A Less trivial resolution proof

1. No one likes anyone of whom he is afraid
   \( \forall x, y. A(x, y) \rightarrow \neg L(x, y) \) (domain: people)

2. No one likes anyone who does not like him
   \( \forall x, y. \neg L(y, x) \rightarrow \neg L(x, y) \)

3. All paranoids are afraid of everyone except their friends
   \( H(x, y) : x \) has friend \( y \)
   \( \forall x, y. (P(x) \land H(x, y)) \rightarrow A(x, y) \)

4. There is a paranoid, and all of his friends are also paranoid
   \( \exists x. P(x) \land \forall y. H(x, y) \rightarrow P(y) \)

Conclusion:
There is someone whom all non-paranoids dislike
\( \exists x \forall y. \neg P(y) \rightarrow \neg L(y, x) \)

Denial
\( \exists x \forall y. \neg P(y) \rightarrow \neg L(y, x) \)

Intuitive argument:
- From 3, all paranoids are afraid of everyone except their friends, so by 1 they don't like anyone except their friends, so by 2, everyone but a paranoid's friends dislike him.
- But by 4 there is a paranoid, so by the previous argument all but his friends dislike him; and since all his friends are paranoids, all non-paranoids dislike him.

So, use 3, 1, 2, 4, 5

Clause form
1. \( \neg A(x, y) \lor \neg L(x, y) \)
2. \( L(y, x) \lor \neg L(x, y) \)
3. \( \neg P(x) \lor H(x, y) \lor A(x, y) \)
4. \( P(C) \)
4'. \( \neg H(C, y) \lor P(y) \)
5. \( \forall x \exists y. \neg (P(y) \lor \neg L(y, x)) \)
   \( \therefore \neg \neg \neg L(f(x), x) \)
5'. \( L(f(x), x) \)
Finding the m.g.u. of \((a_1, \ldots, a_n), (b_1, \ldots, b_n)\)

1. Standardize apart
2. \(\text{mgu} := ()\)
3. Scan the lists in parallel, left to right
   a. If end of lists is reached, return \(\text{mgu}\) (i.e., successful exit, even if \(\text{mgu} = ()\))
   b. If current symbols identical, continue at (a) with the next pair (i.e., advance pointers)
   c. If neither symbol is a variable, return \(\text{FAIL}\)
   d. One symbol is a variable, say \(v\), and the other is the beginning of some term \(t\) (a variable, constant, or functional expression).
      If \(v\) occurs in \(t\), return \(\text{FAIL}\)
      else (i) push \((v, t)\) onto \(\text{mgu}\)
      (ii) make this substitution throughout both lists, from the current pointers forward
      (iii) move the pointers over the (now identical) \(t\) terms

\(e.g., (x, f(x)), (y, y)\) can't be unified

Grads: Study the mathematical example (group theory) in the notes.
Unification example

\[ P(s, g(s), t, h(s, t), u, k(s, t, u)) \]
\[ \Rightarrow P(v, w, m(w), x, n(w, x), y) \]

Factoring example

\[ P(y, x, a) \lor P(f(y), z, z) \lor P(f(a), b, x) \]

Formal statement of resolution (narrow)

\[ \pi(s_1, \ldots, s_k) \lor \alpha_1, \ldots, \alpha_m, \ \Rightarrow \pi(t_1, \ldots, t_k) \lor \beta_1, \ldots, \beta_n \]
\[ \alpha_i \lor \beta_i \lor \alpha_i \lor \beta_i \]

where \((t_1, \ldots, t_k), (s_1, \ldots, s_k)\) are unifiable with \(m_\psi \beta_i\);
\[ a_i = a, x, a_i = a, \beta, \text{ etc.} \]
\[ \pi(\ldots), \ \pi(\ldots) \text{ need not be initial} \]

Deductive Problem Solving & Q.A.

Cordell Green (IJCAI '69, Mach.Int.'69):

To solve a problem or answer a question,
1. prove that a solution/answer exists;
2. extract solution/answer from variable bindings of the proof.

Yes-no questions

Make "parallel" attempts to prove or disprove the proposition in question
\[ \varphi \rightarrow \text{yes, no} \]
\[ \varphi \rightarrow \text{refute } \neg \varphi \rightarrow \text{yes} \]
\[ \varphi \rightarrow \text{refute } \varphi \rightarrow \text{no} \]

Reconsider Snoopy question: Does Spot x bark
**Wh- Questions ("fill in the blank(s)")**

E.g.,
1. \( \forall x. \text{At}(\text{Bob}, x) \rightarrow \text{At}(\text{Carol}, x) \)  Carol is wherever Bob is
2. \( \text{At}(\text{Bob}, \text{dance}) \)  Bob is at the dance

Q: Where is Carol?

Prove: There exists a place where Carol is

\[ \exists x. \text{At}(\text{Carol}, x) \]

Denial: \( \neg \exists x. \text{At}(\text{Carol}, x) \)

3. i.e., \( \forall x. \neg \text{At}(\text{Carol}, x) \)

Refutation & Ans - extraction

1. \( \neg \text{At}(\text{Bob}, x) \lor \text{At}(\text{Carol}, x) \)  given
2. \( \text{At}(\text{Bob}, \text{dance}) \)  given
3. \( \neg \text{At}(\text{Carol}, x) \lor \text{Ans}(x) \)  denial
4. \( \Box \neg \text{Ans}(\text{Bob}) \)  \( r[1,2] \)
5. \( \Box \neg \text{Ans}(\text{dance}) \)  \( r[3,4] \)

**Good answers, bad answers**

E.g.,
1. \( \forall x. \exists y. \text{Works-for}(x, y) \)  Everyone works for someone
2. \( \text{Works-for}(\text{Bob}, \text{Carol}) \)  Bob works for Carol

Q: \( \exists x. \text{Works-for}(\text{Bob}, x) \)  Who does Bob work for?

Denial: \( \forall x. \neg \text{Works-for}(\text{Bob}, x) \)

First proof

1. \( \text{Works-for}(x, \text{F}(x)) \)  given
2. \( \text{Works-for}(\text{Bob}, \text{Carol}) \)  given
3. \( \neg \text{Works-for}(\text{Bob}, x) \lor \text{Ans}(x) \)  denial
4. \( \Box \neg \text{Ans}(\text{Bob}) \)  \( r[1,2] \)

Bob works for his employer

Second proof

4. \( \Box \neg \text{Ans}(\text{Carol}) \)  \( r[2,2] \)

Generally: try to get Skolem-free answers by preferring proof steps that don't substitute Skolem constants/functions into the answer.

But: there may only be Skolem answers — e.g., "Whom does every non-paranoid dislike?"