Inference by Resolution

$$\frac{\alpha \vee \beta \;,\quad \neg\alpha}{\beta}$$

Resolution needs sentences in **Conjunctive Normal Form (CNF)** as a **conjunction of disjunctions of literals.**

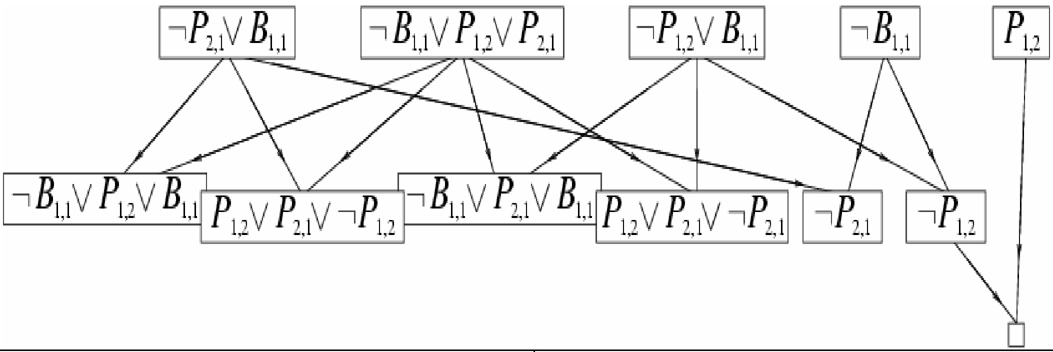
Resolution is sound and complete. It works by proving a contradiction, i.e. to show that KB ⊨ α we show that ($KB \wedge \neg\alpha$) is unsatisfiable.

Example: $KB = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$ $\alpha = \neg P_{1,2}$

The first step is Converting ($KB \wedge \neg \alpha$) into CNF.

Here, will need only to convert ($B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$).

1. Eliminate ⇔, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
   $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$
2. Eliminate ⇒, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$
3. Move ¬ inwards using de Morgan's rules and double-negation:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$
4. Apply distributivity law (∧ over ∨) and flatten to obtain CNF form:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$



Inference by Forward chaining (FC)

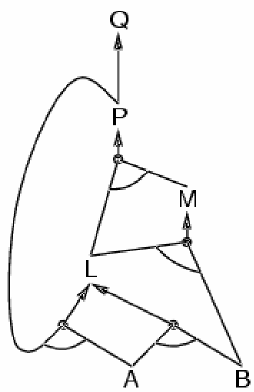
The Idea is to fire any rule whose premises are satisfied in the *KB*, add its conclusion to the *KB*, until query is found. We will first need to construct the to facilitate the process of proof

Inference by Backward chaining (BC)

The Idea is to work backwards from the query *q*: to prove *q* by BC, check if *q* is known already, or prove by BC all premises of some rule concluding *q*. BC Avoids loops by checking if new subgoal is already on the goal stack.

AND-OR graph

$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B



FC is data-driven, automatic, unconscious processing e.g., object recognition, routine decisions. FC May do lots of work that is irrelevant to the goal.

BC is goal-driven, appropriate for problem-solving, e.g., where are my keys? Complexity of BC can be much less than linear in size of KB.