Definizione 1.1. Sia $\mathcal{P}$ un insieme non vuoto, i cui elementi verranno chiamati *lettere proposizionali* o *formule proposizionali atomiche*. $\mathcal{P}$ determina un *linguaggio proposizionale*, le cui formule saranno costruite utilizzando gli elementi di $\mathcal{P}$ e i seguenti *simboli logici proposizionali*:

- i *connettivi proposizionali* sono i quattro simboli $\neg$, $\wedge$, $\vee$ e $\to$, che sono chiamati rispettivamente *negazione*, *congiunzione*, *disgiunzione* e *implicazione*; il primo è un *connettivo unario*, mentre i restanti tre sono *connettivi*

Definizione 1.3. L'insieme delle *formule proposizionali* (o, limitatamente ai primi cinque capitoli, formule) è definito per ricorsione come segue:

- se $p \in \mathcal{P}$ allora p è una formula;
- se F è una formula allora $(\neg F)$ è una formula;
- se F e G sono formule allora $(F \wedge G)$ è una formula;
- se F e G sono formule allora $(F \vee G)$ è una formula;
- se F e G sono formule allora $(F \to G)$ è una formula

Definizione 1.7. Una formula H è:

- una *negazione* se è della forma $(\neg F)$ per qualche formula F;
- una *congiunzione* se è della forma $(F \wedge G)$ per formule F e G, che vengono detti i *congiunti* di H;
- una *disgiunzione* se è della forma $(F \vee G)$ per formule F e G, che vengono detti i *disgiunti* di H;
- un'*implicazione* se è della forma $(F \to G)$ per formule F e G; in questo

Teorema 1.10 (Induzione sulla complessità delle formule). *Sia $\mathcal{A}$ una proprietà che può valere per le stringhe di simboli. Supponiamo di dimostrare che*

- *$\mathcal{A}(p)$ vale per ogni $p \in \mathcal{P}$;*
- *se vale $\mathcal{A}(F)$ per una formula F allora vale anche $\mathcal{A}((\neg F))$;*
- *se valgono $\mathcal{A}(F)$ e $\mathcal{A}(G)$ per formule F e G allora vale anche $\mathcal{A}((F \wedge G))$;*
- *se valgono $\mathcal{A}(F)$ e $\mathcal{A}(G)$ per formule F e G allora vale anche $\mathcal{A}((F \vee G))$;*
- *se valgono $\mathcal{A}(F)$ e $\mathcal{A}(G)$ per formule F e G allora vale anche $\mathcal{A}((F \rightarrow G))$*

DEFINIZIONE 1.12. Il *grado della formula F*, indicato con $g(F)$, è definito da:

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

ESERCIZIO 1.13. Calcolare il grado delle formule dell'esempio 1.4 e dell'esercizio 1.5.

CONVENZIONE 1.15. Nella scrittura delle formule adotteremo le seguenti *convenzioni*:

Definizione 1.19. 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.

La definizione precedente va applicata tenendo a mente la definizione 1.3 di formula, anche quando si utilizza la convenzione 1.15.

Esempio 1.20. Se F è $p \to q \vee \neg r$, allora $q \vee \neg r$ è una sottoformula di F

DEFINIZIONE 1.22. Definiamo per ricorsione sulla complessità della formula F quali sono le *sottoformule* di F:

- 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 $G \to H$ allora le sottoformule di F sono le sottoformule di G, le sottoformule di H e F stessa.

Definizione 2.1. I *valori di verità* sono $\mathbb{V}$ e $\mathbb{F}$, letti rispettivamente "vero" e "falso".

Definizione 2.2. Una *valutazione* per il linguaggio proposizionale $\mathcal{P}$ è una funzione $v : \mathcal{P} \to \{\mathbb{V}, \mathbb{F}\}$ che associa ad ogni lettera proposizionale un valore di verità.

Definizione 2.2. Una *valutazione* per il linguaggio proposizionale $\mathcal{P}$ è una funzione $v : \mathcal{P} \to \{\mathbb{V}, \mathbb{F}\}$ che associa ad ogni lettera proposizionale un valore di verità.

Una valutazione può venir estesa ad una funzione che associa un valore di verità ad ogni formula nel modo seguente.

DEFINIZIONE 2.3. Sia $v : \mathcal{P} \to \{\mathbb{V}, \mathbb{F}\}$ una valutazione. L'*interpretazione* $\bar{v}$ associa ad ogni formula F un valore di verità ed è definita per ricorsione sulla complessità delle formule nel modo seguente:

- se F è una lettera proposizionale $p \in \mathcal{P}$ allora $\bar{v}(F) = v(p)$.
- se F è $\neg G$ allora $\bar{v}(F) = \begin{cases} \mathbb{V}, & \text{se } \bar{v}(G) = \mathbb{F}; \\ \mathbb{F}, & \text{se } \bar{v}(G) = \mathbb{V}. \end{cases}$

Definizione 2.10. Se v è un'interpretazione e F una formula, diciamo che v *soddisfa* F o F *è soddisfatta da* v se $v(F) = \mathbb{V}$. Se T è un insieme di formule, diciamo che v *soddisfa* T se v soddisfa ogni formula di T.

Esercizio 2.11. Stabilire se l'interpretazione v dell'esempio 2.7 soddisfa le formule $(p \to \neg q) \vee \neg(r \wedge q)$ e $\neg(\neg p \to q) \wedge r$.

Definizione 2.12. Diciamo che le formule F e G sono *logicamente equivalenti* (in simboli $F \equiv G$) se per ogni interpretazione v si ha $v(F) = v(G)$.

Nota 2.13. L'equivalenza logica è (come suggerisce il nome) una relazione di equivalenza sull'insieme delle formule. Infatti è facile verificare riflessività ($F \equiv F$), simmetria (da $F \equiv G$ segue $G \equiv F$) e transitività (se $F \equiv G$ e $G \equiv H$ allora $F \equiv H$).

Definizione 2.15. 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.

Nota 2.16. La relazione di conseguenza logica è riflessiva e transitiva: è immediato verificare che $F \models F$, e da $F \models G$ e $G \models H$ segue $F \models H$.

Esempio 2.17. Siano F e G le formule $(p \to q) \land p$ e $p \lor q$ rispettivamente.

Definizione 2.27. Siano T e G un insieme di formule ed una formula. Diciamo che G è *conseguenza logica* di T (in simboli $T \vDash G$) se ogni interpretazione che soddisfa ogni formula di T soddisfa anche G.

Nota 2.28. È immediato che se $F \in T$ e $F \vDash G$ allora $T \vDash G$. È invece possibile che $T \vDash G$ ma che nessuna $F \in T$ sia tale che $F \vDash G$: considerare ad esempio $T = \{p, q\}$, $G = p \wedge q$.

Definizione 2.33. Se F è una formula diciamo che:

- F è *valida* se F è soddisfatta da ogni interpretazione;
- F è *soddisfacibile* se F è soddisfatta da qualche interpretazione;
- F è *insoddisfacibile* se non esiste un'interpretazione che soddisfa F.

Ovviamente una formula è insoddisfacibile se e solo se non è soddisfacibile. Notiamo anche che ogni formula valida è soddisfacibile (ma il viceversa non vale).

TEOREMA 2.37. *Sia F una formula:*

(a) *F è valida se e solo se $\neg F$ è insoddisfacibile;*
(b) *F è insoddisfacibile se e solo se $\neg F$ è valida.*

DIMOSTRAZIONE. (a) F è valida se e solo se per ogni interpretazione v si ha $v(F) = \mathbb{V}$, se e solo se per ogni interpretazione v si ha $v(\neg F) = \mathbb{F}$, se e solo se $\neg F$ è insoddisfacibile.

DEFINIZIONE 2.42. Se T è un insieme di formule diciamo che:

- T è *valido* se ogni interpretazione soddisfa T, cioè soddisfa ogni $F \in T$;
- T è *soddisfacibile* se qualche interpretazione soddisfa T, cioè soddisfa ogni $F \in T$;
- T è *insoddisfacibile* se ogni interpretazione non soddisfa T, cioè non soddisfa qualche $F \in T$ (F può dipendere dall'interpretazione).

Definizione 2.48. Sia $\mathcal{F}$ un insieme di formule proposizionali. Una *procedura di decisione per* $\mathcal{F}$ è un algoritmo che riceve in input una formula F, termina sempre la sua esecuzione e fornisce l'output "sì" se $F \in \mathcal{F}$, l'output "no" se $F \notin \mathcal{F}$.

Un obiettivo della logica è individuare procedure di decisione per gli insiemi delle formule valide e delle formule soddisfacibili. Il teorema 2.37 mostra che una procedura di decisione per uno qualunque di questi insiemi può essere facilmente

Algoritmo 2.49. L'algoritmo delle tavole di verità prende in input una formula proposizionale F e esamina le lettere proposizionali che compaiono in F, che indichiamo con $p_1, \dots, p_n$. Si crea una tabella che contiene $n+1$ colonne, una per ogni p_i ed una per F, e 2^n righe. Le prime n colonne (quelle delle p_i) contengono $\mathbb{V}$ o $\mathbb{F}$ in modo che nelle 2^n righe compaia ogni possibile funzione che associa alle n lettere proposizionali i valori di verità (ciò si ottiene ad esempio se nella prima

DEFINIZIONE 3.1. Un *letterale* è una lettera proposizionale oppure la negazione di una lettera proposizionale.

DEFINIZIONE 3.2. 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.

La proprietà fondamentale delle coppie complementari di letterali è contenuta

DEFINIZIONE 3.2. 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.

La proprietà fondamentale delle coppie complementari di letterali è contenuta nel seguente lemma.

LEMMA 3.3. *Un insieme di letterali è soddisfacibile se e solo se non contiene*

Definizione 3.6. Una formula proposizionale è in *forma normale congiuntiva* se è della forma $F_1 \wedge \cdots \wedge F_m$, dove per $1 \leq i \leq m$, F_i è della forma $G_{i,1} \vee \cdots \vee G_{i,h_i}$, dove per $1 \leq j \leq h_i$, $G_{i,j}$ è un letterale.

Una formula proposizionale è in *forma normale disgiuntiva* se è della forma $F_1 \vee \cdots \vee F_m$, dove per $1 \leq i \leq m$, F_i è della forma $G_{i,1} \wedge \cdots \wedge G_{i,h_i}$, dove per $1 \leq j \leq h_i$, $G_{i,j}$ è un letterale.

Teorema 3.10. *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*

$$F \equiv G_1 \qquad e \qquad F \equiv G_2.$$

L'espressione "può essere trasformata" nell'enunciato del teorema è stata usata per asserire qualcosa di più della semplice esistenza di G_1 e G_2: intendiamo dire

DEFINIZIONE 3.12. 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.

Usando le equivalenze logiche del lemma 2.24 si può mostrare che ogni formula che non è un letterale è una doppia negazione oppure è logicamente equivalente ad una congiunzione o ad una disgiunzione. Per formulare questa osservazione in forma compatta (come faremo nei lemmi 3.14 e 3.15) introduciamo la seguente

Definizione 3.13. Una formula è *una α-formula* se esistono F e G tali che la formula è di uno dei tipi che compaiono nella colonna sinistra della prima delle seguenti tabelle. Una formula è *una β-formula* se esistono F e G tali che la formula è di uno dei tipi che compaiono nella colonna sinistra della seconda delle seguenti tabelle. In entrambi i casi i *ridotti* di una α- o β-formula sono le formule che compaiono nelle due colonne più a destra.

α-formula	ridotti

β-formula	ridotti

Convenzione 3.17. Visto che per i lemmi 2.20.2 e 2.24.6 $\wedge$ è sia commutativo che associativo tutte le formule ottenute combinando in un ordine qualunque $G_1, \ldots, G_n$ per mezzo di $\wedge$ sono logicamente equivalenti a $G_1 \wedge \cdots \wedge G_n$ (che, per la convenzione 1.15.4 sulla scrittura delle formule, è $(((G_1 \wedge G_2) \wedge \ldots G_{n-1}) \wedge G_n))$. Quest'ultima formula è chiamata la *congiunzione generalizzata* di $G_1, \ldots, G_n$ e in questo capitolo la scriviamo $\langle G_1, \ldots, G_n \rangle$.

Analogamente, a partire da commutatività e associatività di $\vee$, si definisce

Algoritmo 3.18. 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 dell'algoritmo avremo una congiunzione generalizzata di disgiunzioni generalizzate di formule. Se tutti gli elementi di queste disgiunzioni generalizzate sono letterali la formula è in forma normale congiuntiva e l'algoritmo si arresta. Se esistono elementi di queste disgiunzioni generalizzate che non sono letterali se ne

ALGORITMO 3.22. 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: $[\langle F \rangle]$. Ad ogni passo dell'algoritmo avremo una disgiunzione generalizzata di congiunzioni generalizzate di formule. Se tutti gli elementi di queste congiunzioni generalizzate sono

Definizione 3.31. Associamo ad ogni formula proposizionale un numero naturale positivo che chiameremo rango, secondo la seguente definizione per ricorsione sul numero dei simboli che compaiono nella formula:

- se F è un letterale allora $\mathrm{rg}(F) = 1$;
- se F è una doppia negazione con ridotto G allora $\mathrm{rg}(F) = \mathrm{rg}(G) + 1$;
- se F è una α-formula o una β-formula con ridotti G e H allora $\mathrm{rg}(F) = \mathrm{rg}(G) + \mathrm{rg}(H) + 1$.

ALGORITMO 4.5. Un tableau per una formula F è un albero in cui ogni nodo è etichettato con un insieme finito di formule. Il tableau è costruito per stadi $\mathcal{T}_0, \ldots, \mathcal{T}_k$: per ogni i, $\mathcal{T}_{i+1}$ è un albero che estende $\mathcal{T}_i$ aggiungendo uno o due nodi con le rispettive etichette e lasciando invariate le etichette dei nodi già appartenenti

Definizione 4.6. Sia n un nodo del tableau che non è una foglia: la *formula su cui si agisce in n* è la G della descrizione dell'algoritmo. Notiamo che G non appartiene all'etichetta di nessuno dei nodi successori di n.

Convenzione 4.7. Per comodità di lettura aggiungeremo sotto le foglie del tableau uno dei simboli $\bigotimes$ e $\bigcirc$: se l'etichetta della foglia contiene una coppia complementare di letterali useremo $\bigotimes$, altrimenti $\bigcirc$. Questo ci permette di vedere

Convenzione 4.7. Per comodità di lettura aggiungeremo sotto le foglie del tableau uno dei simboli $\bigotimes$ e $\bigcirc$: se l'etichetta della foglia contiene una coppia complementare di letterali useremo $\bigotimes$, altrimenti $\bigcirc$. Questo ci permette di vedere facilmente se il tableau contiene solo foglie etichettate con insiemi insoddisfacibili di letterali, oppure se c'è qualche foglia etichettata con un insieme soddisfacibile di letterali (stiamo usando il lemma 3.3). Inoltre, per alleggerire la notazione, evitiamo di indicare le parentesi { e } intorno agli elementi di $E(n)$.

Teorema 4.11. *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.*

Dimostrazione. Se per assurdo la costruzione di un tableau $\mathcal{T}$ non terminasse, essa darebbe origine ad un albero binario infinito. Per il lemma 4.10 un tale albero avrebbe un ramo infinito. Per raggiungere una contraddizione assegnamo

Definizione 4.15. Un tableau è *chiuso* se tutte le sue foglie sono etichettate con insiemi di letterali che contengono una coppia complementare. Un tableau 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* di un tableau è un ramo che collega la radice dell'albero con una foglia etichettata con un insieme di letterali che non contiene una coppia complementare.

Teorema 4.16. *Sia F una formula e $\mathcal{T}$ un tableau per F. F è insoddisfacibile se e solo se $\mathcal{T}$ è chiuso.*

Enunciamo immediatamente due importanti e utili conseguenze del teorema 4.16.

Teorema 4.17. *Sia F una formula e $\mathcal{T}$ un tableau per F. F è soddisfacibile se e solo se $\mathcal{T}$ è aperto.*

Teorema 4.17. *Sia F una formula e $\mathcal{T}$ un tableau per F. F è soddisfacibile se e solo se $\mathcal{T}$ è aperto.*

Dimostrazione. Immediata dal teorema 4.16. $\square$

Teorema 4.18. *Sia F una formula e $\mathcal{T}$ un tableau per $\neg F$. F è valida se e solo se $\mathcal{T}$ è chiuso.*

Teorema 4.18. *Sia F una formula e $\mathcal{T}$ un tableau per $\neg F$. F è valida se e solo se $\mathcal{T}$ è chiuso.*

Dimostrazione. F è valida se e solo se $\neg F$ è insoddisfacibile (teorema 2.37), se e solo se $\mathcal{T}$ è chiuso (teorema 4.16). □

Nota 4.19. Si potrebbe essere tentati di pensare che F sia valida se e solo se

Corollario 4.20. *Sia F una formula e $\mathcal{T}$ e $\mathcal{T}'$ due tableaux per F. $\mathcal{T}$ e $\mathcal{T}'$ sono entrambi chiusi o entrambi aperti.*

Dimostrazione. Se F è soddisfacibile $\mathcal{T}$ e $\mathcal{T}'$ sono entrambi aperti per il teorema 4.17. Se F è insoddisfacibile $\mathcal{T}$ e $\mathcal{T}'$ sono entrambi chiusi per il teorema 4.16. □

Dobbiamo ora dimostrare il teorema 4.16. È opportuno distinguere le due

Teorema 4.21 (Teorema di correttezza). *Se un tableau per la formula F è chiuso allora F è insoddisfacibile.*

Dimostrazione. Fissiamo F e $\mathcal{T}$, tableau chiuso per F. Dimostreremo il seguente fatto, che indichiamo con ($\star$):

per ogni nodo n di $\mathcal{T}$ l'insieme di formule $E(n)$ è insoddisfacibile.

Il caso particolare di ($\star$) ottenuto quando n è la radice di $\mathcal{T}$ (e quindi $E(n) = \{F\}$)

Teorema 4.22 (Teorema di completezza). *Se un tableau per la formula F è aperto allora F è soddisfacibile.*

Schema della dimostrazione. Fissiamo un ramo aperto r di un tableau aperto per F. La dimostrazione si svilupperà in tre passi:

(a) definiremo cosa significa che un insieme di formule è un insieme di Hintikka[4] (definizione 4.23);

Definizione 4.23. Un insieme di formule $\mathcal{H}$ è un *insieme di Hintikka* se soddisfa le seguenti quattro condizioni:

(1) $\mathcal{H}$ non contiene coppie complementari di letterali;
(2) se $G \in \mathcal{H}$ è una doppia negazione con ridotto G_1 allora $G_1 \in \mathcal{H}$;
(3) se $G \in \mathcal{H}$ è una α-formula con ridotti G_1 e G_2 allora $G_1 \in \mathcal{H}$ e $G_2 \in \mathcal{H}$;
(4) se $G \in \mathcal{H}$ è una β-formula con ridotti G_1 e G_2 allora $G_1 \in \mathcal{H}$ oppure $G_2 \in \mathcal{H}$ (è possibile che entrambi siano in $\mathcal{H}$).

Convenzione 4.31. Da questo punto in poi se nell'etichetta di un nodo di un tableau deve comparire una doppia negazione F con ridotto G scriviamo direttamente G, utilizzando la regola della doppia negazione immediatamente e contraendo due nodi in uno.

Nel tableau dell'esempio 4.30 la precedente convenzione avrebbe permesso di scrivere un nodo in meno (il secondo dalla radice, in cui si è agito su $\neg\neg p$).

Convenzione 4.32. Da questo punto in poi se nell'etichetta di un nodo di un tableau compaiono una formula F e la sua negazione $\neg F$ possiamo considerare il nodo in questione chiuso e non operare più su di esso (secondo l'algoritmo 4.5 dovremmo ancora operare su di esso se l'etichetta contiene formule, compresa eventualmente F, che non sono letterali).

Nel tableau dell'esempio 4.30 la precedente convenzione avrebbe permesso di

Algoritmo 4.39. 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 (e l'etichetta priva di coppie complementari di una foglia ci permette di definire un'interpretazione che soddisfa T come nel lemma 4.29), se il tableau è chiuso T è insoddisfacibile.

Per quanto riguarda la validità di $\{F_1, \dots, F_n\}$ il lemma 2.43 e il teorema 4.18

Algoritmo 4.40. Per stabilire se $F_1, \ldots, F_n \vDash G$ costruiamo un tableau la cui radice è etichettata con $\{F_1, \ldots, F_n, \neg G\}$. Se il tableau è chiuso $F_1, \ldots, F_n \vDash G$, se il tableau è aperto $F_1, \ldots, F_n \nvDash G$ (e l'etichetta priva di coppie complementari di una foglia ci permette di definire un'interpretazione che soddisfa $F_1, \ldots, F_n$ ma non G come nel lemma 4.29).

Esercizio 4.41. Studiate con il metodo del tableaux le conseguenze logiche

Definizione 5.2. La formula $\bot$ è una *costante logica* che appartiene all'insieme $\mathcal{P}$ delle *lettere proposizionali* e viene letta "falso" o "contraddizione".

Dal punto di vista della semantica, per ogni interpretazione v si ha $v(\bot) = \mathbb{F}$.

Nota 5.3. La definizione precedente implica che $\bot$ è insoddisfacibile e (quindi) che $\neg\bot$ è valida. Possiamo anzi considerare $\bot$ e $\neg\bot$ come i "prototipi" rispettivamente delle formule insoddisfacibili e delle formule valide. L'esercizio 2.35 si applica

Teorema 5.17 (Teorema di correttezza). *Siano T un insieme di formule proposizionali e F una formula proposizionale. Se $T \rhd F$ allora $T \models F$.*

Dimostrazione. La dimostrazione è per induzione sull'altezza dell'albero della deduzione naturale che testimonia $T \rhd F$. In altre parole, dimostriamo per induzione su n che se l'altezza di un albero di deduzione che mostra $T \rhd F$ è n, allora $T \models F$.

Teorema 5.18 (Teorema di completezza). *Siano T un insieme di formule proposizionali e F una formula proposizionale. Se $T \vDash F$ allora $T \rhd F$.*

Il teorema di completezza asserisce che tutto ciò che è vero è dimostrabile per mezzo della deduzione naturale, che è quindi "completa".

Notiamo che il nostro sistema di deduzione naturale soddisfa anche la componibilità delle deduzioni che avevamo formulato a pagina 42.

Teorema 5.20. *Esistono $a, b \in \mathbb{R}$ tali che a e b sono irrazionali mentre a^b è razionale.*

Dimostrazione. Sia $b = \sqrt{2}$: è noto sin dall'antica Grecia che b è irrazionale. Per individuare a procediamo per casi:

- se $\sqrt{2}^{\sqrt{2}}$ è razionale allora prendiamo $a = \sqrt{2}$;
- se $\sqrt{2}^{\sqrt{2}}$ è irrazionale allora prendiamo $a = \sqrt{2}^{\sqrt{2}}$

Definizione 6.1. Un linguaggio predicativo contiene i seguenti elementi comuni:

- un insieme infinito numerabile Var di *variabili*; tipicamente indicheremo le variabili con le lettere $x, y, z, \ldots$;
- i *simboli logici*: i *connettivi* già visti nel caso proposizionale $\neg$, $\wedge$, $\vee$ e $\rightarrow$ e i *quantificatori* $\forall$ e $\exists$
- la virgola “,” e le parentesi “(” e “)”

Definizione 6.2. Un *linguaggio predicativo* consiste dei seguenti insiemi (che è sempre conveniente siano disgiunti, per evitare di incorrere in confusioni):

- un insieme di *simboli di costante*;
- un insieme di *simboli di funzione*, ciascuno fornito della propria *arietà*, che è un numero naturale ≥ 1;
- un insieme non vuoto di *simboli di relazione* (o *simboli predicativi*), cia-

Definizione 6.9. Sia $\mathcal{L}$ un linguaggio fissato. L'insieme dei *termini di* $\mathcal{L}$ è definito per ricorsione come segue:

- ogni variabile è un termine di $\mathcal{L}$;
- ogni simbolo di costante di $\mathcal{L}$ è un termine di $\mathcal{L}$;
- se $t_1, \dots t_n$ sono termini di $\mathcal{L}$ e f è un simbolo di funzione n-ario di $\mathcal{L}$, allora $f(t_1, \dots, t_n)$ è un termine di $\mathcal{L}$.

TEOREMA 6.17 (Induzione sulla complessità dei termini). *Sia $\mathcal{A}$ una proprietà che può valere per le stringhe di simboli di un certo linguaggio $\mathcal{L}$. Supponiamo di dimostrare che*

- *$\mathcal{A}(x)$ vale per ogni $x \in \mathrm{Var}$;*
- *$\mathcal{A}(c)$ vale per ogni simbolo di costante c di $\mathcal{L}$;*
- *per ogni simbolo di funzione n-ario f di $\mathcal{L}$, se $t_1, \dots t_n$ sono termini di $\mathcal{L}$ per cui valgono $\mathcal{A}(t_1), \dots, \mathcal{A}(t_n)$ allora vale anche $\mathcal{A}(f(t_1, \dots, t_n))$*

DEFINIZIONE 6.18. Se x è una variabile e s e t sono termini definiamo la *sostituzione di x con t in s*, $s\{x/t\}$, per ricorsione sulla complessità di s:

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

Definizione 6.20. Sia $\mathcal{L}$ un linguaggio predicativo. Le *formule atomiche di* $\mathcal{L}$ sono le stringhe di simboli del tipo $p(t_1, \dots, t_n)$ dove p è un simbolo di relazione n-ario e $t_1, \dots, t_n$ sono termini.

L'insieme delle *formule di* $\mathcal{L}$ è definito per ricorsione come segue:

- ogni formula atomica di $\mathcal{L}$ è una formula di $\mathcal{L}$;
- se F è una formula di $\mathcal{L}$ allora $(\neg F)$ è una formula di $\mathcal{L}$;
- se F e G sono formule di $\mathcal{L}$ allora $(F \wedge G)$, $(F \vee G)$ e $(F \rightarrow G)$ sono

Convenzione 6.25. Nella scrittura delle formule adotteremo le seguenti *convenzioni*:

- si omettono le parentesi più esterne;
- $\neg$, $\forall$ e $\exists$ hanno la precedenza su $\wedge$, $\vee$ e $\to$, così che $\forall x\, F \to G$ abbrevia $((\forall x\, F) \to G)$;
- $\wedge$ e $\vee$ hanno la precedenza su $\to$;

Definizione 6.27. Se T è un insieme di formule il *linguaggio di* T è indicato con $\mathcal{L}(T)$ e consiste dei simboli di costante, funzione e relazione che compaiono in qualche elemento di T. Se $T = \{F\}$ scriveremo $\mathcal{L}(F)$, e similmente $\mathcal{L}(F, G)$ sta per $\mathcal{L}(\{F, G\})$.

Esempio 6.28. Se F è

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

Teorema 6.34 (Induzione sulla complessità delle formule). *Sia $\mathcal{A}$ una proprietà che può valere per le stringhe di simboli di un certo linguaggio $\mathcal{L}$. Supponiamo di dimostrare che*

- *$\mathcal{A}(F)$ vale per ogni formula atomica F di $\mathcal{L}$;*
- *se vale $\mathcal{A}(F)$ per una formula F allora vale anche $\mathcal{A}(\neg F)$;*
- *se valgono $\mathcal{A}(F)$ e $\mathcal{A}(G)$ per formule F e G allora valgono anche $\mathcal{A}(F \wedge G)$, $\mathcal{A}(F \vee G)$ e $\mathcal{A}(F \to G)$;*

Definizione 6.35. Il *grado della formula F*, indicato con $g(F)$, è definito da:

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

Esercizio 6.36. Calcolare il grado delle formule dell'esempio 6.22.

Definizione 6.38. Sia F una formula e x una variabile. Definiamo le *occorrenze libere di x in F* come segue:

- se F è atomica allora ogni occorrenza di x in F è libera;
- se F è $\neg G$ allora le occorrenze libere di x in F sono le occorrenze libere di x in G;
- se F è $G \wedge H$, $G \vee H$ oppure $G \to H$, allora le occorrenze libere di x in F sono le occorrenze libere di x in G e le occorrenze libere di x in H;

DEFINIZIONE 6.43.

- Una *formula priva di quantificatori* è una formula in cui non compaiono $\forall$ e $\exists$;
- un *letterale* è una formula atomica o la negazione di una formula atomica.

Se identifichiamo le formule atomiche della logica predicativa con le lettere proposizionali della logica proposizionale, le formule prive di quantificatori sono le

Definizione 6.44. Sia F una formula con variabili libere $x_1, \ldots, x_n$. L'enunciato $\forall x_1 \ldots \forall x_n F$ è una *chiusura universale* di F, mentre l'enunciato $\exists x_1 \ldots \exists x_n F$ è una *chiusura esistenziale* di F.

Convenzione 6.45. Due chiusure universali (esistenziali) di F differiscono per l'ordine in cui le variabili libere vengono quantificate (ad esempio $\forall x \forall y\, r(x, y)$ e $\forall y \forall x\, r(x, y)$ sono (le uniche) due chiusure universali della formula $r(x, y)$). Dato

Convenzione 6.45. Due chiusure universali (esistenziali) di F differiscono per l'ordine in cui le variabili libere vengono quantificate (ad esempio $\forall x\, \forall y\, r(x, y)$ e $\forall y\, \forall x\, r(x, y)$ sono (le uniche) due chiusure universali della formula $r(x, y)$). Dato che il corollario 7.31 mostrerà che tutte le chiusure universali di una formula F sono

Definizione 6.46. 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.

La definizione precedente va applicata tenendo a mente la definizione 6.20 di formula, anche quando si utilizza la convenzione 6.25.

Esempio 6.47. Se F è

Definizione 6.49. Definiamo per ricorsione sulla complessità della formula F quali sono le *sottoformule* di 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 $G \wedge H$, $G \vee H$ oppure $G \rightarrow H$ allora le sottoformule di F sono le sottoformule di G, le sottoformule di H e F stessa.

Definizione 6.50. Se F è una formula atomica $p(s_1, \dots, s_k)$, x è una variabile e t è un termine la *sostituzione di x con t in F* è $p(s_1\{x/t\}, \dots, s_k\{x/t\})$ ed è denotata da $F\{x/t\}$.

Esempio 6.51. $q(x, g(x, f(y)), f(g(z, y)))\{x/f(w)\}$ è

$$q(f(w), g(f(w), f(y)), f(g(z, y))).$$

Definizione 6.52. Un termine t è *libero per la sostituzione* al posto di un'occorrenza libera della variabile x nella formula F, se t non contiene alcuna variabile y tale che esiste una sottoformula di F contenente l'occorrenza di x che stiamo considerando ed è del tipo $\forall y\, G$ o $\exists y\, G$.

Nota 6.53. Se t non contiene variabili diverse da x allora t è libero per la sostituzione al posto di ogni occorrenza libera di x in ogni formula F. In particolare

Definizione 6.55. 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. In questo caso la formula $F\{x/t\}$ è ottenuta sostituendo in F ogni formula atomica A in cui x compare libera con $A\{x/t\}$.

La nozione di sostituzione ammissibile è cruciale per lo sviluppo della teoria. Senza di essa non è possibile formulare e dimostrare il Lemma di Sostituzione

Definizione 6.59. Sia F una formula, x una variabile qualunque e y una variabile che non occorre in F. La *variante* di F in cui x è rimpiazzata da y è indicata con $F_x(y)$ ed è la formula in cui tutte le occorrenze legate di x (comprese quelle che seguono immediatamente un quantificatore) sono sostituite da y.

L'idea è che una variante di F ha lo stesso significato di F (questa affermazione sarà giustificata dal corollario 7.62), così come $\int_0^1 f(x, z)\, dx$ e $\int_0^1 f(y, z)\, dy$ sono la

DEFINIZIONE 6.63. Un linguaggio $\mathcal{L}$ si dice un *linguaggio con uguaglianza* se tra i suoi simboli di relazione binari ne esiste uno denotato con $=$. In questo caso se t_1 e t_2 sono termini di $\mathcal{L}$ la formula atomica $=(t_1, t_2)$ viene scritta $t_1 = t_2$ e il letterale $\neg t_1 = t_2$ viene scritto $t_1 \neq t_2$.

Per cercare di descrivere il ruolo svolto da $=$ in un linguaggio con uguaglianza possiamo definire, per ora a livello puramente sintattico, un insieme di enunciati che

Definizione 6.64. Dato un linguaggio $\mathcal{L}$, con $\mathsf{Eq}_{\mathcal{L}}$ denotiamo l'insieme dei seguenti enunciati di $\mathcal{L} \cup \{=\}$, spesso chiamati *assiomi dell'uguaglianza*:

(e1) $\forall x\, x = x$;

(e2) $\forall x\, \forall y (x = y \to y = x)$;

(e3) $\forall x\, \forall y\, \forall z (x = y \wedge y = z \to x = z)$;

(e4) $\forall x_1 \ldots \forall x_n\, \forall y_1 \ldots \forall y_n (x_1 = y_1 \wedge \cdots \wedge x_n = y_n \to f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n))$ per tutti i simboli di funzione f di $\mathcal{L}$, dove n è l'arietà di f

DEFINIZIONE 7.1. Dato un linguaggio $\mathcal{L}$ una *interpretazione* I per $\mathcal{L}$ è data da:

- un insieme non vuoto D^I, detto *dominio* dell'interpretazione;
- per ogni simbolo di costante c in $\mathcal{L}$, un elemento $c^I \in D^I$;
- per ogni simbolo di funzione n-ario f in $\mathcal{L}$, una funzione $f^I : (D^I)^n \to D^I$;
- per ogni simbolo di relazione n-ario p in $\mathcal{L}$, un insieme $p^I \subseteq (D^I)^n$.

Data un'interpretazione, la useremo per interpretare prima i termini e poi le

DEFINIZIONE 7.2. Un'interpretazione I per il linguaggio $\mathcal{L}$ associa ad ogni termine chiuso t di $\mathcal{L}$ 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

Definizione 7.3. Uno *stato* dell'interpretazione I è una funzione $\sigma : \mathrm{Var} \to D^I$ che ad ogni variabile associa un elemento del dominio di I.

Estendiamo uno stato all'insieme di tutti i termini attraverso una definizione ricorsiva che ha delle analogie con quanto fatto nella definizione 2.3 per estendere una valutazione ad un'interpretazione. Uno stato σ di I associa ad ogni termine t un *valore*, $\sigma(t) \in D^I$, definito per ricorsione sulla complessità di t da:

- se t è una variabile x allora $\sigma(t) = \sigma(x)$;

Definizione 7.8. Siano F una formula di un linguaggio $\mathcal{L}$, I un'interpretazione per $\mathcal{L}$ e σ uno stato di I. Definiamo la relazione $I, \sigma \vDash F$ (da leggersi I *allo stato* σ *soddisfa* F) per ricorsione sulla complessità di F ($I, \sigma \nvDash F$ indica che $I, \sigma \vDash F$ non vale):

- $I, \sigma \vDash p(t_1, \dots, t_n)$ se e solo se $(\sigma(t_1), \dots, \sigma(t_n)) \in p^I$;
- $I, \sigma \vDash \neg G$ se e solo se $I, \sigma \nvDash G$;
- $I, \sigma \vDash G \wedge H$ se e solo se $I, \sigma \vDash G$ e $I, \sigma \vDash H$;

Definizione 7.9. Diciamo che I *soddisfa* F, e scriviamo $I \models F$ se $I, \sigma \models F$ per ogni stato σ di I. In questo caso si dice anche che F *è vera in* I oppure che I *è un modello di* F.

Definizione 7.10. Se T è un insieme di formule, diciamo che I *allo stato* σ *soddisfa* T, e scriviamo $I, \sigma \models T$, se I allo stato σ 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*

Definizione 7.10. Se T è un insieme di formule, diciamo che I *allo stato* σ *soddisfa* T, e scriviamo $I, \sigma \vDash T$, se I allo stato σ 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 $I \vDash T$, se $I, \sigma \vDash T$ per ogni stato σ di I.

Questa definizione va vista come una generalizzazione al caso predicativo della definizione 2.3. Pur tenendo conto della differenza tra le interpretazioni pro-

Definizione 7.27. Siano F e G due formule. Diciamo che F e G sono *logicamente equivalenti* (in simboli $F \equiv G$) se per ogni interpretazione I per $\mathcal{L}(F, G)$ e ogni stato σ di I si ha $I, \sigma \vDash F$ se e solo se $I, \sigma \vDash G$.

Definizione 7.28. Siano F e G due formule. Diciamo che G è *conseguenza logica* di F (in simboli $F \vDash G$) se per ogni interpretazione I per $\mathcal{L}(F, G)$ e ogni stato σ di I tali che $I, \sigma \vDash F$ si ha $I, \sigma \vDash G$.

DEFINIZIONE 7.28. Siano F e G due formule. Diciamo che G è *conseguenza logica* di F (in simboli $F \vDash G$) se per ogni interpretazione I per $\mathcal{L}(F, G)$ e ogni stato σ di I tali che $I, \sigma \vDash F$ si ha $I, \sigma \vDash G$.

DEFINIZIONE 7.29. Siano T e G un insieme di formule ed una formula. Diciamo che G è *conseguenza logica* di T (e scriviamo $T \vDash G$) se per ogni interpretazione I per $\mathcal{L}(T, G)$ ed ogni stato σ di I tale che $I, \sigma \vDash T$ si ha $I, \sigma \vDash G$. Come nel caso

Definizione 7.29. Siano T e G un insieme di formule ed una formula. Diciamo che G è *conseguenza logica* di T (e scriviamo $T \vDash G$) se per ogni interpretazione I per $\mathcal{L}(T, G)$ ed ogni stato σ di I tale che $I, \sigma \vDash T$ si ha $I, \sigma \vDash G$. Come nel caso proposizionale, $\vDash F$ sta ad indicare $\emptyset \vDash F$.

Nota 7.30. Lo stesso simbolo $\vDash$ viene usato per denotare sia la nozione di soddisfazione (definizioni 7.8, 7.9 e 7.10) che quella di conseguenza logica (definizioni

Corollario 7.31. *Due chiusure universali della stessa formula sono logicamente equivalenti. Due chiusure esistenziali della stessa formula sono logicamente equivalenti.*

Esempio 7.32. Siano F e G le formule dell'esercizio 7.19. Dimostriamo che $F \models G$ e che quindi non può esistere un'interpretazione in cui F è vera ma G è falsa.

Definizione 7.43. Se F è una formula diciamo che

- F è *valida* se per ogni interpretazione I per $\mathcal{L}(F)$ e ogni stato σ di I si ha $I, \sigma \models F$;
- F è *soddisfacibile* se esistono un'interpretazione I per $\mathcal{L}(F)$ e uno stato σ di I tali che $I, \sigma \models F$;
- F è *insoddisfacibile* se per ogni interpretazione I per $\mathcal{L}(F)$ e ogni stato σ di I si ha $I, \sigma \not\models F$.

TEOREMA 7.48. *Sia F una formula:*

(a) *F è valida se e solo se $\neg F$ è insoddisfacibile;*
(b) *F è insoddisfacibile se e solo se $\neg F$ è valida.*

ESERCIZIO 7.49. Verificate che i lemmi 2.40, 2.43 e 2.47 valgono anche se le formule coinvolte sono formule predicative.

ESERCIZIO 7.50. Dimostrate la validità dei seguenti enunciati:

Corollario 7.62. *Se y è una variabile che non occorre in F allora $F \equiv F_x(y)$.*

Dimostrazione. Immediata dai lemmi 7.61 e 7.39, nonché dalla definizione 6.59. □

Il prossimo risultato esprime in maniera rigorosa un principio intuitivo: se $\forall x\, F$ è vera allora F è vera per ogni termine t, mentre se F è vera per un termine t allora $\exists x\, F$ è vera. Questo lemma ci sarà utile sia nella discussione dei tableaux predicativi

Corollario 7.64. *Se F è una formula e x una variabile allora $\forall x\, F \vDash F$ e $F \vDash \exists x\, F$.*

Dimostrazione. La sostituzione $\{x/x\}$ è ammissibile in F e $F\{x/x\}$ è F. $\square$

5. Logica con uguaglianza

Nella sezione 6.7 abbiamo introdotto i linguaggi in cui compare il simbolo $=$, e a

DEFINIZIONE 7.65. Un'interpretazione I per un linguaggio $\mathcal{L}$ che comprende il simbolo di relazione binario $=$ è detta *interpretazione normale*, se $=$ è interpretata come la relazione di identità su D^I, vale a dire se $=^I$ è $\{ (d, d) : d \in D^I \}$.

Un esempio di interpretazione normale è quello dell'esempio 7.15.

Nel descrivere un'interpretazione normale si indica, oltre al dominio dell'interpretazione, solo le interpretazioni dei simboli diversi da $=$.

DEFINIZIONE 7.69. Siano T e G un insieme di formule ed una formula dello stesso linguaggio $\mathcal{L}$. Diciamo che G è *conseguenza logica nella logica con uguaglianza* di T (e scriviamo $T \models_{=} G$) se per ogni interpretazione normale I per $\mathcal{L}$ ed ogni stato σ di I tale che $I, \sigma \models_{=} T$ si ha anche $I, \sigma \models_{=} G$. Se $T = \{F\}$ allora scriviamo $F \models_{=} G$ e diciamo che G è *conseguenza logica nella logica con uguaglianza* di F. Diciamo che F e G sono *logicamente equivalenti nella logica con uguaglianza* (e scriviamo $F \equiv_{=} G$) se $F \models_{=} G$ e $G \models_{=} F$.

Definizione 7.72. Sia F una formula.

- F è *valida nella logica con uguaglianza* se per ogni interpretazione normale I per $\mathcal{L}(F)$ si ha $I \models_{=} F$;
- F è *soddisfacibile nella logica con uguaglianza* se per qualche interpretazione normale I per $\mathcal{L}(F)$ e qualche stato σ di I si ha $I, \sigma \models_{=} F$;
- F è *insoddisfacibile nella logica con uguaglianza* se per ogni interpretazione normale I per $\mathcal{L}(F)$ e ogni stato σ di I si ha $I, \sigma \not\models_{=} F$.

DEFINIZIONE 8.1. Una formula F si dice *in forma prenessa* se è priva di quantificatori oppure è della forma $Q_1 x_1 \dots Q_k x_k\, G$, dove G è una formula priva di quantificatori e $Q_1, \dots, Q_k$ sono quantificatori. In questo caso $Q_1 x_1 \dots Q_k x_k$ (che non è una formula) si dice il *prefisso* di F, mentre G è chiamata la *matrice* di F.

ESEMPIO 8.2. Le formule $\forall x\, \exists y\, \exists z\, q(x, y, z)$, $\exists x\, \forall y (p(x) \to q(x, y, f(z)) \wedge p(y))$ e $\forall x\, \forall y\, r(x, f(y))$ sono in forma prenessa. La formula $\exists x\, \neg p(x) \wedge p(a)$ non è in forma

TEOREMA 8.3. *Ogni formula H può essere trasformata in una formula in forma prenessa K che è logicamente equivalente a H.*

L'espressione "può essere trasformata" ha lo stesso significato che aveva nell'enunciato del teorema 3.10: stiamo asserendo più della semplice esistenza di K, ed anche il teorema 8.3 verrà dimostrato attraverso la descrizione di un algoritmo che preso come input H, fornisce come output K.

Definizione 8.23. Data una formula F sia $\mathfrak{q}(F)$ il numero di quantificatori che compaiono in F. Definiamo per ricorsione sulla complessità di F il *p-grado* $\mathfrak{p}(F)$ di F ponendo:

- $\mathfrak{p}(F) = 0$ se F è atomica;
- $\mathfrak{p}(F) = \mathfrak{p}(G) + \mathfrak{q}(F)$ se F è $\neg G$;
- $\mathfrak{p}(F) = \mathfrak{p}(G) + \mathfrak{p}(H) + \mathfrak{q}(F)$ se F è $G * H$, con $*$ uno di $\wedge$, $\vee$ e $\rightarrow$;
- $\mathfrak{p}(F) = \mathfrak{p}(G)$ se F è QxG con Q uno di $\forall$ e $\exists$

Algoritmo 8.29. L'algoritmo per la trasformazione in forma prenessa prende in input una formula H e, iniziando con H, 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 per il lemma 8.27 esiste una sottoformula L di H' di una delle seguenti forme: $\neg Qx\, F$, $Qx\, F * G$, $G * Qx\, F$, dove Q è uno di $\forall$ e $\exists$, e $*$ è uno di $\wedge$, $\vee$ e $\rightarrow$. Fissata L di questo tipo si ottiene una nuova formula

Definizione 10.1. Diciamo che due interpretazioni I e J per un linguaggio $\mathcal{L}$ sono *elementarmente equivalenti rispetto a* $\mathcal{L}$ se per ogni enunciato F di $\mathcal{L}$ si ha che $I \models F$ se e solo se $J \models F$. In questo caso scriviamo $I \equiv_{\mathcal{L}} J$.

Esercizio 10.2. Dimostrare che se per ogni enunciato F di $\mathcal{L}$ si ha che $I \models F$ implica $J \models F$ allora $I \equiv_{\mathcal{L}} J$.
[Suggerimento: se F è un enunciato anche $\neg F$ è un enunciato.]

Definizione 10.3. Date due interpretazioni I e J per un linguaggio $\mathcal{L}$, un *omomorfismo* di I in J è una funzione $\varphi : D^I \to D^J$ tale che:

- per ogni simbolo di costante c di $\mathcal{L}$ si ha $\varphi(c^I) = c^J$;
- per ogni simbolo di funzione n-ario f di $\mathcal{L}$ ed ogni $d_1, \dots, d_n \in D^I$ si ha $\varphi(f^I(d_1, \dots, d_n)) = f^J(\varphi(d_1), \dots, \varphi(d_n))$;
- per ogni simbolo di relazione n-ario p di $\mathcal{L}$ ed ogni $d_1, \dots, d_n \in D^I$ si ha che se $(d_1, \dots, d_n) \in p^I$ allora $(\varphi(d_1), \dots, \varphi(d_n)) \in p^J$.

Teorema 10.13. *Siano I e J due interpretazioni per un linguaggio $\mathcal{L}$, σ uno stato di I e F una formula di $\mathcal{L}$. Se φ è un omomorfismo forte suriettivo di I in J allora $I, \sigma \vDash F$ se e solo se $J, \varphi \circ \sigma \vDash F$.*

Dimostrazione. La dimostrazione è per induzione sulla complessità di F: il caso di base delle formule atomiche ed i passi induttivi relativi ai connettivi sono

Corollario 10.14. *Se I e J sono interpretazioni per un linguaggio $\mathcal{L}$ e esiste un omomorfismo forte suriettivo di I in J, allora $I \equiv_{\mathcal{L}} J$. In particolare due interpretazioni isomorfe sono elementarmente equivalenti.*

Dimostrazione. La prima parte è immediata dal teorema 10.13, ricordando il corollario 7.13. Per la seconda parte basta osservare che un isomorfismo è un omomorfismo forte suriettivo. □

DEFINIZIONE 10.23. Se I è un'interpretazione per $\mathcal{L}$ e $\sim$ una relazione di congruenza su I, allora possiamo definire *l'interpretazione quoziente* $I/\!\sim$ di I rispetto a $\sim$. $I/\!\sim$ è l'interpretazione per $\mathcal{L}$ il cui dominio è $D^{I/\sim} = \left\{ [d] \,:\, d \in D^I \right\}$, dove $[d] = \left\{ d' \in D^I \,:\, d' \sim d \right\}$ è la classe d'equivalenza di d, e le interpretazioni dei simboli di costante, funzione e relazione sono definite come segue:

- per ogni simbolo di costante c, $c^{I/\sim} = [c^I]$;

Definizione 10.29. Se I è un interpretazione per $\mathcal{L}$ e $\sim$ è una relazione di congruenza su I, definiamo la funzione $\pi : D^I \to D^{I/\sim}$ ponendo $\pi(d) = [d]$ per ogni $d \in D^I$. π è detto *omomorfismo canonico*.

La terminologia della definizione precedente è giustificata dal seguente teorema.

Teorema 10.30. *Se I è un interpretazione per $\mathcal{L}$ e $\sim$ è una relazione di congruenza su I, allora l'omomorfismo canonico π è un omomorfismo forte suriettivo*

TEOREMA 10.30. *Se I è un interpretazione per $\mathcal{L}$ e $\sim$ è una relazione di congruenza su I, allora l'omomorfismo canonico π è un omomorfismo forte suriettivo di I su $I/\!\sim$.*

DIMOSTRAZIONE. La suriettività di π è immediata e basta verificare le condizioni della definizione 10.3:

- per ogni simbolo di costante c di $\mathcal{L}$ si ha $\pi(c^I) = [c^I] = c^{I/\sim}$;

COROLLARIO 10.31. *Siano I un interpretazione per $\mathcal{L}$, $\sim$ una relazione di congruenza su I con omomorfismo canonico π, e σ uno stato di I. Per ogni formula F di $\mathcal{L}$, $I, \sigma \vDash F$ se e solo se $I/\!\sim, \pi \circ \sigma \vDash F$. In particolare $I \equiv_{\mathcal{L}} I/\!\sim$.*

DIMOSTRAZIONE. Immediata dai teoremi 10.30 e 10.13. □

3. Applicazione alla logica con uguaglianza

TEOREMA 10.34. *Sia $\mathcal{L}$ un linguaggio con uguaglianza, T un insieme di formule di $\mathcal{L}$ e F una formula di $\mathcal{L}$. Allora*

$$T \models_{=} F \quad \text{se e solo se} \quad T, \mathsf{Eq}_{\mathcal{L}} \models F.$$

DIMOSTRAZIONE. Supponiamo che $T \models_{=} F$. Per mostrare che $T, \mathsf{Eq}_{\mathcal{L}} \models F$ fissiamo un'interpretazione (non necessariamente normale) I per $\mathcal{L}$ e uno stato σ di I tale che $I, \sigma \models T, \mathsf{Eq}_{\mathcal{L}}$. Dato che $I \models \mathsf{Eq}_{\mathcal{L}}$, per il lemma 10.32 $=^I$ è una relazione

Teorema 10.35. *Sia F una formula di un linguaggio con uguaglianza $\mathcal{L}$. F è soddisfacibile nella logica con uguaglianza se e solo se $\mathsf{Eq}_{\mathcal{L}}, F$ è soddisfacibile.*

Dimostrazione. F è soddisfacibile nella logica con uguaglianza se e solo se $\neg F$ non è valido nella logica con uguaglianza se e solo se $\not\models_{=} \neg F$ se e solo se (per il teorema 10.34) $\mathsf{Eq}_{\mathcal{L}} \not\models \neg F$ se e solo se $\mathsf{Eq}_{\mathcal{L}}, F$ è soddisfacibile. $\square$

Esercizio 10.36. Dimostrate il teorema 10.35 usando le interpretazioni, in

Convenzione 11.1. Per semplificare la nostra discussione dei tableaux predicativi nelle prime otto sezioni di questo capitolo restringiamo la nostra attenzione a linguaggi privi di simboli di funzione, in modo che gli unici termini chiusi siano i simboli di costante. Nella sezione 9 descriveremo brevemente come adattare il nostro metodo a linguaggi con simboli di funzione.

1. γ e δ-formule

Definizione 11.2. Se A è una formula atomica $\{A, \neg A\}$ 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.

La proprietà fondamentale delle coppie complementari è contenuta nel seguente lemma di immediata dimostrazione.

Lemma 11.3. *Se un insieme di formule contiene una coppia complementare*

Definizione 11.4. Una formula è *una γ-formula* se esiste F tale che la formula è di uno dei tipi che compaiono nella colonna sinistra della prima delle seguenti tabelle. Una formula è *una δ-formula* se esiste F tale che la formula è di uno dei tipi che compaiono nella colonna sinistra della seconda delle seguenti tabelle. In entrambi i casi un'*istanza* di una γ- o δ-formula è una formula del tipo che compare nella colonna più a destra, dove a è un simbolo di costante.

Algoritmo 11.16. Un tableau per un enunciato F è un albero in cui ogni nodo è etichettato con un insieme finito di enunciati. Il tableau è costruito per stadi $\mathcal{T}_0, \ldots, \mathcal{T}_i, \ldots$: per ogni i, $\mathcal{T}_{i+1}$ è un albero che estende $\mathcal{T}_i$ aggiungendo uno o due nodi con le rispettive etichette e lasciando invariate le etichette dei nodi già appartenenti a $\mathcal{T}_i$. L'unione degli alberi $\mathcal{T}_i$ è il tableau per F. Se n è un nodo di qualche $\mathcal{T}_i$ indichiamo con $E(n)$ l'etichetta di n (per quanto detto prima, la stessa per tutti i $\mathcal{T}_i$ cui appartiene n), che è un insieme di enunciati.

Definizione 11.17. Sia n un nodo del tableau che ha successori nel tableau: l'*enunciato su cui si agisce in n* è la G della descrizione dell'algoritmo.

Nota 11.18. I nodi su cui non possiamo agire nella costruzione del tableau sono quelli la cui etichetta contiene una coppia complementare di letterali oppure contiene solamente letterali. La costruzione del tableau termina se e soltanto se tutte le foglie dell'albero sono di uno di questi tipi.

Convenzione 11.19. Come nel caso proposizionale, per comodità di lettura aggiungeremo sotto i nodi del tableau su cui non possiamo agire uno dei simboli $\otimes$ e $\bigcirc$: se l'etichetta del nodo contiene una coppia complementare di letterali useremo $\otimes$, altrimenti $\bigcirc$.

Inoltre, per alleggerire la notazione, evitiamo di indicare le parentesi $\{$ e $\}$ intorno agli elementi di $E(n)$.

CONVENZIONE 11.21. Da questo punto in poi ogniqualvolta nelle etichette di un nodo di un tableau compare una doppia negazione F con ridotto G scriveremo direttamente G, utilizzando la regola della doppia negazione immediatamente e contraendo due nodi in uno.

ESEMPIO 11.22. Costruiamo un tableau per $\neg(\forall x \neg p(x) \to \neg\exists x\, p(x))$. In ogni nodo sottolineiamo l'enunciato su cui agiamo in quel nodo e utilizziamo la

Definizione 11.28. Un tableau è *chiuso* se non ha rami infiniti e tutte le sue foglie sono etichettate con insiemi di enunciati che contengono una coppia complementare di letterali. Un tableau è *aperto* se non è chiuso, cioè se contiene un ramo infinito oppure una foglia etichettata con un insieme di letterali che non contiene coppie complementari.

Un *ramo aperto* di un tableau è un ramo infinito oppure un ramo che collega la radice dell'albero con una foglia etichettata con un insieme di letterali che non

TEOREMA 11.29 (Teorema di correttezza). *Se un tableau per l'enunciato F è chiuso allora F è insoddisfacibile.*

DIMOSTRAZIONE. La dimostrazione ricalca da vicino quella del caso proposizionale (teorema 4.21). Fissiamo F e $\mathcal{T}$, tableau chiuso per F. Come nel caso proposizionale, dimostreremo il seguente fatto, che indichiamo con $(\star)$:

per ogni nodo n di $\mathcal{T}$ l'insieme di enunciati $E(n)$ è insoddisfacibile.

Corollario 11.30. *Se un tableau per l'enunciato $\neg F$ è chiuso allora F è valido.*

Dimostrazione. Immediata dai teoremi 11.29 e 7.48. $\square$

Esempio 11.31. Costruiamo un tableau per mostrare la validità dell'enunciato

Algoritmo 11.34. Un tableau sistematico per un enunciato F è un albero in cui ogni nodo è etichettato con un insieme finito di enunciati $E(n)$ e con una funzione C_n che associa ad ogni elemento di $\gamma(n) = \{\, G \in E(n) : G$ è γ-formula$\,\}$ un insieme finito di simboli di costante. Il tableau è costruito per stadi $\mathcal{T}_0, \ldots, \mathcal{T}_i, \ldots$: per ogni i, $\mathcal{T}_{i+1}$ è un albero che estende $\mathcal{T}_i$ aggiungendo uno o due nodi con le rispettive etichette e lasciando invariate le etichette dei nodi già appartenenti a $\mathcal{T}_i$. L'unione degli alberi $\mathcal{T}_i$ è il tableau per F.

Teorema 11.36 (Teorema di correttezza per i tableaux sistematici). *Se $\mathcal{T}$ è un tableau sistematico chiuso per l'enunciato F allora F è insoddisfacibile.*

6. La completezza dei tableaux predicativi

L'algoritmo 11.34 dei tableaux sistematici è stato introdotto per ottenere la completezza, cioè per dimostrare che se un enunciato è insoddisfacibile allora il metodo dei tableaux riesce a scoprirlo.

Teorema 11.37 (Teorema di completezza). *Se un tableau sistematico per l'enunciato F è aperto allora F è soddisfacibile.*

Schema della dimostrazione. Seguiamo l'approccio utilizzato nel caso dei tableaux proposizionali (teorema 4.22). Fissiamo un ramo aperto (finito o infinito) r di un tableau sistematico aperto per F. La dimostrazione si sviluppa in tre passi:

(a) definizione di insieme di Hintikka (definizione 11.38);

Definizione 11.38. Un insieme di enunciati $\mathcal{H}$ è un *insieme di Hintikka* se sono soddisfatte le seguenti condizioni:

(0) in $\mathcal{H}$ compare almeno un simbolo di costante;
(1) $\mathcal{H}$ non contiene coppie complementari di letterali;
(2) se $F \in \mathcal{H}$ è una doppia negazione con ridotto H allora $H \in \mathcal{H}$;
(3) se $F \in \mathcal{H}$ è una α-formula con ridotti H_1 e H_2 allora $H_1 \in \mathcal{H}$ e $H_2 \in \mathcal{H}$;
(4) se $F \in \mathcal{H}$ è una β-formula con ridotti H_1 e H_2 allora $H_1 \in \mathcal{H}$ oppure

Teorema 11.48. *Non esiste una procedura di decisione per l'insoddisfacibilità degli enunciati predicativi.*

Quindi una procedura di semidecisione come quella dei tableaux è quanto di meglio si possa ottenere e il caso predicativo è molto diverso dal caso proposizionale, in cui abbiamo introdotto due procedure di decisione per l'insoddisfacibilità (le

Algoritmo 11.50. Per stabilire se un insieme finito $T = \{F_1, \ldots, F_n\}$ di enunciati è soddisfacibile costruiamo un tableau con radice etichettata da T. Se il tableau è chiuso T è insoddisfacibile. Se il tableau è aperto e sistematico T è soddisfacibile (e un ramo aperto ci permette di definire un'interpretazione che soddisfa T).

Similmente la parte dell'esercizio 7.49 riguardante il lemma 2.40 ci permette di

Algoritmo 11.51. Per stabilire se $F_1, \ldots, F_n \models G$ costruiamo un tableau con radice etichettata da $\{F_1, \ldots, F_n, \neg G\}$. Se il tableau è chiuso abbiamo $F_1, \ldots, F_n \models G$. Se il tableau è aperto e sistematico allora $F_1, \ldots, F_n \not\models G$, e un ramo aperto del tableau permette di definire un'interpretazione che soddisfa $F_1, \ldots, F_n$ ma non G.

Esempio 11.52. Usiamo il metodo dei tableaux per stabilire che

Algoritmo 11.54. Per stabilire se $F_1, \ldots, F_n \models_= G$ costruiamo un tableau la cui radice è etichettata con $\{F_1, \ldots, F_n, \neg G\} \cup \mathsf{Eq}_{\mathcal{L}}$, dove $\mathcal{L} = \mathcal{L}(F_1, \ldots, F_n, G)$. Se il tableau è chiuso allora $F_1, \ldots, F_n \models_= G$. Se il tableau è aperto e sistematico allora $F_1, \ldots, F_n \not\models_= G$. In questo secondo caso un ramo aperto del tableau ci permette di definire un'interpretazione I che soddisfa $\mathsf{Eq}_{\mathcal{L}}, F_1, \ldots, F_n$ ma non G. Per ottenere un'interpretazione normale utilizziamo il lemma 10.32: l'interpretazione normale $I/{\equiv^I}$ soddisfa $F_1, \ldots, F_n$ ma non G.

Definizione 11.57. Se G è una γ-formula e t è un termine chiuso, l'istanza di G relativa ad t è $F\{x/t\}$ se G è $\forall x\, F$, e $\neg F\{x/t\}$ se G è $\neg\exists x\, F$.

La sostituzione effettuata nella definizione precedente è sempre ammissibile, perché t non contiene variabili (si veda la nota 6.53). Osserviamo inoltre che il lemma 11.6 (che asserisce che una γ-formula ha come conseguenza logica ogni sua istanza) vale anche per istanze relative a termini chiusi che non siano simboli di

Algoritmo 11.58. Un tableau per un enunciato F in un linguaggio con simboli di funzione è costruito dalla variante dell'algoritmo 11.16 ottenuta sostituendo la condizione (4) con

(4′) se G è una γ-formula sia G_1 l'istanza di G relativa ad un termine chiuso: aggiungiamo un nodo n' sotto n e poniamo $E(n') = E(n) \cup \{G_1\}$.

La correttezza di questo algoritmo viene dimostrata ripetendo la dimostrazione

Teorema 12.11 (Teorema di correttezza). *Siano T un insieme di formule predicative e F una formula predicativa. Se $T \rhd F$ allora $T \models F$.*

Dimostrazione. Come già per il teorema 5.17, la dimostrazione è per induzione sull'altezza dell'albero della deduzione naturale che testimonia $T \rhd F$. In altre parole, dimostriamo per induzione su n che se l'altezza di un albero di deduzione che mostra $T \rhd F$ è n, allora $T \models F$.

Teorema 12.13 (Teorema di completezza). *Siano T un insieme di formule predicative e F una formula predicativa. Se $T \vDash F$ allora $T \rhd F$.*

Come nel caso proposizionale (lemma 5.19) mostriamo che il sistema di deduzione naturale predicativo soddisfa la componibilità delle deduzioni. Nel caso proposizionale avevamo combinato le deduzioni in questo modo:

Teorema 12.58 (Teorema di correttezza con uguaglianza). *Siano T un insieme di formule predicative e F una formula predicativa in un linguaggio con uguaglianza. Se $T \rhd_= F$ allora $T \models_= F$.*

Dimostrazione. La dimostrazione è, come al solito, per induzione sull'altezza dell'albero della deduzione naturale che testimonia $T \rhd_= F$.

Nel caso di base si deve ora considerare anche la possibilità che la deduzione

Teorema 12.59 (Teorema di completezza con uguaglianza). *Siano T un insieme di formule predicative e F una formula predicativa in un linguaggio con uguaglianza. Se $T \models_{=} F$ allora $T \rhd_{=} F$.*

Dimostrazione. Per il teorema 10.34 $T \models_{=} F$ è equivalente a $T, \mathsf{Eq}_{\mathcal{L}} \models F$. Per il teorema di completezza 12.13 abbiamo $T, \mathsf{Eq}_{\mathcal{L}} \rhd F$. Utilizzando il lemma 12.56 è facile trasformare una deduzione naturale che testimonia $T, \mathsf{Eq}_{\mathcal{L}} \rhd F$ in una che