



## **Riassunto Logica Matematica**

Alessandro Gerotto

Università degli studi di Udine

Insegnante *Alberto Marcone*

Anno accademico 2021-22

<b>Cap. 1: Sintassi</b>	<b>6</b>
Convenzioni di lettura $\neg, \wedge, \vee, \rightarrow$	6
Esempio	6
<b>Grado di una formula</b>	<b>6</b>
<b>Sottoformule e sottoformule proprie</b>	<b>6</b>
Esempio	6
<b>Cap. 2: Valutazioni ed interpretazioni</b>	<b>7</b>
Esempio	7
Lemma	7
<b>Soddisfazione di una formula</b>	<b>7</b>
<b>Equivalenza tra due formule <math>\equiv</math></b>	<b>7</b>
<b>Conseguenza logica tra due formule <math>\models</math></b>	<b>7</b>
<b>Formule logicamente equivalenti</b>	<b>8</b>
Lemma	8
Lemma	8
Lemma	8
Lemma	8
<b>Validità, soddisfacibilità e insoddisfacibilità</b>	<b>8</b>
Esempio	8
<b>Algoritmo delle tavole di verità</b>	<b>9</b>
Esempio	9
Esempio	9
<b>Cap. 3: Forma normale congiuntiva e disgiuntiva</b>	<b>10</b>
Letterale e coppia di letterali	10
Lemma	10
Forma normale congiuntiva CNF	10
Esempi di formule in CNF	10
Forma normale disgiuntiva FND	10
Esempi di formule in FND	11
Teorema	11
<b>Doppie negazioni, <math>\alpha</math>-formule e <math>\beta</math>-formule</b>	<b>11</b>
Lemma	11
L'algoritmo di Fitting per la trasformazione in forma normale congiuntiva	11
L'algoritmo di Fitting per la trasformazione in forma normale disgiuntiva	12
<b>Terminazione forte degli algoritmi di Fitting</b>	<b>12</b>
Rango di una formula	12
Lemma	12
<b>Cap. 4: Il metodo dei tableaux</b>	<b>12</b>
<b>Algoritmo</b>	<b>12</b>
Esempio 1	13
Esempio 2	13
<b>Terminazione forte dei tableaux</b>	<b>13</b>
Lemma	13
Teorema	14
<b>Correttezza e completezza del metodo dei tableaux</b>	<b>14</b>
Teorema (tableau chiuso)	14
Teorema (tableau aperto)	14
Corollario	14
<b>Insiemi d'Hintikka</b>	<b>14</b>
Lemma	14
Lemma	14

Lemma	14
Esempio	14
<b>Semplificare i tableaux</b>	<b>15</b>
<b>Tableaux e conseguenza logica</b>	<b>15</b>
<b>Cap. 5: La deduzione naturale proposizionale</b>	<b>16</b>
<b>Caratteristiche e proprietà di un sistema deduttivo</b>	<b>16</b>
La correttezza	16
La completezza	16
La compatibilità (regola di taglio)	16
Uso delle regole	16
AND logico	16
OR logico	17
Implicazione	17
Esempio	17
Negazione logica	17
Regole più complesse	18
Combinazione di regole	19
La deduzione naturale proposizionale	19
Esempio	19
Lo scaricamento	20
Esempio	21
Esempio	21
Esempio	22
Le regole della deduzione naturale proposizionale	23
<b>Cap. 6: Sintassi</b>	<b>24</b>
<b>Alfabeto</b>	<b>24</b>
Esempio	24
Esempio	24
Esempio	24
<b>Termini</b>	<b>24</b>
Esempi	25
Esempio	25
<b>Sostituzione di simboli</b>	<b>25</b>
Esempio	25
<b>Formule predicative</b>	<b>25</b>
Formule atomiche	25
Formule di L	25
<b>Convenzioni di lettura <math>\neg, \wedge, \vee, \rightarrow, \forall, \exists</math></b>	<b>26</b>
<b>Grado di una formula</b>	<b>26</b>
<b>Variabili libere o legate ed enunciati</b>	<b>26</b>
Esempio	26
Esempio	26
<b>Letterali</b>	<b>26</b>
<b>Chiusura universale</b>	<b>27</b>
<b>Sottoformule</b>	<b>27</b>
Esempio	27
<b>Sostituzioni di formule</b>	<b>27</b>
Esempio	27
<b>Linguaggi con uguaglianza</b>	<b>27</b>
<b>Cap. 7: Semantica della logica predicativa</b>	<b>28</b>
<b>Interpretazioni I</b>	<b>28</b>
Esempio	28

<b>Lemma</b>	<b>28</b>
<b>Soddisfazione <math>\models</math></b>	<b>28</b>
Esempio	29
<b>Lemma</b>	<b>29</b>
<b>Corollario</b>	<b>30</b>
<b>Esercizio</b>	<b>30</b>
<b>Equivalenza logica <math>\equiv</math> e conseguenza logica <math>\models</math></b>	<b>30</b>
Nota	30
<b>Lemma di chiusura universale ed esistenziale</b>	<b>30</b>
Esercizio	30
<b>Lemma</b>	<b>31</b>
<b>Lemma</b>	<b>31</b>
<b>Validità e soddisfacibilità</b>	<b>31</b>
Esempi di formule valide, soddisfacibili e non soddisfacibili	31
Esercizio (20.14.1.mp4)	31
<b>Lemma di sostituzione...</b>	<b>32</b>
...per termini	32
...per formule	32
<b>Lemma</b>	<b>32</b>
<b>Corollario</b>	<b>32</b>
<b>Lemma</b>	<b>32</b>
<b>Corollario</b>	<b>32</b>
Logica con uguaglianza Eq	32
Lemma	32
Conseguenza ed equivalenza logica nella logica con uguaglianza	32
Validità, soddisfacibilità e insoddisfacibilità nella logica con uguaglianza	33
Esercizio 20.14.2. mp4	33
Esercizio 20.14.2. mp4	33
<b>Cap. 8: Trasformazione in forma prenessa</b>	<b>33</b>
Teorema	34
L'algoritmo	34
Nota	35
Il p-grado	35
Esempio di calcolo del p-grado	35
Lemma	35
Lemma	35
L'algoritmo per la forma prenessa	35
Lemma	36
Esempio	36
Usare il minimo numero di quantificatori	36
<b>Cap. 9: Traduzioni dal linguaggio naturale</b>	<b>36</b>
Nota:	37
<b>Cap. 10: Interpretazioni elementarmente equivalenti</b>	<b>38</b>
Esempio	38
Omomorfismo	38
Isomorfismo	38
Esempio	38
Lemma	39
Lemma	39
Lemma	39
Teorema	39
Esempio	39

Relazioni di congruenza	39
Esempio	40
<b>L'algoritmo dei tableaux nel caso predicativo</b>	<b>40</b>
Differenza con il caso proposizionale	40
Doppie negazioni, -formule, -formule, -formule, -formula	40
Esempi di domande	41
<b>Cap. 12: La deduzione naturale predicativa</b>	<b>42</b>
La deduzione naturale e i quantificatori	42
Eliminazione $\forall$ ( $\forall e$ )	42
Introduzione $\exists$ ( $\exists i$ )	42
Esempio	42
Introduzione $\forall$ ( $\forall i$ )	43
Esempio	43
Eliminazione $\exists$ ( $\exists e$ )	44
Esempio	44
Riassunto delle regole	44
Generalizzazione di ( $\forall i$ ) e ( $\exists e$ )	45
Esempio	45

## Cap. 1: Sintassi

Un **linguaggio proposizionale** è composto da **lettere proposizionali** e **formule** le cui saranno costruite utilizzando gli **elementi di P** e i **simboli logici proposizionali**  $\neg, \wedge, \vee, \rightarrow, (, )$ .

L'insieme delle **formule proposizionali** è definito per ricorsione come segue:

- se  $p \in P$  allora  $p$  è una formula;
- se  $F$  è una formula allora  $(\neg F)$  è una formula;
- se  $F$  e  $G$  sono formule allora  $(F \wedge G)$  o  $(F \vee G)$  o  $(F \rightarrow G)$  è una formula;

Ogni formula proposizionale è di **uno e uno solo** dei seguenti cinque **tipi**:

- lettera proposizionale ( $p, q, r, s, \dots$ );
- negazione ( $\neg F$ );
- congiunzione ( $F \wedge G$ );
- disgiunzione ( $F \vee G$ );
- implicazione ( $F \rightarrow G$ ).

**Convenzioni di lettura**  $\neg, \wedge, \vee, \rightarrow$

Nella scrittura delle formule adotteremo le seguenti convenzioni:

1. si omettono le parentesi più esterne;
2.  $\neg$  ha la precedenza su  $\wedge, \vee$  e  $\rightarrow$ ;
3.  $\wedge$  e  $\vee$  hanno la precedenza su  $\rightarrow$ ;
4. ulteriori simboli si appoggiano a sinistra.

### Esempio

la formula  $(\neg((\neg p) \rightarrow (q \wedge (\neg r))))$  viene abbreviata da  $\neg(\neg p \rightarrow q \wedge \neg r)$ .

### Grado di una formula

Il grado della formula  $F$ , indicato con  $g(F)$ , è definito da:

- $g(F) = 0$  se  $F$  è una lettera proposizionale;
- $g(\neg F) = g(F) + 1$ ;
- $g(F \wedge G) = g(F \vee G) = g(F \rightarrow G) = g(F) + g(G) + 1$ .

### Sottoformule e sottoformule proprie

Se  $F$  è una formula, diciamo che  $G$  è una **sottoformula** di  $F$  se  $G$  è una formula che è una **sottostringa** di  $F$ .  $G$  è una **sottoformula propria** di  $F$  se è diversa da  $F$ .

Le sottoformule di  $F$  sono:

- se  $F$  è una lettera proposizionale,  $F$  è la sua unica sottoformula;
- se  $F$  è  $\neg G$  allora le sottoformule di  $F$  sono le sottoformule di  $G$  e  $F$  stessa;
- se  $F$  è  $G \wedge H, G \vee H$  oppure  $G \rightarrow H$  allora le sottoformule di  $F$  sono le sottoformule di  $G$ , di  $H$  e di  $F$  stessa.

### Esempio

Se  $F$  è  $p \rightarrow q \vee \neg r$ , allora  $q \vee \neg r$  è una sottoformula di  $F$  (attenzione,  $p \rightarrow q$  non è una sottoformula!)

## Cap. 2: Valutazioni ed interpretazioni

Una **valutazione** per il linguaggio proposizionale  $P$  è una **funzione**  $v: P \rightarrow \{V, F\}$  che associa ad ogni **lettera proposizionale** un **valore di verità**.

Sia  $v: P \rightarrow \{V, F\}$  una valutazione. L'interpretazione  $v$  **associa** ad ogni formula  $F$  un **valore di verità** ed è definita nel modo seguente:

$G$	$H$	$G \wedge H$	$G \vee H$	$G \rightarrow H$
V	V	V	V	V
V	F	F	V	F
F	V	F	V	V
F	F	F	F	V

### Esempio

Consideriamo la valutazione  $v$  definita da  $v(p) = \text{FALSO}$ ,  $v(q) = \text{VERO}$ ,  $v(r) = \text{VERO}$  e supponiamo di voler calcolare  $v(F)$  dove  $F$  è  $\neg(\neg p \rightarrow q \wedge \neg r)$ .

Applicando le varie parti della definizione di interpretazione troveremo:

$$\neg(\neg p \rightarrow q \wedge \text{FALSO}), \neg(\neg p \rightarrow \text{FALSO}), \neg(\text{VERO} \rightarrow \text{FALSO}), \neg(\text{FALSO}), \text{VERO}$$

La formula  $F$  risulta dunque essere vera nell'interpretazione  $v$ .

### Lemma

Sia  $F$  una formula e sia  $P' \subseteq P$  l'insieme delle lettere proposizionali che compaiono in  $F$ . Siano  $v_1$  e  $v_2$  due valutazioni tali che  $v_1(p) = v_2(p)$  per ogni  $p \in P'$ . Allora  $v_1(F) = v_2(F)$ .

### Soddisfazione di una formula

Se  $v$  è un'interpretazione e  $F$  una formula, diciamo che  $v$  **soddisfa**  $F$  se  $v(F) = \text{VERO}$ . Se  $T$  è un insieme di formule, diciamo che  $v$  **soddisfa**  $T$  se  $v$  soddisfa ogni formula di  $T$ .

### Equivalenza tra due formule $\equiv$

Diciamo che le formule  $F$  e  $G$  sono **logicamente equivalenti** (in simboli  $F \equiv G$ ) se per ogni interpretazione  $v$  si ha  $v(F) = v(G)$  (quindi o tutte due false o tutte due vere).

**A.e.** Siano  $F = p \rightarrow \neg p$  e  $G = \neg p$  due formule. Ci sono due possibilità:

- se  $v(p) = \text{VERO}$  allora si verifica che  $v(F) = v(G) = \text{FALSO}$ ;
- se  $v(p) = \text{FALSO}$  invece  $v(F) = v(G) = \text{VERO}$ .

In ogni caso  $v(p) = v(G)$ , e perciò  $F \equiv G$ .

### Conseguenza logica tra due formule $\models$

Siano  $F$  e  $G$  due formule. Diciamo che  $G$  è **conseguenza logica** di  $F$  (in simboli  $F \models G$ ) se **ogni interpretazione che soddisfa  $F$  soddisfa anche  $G$** .

**A.e.** Siano  $F = (p \rightarrow q) \wedge p$  e  $G = p \vee q$  due formule.  $F \models G$ : infatti se  $v$  è un'interpretazione che soddisfa  $F$  allora in particolare deve essere  $v(p) = \text{VERO}$ , da cui segue che  $v$  soddisfa  $G$ .

$G \not\models F$ : si consideri ad esempio  $v$  tale che  $v(p) = F$  e  $v(q) = V$ ;  $v$  soddisfa  $G$  ma non soddisfa  $F$ .

L'esempio precedente evidenzia come per mostrare che  $G \models F$  sia sufficiente trovare una singola interpretazione che soddisfa  $G$  e non soddisfa  $F$ . Invece per mostrare che  $F \models G$  è necessario considerare tutte le interpretazioni che soddisfano  $F$ .

## Formule logicamente equivalenti

### Lemma

Due formule  $F$  e  $G$  sono **logicamente equivalenti** sse  $F \models G$  e  $G \models F$ .

### Lemma

Se  $F$  e  $G$  sono formule allora:

- 1)  $F \equiv \neg\neg F$ ;
- 2)  $F \wedge G \equiv G \wedge F$ ;
- 3)  $F \vee G \equiv G \vee F$ ;
- 4)  $F \wedge F \equiv F$ ;
- 5)  $F \vee F \equiv F$ ;
- 6)  $F \models F \vee G$  e  $G \models F \vee G$ ;
- 7)  $F \wedge G \models F$  e  $F \wedge G \models G$ ;
- 8)  $G \models F \rightarrow G$  e  $\neg F \models F \rightarrow G$

Quasi tutte le future affermazioni sulle conseguenze ed equivalenze logiche saranno schemi di questo tipo.

### Lemma

Fissiamo le formule logicamente equivalenti  $F$  e  $G$ . Se  $F$  è una sottoformula di  $H$  e  $F \equiv G$  (cioè  $v(F)=v(G)$ ) allora  $H \equiv H'$  (cioè  $v(H)=v(H')$ ) dove  $H'$  è la formula ottenuta da  $H$  rimpiazzando  $F$  con  $G$ .

Ossia, in altri termini: sostituendo dentro una formula una sottoformula con una formula ad essa equivalente, si ottiene una formula equivalente a quella di partenza.

### Lemma

Se  $F$ ,  $G$  e  $H$  sono formule allora:

- 1)  $\neg(F \wedge G) \equiv \neg F \vee \neg G$  (De Morgan)
- 2)  $\neg(F \vee G) \equiv \neg F \wedge \neg G$  (De Morgan)
- 3)  $F \rightarrow G \equiv \neg F \vee G$ ; (Eliminare implicazione a favore della disgiunzione)
- 4)  $\neg(F \rightarrow G) \equiv F \wedge \neg G$ ; (Eliminare implicazione a favore della disgiunzione)
- 5)  $F \rightarrow G \equiv \neg G \rightarrow \neg F$ ; (equivalenza tra un'implicazione e la sua contrappositiva)
- 6)  $(F \wedge G) \wedge H \equiv F \wedge (G \wedge H)$ ; (associatività di congiunzione)
- 7)  $(F \vee G) \vee H \equiv F \vee (G \vee H)$ ; (associatività di disgiunzione)
- 8)  $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$ ; (distributivà della congiunzione rispetto alla disgiunzione)
- 9)  $(G \vee H) \wedge F \equiv (G \wedge F) \vee (H \wedge F)$ ; (distributivà della congiunzione rispetto alla disgiunzione)
- 10)  $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$ ; (distributivà della disgiunzione rispetto alla congiunzione)
- 11)  $(G \wedge H) \vee F \equiv (G \vee F) \wedge (H \vee F)$ . (distributivà della disgiunzione rispetto alla congiunzione)

## Validità, soddisfacibilità e insoddisfacibilità

Se  $F$  è una formula:

- $F$  è **valida** se  $F$  è vera in ogni sua interpretazione.  $F$  è valida sse  $\neg F$  è insoddisfacibile;
  - $F \models G$  se e solo se  $F \rightarrow G$  valida.
  - $F \models G$  se e solo se  $F \wedge \neg G$  soddisfacibile.
- $F$  è **soddisfacibile** se esiste una valutazione  $v$  tale che  $v(A) = \text{VERO}$ ;
- $F$  è **insoddisfacibile** se **non esiste** un'interpretazione che soddisfa  $F$ .  $F$  è insoddisfacibile sse  $\neg F$  è valida;

### Esempio

Verifichiamo che la formula  $(p \rightarrow q) \wedge (q \rightarrow \neg p)$ , che indicheremo con  $F$ , è soddisfacibile:



Basta trovare un'interpretazione  $v$  che soddisfa  $F$ . Ponendo  $v(p) = \text{FALSO}$  e  $v(q) = \text{VERO}$  è facile verificare che vale  $v(F) = \text{VERO}$ .

### Algoritmo delle tavole di verità

- 1) Prende in input una formula  $F$  e esamina le sue lettere;
- 2) Crea una tabella con  $n + 1$  colonne (dove  $n = \# \text{lettere in } F \text{ e } 1 \text{ per } F$ ) e  $2^n$  linee;
- 3) Scompone nelle varie sottoformule e ne fa la valutazione;
- 4) Riassembra  $F$ .

La colonna corrispondente a  $F$  contiene VERO o FALSO e permette di stabilire:

- $F$  **valida** se contiene **solo VERO**;
- $F$  **soddisfacibile** se contiene **almeno un VERO**;
- $F$  **insoddisfacibile** se contiene **solo F**.

### Esempio

Verificare che la formula  $(p \rightarrow q) \vee (p \rightarrow \neg q)$  è valida.

$p$	$q$	$p \rightarrow q$	$\neg q$	$p \rightarrow \neg q$	$(p \rightarrow q) \vee (p \rightarrow \neg q)$
V	V	V	F	F	V
V	F	F	V	V	V
F	V	V	F	V	V
F	F	V	V	V	V

Come si nota l'ultima colonna è composta interamente da VERO.

### Esempio

Verificare che la formula  $(\neg p \vee q) \wedge (q \rightarrow \neg r \wedge \neg p) \wedge (p \vee r)$  è soddisfacibile.

$p$	$q$	$r$	$\neg p \vee q$	$\neg r \wedge \neg p$	$q \rightarrow \neg r \wedge \neg p$	$p \vee r$	$F$
V	V	V	V	F	F	V	F
V	V	F	V	F	F	V	F
V	F	V	F	F	V	V	F
V	F	F	F	F	V	V	F
F	V	V	V	F	F	V	F
F	V	F	V	V	V	F	F
F	F	V	V	F	V	V	F
F	F	F	V	V	V	F	F

Esiste un'interpretazione che soddisfa  $F$  e la formula è soddisfacibile.

## Cap. 3: Forma normale congiuntiva e disgiuntiva

Una delle ragioni dell'importanza delle forme normali congiuntiva e disgiuntiva è la **semplicità** con cui si può **verificare** se un'interpretazione **soddisfa o meno una formula** di questo tipo.

### Letterale e coppia di letterali

Un letterale è una **lettera proposizionale** oppure la **negazione di una lettera proposizionale**.

Se  $p$  è una lettera proposizionale  $\{p, \neg p\}$  è una **coppia complementare di letterali**. Più in generale se  $F$  è una formula  $\{F, \neg F\}$  è una coppia complementare. Diciamo che  $F$  e  $\neg F$  sono ciascuno il **complemento** dell'altro.

L'algoritmo che introdurremo, è **non deterministico** (ad ogni passo abbiamo diverse scelte possibili), ma gode della proprietà della **terminazione forte** (termina qualunque sia la successione delle scelte).

### Lemma

Un insieme di letterali è **soddisfacibile** sse **non contiene** nessuna **coppia complementare**.

### Forma normale congiuntiva CNF

Per verificare se un'interpretazione **soddisfa o meno una formula** basta verificare **se in ogni congiunto esiste almeno un letterale soddisfatto** dall'interpretazione.

Una formula proposizionale è in forma normale congiuntiva se è della forma:

$$F_1 \wedge \dots \wedge F_m$$

dove  $F_i$  è della forma:

$$G_1 \vee \dots \vee G_h$$

con  $G_i$  letterale.

### Esempi di formule in CNF

- $(\neg A \vee B) \wedge (B \vee C)$ ;
- $(A \vee B) \wedge (\neg B \vee C \vee D) \wedge (D \vee \neg E)$ ;
- $(\neg B \vee C)$ ;
- $(\neg B \vee C) \wedge A$ ;
- $A \wedge B$ .

### Forma normale disgiuntiva FND

Per verificare se un'interpretazione **soddisfa o meno una formula** basta verificare **se esiste un disgiunto** i cui **letterali sono tutti soddisfatti** dall'interpretazione.

Una formula proposizionale è in forma normale disgiuntiva se è della forma:

$$F_1 \vee \dots \vee F_m$$

dove  $F_i$  è della forma:

$$G_1 \wedge \dots \wedge G_h$$

con  $G_i$  letterale.

### Esempi di formule in FND

- $(\neg A \wedge B) \vee (B \wedge C)$ ;
- $(A \wedge B) \vee (\neg B \wedge C \wedge D) \vee (D \wedge \neg E)$ ;
- $(\neg B \wedge C)$ ;
- $(\neg B \wedge C) \vee A$ ;
- $A \wedge B$ .

### Teorema

Ogni formula proposizionale  $F$  può essere trasformata in due formule  $G_1$  e  $G_2$ , la prima in **forma normale congiuntiva** e la seconda in **forma normale disgiuntiva**, tali che:

$$F \equiv G_1 \text{ e } F \equiv G_2$$

### Doppie negazioni, $\alpha$ -formule e $\beta$ -formule

Una formula  $F$  è una **doppia negazione** se è del tipo  $\neg\neg G$  per qualche formula  $G$ . In questo caso diciamo che  $G$  è il **ridotto** di  $F$ .

Una formula è una  **$\alpha$ -formula** se esistono  $F$  e  $G$  tali che la formula è di uno dei tipi che compaiono nella colonna sinistra della seguente tabella.

$\alpha$ -formula	ridotti	
$F \wedge G$	$F$	$G$
$\neg(F \vee G)$	$\neg F$	$\neg G$
$\neg(F \rightarrow G)$	$F$	$\neg G$

Una formula è una  **$\beta$ -formula** se esistono  $F$  e  $G$  tali che la formula è di uno dei tipi che compaiono nella colonna sinistra della seguente tabella.

$\beta$ -formula	ridotti	
$F \vee G$	$F$	$G$
$\neg(F \wedge G)$	$\neg F$	$\neg G$
$F \rightarrow G$	$\neg F$	$G$

### Lemma

Ogni **doppia negazione** è logicamente equivalente al suo **ridotto**.

Ogni  **$\alpha$ -formula** è logicamente equivalente alla congiunzione dei suoi **ridotti**.

Ogni  **$\beta$ -formula** è logicamente equivalente alla disgiunzione dei suoi **ridotti**.

### L'algoritmo di Fitting per la trasformazione in forma normale congiuntiva

L'algoritmo di Fitting per la trasformazione in forma normale congiuntiva prende in input una formula proposizionale  $F$  e la considera come una congiunzione generalizzata di una disgiunzione generalizzata:  $\langle [F] \rangle$ . Ad ogni passo controlla:

- Se tutti gli **elementi sono letterali** la formula è in forma normale congiuntiva e l'**algoritmo si arresta**.
- Se esistono **elementi che non sono letterali** se ne sceglie uno, indicato con  $G$  e se  $G$  è:
  - **$G$  doppia negazione** con ridotto  $H$ : si sostituisce  $G$  con  $H$  nel congiunto in cui appariva  $G$ ;
  - **$G$   $\beta$ -formula** con ridotti sono  $G_1$  e  $G_2$ : si sostituisce  $G$  con  $G_1, G_2$  nel congiunto in cui appariva  $G$ ;
  - **$G$   $\alpha$ -formula** con ridotti sono  $G_1$  e  $G_2$ :  $\langle [\alpha, \dots] \rangle = \langle [G_1, \dots], [G_2, \dots] \rangle$ .

### L'algoritmo di Fitting per la trasformazione in forma normale disgiuntiva

L'algoritmo di Fitting per la trasformazione in forma normale disgiuntiva prende in input una formula proposizionale  $F$  e la considera come una disgiunzione generalizzata di una congiunzione generalizzata:  $[F]$ . Ad ogni passo controlla:

- Se tutti gli **elementi sono letterali** la formula è in forma normale congiuntiva e l'**algoritmo si arresta**.
- Se esistono **elementi che non sono letterali** se ne sceglie uno, indicato con  $G$  e se  $G$  è:
  - $G$  **doppia negazione** con ridotto  $H$ : si sostituisce  $G$  con  $H$  nel congiunto in cui appariva  $G$ ;
  - $G$   **$\alpha$ -formula** con ridotti sono  $G_1$  e  $G_2$ : si sostituisce  $G$  con  $G_1, G_2$  nel congiunto in cui appariva  $G$ ;
  - $G$   **$\beta$ -formula** con ridotti sono  $G_1$  e  $G_2$ :  $\langle [\beta, \dots] \rangle = \langle [G_1, \dots], [G_2, \dots] \rangle$ .

### Terminazione forte degli algoritmi di Fitting

#### Rango di una formula

Il **rango** è un **numero naturale positivo** associato ad ogni formula proposizionale che vale:

- se  $F$  è un **letterale** allora  $rg(F) = 1$ ;
- se  $F$  è una **doppia negazione** con ridotto  $G$  allora  $rg(F) = rg(G) + 1$ ;
- se  $F$  è una  **$\alpha$ -formula o una  $\beta$ -formula** con ridotti  $G$  e  $H$  allora  $rg(F) = rg(G) + rg(H) + 1$ .

#### Lemma

Gli algoritmi di fitting (CNF e FND) godono della **proprietà della terminazione forte**, cioè **terminano** qualunque sia la formula su cui si decide di operare ad ogni singolo passo.

## Cap. 4: Il metodo dei tableaux

Il metodo dei **tableaux proposizionali** è un calcolo logico che fornisce una **procedura** (non deterministica e con terminazione forte) di **decisione per la soddisfacibilità** delle formule proposizionali.

Il principio che ispira questo algoritmo è piuttosto semplice: si **cerca** un'interpretazione che **soddisfi la formula** in esame. Se la ricerca ha successo la formula sarà **soddisfacibile**, altrimenti è **insoddisfacibile**.

### Algoritmo

Un **tableau per una formula  $F$**  è un **albero**  $T_k$  in cui ogni nodo è etichettato con un insieme finito di formule. All'inizio della costruzione l'albero consiste di un solo nodo (**radice**  $T_0$ ) etichettato con  $\{F\}$ .

Il tableau è **costruito per livelli** ( $T_0 \dots T_k$ ) dove, l'ultimo di essi ( $T_k$ ) è il **tableau per  $F$** .

Se  $n$  è un **nodo** di qualche sottoalbero  $T_i$  si indica con  $E(n)$  l'**etichetta** di  $n$ . Allo stadio  $i$  cerchiamo una foglia  $n$  dell'albero  $T_i$  tale che  $E(n)$  **non sia un insieme di letterali**. Se una tale foglia:

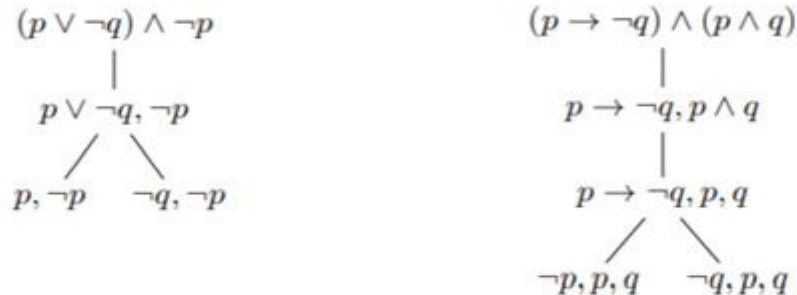
- **Non esiste**: la costruzione del tableau è **terminata** e l'algoritmo si arresta.
- **Esiste**: fissiamo  $n$  e scegliamo una formula  $G \in E(n)$  che non è un letterale. Allora  $G$  può essere:
  - Una **doppia negazione** con ridotto  $G_1$ : si aggiunge un nuovo nodo  $n'$  sotto  $n$  e si pone:  
 $E(n') = (E(n) \setminus \{G\}) \cup \{G_1\}$ ;
  - Una  **$\alpha$ -formula** con ridotti  $G_1$  e  $G_2$ : si aggiunge un nuovo nodo  $n'$  sotto  $n$  e si pone:  
 $E(n') = (E(n) \setminus \{G\}) \cup \{G_1, G_2\}$ ;
  - Una  **$\beta$ -formula** con ridotti  $G_1$  e  $G_2$ : si aggiungono due nodi  $n_1$  e  $n_2$  sotto  $n$  e si pone:  
 $E(n_1) = (E(n) \setminus \{G\}) \cup \{G_1\}$  e  $E(n_2) = (E(n) \setminus \{G\}) \cup \{G_2\}$ .

Per comodità di lettura aggiungeremo sotto le foglie del tableau uno dei simboli:

- $\otimes$ : se l'etichetta della foglia contiene una coppia complementare di letterali;
- $\odot$ : altrimenti.

### Esempio 1

Costruiamo un tableau per  $(p \vee \neg q) \wedge \neg p$  e poi per  $(p \rightarrow \neg q) \wedge (p \wedge q)$

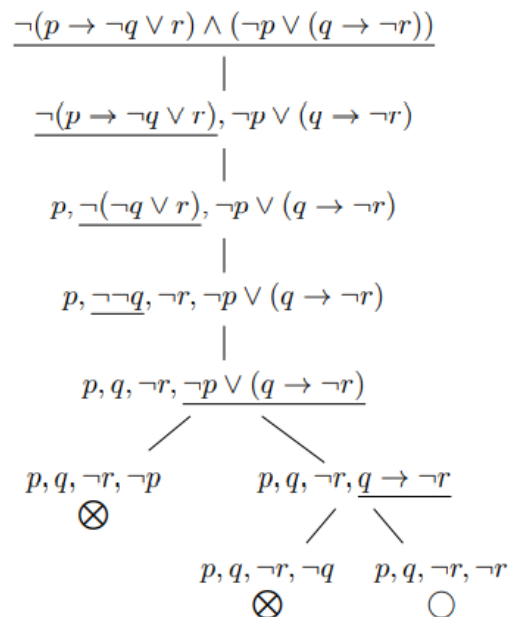


Il primo tableau contiene una foglia (quella di destra) etichettata con un insieme di letterali che **non contiene coppie complementari**: questo testimonia la **soddisfacibilit ** della formula originaria.

Il secondo tableau contiene **solo foglie etichettate con insiemi di letterali che contengono coppie complementari**: ci  implica che la formula di partenza   **insoddisfacibile**.

### Esempio 2

Costruiamo un tableau per  $\neg(p \rightarrow \neg q \vee r) \wedge (\neg p \vee (q \rightarrow \neg r))$



Il tableau contiene una foglia (quella di destra) etichettata con un **insieme di letterali che non contiene coppie complementari**: questo testimonia la **soddisfacibilit ** della formula originaria.

### Terminazione forte dei tableaux

#### Lemma

Se un albero binario   infinito allora ha un ramo infinito.

### **Teorema**

L'algoritmo di costruzione dei tableaux gode della **proprietà della terminazione forte**, cioè **termina** qualunque siano il nodo e la formula su cui si decide di operare ad ogni singolo passo.

### **Correttezza e completezza del metodo dei tableaux**

#### **Teorema (tableau chiuso)**

Sia  $F$  una formula e  $T$  un tableau per  $F$ .  $F$  è **insoddisfacibile** sse  $T$  è **chiuso** (se tutte le sue **foglie** sono etichettate con insiemi di letterali che contengono una **coppia complementare**).

#### **Teorema (tableau aperto)**

Sia  $F$  una formula e  $T$  un tableau per  $F$ .  $F$  è soddisfacibile se e solo se  $T$  è **aperto** (se non è chiuso, cioè se **almeno una foglia** è etichettata con un insieme di letterali che **non contiene una coppia complementare**). Un **ramo aperto** è un ramo che collega la **radice** dell'albero con una **foglia** etichettata con un insieme di letterali che **non contiene una coppia complementare**.

### **Corollario**

Sia  $F$  una formula e  $T$  e  $T'$  due tableaux per  $F$ .  $T$  e  $T'$  sono **entrambi chiusi** o **entrambi aperti**.

### **Insiemi d'Hintikka**

Un insieme di **Hintikka** è l'insieme di formule che sono **vere in una qualche interpretazione**  $v$ . Un **insieme di formule  $H$**  è un insieme di Hintikka se soddisfa le seguenti quattro condizioni:

- 1)  $H$  **non contiene coppie complementari** di letterali;
- 2) se  $G \in H$  è una **doppia negazione** con ridotto  $G_1$  allora  $G_1 \in H$ ;
- 3) se  $G \in H$  è una  **$\alpha$ -formula** con ridotti  $G_1$  e  $G_2$  allora  $G_1 \in H$  e  $G_2 \in H$ ;
- 4) se  $G \in H$  è una  **$\beta$ -formula** con ridotti  $G_1$  e  $G_2$  allora  $G_1 \in H$  oppure  $G_2 \in H$  (o entrambi siano in  $H$ ).

Quindi, anche un **insieme di letterali privo di coppie complementari** è un insieme di Hintikka.

### **Lemma**

Ogni insieme di Hintikka è soddisfacibile.

### **Lemma**

Se  $r$  è un ramo aperto di un tableau allora  $H = \bigcup_{n \in r} E(n)$  è un insieme di Hintikka.

### **Lemma**

Se una foglia  $f$  di un tableau per  $F$  non contiene coppie complementari allora l'interpretazione  $v$  definita da

$$v(p) = \begin{cases} \mathbb{V} & \text{se } p \in E(f); \\ \mathbb{F} & \text{se } p \notin E(f). \end{cases}$$

### **Esempio**

L'insieme  $\{p \vee \neg r \rightarrow q, \neg(p \vee \neg r), \neg p, \neg \neg r, r\}$  è un insieme di Hintikka. Infatti

- 1) Non contiene **coppie complementari** di letterali,
- 2) Contiene **il ridotto della doppia negazione**  $\neg\neg r$ .
- 3) Contiene **entrambi i ridotti della  $\alpha$ -formula**  $\neg(p \vee \neg r)$ ,
- 4) Contiene **uno dei ridotti della  $\beta$ -formula**  $p \vee \neg r \rightarrow q$ ,

### Semplificare i tableaux

- Se nell'etichetta di un nodo di un tableau compaiono una **formula G** e la **sua negazione  $\neg G$**  possiamo considerare il nodo in questione **chiuso** e **non operare più** su di esso.
- **Privilegiare** operazioni di **doppie negazioni** o  **$\alpha$ -formule** ogniquale volta ciò sia possibile per limitare il numero di diramazioni.

### Tableaux e conseguenza logica

Se  $T = \{F_1 \dots F_n\}$  è un **insieme finito di formule** allora la validità (soddisfacibilità, insoddisfacibile) di T è equivalente alla validità (soddisfacibilità, insoddisfacibilità) della formula  $F_1 \wedge \dots \wedge F_n$ . Grazie a questo lemma si può usare il metodo dei tableaux per studiare la validità o la soddisfacibilità di un insieme finito di formule  $\{F_1 \dots F_n\}$ .

Per stabilire se un insieme finito  $T = \{F_1 \dots F_n\}$  di formule è soddisfacibile costruiamo un tableau la cui radice è etichettata con T. Se il tableau è aperto T è soddisfacibile, se il tableau è chiuso T è insoddisfacibile

Per stabilire se  $F_1 \dots F_n \models G$  costruiamo un tableau la cui radice è etichettata con  $\{F_1 \dots F_n, \neg G\}$ . Se il tableau è **chiuso**  $F_1 \dots F_n \models G$ , altrimenti (tableau **aperto**)  $F_1 \dots F_n \not\models G$  (e l'etichetta priva di coppie complementari di una foglia ci permette di definire un'interpretazione che soddisfa  $F_1 \dots F_n$  ma non G).

## Cap. 5: La deduzione naturale proposizionale

La deduzione naturale è un tipo di **sistema deduttivo**. Astrattamente, un sistema deduttivo è definito come una relazione tra un insieme di formule e una singola formula. Se una formula è deducibile da un insieme di formule, indicato con  $T \vdash F$ , significa che  $F$  può essere **logicamente** derivata dall'insieme  $T$ .

### Caratteristiche e proprietà di un sistema deduttivo

Le caratteristiche essenziali che sono desiderabili in ogni sistema deduttivo includono:

#### La correttezza

La prima proprietà fondamentale che il sistema deduttivo dovrebbe possedere è quella della **correttezza** delle deduzioni. Tale proprietà serve per garantire che ciò che viene dedotto nel sistema dev'essere vero. Tale formula è:

$$(1) \text{ Se } T \vdash F \text{ allora } T \models F$$

#### La completezza

Idealmente desideriamo anche che valga l'implicazione inversa:  $T \models F$  dovrebbe implicare  $T \vdash F$ . Questa proprietà (la **completezza** di  $\vdash$ ) è assai più difficile da ottenere e da dimostrare: per raggiungerla è necessario che il sistema deduttivo rappresentato da  $\vdash$  sia sufficientemente "ricco":

$$(2) \text{ Se } T \models F \text{ allora } T \vdash F$$

#### La compatibilità (regola di taglio)

Una proprietà fondamentale che il sistema deduttivo dovrebbe possedere è quella della **compatibilità** delle deduzioni. Spesso per dimostrare un teorema  $F$  iniziamo con il dimostrare un "lemma" intermedio  $G$ , per poi utilizzare  $G$  nella dimostrazione di  $F$ . Questo principio è chiamato **regola di taglio** (nel passaggio dalle ipotesi alla conclusione "tagliamo"  $G$ ), ed è implicito nel sistema di deduzione naturale che presenteremo.

Utilizzando il nostro simbolismo, esprimiamo questa proprietà come:

$$(3) \text{ Se } T \vdash G \text{ e } T', G \vdash F \text{ allora } T, T' \vdash F$$

### Uso delle regole

Nell'**individuare le regole** che vogliamo **inserire** nel nostro sistema deduttivo (e che saranno le regole fondamentali della deduzione naturale) ci facciamo guidare dall'esigenza di **assicurare la correttezza**, ed esaminiamo i vari connettivi per isolare le regole di base che li riguardano.

#### AND logico

Dato che da  $F \wedge G$  possiamo dedurre sia  $F$  che  $G$ , mentre se supponiamo sia  $F$  che  $G$  possiamo dedurre  $F \wedge G$ , le seguenti regole appaiono del tutto naturali:

$$\begin{array}{ll} F \wedge G \vdash F & (\wedge e.1) \\ F \wedge G \vdash G & (\wedge e.2) \\ F, G \vdash F \wedge G & (\wedge i) \end{array}$$



Le lettere  $e$  e  $i$  sono state scelte perché nelle prime due righe stiamo eliminando il connettivo  $\wedge$ , mentre nell'ultima lo stiamo introducendo.

### ***OR logico***

Nel caso di  $\vee$  è evidente come possiamo introdurre il connettivo:

$$\begin{array}{l} F \vdash F \vee G \quad (\vee i.1) \\ G \vdash F \vee G \quad (\vee i.2) \end{array}$$

Non è invece immediatamente chiaro come eliminare  $\vee$ : se sappiamo che vale  $F \vee G$  non sappiamo quale dei due disgiunti è vero, e quindi sembra che non possiamo dedurre nulla. Vedremo più avanti come risolvere questo problema.

### ***Implicazione***

Nel caso di  $\rightarrow$  la regola di eliminazione è nota come **modus ponens**, e rappresenta il modo tipico di utilizzare un'implicazione:

$$F, F \rightarrow G \vdash G \quad (\rightarrow e)$$

Essa corrisponde al ragionamento per cui sapendo che se piove allora Marco non prende la bici ed osservando che piove, possiamo giungere alla conclusione che Marco non prende la bici:

$$piove, se\ piove \rightarrow Marco\ non\ prende\ la\ bici \vdash Marco\ non\ prende\ la\ bici$$

### ***Esempio***

Vediamo ora come combinando (attraverso l'uso ripetuto della regola di taglio) queste prime regole è possibile ottenere qualche ragionamento non banale.

Vogliamo mostrare che

$$F \wedge (G \rightarrow H), F \rightarrow G \vdash G \wedge H$$

ovvero che assumendo  $F \wedge (G \rightarrow H)$  e  $F \rightarrow G$  possiamo dedurre  $G \wedge H$ .

- Per  $(\wedge e.1)$  abbiamo  $F \wedge (G \rightarrow H) \vdash F$  ed abbiamo ottenuto  $F$  a partire da una delle nostre ipotesi.
- Per  $(\rightarrow e)$  abbiamo  $F, F \rightarrow G \vdash G$  ed abbiamo ottenuto  $G$  da una delle ipotesi e da una conclusione già ottenuta.
- Per  $(\wedge e.2)$  abbiamo  $F \wedge (G \rightarrow H) \vdash G \rightarrow H$
- Per  $(\rightarrow e)$  otteniamo  $G, G \rightarrow H \vdash H$ .
- Abbiamo quindi ottenuto sia  $G$  che  $H$ , e possiamo usare  $(\wedge i)$  per ottenere  $G \wedge H$ .

### ***Negazione logica***

Tornando all'analisi delle regole, abbiamo lasciato finora da parte il connettivo  $\neg$ , che è senz'altro il più delicato. Per eliminare  $\neg$  è naturale supporre di aver ottenuto sia  $F$  che  $\neg F$ , e da queste dedurre una contraddizione. La negazione è quindi strettamente connessa con la nozione di contraddizione, e perciò è conveniente introdurre una formula che rappresenti quest'ultima.

Introduciamo  $\perp$  ossia una costante logica che appartiene all'insieme  $P$  delle lettere proposizionali e viene letta "falso" o "contraddizione". La definizione implica che  $\perp$  è **insoddisfacibile** e (quindi) che  $\neg \perp$  è **valida**. Possiamo anzi considerare  $\perp$  e  $\neg \perp$  come i "prototipi" rispettivamente delle formule insoddisfacibili e delle formule valide.

Otteniamo quindi la regola:

$$F, \neg F \vdash \perp \quad (\neg\text{-e}) \text{ o } (\perp\text{-i})$$

Questa regola, può anche essere vista come la regola di introduzione di  $\perp$ , ( $\perp\text{-i}$ ). Un'altra regola riguardante la negazione è l'eliminazione della doppia negazione:

$$\neg\neg F \vdash F \quad (\neg\text{-e})$$

### **Regole più complesse**

Passiamo ora a considerare regole più complesse, in cui a partire da una deduzione di partenza si arriva ad una deduzione in cui alcune delle ipotesi sono state eliminate. La regola più semplice di questo tipo è quella che riguarda l'introduzione di  $\rightarrow$ . Per dimostrare l'implicazione  $F \rightarrow G$  è naturale assumere  $F$  e dedurre  $G$  da essa.

In altre parole, se abbiamo una deduzione di  $G$  a partire da  $F$ , allora possiamo dedurre  $F \rightarrow G$ . La correttezza di questo principio è verificata osservando che:

- se  $F$  è falsa,  $F \rightarrow G$  è certamente vera
- se  $F$  è vera la deduzione di partenza (che supponiamo corretta) assicura che  $G$  è vera e quindi anche  $F \rightarrow G$  lo è.

Utilizzando la nostra simbologia:

$$\text{se } F \vdash G \text{ allora } \vdash F \rightarrow G \quad (\rightarrow\text{-i})$$

Consideriamo ora l'introduzione di  $\neg$ : un tipico modo di dimostrare  $\neg F$  è quello di supporre  $F$  e dedurre una contraddizione. La regola che otteniamo è:

$$\text{se } F \vdash \perp \text{ allora } \vdash \neg F \quad (\neg\text{-i}) \text{ o } (\perp\text{-e})$$

Consideriamo ora l'eliminazione di  $\vee$ , che avevamo rimandato in precedenza. Se sappiamo che una formula  $H$  è deducibile sia da  $F$  che da  $G$ , allora possiamo affermare che da  $F \vee G$  possiamo dedurre  $H$  (si tratta in sostanza di una **dimostrazione per casi**). Questo ragionamento può venire riassunto dalla regola:

$$\text{se } F \vdash H \text{ e } G \vdash H \text{ allora } F \vee G \vdash H \quad (\vee\text{-e})$$

Un esempio di ragionamento che viene formalizzato da questa regola è il seguente: se dal fatto che piova possiamo dedurre che Marco non prende la bici e dal fatto che nevichi possiamo arrivare alla stessa conclusione, a partire dall'ipotesi che piove o nevica possiamo giungere alla conclusione che Marco non prende la bici (senza necessariamente sapere quale delle due ipotesi si verifichi).

*piove*  $\vdash$  *Marco non prende la bici* e *nevica*  $\vdash$  *Marco non prende la bici* allora se *piove* o *nevica*  $\vdash$  *Marco non prende la bici*

### Combinazione di regole

Le varie regole che abbiamo introdotto possono essere combinate tra di loro ottenendo nuove regole. Se ad esempio  $\neg F \vdash \perp$ , allora per ( $\neg$ i) abbiamo  $\vdash \neg\neg F$ , e per ( $\neg$ e) otteniamo  $\vdash F$ . Quindi

$$\text{se } \neg F \vdash \perp \text{ allora } \vdash F \quad (\text{RAA})$$

dove il nome della regola è l'abbreviazione del latino **reductio ad absurdum**, cioè **riduzione all'assurdo**. Questa regola riproduce le usuali dimostrazioni per assurdo o per contraddizione:

*se vogliamo dimostrare  $F$  supponiamo  $\neg F$  con l'obiettivo di raggiungere una contraddizione.*

### La deduzione naturale proposizionale

Per ogni deduzione naturale, le regole precedentemente analizzate vengono **combinare** tra loro, utilizzando la **regola di taglio** e altre tecniche descritte successivamente, per formare un albero di deduzione. A differenza del metodo dei tableaux, nell'approccio della deduzione naturale è più comune che l'albero abbia la **radice in basso**. Ogni **nodo** dell'albero è associato a una **singola formula** (in contrasto con gli insiemi di formule nei tableaux). La **radice** dell'albero è contrassegnata con il **risultato della deduzione**, mentre le **foglie** sono contrassegnate con le **ipotesi della deduzione**.

Nel contesto della deduzione naturale, utilizziamo il simbolo  $\triangleright$  come simbolo deduttivo anziché  $\vdash$ . Quindi,  $T \triangleright F$  indica che esiste una deduzione naturale in cui  $F$  è l'etichetta della radice e tutte le etichette delle foglie sono contenute nell'insieme  $T$ .

Ogni regola della deduzione naturale corrisponde all'aggiunta di un nodo sotto uno o più nodi preesistenti, seguendo un insieme specifico di regole e linee guida. La linea orizzontale è il modo tradizionale di rappresentare una regola deduttiva.

È facile dunque rappresentare ( $\wedge$ i), ( $\wedge$ e.1), ( $\wedge$ e.2), ( $\vee$ i.1), ( $\vee$ i.2), ( $\rightarrow$ e), ( $\neg$ e) e ( $\neg$ e), rispettivamente come:

$$\begin{array}{c} \frac{F \quad G}{F \wedge G} \quad \frac{\frac{F \wedge G}{F} \quad \frac{F \wedge G}{G}}{F \rightarrow G} \quad \frac{F \quad \neg F}{\perp} \quad \frac{G}{F \vee G} \quad \frac{F}{F \vee G} \\ \frac{F \quad \neg\neg F}{F} \end{array}$$

### Esempio

Vediamo ora come combinando (attraverso l'uso ripetuto della regola di taglio) queste prime regole è possibile ottenere qualche ragionamento non banale. Vogliamo mostrare che

$$F \wedge (G \rightarrow H), F \rightarrow G \vdash G \wedge H$$

ovvero che assumendo  $F \wedge (G \rightarrow H)$  e  $F \rightarrow G$  possiamo dedurre  $G \wedge H$ .

- Per ( $\wedge$ e.1) abbiamo  $F \wedge (G \rightarrow H) \vdash F$  ed abbiamo ottenuto  $F$  a partire da una delle nostre ipotesi.
- Per ( $\rightarrow$ e) abbiamo  $F, F \rightarrow G \vdash G$  ed abbiamo ottenuto  $G$  da una delle ipotesi e da una conclusione già ottenuta.
- Per ( $\wedge$ e.2) abbiamo  $F \wedge (G \rightarrow H) \vdash G \rightarrow H$
- Per ( $\rightarrow$ e) otteniamo  $G, G \rightarrow H \vdash H$ .

Abbiamo quindi ottenuto sia  $G$  che  $H$ , e possiamo usare  $(\wedge i)$  per ottenere  $G \wedge H$ . Mostriamo ora lo stesso ragionamento utilizzando la nuova notazione:

$$\frac{\frac{\frac{F \wedge (G \rightarrow H)}{F} \quad F \rightarrow G}{G} \quad \frac{\frac{\frac{F \wedge (G \rightarrow H)}{F} \quad F \rightarrow G}{G} \quad \frac{F \wedge (G \rightarrow H)}{G \rightarrow H}}{H}}{G \wedge H}$$

Abbiamo così ottenuto  $F \wedge (G \rightarrow H)$ ,  $F \rightarrow G \triangleright G \wedge H$ . Notiamo come la deduzione naturale dell'esempio sia in realtà uno **schema**. Essa vale **qualunque** siano le formule  $F$ ,  $G$  e  $H$  e quindi abbiamo mostrato ad esempio che

$$\neg p \wedge (q \rightarrow (r \rightarrow s)), \neg p \rightarrow q \triangleright q \wedge (r \rightarrow s)$$

Tutte le deduzioni naturali che presenteremo in questo capitolo (e nel capitolo 12 dedicato alla deduzione naturale per la logica predicativa) sono in effetti **schemi**, e non noteremo più esplicitamente questo fatto.

### Lo scaricamento

Per trattare le **regole condizionali**, è opportuno considerare alcune delle formule coinvolte come delle “**ipotesi ausiliarie**”. Ad esempio in  $(\rightarrow i)$ , il nostro obiettivo è dedurre  $F \rightarrow G$ , e per ottenerlo facciamo l'ipotesi ausiliaria  $F$ . Quando, utilizzando questa ipotesi ausiliaria, avremo ottenuto  $G$ , potremo eliminare (la terminologia della deduzione naturale usa l'espressione **scaricare**) l'ipotesi ausiliaria e dedurre  $F \rightarrow G$ . Per indicare lo scaricamento di  $F$  metteremo questa formula **tra parentesi quadre**. La deduzione naturale che corrisponde a  $(\rightarrow i)$  è dunque

$$\frac{\frac{T, [F]}{\nabla} \quad G}{F \rightarrow G}$$

In questa deduzione il simbolo  $\triangleright$  indica l'esistenza di una deduzione naturale che mostra  $T, F \triangleright G$ . Le etichette di foglie che sono messe tra  $[ ]$  **non sono ipotesi** della deduzione naturale. In questo modo abbiamo effettivamente ottenuto  $T \triangleright F \rightarrow G$ . Similmente  $(\vee e)$  e  $(\neg i)$  sono rappresentate dalle deduzioni naturali

$$\frac{\frac{T}{\nabla} \quad F \vee G \quad \frac{\frac{T', [F]}{\nabla} \quad H}{H} \quad \frac{T'', [G]}{\nabla} \quad H}{H}$$

$$\frac{\frac{T, [F]}{\nabla} \quad \perp}{\neg F}$$

Si noti che quando scarichiamo le ipotesi ausiliarie esse non “spariscono”, ma diventano sottoformule della conclusione (è il caso di  $(\rightarrow i)$  e  $(\neg i)$ ) o di una nuova ipotesi (nel caso di  $(\vee e)$ ).

### Esempio

Vogliamo dimostrare in deduzione naturale l'inverso della regola della doppia negazione, cioè che  $F \triangleright \neg\neg F$ . Consideriamo la deduzione naturale

$$\frac{F \quad [\neg F]^1}{\frac{\perp}{\neg\neg F} 1}$$

La prima riga della deduzione naturale è un'applicazione di ( $\neg$ e): da  $F$  e  $\neg F$  abbiamo dedotto  $\perp$ . Infine, abbiamo applicato la regola di ( $\neg$ i) per dedurre  $\neg\neg F$ , scaricando l'ipotesi  $\neg F$

Le **parentesi quadre** intorno a  $\neg F$  indicano che questa ipotesi è stata scaricata, mentre l'**indice 1** (riportato anche sulla linea che corrisponde al passaggio relativo a ( $\neg$ i)) serve ad indicare quando lo scaricamento è avvenuto.

Osservando la deduzione naturale globalmente, è ora possibile vedere che l'unica ipotesi non scaricata è  $F$ , mentre la conclusione è  $\neg\neg F$ , e perciò abbiamo ottenuto ciò che volevamo:  $F \triangleright \neg\neg F$ .

### Esempio

Come primo esempio di applicazione della regola ( $\vee$ e) consideriamo la seguente deduzione naturale, che mostra che

$$\neg F \wedge \neg G \triangleright \neg(F \vee G)$$

$$\frac{[F \vee G]^2 \quad \frac{[F]^1 \quad \frac{\neg F \wedge \neg G}{\neg F}}{\perp} \quad \frac{[G]^1 \quad \frac{\neg F \wedge \neg G}{\neg G}}{\perp} 1}{\frac{\perp}{\neg(F \vee G)} 2}$$

Proviamo a percorrere a parole il ragionamento rappresentato da questa deduzione naturale. Dato che vogliamo dimostrare  $\neg(F \vee G)$ , facciamo l'ipotesi ausiliaria  $F \vee G$  con l'obiettivo di ottenere  $\perp$  e utilizzare la regola ( $\neg$ i), in pratica si tratta di ottenere

$$\neg F \wedge \neg G, F \vee G \triangleright \perp.$$

Per sfruttare l'ipotesi appena fatta dobbiamo usare ( $\vee$ e), e quindi supporre separatamente  $F$  e  $G$ , ottenendo  $\perp$  in entrambi i casi, cioè ottenere:

- $\neg F \wedge \neg G, F \triangleright \perp$
- $\neg F \wedge \neg G, G \triangleright \perp$

Ciò non è difficile, applicando le regole di eliminazione di  $\wedge$  all'ipotesi  $\neg F \wedge \neg G$ . Avendo ottenuto  $\perp$  a partire sia da  $F$  che da  $G$  otteniamo  $\perp$  da  $F \vee G$  (notiamo che a questo punto della deduzione naturale, cioè dopo la linea marcata con 1, l'ipotesi  $F \vee G$  non è ancora stata scaricata, come testimoniato dall'indice 2 che rimanda ad una linea successiva). A questo punto è sufficiente utilizzare, come previsto, ( $\neg$ i) per concludere la dimostrazione.

La regola di riduzione all'assurdo (RAA), che avevamo dedotto astrattamente nelle sezione precedente, è rappresentata da

$$\frac{T, [\neg F]}{\frac{\perp}{F}}$$

### Esempio

Consideriamo la seguente deduzione naturale, che mostra che  $F \vee \neg G, G \triangleright F$ :

$$\frac{F \vee \neg G \quad \frac{[F]^2 \quad \frac{\frac{G \quad [\neg G]^2}{\perp} 1}{\neg \neg F} 1}{F} 2}{F}$$

Dato che il risultato di  $(\neg i)$  è  $\neg \neg F$ , la formula a cui applichiamo la regola è  $\neg F$ , e siamo quindi autorizzati a scaricare l'ipotesi ausiliaria  $\neg F$ . In questo caso però non c'è nessuno scaricamento (ed in effetti non vi sono ipotesi ausiliarie contrassegnate con 1), perché l'ipotesi ausiliaria non è mai stata utilizzata. Questa situazione evidenzia come **lo scaricamento** di un'ipotesi sia una **possibilità** e non un obbligo, nella costruzione della deduzione naturale.

L'utilizzo consecutivo di  $(\neg i)$  e  $(\neg e)$  nell'esempio precedente può essere isolato considerando la deduzione naturale

$$\frac{\frac{\perp}{\neg \neg F} 1}{F}$$

Otteniamo perciò la regola

$$\frac{\perp}{F}$$

che asserisce che da una **contraddizione** è possibile **dedurre qualsiasi formula** e formalizza il principio logico tradizionalmente noto come **ex-falso quodlibet sequitur** (dal falso segue qualunque cosa si desideri) e perciò è nota come **ex-falso**.

Un altro esempio di regola derivata è la regola (MT) di **modus tollens** ("toglie" la verità a di una proposizione "togliendo" quella di un'altra)

$$\frac{F \rightarrow G \quad \neg G}{\neg F}$$

### Le regole della deduzione naturale proposizionale

In questa sezione riassumiamo tutte le regole del sistema di deduzione naturale. Nel formulare le regole inseriamo esplicitamente le ipotesi delle deduzioni naturali.

$$\begin{array}{c}
 (\wedge i) \quad \frac{\begin{array}{c} T \\ \nabla \\ F \end{array} \quad \begin{array}{c} T' \\ \nabla \\ G \end{array}}{F \wedge G} \qquad (\wedge e.1) \quad \frac{\begin{array}{c} T \\ \nabla \\ F \wedge G \end{array}}{F} \qquad (\wedge e.2) \quad \frac{\begin{array}{c} T \\ \nabla \\ F \wedge G \end{array}}{G} \\
 \\
 (\vee i.1) \quad \frac{\begin{array}{c} T \\ \nabla \\ F \end{array}}{F \vee G} \qquad (\vee i.2) \quad \frac{\begin{array}{c} T \\ \nabla \\ G \end{array}}{F \vee G} \qquad (\vee e) \quad \frac{\begin{array}{c} T \\ \nabla \\ F \vee G \end{array} \quad \begin{array}{c} T', [F] \\ \nabla \\ H \end{array} \quad \begin{array}{c} T'', [G] \\ \nabla \\ H \end{array}}{H} \\
 \\
 (\rightarrow i) \quad \frac{\begin{array}{c} T, [F] \\ \nabla \\ G \end{array}}{F \rightarrow G} \qquad (\rightarrow e) \quad \frac{\begin{array}{c} T \\ \nabla \\ F \end{array} \quad \begin{array}{c} T' \\ \nabla \\ F \rightarrow G \end{array}}{G} \\
 \\
 (\neg i) \quad \frac{\begin{array}{c} T, [F] \\ \nabla \\ \perp \end{array}}{\neg F} \qquad (\neg e) \quad \frac{\begin{array}{c} T \\ \nabla \\ F \end{array} \quad \begin{array}{c} T' \\ \nabla \\ \neg F \end{array}}{\perp} \qquad (\neg\neg e) \quad \frac{\begin{array}{c} T \\ \nabla \\ \neg\neg F \end{array}}{F} \\
 \\
 (RAA) \quad \frac{\begin{array}{c} T, [\neg F] \\ \nabla \\ \perp \end{array}}{F} \qquad (ex-falso) \quad \frac{\begin{array}{c} T \\ \nabla \\ \perp \end{array}}{F} \\
 \\
 (MT) \quad \frac{\begin{array}{c} T \\ \nabla \\ F \rightarrow G \end{array} \quad \begin{array}{c} T' \\ \nabla \\ \neg G \end{array}}{\neg F} \qquad (TE) \quad [F \vee \neg F]
 \end{array}$$

## La logica predicativa

### Cap. 6: Sintassi

#### Alfabeto

É diviso in due parti:

Una parte **fissa**, comune ad ogni linguaggio:

- Un insieme di **variabili**:  $x, y, z, \dots$ ;
- I **simboli logici**:  $\neg, \wedge, \vee, \rightarrow$ ;
- I **quantificatori**  $\forall$  e  $\exists$ ;
- La **virgola** “,” e le **parentesi** “(” e “)”.

Una parte variabile (di insiemi disgiunti):

- Un insieme di **simboli di costante**;
- Un insieme di **simboli di funzione**, a ciascuno dei quali è associato il suo numero di argomenti (**arietà**);
- Un insieme di **simboli di relazione** (o simboli predicativi, non vuoto), ciascuno fornito della propria arietà.

#### Esempio

Il linguaggio *Lar*<sub>it</sub>, adatto a fare affermazioni sull’aritmetica, consiste dei:

- simboli di costante 0 e 1;
- dei simboli di funzione binari  $+$  e  $\times$ ;
- dei simboli di relazione binari  $<, >$  e  $=$ .

#### Esempio

Il linguaggio *Lfam*, consiste dei:

- simboli di costante per ogni membro della famiglia (Anna, mamma, papà, zio, ...);
- simboli di funzione binari “la madre di”, “il padre di”, “lo zio di”, ...;
- simboli di relazione binari “essere fratelli”, “essere figlio di”, ....

#### Esempio

Se  $F$  è  $p(a) \wedge \forall x(p(f(x)) \rightarrow \neg r(x, b)) \rightarrow \exists y(q(a, y, g(y, b)) \vee q(y, g(y, a), f(b)))$ :

$L(F)$  consiste dei:

- simboli di costante  $a$  e  $b$ ;
- simboli di funzione  $f$  (unario) e  $g$  (binario);
- simboli di relazione  $p$  (unario),  $r$  (binario) e  $q$  (ternario).

#### Termini

L'insieme dei **termini** di un linguaggio è costituito dal minimo insieme di stringhe dell'alfabeto che possano denotare degli oggetti specifici, formalmente si dà la seguente definizione ricorsiva:

- Una **variabile** è un termine;
- Una **simbolo di costante** è un termine;
- Se  $t_1, \dots, t_n$  sono **termini** e  $f$  è un **simbolo di funzione**  $n$ -ario di  $L$ , allora  $f(t_1, \dots, t_n)$  è un termine di  $L$ .

Un termine è **chiuso** se in esso non compare nessuna variabile.



### Esempi

Alcuni termini del linguaggio *Larit* sono:  $x$ ,  $1$ ,  $+$ ,  $(x, 0)$ ,  $(x \cdot y) + (0 \cdot (y + x))$  (solo il secondo è chiuso).

- $x$
- $1$   
(termine chiuso)
- $+$  ( $x, 0$ )
- $(x \cdot y) + (0 \cdot (y + x))$

### Esempio

Alcuni termini del linguaggio *Lfam* sono:

- $p(m(Anna))$  "il padre della madre di Anna", con  $p$  ed  $m$  funzioni unarie e  $Anna$  costante (termine chiuso).

### Sostituzione di simboli

Se  $x$  è una variabile e  $s$  e  $t$  sono termini definiamo la **sostituzione di  $x$  con  $t$  in  $s$** , ( $s\{x/t\}$ ):

- Se  $s$  è la **variabile**  $x$  allora  $s\{x/t\} = t$ ;
- Se  $s$  è una **variabile diversa da  $x$**  oppure un **simbolo di costante** allora  $s\{x/t\} = s$ ;
- Se  $s$  è  $f(s_1, \dots, s_n)$  allora  $s\{x/t\} = f(s_1\{x/t\}, \dots, s_n\{x/t\})$ .

### Esempio

Consideriamo un linguaggio contenente le costanti  $a$  e  $b$  e le funzioni  $f$  e  $g$ , il primo binario e il secondo unario.

$$\begin{aligned}f(g(x), a)\{x/b\} &\text{ è } f(g(b), a); \\f(x, g(x))\{x/g(a)\} &\text{ è } f(g(a), g(g(a))); \\f(x, g(x))\{x/g(x)\} &\text{ è } f(g(x), g(g(x))); \\g(y)\{x/a\} &\text{ è } g(y); \\x\{x/a\} &\text{ è } a.\end{aligned}$$

Notiamo che prevediamo di effettuare sostituzioni solo di variabili: nel linguaggio dell'esempio precedente le espressioni  $f(g(x), a)\{a/b\}$  e  $f(g(x), a)\{g(x)/y\}$  non sono accettabili, perché non possiamo sostituire un simbolo di costante o un termine costruito con simboli di funzione.

### Formule predicative

#### Formule atomiche

Le **formule atomiche di  $L$**  sono stringhe di simboli del tipo  $p(t_1, \dots, t_n)$  con  $p$  simbolo di relazione  $n$ -ario e  $t_1, \dots, t_n$  termini.

#### Formule di $L$

L'**insieme delle formule di  $L$**  è definito come segue:

- Ogni formula atomica è una formula;
- Se  $F$  è una formula allora  $(\neg F)$  è una formula;
- Se  $F$  e  $G$  sono formule allora  $(F \wedge G)$ ,  $(F \vee G)$  e  $(F \rightarrow G)$  sono formule;
- se  $F$  è una formula di  $L$  e  $x$  è una variabile, allora  $(\forall x F)$  e  $(\exists x F)$  sono formule di  $L$ .

Notate che se un linguaggio fosse **privo di simboli di relazione** non avrebbe **formule atomiche** e di conseguenza non avrebbe **formule**: per questo l'insieme dei simboli di relazione deve essere non vuoto.

**Convenzioni di lettura**  $\neg, \wedge, \vee, \rightarrow, \forall, \exists$

Nella scrittura delle formule adotteremo le seguenti convenzioni:

- 1) si omettono le parentesi più esterne;
- 2)  $\neg, \forall$  e  $\exists$  hanno la precedenza su  $\wedge, \vee$  e  $\rightarrow$ ;
- 3)  $\wedge$  e  $\vee$  hanno la precedenza su  $\rightarrow$ ;
- 4) ulteriori simboli si appoggiano a sinistra.

### Grado di una formula

Il grado della formula  $F$ , indicato con  $g(F)$ , è definito da:

- $g(F) = 0$  se  $F$  è atomica;
- $g(\neg F) = g(\forall x F) = g(\exists x F) = g(F) + 1$ ;
- $g(F \wedge G) = g(F \vee G) = g(F \rightarrow G) = g(F) + g(G) + 1$ .

### Variabili libere o legate ed enunciati

Il ruolo della variabile  $x$  nelle formule  $p(x)$  e  $\forall x p(x)$  ( $p$  è un simbolo di relazione unario) è **diverso**:

- $p(x)$ :  $x$  è biondo;
- $\forall x p(x)$ : ogni  $x$  è biondo.

Per decidere se la prima formula è vera o falsa è necessario dare un significato, oltre che a  $p$ , anche a  $x$ , mentre la verità o falsità della seconda formula dipende solo da come interpretiamo  $p$ .

Sia  $F$  una formula e  $x$  una variabile. Definiamo le **occorrenze libere di  $x$  in  $F$**  come segue:

- Se  $F$  è atomica: **ogni occorrenza** di  $x$  in  $F$  è libera (*a.e.*:  $p(x)$  è atomica quindi  $x$  è libera);
- Se  $F$  è  $\neg G$ : le **occorrenze libere di  $x$  in  $F$**  sono le **occorrenze libere di  $x$  in  $G$** ;
- Se  $F$  è  $G \wedge H, G \vee H$  o  $G \rightarrow H$ : le **occorrenze libere di  $x$  in  $F$**  sono le **occorrenze libere di  $x$  in  $G$**  e le **occorrenze libere di  $x$  in  $H$** ;
- Se  $F$  è  $\forall x G$  o  $\exists x G$ : **nessuna occorrenza** di  $x$  in  $F$  è libera;
- Se  $F$  è  $\forall y G$  o  $\exists y G$  con  $y \neq x$  variabili: le **occorrenze libere di  $x$  in  $F$**  sono le **occorrenze libere di  $x$  in  $G$** .

Le occorrenze di una variabile in  $F$  che **non sono libere** si dicono **occorrenze legate** (in particolare l'occorrenza di una variabile **subito dopo un quantificatore** è legata).

Una formula **priva di variabili libere** è chiamata **enunciato o formula chiusa**.

### Esempio

In *Larit*

- $x + 1 = y$  ha  $x$  e  $y$  come variabili libere
- $\forall x(x + 1 = y)$  ha  $y$  come variabile libera
- $\exists y \forall x (x + 1 = y)$  è un enunciato
- $(\exists y y = 0) \wedge y = 0 + 1$  l'ultima  $y$  è libera, nonostante le sue due prime occorrenze siano legate.

### Esempio

Le variabili sottolineate nella formula seguente sono libere:

$$\forall x(r(f(x), \underline{y}) \rightarrow \exists y r(f(y), x)) \rightarrow \forall y(\neg r(y, \underline{x}) \vee r(\underline{z}, f(\underline{z})) \rightarrow \exists w \forall x r(x, w))$$

### Letterali

Un **letterale** è una **formula atomica** o la **negazione di una formula atomica**.

## Chiusura universale

Sia  $F$  una formula con variabili libere  $x_1, \dots, x_n$  l'enunciato:

- $\forall x_1 \dots \forall x_n F$  è una **chiusura universale** di  $F$  e si può indicare con  $\forall F$  dato che le chiusure universali di  $F$  sono uguali;
- $\exists x_1 \dots \exists x_n F$  è una **chiusura esistenziale** di  $F$  e si può indicare con  $\exists F$  dato che tutte le chiusure esistenziali di  $F$  sono uguali.

## Sottoformule

Se  $F$  è una formula, diciamo che  $G$  è una **sottoformula di  $F$**  se  $G$  è una formula che è una sottostringa di  $F$ .  $G$  è una sottoformula propria di  $F$  se è diversa da  $F$ :

- se  $F$  è atomica,  **$F$  è la sua unica sottoformula**;
- se  $F$  è  $\neg G$ ,  $\forall x G$  oppure  $\exists x G$  allora le **sottoformule di  $F$**  sono le **sottoformule di  $G$  e  $F$  stessa**;
- se  $F$  è  $G \wedge H$ ,  $G \vee H$  o  $G \rightarrow H$  allora le **sottoformule di  $F$**  sono le **sottoformule di  $G$ , di  $H$  e  $F$  stessa**.

## Esempio

Se  $F$  è  $\forall x(p(x) \rightarrow \exists y p(y) \vee r(x, y))$  allora, ad esempio,  $\exists y p(y) \vee r(x, y)$  è una sottoformula di  $F$ , mentre, ad esempio,  $p(x) \rightarrow \exists y p(y)$  non lo è.

## Sostituzioni di formule

Se  $F$  è una **formula atomica**  $p(s_1, \dots, s_n)$ ,  $x$  è una variabile e  $t$  è un termine, la **sostituzione di  $x$  con  $t$  in  $F$**  è  $p(s_1\{x/t\}, \dots, s_n\{x/t\})$  ed è denotata da  $F\{x/t\}$ .

Se  $F$  **non è una formula atomica**, la sostituzione della variabile  $x$  con il termine  $t$  è ammissibile in  $F$  se  $t$  è libero per la sostituzione al posto di ogni occorrenza libera di  $x$  in  $F$  (ossia se  $t$  non contiene nessuna variabile  $y$  per cui esiste una sottoformula di  $F$  del tipo  $\forall y G$  o  $\exists y G$  che contiene l'occorrenza di  $x$  che stiamo considerando). In questo caso la formula  $F\{x/t\}$  è ottenuta sostituendo in  $F$  ogni formula atomica  $A$  in cui  $x$  compare libera con  $A\{x/t\}$ .

## Esempio

- $q(x, g(x, f(y)), f(g(z, y)))\{x/f(w)\}$ :  
 $q(f(w), g(f(w), f(y)), f(g(z, y)))$ .
- $\forall z r(x, z) \wedge \exists y r(x, f(y)) \rightarrow \neg \exists x r(x, x)$ :  
le sostituzioni  $\{x/y\}$ ,  $\{x/f(y)\}$  e  $\{x/z\}$  non sono ammissibili in  $F$ , mentre le sostituzioni  $\{x/c\}$ ,  $\{x/f(x)\}$  e  $\{x/w\}$  sono ammissibili e conducono rispettivamente a  
 $\forall z r(c, z) \wedge \exists y r(c, f(y)) \rightarrow \neg \exists x r(x, x)$ ,  
 $\forall z r(f(x), z) \wedge \exists y r(f(x), f(y)) \rightarrow \neg \exists x r(x, x)$ ,  
 $\forall z r(w, z) \wedge \exists y r(w, f(y)) \rightarrow \neg \exists x r(x, x)$ .

## Linguaggi con uguaglianza

Un linguaggio  $L$  si dice un **linguaggio con uguaglianza** se tra i suoi simboli di relazione binari c'è  $=$ .

## Cap. 7: Semantica della logica predicativa

### Interpretazioni I

Dato un linguaggio  $L$  una **interpretazione**  $I$  per  $L$  è data da:

- Un insieme non vuoto  $D^I$  **dominio dell'interpretazione** (a.e. insieme dei naturali, insieme studenti, ...);
- Per ogni **simbolo di costante**  $c$  in  $L$ , un elemento  $c^I \in D^I$  (a.e. "Anna");
- Per ogni **simbolo di funzione**  $n$ -ario  $f$  in  $L$ , una funzione  $f^I: (D^I)^n \rightarrow D^I$  (a.e.  $+$  ( $12 + 23$ ) dove il simbolo di funzione  $+$  prende due elementi del dominio ( $N$ ) e restituisce un altro elemento del dominio ( $N$ ));
- Per ogni **simbolo di relazione**  $n$ -ario  $p$  in  $L$ , un insieme  $p^I \subseteq (D^I)^n$ .

Un'interpretazione  $I$  per il linguaggio  $L$  associa ad ogni termine chiuso  $t$  di  $L$  la sua **interpretazione** in  $I$ , che è un elemento  $t^I \in D^I$  definito ricorsivamente da:

- se  $t$  è un simbolo di costante  $c$  allora  $t^I = c^I$ ;
- se  $t = f(t_1, \dots, t_n)$  allora  $t^I = f^I(t_1^I, \dots, t_n^I)$ .

Se un termine non è chiuso (cioè contiene delle variabili) l'interpretazione non basta per stabilire quale elemento del dominio associare al termine.

Uno **stato dell'interpretazione**  $I$  è una funzione  $\sigma: Var \rightarrow D^I$  che ad ogni variabile associa un elemento del dominio dell'interpretazione  $I$ . Uno stato  $\sigma$  di  $I$  associa ad ogni termine  $t$  un valore,  $\sigma(t) \in D^I$ , dove  $\sigma(t)$  è definito per ricorsione sulla complessità di  $t$  da:

- se  $t$  è una variabile  $x$  allora  $\sigma(t) = \sigma(x)$ ;
- se  $t$  è una costante  $c$  allora  $\sigma(t) = c^I$ ;
- se  $t = f(t_1, \dots, t_n)$  allora  $\sigma(t) = f^I(\sigma(t_1), \dots, \sigma(t_n))$ .

### Esempio

Sia  $L = \{c, f, p, q, r\}$  un linguaggio con un simbolo di costante  $c$ , un simbolo funzionale unario  $f$ , due simboli relazionali unari  $p$  e  $q$  e un simbolo relazionale binario  $r$ . Definiamo un'interpretazione  $I$  per  $L$ :

$$D^I = \{0, 1, 2\}, \quad c^I = 1, \quad f^I(0) = 1, \quad f^I(1) = 2, \quad f^I(2) = 1, \\ p^I = \{1, 2\}, \quad q^I = \{0, 2\}, \quad r^I = \{(0, 0), (0, 2), (1, 2)\}.$$

Definiamo uno stato di  $I$  ponendo  $\sigma(x) = 0$ ,  $\sigma(y) = 1$  e  $\sigma(v) = 2$  per tutte le variabili  $v$  diverse da  $x$  e  $y$ .

L'interpretazione in  $I$  del termine chiuso  $f(c)$  è  $(f(c))^I = f^I(c^I) = f^I(1) = 2$ . Il valore secondo  $\sigma$  del termine  $f(x)$  è  $\sigma(f(x)) = f^I(\sigma(x)) = f^I(0) = 1$ .

### Lemma

Sia  $t$  un termine di un linguaggio  $L$ , e siano  $I$  e  $I'$  interpretazioni per  $L$  che hanno lo stesso dominio e coincidono sulle interpretazioni dei simboli di costante e di funzione che occorrono in  $t$ . Siano inoltre  $\sigma$  e  $\sigma'$  stati rispettivamente di  $I$  e  $I'$  che coincidono sulle variabili che occorrono in  $t$ . Allora  $\sigma(t) = \sigma'(t)$ .

### Soddisfazione $\models$

Siano  $F$  una formula di un linguaggio  $L$ ,  $I$  un'interpretazione per  $L$  e  $\sigma$  uno stato di  $I$ . Definiamo la relazione

$$I, \sigma \models F \text{ (} I \text{ allo stato } \sigma \text{ soddisfa } F \text{)}$$

per ricorsione sulla complessità di  $F$  ( $I, \sigma \not\models F$  indica che  $I, \sigma \models F$  non vale):

- $I, \sigma \models p(t_1, \dots, t_n)$  sse  $(\sigma(t_1), \dots, \sigma(t_n)) \in p^I$ ;

- $I, \sigma \models \neg G$  sse  $I, \sigma \not\models G$ ;
- $I, \sigma \models G \wedge H$  sse  $I, \sigma \models G$  e  $I, \sigma \models H$ ;
- $I, \sigma \models G \vee H$  sse  $I, \sigma \models G$  oppure  $I, \sigma \models H$ ;
- $I, \sigma \models G \rightarrow H$  sse  $I, \sigma \not\models G$  oppure  $I, \sigma \models H$ ;
- $I, \sigma \models \forall x G$  sse per ogni  $d \in D$  si ha che  $I, \sigma[x/d] \models G$ ;
- $I, \sigma \models \exists x G$  sse esiste  $d_0 \in D$  tale che  $I, \sigma[x/d_0] \models G$ .

Diciamo che **I soddisfa F** ( $I \models F$ ) se  $I, \sigma \models F$  per ogni stato  $\sigma$  di  $I$ . In questo caso si dice anche che **F è vera in I** oppure che **I è un modello di F**.

Se  $T$  è un insieme di formule, diciamo che **I allo stato  $\sigma$  soddisfa T** ( $I, \sigma \models T$ ) se  $I$  allo stato  $\sigma$  soddisfa ogni  $F \in T$ . Anche in questo caso diciamo che  $I$  soddisfa  $T$ , o che  $T$  è vera in  $I$  oppure che  $I$  è un modello di  $T$ , e scriviamo  $I \models T$ , se  $I, \sigma \models T$  per ogni stato  $\sigma$  di  $I$ .

### Esempio

Sia  $L = \{c, f, p, q, r\}$  un linguaggio con un simbolo di costante  $c$ , un simbolo funzionale unario  $f$ , due simboli relazionali unari  $p$  e  $q$  e un simbolo relazionale binario  $r$ . Definiamo un'interpretazione  $I$  per  $L$ :

$$D^I = \{0, 1, 2\}, \quad c^I = 1, \quad f^I(0) = 1, \quad f^I(1) = 2, \quad f^I(2) = 1, \\ p^I = \{1, 2\}, \quad q^I = \{0, 2\}, \quad r^I = \{(0, 0), (0, 2), (1, 2)\}.$$

Definiamo uno stato di  $I$  ponendo  $\sigma(x) = 0, \sigma(y) = 1$  e  $\sigma(v) = 2$  per tutte le variabili  $v$  diverse da  $x$  e  $y$ . Questa interpretazione  $I$ , nello stato  $\sigma$ , soddisfa la formula  $(q(f(y)) \rightarrow \neg p(c)) \vee \exists z (r(x, z) \wedge p(z) \wedge r(y, z))$  ossia:

$$I, \sigma \models (q(f(y)) \rightarrow \neg p(c)) \vee \exists z (r(x, z) \wedge p(z) \wedge r(y, z))?$$

Tale formula è una **disgiunzione** (i due disgiunti sono infatti:  $(q(f(y)) \rightarrow \neg p(c))$  e  $\exists z (r(x, z) \wedge p(z) \wedge r(y, z))$ ). Allora passo ad analizzare le sottoformule:

- ★  $I, \sigma \models (q(f(y)) \rightarrow \neg p(c))$ ? Tale formula è una **implicazione**, allora passo ad analizzare le sottoformule:
  - $I, \sigma \models (f(y))$ ?  $\sigma(f(y)) = f^I(\sigma(y)) = f^I(1) = 2$  e dato che  $2 \in q^I$  si ha  $I, \sigma \models q(f(y))$ ;
  - $I, \sigma \models \neg p(c)$ ?  $\sigma(c) = c^I = 1$  e dato che  $1 \in p^I$  si ha  $I, \sigma \models p(c)$  e quindi  $I, \sigma \not\models \neg p(c)$ ;
  - Perciò  $I, \sigma \not\models q(f(y)) \rightarrow \neg p(c)$ . L'unica possibilità che  $I, \sigma \models F$  è che  $I, \sigma \models \exists z (r(x, z) \wedge p(z) \wedge r(y, z))$  dato che il primo disgiunto ha valore **FALSO**.
- ★  $I, \sigma \models \exists z (r(x, z) \wedge p(z) \wedge r(y, z))$ ? Almeno una delle tre perturbazioni  $[z/\text{elemento}]$  è **VERA**?:
  - $I, \sigma[z/0] \models \exists z (r(x, z) \wedge p(z) \wedge r(y, z))$ ?
    - $I, \sigma[z/0] \models r(x, z)$  ok;
    - $I, \sigma[z/0] \not\models p(z)$  non va bene, 0 non è l'elemento adatto.
  - $I, \sigma[z/1] \models \exists z (r(x, z) \wedge p(z) \wedge r(y, z))$ ?
    - $I, \sigma[z/1] \not\models r(x, z)$  non va bene, neanche 1 è l'elemento adatto.
  - $I, \sigma[z/2] \models \exists z (r(x, z) \wedge p(z) \wedge r(y, z))$ ?
    - $I, \sigma[z/2] \models r(x, z)$  ok;
    - $I, \sigma[z/2] \models p(z)$  ok;
    - $I, \sigma[z/2] \models r(y, z)$  ok;

Quindi  $I, \sigma[z/2] \models r(x, z) \wedge p(z) \wedge r(y, z)$ .

Perciò  $I, \sigma \models \exists z (r(x, z) \wedge p(z) \wedge r(y, z))$  e quindi  $I, \sigma \models F$ .

### Lemma

Sia  $F$  una formula, e siano  $I$  e  $I'$  interpretazioni con lo stesso dominio e con gli stessi simboli di costante, di funzione e di relazione che occorrono in  $F$ . Siano inoltre  $\sigma$  e  $\sigma'$  stati rispettivamente di  $I$  e  $I'$  che coincidono sulle variabili libere di  $F$ . Allora:

$$I, \sigma \models F \text{ sse } I', \sigma' \models F$$

### Corollario

Se  $F$  è un enunciato di  $L$ ,  $I$  un'interpretazione per  $L$  e  $\sigma$  e  $\sigma'$  due stati di  $I$  allora  $I, \sigma \models F$  se e solo se  $I, \sigma' \models F$ . Quindi  $I \models F$  sse  $I, \sigma \models F$  per qualche stato  $\sigma$  di  $I$ .

Ossia possiamo ignorare lo stato nello stabilire la soddisfazione di un enunciato in un'interpretazione. Per la verifica di questa soddisfazione uno stato ausiliario sarà però spesso utile: ad esempio per verificare se  $I \models \forall x F$ , dove  $F$  è una formula in cui  $x$  è l'unica variabile libera, dobbiamo considerare

$$\text{per ogni } d \in D' \text{ se } I, \sigma[x/d] \models F;$$

In questo caso però per il lemma sopra possiamo **scegliere  $\sigma$  arbitrariamente**: l'unica proprietà di  $\sigma[x/d]$  che ci interessa è che manda  $x$  in  $d$ .

### Esercizio

Sia  $H$  l'enunciato  $\forall x \forall y (r(x, y) \rightarrow \neg r(y, x)) \wedge \forall x \exists y r(x, y)$ . Trovare  $I$  tale che  $I \models H$ .

$$D' = \{0, 1, 2\}$$

$$r' = \{(0, 1), (1, 2), (2, 0)\} \quad (0, 0), (1, 1), (1, 0), (2, 2), (2, 1) \notin r'$$

### Equivalenza logica $\equiv$ e conseguenza logica $\models$

Siano  $F$  e  $G$  due formule. Diciamo che  $F$  e  $G$  sono **logicamente equivalenti** ( $F \equiv G$ ) se per ogni interpretazione  $I$  per  $L(F, G)$  e ogni stato  $\sigma$  di  $I$  si ha

$$I, \sigma \models F \text{ sse } I, \sigma \models G.$$

Siano  $F$  e  $G$  due formule. Diciamo che  $G$  è **conseguenza logica** di  $F$  ( $F \models G$ ) se per ogni interpretazione  $I$  per  $L(F, G)$  e ogni stato  $\sigma$  di  $I$  tali che:

$$I, \sigma \models F \text{ si ha } I, \sigma \models G.$$

Siano  $T$  e  $G$  un insieme di formule ed una formula. Diciamo che  $G$  è **conseguenza logica** di  $T$  ( $T \models G$ ) se per ogni interpretazione  $I$  per  $L(T, G)$  ed ogni stato  $\sigma$  di  $I$  tale che

$$I, \sigma \models T \text{ si ha } I, \sigma \models G.$$

Come nel caso proposizionale,  $\models F$  sta ad indicare  $\emptyset \models F$ .

### Nota

Lo stesso simbolo  $\models$  viene usato per denotare sia la nozione di **soddisfazione** che quella di **conseguenza logica**. E sempre possibile capire con quale delle due nozioni si ha a che fare semplicemente esaminando ciò che compare a sinistra di  $\models$ :

- Se c'è una **interpretazione** con uno stato: soddisfazione;
- Se c'è una **formula**: conseguenza logica.

### Lemma di chiusura universale ed esistenziale

Sia  $F$  una formula e  $I$  un'interpretazione per  $L(F)$ . Se  $G$  è una **chiusura universale** di  $F$ , allora  $I \models G$  sse  $I \models F$  (cioè  $I, \sigma \models F$  per ogni stato  $\sigma$  di  $I$ ). Due chiusure universali della **stessa formula**  $F$  sono **logicamente equivalenti**.

Se  $H$  è una **chiusura esistenziale** di  $F$  allora  $I \models H$  sse esiste uno stato  $\sigma$  di  $I$  tale che  $I, \sigma \models F$ . Due chiusure esistenziali della **stessa formula** sono **logicamente equivalenti**.

### Esercizio

Sia  $F$  la formula  $\forall x (p(x) \rightarrow \neg r(x, x))$  e sia  $G$  la formula  $\neg \exists x (p(x) \wedge \forall y r(y, x))$ . Dimostrare  $F \models G$  (ossia che non può esistere un'interpretazione in cui  $F$  è vera ma  $G$  è falsa).

Il  $L = \{p, r\}$  dove  $p$  è un simbolo di relazione unario ed  $r$  è un simbolo di relazione binario. Sia  $I$  una interpretazione tale che  $I \models F$ , voglio quindi ottenere  $I \models G$  ragionando per assurdo:

Suppongo che  $I \models G$ : allora  $I \models \exists x(p(x) \wedge \forall y r(y, x))$ ;

Questo significa che esiste  $d_0 \in D'$  tale che  $I, \sigma[x/d_0] \models p(x) \wedge \forall y r(y, x)$  e allora  $d_0 \in p'$  e  $\forall d \in D'$   $I, \sigma[x/d_0][y/d] \models r(y, x)$ . Cioè la coppia  $(d, d_0) \in r'$  per ogni  $\forall d \in D'$ .

Dato che  $I \models F$ , avrò che  $I, \sigma[x/d_0] \models p(x) \rightarrow \neg r(x, x)$  e per rendere vera l'implicazione si deve avere:

- $I, \sigma[x/d_0] \models p(x)$
- $I, \sigma[x/d_0] \models \neg r(x, x)$

Di conseguenza  $(d_0, d_0) \notin r'$  che è un assurdo dato che:  $(d, d_0) \in r'$  per ogni  $\forall d \in D'$ .

### Lemma

Sia  $F$  sottoformula di  $H$  e sia  $G$  una formula tale che  $F \equiv G$ . Sia  $H'$  una formula ottenuta da  $H$  sostituendo  $F$  con  $G$ . allora  $H' = H$ .

### Lemma

Se  $x$  non è una variabile libera della formula  $F$  allora  $F \equiv \forall x F \equiv \exists x F$ .

### Validità e soddisfacibilità

Se  $F$  è una formula diciamo che:

- $F$  è **valida** se  $\forall I, \sigma$  si ha  $I, \sigma \models F$  ( $F$  è valida sse  $\neg F$  è sempre insoddisfacibile);
- $F$  è **soddisfacibile** se  $\exists I, \sigma$  tali che  $I, \sigma \models F$ ;
- $F$  è **insoddisfacibile** se  $\forall I, \sigma$  si ha  $I, \sigma \models F$  si ha  $I, \sigma \not\models F$  ( $F$  è insoddisfacibile sse  $\neg F$  è valida).

### Esempi di formule valide, soddisfacibili e non soddisfacibili

- $p(a) \vee \neg p(a)$  è valida;
- $\forall x p(x) \rightarrow p(a)$  è valida;
- $p(a) \rightarrow \exists x p(x)$  è valida;
- $p(a)$  è soddisfacibile ma non valida;
- $p(a) \wedge \exists x \neg p(x)$  è soddisfacibile ma non valida;
- $\exists x p(x) \wedge \forall x \neg p(x)$  è insoddisfacibile;
- $p(a) \wedge \neg p(a)$  è insoddisfacibile;

Più in generale, le formule  $\forall x F \rightarrow F\{x/a\}$  e  $F\{x/a\} \rightarrow \exists x F$  sono **valide**, mentre  $\exists x F \wedge \forall x \neg F$  è **insoddisfacibile**.

### Esercizio (20.14.1.mp4)

Dimostrare che l'insieme di formule  $\exists x (p(x) \wedge \forall y \neg r(y, x))$ ,  $p(c)$ ,  $r(c, c)$ ,  $\forall x \exists y (r(x, y) \wedge \neg p(y))$  è soddisfacibile (basta che uno dei quattro enunciati sia vero).

$$D' = \{0, 1, 2\}$$

$$c' = 0$$

$$p' = \{0, 1\}$$

$$r' = \{(0, 0), (0, 2), (1, 2), (2, 2)\}$$

### **Lemma di sostituzione...**

#### **...per termini**

Siano  $\sigma$  uno stato dell'interpretazione  $I$ ,  $x$  una variabile e  $s$  e  $t$  due termini. Allora

$$\sigma(s\{x/t\}) = \sigma[x/\sigma(t)](s)$$

#### **...per formule**

Siano  $\sigma$  uno stato dell'interpretazione  $I$ ,  $x$  una variabile,  $t$  un termine e  $F$  una formula. Se la sostituzione  $\{x/t\}$  è ammissibile in  $F$ , allora

$$I, \sigma \models F\{x/t\} \text{ sse } I, \sigma[x/\sigma(t)] \models F$$

### **Lemma**

Se  $F$  è una formula,  $y$  una variabile non libera in  $F$  e la sostituzione  $\{x/y\}$  è ammissibile in  $F$  allora

$$\forall x F \equiv \forall y F\{x/y\} \text{ e } \exists x F \equiv \exists y F\{x/y\}.$$

### **Corollario**

Se  $y$  è una variabile che non occorre in  $F$  allora  $F \equiv Fx(y)$ .

### **Lemma**

Se  $F$  è una formula,  $x$  una variabile,  $t$  un termine e  $\{x/t\}$  una sostituzione ammissibile in  $F$ , allora

$$\forall x F \models F\{x/t\} \text{ e } F\{x/t\} \models \exists x F.$$

### **Corollario**

Se  $F$  è una formula e  $x$  una variabile allora  $\forall x F \models F$  e  $F \models \exists x F$ .

### **Logica con uguaglianza Eq**

Una  $I$  per  $L$  è **normale** se  $='$  è la relazione di identità su  $D'$ , cioè  $='$  è  $\{(d, d) : d \in D'\}$ .

Scrivendo  $I, \sigma \models_{=} F$  intendiamo dire che  $I, \sigma \models F$  e che  $I$  è un'interpretazione normale.

### **Lemma**

Se  $I$  è normale allora  $I \models Eq$  ( $I$  soddisfa gli assiomi di uguaglianza). Vale il contrario? No Ad esempio:

$$L = \{p, =\}, D' = \{0, 1, 2\}, p' = \{0, 2\}, =^I = \{(0, 0), (0, 2), (1, 1), (2, 0), (2, 2)\}$$

**I non è normale** dato che questa interpretazione non soddisfa la condizione  $='$  è  $\{(d, d) : d \in D'\}$  dato che ci sono coppie dell'interpretazione dell'uguaglianza in  $I$  che non sono nella forma  $\{(d, d) : d \in D'\}$ . Vale però  $I \models Eq$ .

### **Conseguenza ed equivalenza logica nella logica con uguaglianza**

Siano  $T$  e  $G$  un insieme di formule ed una formula dello stesso linguaggio  $L$ .

- $G$  è **conseguenza logica nella logica con uguaglianza** di  $T$  ( $T \models_{=} G$ ) se per ogni interpretazione normale  $I$  per  $L$  ed ogni stato  $\sigma$  di  $I$  tale che  $I, \sigma \models_{=} T$  si ha anche  $I, \sigma \models_{=} G$ .
- Se  $T = \{F\}$  diciamo che  $G$  è **conseguenza logica nella logica con uguaglianza** di  $F$  ( $F \models_{=} G$ ).
- $F$  e  $G$  sono **logicamente equivalenti nella logica con uguaglianza** ( $F \equiv_{=} G$ ) se  $F \models_{=} G$  e  $G \models_{=} F$ .



### Validità, soddisfacibilità e insoddisfacibilità nella logica con uguaglianza

- F è **valida** nella logica con uguaglianza se  $\forall I$  normale si ha  $I \models F$ ;
- F è **soddisfacibile** nella logica con uguaglianza se  $\exists I$  normale e  $\exists \sigma$  stato si ha  $I, \sigma \models F$ ;
- F è **insoddisfacibile** nella logica con uguaglianza se  $\forall I$  normale e  $\forall \sigma$  stato si ha  $I, \sigma \not\models F$ .

### Esercizio 20.14.2. mp4

$\forall x (p(x) \vee \forall y \neg r(x, y)), \exists z r(c, f(z)) \models p(c)$

- 1) Fisso  $I, \sigma$  tali che  $I, \sigma \models \forall x (p(x) \vee \forall y \neg r(x, y)), \exists z r(c, f(z))$ . Voglio che  $I, \sigma \models p(c)$ . (tip.: parti da  $\exists$ )
- 2) Dato che  $I, \sigma \models \exists z r(c, f(z))$ , Esiste  $d_0 \in D'$  tale che  $I, \sigma[z/d_0] \models r(c, f(z))$  ossia  $(c', f'(d_0)) \in r'$ .
- 3) Dato che  $I, \sigma \models \forall x (p(x) \vee \forall y \neg r(x, y))$ , in particolare  $I, \sigma[x/c'] \models p(x) \vee \forall y \neg r(x, y)$ :
- 4) Se  $I, \sigma[x/c'] \models p(x)$  allora  $c' \in p'$  e  $I, \sigma \models p(c)$  ok;
- 5) Se  $I, \sigma[x/c'] \models \forall y \neg r(x, y)$  in particolare  $(c', f'(d_0)) \notin r'$ , assurdo.

### Esercizio 20.14.2. mp4

Sia F la formula  $\exists x \exists y r(x, y)$  e sia G la formula  $\forall x \forall y (r(x, y) \rightarrow \exists z (x \neq z \wedge y \neq z \wedge \neg r(z, x) \wedge r(y, z)))$ . Mostrare che  $\{F, G\}$  è soddisfacibile nella logica con uguaglianza (definire una I normale tale che  $I \models F, G$ ).

Qual è il dominio (che deve contenere almeno un elemento)  $D' = \{0, \dots$  come interpreto la  $r' = \{\dots\}$ ?

Devono esistere due elementi in relazione tra di loro  $r(x, y)$ . proviamo con  $(0, 0)$  per soddisfare la F.

In G, interpretando  $x = 0, y = 0$ , deve esistere un elemento  $z \neq x \wedge z \neq y$  (quindi devo aggiungere un elemento al dominio  $D' = \{0, 1, \dots\}$ ), tale elemento deve stare in relazione con z (fare la coppia  $(0, 1)$ ) ma non deve stare in relazione con x (la coppia  $(1, 0) \notin r'$ ).

La F ormai è soddisfatta, avendo una coppia che la soddisfa in  $r'$ . La G necessita di qualche accorgimento in più. Per adesso G è soddisfatta per la coppia  $(0, 0)$  dato che ho uno z con tutte le proprietà che vengono richieste.

$$D' = \{0, 1 \dots\}, r' = \{(0, 0), (0, 1) \dots\}$$

Però, anche se scelgo  $x = 0, y = 1$  devo soddisfare la condizione  $\exists z (x \neq z \wedge y \neq z \wedge \neg r(z, x) \wedge r(y, z))$  e questo z deve essere diverso sia da x che da y, quindi devo aggiungere un elemento al dominio  $D' = \{0, 1, 2, \dots\}$ , tale elemento deve stare in relazione con z (fare la coppia  $(1, 2)$ ) ma non deve stare in relazione con x (la coppia  $(2, 0) \notin r'$ ).

$$D' = \{0, 1, 2 \dots\}, r' = \{(0, 0), (0, 1), (1, 2) \dots\}$$

Però, anche se scelgo  $x = 1, y = 2$  devo soddisfare la condizione  $\exists z (x \neq z \wedge y \neq z \wedge \neg r(z, x) \wedge r(y, z))$  e questo z deve essere diverso sia da x che da y. Tale elemento potrebbe essere 0? No perché significherebbe mettere la coppia  $(2, 0)$  in  $r'$ , cosa non possibile (vedi sopra), quindi devo aggiungere un elemento al dominio  $D' = \{0, 1, 2, 3, \dots\}$ , tale elemento deve stare in relazione con z (fare la coppia  $(2, 3)$ ) ma non deve stare in relazione con x (la coppia  $(3, 1) \notin r'$ ).

$$D' = \{0, 1, 2, 3 \dots\}, r' = \{(0, 0), (0, 1), (1, 2), (2, 3) \dots\}$$

Però, anche se scelgo  $x = 2, y = 3$  devo soddisfare la condizione  $\exists z (x \neq z \wedge y \neq z \wedge \neg r(z, x) \wedge r(y, z))$  e questo z deve essere diverso sia da x che da y. Tale elemento potrebbe essere 0? Significherebbe promettere  $(0, 2) \notin r'$  e aggiungere ad  $r'$  la coppia  $(3, 0)$ . A questo punto, per  $(3, 0)$  ho già uno  $z = 1$  che soddisfa le condizioni  $(3 \neq 1 \wedge 0 \neq 1 \wedge \neg r(1, 3) \wedge r(0, 1))$  ossia 1 è diverso da tre e zero, la coppia  $(1, 3)$  non sta in  $r'$  e la coppia  $(0, 1)$  sta in  $r'$ .

$$D' = \{0, 1, 2, 3\}, r' = \{(0, 0), (0, 1), (1, 2), (2, 3), (3, 0)\}$$

### Cap. 8: Trasformazione in forma prenessa

Una formula F si dice in **forma prenessa** se è **priva di quantificatori** oppure è della forma

$$Q_1 x_1 \dots Q_n x_n G$$

## Teorema

Ogni formula  $H$  può essere trasformata in una formula in forma prenessa  $K$  che è logicamente equivalente a  $H$ .

## L'algoritmo

Prende in input una formula  $H$  e, ad ogni passo produce una formula  $H'$  logicamente equivalente a  $H$ .

- Se  $H'$  è in forma prenessa allora l'algoritmo si arresta fornendo output  $H'$ .
- Se  $H'$  non è in forma prenessa allora esiste una sottoformula  $L$  di  $H'$  nella forma  $\neg Qx F$ ,  $Qx F \star G$  o  $G \star Qx F$  dove  $Q = \{\forall, \exists\}$ , e  $\star = \{\vee, \wedge, \rightarrow\}$ .

Fissata  $L$  di questo tipo si ottiene una nuova formula  $H''$  sostituendo  $L$  con una formula ad essa logicamente equivalente:

- $\neg Qx F$ :
  - $\neg \forall x F \equiv \exists x \neg F$
  - $\neg \exists x F \equiv \forall x \neg F$
- $Qx F \star G$ :
  - Se  $x$  non è libera in  $G$ :
    - $Qx F * G \equiv Qx(F * G)$
    - $G * Qx F \equiv Qx(G * F)$
  - Se  $x$  è libera in  $G$ : si sceglie una variabile  $y$  e la si usa:
    - $Qx F * G \equiv Qy(F\{x/y\} * G)$
    - $G * Qx F \equiv Qy(G * F\{x/y\})$
    - $\forall x F \rightarrow G \equiv \exists y(F\{x/y\} \rightarrow G)$
    - $G \rightarrow \forall x F \equiv \forall y(G \rightarrow F\{x/y\})$
    - $\exists x F \rightarrow G \equiv \forall y(F\{x/y\} \rightarrow G)$
    - $G \rightarrow \exists x F \equiv \exists y(G \rightarrow F\{x/y\})$
- $G \star Qx F$ :
  - Se  $x$  non è libera in  $G$ :
    - $\forall x F \rightarrow G \equiv \exists x(F \rightarrow G)$
    - $\exists x F \rightarrow G \equiv \forall x(F \rightarrow G)$
    - $G \rightarrow \forall x F \equiv \forall x(G \rightarrow F)$
    - $G \rightarrow \exists x F \equiv \exists x(G \rightarrow F)$
  - Se  $x$  è libera in  $G$ : si sceglie una variabile  $y$  e la si usa:
    - $Qx F * G \equiv Qy(F\{x/y\} * G)$
    - $G * Qx F \equiv Qy(G * F\{x/y\})$
    - $\forall x F \rightarrow G \equiv \exists y(F\{x/y\} \rightarrow G)$
    - $G \rightarrow \forall x F \equiv \forall y(G \rightarrow F\{x/y\})$
    - $\exists x F \rightarrow G \equiv \forall y(F\{x/y\} \rightarrow G)$
    - $G \rightarrow \exists x F \equiv \exists y(G \rightarrow F\{x/y\})$

## Lemma 8.35

Se  $z$  è una variabile che non occorre libera né in  $\forall x F$  (equivalentemente, in  $\exists x F$ ) né in  $\forall y G$  (equivalentemente, in  $\exists y G$ ) e le sostituzioni  $\{x/z\}$  e  $\{y/z\}$  sono ammissibili rispettivamente in  $F$  e in  $G$  allora:

$$\forall x F \wedge \forall y G \equiv \forall z(F\{x/z\} \wedge G\{y/z\})$$

$$\exists x F \vee \exists y G \equiv \exists z(F\{x/z\} \vee G\{y/z\})$$

$$\forall x F \rightarrow \exists y G \equiv \exists z(F\{x/z\} \rightarrow G\{y/z\})$$

### Nota

non è possibile operare su uno dei quantificatori su  $y$  prima di aver operato su quello su  $x$  che lo precede. Infatti  $\forall x \exists y r(x, y)$  non è logicamente equivalente a  $\exists y \forall x r(x, y)$ .

### Il p-grado

È una misura di complessità della formula, ossia uno strumento per sapere **quanti passi** servono per **raggiungere la forma prenessa**.

Data una formula  $F$  sia  $q(F)$  il numero di quantificatori che compaiono in  $F$ . Definiamo per ricorsione sulla complessità di  $F$  il p-grado  $p(F)$  di  $F$  ponendo:

- $p(F) = 0$  se  $F$  è atomica;
- $p(F) = p(G) + q(F)$  se  $F$  è  $\neg G$ ;
- $p(F) = p(G) + p(H) + q(F)$  se  $F$  è  $G \wedge H$ ,  $G \vee H$  o  $G \rightarrow H$ ;
- $p(F) = p(G)$  se  $F$  è  $\forall x G$  o  $\exists x G$ .

### Esempio di calcolo del p-grado

Calcoliamo  $p(F)$  dove  $F$  è:  $\exists x(p(x) \wedge \neg \forall y r(x, y))$ :

Chiamo  $G$  la sottoformula  $p(x) \wedge \neg \forall y r(x, y)$  ottenendo che  $p(F) = p(G)$ , dove

$$p(G) = p(p(x)) + p(\neg \forall y r(x, y)) + 1 = 0 + p(\forall y r(x, y)) + 1 + 1 = p(r(x, y)) + 2 = 2$$

### Lemma

Una formula  $F$  è in **forma prenessa** sse  $p(F) = 0$ .

### Lemma

Sia  $H$  una formula tale che nessuna sottoformula di  $H$  è di una delle seguenti forme:  $\neg Qx F$ ,  $Qx F * G$ ,  $G * Qx F$ , dove  $Q$  è uno di  $\forall$  e  $\exists$ ,  $*$  è uno di  $\wedge$ ,  $\vee$  e  $\rightarrow$ . Allora  $H$  è in **forma prenessa**.

### L'algoritmo per la forma prenessa

Avendo in input una formula  $H$ , ad ogni passo, se  $H$  non in forma prenessa, produce una formula  $H'$  logicamente equivalente ad  $H$ , altrimenti ritorna  $H$ .

Se  $H'$  non è in forma prenessa, contiene almeno una sottoformula del tipo:  $\neg Qx F$ ,  $Qx F * G$ ,  $G * Qx F$ . Fissiamo  $L$  di questo tipo e produciamo  $H''$  sostituendo dentro  $H'$ , al posto di  $L$ , una formula logicamente equivalente ad  $L$  ottenuta con uno delle equivalenze logiche notevoli.

1.  $\neg \forall x F \equiv \exists x \neg F$
2.  $\neg \exists x F \equiv \forall x \neg F$
3.  $Qx F * G \equiv Qx(F * G)$
4.  $G * Qx F \equiv Qx(G * F)$
5.  $\forall x F \rightarrow G \equiv \exists x(F \rightarrow G)$
6.  $\exists x F \rightarrow G \equiv \forall x(F \rightarrow G)$
7.  $G \rightarrow \forall x F \equiv \forall x(G \rightarrow F)$
8.  $G \rightarrow \exists x F \equiv \exists x(G \rightarrow F)$
9.  $Qx F * G \equiv Qy(F\{x/y\} * G)$
10.  $G * Qx F \equiv Qy(G * F\{x/y\})$
11.  $\forall x F \rightarrow G \equiv \exists y(F\{x/y\} \rightarrow G)$
12.  $G \rightarrow \forall x F \equiv \forall y(G \rightarrow F\{x/y\})$
13.  $\exists x F \rightarrow G \equiv \forall y(F\{x/y\} \rightarrow G)$
14.  $G \rightarrow \exists x F \equiv \exists y(G \rightarrow F\{x/y\})$

### Lemma

L'algoritmo gode della **proprietà della terminazione forte**, cioè termina qualunque sia la formula su cui si decide di operare ad ogni singolo passo.

### Esempio

Utilizziamo l'algoritmo per ottenere tre diverse formule in forma prenessa logicamente equivalenti alla formula  $\exists x p(x) \rightarrow \forall x \exists y q(x, y)$ .

$$\begin{aligned}\exists x p(x) \rightarrow \forall x \exists y q(x, y) &\equiv \forall x (p(x) \rightarrow \forall x \exists y q(x, y)) \\ &\equiv \forall x \forall z (p(x) \rightarrow \exists y q(z, y)) \\ &\equiv \forall x \forall z \exists y (p(x) \rightarrow q(z, y)). \\ \exists x p(x) \rightarrow \forall x \exists y q(x, y) &\equiv \forall x (\exists x p(x) \rightarrow \exists y q(x, y)) \\ &\equiv \forall x \forall z (p(z) \rightarrow \exists y q(x, y)) \\ &\equiv \forall x \forall z \neg y (p(z) \rightarrow q(x, y)). \\ \exists x p(x) \rightarrow \forall x \exists y q(x, y) &\equiv \forall x (\exists x p(x) \rightarrow \exists y q(x, y)) \\ &\equiv \forall x \exists y (\exists x p(x) \rightarrow q(x, y)) \\ &\equiv \forall x \exists y \forall z (p(z) \rightarrow q(x, y)).\end{aligned}$$

### Usare il minimo numero di quantificatori

Per ogni formula  $F$  e  $G$  si ha

$$\forall x F \wedge \forall x G \equiv \forall x (F \wedge G)$$

$$\exists x F \vee \exists x G \equiv \exists x (F \vee G)$$

$$\forall x F \rightarrow \exists x G \equiv \exists x (F \rightarrow G)$$

Se  $z$  è una variabile che non occorre libera né in  $\forall x F$  o  $\exists x F$  né in  $\forall y G$  o  $\exists y G$  e le sostituzioni  $\{x/z\}$  e  $\{y/z\}$  sono ammissibili in  $F$  e in  $G$  allora:

$$\forall x F \wedge \forall y G \equiv \forall z (F\{x/z\} \wedge G\{y/z\})$$

$$\exists x F \vee \exists y G \equiv \exists z (F\{x/z\} \vee G\{y/z\})$$

$$\forall x F \rightarrow \exists y G \equiv \exists z (F\{x/z\} \rightarrow G\{y/z\})$$

### Cap. 9: Traduzioni dal linguaggio naturale

Il linguaggio degli esempi sarà formato sui:

Simboli di **costante**:

$d$  Davide,  $b$  Bobi,  $f$  Fifi,  $m$  Micio

Simboli di **funzione unari**:

$p(x)$  il padrone di  $x$ ,  $m(x)$  la madre di  $x$

Simboli di **relazione unari**:

$c(x)$   $x$  è un cane,  $t(x)$   $x$  è un gatto,  $a(x)$   $x$  si arrabbia;

Simboli di **relazione binari**:

$i(x, y)$   $x$  insegue  $y$ ,  $g(x, y)$   $x$  graffia  $y$

Linguaggio naturale	Traduzione
Fifi è un cane o un gatto	$c(f) \vee t(f)$

Se Bobi insegue Micio, allora Micio graffia il padrone di Bobi	$i(b, m) \rightarrow g(b, p(b))$
Se Fifi è un gatto che graffia Micio, allora il padrone di Micio si arrabbia e insegue Fifi	$t(f) \wedge g(f, m) \rightarrow a(p(m)) \wedge i(p(m), f)$
Qualche gatto è arrabbiato	$\exists x (t(x) \wedge a(x))$
Tutti i cani sono arrabbiati	$\forall x (c(x) \rightarrow a(x))$
La madre di un gatto è un gatto	$\forall x (t(x) \rightarrow t(m(x)))$
Bobi insegue tutti i gatti arrabbiati	$\forall x (t(x) \wedge a(x) \rightarrow i(b, x))$
Bobi insegue qualche gatto arrabbiato	$\exists x (t(x) \wedge a(x) \wedge i(b, x))$
Fifi insegue i gatti che lo graffiano e li graffia	$\forall x (t(x) \wedge g(x, f) \rightarrow i(f, x) \wedge g(f, x))$
Ogni gatto è inseguito da un cane	$\forall x (t(x) \rightarrow \exists y (c(y) \wedge i(y, x)))$
Ogni cane arrabbiato insegue un gatto che lo graffia	$\forall x (c(x) \wedge a(x) \rightarrow \exists y (t(y) \wedge g(y, x) \wedge i(x, y)))$
Qualche gatto non graffia nessun cane	$\exists x (t(x) \wedge \neg \exists y (c(y) \wedge g(x, y)))$
Tutti i cani e i gatti non graffiano i loro padroni	$\forall x (t(x) \vee c(x) \rightarrow \neg g(x, p(x)))$
Ogni padrone di un cane non insegue nessun gatto	$\forall x (c(x) \rightarrow \neg \exists y (t(y) \wedge i(p(x), y)))$
Davide è il padrone di Bobi	$p(b) = d$
Bobi è l'unico cane di Davide	$c(b) \wedge p(b) = d \wedge \neg \exists x (c(x) \wedge x \neq b \wedge p(x) = d)$
Davide non è il padrone di nessun gatto	$\neg \exists x (p(x) \wedge t(x)) = d$
Davide è il padrone di (almeno) due cani	$\exists x \exists y (c(x) \wedge c(y) \wedge p(y) = d \wedge p(x) = d \wedge x \neq y)$
I cani di Davide inseguono Fifi	$\forall x (c(x) \wedge p(x) = d \rightarrow i(x, f))$
Bobi e Fifi hanno lo stesso padrone	$p(b) = p(f)$

**Nota:**

- Dopo un **quantificatore esistenziale** è normale trovare delle **coniunzioni**  $(\exists \dots (\dots \wedge \dots))$ ;
- Dopo un **quantificatore universale** è normale trovare un **implicazione**  $\forall \dots (\dots \rightarrow \dots)$ ;

## Cap. 10: Interpretazioni elementarmente equivalenti

Diciamo che due interpretazioni  $I$  e  $J$  per un linguaggio  $L$  sono **elementarmente equivalenti** se per ogni enunciato  $F$

$$I \models F \text{ sse } J \models F \text{ e si scrive } I \equiv_L J$$

E quindi **soddisfano gli stessi enunciati**.

### Esempio

$L = \{p\}$  con  $p$  simbolo di relazione unario. Le interpretazioni  $I$  e  $J$  sono elementarmente equivalenti ( $I \equiv_L J$ ):

$$I: D^I = \{0, 1\}, \quad p^I = \{0\} \equiv_L J: D^J = \{2, 3\}, \quad p^J = \{2\}$$

### Omomorfismo

Date due interpretazioni  $I$  e  $J$ , un **omomorfismo** di  $I$  in  $J$  è una funzione  $\varphi: D^I \rightarrow D^J$  tale che per ogni simbolo di:

- 1) **Costante**  $c$  si ha  $\varphi(c^I) = c^J$ ;
- 2) **Funzione**  $n$ -ario  $f$  ed ogni  $d_1 \dots d_n \in D^I$  si ha  $\varphi(f^I(d_1 \dots d_n)) = f^J(\varphi(d_1) \dots \varphi(d_n))$ ;
- 3) **Relazione**  $n$ -ario  $p$  ed ogni  $d_1 \dots d_n \in D^I$  si ha che se  $d_1 \dots d_n \in p^I$  allora  $(\varphi(d_1) \dots \varphi(d_n)) \in p^J$ .

$\varphi$  è un **omomorfismo forte** se la terza condizione è sostituita da:

- 4) **Relazione**  $n$ -ario  $p$  ed ogni  $d_1 \dots d_n \in D^I$  si ha che se  $d_1 \dots d_n \in p^I$  sse  $(\varphi(d_1) \dots \varphi(d_n)) \in p^J$ .

Per **dimostrare che un omomorfismo**  $\varphi$  è un **omomorfismo forte** è sufficiente verificare che per ogni simbolo di relazione  $n$ -ario  $p$  ed ogni elemento del dominio  $D^I$  non contenuto in  $p^I$  si ha  $(\varphi(d_1) \dots \varphi(d_n)) \notin p^J$ .

### Isomorfismo

Se  $\varphi$  è un omomorfismo forte che è anche una **biiezione** (corrispondenza biunivoca) allora  $\varphi$  è un **isomorfismo**, le due interpretazioni  $I$  e  $J$  si dicono isomorfe e scriviamo  $I \cong J$ .

Due interpretazioni isomorfe sono sostanzialmente la **stessa interpretazione** (e sono elementarmente equivalenti): ciò che **cambia** è solo il **nome degli elementi del dominio**, secondo la corrispondenza descritta dall'isomorfismo. L'essere isomorfi è una **relazione d'equivalenza** tra le interpretazioni di un linguaggio fissato.

### Esempio

Sia  $L$  il linguaggio che consiste del solo simbolo di relazione unario  $p$ . Definiamo tre interpretazioni  $I$ ,  $J$  e  $K$

$$D^I = \{A, B\}, \quad p^I = \{A\}; \quad D^J = \{0, 1\}, \quad p^J = \{1\}; \quad D^K = \mathbb{N}; \quad p^K = \mathbb{N}.$$

Alcune osservazioni:

- La funzione  $\varphi: D^I \rightarrow D^J$  definita da  $\varphi(A) = 1$  e  $\varphi(B) = 0$  è una biiezione ed è un omomorfismo forte: perciò è un isomorfismo e  $I \cong J$ .
- La funzione  $\psi: D^K \rightarrow D^I$  definita ponendo  $\psi(n) = A$  per ogni  $n \in \mathbb{N}$  è omomorfismo forte. Dato che  $D^I$  e  $D^K$  hanno cardinalità diversa, non può esistere una biiezione tra  $I$  e  $K$  e quindi non sono isomorfi.
- La funzione  $\chi: D^I \rightarrow D^K$  definita da  $\chi(A) = 7$ ,  $\chi(B) = 4$  è un omomorfismo, ma non è un omomorfismo forte.

### Lemma

Siano  $I$  e  $J$  due interpretazioni per lo stesso linguaggio  $L$  e sia  $\varphi : D^I \rightarrow D^J$  una funzione. Allo stato  $\sigma$  di  $I$  corrisponde uno stato  $\varphi \circ \sigma$  di  $J$ , ottenuto componendo  $\sigma$  e  $\varphi$ , cioè tale che per ogni variabile  $v$ ,  $(\varphi \circ \sigma)(v) = \varphi(\sigma(v))$ . Notiamo che  $\varphi \circ \sigma$  è effettivamente uno stato di  $J$ . Infatti, ricordando che  $\sigma : Var \rightarrow D^I$ , si ha che  $\varphi \circ \sigma : Var \rightarrow D^J$ .

Siano  $I$  e  $J$  due interpretazioni,  $\sigma$  uno stato di  $I$ . Se  $\varphi$  è un omomorfismo di  $I$  in  $J$  e  $t$  è un termine allora  $\varphi(\sigma(t)) = (\varphi \circ \sigma)(t)$ .

### Lemma

Siano  $I$  e  $J$  due interpretazioni,  $\sigma$  uno stato di  $I$ . Se  $\varphi$  è un omomorfismo forte di  $I$  in  $J$  e  $F$  è una formula priva di quantificatori allora

$$I, \sigma \models F \text{ sse } J, \varphi \circ \sigma \models F.$$

### Lemma

Siano  $I$  e  $J$  due interpretazioni,  $\varphi : D^I \rightarrow D^J$  una funzione qualsiasi e  $\sigma$  uno stato di  $I$ . Per ogni  $x \in Var$  e ogni  $d \in D^I$  gli stati

$$\varphi(\sigma[x/d]) \text{ e } (\varphi \circ \sigma)[x/\varphi(d)]$$

di  $J$  coincidono.

### Teorema

Siano  $I$  e  $J$  due interpretazioni per un linguaggio  $L$ ,  $\sigma$  uno stato di  $I$  e  $F$  una formula di  $L$ . Se  $\varphi$  è un omomorfismo forte suriettivo di  $I$  in  $J$  allora

$$I, \sigma \models F \text{ sse } J, \varphi \circ \sigma \models F.$$

### Esempio

Sia  $L = \{c, p, q\}$  il linguaggio con  $c$  simbolo di costante, e  $p$  e  $q$  simboli di relazione unari. Siano  $I$  e  $J$  le seguenti interpretazioni per  $L$ :

$$\begin{aligned} D^I &= \{0, 1, 2, 3\}, & c^I &= 1, & p^I &= \{0, 2, 3\}, & q^I &= \{0, 1, 2, 3\}; \\ D^J &= \{A, B, C\}, & c^J &= C, & p^J &= \{A, B\}, & q^J &= \{A, C\}. \end{aligned}$$

L'unico omomorfismo forte  $\varphi$  di  $I$  in  $J$  è dato da

- $\varphi(0) = A$  (perché  $0 \in p^I \cap q^I$  e  $A$  è l'unico elemento in  $p^J \cap q^J$ ),
- $\varphi(1) = C$  (perché  $1 = c^I$  e  $c^J = C$ ; notare anche che  $1 \notin p^I$ ,  $1 \in q^I$ ,  $C \notin p^J$ ,  $C \in q^J$ ),
- $\varphi(2) = \varphi(3) = A$  (perché  $2, 3 \in p^I \cap q^I$ ).

Dato che  $\varphi$  non è suriettivo, non possiamo dedurre  $I \equiv J$ . Infatti  $\exists x(p(x) \wedge \neg q(x))$  è vero in  $J$  ma non in  $I$  e quindi  $I \not\equiv J$ .

### Relazioni di congruenza

Sia  $I$  un'interpretazione per  $L$ . Una relazione binaria  $\sim$  sul dominio  $D^I$  di  $I$  si dice **relazione di congruenza su  $I$**  se:

- $\sim$  è una relazione di **equivalenza**;
- Per ogni simbolo di **funzione**  $n$ -ario  $f$  e per ogni  $d_1, d'_1, \dots, d_n, d'_n \in D^I$ , tali che  $d_1 \sim d'_1, \dots, d_n \sim d'_n$  vale  $f^I(d_1, \dots, d_n) \sim f^I(d'_1, \dots, d'_n)$ ;
- Per ogni simbolo di **relazione**  $n$ -ario  $p$  e per ogni  $d_1, d'_1, \dots, d_n, d'_n \in D^I$ , tali che  $d_1 \sim d'_1, \dots, d_n \sim d'_n$  vale  $(d_1, \dots, d_n) \in p^I$  sse  $(d'_1, \dots, d'_n) \in p^I$ .

La nozione di congruenza deriva il suo nome dalle relazioni di congruenza modulo  $n$  sugli interi, di cui è una generalizzazione.

### Esempio

L'esempio più semplice di relazione di congruenza è dato dalle identità. Se  $I$  è un'interpretazione qualunque, si verifica facilmente che la relazione  $\sim$  su  $D^I$  definita da:

$$d_0 \sim d_1 \text{ se e solo se } d_0 = d_1$$

è una relazione di congruenza su  $I$ .

### L'algoritmo dei tableaux nel caso predicativo

#### Differenza con il caso proposizionale

	Tableaux proposizionali	Tableaux predicativi
<b>Deterministici</b>	No	No
<b>Terminazione</b>	Sì (forte)	No
<b>Correttezza</b> (tableau chiuso, F insoddisfacibile)	Sì	?
<b>Completezza</b> (tableau aperto, F soddisfacibile)	Sì	No

#### Doppie negazioni, $\alpha$ -formule, $\beta$ -formule, $\delta$ -formule, $\gamma$ -formula

Una formula  $F$  è una **doppia negazione** se è del tipo  $\neg\neg G$  per qualche formula  $G$ . In questo caso diciamo che  $G$  è il **ridotto** di  $F$ .

Una formula è una  **$\alpha$ -formula** se esistono  $F$  e  $G$  tali che la formula è di uno dei tipi che compaiono nella colonna sinistra della seguente tabella.

$\alpha$ -formula	ridotto 1	ridotto 2
$F \wedge G$	$F$	$G$
$\neg(F \vee G)$	$\neg F$	$\neg G$
$\neg(F \rightarrow G)$	$F$	$\neg G$

Una formula è una  **$\beta$ -formula** se esistono  $F$  e  $G$  tali che la formula è di uno dei tipi che compaiono nella colonna sinistra della seguente tabella.

$\beta$ -formula	ridotto 1	ridotto 2
$F \vee G$	$F$	$G$
$\neg(F \wedge G)$	$\neg F$	$\neg G$
$F \rightarrow G$	$\neg F$	$G$



Una formula è una  **$\delta$ -formula** se esiste  $F$  tale che la formula è di uno dei tipi che compaiono nella colonna sinistra della seguente tabella. Un'istanza di una  $\delta$ -formula è una formula del tipo che compare nella colonna più a destra, dove  $a$  è un simbolo di costante. Gli **esistenziali** delle  **$\delta$ -formule** vanno sempre istanziati con **nuove costanti** mai usate prima d'ora.

$\delta$ -formula	istanza
$\exists xF$	$F\{x/a\}$
$\neg\forall xF$	$\neg F\{x/a\}$

Una formula è una  **$\gamma$ -formula** se esiste  $F$  tale che la formula è di uno dei tipi che compaiono nella colonna sinistra della seguente tabella. Un'istanza di una  $\gamma$ -formula è una formula del tipo che compare nella colonna centrale, dove  $a$  è un simbolo di costante. Possono essere istanziate con la **costante che si vuole** (anche già esistente), ma **non vengono eliminate** nel nuovo nodo.

$\gamma$ -formula	istanza 1	istanza 2
$\forall xF$	$F\{x/a\}$	$\forall xF$
$\neg\exists xF$	$\neg F\{x/a\}$	$\neg\exists xF$

#### Esempi di domande

- Dimostrare che la formula data è soddisfacibile?;
- Dimostrare che la formula data è insoddisfacibile: costruire un tableau chiuso;
- Dimostrare che la formula data è valida: costruire un tableau chiuso su  $\neg F$ ;
- Dimostrare che  $F \models G$ : costruire un tableau chiuso con la radice etichettata con  $F, \neg G$ .

## Cap. 12: La deduzione naturale predicativa

Nel presente capitolo, estendiamo il sistema deduttivo introdotto per la logica proposizionale nel capitolo 5 alla logica predicativa. Continua ad essere fondamentale l'utilizzo della formula  $\perp$ , che rappresenta il concetto di falso: nel contesto predicativo, si tratta di un enunciato atomico che non trova soddisfazione in nessuna interpretazione in nessuno stato.

Continueremo a denotare con  $T \triangleright F$  l'esistenza di una deduzione naturale in cui  $F$  è l'etichetta della radice e tutte le etichette delle foglie appartengono all'insieme  $T$ . La struttura di una deduzione come albero etichettato rimane invariata e tutte le regole della deduzione naturale proposizionale sono incorporate nel nuovo sistema. Di conseguenza, è necessario solamente aggiungere regole di introduzione ed eliminazione per i quantificatori.

### La deduzione naturale e i quantificatori

Per effettuare l'eliminazione del quantificatore universale ( $\forall$ ), dobbiamo comprendere cosa possiamo dedurre una volta che abbiamo già dedotto  $\forall x F$ . In questo caso, sappiamo che  $F$  è vero per qualsiasi elemento che vogliamo considerare: abbiamo quindi la **possibilità** di passare dall'**affermazione universale al particolare**. Poiché gli elementi sono rappresentati dai termini, è naturale poter dedurre  $F\{x/t\}$  per qualsiasi termine  $t$ , a **condizione che la sostituzione di  $x$  con  $t$  sia ammissibile in  $F$** .

#### Eliminazione $\forall$ ( $\forall e$ )

Utilizzando la notazione della sezione 5.1, possiamo quindi scrivere  $\forall x F \vdash F\{x/t\}$ , quando la sostituzione  $\{x/t\}$  è ammissibile in  $F$ . In termini di deduzione naturale, questa regola corrisponde a:

$$\frac{\forall x F}{F\{x/t\}} \quad \{x/t\} \text{ ammissibile in } F$$

#### Introduzione $\exists$ ( $\exists i$ )

L'introduzione dell'esistenziale ( $\exists$ ) è in un certo senso l'operazione inversa: se abbiamo ottenuto  $F\{x/t\}$  per qualche termine  $t$ , la cui sostituzione al posto di  $x$  è ammissibile in  $F$ , possiamo dedurre  $\exists x F$ . Infatti, il termine  $t$  rappresenta un qualche individuo, e quindi  $F\{x/t\}$  implica l'esistenza di qualcosa che soddisfa  $F$ . In termini di deduzione naturale, otteniamo:

$$\frac{F\{x/t\}}{\exists x F} \quad \{x/t\} \text{ ammissibile in } F$$

Utilizzando queste due regole per i quantificatori possiamo dare un primo esempio di deduzione naturale predicativa.

#### Esempio

Mostriamo che  $\forall x(p(x) \rightarrow q(x)), p(f(a)) \triangleright \exists x q(f(x))$

$$\frac{\frac{p(f(a)) \quad \frac{\forall x(p(x) \rightarrow q(x))}{p(f(a)) \rightarrow q(f(a))}}{q(f(a))}}{\exists x q(f(x))}$$

### Introduzione $\forall$ ( $\forall i$ )

Una tipica dimostrazione di  $\forall x F$  avviene considerando un elemento "generico" e dimostrando che  $F$  vale per quell'elemento. Ad esempio, se sosteniamo di poter aprire qualsiasi bottiglia di birra con le mani, i nostri amici ci proporranno di dimostrarlo aprendo una **specific**a bottiglia di birra (non certo tutte le bottiglie di birra mai prodotte), ma esigeranno che quella bottiglia sia "**generica**": ad esempio, non può certo essere una bottiglia già aperta e poi richiusa. Si passa quindi dal particolare al generale, ma a condizione che il particolare riguardi un elemento generico.

Gli elementi generici sono rappresentati nei linguaggi predicativi dalle variabili, ed in particolare possiamo utilizzare la variabile  $x$  stessa. Perciò è sufficiente dimostrare  $F$  a condizione che la variabile  $x$  sia "generica". Tuttavia, dobbiamo rendere rigorosa la condizione di genericità: perché  $x$  sia "generica" è necessario che nessuna ipotesi venga fatta su  $x$ . Per rendersene conto, consideriamo la seguente deduzione naturale  $\forall x(c(x) \rightarrow m(x)), p(x) \nvdash \forall m(x)$ :

$$\frac{\begin{array}{c} \text{Libera} \\ \downarrow \\ c(x) \end{array} \quad \frac{\forall x(c(x) \rightarrow m(x))}{c(x) \rightarrow m(x)}}{m(x)} \quad \frac{}{\forall m(x)}$$

Se interpretiamo  $c(x)$  come " $x$  è un cane" e  $m(x)$  come " $x$  è un mammifero", possiamo concludere che  $x$  è un mammifero a partire dalle ipotesi che ogni cane sia un mammifero e che  $x$  sia un cane. Sarebbe scorretto a questo punto concludere  $\forall x m(x)$  (cioè "tutti sono mammiferi"), proprio perché su  $x$  è stata fatta l'ipotesi che rappresenti un cane. Questa ipotesi è espressa da una formula che ha  $x$  tra le variabili libere. La condizione di genericità di  $x$  viene quindi resa precisa con la richiesta che se abbiamo  $T \vdash F$ , per poter dedurre  $\forall x F$  è necessario che  $x$  non sia libera in nessuna formula di  $T$ , cioè in nessuna ipotesi della deduzione.

Riassumendo: se

- abbiamo  $T \vdash F$ ;
- $x$  non è libera in nessuna formula di  $T$ .

allora possiamo dedurre  $\forall x F$ . Otteniamo dunque:

$$\frac{\begin{array}{c} T \\ \forall \\ F \end{array}}{\forall x F} \quad x \text{ non libera in } T$$

### Esempio

Mostriamo che  $\forall x(p(x) \rightarrow q(x)), \forall x p(x) \vdash \forall x q(x)$ :

$$\frac{\frac{\forall x p(x)}{p(x)} \quad \frac{\forall x(p(x) \rightarrow q(x))}{p(x) \rightarrow q(x)}}{q(x)} \quad \frac{}{\forall x q(x)}$$

Anche in questo caso, il ragionamento rappresentato da questa deduzione naturale è "naturale": se sappiamo che ogni elemento con la proprietà  $p$  ha anche la proprietà  $q$ , e sappiamo che tutti gli elementi hanno la proprietà  $p$ , possiamo applicare le due informazioni ad un generico  $x$ , ottenendo che esso soddisfa la proprietà  $q$ ; la genericità di  $x$  ci assicura che ogni elemento ha quest'ultima proprietà.

### Eliminazione $\exists$ ( $\exists e$ )

Rimane da considerare la regola di eliminazione di  $\exists$ . In questo caso ci può aiutare l'osservazione che la quantificazione esistenziale è una generalizzazione della disgiunzione. Perciò la regola ( $\exists e$ ) ha degli aspetti in comune con ( $\vee e$ ): in particolare è una regola che prevede lo **scaricamento**. Per sfruttare il fatto che  $\exists x F$  tipicamente

1. si assegna un **nome generico** (a.e. la variabile  $x$  stessa) all'elemento di cui si conosce l'esistenza
2. si **assume** come ipotesi ausiliaria che  $F$  valga.

Se, utilizzando l'ipotesi ausiliaria, giungiamo ad una conclusione  $G$  vorremmo scaricare  $F$  e concludere  $G$  a partire da  $\exists x F$ . Per far ciò è però necessario che  **$G$  non contenga informazioni specifiche su  $x$**  (altrimenti  $G$  non sarebbe conseguenza solo di  $\exists x F$ , ma anche del fatto che l'elemento di cui sappiamo l'esistenza sia proprio  $x$ ).

Similmente, è necessario che **le altre ipotesi  $T$  utilizzate nella deduzione naturale non contengano ipotesi su  $x$  oltre a  $F$**  (altrimenti  $x$  non sarebbe generico). Queste condizioni si traducono nel fatto che  $x$  non può essere libera né in  $G$ , né nelle formule di  $T \setminus \{F\}$ . Riassumendo:

- se da  $T, F$  possiamo dedurre  $G$
- se  $x$  non è libera in nessuna formula di  $T \setminus \{F\}$  e neppure in  $G$

allora possiamo dedurre  $G$  da  $T, \exists x F$ . Otteniamo dunque

$$\frac{\frac{T, [F]}{\nabla} \quad \frac{\exists x F \quad G}{G}}{x \text{ non libera in } T \setminus \{F\}, G}$$

### Esempio

Mostriamo che  $\forall x(p(x) \rightarrow q(x)), \exists x p(x) \triangleright \exists x q(x)$ :

$$\frac{\frac{\frac{\forall x(p(x) \rightarrow q(x))}{p(x) \rightarrow q(x)} \quad [p(x)]^1}{q(x)} \quad \frac{\exists x p(x)}{\exists x q(x)}_1}{\exists x q(x)}$$

Il ragionamento rappresentato da questa deduzione naturale inizia assegnando il nome  $x$  all'elemento con la proprietà  $p$  che sappiamo esistere. Dato che sappiamo anche che ogni elemento con la proprietà  $p$  ha la proprietà  $q$ ,  $x$  soddisfa anche  $q$  e quindi esiste un elemento con quest'ultima proprietà. Quest'ultima affermazione non riguarda  $x$  e quindi essa non dipende dall'aver usato quel nome per designare un elemento con la proprietà  $p$ .

### Riassunto delle regole

$$\begin{array}{ll} (\forall i) \quad \frac{T}{\frac{\nabla}{F}} \quad x \text{ non libera in } T & (\forall e) \quad \frac{T}{\frac{\nabla}{\forall x F}} \quad \{x/t\} \text{ ammissibile in } F \\ (\exists i) \quad \frac{T}{\frac{\nabla}{F\{x/t\}}} \quad \{x/t\} \text{ ammissibile in } F & \\ (\exists e) \quad \frac{T' \quad T, [F]}{\frac{\nabla}{\exists x F} \quad \frac{\nabla}{G}} \quad x \text{ non libera in } T \setminus \{F\}, G \end{array}$$

### Generalizzazione di ( $\forall$ i) e ( $\exists$ e)

Le regole ( $\forall$ i) e ( $\exists$ e) possono essere **generalizzate** osservando che non è necessario che l'elemento "generico" sia rappresentato proprio dalla variabile  $x$  che viene quantificata: qualunque altra variabile sufficientemente generica può essere utilizzata. In certi casi ciò è effettivamente necessario, come mostrano i seguenti esempi.

#### Esempio

Supponiamo che  $F$  sia una formula qualunque e di voler mostrare che  $\forall x(p(x) \rightarrow q(x))$ ,  $F \wedge \forall x p(x) \triangleright \forall x q(x)$ .

$$\frac{\frac{\frac{F \wedge \forall x p(x)}{\forall x p(x)} \quad \frac{\forall x(p(x) \rightarrow q(x))}{p(x) \rightarrow q(x)}}{p(x)} \quad q(x)}{\forall x q(x)}$$

Se

- $x$  non è libera in  $F$  questa deduzione naturale è corretta.
- $x$  è libera in  $F$  siamo in presenza di una violazione della condizione relativa a ( $\forall$ i) e la deduzione non è accettabile.

Un esempio del tutto analogo può essere fatto per ( $\exists$ e) mostrando che  $F \wedge \forall x(p(x) \rightarrow q(x))$ ,  $\exists x p(x) \triangleright \exists x q(x)$ . Anche in questo caso se  $x$  è libera in  $F$  la deduzione naturale non è accettabile.

$$\frac{\frac{\frac{F \wedge \forall x(p(x) \rightarrow q(x))}{\forall x(p(x) \rightarrow q(x))} \quad \frac{[p(x)]^1}{p(x) \rightarrow q(x)}}{q(x)} \quad \frac{\exists x p(x)}{\exists x q(x)} \quad 1$$

La soluzione è quella di utilizzare, per indicare l'elemento generico, una **variabile diversa** da  $x$ . Si ottengono quindi

Vecchia soluzione	Nuova soluzione
$\frac{\frac{\frac{F \wedge \forall x p(x)}{\forall x p(x)} \quad \frac{\forall x(p(x) \rightarrow q(x))}{p(x) \rightarrow q(x)}}{p(x)} \quad q(x)}{\forall x q(x)}$	$\frac{\frac{\frac{F \wedge \forall x p(x)}{\forall x p(x)} \quad \frac{\forall x(p(x) \rightarrow q(x))}{p(y) \rightarrow q(y)}}{p(y)} \quad q(y)}{\forall x q(x)}$
$\frac{\frac{\frac{F \wedge \forall x(p(x) \rightarrow q(x))}{\forall x(p(x) \rightarrow q(x))} \quad \frac{[p(x)]^1}{p(x) \rightarrow q(x)}}{q(x)} \quad \frac{\exists x p(x)}{\exists x q(x)} \quad 1$	$\frac{\frac{\frac{F \wedge \forall x(p(x) \rightarrow q(x))}{\forall x(p(x) \rightarrow q(x))} \quad \frac{[p(y)]^1}{p(y) \rightarrow q(y)}}{q(y)} \quad \frac{\exists x p(x)}{\exists x q(x)} \quad 1$

Per poter enunciare le generalizzazioni di  $(\forall i)^g$  e  $(\exists e)^g$ , è necessario precisare le condizioni sulla variabile  $y$  utilizzata per l'elemento generico. In ogni caso è chiaramente necessario che la sostituzione  $\{x/y\}$  sia ammissibile in  $F$ .

Nel caso di  $(\forall i)^g$ , se abbiamo  $T \vdash F\{x/y\}$ , per poter dedurre  $\forall x F$  è necessario che  $y$  non sia libera in nessuna formula di  $T$ . Queste condizioni non sono però ancora sufficienti per dedurre  $\forall x F$ . Per rendersene conto è bene considerare un nuovo esempio: se abbiamo ottenuto  $T \vdash r(y, y)$  con  $y$  non libera in  $T$ , è corretto dedurre  $\forall x r(x, x)$ , mentre sarebbe scorretto ottenere  $\forall x r(x, y)$ . Infatti da quest'ultima formula, con un'ulteriore applicazione di  $(\forall i)$ , è possibile ottenere  $T \vdash \forall y \forall x r(x, y)$ , che non è giustificato da  $T \vdash r(y, y)$ . In questo esempio le due occorrenze di  $y$  in  $r(y, y)$  sono considerate indipendentemente l'una dall'altra, e ciò **non è accettabile**. La condizione che è necessario aggiungere è che  $y$  non sia libera neppure in  $\forall x F$ , così che la genericità di  $y$  sia sfruttabile una sola volta. Otteniamo dunque

$$\frac{\begin{array}{c} T \\ \nabla \\ F\{x/y\} \end{array}}{\forall x F} \quad \begin{array}{l} \{x/y\} \text{ ammissibile in } F \\ y \text{ non libera in } T, \forall x F \end{array}$$

Nel caso di  $(\exists e)^g$  è necessario richiedere che  $y$  non sia libera in  $\exists x F$ , per fare in modo che essa rappresenti un elemento veramente generico. Otteniamo dunque

$$\frac{\begin{array}{c} T, [F\{x/y\}] \\ \nabla \\ \exists x F \quad G \end{array}}{G} \quad \begin{array}{l} \{x/y\} \text{ ammissibile in } F \\ y \text{ non libera in } T \setminus \{F\{x/y\}\}, \exists x F, G \end{array}$$

Notiamo che in  $(\forall i)^g$  e  $(\exists e)^g$  nulla proibisce che la variabile  $y$  sia proprio  $x$ . In questo caso la sostituzione è ovviamente ammissibile e il fatto che  $y$  non sia libera in  $\forall x F$  o in  $\exists x F$  è immediato: si ottengono quindi  $(\forall i)$  e  $(\exists e)$  come **casi particolari**.