RESOLUTION STRATEGIES

- Improve efficiency
- Preserve completeness

Clause elimination

- "pure" literal elimination
  \( P(A) \lor Q(x, A) \)  
  \( \neg P(...) \lor \ldots \)  
  no such literal

- unicity elimination
  \( P(A) \lor \neg P(A) \lor P(B), \ P(x) \lor \neg P(x) \lor Q(A, x) \)  
  (cannot help in a refutation)
  e.g., we would have to have clauses with
  literals resolvable against \( P(A), \neg P(A) \ldots \)  
  but then these clauses can be resolved against each other.

- Subsumption elimination
  \( P(C) \)  
  \( P(A) \lor Q(A, B) \)  
  \( P(x) \)  
  \( P(y) \lor Q(y) \)  
  \( P(F(x), B) \lor Q(y) \)  
  subsumes \( P(F(B), B) \lor Q(A) \lor R(w) \)

Proof strategies

- Breadth-first
  complete but inefficient

- Depth-first
  efficient but incomplete

- Unit resolution
  (one or both parents length 1)
  efficient but incomplete

- Unit preference
  (use unit resolution if possible)
  efficient & complete

- Set of support
  denial clauses "guide" proof

- Sos steps
  \( \neg P(x) \lor Q(x) \)  
  \( \neg R(x, y) \)  
  \( P(A) \lor R(A, B) \)  
  \( \neg Q(x) \lor P(x) \)  
  \( Q(A) \lor R(A, B) \)  
  \( P(A) \lor R(A, B) \)  
  \( \neg Q(A) \lor P(A) \)  
  non-Sos steps
- Connection-graph method: (grads)
- Ordered resolution
  Resolve initial literals only
  Complete for Horn clauses (prolog)
    \[ P(A) :\neg \quad \rightarrow \quad Q(x), Q(x) :\neg P(x) \]
  \[ P(A) \leftarrow \text{True}, \quad \text{False} \equiv Q(x), \quad Q(x) \equiv P(x) \]
  \[ P(A), \neg Q(x) \quad P(x) \rightarrow Q(x) \]

Quacks \( x \) := Happy \( x \), Duck \( x \)
  \[ \neg \quad \text{Denial} \]
  \[ \neg \quad \text{Model} \]

If we also have Happy(Daffy) := & Duck(Daffy) :=
we'll have a refutation & thus a proof of Quacks (Daffy);

The latex notes contain James Allen's less trivial example,
Where we show that some bird quacks: \(-\) Bird \( x \), Quacks \( x \)

- Model Generation (elimination)
  Efficient!
    Think of "model" as "all true ground atoms"

Use Prolog on Horn subset
If refuted \( \rightarrow \) done
If fail \( \rightarrow \) expand Horn set using a +ve literal from a non-Horn clause whose antecedent (if any) we can prove.

If we refute all possibilities \( \rightarrow \) done
O.w. expand Horn set again, ...

E.g. (BTW, SAT methods (later) can solve this too)

\[ \neg P, \neg Q, \neg RV \quad Q, \neg P, \neg Q, \neg R, \neg P, \neg Q \]

(1) (2) (3) (4) (5) (6)

\[ \text{can't refute} \]
\[ \text{can prove antecedent, which is empty} \]
\[ \text{Add } R \text{ from (6)} \]
\[ \text{can't refute} \]
\[ \text{retain } R \text{ in model} \]
\[ \text{can prove } S \text{ (eliminate -ve literals)} \]

\[ \text{Add } P \text{ from (5)} \]
\[ \text{can refute} \]
\[ \text{Add } Q \text{ ... can refute} \]

Retract R, Q; back to (6), try options P, Q; fail; DONE!