Proof Approaches
(A Review)

How Should We Approach a Proof?

- In general, there are three types of approaches you can take at any given step in a proof:
    1. Attack the proof goal (consequent) directly, either by simplifying it (e.g., “apply”ing an introduction rule) or showing how the proof goal follows “exact”ly from existing facts (antecedents)
    2. Use existing facts (e.g., using an elimination rule) to create new facts
    3. Introduce new facts from introduction rules (e.g., or.inl and true.intro), lemmas, or axioms (e.g., the axiom of the excluded middle — if allowed)
- There’s never only one way to solve a proof$^{\text{[citation needed]}}$, but familiarizing yourself with proof patterns that work will be useful

Equality

- The introduction rule for equality is eq.refl, but rfl usually works just as well
- The most common elimination rule for equality would probably be eq.subst (see also rewrite or rw)
- Other elimination rules include eq.symm and eq.trans

Conjunction

- For dealing with conjunction as a goal, you can either use the and.intro introduction rule, or use split to simplify the goal into two smaller goals
- For using conjunction as a fact (antecedent), you will usually use .left or .right on the antecedent to pull off the individual pieces
  - You can also uses cases

Disjunction

- For dealing with disjunction as a goal, you will usually use or.inl or or.inr
- For using disjunction as a fact, you will usually want to use cases

Implication/Negation

- As with universal quantification, for implication we don’t have a simple introduction rule
  - Our most common approach is to simplify the goal by simply ”assuming the antecedent” of the implication, and then proving the consequent follows
- For elimination, we will usually apply the implication to a proof of the antecedent, resulting in a proof of the consequent
- Negation is simply a special type of implication — that something implies false

Bi-implication

- When dealing with a bi-implication as a goal, we will almost always use split to simplify the goal into two goals (one in either direction)
- When dealing with bi-implication as a fact, we will typically want to use .left or .right to introduce implication facts in either the forward or backward direction

Universal Quantification

- Unlike with equality (`eq.refl`) we don't have a simple introduction rule for universal quantification
    - Instead we have an algorithm: for a type T and predicate Q that takes an instance of type T, if we want to prove ∀(t: T), (Q t), we
        - Assume we have an arbitrary t of type T
        - Prove that from that assumption the predicate Q holds for that arbitrary
- As with introduction, there is not a named elimination rule, but rather a simple algorithm
    - To use an existing proof (i.e., a fact in the antecedent or "above the turnstile" for a goal), the approach is to *instantiate* the universal quantifier with a specific value

Existence proof

- To approach a proof where the goal is existence, we typically apply exists.intro to an existence goal with an instance where the goal should be true, then prove that the goal is true for that instance
- To use an existence proof as a fact, we will typically follow the pattern of:
  1. Applying exists.elim to the fact to get a forall statement with an implication
  2. Then assuming a witness from the forall statement, and then assuming the antecedent of the implication
     - See forgetAProperty and reverseAProperty in exists_properties.lean and problems 8a and 8c from the practice exam

Lack of existence

- Proving something of the form ¬∃(A B C: Prop), (sat_pred A B C) (see problem 8b from practice exam) can usually be accomplished with the following pattern:
  1. Remember that this goal is actually an implication that the existence of an A B and C that work for sat_pred implies false, so assume the antecedent that there does exist such an A, B, and C
  2. Apply exists elimination on the assumed antecedent (which antecedent will change on repeated application of this step)
  3. Assume a witness and a new proof of the witness as an antecedent
  4. Repeat steps 2 and 3 for each variable in the existence statement until you have an antecedent without an existence
  5. Use the axiom of the excluded middle to effectively build a truth table, for which you can prove sat_pred is not true for any values of A, B, or C
- The last step might require up to 2^n – 1 (i.e, 7 when there are 3 propositions) applications of the axiom of the excluded middle

Fin