More on propositional logic

1 Distribution laws

Whereas De Morgan's Laws allow us to simplify formulae with respect to negations we often have "combinations" of disjunctions and conjunctions.

The Distributive Law of Disjunction over Conjunction is:

$$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$$

And the **Distributive law of conjunction over disjunction is**

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

Just as before there are the **generalised Distributive laws**

$$X \wedge (Y_1 \vee Y_2 \vee ... \vee Y_n) \equiv (X \wedge Y_1) \vee (X \wedge Y_2) \vee ... \vee (X \wedge Y_n)$$

$$X \vee (Y_1 \wedge Y_2 \wedge ... \wedge Y_n) \equiv (X \vee Y_1) \wedge (X \vee Y_2) \wedge ... \wedge (X \vee Y_n)$$

Of course we can apply these laws to combinations of formulae and to sub-formulae not just with propositional variables.

2 Functional Completeness

We defined propositional logic using the connectives: $\wedge \vee \neg \Rightarrow \Leftrightarrow$, but we could have chosen other connectives

We say that a set C of logical connectives is **functionally complete** if any propositional formula is equivalent to one constructed using only the connectives from C. - This is kind of like Turing complete for logic

In fact $\wedge \vee \neg$ is functionally complete

- Let φ be a propositional formula involving the variables $p_1, p_2, ..., p_n$
- Build the truth table for φ and let f be some truth assignment that evaluates to **true**
- Suppose that in this truth assignment f each p_i has the truth value of v_i
- Build a conjunction χ_f of literals as follows: for each i
  - if v_i is **true** then include the literal p_i in the conjunction χ_f
  - if v_i is **false** then include the literal $\neg p_i$ in the conjunction χ_f

2.1 Example

Example

Durham University

- Consider the following truth table for φ

p	q	r	s	φ	
T	T	T	T	F	
T	T	T	F	F	
T	T	F	T	T	$\leftarrow f_1$
T	T	F	F	F	
T	F	T	T	F	
T	F	T	F	F	
T	F	F	T	T	$\leftarrow f_2$
T	F	F	F	T	$\leftarrow f_3$

p	q	r	s	φ	
F	T	T	T	F	
F	T	T	F	F	
F	T	F	T	F	
F	T	F	F	T	$\leftarrow f_4$
F	F	T	T	F	
F	F	T	F	F	
F	F	F	T	F	
F	F	F	F	T	$\leftarrow f_5$

- So

$\chi_{f_1} = p \wedge q \wedge \neg r \wedge s$ $\chi_{f_2} = p \wedge \neg q \wedge \neg r \wedge s$ $\chi_{f_3} = p \wedge \neg q \wedge \neg r \wedge \neg s$

$\chi_{f_4} = \neg p \wedge q \wedge \neg r \wedge \neg s$ $\chi_{f_5} = \neg p \wedge \neg q \wedge \neg r \wedge \neg s$

and

$$\psi = (p \wedge q \wedge \neg r \wedge s) \vee (p \wedge \neg q \wedge \neg r \wedge s) \vee (p \wedge \neg q \wedge \neg r \wedge \neg s) \vee (\neg p \wedge q \wedge \neg r \wedge \neg s) \vee (\neg p \wedge \neg q \wedge \neg r \wedge \neg s)$$

slide 4 of 16

- f_1 can be specified by writing a truth assignment $p \wedge q \wedge \neg r \wedge s$
- The disjunction can then be written as the lor of all the inputs that make it true, so $\psi = \chi_{f_1} \vee ...$
- Only in the case of truth assignment f_1 will χ_1 be true and etc for the rest of the fs
- So this can be used to show that two truth tables are exactly the same, and so that $\wedge \vee \neg$ is **functionally complete**

2.2 More on Functional Completeness

- Now let ψ be the disjunction of all those conjunctions χ_f we have just built. Remember, we only build disjunctions corresponding to the rows of the truth table evaluating to **true**
- We claim that φ and ψ are logically equivalent
  - Suppose that f is some truth statement making φ true, so we have indeed built the conjunction χ_f
  - Key Point: The only truth assignment making the conjunction χ_f true is the truth assignment f itself
  - In particular the truth assignment f must make χ_f true
    - For example with regard to the truth assignment f in the example χ_f is $p_1 \wedge \neg p_2 \wedge ... \wedge \neg p_n$
      Which is made **true** only by the truth assignment f
  - Hence, f makes ψ true
- Conversely

  - Suppose that g is some truth assignment making ψ true
    - So at least one conjunct χ_f say, is made **true** by g
  - But the only truth assignment making χ_f true is f
    - Hence, f=g
  - The reason χ_f appears as a conjunct is because f makes φ true
    - So g=f is a truth assignment making φ true
- Consequently, for any truth assignment f
  - f satisfies φ if, and only if, f satisfies ψ
    - That is, $\varphi \equiv \psi$
- Our proof yields even more
  - Every formula of propositional logic is equivalent to a formula in **disjunctive normal form**
    - A disjunction of conjunctions of literals
  - Also, every truth table is the truth table of some propositional formula

3 Conjunctive normal form

- Let φ be some formula of propositional logic
- The formula $\neg\varphi$ is equivalent to one in disjunctive normal form
  - That is, one of the form
    $\chi_1 \vee \chi_2 \vee ... \vee \chi_m$
    Where each χ_i is a conjunction of literals
- So, φ is equivalent to the formula
  $\neg(\chi_1 \vee \chi_2 \vee ... \vee \chi_m)$
  Which in turn, by using generalised De Morgan's Laws, is equivalent to
  $\neg\chi_1 \wedge \neg\chi_2 \wedge ... \wedge \neg\chi_m$
- Each $\neg\chi_i$ is equivalent to a disjunction of literals, by again using generalised De Morgan's Laws
- Thus
  - Every formula of propositional logic is logically equivalent to a conjunction of disjunctions of literals, i.e., a conjunction of **clauses**
    - That is, every formula of propositional logic is equivalent to a formula in **conjunctive normal form**

4 A spot of practice

A spot of practice

Durham University

- We wish to convert the formula $\varphi = ((\neg p \wedge q) \vee r) \wedge \neg((r \wedge p) \vee \neg q)$ into c.n.f.

p	q	r	$((\neg p \wedge q) \vee r) \wedge \neg((r \wedge p) \vee \neg q)$	$\neg\varphi$
T	T	T	F T F T T T **F** F T T T TFT	**T**
T	T	F	F T F T F F **F** T F F T FFT	**T**
T	F	T	F T F F T T **F** F T T T TTF	**T**
T	F	F	F T F F F F **F** F F F T TTF	**T**
F	T	T	T F T T T T **T** T T F F FFT	**F**
F	T	F	T F T T T F **T** T F F F FFT	**F**
F	F	T	T F F F T T **F** F T F F TTF	**T**
F	F	F	T F F F F F **F** F F F F TTF	**T**

- So, $\neg\varphi$ is equivalent to

$$(p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge r) \vee (p \wedge \neg q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge r) \vee (\neg p \wedge \neg q \wedge \neg r).$$

- Hence, φ is equivalent to the c.n.f. formula

$$(\neg p \vee \neg q \vee \neg r) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge (\neg p \vee q \vee r) \wedge (p \vee q \vee \neg r) \wedge (p \vee q \vee r).$$

slide 9 of 16

5 Converting to c.n.f syntactically

- We can often establish normal forms ”syntactically”
- Consider the formula

$$\begin{aligned}\varphi \quad & ((\neg p \wedge q) \vee r) \wedge \neg((r \wedge p) \vee \neg q) \\ & \equiv ((\neg p \vee r) \wedge (q \vee r)) \wedge (\neg(r \wedge p) \wedge q) \\ & \equiv (\neg p \vee r) \wedge (q \vee r) \wedge ((\neg r \vee \neg p) \wedge q) \\ & \equiv (\neg p \vee r) \wedge (q \vee r) \wedge ((\neg r \wedge q) \vee (\neg p \wedge q)) \\ & \equiv (\neg p \vee r) \wedge (q \vee r) \wedge (((\neg r \wedge q) \vee \neg p) \wedge ((\neg r \wedge q) \vee q)) \\ & \equiv (\neg p \vee r) \wedge (q \vee r) \wedge (\neg r \vee \neg p) \wedge (q \vee \neg p) \wedge (\neg r \vee q) \wedge (q \vee q) \\ & \equiv (\neg p \vee r) \wedge (q \vee r) \wedge (\neg r \vee \neg p) \wedge (q \vee \neg p) \wedge (\neg r \vee q) \wedge q\end{aligned}$$

- In the ”semantic” approach, i.e., using truth tables we are stuck with using exponentially sized truth tables
- However with the ”syntactic” approach, i.e., using known equivalences
  - We can often achieve our aims much more quickly, though this often requires cunning

Non-Examinable from here on

6 An application: SAT-solving

- The power of propositional logic of quite remarkable as computationally complex problems can be described using logic
- The aim of SAT-solving is
  - To encode a problem X as a propositional formula φ so that
    - A solution to X corresponds to φ having a satisfying truth assignment
  - To employ algorithms to solve the satisfiability problem (SAT) for φ (and so X)
- The SAT problem is to decide if a propositional formula has a satisfying truth assignment. It is extremely hard to solve
  - In fact, it is NP-complete, even if the formula is given in c.n.f, so it takes time exponential in the size of the formula to solve
- However, modern day SAT-solvers can give extremely good results
  - Note that all modern day SAT-solvers need their inputs to be in c.n.f