

Intelligenza Artificiale

Logica proposizionale classica (Parte 2)

Marco Piastra

Introduzione alla logica formale

Parte 1. Preambolo: algebra di Boole, proposizioni, conseguenza logica

Parte 2. Logica proposizionale

Parte 3. Calcolo dei predicati (in forma ridotta)

Parte 2

Logica proposizionale

Logica proposizionale – Linguaggio

- Un linguaggio proposizionale L_p contiene:
 - un insieme non vuoto P di **lettere proposizionali**: a, b, c, \dots
 - due *connettivi* principali: \neg, \rightarrow
 - due simboli ausiliari: $(,)$ (le parentesi)
 - tre *connettivi* derivati: $\wedge, \vee, \leftrightarrow$
 - regole sintattiche per la composizione di formule, dette **formule ben formate** - *fbf*
 - l'insieme di tutte le fbf di L_p si indica con $\text{fbf}(L_p)$
 - $P \in \text{fbf}(L_p)$
 - $\varphi \in \text{fbf}(L_p) \Rightarrow (\neg\varphi) \in \text{fbf}(L_p)$
 - $\varphi, \psi \in \text{fbf}(L_p) \Rightarrow (\varphi \rightarrow \psi) \in \text{fbf}(L_p)$
 - $\varphi, \psi \in \text{fbf}(L_p), (\varphi \vee \psi) \Leftrightarrow ((\neg\varphi) \rightarrow \psi)$
 - $\varphi, \psi \in \text{fbf}(L_p), (\varphi \wedge \psi) \Leftrightarrow (\neg(\varphi \rightarrow (\neg\psi)))$
 - $\varphi, \psi \in \text{fbf}(L_p), (\varphi \leftrightarrow \psi) \Leftrightarrow ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$

LP – Regole semantiche

- Un' *interpretazione* di L_p è una funzione $v : \text{fbf}(L_p) \rightarrow \{0, 1\}$
 - $\varphi, \psi \in \text{fbf}(L_p)$
 - $\varphi \in P$, $v(\varphi) =$ valore assegnato alla lettera proposizionale
 - $v(\neg\varphi) = 1$ sse $v(\varphi) = 0$
 - $v(\varphi \wedge \psi) = 1$ sse $v(\varphi) = 1$ e $v(\psi) = 1$
 - $v(\varphi \vee \psi) = 1$ sse $v(\varphi) = 1$ o $v(\psi) = 1$
 - $v(\varphi \rightarrow \psi) = 1$ sse non $v(\varphi) = 1$ e $v(\psi) = 0$
 - $v(\varphi \leftrightarrow \psi) = 1$ sse $v(\varphi) = v(\psi)$
- in sintesi, un'interpretazione v di L_p è coerente con le tavole di verità viste in precedenza

LP – Modelli e soddisfacibilità

- Data una fbf φ ed un'interpretazione v tale per cui $v(\varphi) = 1$
- Si dice che:
 - v **soddisfa** φ
 - v è un **modello** di φ
 - si scrive anche $v \models \varphi$
- La definizione è facilmente estesa agli insiemi di fbf $\Gamma = \{\varphi_1, \dots, \varphi_n\}$
 - $v \models \Gamma$ sse per $\forall \varphi_i \in \Gamma, v \models \varphi_i$
- Una fbf è una **tautologia** (o una fbf **valida**) se è soddisfatta da qualsiasi interpretazione
 - φ è una tautologia si scrive anche $\models \varphi$
- Una fbf φ è una **contraddizione** se non ha un modello

LP – Conseguenza, decidibilità

- Una fbf ψ è una *conseguenza logica* di un insieme di fbf Γ se e solo se qualsiasi modello di Γ è anche modello di ψ

– si scrive anche:

$$\Gamma \models \psi$$

- Il problema della **decidibilità** in LP:
esistenza di un algoritmo per stabilire se:

$$\models \varphi$$

$$\Gamma \models \psi$$

- LP è **decidibile**
 - basta usare il metodo delle tavole di verità

LP – Derivazione

- Una regola di **derivazione** (anche regola di **inferenza**) permette di *derivare* fbf da altre fbf
- In logica proposizionale si ha una sola regola di derivazione

– *modus ponens* (*mp*):

$$\frac{\varphi \rightarrow \psi \quad \varphi}{\psi}$$

– si può scrivere anche così:

$$\varphi \rightarrow \psi, \varphi \vdash \psi \quad (\text{da } \varphi \rightarrow \psi \text{ e } \varphi \text{ è derivabile } \psi)$$

- Esempio di applicazione:

– dalle due formule

$$(\neg a \rightarrow b) \rightarrow (c \rightarrow \neg d)$$

$$(\neg a \rightarrow b)$$

– si può derivare

$$(c \rightarrow \neg d)$$

Una regola di derivazione è di tipo *sintattico* in quanto opera sulla struttura delle fbf

LP – Assiomi

- Gli assiomi di un sistema logico esprimono *leggi logiche* di validità generale (nel sistema stesso)
- In logica proposizionale si usano degli *schemi di assioma* :
 - $\forall \varphi, \psi, \chi \in \text{fbf}(L_p)$
 - Ax1 $\varphi \rightarrow (\psi \rightarrow \varphi)$
 - Ax2 $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$
 - Ax3 $(\neg\varphi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \varphi)$
 - le lettere φ, ψ, χ sono fbf qualsiasi di L_p
 - ogni sostituzione delle lettere di Ax1, Ax2, Ax3 con fbf è un assioma
- Esempi
 - $a \rightarrow (a \rightarrow a)$ [Ax1, $\varphi/a, \psi/a$]
 - $(\neg(b \vee c) \rightarrow \neg d) \rightarrow (d \rightarrow (b \vee c))$ [Ax3, $\varphi/(b \vee c), \psi/d$]
 - Notare che ogni istanziazione è anche una fbf *valida* (o *tautologia*)

LP – Derivazioni

- Una *dimostrazione* (o *derivazione*) di una fbf ψ da un insieme di fbf Γ
 - è una successione *finita* di passi $\langle \alpha_1, \dots, \alpha_n \rangle$
 - per ogni passo α_j si hanno tre alternative:
 - 1) $\alpha_j \in$ istanza di Ax_n
 - 2) $\alpha_j \in \Gamma$
 - 3) α_j è ottenibile dalle fbf dei passi precedenti, tramite *modus ponens*
 - $\alpha_n = \psi$
 - quando la derivazione esiste, si scrive $\Gamma \vdash \psi$ “ ψ è derivabile da Γ ”
- Ne segue che:
 - $\vdash Ax$ (un assioma è derivabile anche da un Γ vuoto)
 - $\Gamma \vdash Ax$ (un assioma è derivabile da qualsiasi Γ)
 - $\{\varphi, \dots\} \vdash \varphi$ (qualsiasi φ è derivabile da un Γ che già la contiene)

LP – Regole aggiuntive

- Nelle dimostrazioni si possono anche utilizzare 'scorciatoie'
 - Regole aggiuntive, dimostrate a partire dalle definizioni
 - Teoremi, cioè schemi del tipo $\Gamma \vdash \varphi$ dimostrati in precedenza

- *Riflessività*

$$\varphi \vdash \varphi$$

(Dimostrazioni sul testo)

- *Monotonia sintattica*

$$\text{se } \Gamma \subseteq \Sigma, \Gamma \vdash \varphi \text{ allora } \Sigma \vdash \varphi$$

- *Compattezza*

$$\text{se } \Gamma \vdash \varphi \text{ allora esiste un insieme } \Delta \text{ **finito**, con } \Delta \subseteq \Gamma, \text{ per cui } \Sigma \vdash \varphi$$

- *Transitività*

$$\text{se per ogni } \varphi \in \Delta \text{ si ha che } \Gamma \vdash \varphi \text{ e } \Delta \vdash \psi \text{ allora } \Gamma \vdash \psi$$

- *Regola di deduzione*

$$\Gamma \cup \{\varphi\} \vdash \psi \text{ equivale a } \Gamma \vdash (\varphi \rightarrow \psi)$$

LP – Derivazioni, esempio 1

- Doppia negazione implica affermazione

$$\vdash \neg\neg\varphi \rightarrow \varphi$$

| | | |
|----|--|----------|
| 1: | $\vdash \neg\neg\varphi \rightarrow (\neg\neg\neg\neg\varphi \rightarrow \neg\neg\varphi)$ | (Ax1) |
| 2: | $\neg\neg\varphi \vdash \neg\neg\neg\neg\varphi \rightarrow \neg\neg\varphi$ | (Ded) |
| 3: | $\neg\neg\varphi \vdash (\neg\neg\neg\neg\varphi \rightarrow \neg\neg\varphi) \rightarrow (\neg\varphi \rightarrow \neg\neg\neg\varphi)$ | (Ax3) |
| 4: | $\neg\neg\varphi \vdash \neg\varphi \rightarrow \neg\neg\neg\varphi$ | (MP 2,1) |
| 5: | $\neg\neg\varphi \vdash (\neg\varphi \rightarrow \neg\neg\neg\varphi) \rightarrow (\neg\neg\varphi \rightarrow \varphi)$ | (Ax3) |
| 6: | $\neg\neg\varphi \vdash \neg\neg\varphi \rightarrow \varphi$ | (MP 4,3) |
| 7: | $\neg\neg\varphi \vdash \neg\neg\varphi$ | (Rif) |
| 8: | $\neg\neg\varphi \vdash \varphi$ | (MP 5,6) |
| 9: | $\vdash \neg\neg\varphi \rightarrow \varphi$ | (Ded) |

LP – Derivazioni, esempio 2

- “Ex absurdo sequitur quodlibet”:

$\vdash \varphi \rightarrow (\neg\varphi \rightarrow \psi)$ (cioè $\varphi, \neg\varphi \vdash \psi$)

| | | |
|----|---|-------------|
| 1: | $\varphi, \neg\varphi \vdash \neg\varphi \rightarrow (\neg\psi \rightarrow \neg\varphi)$ | (Ax1 + Mon) |
| 2: | $\varphi, \neg\varphi \vdash \neg\varphi$ | (Rif + Mon) |
| 3: | $\varphi, \neg\varphi \vdash \neg\psi \rightarrow \neg\varphi$ | (MP 1,2) |
| 4: | $\varphi, \neg\varphi \vdash (\neg\psi \rightarrow \neg\varphi) \rightarrow (\varphi \rightarrow \psi)$ | (Ax3) |
| 5: | $\varphi, \neg\varphi \vdash \varphi \rightarrow \psi$ | (MP 4,3) |
| 6: | $\varphi, \neg\varphi \vdash \varphi$ | (Rif + Mon) |
| 7: | $\varphi, \neg\varphi \vdash \psi$ | (MP 5,6) |
| 8: | $\varphi \vdash \neg\varphi \rightarrow \psi$ | (Ded) |
| 9: | $\vdash \varphi \rightarrow (\neg\varphi \rightarrow \psi)$ | (Ded) |

LP – Correttezza e completezza

- *Correttezza*

- la regola del *modus ponens* preserva la conseguenza logica

$$\Gamma \models \{\varphi \rightarrow \psi, \varphi\} \Rightarrow \Gamma \models \psi$$

- le fbf φ *derivabili* da un insieme di fbf Γ sono una *conseguenza logica* di Γ (sono soddisfatte dai modelli di Γ)

$$\Gamma \vdash \varphi \Rightarrow \Gamma \models \varphi$$

- tutte le fbf *derivabili* dagli assiomi A_{Xn} assiomi sono *valide* (cioè sono *tautologie*)

$$\vdash \varphi \Rightarrow \models \varphi$$

- *Completezza*

- le *conseguenze logiche* di Γ sono le fbf φ *derivabili*

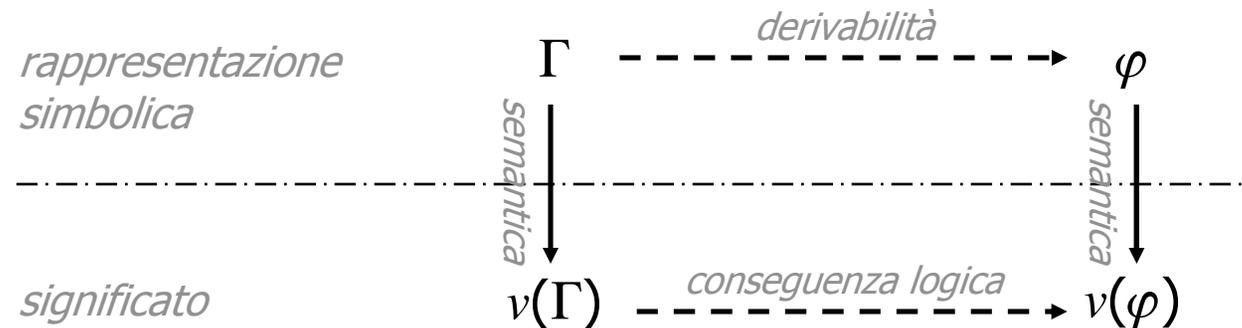
$$\Gamma \vdash \varphi \Leftrightarrow \Gamma \models \varphi$$

- le fbf *valide* sono tutte e sole le fbf *derivabili* dagli assiomi A_{Xn}

$$\vdash \varphi \Leftrightarrow \models \varphi$$

LP - Calcolo simbolico

- Per la proprietà di *completezza*, la *derivazione* simbolica è rappresentativa delle relazioni tra i significati



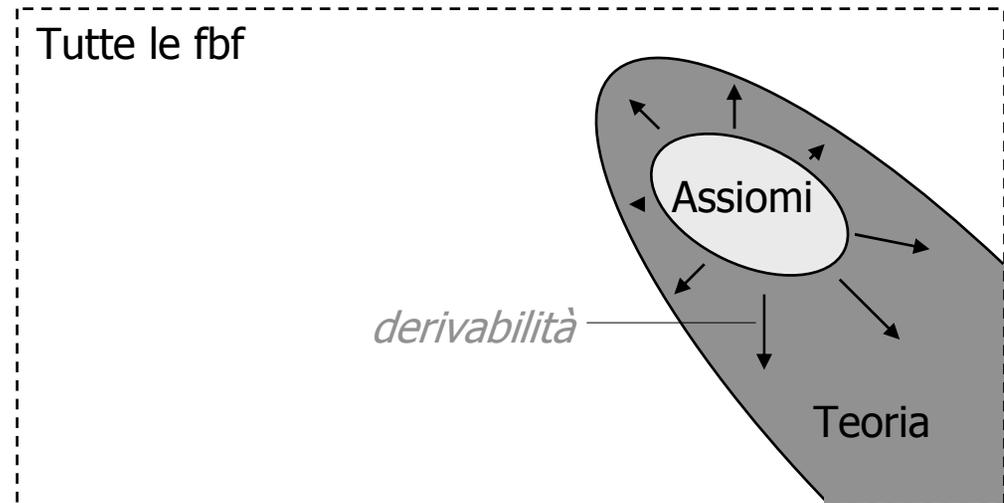
- Nel caso della logica proposizionale, la relazione di *conseguenza logica* può essere determinata in modo diretto
- In molti altri casi, la derivazione simbolica è l'unica possibilità

LP – Teorie, assiomatizzazione

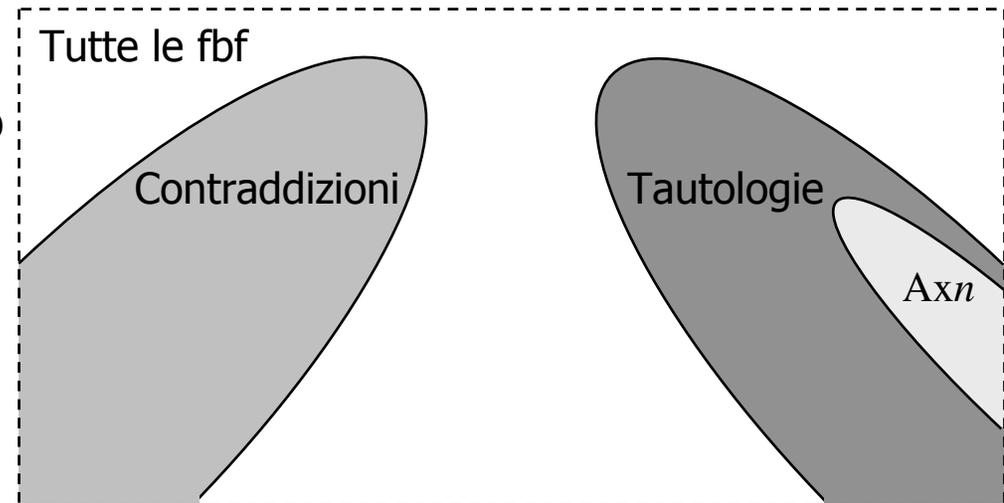
- Un qualsiasi insieme di fbf Σ può essere detto una **teoria**
- Dato un insieme di fbf Γ , l'insieme dei **teoremi** di Γ è l'insieme di tutte le fbf *derivabili* a partire da Γ
$$Teo(\Gamma) = \{\varphi : \Gamma \vdash \varphi\}$$
- Un insieme di fbf Γ è un'**assiomatizzazione** di Σ sse
$$\Sigma \equiv Teo(\Gamma)$$
- Il sistema di assiomi A_{Xn} descrive la *teoria* delle fbf *valide* del **calcolo proposizionale classico (LP)**
 - le fbf *valide* si applicano a qualsiasi 'ragionamento' in LP (sono 'leggi logiche' nel senso di leggi di LP)

LP – fbf e teorie

- Una **teoria** può essere definita tramite **assiomi**
 - In questo caso, la teoria coincide con i **teoremi** (fbf derivabili dagli assiomi)



- Nel caso di A_{xn} per LP
 - L'insieme di assiomi è infinito (esistono assiomatizzazioni finite)
 - La teoria è l'insieme delle tautologie (o fbf **valide**)



LP – Logica, teorie ed applicazioni

- In generale, l'**assiomatizzazione** di una **logica** è la descrizione concisa dell'insieme delle fbf **valide** di quella logica
 - Il sistema di assiomi A_{Xn} in LP è la descrizione dell'insieme delle *tautologie*
- La teoria delle fbf valide di una logica (p.es. LP) rappresenta l'insieme delle leggi logiche **generali** (di quella logica) e non contiene alcuna rappresentazione specifica
- Ai fini dell'applicazione a problemi pratici (non solo logici), le teorie devono rappresentare conoscenze specifiche
 - ad esempio il funzionamento di un dispositivo o di un impianto
- Esistono quindi anche **assiomatizzazioni** di **teorie** specifiche
 - cioè le applicazioni pratiche ...

LP – Esempio di teoria specifica

- “La macchina non parte”, proposizioni atomiche:

a : “La batteria è carica”
 u : “Le luci si accendono”
 r : “L’autoradio funziona”
 m : “Il motorino d’avviamento gira”
 g : “Il motorino d’avviamento è guasto”
 p : “Il motore parte”

- Assiomi della Teoria K

AxK_1 : $\neg a \rightarrow (\neg u \wedge \neg r \wedge \neg m)$

AxK_2 : $g \rightarrow \neg m$

AxK_3 : $\neg m \rightarrow \neg p$

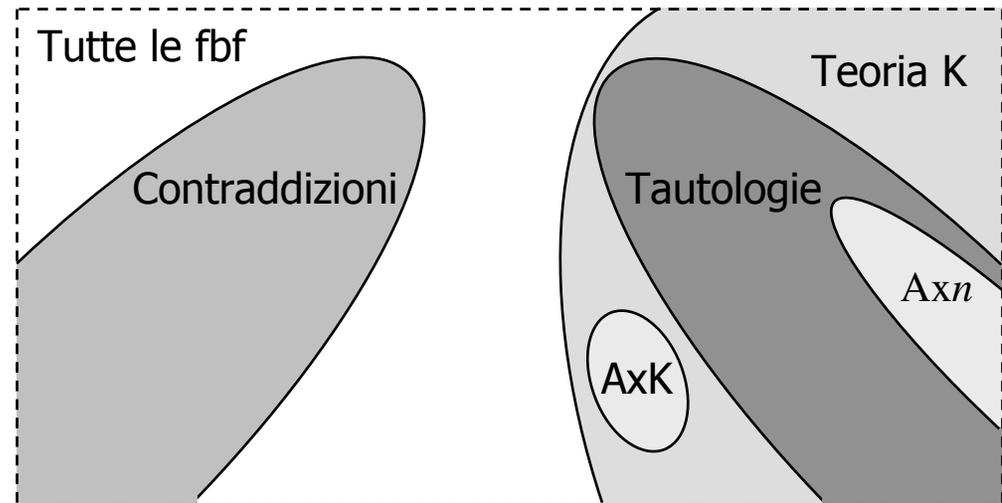
- Esempi di derivazione

$\neg a \vdash \neg p$ applicando AxK_1 e AxK_3
 $g \vdash \neg p$ applicando AxK_2 e AxK_3

LP – Tautologie e teorie specifiche

- La Teoria K è definita dagli assiomi AxK
 - coincide con l'insieme di fbf derivabili da AxK
 - l'assiomatizzazione è finita, la teoria no
- La Teoria K include Axn e tutte le tautologie (o fbf valide)
 - si rammenti la definizione di derivazione
 - non può contenere contraddizioni
 - altrimenti include *tutte* le fbf

- Tutte le teorie specifiche si comportano come K



LP – Decidibilità e derivazione

- Un sistema logico è detto *decidibile* se esiste un algoritmo di validità generale per stabilire se
$$\Gamma \models \varphi$$
- In LP ciò equivale a dire (per la completezza)
$$\Gamma \vdash \varphi$$
- La logica proposizionale è senz'altro decidibile
 - alla peggio, si provano tutte le 2^n possibili interpretazioni per stabilire se $\Gamma \models \varphi$
- Il procedimento di *derivazione* non è un algoritmo deterministico
 - ad ogni passo occorre scegliere 'la mossa giusta'
 - quale formula o quale assioma introdurre, derivare per *MP*
 - si tratta di una tecnica per la derivazione manuale

LP – Forme normali (1)

- **Forma normale congiuntiva (FNC)**

- Una formula in cui il connettivo \wedge appare solo al livello più esterno

$\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n$ dove le α_i sono fbf in cui si usa solo \vee e \neg
ed il \neg compare solo davanti alle lettere

- Esempio:

$(a \vee \neg b) \wedge (\neg c \vee \neg d \vee a) \wedge d$

$a, \neg b, \neg d, d, \dots$ sono **letterali**

- Qualsiasi fbf di LP può essere tradotta in una fbf equivalente in FNC
(dimostrazione sul testo)

- Algoritmo per la traduzione:

1) Si eliminano \rightarrow e \leftrightarrow in base alle regole di abbreviazione (vedi pag. 5)

2) Si muove \neg all'interno

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

3) Si distribuisce \vee

proprietà del connettivo: $((\varphi \wedge \psi) \vee \chi) \leftrightarrow ((\varphi \vee \chi) \wedge (\psi \vee \chi))$

LP – Forme normali (2)

- **Forma normale congiuntiva (FNC)**

- Esempio:

$$a \rightarrow \neg(b \vee c)$$

$$\neg a \vee \neg(b \vee c) \quad (\text{eliminazione di } \rightarrow)$$

$$\neg a \vee (\neg b \wedge \neg c) \quad (\text{De Morgan})$$

$$(\neg a \vee \neg b) \wedge (\neg a \vee \neg c) \quad (\text{distribuzione di } \wedge)$$

- **Forma normale disgiuntiva (FND)**

- Una formula in cui il connettivo \vee appare solo al livello più esterno

$$\alpha_1 \vee \alpha_2 \vee \dots \vee \alpha_n \quad \text{dove le } \alpha_i \text{ sono fbf in cui si usa solo } \wedge \text{ e } \neg$$

ed il \neg compare solo davanti alle lettere

- Qualsiasi fbf di LP può essere tradotta in una fbf equivalente in FND

LP – Forma a clausole

- **Forma a clausole (FC)**

- Un qualunque insieme di fbf Γ in cui le fbf sono formate solo da **disgiunzioni di letterali**

$\alpha_1 \vee \alpha_2 \vee \dots \vee \alpha_n$ dove le α_i sono **letterali** tipo $a, \neg a$

- Esempio:

$\Gamma = \{a \vee \neg b, \neg c \vee \neg d \vee a, d\}$

si può scrivere anche $\Gamma = \{\{a, \neg b\}, \{\neg c, \neg d, a\}, \{d\}\}$

- Qualunque insieme di fbf Γ di LP può essere tradotto in un'equivalente FC (dimostrazione sul testo)
- Algoritmo, a partire da un Γ qualsiasi:
 - 1) Si traduce ciascuna le fbf di Γ in FNC
si otterranno fbf del tipo $\beta_1 \wedge \beta_2 \wedge \dots \wedge \beta_n$
dove ogni β_i è una disgiunzione di letterali
 - 2) Si 'spezza' ciascuna fbf in FNC in un insieme di fbf disgiuntive
da $\beta_1 \wedge \beta_2 \wedge \dots \wedge \beta_n$ si ottiene $\{\beta_1, \beta_2, \dots, \beta_n\}$
 - 3) L'insieme Γ' (in FC) è l'unione degli insiemi di fbf così ottenute

LP – Inferenza per risoluzione

- Regola di inferenza per **risoluzione**

- a partire da due formule $\varphi \vee \chi$ e $\neg\chi \vee \psi$ si deriva $\varphi \vee \psi$
- sulla base del fatto che $\varphi \vee \chi, \neg\chi \vee \psi \models \varphi \vee \psi$

$$\frac{\varphi \vee \chi}{\neg\chi \vee \psi} \quad \varphi \vee \psi$$

↑
risolvente

Si può
verificare
direttamente

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

- Il *modus ponens* è un caso particolare di *risoluzione*

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

LP - Derivazione per risoluzione

- **Derivazione per risoluzione:**
 - Procedimento per stabilire se $\Gamma \vdash \varphi$:
 - 1) Si traduce Γ in un equivalente Γ' in FC
 - 2) Si traduce φ in un equivalente φ' in FC
 - 3) Si applica a Γ' in modo esaustivo la regola di risoluzione *in forma ristretta* (cioè si risolve **un letterale alla volta**: $\varphi \vee a, \neg a \vee \psi \vdash \varphi \vee \psi$)
 - 4) al termine si ottiene un insieme Σ (di fbf in FC), dove $\Sigma \supseteq \Gamma'$
 - 5) Se $\varphi' \subseteq \Sigma$, allora $\Gamma \vdash \varphi$
- **Vantaggi:**
 - esiste un'unica operazione di derivazione (la risoluzione)
 - non necessita di *assiomi logici*
- **Purtroppo:**
 - questo metodo è *corretto* ma non è *completo*
 - non è possibile derivare tutte le conseguenze logiche di un insieme di premesse

LP – Refutazione e risoluzione

- Conseguenza logica e insoddisfacibilità

$\Gamma \models \varphi$ sse $\Gamma \cup \{\neg\varphi\}$ è **insoddisfacibile**

| φ | ψ | a | $\varphi \vee a$ | $\neg a \vee \psi$ | $\neg(\varphi \vee \psi)$ |
|-----------|--------|-----|------------------|--------------------|---------------------------|
| 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 |

- Una **clausola vuota** è testimone di **insoddisfacibilità** per $\Gamma \cup \{\neg\varphi\}$
 - da $\{\{\neg a\}, \{a\}\}$ si deriva per risoluzione $\{ \}$ (i.e. una contraddizione)
- **Risoluzione per refutazione**
 - dovendo dimostrare $\Gamma \models \varphi$
 - si parte da $\Gamma \cup \{\neg\varphi\}$ tradotte in FC
 - e si cerca di derivare $\{ \}$ (insoddisfacibilità)

LP – Risoluzione come algoritmo

- Il metodo della risoluzione per refutazione:
 - è corretto
 - è anche completo (si può derivare qualsiasi conseguenza logica)
- Come algoritmo:
 - è sempre terminante (in LP)
 - infatti, dato un generico problema $\Gamma \cup \{\neg\varphi\}$
 - a) si deriva la clausola vuota $\{ \}$
 - b) oppure l'algoritmo si arresta quando non sono possibili ulteriori risoluzioni
 - ha una complessità esponenziale $O(2^n)$ dove n è il numero delle *lettere*

LP – Clausole di Horn

- Una **clausola di Horn**
 - è una fbf in cui compaiono solo \vee e \neg
 - in cui si ha **al massimo un letterale** in forma positiva
- Tre tipi di clausole di Horn:
 - singoli letterali (o **fatti**): $a, \neg b, c$
 - implicazioni (o **regole**): $(a \wedge b \wedge \dots) \rightarrow c$ cioè $\neg a \vee \neg b \vee \dots \vee c$
 - obiettivi o **goal**: $(c \wedge d \wedge \dots)$, la cui forma negata è $\neg c \vee \neg d \vee \dots$
- Tecnica generale
 - si esprimono le premesse come *fatti* e *regole*
 - si definisce il risultato atteso (“Giorgio è contento”?) come *goal*
 - si applica la tecnica della *risoluzione per refutazione*
- Limitazioni e vantaggi
 - non tutte i problemi LP sono esprimibili come clausole di Horn
 - ma molti problemi pratici lo sono
 - la risoluzione diretta ha complessità lineare $O(n)$

LP – Metodo a tableau

- Anche detto tableau semantici o a “tableaux”
- E` un metodo basato sulla refutazione:
 - per stabilire che $\Gamma \models \varphi$ si tenta di mostrare che $\Gamma \cup \{\neg\varphi\}$ è inconsistente
- Caratteristiche generali:
 - l’insieme $\Gamma \cup \{\neg\varphi\}$ viene completamente espanso in forma di *albero*
 - si usano diverse regole di inferenza (9 per LP) per l’espansione
 - ogni nodo dell’albero rappresenta una **congiunzione** di fbf
 - ogni biforcazione rappresenta una disgiunzione
 - un ramo è chiuso non appena si incontra una contraddizione $\{\varphi, \neg\varphi\}$
 - $\Gamma \cup \{\neg\varphi\}$ è inconsistente (o insoddisfacibile) se tutti i rami dell’albero sono **chiusi**
 - $\Gamma \cup \{\neg\varphi\}$ non è inconsistente se almeno un ramo che non si chiude
 - non è necessario che $\Gamma \cup \{\neg\varphi\}$ sia espresso in una forma normale

LP – Regole di inferenza per i tableau

- Regole alfa (o di espansione)

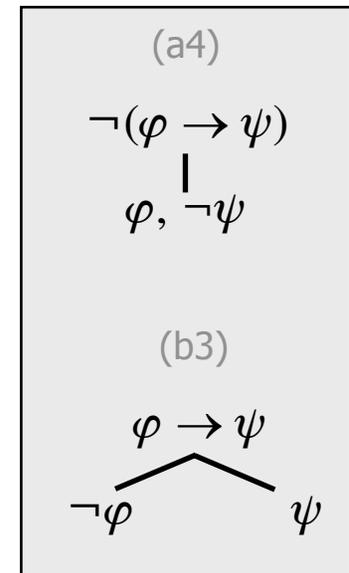
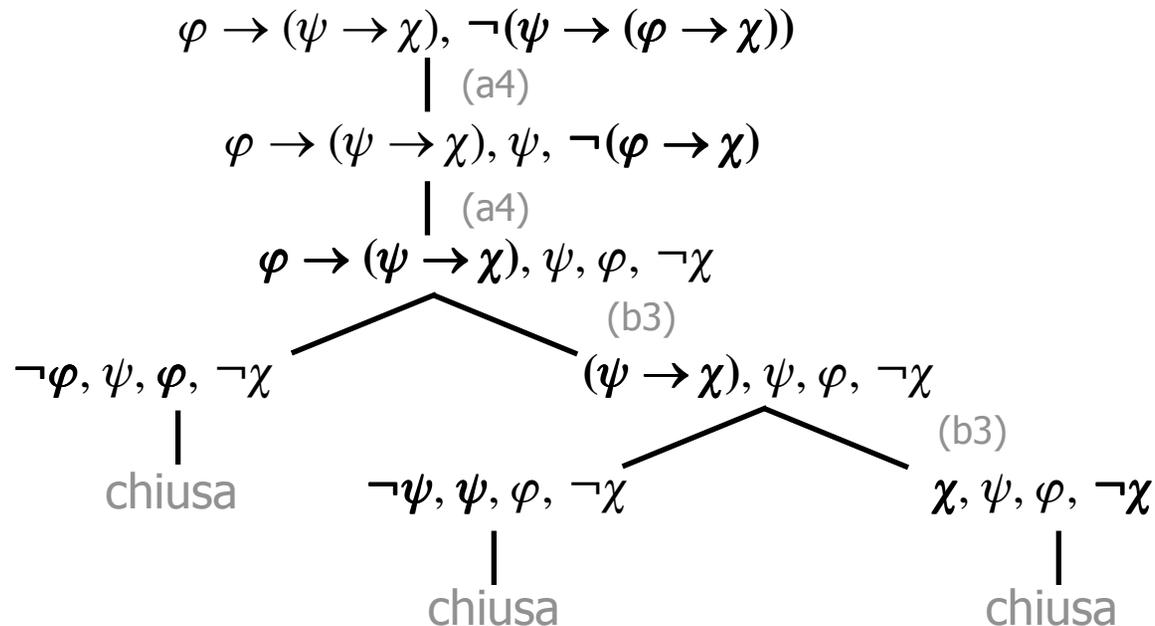
| | | | |
|---------------------|-----------------------|---------------------------|----------------------------------|
| (a1) | (a2) | (a3) | (a4) |
| $\neg(\neg\varphi)$ | $\varphi \wedge \psi$ | $\neg(\varphi \vee \psi)$ | $\neg(\varphi \rightarrow \psi)$ |
| | | | |
| φ | φ, ψ | $\neg\varphi, \neg\psi$ | $\varphi, \neg\psi$ |

- Regole beta (o di biforcazione)

| | | | | |
|----------------------|------------------------------|----------------------------|---|---|
| (b1) | (b2) | (b3) | (b4) | (b5) |
| $\varphi \vee \psi$ | $\neg(\varphi \wedge \psi)$ | $\varphi \rightarrow \psi$ | $\varphi \leftrightarrow \psi$ | $\neg(\varphi \leftrightarrow \psi)$ |
| / \ | / \ | / \ | / \ | / \ |
| $\varphi \quad \psi$ | $\neg\varphi \quad \neg\psi$ | $\neg\varphi \quad \psi$ | $\neg\varphi, \neg\psi \quad \varphi, \psi$ | $\neg\varphi, \psi \quad \varphi, \neg\psi$ |

LP – Esempio del metodo a tableau

- Obiettivo
 - stabilire che $\varphi \rightarrow (\psi \rightarrow \chi) \models \psi \rightarrow (\varphi \rightarrow \chi)$
- Espansione



LP – Algoritmo del metodo a tableau

- Procedura per $\Gamma \models \varphi$
 - 1) Definire $\Gamma \cup \{\neg\varphi\}$ come nodo (*tableau*) iniziale
 - 2) Agire su ciascun ramo, in modalità *depth first*
 - 3) Se il nodo contiene solo letterali, allora se il nodo contiene una contraddizione, chiudere altrimenti il procedimento è fallito, uscire
 - 4) Se il nodo contiene formule composite:
 - a) applicare le regole alfa
 - b) applicare le regole beta
- Notare che le regole alfa vengono applicate prima delle regole beta

LP – Caratteristiche del metodo a tableau

- Efficienza
 - anche se la complessità *worst case* è sempre $O(2^n)$
 - l'efficienza del metodo dipende in generale dalla struttura delle fbf, più che dal numero di letterali
 - in molti casi pratici, il metodo a tableau è estremamente efficace
- Leggibilità
 - si applica a fbf in forma qualsiasi
 - non richiede assiomi logici
- Generalità
 - modificando opportunamente le regole di inferenza si possono costruire versioni del metodo anche per altre logiche (anche non classiche, come le logiche modali e multivalenti)

LP – Logica classica ed applicazioni

- Ipotesi del mondo aperto (*open world assumption* – OWA)
 - vale in logica classica
 - dati Γ e φ qualsiasi, si hanno tre casi possibili
 - 1) $\Gamma \models \varphi$
 - 2) $\Gamma \models \neg\varphi$
 - 3) $\Gamma \not\models \varphi, \Gamma \not\models \neg\varphi$
- Ipotesi del mondo chiuso (*closed world assumption* – CWA)
 - si utilizza la regola aggiuntiva:

$$\frac{\Gamma \not\models \varphi}{\varphi}$$
 - **non è più logica classica**
infatti: $\Gamma \subseteq \Sigma, \Gamma \vdash \varphi \not\Rightarrow \Sigma \vdash \varphi$
(**non** monotonia)
- Negazione come fallimento (*negation as failure* – NAF)
 - una regola aggiuntiva ancor più debole:

$$\frac{\Gamma \not\vdash \varphi}{\varphi}$$
 - Dove \vdash indica la derivabilità con un algoritmo effettivo, non necessariamente completo (si pensi alla congettura $P \neq NP \dots$)