If Σ is inconsistent, $\Box$ will be eventually derivable using resln.

$\{\Sigma, \neg\alpha\} \vdash \Box$

$X, X_a, \cancel{X}\{a\} \cup \{[ax]^n d\, X \in \{b, c\}\}$

Rule ① Rule ② $bab \to c$

$a \to \neg abv$

Strategy : Which rule to choose next

$\underbrace{\Sigma, \vdash R + \text{selection}}_{\text{proof procedure.}}$ strategy

Refutation Proof Procedure

(1) Let S be a set of clauses, α be a prop formula (clause)

$C = S \cup \{\neg\alpha\}$

(2) Repeatedly do the following:-

(a) Select a pair of clauses C_1 & C_2 from C
that can be resolved on a proposition P

(b) Resolve C_1 & C_2 to give R ; could be multiple Rs

(c) If $R = \square$ then stop $\{$Proof proc failed But $\Rightarrow S \models \alpha$ .... success$]$

o/w if R contains both a prop Q & $\neg Q$
then discard R .... $Q \vee \neg Q \vee$ .... $\{$Proof proc success but $S \not\models \alpha\}$

o/w add R to C

Remember: Resln is sound

So if $S \cup \{\neg\alpha\} \vdash_{R}$ substantial $\neq \Box$.

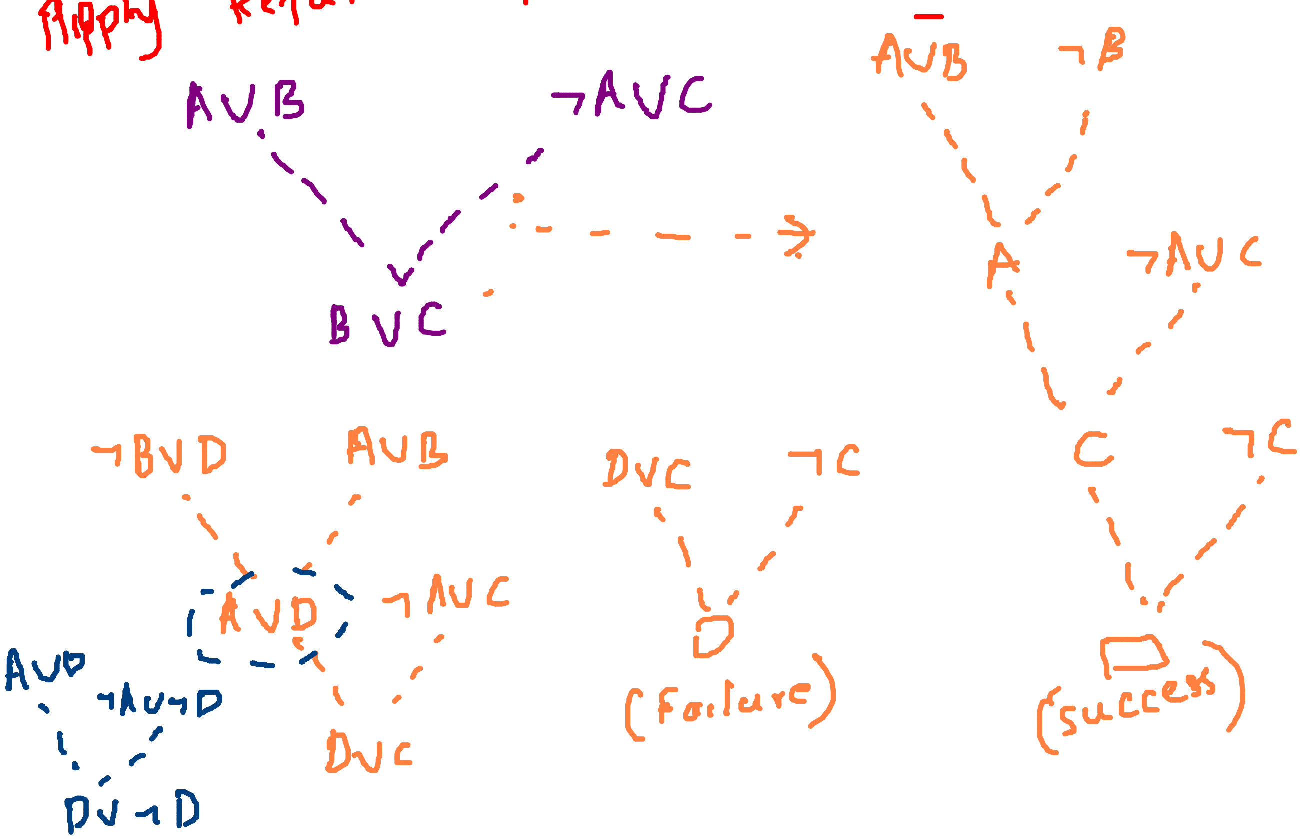
ie $S \cup \{\neg\alpha\} \models$ substantial

$\Rightarrow S \cup \{\neg\alpha\}$ is NOT a contradiction.

$\Sigma = \{ (A \vee B), (\neg A \vee C), (\neg B \vee D), (\neg C \vee F), \neg B, \neg C \}$

$\alpha = B \vee C$: clause $\qquad (\neg A \vee \neg D)$

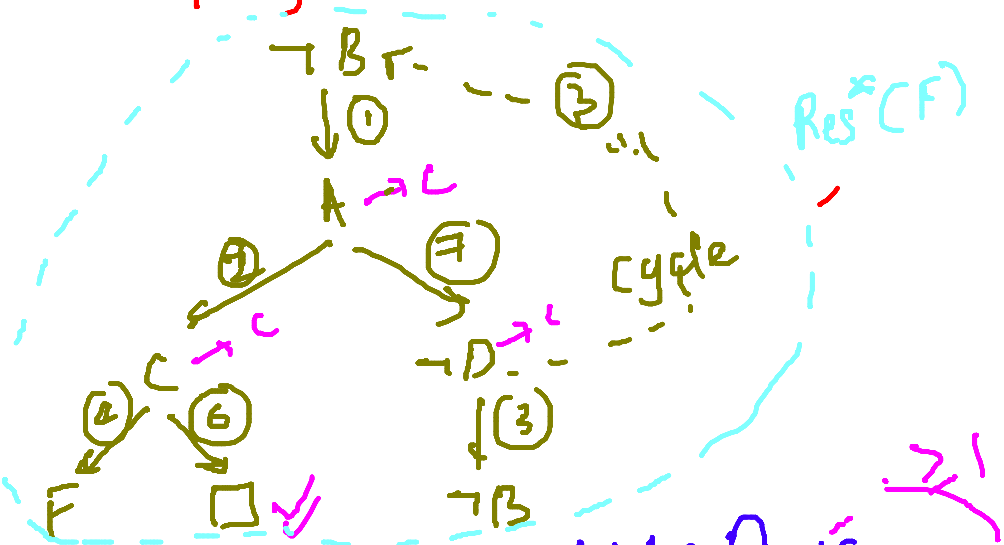
Apply Refutation proof.



$\Sigma = \{ \overset{1}{(A \lor B)}, \overset{2}{(\neg A \lor C)}, \overset{3}{(\neg B \lor D)}, \underset{4}{(\neg C \lor F)}, \underset{5}{\neg B}, \underset{6}{\neg C} \}$

$\underset{7}{(\neg A \lor \neg D)}$

$\alpha = \underbrace{B \lor C}_{\text{query}}$: clause



1. some leaves could be □ : Success
2. some leaves could fail
3. some paths have infinite loops

Naive proc:

(1) F : CNF $\Rightarrow \Sigma$ $\quad$ (eg:- $Res^0(F) = F = \Sigma \cup \{\neg\alpha\}$)

$F \equiv \Sigma$

(2) $Res^0(F)$: Be clauses in F

(3) $Res^n(F)$ $(n>0)$ :: clauses in $Res^{n-1}(F)$

$\cup$ clauses obtained by resolving pair of clauses in $Res^{n-1}(F)$

Variants of proof procedure

(4) Since there are only finite prop symbols in F,

$\Rightarrow$ only finite clauses in CNF, only finite clauses obtained by resoln ...

$\therefore$ For some n

$Res^{n-1}(F) = Res^n(F)$

$Res^*(F)$

n symbols, # clauses $= 2^n$ $\quad$ # 2^{2^n}

$\underbrace{\Sigma}_{\text{axioms}}$ $\underbrace{\alpha}_{\text{theorem}}$

$\Sigma \models \alpha$?

Refutation proof proc.

$$\Sigma \cup \{\neg \alpha\} \vdash_{Res^*} \Box$$

Theorem: for some F, $\Box \in Res^*(F)$ iff F is unsatisfiable.

Resolution is Refutation complete

completeness & soundness

$\Sigma \models \alpha \dashrightarrow \Sigma \vdash \alpha$ — Does not hold for resolution with arb. Σ & α

$\Sigma \vdash \alpha \dots \Sigma \models \alpha$ — Does hold for resolution with arb Σ & α

$\{A, B\} \nvdash A \leftarrow B$

$\{A, B\} \models A \leftarrow B$

Proof: (of only if)

If Σ :- inconsistent axiom (refutable)

α :- $\Box$

then for this restriction, resolution is complete.

Proof (only-if):-

If $\square \in Res^*(F)$ then F is unsatisfiable

① Since $F =$ CNF form $\square \notin F = Res^0(F)$

Because $\square$ is NOT a clause

Mostly :- $F = \Sigma \cup \{\neg\alpha\}$

application

$A \cup \neg A = T$ $\quad \cancel{A \wedge \neg A} = \square$

Not a clause

② ∴ For some k, $\square \notin Res^k(F)$, $\square \in Res^{k+1}(F)$

ie $\square$ was derived using resolution on $Res^k(F)$

$\{A, \neg A\} \vdash_{Res} \square$

This only means that $L, \neg L \in Res^k(F)$ (for some literal L)

Since resolution is SOUND

$M_F = \emptyset$ ie F is unsatisfiable

$F \vdash L \Rightarrow F \models L \qquad F \vdash \neg L \Rightarrow F \models \neg L$

$M_F \subseteq M_L \quad M_F \subseteq M_{\neg L}$. But $M_L \wedge M_{\neg L} = \emptyset$ ∴

Resolution
sound
Not complete

$+$

Subsumption

$||$

sound + complete

Σ :- Given

Need :- $\underbrace{\text{All theorems}}_{\text{Enumerate}}$ α $(\Sigma \models \alpha)$

Restricted form
of completeness
of Resolution

Need to be
given α

$\underbrace{\Sigma \cup \{\neg\alpha\} \vdash \Box}_{\text{Query answering}}$

Subsumption :-

A prop clause C subsumes prop clause D if $C \subseteq D$.

$A \vee B \equiv \{A, B\} \subseteq A \vee B \vee C \equiv \{A, B, C\}$

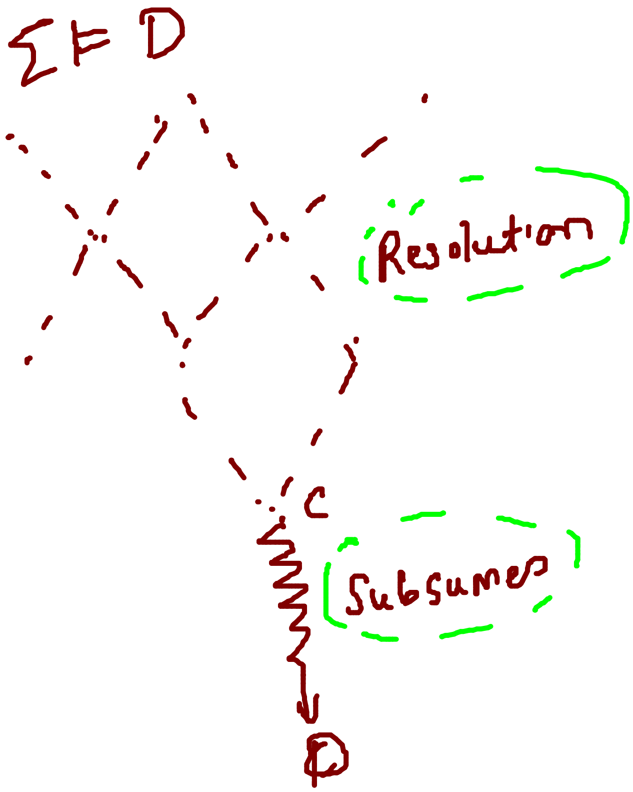
i.e. every literal in C appears in D.

Claim:- If $C \subseteq D$ then $C \models D$

Subsumption theorem :- If Σ is a set of clauses & D is a clause, then $\Sigma \models D$ iff either

(a) D is a tautology OR

(b) $\exists$ clause C st $\Sigma \vdash_{Res} C$ & $C \subseteq D$



Logical implication can be decomposed into a sequence of resolution steps . . . . followed by a subsumption.

(100%)
Precision : Measure of soundness

(100%)
Recall : Measure of completeness

For SUM/NB, Procedure is trivial
For FOL/PL, proc. is non-trivial

Proof :- "only if"

If $\Sigma \models D$ then $\exists$ C s.t $\Sigma \vdash_{Res} C$ & $C \subseteq D$

Induction on $|\Sigma|$:-

① $|\Sigma| = 1$ $\quad \Sigma = \{\underset{\text{clause}}{\alpha_1}\}$ $\quad \Sigma \models \underset{\text{clause}}{D}$

Since $\Sigma \vdash_{Res} \alpha_1$, need to show $\alpha_1 \subseteq D$

If α_1, D are clauses & $\alpha_1 \models D$, $\alpha_1 \subseteq D$

Proof by contr :- $L \in \alpha_1$ $L \notin D$

(2) Let thm hold for $|\Sigma| = n$. We will see if it holds for $|\Sigma| = n+1$.

$\Sigma = \{\alpha_1, \alpha_2 \ldots \alpha_{n+1}\}$ & $\Sigma \models D$

Using deduction thm, equivalently

$\Sigma - \alpha_{n+1} \models (D \leftarrow \alpha_{n+1})$ ie $\underbrace{\{\alpha_1 \ldots \alpha_n\}}_{\Sigma'} \models (D \vee \neg \alpha_{n+1})$

Let $L_1, L_2 \ldots L_k$ be literals in α_{n+1}, that do not appear in D

ie $\alpha_{n+1} = L_1 \vee L_2 \ldots \vee L_k \vee D'$ $\quad D' \subseteq D$

$\boxed{\Sigma' \models (D \vee \neg L_i) \quad \forall \quad i = 1 \ldots k}$ $\quad |\Sigma'| = n$

$\neg \alpha_{n+1} \vee D \equiv D \vee (\neg L_1 \wedge \neg L_2 \ldots \wedge \neg L_k \wedge \neg D')$

$\equiv (D \vee \neg L_1) \wedge (D \vee \neg L_2) \ldots \wedge (D \vee L_k)$

$\wedge \underbrace{(D \vee \neg D')}_{\text{true}}$

Using induction assumption

$\Sigma' \vdash_{Res} \beta_i \subseteq D \vee \neg L_i$

[To prove $\Sigma \vdash_{Res} C' \subseteq D$]

(a) Case 1 :- $\neg L_i \notin \beta_i$ (even for one 'i')

then $\beta_i \subseteq D$

$\therefore \Sigma' \vdash_{Res} \beta_i \subseteq D$.

$\therefore \Sigma \vdash_{Res} \beta_i \subseteq D$

(b) Case 2 :- $\neg L_i \in \beta_i$ for all i

ie $\beta_i = \neg L_i \vee \beta_i'$, $\beta_i' \subseteq D$

More Resolution using α_{n+1} & all β_i

Key: $\underbrace{\Sigma'}_{\Sigma \setminus \{\alpha_{n+1}\}} \vdash \beta_1, \beta_2 \ldots \beta_k \vdash \underbrace{\alpha_{n+1}}_{\in \Sigma} \vdash C \ldots \subseteq D$

$\beta_1' \vee \beta_2' \cdots \vee \beta_k' \vee D'$

$\alpha_{n+1} = L_1 \vee L_2 \ldots \vee L_k \vee D'$

'If part'

If $\Sigma \vdash C \sqsubseteq D$ then $\Sigma \models D$

soundness $\therefore \Sigma \models C$ $\quad C \models D$??

Proof of Refutation Comp.:

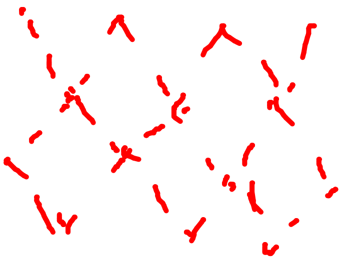
Prove its if part using subsumption thm.

If Σ is unsat, $\Sigma \models \Box$

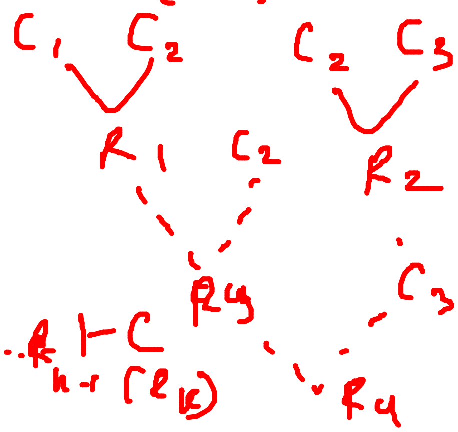
$\Rightarrow \Sigma \vdash C \sqsubseteq \Box$ But $C = \Box$

$\therefore \Sigma \vdash \Box$

Proof strategies (intelligent choice of which clauses to resolve)

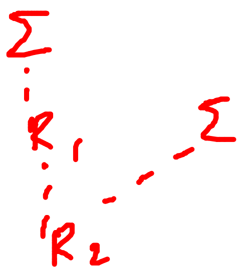


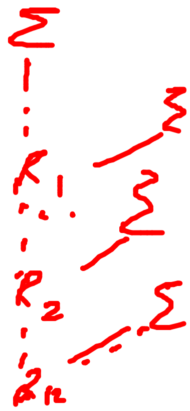
$\Sigma = \{C_1, C_2, C_3\}$



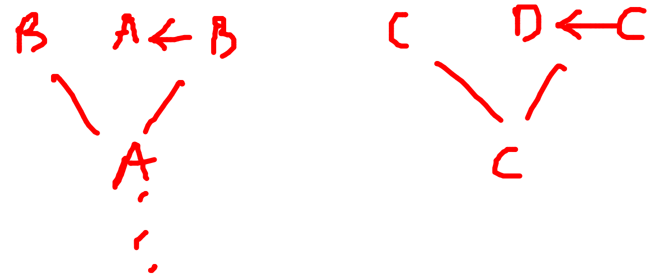
$\Sigma \vdash_{R^*} C \qquad \Sigma \underset{\tilde{\Sigma}}{\vdash} R_1 \vdash R_2 \cdots R_{k-1} \vdash C\ (R_k)$

Linear derivation





$\Sigma = \{A, B, A \leftarrow B, D \leftarrow C, C\}$

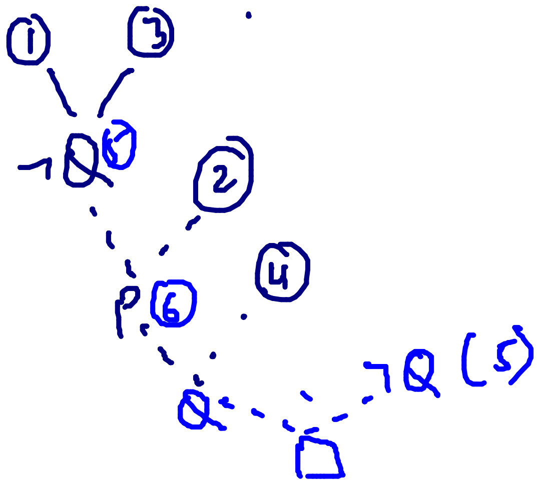


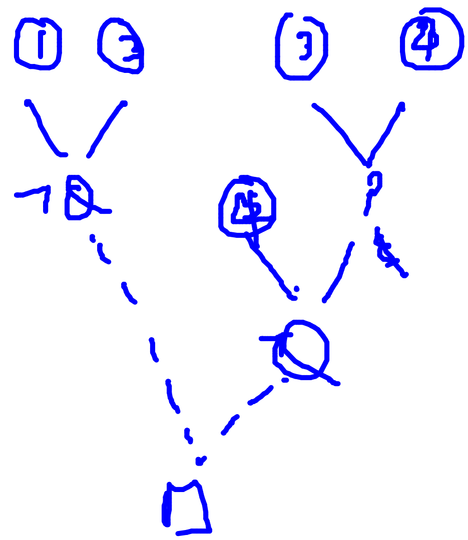
$\{P \leftarrow Q,\ P \leftarrow \neg Q,\ \neg P \leftarrow Q,\ \neg P \leftarrow \neg Q\}$

1 2 3 4

Linear derivation

~~Normal~~ deriv. (Brute force)





Much less search overhead.

Linear: $\Sigma \vdash R_1 \vdash_{\Sigma \cup \{R_1\}} R_2 \vdash_{\Sigma \cup \{R_1, R_2\}} R_3 \cdots R_{i-1} \vdash_{\Sigma \cup \{R_1 \ldots R_{i-2}\}} R_i$

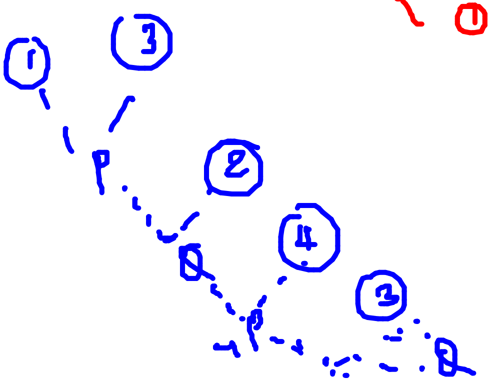
Q: Is this refutation complete? : Yes.

Input: $\Sigma \vdash R_1 \vdash_{\Sigma} R_2 \vdash_{\Sigma} R_3 \vdash \cdots R_{i-1} \vdash_{\Sigma} R_i$

Q: Is this refutation complete : No

$\{P \leftarrow Q, \; Q \leftarrow P, P, \; P \leftarrow \neg Q, \; \neg Q \leftarrow P\}$

1, 2, 3, 4



not comp

Input $\subsetneq$ Linear $\subseteq$ Brute.

soundness

Claim: If Σ is unsat. $(\Sigma \models \square)$ & $C \in \Sigma$ s.t. $\Sigma \setminus \{C\}$ is sat, then $\exists$ a linear ref of Σ with C as top clause

$C = \neg\alpha$.

$(\Sigma' \cup \{\neg\alpha\})$
consistent

Proof: ② Assume Σ is finite. Let $|\Sigma|$ be # of distinct atoms in Σ. eg: $\Sigma = \{\neg A \vee B, A \vee \neg B, A \vee B\}$

$|\Sigma| = 2$

Proof by induction on $|\Sigma| = n$

① $n = 0$, $\underbrace{\Sigma = \{\square\}}_{\text{sat?}}$ $\underbrace{\Sigma \setminus \{C\} = \{\}}_{\text{satisfiable}}$ $C = \square$

Case 2 : Assume for $n \leq m$. Prove for $m+1$

(a) $C = L$ (literal) . . . $\top$.

Removing clauses containing L

Removing $\neg L$ from o/r clauses