Lecture notes for *Principles of Artificial Intelligence* (COMS 4720/5720)
Yan-Bin Jia, Iowa State University

Proof Using Resolution

Outline

I. Rule of resolution

II. Conjunctive normal forms

III. Resolution refutation

* Figures are from the textbook site unless the source is specifically cited.

I. Resolution

An inference algorithm i is

sound if $KB \vDash \alpha$ whenever $KB \vdash_i \alpha$

complete if $KB \vdash_i \alpha$ whenever $KB \vDash \alpha$

I. Resolution

An inference algorithm i is

sound if $KB \models \alpha$ whenever $KB \vdash_i \alpha$

complete if $KB \vdash_i \alpha$ whenever $KB \models \alpha$

- Inference rules covered so far are sound.
- The inference algorithms using them may not be complete.

I. Resolution

An inference algorithm i is

sound if $KB \vDash \alpha$ whenever $KB \vdash_i \alpha$

complete if $KB \vdash_i \alpha$ whenever $KB \vDash \alpha$

- Inference rules covered so far are sound.
- The inference algorithms using them may not be complete.

resolution + a complete search algorithm $=$ a complete inference algorithm

I. Resolution

An inference algorithm i is

sound if $KB \models \alpha$ whenever $KB \vdash_i \alpha$

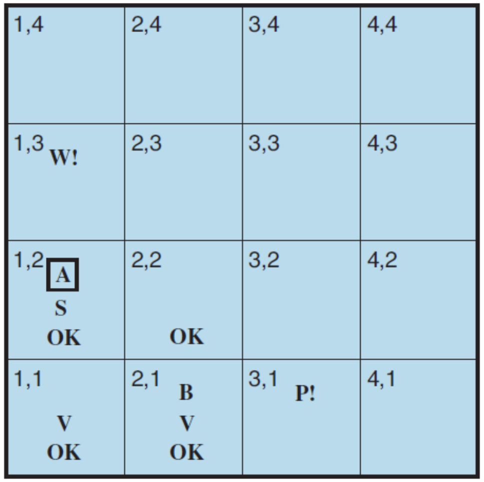
complete if $KB \vdash_i \alpha$ whenever $KB \models \alpha$

- Inference rules covered so far are sound.
- The inference algorithms using them may not be complete.

resolution + a complete search algorithm $=$ a complete inference algorithm

single inference rule

Wumpus World Revisited



KB:

R_1: $\neg P_{1,1}$

R_2: $B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})$

R_3: $B_{2,1} \Leftrightarrow (P_{1,1} \lor P_{2,2} \lor P_{3,1})$

R_4: $\neg B_{1,1}$

R_5: $B_{2,1}$

Rules

$$\frac{R_2\colon\ B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})}{R_6\colon\ \left(B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})\right) \land \left((P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1}\right)}$$

biconditional elimination

$$\frac{}{R_7\colon\ (P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1}}$$

and-elimination

$$\frac{R_4\colon\ \neg B_{1,1} \qquad R_8\colon\ \neg B_{1,1} \Rightarrow \neg(P_{1,2} \lor P_{2,1})}{R_9\colon\ \neg(P_{1,2} \lor P_{2,1})}$$

logical equivalence

modus ponens

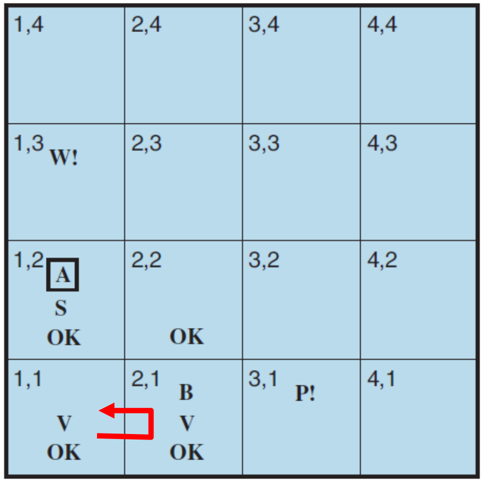
$$\frac{}{R_{10}\colon\ \neg P_{1,2} \land \neg P_{2,1}}$$

De Morgan's rule

$$\frac{}{R_{11}\colon\ \neg P_{1,2}}$$

and-elimination

Wumpus World Revisited



Agent: [1,1] → [2,1] → [1,1]

KB:

R_1: $\neg P_{1,1}$

R_2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

R_3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

R_4: $\neg B_{1,1}$

R_5: $B_{2,1}$

Rules

$$\frac{R_2:\ B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})}{R_6:\ \big(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})\big) \wedge \big((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}\big)}$$

biconditional elimination

$$\frac{R_6:\ \big(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})\big) \wedge \big((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}\big)}{R_7:\ (P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}}$$

and-elimination

$$\frac{R_7:\ (P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}}{R_8:\ \neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1})}$$

logical equivalence

$$\frac{R_4:\ \neg B_{1,1} \qquad R_8:\ \neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1})}{R_9:\ \neg(P_{1,2} \vee P_{2,1})}$$

modus ponens

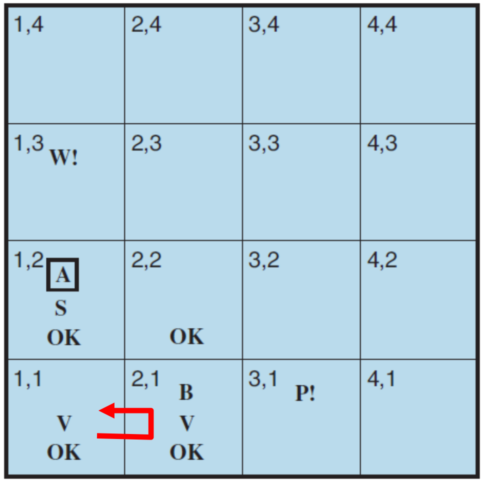
$$\frac{R_9:\ \neg(P_{1,2} \vee P_{2,1})}{R_{10}:\ \neg P_{1,2} \wedge \neg P_{2,1}}$$

De Morgan's rule

$$\frac{R_{10}:\ \neg P_{1,2} \wedge \neg P_{2,1}}{R_{11}:\ \neg P_{1,2}}$$

and-elimination

Wumpus World Revisited



KB:

R_1: $\neg P_{1,1}$

R_2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

R_3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

R_4: $\neg B_{1,1}$

R_5: $B_{2,1}$

Rules

Agent: [1,1] → [2,1] → [1,1]

R_2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

biconditional elimination

R_6: $\big(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})\big) \wedge \big((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}\big)$

and-elimination

R_7: $(P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}$

logical equivalence

R_4: $\neg B_{1,1}$ R_8: $\neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1})$

modus ponens

R_9: $\neg(P_{1,2} \vee P_{2,1})$

De Morgan's rule

R_{10}: $\neg P_{1,2} \wedge \neg P_{2,1}$

and-elimination

R_{11}: $\neg P_{1,2}$

R_6: $\big(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})\big) \wedge \big((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}\big)$

R_7: $(P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}$

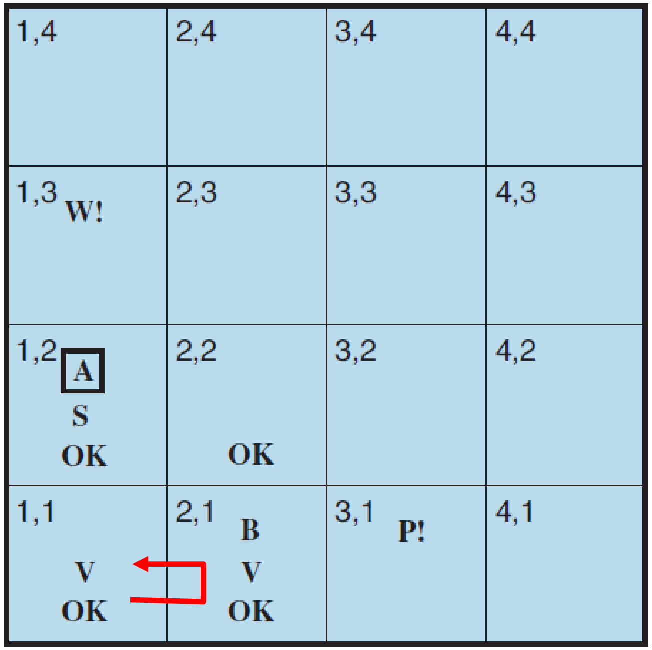
R_8: $\neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1})$

R_9: $\neg(P_{1,2} \vee P_{2,1})$ // R_4, R_8

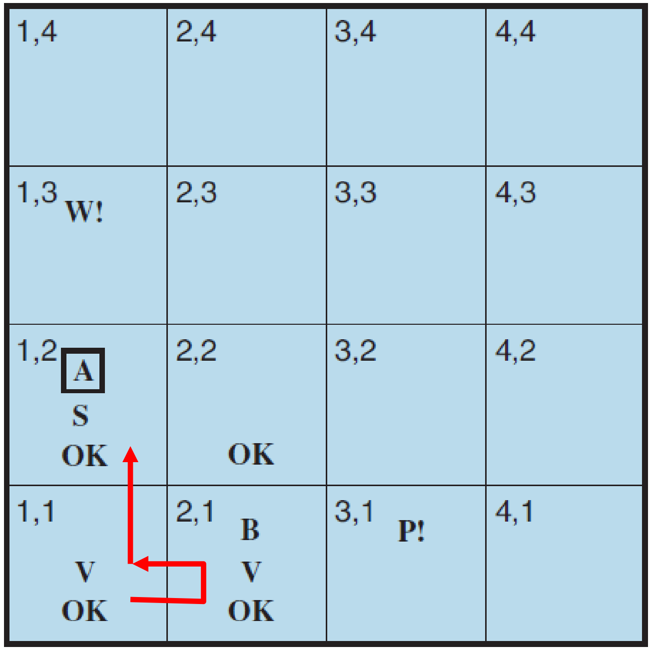
R_{10}: $\neg P_{1,2} \wedge \neg P_{2,1}$

Added to KB via inferences

(cont'd)

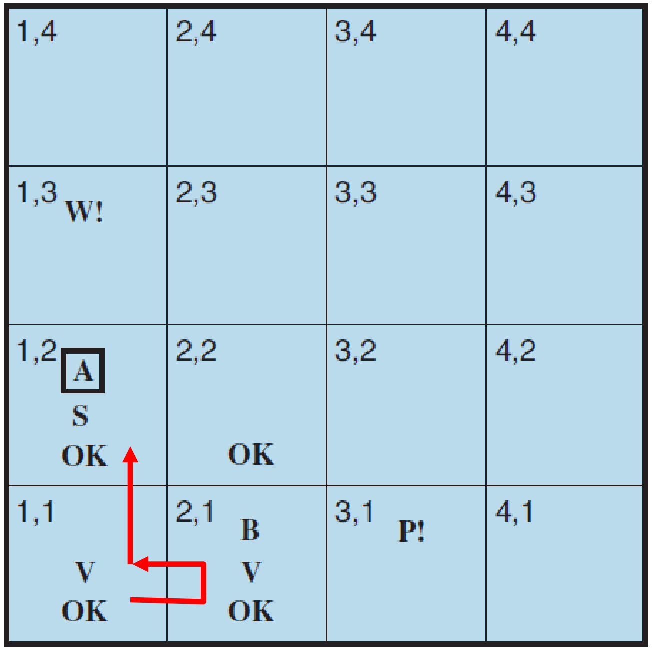


(cont’d)



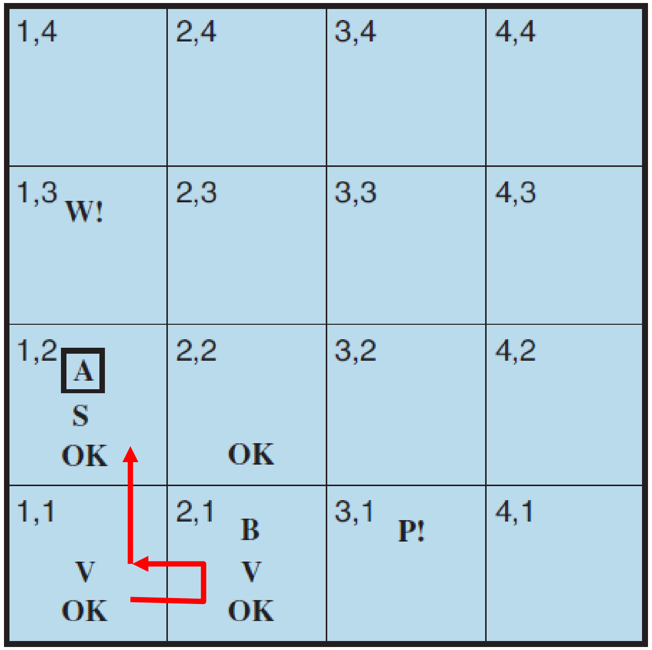
1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

(cont'd)



[1,1] → [1,2]: stench but no breeze

(cont'd)



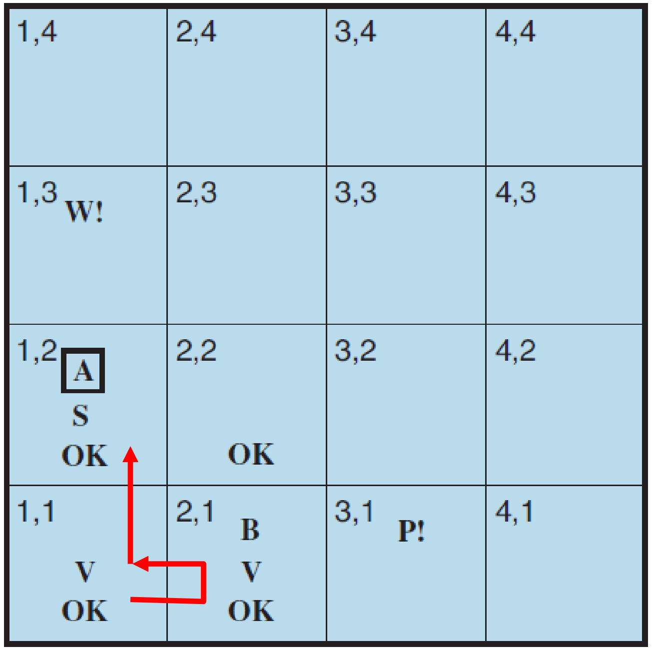
[1,1] → [1,2]: stench but no breeze

Add to KB:

R_{11}: $\neg B_{1,2}$

R_{12}: $B_{1,2} \Leftrightarrow \left(P_{1,1} \vee P_{2,2} \vee P_{1,3}\right)$

(cont'd)



[1,1] → [1,2]: stench but no breeze

Add to KB:

R_{11}: $\neg B_{1,2}$

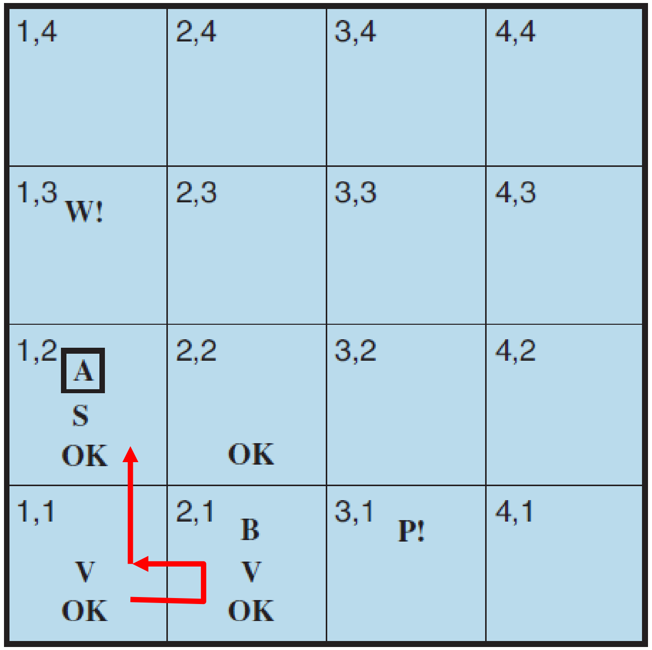
R_{12}: $B_{1,2} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3})$

$$\Downarrow \quad \neg B_{1,2} \Rightarrow \neg P_{1,1} \wedge \neg P_{2,2} \wedge P_{1,3}$$

R_{13}: $\neg P_{2,2}$

R_{14}: $\neg P_{1,3}$

(cont’d)



[1,1] → [1,2]: stench but no breeze

Add to KB:

$R_{11}: \quad \neg B_{1,2}$

$R_{12}: \quad B_{1,2} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3})$

$\Downarrow$

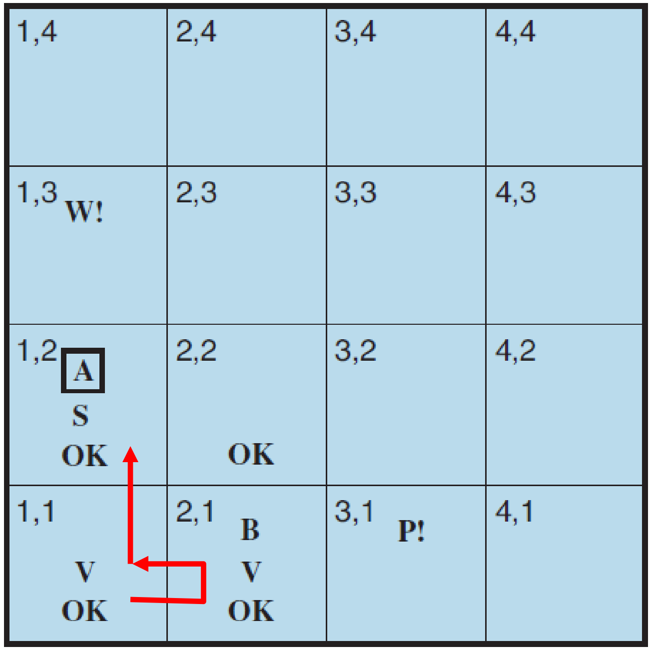
$\neg B_{1,2} \Rightarrow \neg P_{1,1} \wedge \neg P_{2,2} \wedge P_{1,3}$

$R_{13}: \quad \neg P_{2,2}$

$R_{14}: \quad \neg P_{1,3}$

$R_3\!: \; B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

(cont’d)



[1,1] → [1,2]: stench but no breeze

Add to KB:

R_{11}: $\neg B_{1,2}$

R_{12}: $B_{1,2} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{1,3})$

$\Downarrow$ $\neg B_{1,2} \Rightarrow \neg P_{1,1} \wedge \neg P_{2,2} \wedge P_{1,3}$

R_{13}: $\neg P_{2,2}$

R_{14}: $\neg P_{1,3}$

R_3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$

$\Downarrow$ biconditional elimination

R_5: $B_{2,1}$

R_{15}: $P_{1,1} \vee P_{2,2} \vee P_{3,1}$

Resolvent

R_{13}: $\neg P_{2,2}$ R_{15}: $P_{1,1} \lor P_{2,2} \lor P_{3,1}$

Resolvent

$R_{13}\colon\ \neg P_{2,2}$ $R_{15}\colon\ P_{1,1} \vee P_{2,2} \vee P_{3,1}$

resolving the two literals that are negations of each other

Resolvent

R_{13}: $\neg P_{2,2}$ R_{15}: $P_{1,1} \lor P_{2,2} \lor P_{3,1}$

resolving the two literals that are negations of each other

R_{16}: $P_{1,1} \lor P_{3,1}$ (*resolvent*)

Resolvent

R_{13}: $\neg P_{2,2}$ R_{15}: $P_{1,1} \vee P_{2,2} \vee P_{3,1}$

resolving the two literals that are negations of each other

R_{16}: $P_{1,1} \vee P_{3,1}$ (*resolvent*)

If there's a pit in one of [1,1], [2,2], and [3,1] and it's not in [2,2], then it's in [1,1] or [3,1].

Resolvent

R_{13}: $\neg P_{2,2}$

R_{15}: $P_{1,1} \vee P_{2,2} \vee P_{3,1}$

resolving the two literals that are negations of each other

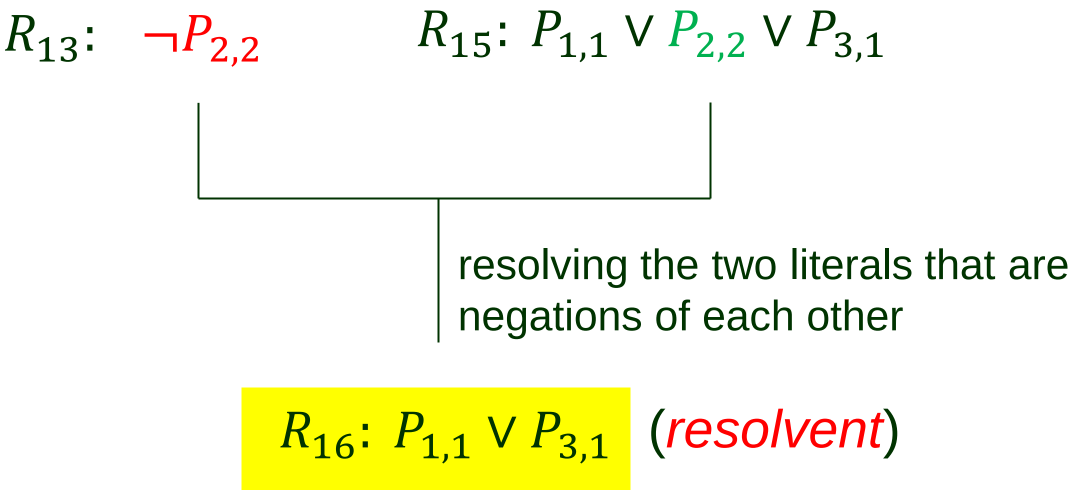
R_{16}: $P_{1,1} \vee P_{3,1}$ (*resolvent*)

If there's a pit in one of [1,1], [2,2], and [3,1] and it's not in [2,2], then it's in [1,1] or [3,1].

R_1: $\neg P_{1,1}$

R_{16}: $P_{1,1} \vee P_{3,1}$

Resolvent



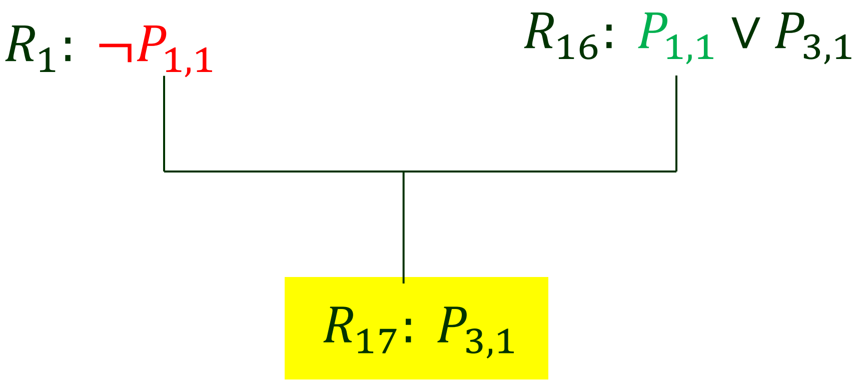
R_{13}: $\neg P_{2,2}$

R_{15}: $P_{1,1} \lor P_{2,2} \lor P_{3,1}$

resolving the two literals that are negations of each other

R_{16}: $P_{1,1} \lor P_{3,1}$ (*resolvent*)

If there’s a pit in one of [1,1], [2,2], and [3,1] and it’s not in [2,2], then it’s in [1,1] or [3,1].



R_1: $\neg P_{1,1}$

R_{16}: $P_{1,1} \lor P_{3,1}$

R_{17}: $P_{3,1}$

Simple Resolution Rule

$$\frac{l_1 \vee \cdots \vee l_i \vee \cdots \vee l_k, \qquad m}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k}$$

(l_i and m are complementary literals, i.e.,$l_i = \neg m$ or $m = \neg l_i$.)

Simple Resolution Rule

$$\frac{l_1 \vee \cdots \vee l_i \vee \cdots \vee l_k, \qquad m}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k}$$

(l_i and m are complementary literals, i.e.,$l_i = \neg m$ or $m = \neg l_i$.)

Since m is true, then l_i must be false. But one of $l_1, \dots, l_k$ must be true. Therefore, we can exclude l_i and assert that one of the remaining $k - 1$ literals must be true.

Simple Resolution Rule

$$\frac{l_1 \vee \cdots \vee l_i \vee \cdots \vee l_k, \qquad m}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k}$$

(l_i and m are complementary literals, i.e.,$l_i = \neg m$ or $m = \neg l_i$.)

Since m is true, then l_i must be false. But one of $l_1, \dots, l_k$ must be true. Therefore, we can exclude l_i and assert that one of the remaining $k - 1$ literals must be true.

Clause: a disjunction of literals.

$$R_{15}\colon\ P_{1,1} \vee P_{2,2} \vee P_{3,1}$$

Simple Resolution Rule

$$\frac{l_1 \vee \cdots \vee l_i \vee \cdots \vee l_k, \qquad m}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k}$$

(l_i and m are complementary literals, i.e.,$l_i = \neg m$ or $m = \neg l_i$.)

Since m is true, then l_i must be false. But one of $l_1, \ldots, l_k$ must be true. Therefore, we can exclude l_i and assert that one of the remaining $k - 1$ literals must be true.

Clause: a disjunction of literals.

$$R_{15}\colon\ P_{1,1} \vee P_{2,2} \vee P_{3,1}$$

Unit clause: a single literal.

Simple Resolution Rule

$$\frac{l_1 \vee \cdots \vee l_i \vee \cdots \vee l_k, \qquad m}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k}$$

(l_i and m are complementary literals, i.e.,$l_i = \neg m$ or $m = \neg l_i$.)

Since m is true, then l_i must be false. But one of $l_1, \ldots, l_k$ must be true. Therefore, we can exclude l_i and assert that one of the remaining $k - 1$ literals must be true.

Clause: a disjunction of literals.

$$R_{15}:\ P_{1,1} \vee P_{2,2} \vee P_{3,1}$$

Unit clause: a single literal.

$$R_1:\ \neg P_{2,2} \qquad\qquad R_5:\ B_{2,1}$$

Simple Resolution Rule

$$\frac{l_1 \lor \cdots \lor l_i \lor \cdots \lor l_k, \qquad m}{l_1 \lor \cdots \lor l_{i-1} \lor l_{i+1} \lor \cdots \lor l_k}$$

(l_i and m are complementary literals, i.e.,$l_i = \neg m$ or $m = \neg l_i$.)

Since m is true, then l_i must be false. But one of $l_1, \dots, l_k$ must be true. Therefore, we can exclude l_i and assert that one of the remaining $k - 1$ literals must be true.

Clause: a disjunction of literals.

$$\frac{P_{1,1} \lor P_{2,2} \lor P_{3,1}, \qquad \neg P_{2,2}}{P_{1,1} \lor P_{3,1}}$$

R_{15}: $P_{1,1} \lor P_{2,2} \lor P_{3,1}$

Unit clause: a single literal.

R_1: $\neg P_{2,2}$ $\qquad$ R_5: $B_{2,1}$

Full Resolution Rule

l_i and m_j are complementary literals:

$$\frac{l_1 \lor \cdots \lor l_i \lor \cdots \lor l_k, \qquad m_1 \lor \cdots \lor m_j \lor \cdots \lor m_n}{l_1 \lor \cdots \lor l_{i-1} \lor l_{i+1} \lor \cdots \lor l_k \lor m_1 \lor \cdots \lor m_{j-1} \lor m_{j+1} \lor \cdots \lor m_n}$$

Full Resolution Rule

l_i and m_j are complementary literals:

$$\frac{l_1 \vee \cdots \vee l_i \vee \cdots \vee l_k, \qquad m_1 \vee \cdots \vee m_j \vee \cdots \vee m_n}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

If l_i is true, then m_j is false. Hence $m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n$ must be true.
If l_i is false, then $l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k$ must be true.

Full Resolution Rule

l_i and m_j are complementary literals:

$$\frac{l_1 \vee \cdots \vee l_i \vee \cdots \vee l_k, \qquad m_1 \vee \cdots \vee m_j \vee \cdots \vee m_n}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

If l_i is true, then m_j is false. Hence $m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n$ must be true.
If l_i is false, then $l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k$ must be true.

$$\frac{P_{1,1} \vee P_{3,1}, \qquad \neg P_{1,1} \vee \neg P_{2,2}}{P_{3,1} \vee \neg P_{2,2}}$$

One Pair at a Time

Only one pair of complementary literals can be resolved at each step.

$P \lor \neg Q \lor R, \qquad \neg P \lor Q$

One Pair at a Time

Only one pair of complementary literals can be resolved at each step.

$$\frac{P \lor \neg Q \lor R, \qquad \neg P \lor Q}{\neg Q \lor R \lor Q}$$

One Pair at a Time

Only one pair of complementary literals can be resolved at each step.

$$\frac{P \lor \neg Q \lor R, \qquad \neg P \lor Q}{\neg Q \lor R \lor Q \quad \equiv \textit{true}}$$

One Pair at a Time

Only one pair of complementary literals can be resolved at each step.

$$\frac{P \lor \neg Q \lor R, \qquad \neg P \lor Q}{\neg Q \lor R \lor Q \quad \equiv \mathit{true}}$$

✓

One Pair at a Time

Only one pair of complementary literals can be resolved at each step.

$$\frac{P \lor \neg Q \lor R, \qquad \neg P \lor Q}{\neg Q \lor R \lor Q \quad \equiv \mathit{true}}$$ ✓

$$\frac{P \lor \neg Q \lor R, \qquad \neg P \lor Q}{R}$$

One Pair at a Time

Only one pair of complementary literals can be resolved at each step.

$$\frac{P \vee \neg Q \vee R, \qquad \neg P \vee Q}{\neg Q \vee R \vee Q \quad \equiv \textit{true}}$$

✓

$$\frac{P \vee \neg Q \vee R, \qquad \neg P \vee Q}{R}$$

Incorrect conclusion!

One Pair at a Time

Only one pair of complementary literals can be resolved at each step.

$$\frac{P \lor \neg Q \lor R, \qquad \neg P \lor Q}{\neg Q \lor R \lor Q \quad \equiv \mathit{true}}$$

✓

$$\frac{P \lor \neg Q \lor R, \qquad \neg P \lor Q}{R}$$

✗

Incorrect conclusion!

II. Conjunctive Normal Form

The resolution rule applies to clauses only.

Conjunctive normal form (CNF): a conjunction of clauses

II. Conjunctive Normal Form

The resolution rule applies to clauses only.

Conjunctive normal form (CNF): a conjunction of clauses

$$
\begin{array}{rcl}
CNFSentence & \rightarrow & Clause_1 \land \cdots \land Clause_n \\
Clause & \rightarrow & Literal_1 \lor \cdots \lor Literal_m \\
Fact & \rightarrow & Symbol \\
Literal & \rightarrow & Symbol \mid \neg Symbol \\
Symbol & \rightarrow & P \mid Q \mid R \mid \ldots
\end{array}
$$

II. Conjunctive Normal Form

The resolution rule applies to clauses only.

Conjunctive normal form (CNF): a conjunction of clauses

$$\begin{array}{rcl}
\mathit{CNFSentence} & \rightarrow & \mathit{Clause}_1 \land \cdots \land \mathit{Clause}_n \\
\mathit{Clause} & \rightarrow & \mathit{Literal}_1 \lor \cdots \lor \mathit{Literal}_m \\
\mathit{Fact} & \rightarrow & \mathit{Symbol} \\
\mathit{Literal} & \rightarrow & \mathit{Symbol} \mid \neg \mathit{Symbol} \\
\mathit{Symbol} & \rightarrow & P \mid Q \mid R \mid \ldots
\end{array}$$

Every sentence of propositional logic is equivalent to a CNF.

Converting to CNF

1. Eliminate ⇔.

$$\alpha \Leftrightarrow \beta$$

Converting to CNF

1. Eliminate ⇔.

$$\alpha \Leftrightarrow \beta$$

replaced with

$$(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)$$

Converting to CNF

1. Eliminate ⇔.

$$\alpha \Leftrightarrow \beta$$

replaced with

$$(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$$

$$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$$

$$(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Longrightarrow B_{1,1})$$

Converting to CNF

1. Eliminate ⇔.

$$\begin{tikzcd} \alpha \Leftrightarrow \beta \arrow[d, dashed, "\text{replaced with}"'] \\ (\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha) \end{tikzcd}$$

$$\begin{tikzcd} B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1}) \arrow[d, dashed] \\ (B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \Longrightarrow B_{1,1}) \end{tikzcd}$$

2. Eliminate ⟹.

Converting to CNF

1. Eliminate ⇔.

$$\alpha \Leftrightarrow \beta$$

replaced with $\downarrow$

$$(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)$$

$$B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})$$

$$\downarrow$$

$$(B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \Longrightarrow B_{1,1})$$

2. Eliminate ⟹.

$$\alpha \Rightarrow \beta$$

$$\downarrow$$

$$\neg\alpha \lor \beta$$

Converting to CNF

1. Eliminate ⇔.

$$\begin{tikzcd}\alpha \Leftrightarrow \beta \arrow[d, dashed, "\text{replaced with}"'] \\ (\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)\end{tikzcd}$$

2. Eliminate ⟹.

$$\begin{tikzcd}\alpha \Rightarrow \beta \arrow[d, dashed] \\ \neg\alpha \lor \beta\end{tikzcd}$$

$$\begin{tikzcd}B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1}) \arrow[d, dashed] \\ (B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \Longrightarrow B_{1,1}) \arrow[d, dashed] \\ (\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg(P_{1,2} \lor P_{2,1}) \lor B_{1,1})\end{tikzcd}$$

Converting to CNF

1. Eliminate ⇔.

$$\alpha \Leftrightarrow \beta$$

replaced with

$$(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)$$

2. Eliminate ⟹.

$$\alpha \Rightarrow \beta$$

$$\neg\alpha \lor \beta$$

$$B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})$$

$$(B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \Longrightarrow B_{1,1})$$

$$(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg(P_{1,2} \lor P_{2,1}) \lor B_{1,1})$$

3. Move ¬ inwards, repeatedly applying

$$\neg(\neg\alpha) \equiv \alpha$$
$$\neg(\alpha \lor \beta) \equiv (\neg\alpha \land \neg\beta)$$
$$\neg(\alpha \land \beta) \equiv (\neg\alpha \lor \neg\beta)$$

Converting to CNF

1. Eliminate ⇔.

$$\alpha \Leftrightarrow \beta$$

replacedwith

$$\downarrow$$

$$(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)$$

2. Eliminate ⟹.

$$\alpha \Rightarrow \beta$$

$$\downarrow$$

$$\neg\alpha \lor \beta$$

3. Move ¬ inwards, repeatedly applying

$$\neg(\neg\alpha) \equiv \alpha$$
$$\neg(\alpha \lor \beta) \equiv (\neg\alpha \land \neg\beta)$$
$$\neg(\alpha \land \beta) \equiv (\neg\alpha \lor \neg\beta)$$

$$B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})$$

$$\downarrow$$

$$(B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \Longrightarrow B_{1,1})$$

$$\downarrow$$

$$(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg(P_{1,2} \lor P_{2,1}) \lor B_{1,1})$$

$$\downarrow$$

$$(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land ((\neg P_{1,2} \land \neg P_{2,1}) \lor B_{1,1})$$

Converting to CNF

1. Eliminate ⇔.

$$\alpha \Leftrightarrow \beta$$

replaced with $\downarrow$

$$(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)$$

2. Eliminate ⟹.

$$\alpha \Rightarrow \beta$$

$$\downarrow$$

$$\neg\alpha \lor \beta$$

3. Move ¬ inwards, repeatedly applying

$$\neg(\neg\alpha) \equiv \alpha$$
$$\neg(\alpha \lor \beta) \equiv (\neg\alpha \land \neg\beta)$$
$$\neg(\alpha \land \beta) \equiv (\neg\alpha \lor \neg\beta)$$

4. Apply the distributivity law

$$B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})$$

$$\downarrow$$

$$(B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \Longrightarrow B_{1,1})$$

$$\downarrow$$

$$(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land (\neg(P_{1,2} \lor P_{2,1}) \lor B_{1,1})$$

$$\downarrow$$

$$(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}) \land ((\neg P_{1,2} \land \neg P_{2,1}) \lor B_{1,1})$$

Converting to CNF

1. Eliminate ⇔.

$$\begin{tikzcd} \alpha \Leftrightarrow \beta \arrow[d, dashed, "\text{replaced with}"'] \\ (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha) \end{tikzcd}$$

2. Eliminate ⟹.

$$\begin{tikzcd} \alpha \Rightarrow \beta \arrow[d, dashed] \\ \neg\alpha \vee \beta \end{tikzcd}$$

3. Move ¬ inwards, repeatedly applying

$$\neg(\neg\alpha) \equiv \alpha$$
$$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$$
$$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$$

4. Apply the distributivity law

$$\begin{tikzcd} B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1}) \arrow[d, dashed] \\ (B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Longrightarrow B_{1,1}) \arrow[d, dashed] \\ (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1}) \arrow[d, dashed] \\ (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1}) \arrow[d, dashed] \\ (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1}) \end{tikzcd}$$

CNF Conversion Algorithm (Optional)

- Parse every propositional sentence in the KB as an arithmetic expression to construct an expression tree (Com S 228).

  Operators (connectives): ¬, ∧, ∨, ⇒, ⇔

  Operands (atomic sentences): $P, Q, R, S, T,$ ...

CNF Conversion Algorithm (Optional)

- Parse every propositional sentence in the KB as an arithmetic expression to construct an expression tree (Com S 228).

Operators (connectives): ¬, ∧, ∨, ⇒, ⇔

Operands (atomic sentences): $P, Q, R, S, T, \ldots$

$$\neg(P \wedge \neg Q) \vee R \Rightarrow S \wedge \neg T$$

(infix expression)

CNF Conversion Algorithm (Optional)

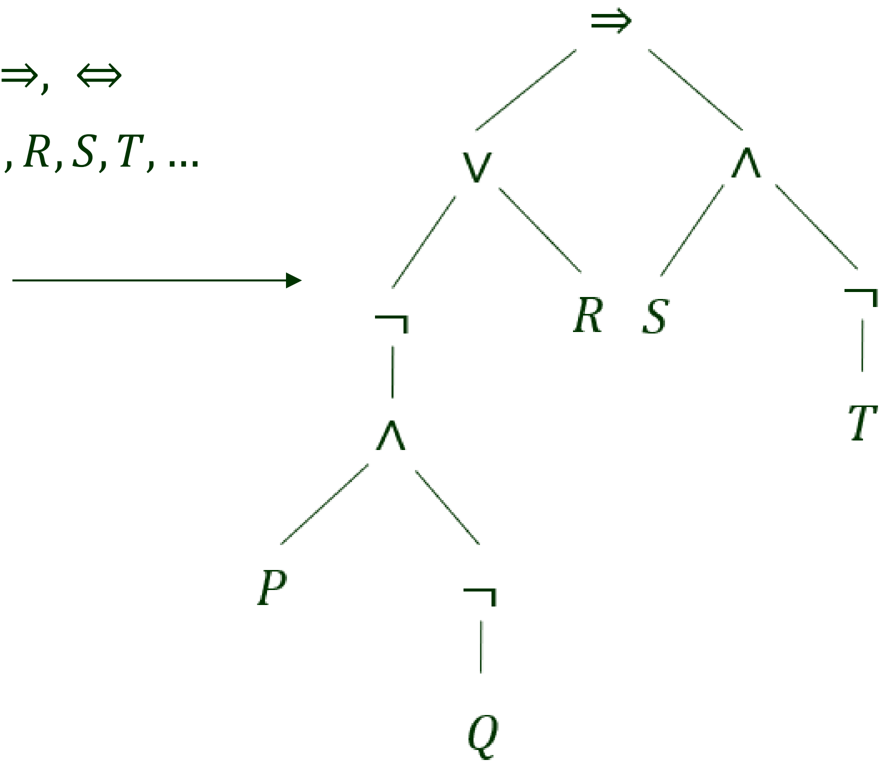
- Parse every propositional sentence in the KB as an arithmetic expression to construct an expression tree (Com S 228).

Operators (connectives): ¬, ∧, ∨, ⇒, ⇔

Operands (atomic sentences): $P, Q, R, S, T, \ldots$

$$\neg(P \wedge \neg Q) \vee R \Rightarrow S \wedge \neg T$$

(infix expression)



CNF Conversion Algorithm (Optional)

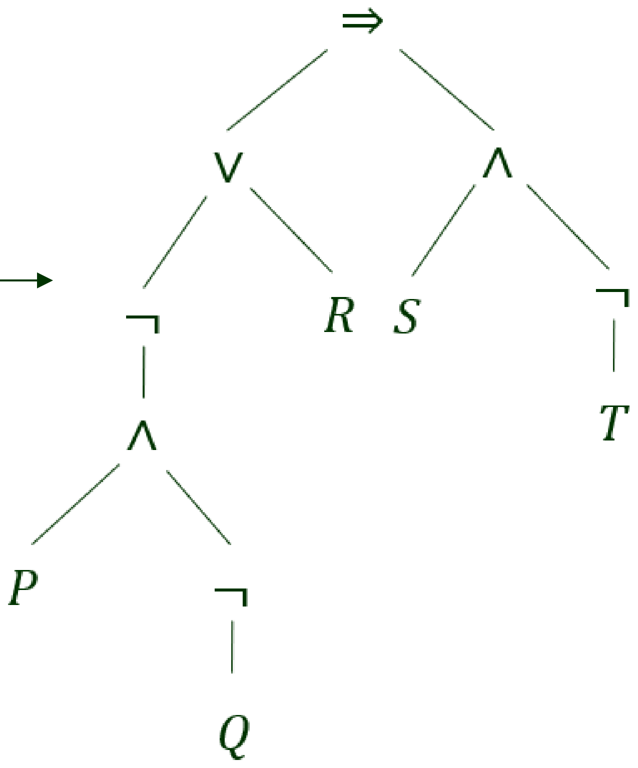
- ♦ Parse every propositional sentence in the KB as an arithmetic expression to construct an expression tree (Com S 228).

Operators (connectives): ¬, ∧, ∨, ⇒, ⇔

Operands (atomic sentences): $P, Q, R, S, T,$ ...

$$\neg(P \wedge \neg Q) \vee R \Rightarrow S \wedge \neg T \longrightarrow$$

(infix expression)



- Construction is similar to the algorithm employing a stack to convert an infix expression into a postfix expression (Com S 228).

CNF Conversion Algorithm (Optional)

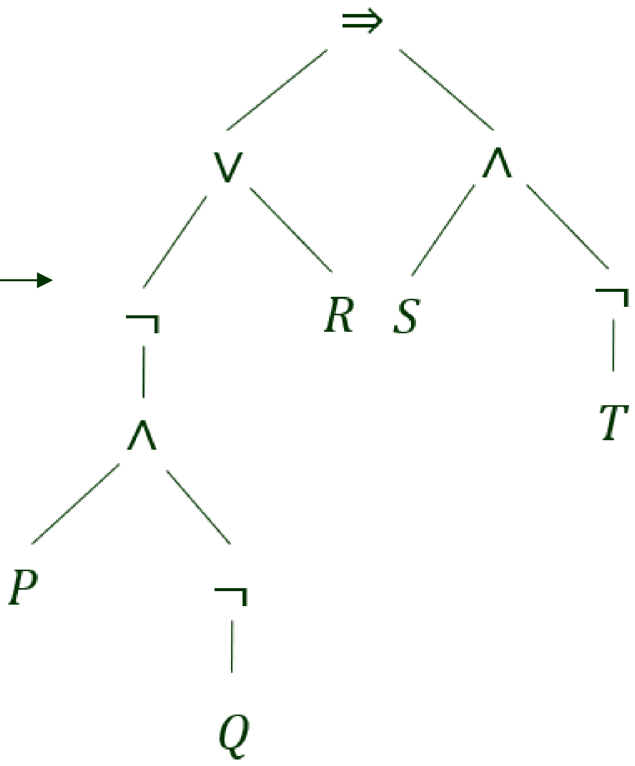
- Parse every propositional sentence in the KB as an arithmetic expression to construct an expression tree (Com S 228).

Operators (connectives): ¬, ∧, ∨, ⇒, ⇔

Operands (atomic sentences): $P, Q, R, S, T, \ldots$

$$\neg(P \land \neg Q) \lor R \Rightarrow S \land \neg T \longrightarrow$$

(infix expression)



  - Construction is similar to the algorithm employing a stack to convert an infix expression into a postfix expression (Com S 228).

$$PQ\neg \land \neg R \lor ST\neg \land \Rightarrow$$

(postfix expression)

CNF Conversion Algorithm (Optional)

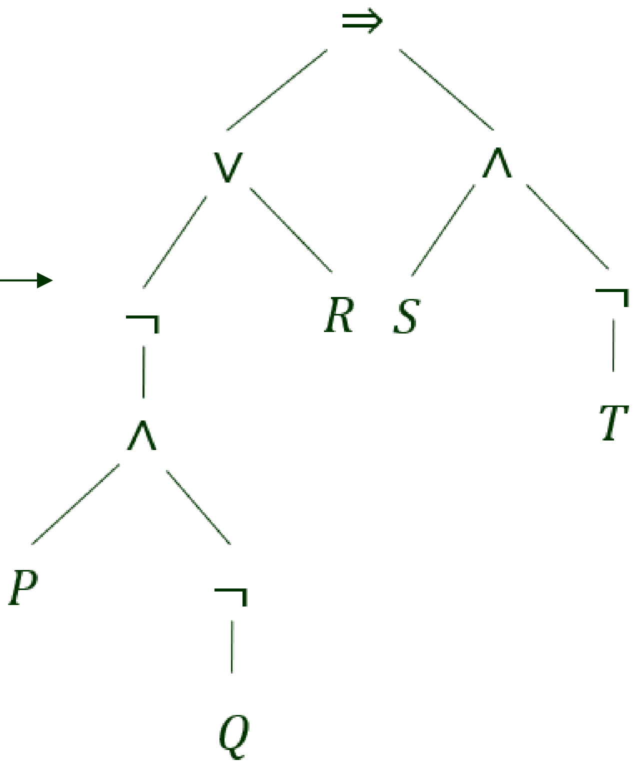
- Parse every propositional sentence in the KB as an arithmetic expression to construct an expression tree (Com S 228).

Operators (connectives): $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$

Operands (atomic sentences): $P, Q, R, S, T, \ldots$

$$\neg(P \wedge \neg Q) \vee R \Rightarrow S \wedge \neg T$$

(infix expression)



- Construction is similar to the algorithm employing a stack to convert an infix expression into a postfix expression (Com S 228).

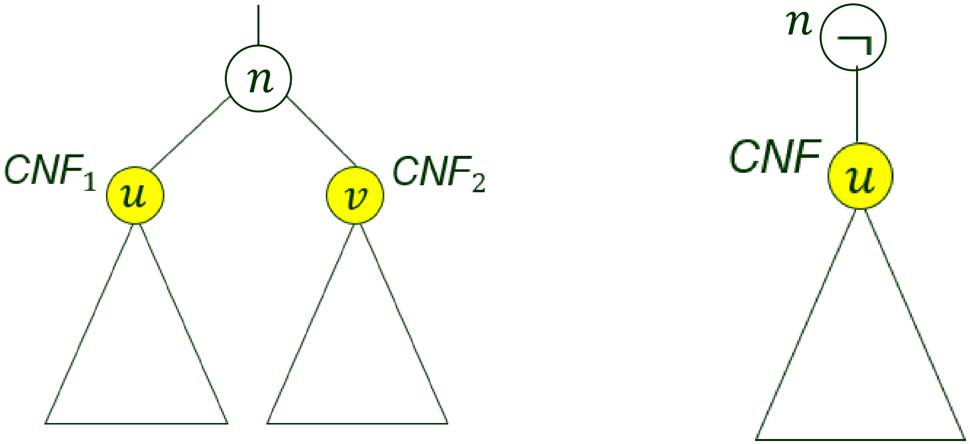
$$PQ\neg \wedge \neg R \vee ST\neg \wedge\Rightarrow$$

(postfix expression)

- Instead of outputting an operator after its two operands in the postfix format, now you just make the logical operator the parent of the two roots of the subtrees that store the same operator's subexpression operands.

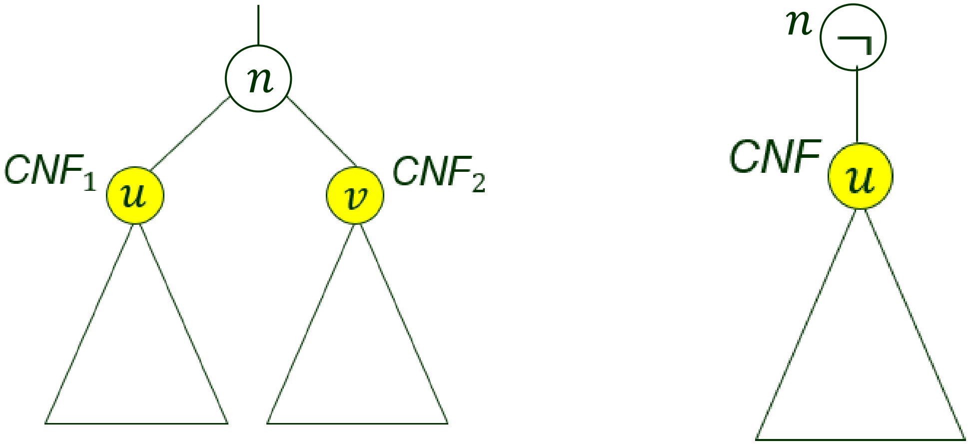
Postorder Traversal

- Perform a postorder traversal of the expression tree.
  - When visiting an internal node n (representing a connective), its left and right children (or its unique child in the case of a ¬ node) store the CNFs for the expressions represented by the left and right subtrees.



Postorder Traversal

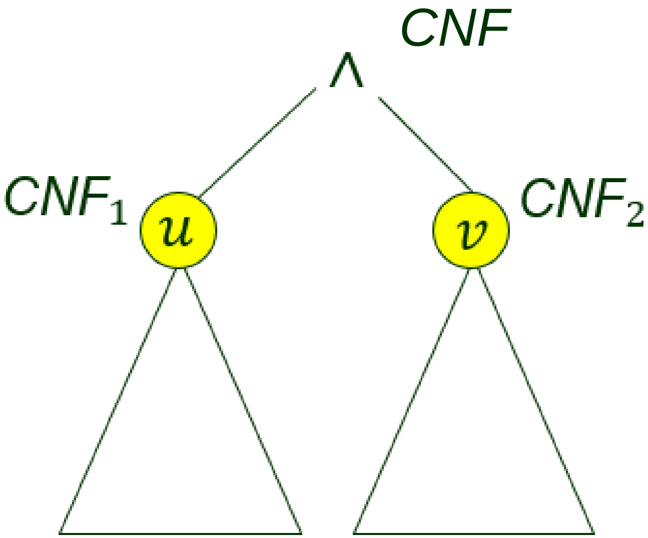
- Perform a postorder traversal of the expression tree.
  - When visiting an internal node n (representing a connective), its left and right children (or its unique child in the case of a ¬ node) store the CNFs for the expressions represented by the left and right subtrees.



  - Conversion is done in five cases depending on the logical operator stored at n.

Case 1

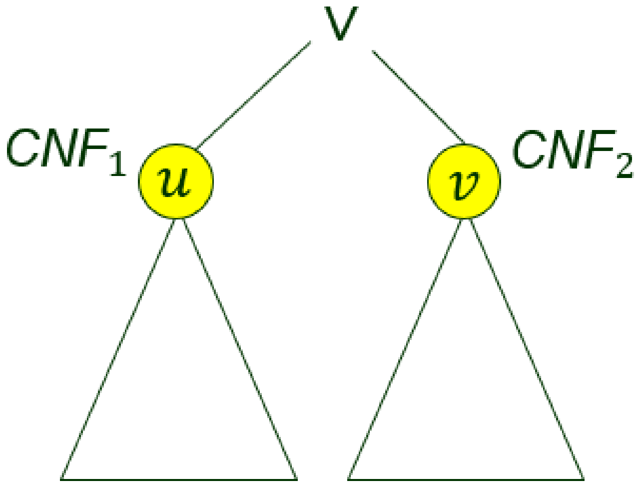
The node n stores ∧.



$$CNF \equiv CNF_1 \land CNF_2$$

Case 2

The node n stores V.



$$CNF_1 \equiv \; C_1 \wedge \cdots \wedge C_k$$

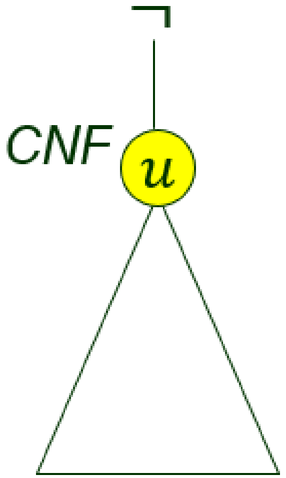
$$CNF_2 \equiv \; C_1' \wedge \cdots \wedge C_m'$$

$$CNF \equiv CNF_1 \vee CNF_2 \equiv \quad \bigwedge_{\substack{i=1,\ldots,k \\ j=1,\ldots,m}} \left(C_i \vee C_j'\right)$$

Case 3

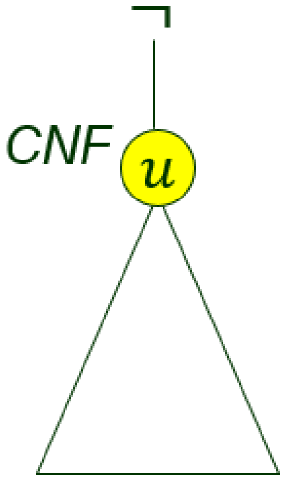
The node n stores $\neg$.

$$CNF \equiv (l_{11} \vee \cdots \vee l_{1k_1}) \wedge (l_{21} \vee \cdots \vee l_{2k_2}) \wedge \cdots \wedge (l_{r1} \vee \cdots \vee l_{rk_r})$$



Case 3

The node n stores ¬.



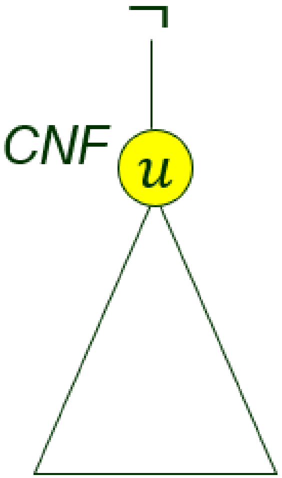
$$CNF \equiv \left(l_{11} \vee \cdots \vee l_{1k_1}\right) \wedge \left(l_{21} \vee \cdots \vee l_{2k_2}\right) \wedge \cdots \wedge \left(l_{r1} \vee \cdots \vee l_{rk_r}\right)$$

$$\Downarrow$$

$$\neg CNF \equiv \quad \bigwedge_{\substack{1\le j_1 \le k_1 \\ \vdots \\ 1 \le j_r \le k_r}} \left(\neg l_{1j_1} \vee \neg l_{2j_2} \vee \cdots \vee \neg l_{rj_r}\right)$$

Case 3

The node n stores $\neg$.



$$CNF \equiv \left(l_{11} \vee \cdots \vee l_{1k_1}\right) \wedge \left(l_{21} \vee \cdots \vee l_{2k_2}\right) \wedge \cdots \wedge \left(l_{r1} \vee \cdots \vee l_{rk_r}\right)$$

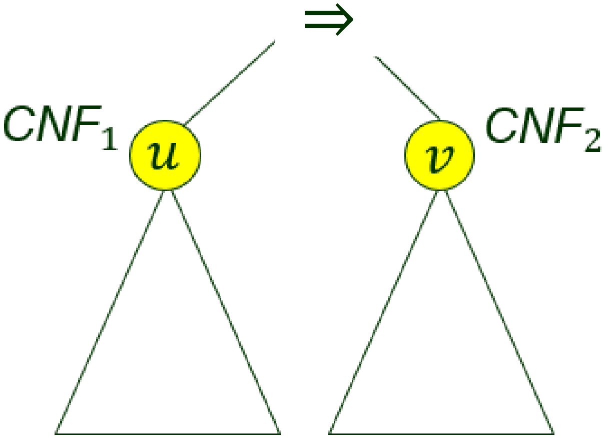
$$\Downarrow$$

$$\neg CNF \equiv \quad \bigwedge_{\substack{1\le j_1\le k_1 \\ \vdots \\ 1\le j_r\le k_r}} \left(\neg l_{1j_1} \vee \neg l_{2j_2} \vee \cdots \vee \neg l_{rj_r}\right)$$

- If l_{ij} is a negative literal, i.e., $l_{ij} = \neg p_{ij}$, then $\neg l_{ij}$ reduces to p_{ij} because the two occurrences of $\neg$ cancel out.

Case 4

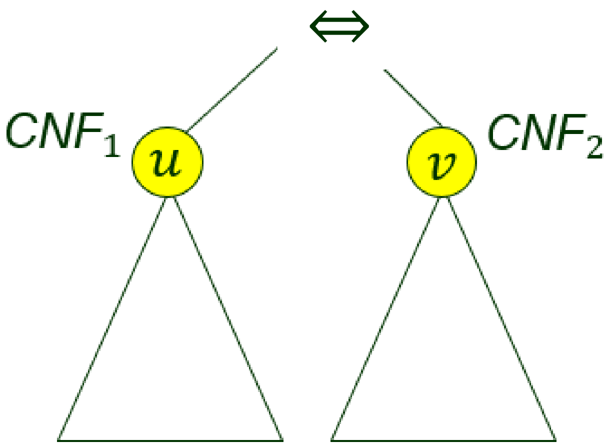
The node n stores ⇒.



- Logically equivalent to $\neg CNF_1 \vee CNF_2$.
- First, convert $\neg CNF_1$ into conjunctive normal form CNF_1' (see case 3).
- Then, convert the disjunction $CNF_1' \vee CNF_2$ into conjunctive normal form (see case 2)

Case 5

The node n stores $\Leftrightarrow$.



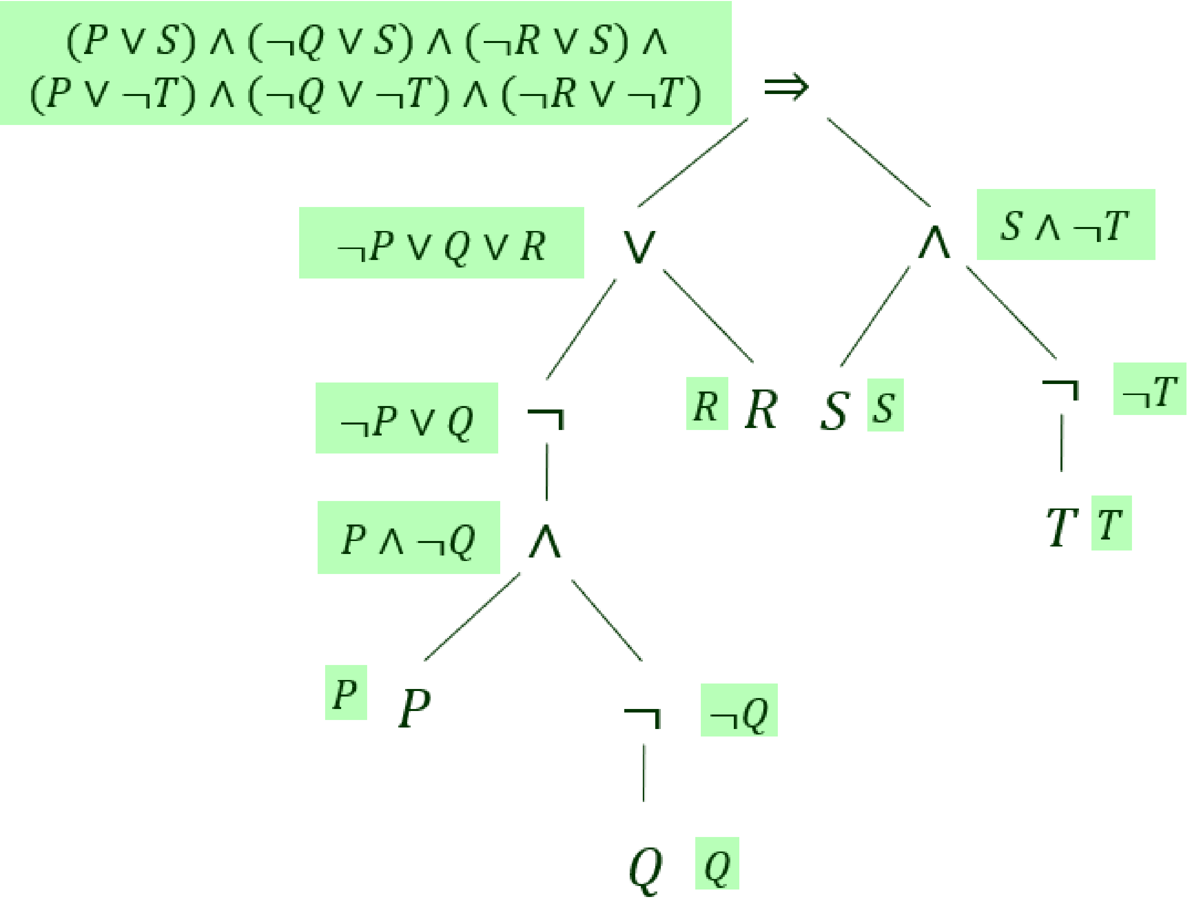
- Logically equivalent to $(CNF_1 \Rightarrow CNF_2) \wedge (CNF_2 \Rightarrow CNF_1)$.

- Convert $CNF_1 \Rightarrow CNF_2$ and $CNF_2 \Rightarrow CNF_1$ separately into conjunctive normal forms CNF_1' and CNF_2' (see case 4).

- Return $CNF_1' \wedge CNF_2'$.

Example

$\neg(P \wedge \neg Q) \vee R \Rightarrow S \wedge \neg T$ has the following conjunctive normal form:



$$(P \vee S) \wedge (\neg Q \vee S) \wedge (\neg R \vee S) \wedge$$
$$(P \vee \neg T) \wedge (\neg Q \vee \neg T) \wedge (\neg R \vee \neg T)$$

$\Rightarrow$

$\neg P \vee Q \vee R$ $\quad \vee$

$\wedge$ $\quad S \wedge \neg T$

$\neg P \vee Q$ $\quad \neg$

R $\quad R$ $\quad S$ $\quad S$

$\neg$ $\quad \neg T$

$P \wedge \neg Q$ $\quad \wedge$

T $\quad T$

P $\quad P$

$\neg$ $\quad \neg Q$

Q $\quad Q$

III. Proof by Resolution – An Example

KB:

P

$P \rightarrow (Q \vee R)$

$Q \rightarrow S$

$R \rightarrow (S \wedge T)$

III. Proof by Resolution – An Example

KB:

P

$P \rightarrow (Q \vee R)$

$Q \rightarrow S$

$R \rightarrow (S \wedge T)$

Q: $KB \vdash S$?

III. Proof by Resolution – An Example

KB:

$$P$$
$$P \rightarrow (Q \vee R)$$
$$Q \rightarrow S$$
$$R \rightarrow (S \wedge T)$$

Q: *KB* $\vdash S$?

1. Converting sentences to CNF

III. Proof by Resolution – An Example

KB:

P

$P \rightarrow (Q \vee R) \dashrightarrow \neg P \vee Q \vee R$

$Q \rightarrow S$

$R \rightarrow (S \wedge T)$

Q: $KB \vdash S$?

1. Converting sentences to CNF

III. Proof by Resolution – An Example

KB:

P

$P \rightarrow (Q \vee R)$ - - - - - -▸ $\neg P \vee Q \vee R$

$Q \rightarrow S$ - - - - - -▸ $\neg Q \vee S$

$R \rightarrow (S \wedge T)$

Q: $KB \vdash S$?

1. Converting sentences to CNF

III. Proof by Resolution – An Example

KB:

P

$P \rightarrow (Q \vee R) \quad \dashrightarrow \quad \neg P \vee Q \vee R$

$Q \rightarrow S \quad \dashrightarrow \quad \neg Q \vee S$

$R \rightarrow (S \wedge T) \quad \dashrightarrow \quad \neg R \vee (S \wedge T)$

Q: *KB* $\vdash S$?

1. Converting sentences to CNF

III. Proof by Resolution – An Example

KB:

P

$P \rightarrow (Q \vee R) \dashrightarrow \neg P \vee Q \vee R$

$Q \rightarrow S \dashrightarrow \neg Q \vee S$

$R \rightarrow (S \wedge T) \dashrightarrow \neg R \vee (S \wedge T)$

$\dashrightarrow (\neg R \vee S) \wedge (\neg R \vee T)$

Q: $KB \vdash S$?

1. Converting sentences to CNF

III. Proof by Resolution – An Example

KB:

P

$P \rightarrow (Q \vee R)$ ------→ $\neg P \vee Q \vee R$

$Q \rightarrow S$ ------→ $\neg Q \vee S$

$R \rightarrow (S \wedge T)$ ------→ $\neg R \vee (S \wedge T)$

------→ $(\neg R \vee S) \wedge (\neg R \vee T)$

Q: *KB* $\vdash S$?

1. Converting sentences to CNF

KB: P

$\neg P \vee Q \vee R$

$\neg Q \vee S$

$(\neg R \vee S) \wedge (\neg R \vee T)$

III. Proof by Resolution – An Example

KB:

P

$P \rightarrow (Q \vee R)$ ------→ $\neg P \vee Q \vee R$

$Q \rightarrow S$ ------→ $\neg Q \vee S$

$R \rightarrow (S \wedge T)$ ------→ $\neg R \vee (S \wedge T)$

------→ $(\neg R \vee S) \wedge (\neg R \vee T)$

Q: *KB* $\vdash S$?

1. Converting sentences to CNF
2. Spilt each conjunction into clauses.

KB: P

$\neg P \vee Q \vee R$

$\neg Q \vee S$

$(\neg R \vee S) \wedge (\neg R \vee T)$

III. Proof by Resolution – An Example

KB:

P

$P \to (Q \lor R) \dashrightarrow \neg P \lor Q \lor R$

$Q \to S \dashrightarrow \neg Q \lor S$

$R \to (S \land T) \dashrightarrow \neg R \lor (S \land T)$

$\dashrightarrow (\neg R \lor S) \land (\neg R \lor T)$

Q: *KB* ⊢ S ?

1. Converting sentences to CNF
2. Spilt each conjunction into clauses.

KB:

P

$\neg P \lor Q \lor R$

$\neg Q \lor S$

$$(\neg R \lor S) \land (\neg R \lor T) \longrightarrow \begin{cases} \neg R \lor S \\ \neg R \lor T \end{cases}$$

III. Proof by Resolution – An Example

KB:

P

$P \rightarrow (Q \lor R)$ - - - - - → $\neg P \lor Q \lor R$

$Q \rightarrow S$ - - - - - → $\neg Q \lor S$

$R \rightarrow (S \land T)$ - - - - - → $\neg R \lor (S \land T)$

- - - - - → $(\neg R \lor S) \land (\neg R \lor T)$

Q: *KB* $\vdash S$?

1. Converting sentences to CNF
2. Spilt each conjunction into clauses.

KB:

P

$\neg P \lor Q \lor R$

$\neg Q \lor S$

$(\neg R \lor S) \land (\neg R \lor T)$

KB:

P

$\neg P \lor Q \lor R$

$\neg Q \lor S$

$\neg R \lor S$

$\neg R \lor T$

Proof by Resolution

KB (updated):

(1) P

(2) $\neg P \lor Q \lor R$

(3) $\neg Q \lor S$

(4) $\neg R \lor S$

(5) $\neg R \lor T$

Q: *KB* $\vdash S$?

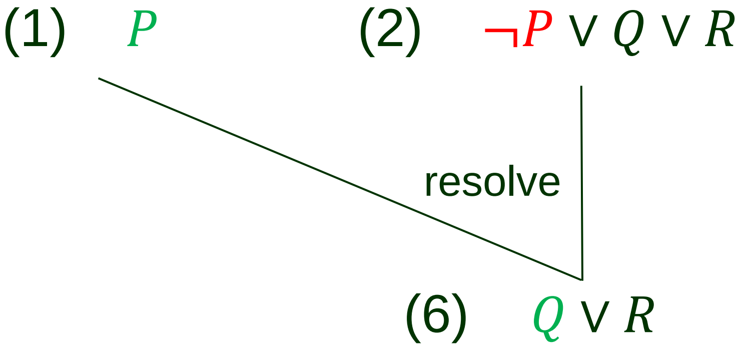
(1) P (2) $\neg P \lor Q \lor R$

Proof by Resolution

KB (updated):

(1) P
(2) $\neg P \vee Q \vee R$
(3) $\neg Q \vee S$
(4) $\neg R \vee S$
(5) $\neg R \vee T$

Q: $KB \vdash S$?



(1) P

(2) $\neg P \vee Q \vee R$

resolve

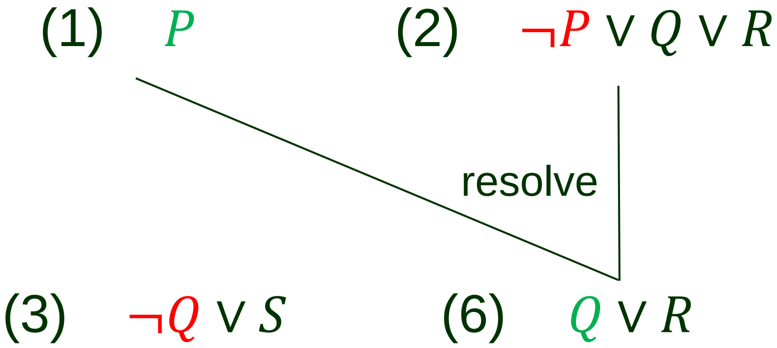
(6) $Q \vee R$

Proof by Resolution

KB (updated):

(1) P
(2) $\neg P \lor Q \lor R$
(3) $\neg Q \lor S$
(4) $\neg R \lor S$
(5) $\neg R \lor T$

Q: $KB \vdash S$?



(1) P

(2) $\neg P \lor Q \lor R$

resolve

(3) $\neg Q \lor S$

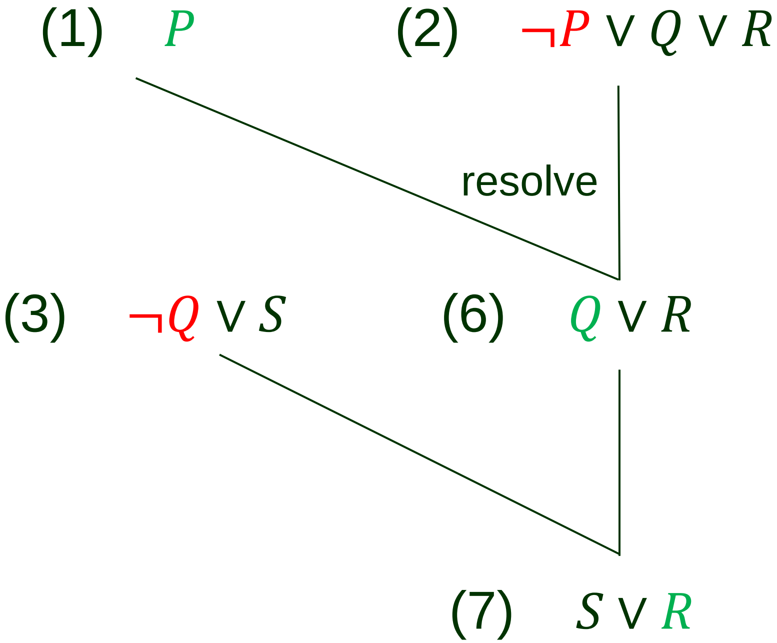
(6) $Q \lor R$

Proof by Resolution

KB (updated):

(1) P
(2) $\neg P \lor Q \lor R$
(3) $\neg Q \lor S$
(4) $\neg R \lor S$
(5) $\neg R \lor T$

Q: *KB* $\vdash S$?



(1) P
(2) $\neg P \lor Q \lor R$

resolve

(3) $\neg Q \lor S$
(6) $Q \lor R$

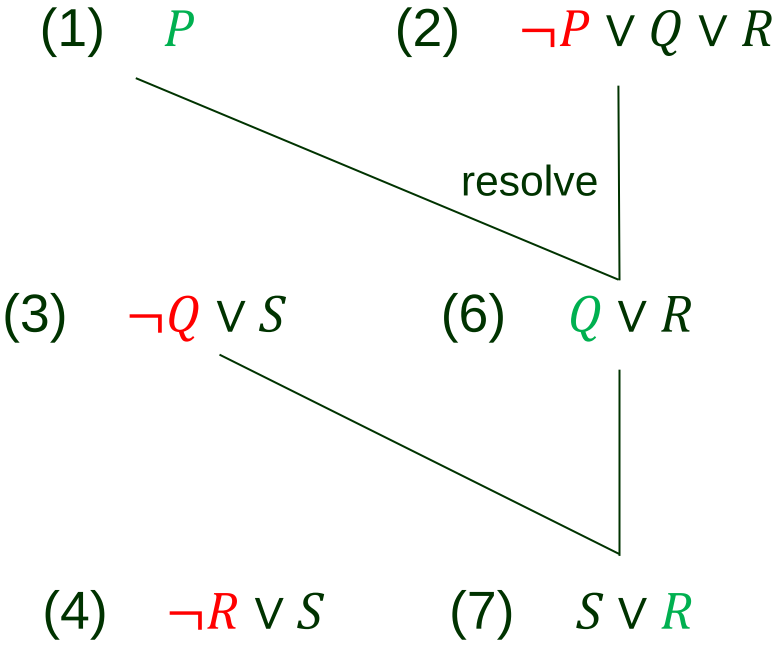
(7) $S \lor R$

Proof by Resolution

KB (updated):

(1) P
(2) $\neg P \lor Q \lor R$
(3) $\neg Q \lor S$
(4) $\neg R \lor S$
(5) $\neg R \lor T$

Q: *KB* $\vdash S$?



(1) P

(2) $\neg P \lor Q \lor R$

resolve

(3) $\neg Q \lor S$

(6) $Q \lor R$

(4) $\neg R \lor S$

(7) $S \lor R$

Proof by Resolution

KB (updated):

(1) P

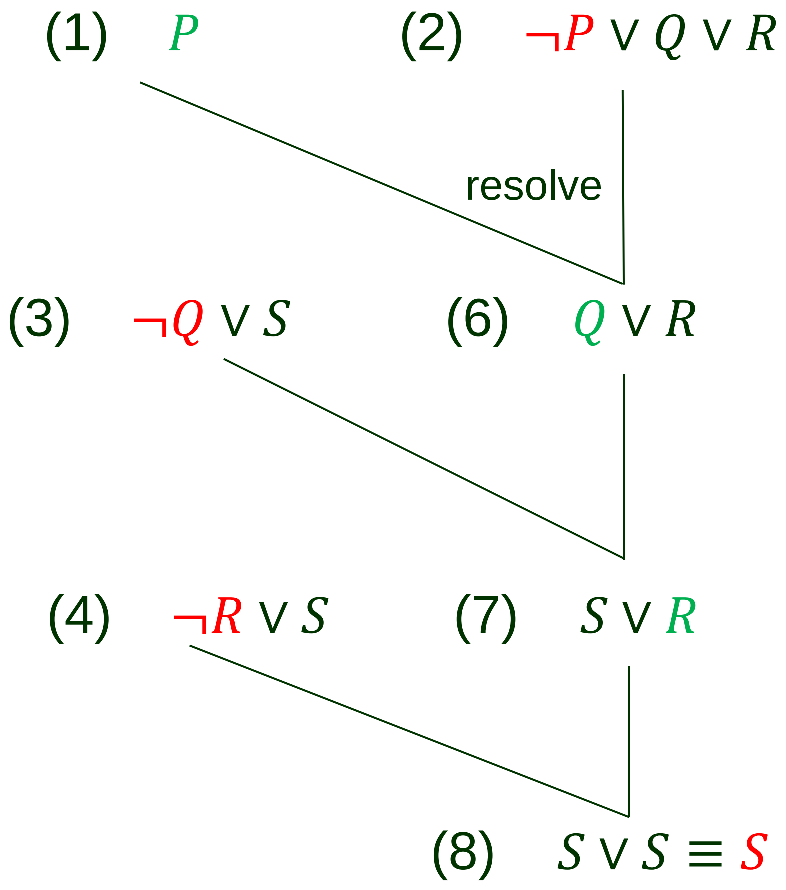
(2) $\neg P \lor Q \lor R$

(3) $\neg Q \lor S$

(4) $\neg R \lor S$

(5) $\neg R \lor T$

Q: *KB* $\vdash S$?



(1) P

(2) $\neg P \lor Q \lor R$

resolve

(3) $\neg Q \lor S$

(6) $Q \lor R$

(4) $\neg R \lor S$

(7) $S \lor R$

(8) $S \lor S \equiv S$

Proof by Resolution

KB (updated):

(1) P

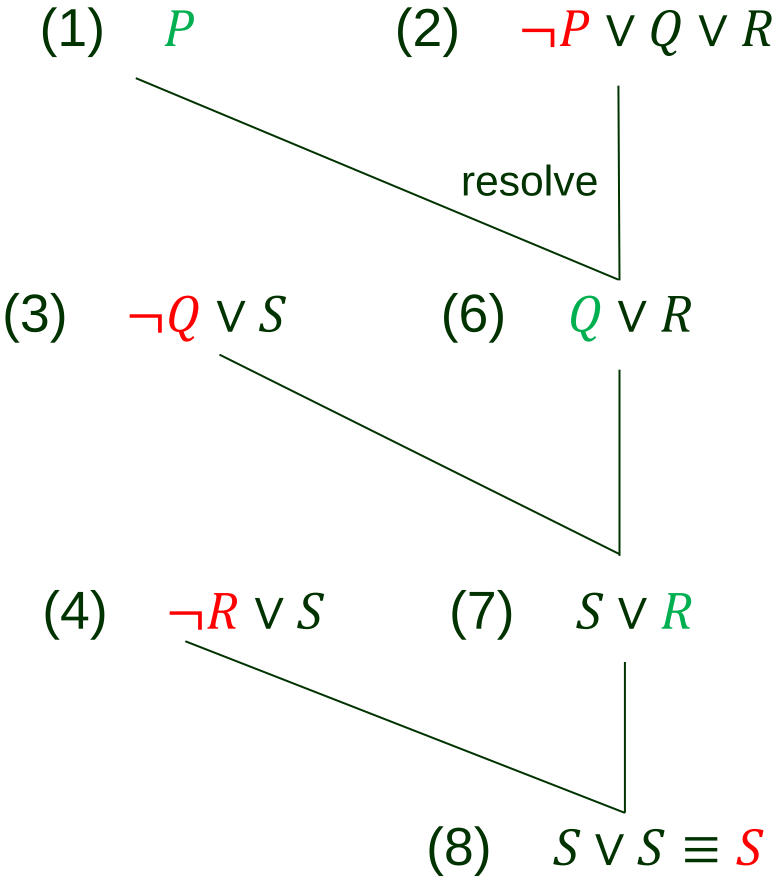
(2) $\neg P \lor Q \lor R$

(3) $\neg Q \lor S$

(4) $\neg R \lor S$

(5) $\neg R \lor T$

Q: $KB \vdash S$?



(1) P

(2) $\neg P \lor Q \lor R$

resolve

(3) $\neg Q \lor S$

(6) $Q \lor R$

(4) $\neg R \lor S$

(7) $S \lor R$

(8) $S \lor S \equiv S$

Resolution tree

Resolution Refutation

(Proof by contradiction)

To show that $KB \vDash \alpha$, we show that $KB \wedge \neg\alpha$ is unsatisfiable. .

Resolution Refutation

(Proof by contradiction)

To show that $KB \models \alpha$, we show that $KB \land \neg\alpha$ is unsatisfiable. .

KB (about a summer day):

```
(1) If it is raining and you are outside then you will get wet.
(2) If it is warm and there is no rain then it is a pleasant day.
(3) You are not wet.
(4) You are outside.
(5) It is a warm day.
```

* Example taken from http://watson.latech.edu/book/intelligence/intelligenceApproaches2b2.html

Resolution Refutation

(Proof by contradiction)

To show that $KB \vDash \alpha$, we show that $KB \land \neg\alpha$ is unsatisfiable. .

KB (about a summer day):

```
(1) If it is raining and you are outside then you will get wet.
(2) If it is warm and there is no rain then it is a pleasant day.
(3) You are not wet.
(4) You are outside.
(5) It is a warm day.
```

Prove

```
It is a pleasant day.
```

* Example taken from http://watson.latech.edu/book/intelligence/intelligenceApproaches2b2.html

KB in Propositional Sentences

KB (rewritten):

(1) (rain ∧ outside) ⇒ wet

(2) (warm ∧ ¬ rain) ⇒ pleasant

(3) ¬ wet

(4) outside

(5) warm

KB in Propositional Sentences

KB (rewritten):

(1) $(\mathrm{rain} \land \mathrm{outside}) \Rightarrow \mathrm{wet}$
(2) $(\mathrm{warm} \land \neg \mathrm{rain}) \Rightarrow \mathrm{pleasant}$
(3) $\neg \mathrm{wet}$
(4) $\mathrm{outside}$
(5) warm

$\Downarrow$ converted into clauses

(1) $\neg \mathrm{rain} \lor \neg \mathrm{outside} \lor \mathrm{wet}$
(2) $\neg \mathrm{warm} \lor \mathrm{rain} \lor \mathrm{pleasant}$
(3) $\neg \mathrm{wet}$
(4) $\mathrm{outside}$
(5) warm

KB in Propositional Sentences

KB (rewritten):

```
(1)  ( rain ∧ outside ) ⇒ wet
(2)  ( warm ∧ ¬rain ) ⇒ pleasant
(3)  ¬wet
(4)  outside
(5)  warm
```

⇓ converted into clauses

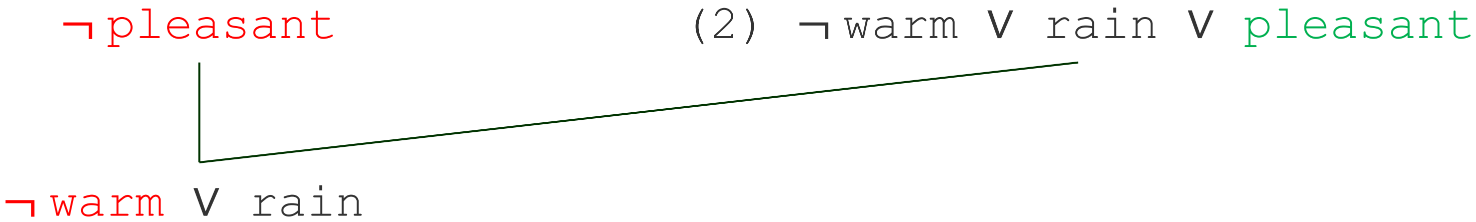
```
(1)  ¬rain ∨ ¬outside ∨ wet
(2)  ¬warm ∨ rain ∨ pleasant
(3)  ¬wet
(4)  outside
(5)  warm
```

We add ¬`pleasant` to KB and try to derive an empty clause Ø.

Resolution Refutation Tree

¬pleasant (2) ¬warm ∨ rain ∨ pleasant

Resolution Refutation Tree

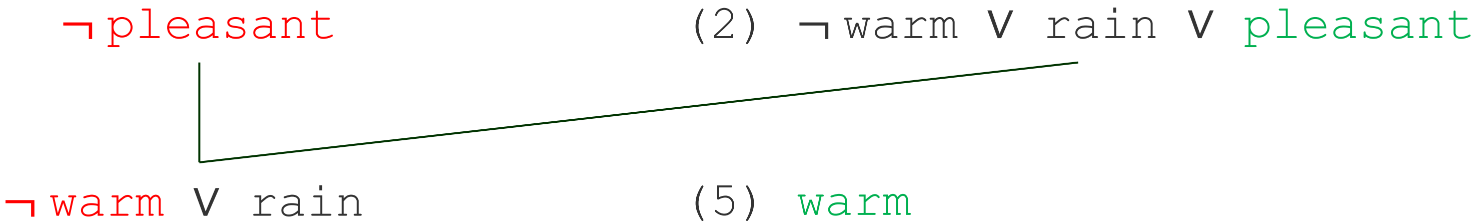


¬pleasant

(2) ¬warm ∨ rain ∨ pleasant

¬warm ∨ rain

Resolution Refutation Tree



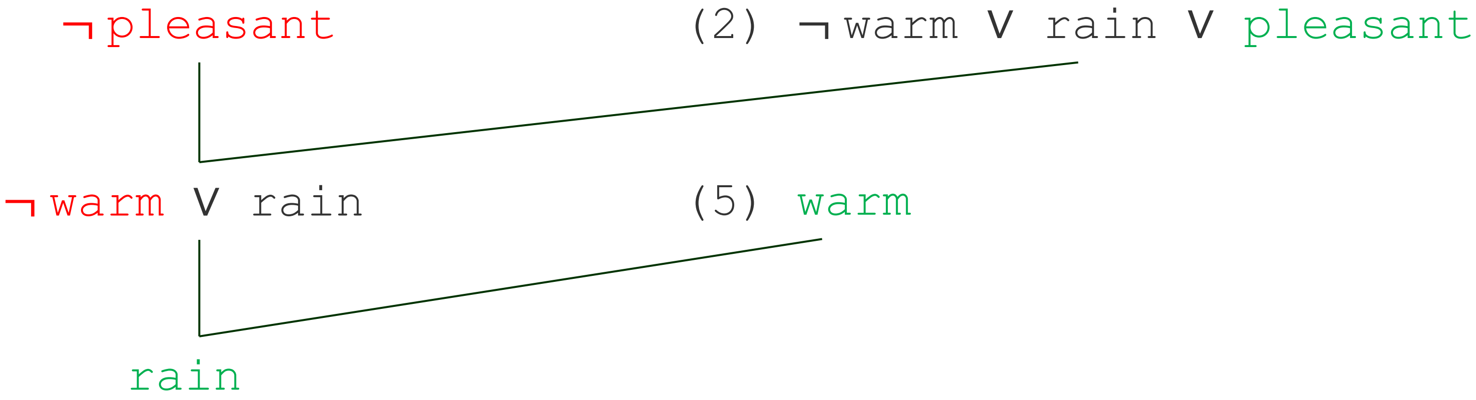
¬pleasant

(2) ¬warm ∨ rain ∨ pleasant

¬warm ∨ rain

(5) warm

Resolution Refutation Tree



¬pleasant

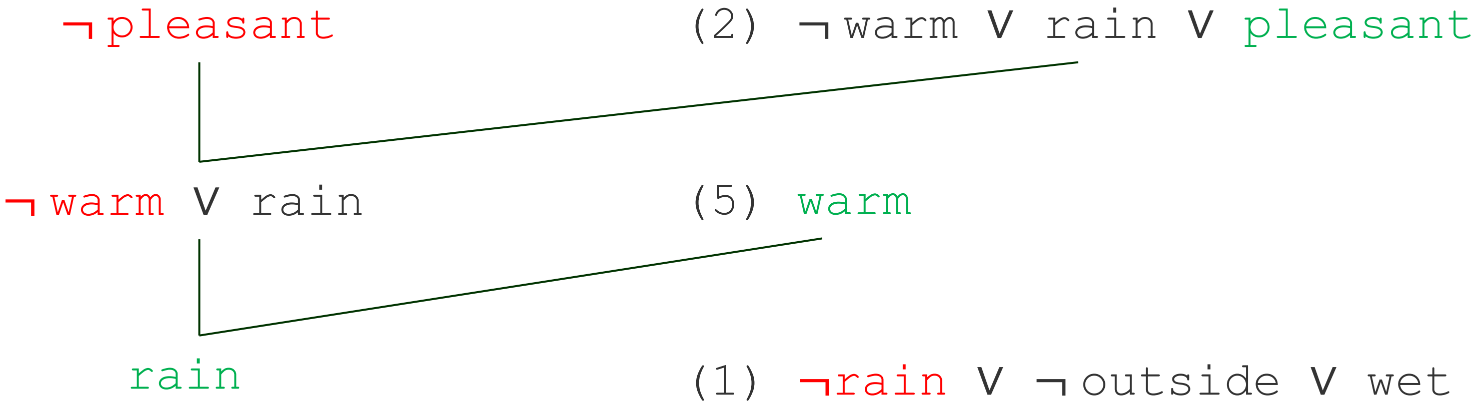
(2) ¬warm ∨ rain ∨ pleasant

¬warm ∨ rain

(5) warm

rain

Resolution Refutation Tree



¬pleasant

(2) ¬warm ∨ rain ∨ pleasant

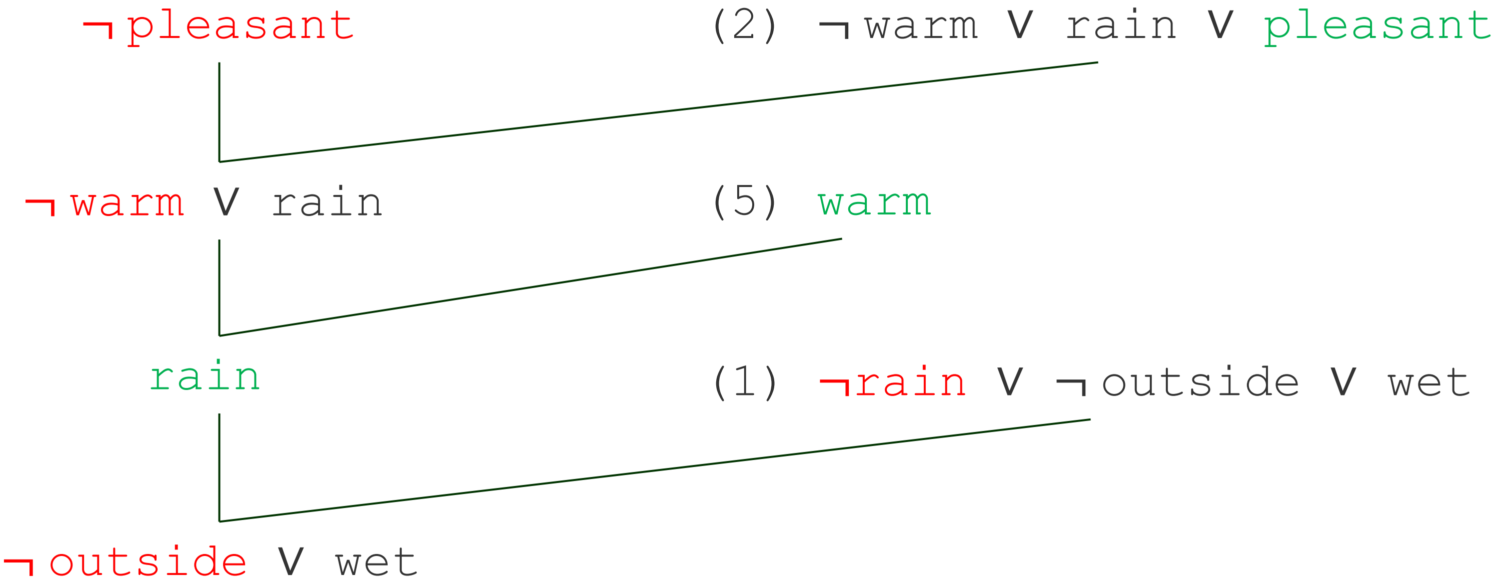
¬warm ∨ rain

(5) warm

rain

(1) ¬rain ∨ ¬outside ∨ wet

Resolution Refutation Tree



¬pleasant

(2) ¬warm ∨ rain ∨ pleasant

¬warm ∨ rain

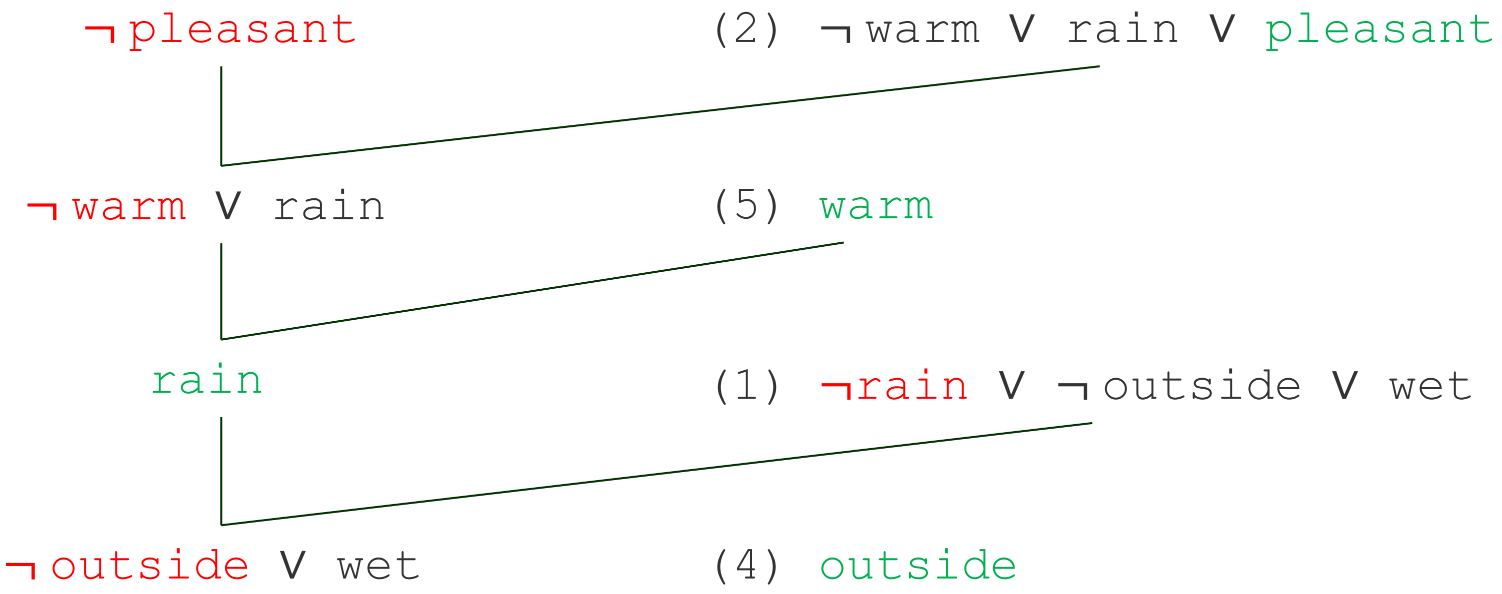
(5) warm

rain

(1) ¬rain ∨ ¬outside ∨ wet

¬outside ∨ wet

Resolution Refutation Tree



¬pleasant

(2) ¬warm ∨ rain ∨ pleasant

¬warm ∨ rain

(5) warm

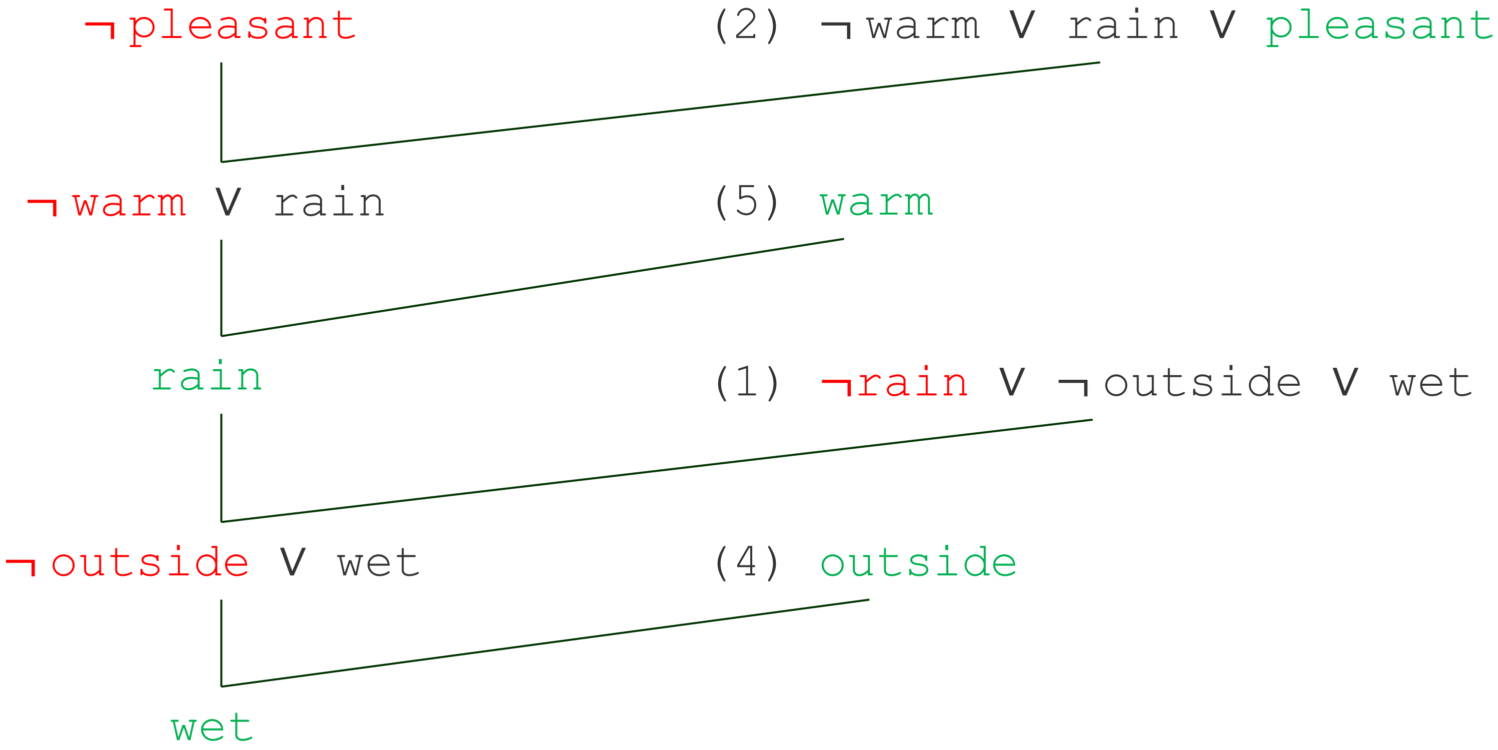
rain

(1) ¬rain ∨ ¬outside ∨ wet

¬outside ∨ wet

(4) outside

Resolution Refutation Tree



¬pleasant

(2) ¬warm V rain V pleasant

¬warm V rain

(5) warm

rain

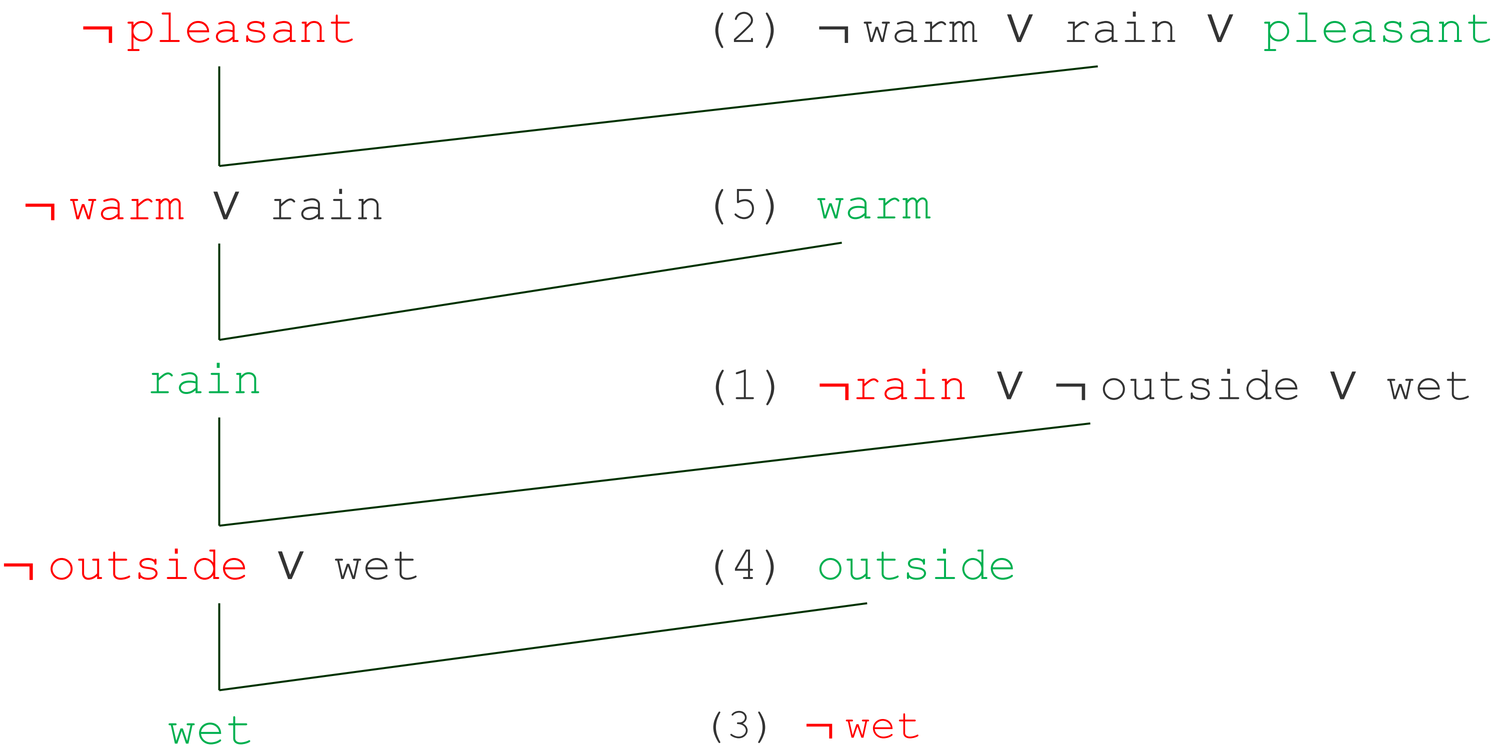
(1) ¬rain V ¬outside V wet

¬outside V wet

(4) outside

wet

Resolution Refutation Tree



¬pleasant (2) ¬warm ∨ rain ∨ pleasant

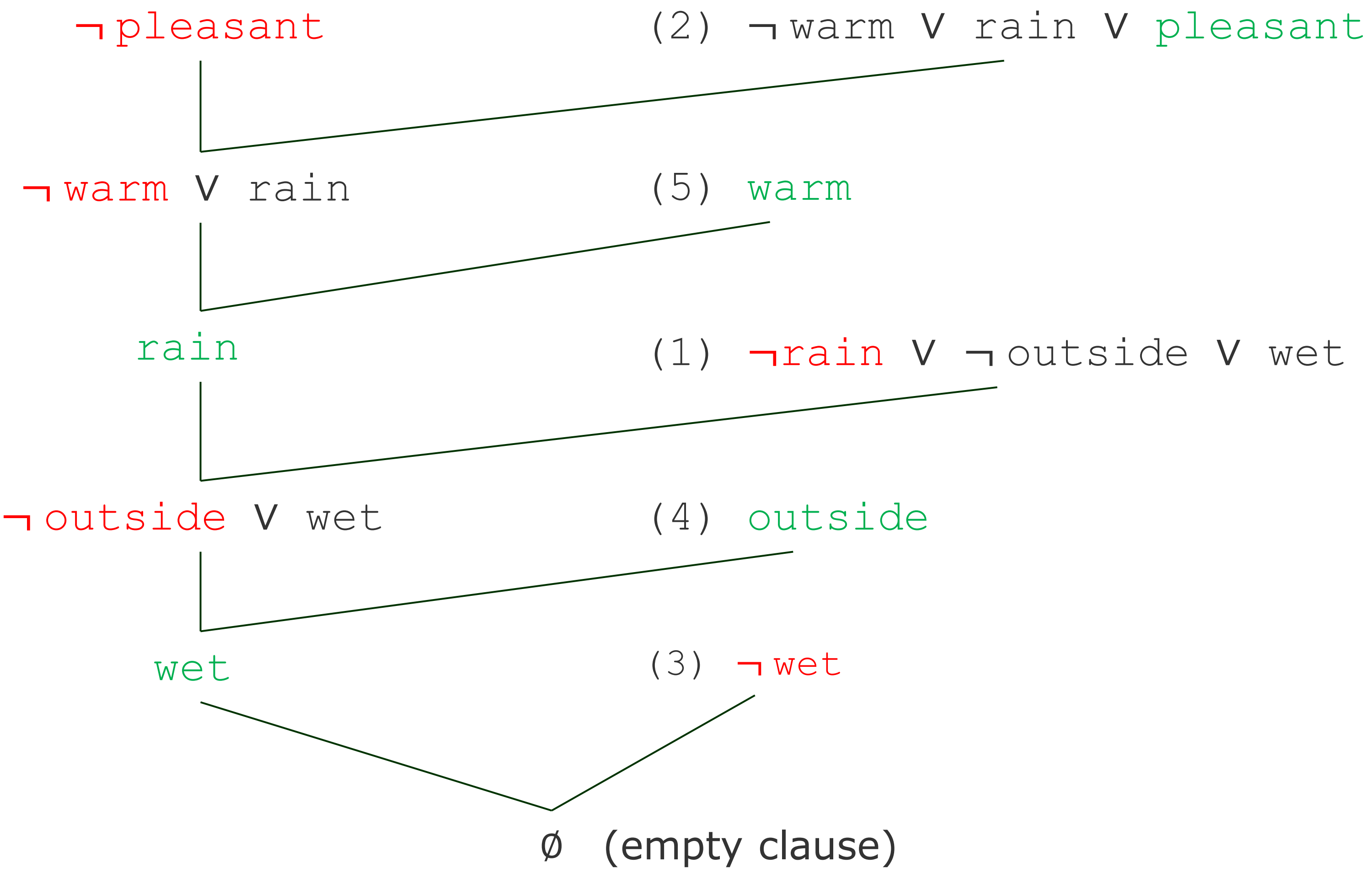
¬warm ∨ rain (5) warm

rain (1) ¬rain ∨ ¬outside ∨ wet

¬outside ∨ wet (4) outside

wet (3) ¬wet

Resolution Refutation Tree



¬pleasant

(2) ¬warm ∨ rain ∨ pleasant

¬warm ∨ rain

(5) warm

rain

(1) ¬rain ∨ ¬outside ∨ wet

¬outside ∨ wet

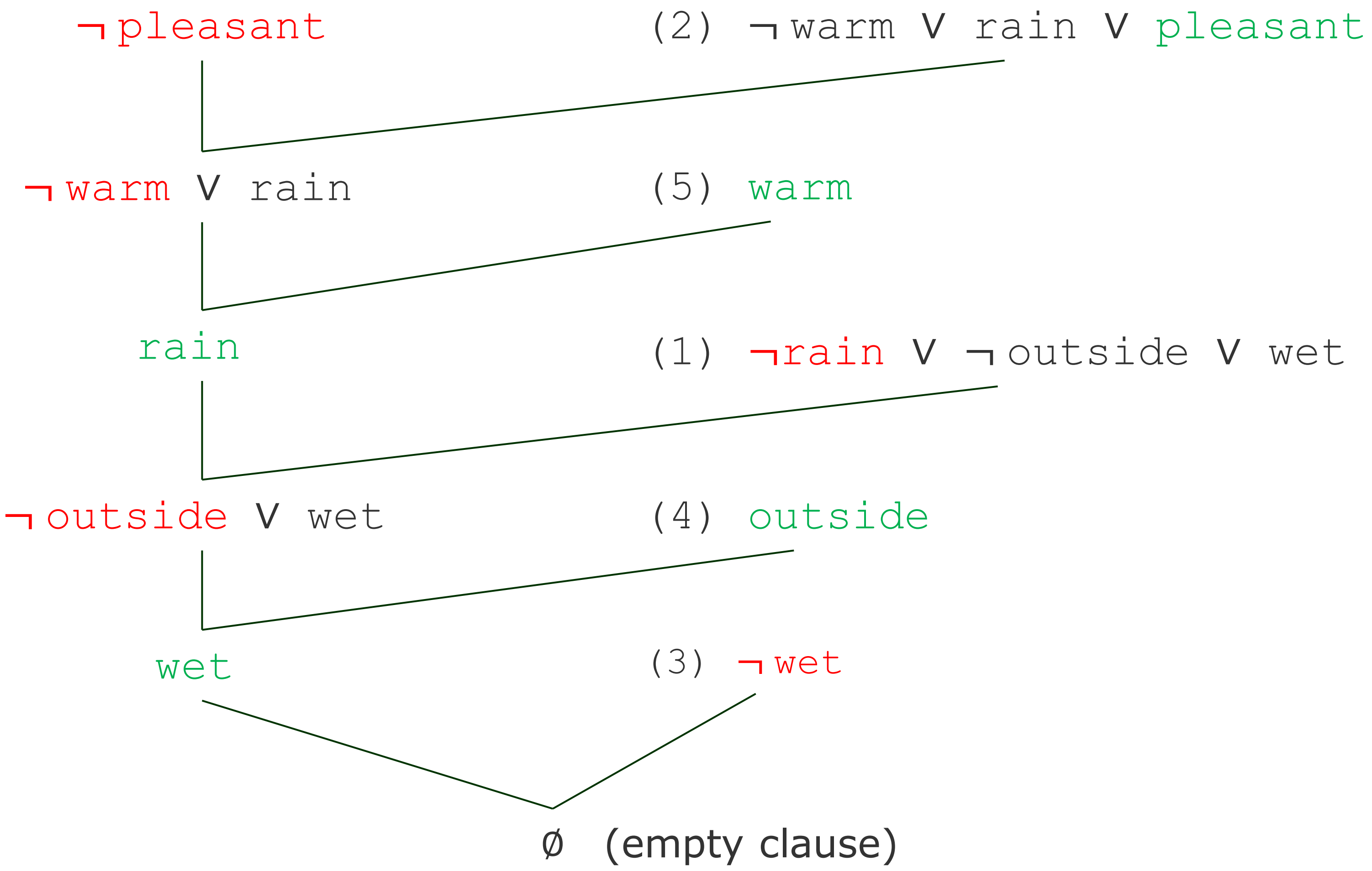
(4) outside

wet

(3) ¬wet

∅ (empty clause)

Resolution Refutation Tree



¬pleasant

(2) ¬warm V rain V pleasant

¬warm V rain

(5) warm

rain

(1) ¬rain V ¬outside V wet

¬outside V wet

(4) outside

wet

(3) ¬wet

Ø (empty clause)

contradiction!