Lecture 6:
Proof

Aims:

- To see more examples of natural deduction;
- To see examples of formal proofs;
- To discuss what we mean by the soundness and completeness of a deduction system;
- To see examples of informal proofs.

6.1 More Derivations

- We start with three more examples of what we were doing at the end of the previous lecture.

  Show $\{p \Leftrightarrow q, q \Rightarrow \neg r\} \vdash p \Rightarrow \neg r$

  Show $\{p \Rightarrow r, q \Rightarrow r, p \vee q\} \vdash r$

  Show $\{p \Rightarrow q, \neg q\} \vdash \neg p$

6.2 Formal Proof

- A *proof* is a derivation from no premisses. Obviously, they're almost bound to begin with some assumptions!
- E.g. prove that $\vdash p \Rightarrow p$

  1. Subderivation:
     1.1 p assumption
     1.2 p 1.1, repetition
  2. $p \Rightarrow p$ 1, $\Rightarrow$-I

- Here's another example of a proof. We'll prove that $\vdash ((p \wedge q) \vee (\neg p \wedge r)) \Rightarrow (q \vee r)$

1.
    1.1
    1.2
        1.2.1
        1.2.2
        1.2.3
    1.3
        1.3.1
        1.3.2
        1.3.3
    1.4
2.

6.3 Soundness and Completeness

- A deduction system derives wffs using syntactic operations alone, without reference to the semantics. We could invent a set of inference rules that license even quite bizarre inferences. However, if a deduction system is to be useful, the wffs we can derive from a set of wffs must tie in with logical consequences of that set.

  At least two things may go wrong.

- An inference rule might license an inference that is not a logical consequence. For example,

  $\Rightarrow$-BAD!

$$\frac{W_1 \Rightarrow W_2, W_2}{W_1}$$

  We can use this rule to show, e.g., $\{p \Rightarrow q, q\} \vdash p$. But, we know that $\{p \Rightarrow q, q\} \not\models p$. (We showed this earlier. If you can't find it, draw a truth table and confirm it again!)

- An inference rule of this kind is said to be *unsound*. A deduction system that contains such a rule is *unsound*.

  An inference rule is *sound* if the conclusions one can infer from any set of wffs using the rule are logical consequences of the set of wffs.

  A deduction system is *sound* if it contains only sound inference rules.

- Another thing that may go wrong is that there may be logical consequences of a set of wffs that the deduction system fails to derive.

  For example, we know that $\{p \vee q, \neg p\} \models q$. (Confirm it with a truth table if you don't believe it!). But if our deduction system has insufficient inference rules or insufficiently 'powerful' inference rules, it may be that we cannot show that $\{p \vee q, \neg p\} \vdash q$. In this case, the deduction system is said to be *incomplete*.

  A deduction system is *complete* if every logical consequence of any set of wffs can also be derived from the set of wffs.

- Here's a summary of soundness and completeness.

  A deduction system is *sound* so long as for any set of wffs A and any wff W,

$$\text{if } A \vdash W \text{ then } A \models W$$

A deduction system is *complete* so long as for any set of wffs A and any wff W,

$$\text{if } A \models W \text{ then } A \vdash W$$

- Soundness is essential, but completeness might be sacrificed in Computer Science uses of deduction systems, where the efficiency of the automated deduction system is important.

  The soundness and completeness of a deduction system is obviously something that the logician who proposes the system should prove.

- If a deduction system is sound and complete, then for any wff W,

  $$W \text{ is valid if and only if } \vdash W$$

  i.e. any valid wff will be provable and vice versa.

  And, for any set of wffs A and wff W,

  there will be no interpretation that satisfies all the wffs in A
  if and only if $A \vdash W \wedge \neg W$

  (When a wff of the form $W \wedge \neg W$ can be derived from a set of wffs A, we say that A is *inconsistent*.)

6.4 Informal Proof

- In the correctness proofs that we will be doing soon, we won't be working with propositional symbols (ps and qs). We will have statements about programs, e.g. statements about the values of the variables in the programs, such as $x > 2$ or $x = 0 \wedge y > 10$.

  Mostly, we'll need to prove that a conditional holds.

- Suppose we need to prove, e.g., $\vdash x > 5 \Rightarrow x > 0$.

  Strictly, in propositional logic this cannot be proved. Think of it written using ps and qs! There are two different statements, linked by a conditional. So you are being asked to prove $\vdash p \Rightarrow q$, which is simply not provable.

  But you can't help inspecting the inner details of the two statements and realising, *given additionally what you know about arithmetic*, that it is provable.

- We will prove such statements in two (informal) ways. (It is up to you to be smart enough to choose which of the two to use in any particular case.)

- **Method 1.** Use the laws of the algebra of propositions, plus whatever you know about arithmetic, to show the statement is valid (i.e. always true).

- Example. To show $\vdash \textbf{True} \Rightarrow x + x = 2x$.

- **Method 2.** Use the rules of natural deduction, plus whatever you know about arithmetic.

- Example. To show $\vdash x = n \Rightarrow x + 1 = n + 1$.