CS 6100 – Foundations of CS – Spring 2012 – Notes for L3, 1/18/12

ALSO Assignment 2 – given at the end

1 Discussion of Assignment 1

In these notes, I am mainly demonstrating how I think. You are urged to stick with the textbook notation, at least for one of the problems in your set.

For example, if you are asked to solve

"1.1 a,e,h"

show it the painstakingly detailed way at least for 1.1 a. For the other parts, you may practice accelerating the calculations using any sound method. I am trying to demonstrate some of these methods in these notes.

It is also important to view things through Venn diagrams and also sets, whenever possible.

1. B/M, 1.1, (g)

   $(P \rightarrow Q \rightarrow R) \;\rightarrow\; \neg R \rightarrow \neg Q \rightarrow \neg P$

   As soon as I see this, I apply some identities, and rewrite

   $(P \rightarrow Q \rightarrow R) \;\rightarrow\; \neg R \rightarrow (P \rightarrow Q)$

   $(P \rightarrow Q \rightarrow R) \;\rightarrow\; (P \wedge \neg Q) \rightarrow R$

   $(P \wedge Q \rightarrow R) \;\rightarrow\; (P \wedge \neg Q) \rightarrow R$

   "Obviously false."

   $P \wedge \neg Q \wedge \neg R$ will falsify

2. B/M, 1.2, (x) "Template equivalence"

   Two formulae are equivalent ($\Leftrightarrow$) if they evaluate to the same truth value ("under all interpretations I")

   We are asked to establish this template equivalence $(F_1 \rightarrow F_2) \wedge (F_1 \rightarrow F_3) \Leftrightarrow F_1 \rightarrow (F_2 \wedge F_3)$

   Immediately see a Venn diagram where the satisfying set of F_1 is contained in that of F_2, and the satisfying set of F_1 is (also) contained in that of F_3.

   Then it must be the case that the satisfying set of F_1 is contained in that of $F_2 \wedge F_3$.

   So the proof must go through

   - We can prove the validity of $(F_1 \rightarrow F_2) \wedge (F_1 \rightarrow F_3) \leftrightarrow F_1 \rightarrow (F_2 \wedge F_3)$
   - Suppose $(F_1 \rightarrow F_2) \wedge (F_1 \rightarrow F_3)$

     F_1

     But not $(F_2 \wedge F_3)$.

     Contradiction.
   - Suppose $F_1 \rightarrow (F_2 \wedge F_3)$

     $F_1 \wedge \neg F_2$ or $F_1 \wedge \neg F_3$

     Contradiction.

3. You must be able to do 1.3, 1.4, and 1.5 – if not, please ask for help **urgently**! There is a midterm exam and this level of knowledge will be assumed.

4. Study of Tyler's SAT routine: your ability to understand Python at this level will be assumed. If not, study up!
5. Study of the DNF to CNF converter (written in Ocaml, and presented in the notes of Lecture 2).
   - I will provide an intro to Ocaml today
   - You should practice some basic Ocaml
   - Midterm-1 will contain a take-home part where simple knowledge of Python and Ocaml will be tested (Without studying declarative languages, your study of these topics will be shortchanged. In fact, advances in Logic do reflect into advances in prog. languages.)

2 Requirement to meet me during office hours next week

I require all students to meet me for 10 mins during my office hours of next week (week of 1/23/12). I have office hours on Mon 14:00 to 15:20 and Tue 15:00 to 16:30. If these slots don't work, you must ask me for other times (I can make some time next week Tuesday 1/24/12. The idea is to make office-hour visits a habit, but also to (i) discuss how well you are doing in the class, and (ii) help you select a project. Be prepared to discuss these topics. All meetings must occur next week, so we must find a time that works.

3 Lecture topics for 1/18/12

- Examining the CNF converter in Ocaml (also drives home many valuable ideas for designing these kind of routines in practice)
- Equisatisfiable explained better. See `http://www.inf.ed.ac.uk/teaching/courses/propm/papers/main_lan1.pdf`. These discussions help introduce FOL gradually.
- Read-up Section 1.7.3 from B/M on how to generate Equisatisfiable formule while preserving CNF-size
- Resolution; BCP as unit resolution – read from the B/M chapter
- Examine the SAT solver in Python – we will read through the code
- 2SAT and complexity thereof – read from Chapter 18 of my book (online now). You'll be doing a problem in Asg2.
- Assignment 2 – given today 1/18/12, discussed 1/23/12, due 1/30/12 (again, all due-dates are AOE – "anywhere on earth" – look up AOE).

4 CNF converter, and Basic Ocaml

5 Equisatisfiable

Two formulae F_1 and F_2 are equisatisfiable if

$$\exists I_1 : I_1 \models F_1 \Leftrightarrow \exists I_2 : I_2 \models F_2$$

Two formulae F_1 and F_2 are equivalent if

$$\forall I : \ I \models F_1 \Leftrightarrow I \models F_2$$

Compare these:

- $(x \rightarrow p) \wedge (\neg x \rightarrow q)$ and $p \vee q$ are equisatisfiable.
- When comparing two formulae with the same variables, these two notions coincide.

5.1 Generating Equisatisfiable CNF, Preserving Sizes

An example is provided in my book's Chapter 18. You'll also be doing a problem in Asg2.

6 Assignment 2

Points are shown

1. (10%) Prove that XOR is not a universal gate (check 1.4 of B/M which asks you a similar question about Nand)
2. (30%) (2SAT)

   Consider the 2CNF formula

   $$(a \vee b) \wedge (\neg c \rightarrow \neg b) \wedge (c \leftrightarrow d) \wedge (a \leftrightarrow \neg d)$$

   Apply the polynomial-time algorithm for 2SAT discussed in Chapter 18 of my book, and draw conclusions on the satisfiability of this formula.
3. (50%) (Conversion to Equisatisfiable CNF, preventing size explosion)

   Convert the formula below to an equisatisfiable CNF formula, and run your Python SAT routine (the one you fixed up to generate satisfying assignments). In your conversion, you must assume the availability of "implication gates." Connect-up these into a gate network that is equisatisfiable. Obtain the satisfying assignment for the resulting formula.

   $$\neg((P \rightarrow Q \rightarrow R) \ \rightarrow \ \neg R \rightarrow \neg Q \rightarrow \neg P)$$
4. (10%)

   Reading assignment:

   - Technical report on GRASP (on the class webpage). Read as much as you can.
   - All of Chapter 1 of B/M, and my Chapter 18. Read as much as you can.

   Write a 1-page summary of each of these items.