4/1 - Natural Deduction

Let's start with something totally different:

- Thm: there are infinitely many prime numbers
- We can write out a proof sketch by contradiction

- Do we believe this? Seems reasonable.
- What assumptions did we make? Lots of them:
  - There are such things as primes
  - That we're working with all integers, etc.
- What kind of thing is the theorem/proof?
  - For humans: a proof that can we can use to convince ourselves
  - For computers: can't really use it to formally verify it with a computer
  - It's a social object - just for us humans to convince ourselves
- What do we want to do with proofs?
  - We want to automatically *check* proofs
  - We want to automatically *find* proofs (search the proof space)
  - There are some limits to what we can do automatically

We want to focus now on **validity** as opposed to **satisfiability**:

- *Satisfiability* - is there some assignment that makes the thing true?
- *Validity* - is this thing always true
  - If the preconditions hold, the statement always holds
- Essentially, two approaches to proving things
  - We've seen searching for a counterexample (using satisfiability)
  - Now, we'll look at finding a deductive proof

Natural Deduction:

- System for proving things in propositional logic
- We have generally:
  - `premises |- conclusion`
- In natural deduction, we have a few rules we can use to ***rewrite*** formulas:
  - `And elimination (left): x ∧ y |- x`
  - `And elimination (right): x ∧ y |- y`
  - `And introduction: x, y |- x ∧ y`
  - And similarly for or (`∨`), implication (`=>`) as well
- Notice we generally have two types of rules:
  - Elimination rules (where we eliminate the symbol)
  - Introduction rules (where we introduce the symbol)
- For implication introduction,
  - If we have a subproof `a … b`, then we get `a => b`
  - i.e. `a … b |- a => b`