Isabelle/HOL Exercises
Logic and Sets

A Riddle: Rich Grandfather

First prove the following formula, which is valid in classical predicate logic, informally with pen and paper. Use case distinctions and/or proof by contradiction.

If every poor man has a rich father,
then there is a rich man who has a rich grandfather.

theorem
```
"∀x. ¬ rich x ⟶ rich (father x) ⟹
∃x. rich (father (father x)) ∧ rich x"
```

Proof
(1) We first show: `∃x. rich x`.
Proof by contradiction.
  Assume `¬ (∃x. rich x)`.
    Then `∀x. ¬ rich x`.
    We consider an arbitrary `y` with `¬ rich y`.
    Then `rich (father y)`.
(2) Now we show the theorem.
Proof by cases.
  Case 1: `rich (father (father x))`.
    We are done.
  Case 2: `¬ rich (father (father x))`.
    Then `rich (father (father (father x)))`.
    Also `rich (father x)`,
    because otherwise `rich (father (father x))`.
qed

Now prove the formula in Isabelle using a sequence of rule applications (i.e. only using the methods *rule*, *erule* and *assumption*).

theorem
```
"∀x. ¬ rich x ⟶ rich (father x) ⟹
∃x. rich (father (father x)) ∧ rich x"
```

```
apply (rule classical)
apply (rule exI)
apply (rule conjI)

  apply (rule classical)
  apply (rule allE) apply assumption
  apply (erule impE) apply assumption
  apply (erule notE)
  apply (rule exI)
  apply (rule conjI) apply assumption
  apply (rule classical)
  apply (erule allE)
  apply (erule notE)
  apply (erule impE) apply assumption
  apply assumption

apply (rule classical)
apply (rule allE) apply assumption
apply (erule impE) apply assumption
apply (erule notE)
apply (rule exI)
apply (rule conjI) apply assumption
apply (rule classical)
apply (erule allE)
apply (erule notE)
apply (erule impE) apply assumption
apply assumption
done
```

Here is a proof in Isar that resembles the informal reasoning above:

```
theorem rich_grandfather: "∀x. ¬ rich x ⟶ rich (father x) ⟹
  ∃x. rich x ∧ rich (father (father x))"
proof -
  assume a: "∀x. ¬ rich x ⟶ rich (father x)"
```

(1)

```
 have "∃x. rich x"
  proof (rule classical)
    fix y
    assume "¬ (∃x. rich x)"
    then have "∀x. ¬ rich x" by simp
    then have "¬ rich y" by simp
    with a have "rich (father y)" by simp
```

```
    then show ?thesis by rule
  qed
  then obtain x where x: "rich x" by auto
```

(2)

```
 show ?thesis
  proof cases
    assume "rich (father (father x))"
    with x show ?thesis by auto
  next
    assume b: "¬ rich (father (father x))"
    with a have "rich (father (father (father x)))" by simp
    moreover have "rich (father x)"
    proof (rule classical)
      assume "¬ rich (father x)"
      with a have "rich (father (father x))" by simp
      with b show ?thesis by contradiction
    qed
    ultimately show ?thesis by auto
  qed
qed
```