Risoluzione e clausole di Horn

Marco Piastra

Sistemi deduttivi e automazione

- Il problema $\Gamma \vdash \varphi$ è decidibile?

  La definizione di derivabilità 'a la Hilbert' (derivazione da schemi di assioma) non è direttamente traducibile in una procedura effettiva

  Perché:

  Volendo costruire una dimostrazione di $\Gamma \vdash \varphi$

  - Possiamo inserire una ad una le fbf $\psi \in \Gamma$ (OK)
  - Possiamo applicare la regola MP non appena possibile (OK)
  - NON possiamo inserire tutte le istanze degli assiomi Axn (KO)
    - A rigor di teoria, tali istanze sono infinite

  Morale: *il problema sono gli schemi di assioma, che ammettono infinite istanziazioni*

Regola di risoluzione

- Una nuova regola di inferenza (calcolo simbolico)

$$\varphi \vee \chi, \neg\chi \vee \psi \vdash \varphi \vee \psi$$

La fbf $\varphi \vee \psi$ viene anche detta *risolvente* di $\varphi \vee \chi$ e $\neg\chi \vee \psi$

La regola di risoluzione è corretta

φ	ψ	χ	$\varphi \vee \chi$	$\neg\chi \vee \psi$	$\varphi \vee \psi$
0	0	0	0	1	0
0	0	1	1	0	0
0	1	0	0	1	1
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	1

La regola del *modus ponens* è un caso particolare di *risoluzione*

$\chi \rightarrow \psi, \chi \vdash \psi$ può essere riscritta come $\chi, \neg\chi \vee \psi \vdash \psi$

Forme normali

= Traduzione di una fbf qualsiasi in una fbf logicamente equivalente

Le fbf in forma normale, in generale, hanno una struttura particolare

- **Forma normale congiuntiva** (FNC)

  Una fbf in cui ∧ appare solo al livello più esterno

  $$\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n$$

  dove α_i sono fbf in cui compaiono solo ∨ e ¬
  (¬ compare solo davanti ai simboli proposizionali)

  Esempi:

  $$(B \vee D) \wedge (A \vee \neg C) \wedge C$$
  $$(B \vee \neg A \vee \neg C) \wedge (\neg D \vee \neg A \vee \neg C)$$

- **Forma normale disgiuntiva** (FND)

  Come la precedente, ma si scambiano i ruoli di ∨ e ∧:

  $$\beta_1 \vee \beta_2 \vee \ldots \vee \beta_n$$

  dove β_i sono fbf in cui compaiono solo ∧ e ¬
  (¬ compare solo davanti ai simboli proposizionali)

Forma normale congiuntiva

- Traduzione automatica in FNC

  Applicazione esaustiva delle seguenti regole:

  1) Si eliminano $\rightarrow$ e $\leftrightarrow$ in base alle regole di riscrittura

  2) Si muove $\neg$ all'interno

  con le "leggi De Morgan": $\neg(\varphi \wedge \psi) \equiv (\neg\varphi \vee \neg\psi)$

  $\neg(\varphi \vee \psi) \equiv (\neg\varphi \wedge \neg\psi)$

  3) Si eliminano le doppie negazioni $\neg$

  4) Si distribuisce $\vee$

  proprietà distributiva di $\vee$: $((\varphi \wedge \psi) \vee \chi) \equiv ((\varphi \vee \chi) \wedge (\psi \vee \chi))$

Esempi:

$(\neg B \rightarrow D) \vee \neg(A \wedge C)$

$B \vee D \vee \neg(A \wedge C)$ (eliminazione di $\rightarrow$)

$B \vee D \vee \neg A \vee \neg C$ (De Morgan)

$\neg(B \rightarrow D) \vee \neg(A \wedge C)$

$\neg(\neg B \vee D) \vee \neg(A \wedge C)$ (eliminazione di $\rightarrow$)

$(B \wedge \neg D) \vee (\neg A \vee \neg C)$ (De Morgan)

$(B \vee \neg A \vee \neg C) \wedge (\neg D \vee \neg A \vee \neg C)$ (distribuzione di $\vee$)

Risoluzione per refutazione

- Algoritmo

  Di decisione del problema: $\Gamma \vdash \varphi$?

  Si trasforma il problema in $\Gamma \cup \{\neg\varphi\}$

  Se $\Gamma \vdash \varphi$, $\Gamma \cup \{\neg\varphi\}$ è incoerente, quindi è possibile derivare una contraddizione

  Si traduce $\Gamma \cup \{\neg\varphi\}$ in FNC e quindi in FC

  Si applica esaustivamente la regola di risoluzione ad un coppia di clausole risolvendo un **letterale** complementare

  Ciascuna possibile risoluzione viene effettuata una volta soltanto

  Terminazione:

  L'algoritmo termina con successo quando viene derivata una **clausola vuota** { }

  Diversamente, l'algoritmo termina quando la regola non è più applicabile (fallimento)

  Vantaggi:

  Niente assiomi

  Una sola operazione (applicazione della regola di risoluzione)

  Procedura effettiva (algoritmo)

Derivazioni: risoluzione per refutazione

- L'esempio visto in precedenza

  $B \vee D \vee \neg A \vee \neg C, B \vee C, A \vee D, \neg B \vdash D$

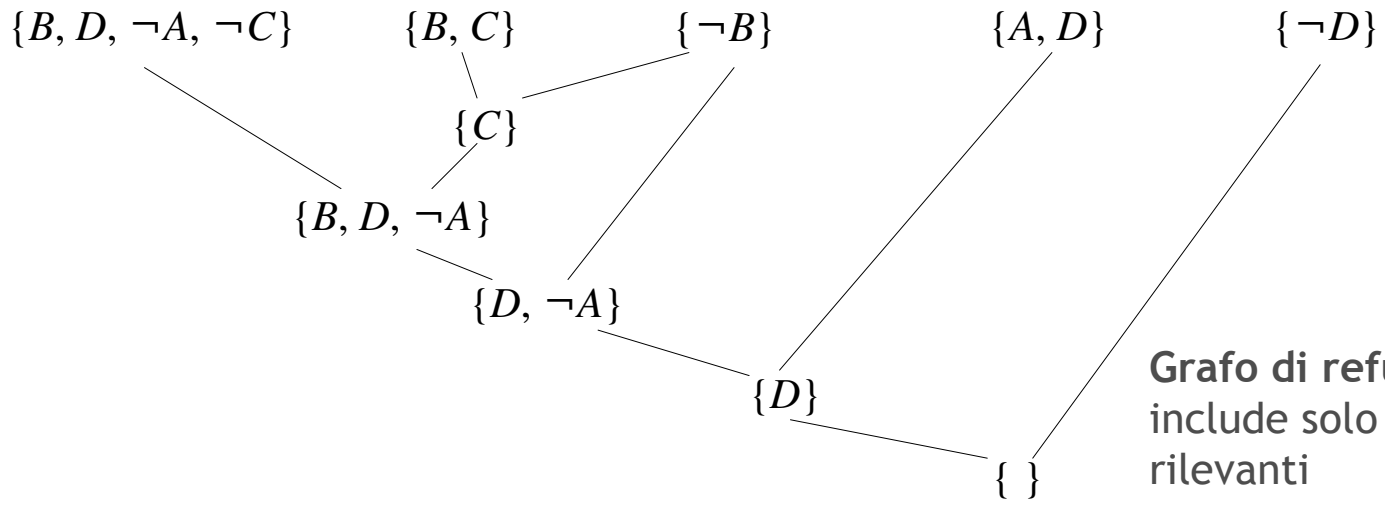
  Refutando la tesi e riscrivendo il tutto in FNC:

  $B \vee D \vee \neg A \vee \neg C, B \vee C, A \vee D, \neg B, \neg D$

  In FC:

  $\{B, D, \neg A, \neg C\}, \{B, C\}, \{A, D\}, \{\neg B\}, \{\neg D\}$

  Applicando la regola di risoluzione:



Grafo di refutazione: include solo le risoluzioni rilevanti

Derivazioni: risoluzione per refutazione

- L'esempio visto in precedenza

  $B \vee D \vee \neg A \vee \neg C, B \vee C, A \vee D, \neg B \vdash D$

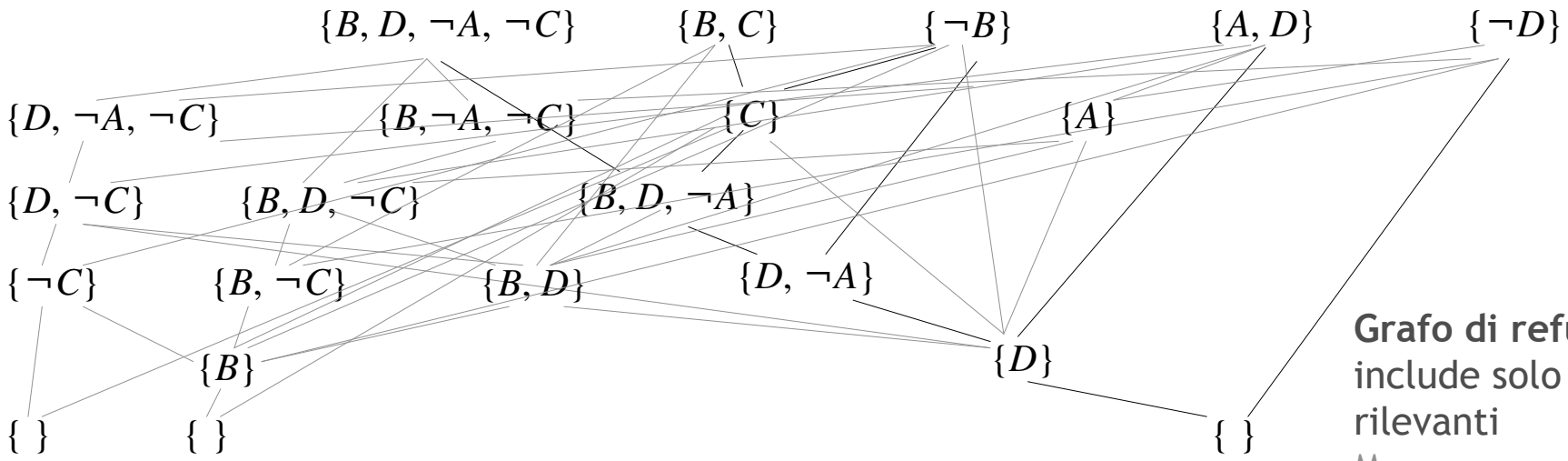
  Refutando la tesi e riscrivendo il tutto in FNC:

  $B \vee D \vee \neg A \vee \neg C, B \vee C, A \vee D, \neg B, \neg D, \neg E$

  In FC:

  $\{B, D, \neg A, \neg C\}, \{B, C\}, \{A, D\}, \{\neg B\}, \{\neg D\}$

  Applicando la regola di risoluzione:



Grafo di refutazione:
include solo le risoluzioni rilevanti
Ma ce ne sono altre ...

Risoluzione per refutazione

- Il metodo della risoluzione per refutazione in L_P
  - E` corretto: $\Gamma \vdash \varphi \Rightarrow \Gamma \models \varphi$
  - E` completo (in senso forte): $\Gamma \models \varphi \Rightarrow \Gamma \vdash \varphi$
    - Si intende: se $\Gamma \models \varphi$ allora esiste un grafo di refutazione

- Algoritmo di risoluzione per refutazione in L_P
  - E' una procedura di decisione per $\Gamma \vdash \varphi$
    - Se esiste un grafo di refutazione, l'algoritmo lo trova
  - Ha complessità di tempo $O(2^n)$
    - Dove n è il numero dei simboli proposizionali che compaiono in $\Gamma \cup \{\neg\varphi\}$

Minore potenza espressiva, maggiore efficienza di calcolo: *clausole di Horn*

Clausole di Horn

- Definizione

  Una **clausola di Horn** è una fbf in FC
  in cui si ha al massimo un letterale in forma positiva

- Tre tipi di clausole di Horn:

  Regole: più letterali, uno solo positivo

  Esempi: $\{B, \neg D, \neg A, \neg C\}$, $\{A, \neg B\}$ (traduzione: $(D \wedge A \wedge C) \rightarrow B$, $B \rightarrow A$)

  Fatti: un solo letterale, positivo

  Esempi: $\{B\}$, $\{A\}$

  Goal: uno o più letterali, tutti negativi

  Esempi: $\{\neg B\}$, $\{\neg A, \neg B\}$ (refutazione di: B, $A \wedge B$)

  Ulteriore terminologia

  Regole e fatti si dicono anche clausole definite (*definite clauses*)

  I goal si dicono anche clausole negative (*negative clauses*)

Traduzione in clausole di Horn

Molte fbf sono traducibili in clausole di Horn:

$(A \wedge B) \rightarrow C$
$\neg(A \wedge B) \vee C$ (eliminazione di $\rightarrow$)
$\neg A \vee \neg B \vee C$ (De Morgan - FC - è una regola)

$A \rightarrow (B \wedge C)$
$\neg A \vee (B \wedge C)$ (eliminazione di $\rightarrow$)
$(\neg A \vee B) \wedge (\neg A \vee C)$ (distribuzione di $\vee$)
$(\neg A \vee B), (\neg A \vee C)$ (FC – si generano due regole)

$(A \vee B) \rightarrow C$
$\neg(A \vee B) \vee C$ (eliminazione di $\rightarrow$)
$(\neg A \wedge \neg B) \vee C$ (De Morgan)
$(\neg A \vee C) \wedge (\neg B \vee C)$ (distribuzione di $\vee$)
$(\neg A \vee C), (\neg B \vee C)$ (FC – si generano due regole)

Ma non tutte:

$(A \wedge \neg B) \rightarrow C$
$\neg(A \wedge \neg B) \vee C$ (eliminazione di $\rightarrow$)
$\neg A \vee B \vee C$ (De Morgan)

$A \rightarrow (B \vee C)$
$\neg A \vee B \vee C$ (eliminazione di $\rightarrow$)

Risoluzione SLD

Linear resolution with Selection function for Definite clauses

Metodo particolare di risoluzione per refutazione per le clausole di Horn

- Algoritmo

  Si applica ad un set di *clausole definite* (*programma*) + un *goal*

  1) La funzione di selezione (*selection function*) identifica un *letterale* (*subgoal*)
  2) Si seleziona una *clausola definita* applicabile al *subgoal* selezionato
  3) Si applica la regola di risoluzione, sostituendo al *subgoal* risolto i *subgoal* risolventi della *clausola definita* applicata

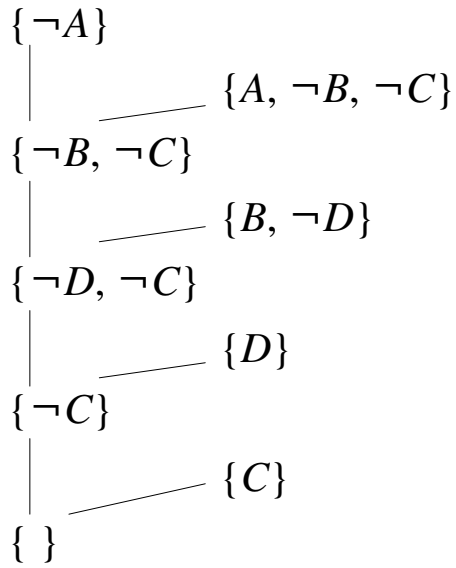
  Terminazione:
  si procede finché si ottiene { } oppure il passo 2) fallisce.

  Esempio:

  Selection function: *leftmost subgoal first*

  Clausole definite: $\{C\}$, $\{D\}$, $\{B, \neg D\}$, $\{A, \neg B, \neg C\}$

  Goal: $\{\neg A\}$



Alberi SLD

Derivazioni SLD

Esempio: $\{C\}$, $\{D\}$, $\{B, \neg D\}$, $\{A, \neg B, \neg C\}$ goal $\{\neg A\}$

In questo caso, c'è al massimo un modo per risolvere ciascun *subgoal*

In generale non è così

Progressione *lineare* →

$\{\neg A\}$
|
$\{\neg B, \neg C\}$
|
$\{\neg D, \neg C\}$
|
$\{\neg C\}$
|
$\{\ \}$

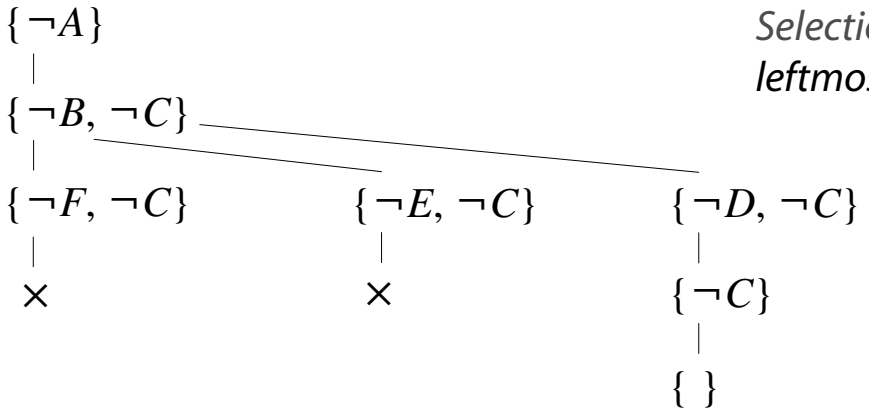
- Alberi SLD (= tutte le possibili derivazioni SLD a partire da un goal)

Una diramazione per ciascun modo di risolvere un *subgoal*

Esempio: $\{C\}$, $\{D\}$, $\{B, \neg F\}$, $\{B, \neg E\}$, $\{B, \neg D\}$, $\{A, \neg B, \neg C\}$ goal $\{\neg A\}$

Sono state aggiunte nuove regole: esistono ora diverse possibilità

Albero SLD



Selection function:
leftmost subgoal first

Metodo della risoluzione SLD

- Il metodo della risoluzione SLD in L_P

  Si applica solo alle clausole di Horn (clausole definite + *goal*)

  E` corretto: $\Gamma \vdash \varphi \Rightarrow \Gamma \models \varphi$

  Termina sempre

  E` completo: $\Gamma \models \varphi \Rightarrow \Gamma \vdash \varphi$

- Complessità computazionale:

  Ha complessità polinomiale (rispetto al numero di lettere ed alla dimensione delle fbf)

  Rispetto ai metodi generali, limitandosi alle clausole di Horn
  si ottiene un'efficienza molto elevata

- Limitazioni

  Non tutti i problemi sono traducibili in clausole di Horn

  "Giorgio è contento" non lo è

  Solo combinazioni di regole e fatti (con fatti – negati – come goal)

Forward Chaining

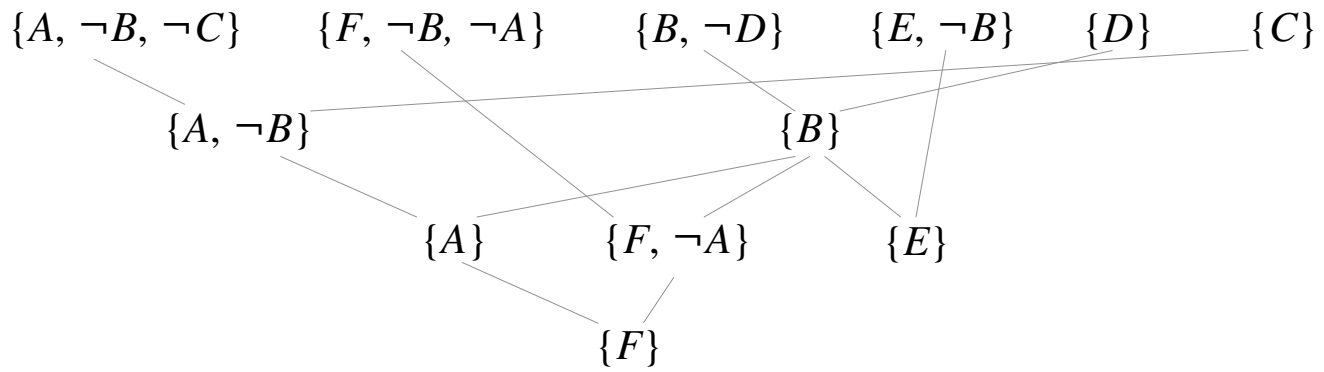
- Risoluzione senza refutazione per le clausole di Horn

  Esempio (clausole di Horn):

  $\{C\}, \{D\}, \{E, \neg B\}, \{B, \neg D\}, \{F, \neg B, \neg A\}, \{A, \neg B, \neg C\} \vdash \{A\}$

  (Può essere riscritto come: $C,\ D,\ B \rightarrow E,\ D \rightarrow B, (B \wedge A) \rightarrow F, (B \wedge C) \rightarrow A \vdash A$)

  Si applica *esaustivamente* la regola di risoluzione,
  CON UN VINCOLO: risoluzione solo tra regole e fatti



La risposta è positiva: $\Gamma \vdash \{A\}$. Ma si ottiene anche $\Gamma \vdash \{B\}, \Gamma \vdash \{E\}, \Gamma \vdash \{F\}$

L'insieme $\{\{A\}, \{B\}, \{E\}, \{F\}\}$ contiene tutti i *fatti* derivabili da Γ

Si può dimostrare che il metodo è completo (nel senso di cui sopra)

Forward Chaining

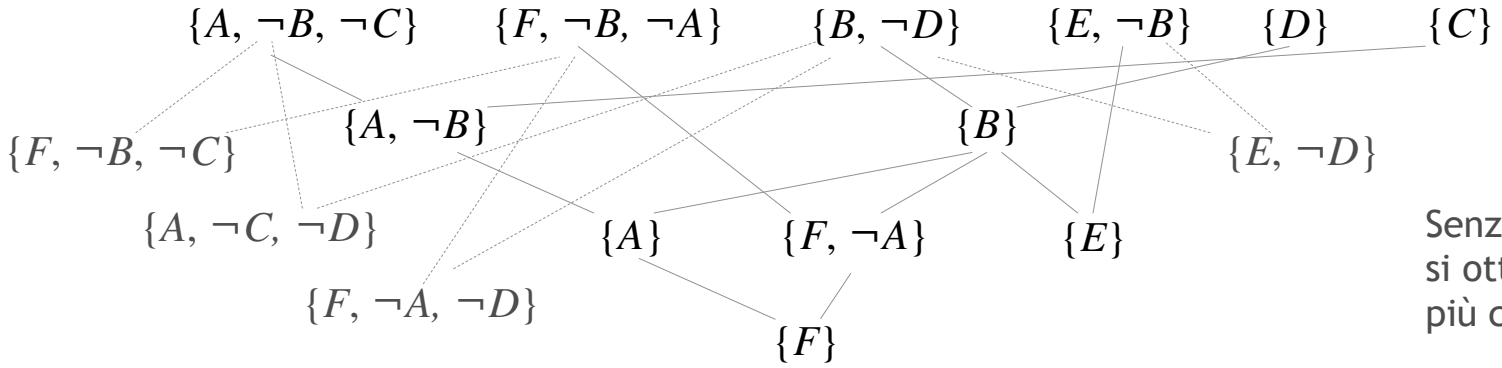
- Risoluzione <u>senza</u> refutazione per le clausole di Horn

  Esempio (clausole di Horn):

  $\{C\}, \{D\}, \{E, \neg B\}, \{B, \neg D\}, \{F, \neg B, \neg A\}, \{A, \neg B, \neg C\} \vdash \{A\}$

  (Può essere riscritto come: $C,\ D,\ B \rightarrow E,\ D \rightarrow B, (B \wedge A) \rightarrow F, (B \wedge C) \rightarrow A \vdash A$)

  Si applica *esaustivamente* la regola di risoluzione,
  CON UN VINCOLO: risoluzione <u>solo</u> tra regole e fatti



Senza il VINCOLO
si otterrebbero
più clausole

La risposta è positiva: $\Gamma \vdash \{A\}$. Ma si ottiene anche $\Gamma \vdash \{B\}, \Gamma \vdash \{E\}, \Gamma \vdash \{F\}$

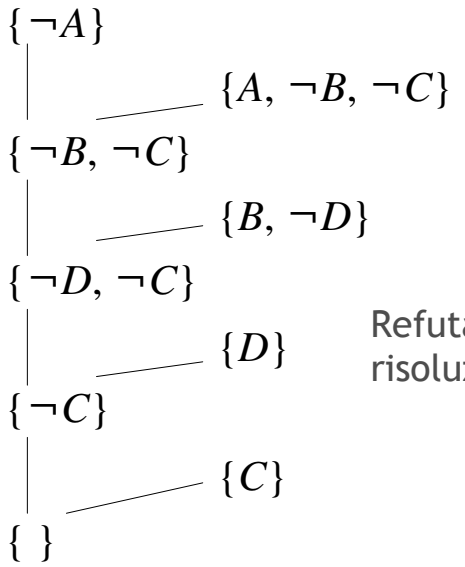
L'insieme $\{\{A\}, \{B\}, \{E\}, \{F\}\}$ contiene tutti i *fatti* derivabili da Γ

Si può dimostrare che il metodo è completo (nel senso di cui sopra)

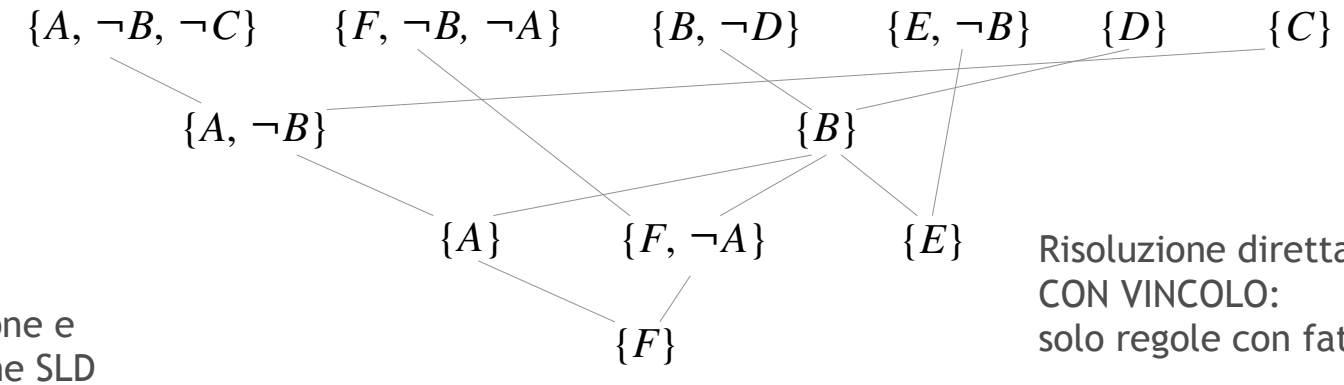
Forward & Backward

- Confronto tra risoluzione <u>con</u> e <u>senza</u> refutazione

  Esempio (come prima): $\{C\}, \{D\}, \{E, \neg B\}, \{B, \neg D\}, \{F, \neg B, \neg A\}, \{A, \neg B, \neg C\} \vdash \{A\}$



Refutazione e risoluzione SLD



Risoluzione diretta CON VINCOLO: solo regole con fatti

La risoluzione con refutazione agisce all'indietro (*backward*): si parte dal goal e si cerca { }

Risponde esclusivamente alla domanda: $\Gamma \vdash \{A\}$?

La risoluzione diretta agisce in avanti (*forward*): si concatenano regole e fatti

Trova tutti i fatti derivabili: $\{\varphi \mid \Gamma \vdash \{\varphi\}\}$, dove φ è un simbolo proposizionale