalute to true.

Inductive Hypothesis: Assume that a formula ϕ made up of n connectives from C always evaluates to True.

Show for n +1: Take a formula ϕ of length n+1, since we know the only connectives available are $\wedge, \vee, \rightarrow$ we know that ϕ can be made up of two other well formed formulas α and β with $length \leq n$. From the inductive hypothesis α and β both evaluate to True. So the possible combinations of ϕ are $\alpha \vee \beta$, $\alpha \wedge \beta$, $\alpha \rightarrow \beta$ where α, β evaluate to True from the induction hypothesis. Then all these possible expressions of ϕ evaluate to True, and so all ϕ of length n+1 evaluate to True.

(c) Is $\{\leftrightarrow, \neg\}$ adequate?
Taking the definition of $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$ Looking at the truth table we see:

p	q	$(p \rightarrow q)$	$(q \rightarrow p)$	$p \leftrightarrow q$	$\neg p$
T	**T**	**T**	**T**	**T**	**F**
T	**F**	**F**	**T**	**F**	**F**
F	**T**	**T**	**F**	**F**	**T**
F	**F**	**T**	**T**	**T**	**T**

Comparing to the truth tables on page 38 the truth table above has an even number of True and False valuations for anything created by $\leftrightarrow$ or $\neg$. However the truth tables fpr $\vee, \wedge, \rightarrow$ on page 38 have an odd number of True and False valuations. If I can show that any clause of any length is formed by $\{\leftrightarrow, \neg\}$, has truth tables that evaluate to an even number of True and False clauses, then there will be no way to recreate $\wedge\ \vee \rightarrow$.

Proof by induction: On the length of the formula

Base Case: $p \leftrightarrow q$ or$\neg p$
As argued above this has an even number of True and False valuations in the truth table.

Induction Hypothesis: Assume a formula of length n formed by $\{\leftrightarrow, \neg\}$ coressponds to a truth table that has an even number of True valuations and False valuations.

Show for n+1:
Take a formula ϕ of length n+1. So either $\phi = \neg\phi'$ or $\phi = \rho \leftrightarrow \lambda$ where ϕ', λ, ρ are of length n and thus by the induction hypothesis have an even number of True, False valuations in the truth table. But the situation we

have now simply is an extended version of the base case. So then and version of ϕ will have an even number of True and False Valuations in the truth tables.

4

Let $\#$ denote the ternary majority connective. Show that $\{\neg, \#\}$ is not adequate set of connectives for propositonal logic.

Looking at the truth table:

p	q	r	$\#pqr$	$\neg p$
T	**T**	**T**	**T**	**F**
T	**F**	**T**	**T**	**F**
T	**F**	**T**	**T**	**F**
T	**F**	**F**	**F**	**F**
F	**T**	**T**	**T**	**T**
F	**T**	**F**	**F**	**T**
F	**F**	**T**	**F**	**T**
F	**F**	**F**	**F**	**T**

We see that the there is an even number of True and False valuations in the truth table similar to question 1.5.3(c). However the truth tables for $\vee, \wedge, \rightarrow$ have an odd number of True and False valuations. If I can show that any clause of any length is formed by $\{\leftrightarrow, \neg\}$, has truth tables that evaluate to an even number of True and False clauses, then there will be no way to recreate $\wedge\ \vee \rightarrow$.

So once again induction to show any clause will have an even number to Truth and False valuations for its truth table. This proof is really similar to 3(c) so please refer to that logic and use $\#$ instead.

5 1.5.16

(a)Explain why the algorithm HORN fails to work correctly if we change the conept of Horn formulas by by extending the clause to $P ::= \bot|\top|p|\neg p$

For a specific case take $(p \to \neg p) \wedge (\neg p \to p)$ this is not satisfiable by design but if we put it throught the alogirthm HORN it would come out statisfiable. Another case that is unstatisfiable but is satisfiable would be: $(x_1 \wedge \neg x_2 \to x_2) \wedge (\top \to x_1) \wedge (\top \to x_2)$
In general the algorithm fails to work correctly since when it marks P_j or P' it doesn't take into account or check the contradictions that may arise by extending the definiton of P to include $\neg p$.

(b) What can you say about the CNF of Horn formulas. Mre precisely, can you specify syntatic criteria for a CNF that ensure that there is an equivalent Horn formula? Can you describe informally programs which would translate from one form representation into another?

Horn clauses are made up of conjucntion of C's : $C_0 \wedge C_1 ...... \wedge C_n$ where each C is made up of conjuntions of atoms with an implication. so each C has the form $p_1 \wedge p_2 \wedge .....p_n \to q$ from before we know that $\phi \to \rho \equiv \neg\phi \vee \rho$.
So the form $p_1 \wedge p_2 \wedge .....p_n \to q \equiv \neg(p_1 \wedge p_2 \wedge .....p_n) \vee q \equiv \neg p_1 \vee \neg p_2 \vee ...\neg p_n \vee q$
So an equivelent Horn Formula is one where the C clause is made up of negative literals and at most one positive literal joined by $\vee$. Informally programs that would convert from one representation to another would be to find a $\to$, negate all the atoms before the $\to$ and switching all the conneting conjunctions to logical or's, and then turning the $\to$ to $\vee$. A informal prgoram to convert the other way would be to search for a positive literal, change the connecting $\vee$ to $\to$ and then turing all negative literal's to its positive literal and change the $\vee$ to $\wedge$.

6 1.6.1

(a) show $T(\phi)$ can be generated by 1.10 on page 69 by clauses from pg 33. To be able to generate $T(\phi)$ need to show that any $\phi ::== p|(\neg\phi|(\phi \wedge \psi)|\phi \vee \psi|\phi \rightarrow \psi$
can be turned into $\phi ::== p|\neg\phi|(\phi \wedge \phi)$ and so $T(\phi)$ can be generated. For p and $\phi \wedge \phi$ this is trivial, for the others we will need to show that by induction that they can converted to a form using $\wedge$. This can be done based on indutions of the size formulas similar to what we have done throughout the homework. Then at the end of proofs we will have that we can write any formula from pg 33 to a formula og pg 69 and thus use the translation formula.

(b) we can use induction based on truth table valuations so if I can argue the base case, the logic can be extended to the larger cases using induction. For the negation case we can just write out the truth tables and compare and see that the valuations turn out to be the same. The expressions using the other operators can be proved by using DeMorgan's Laws or by converting them into an equivelent for and then using DeMorgan's Laws.

(c) Since we have shown that ϕ and $T(\phi)$ have the same valuations and that the phrases and be converted into equivelent forms from part (a), then the set of valuations that make ϕ true must make $T(\phi)$ true.

7

File attached.