Università di Udine
DIPARTIMENTO DI SCIENZE MATEMATICHE, INFORMATICHE E FISICHE
Corso di Laurea in Informatica

dispense di

Logica Matematica

Alberto Marcone

e-mail: alberto.marcone@uniud.it
pagina web: http://users.dimi.uniud.it/~alberto.marcone/LMinf.html

Versione provvisoria del 1 ottobre 2021

Anno Accademico 2021–2022

Indice

Presentazione

Queste dispense raccolgono il materiale coperto nell'anno accademico 2021–2022 nel corso di Logica Matematica del Corso di Laurea triennale in Informatica dell'Università di Udine. Il corso si sviluppa per 6 crediti, corrispondenti a 48 ore di lezione frontale, ed è obbligatorio per gli studenti iscritti al secondo anno del corso di laurea.

Dall'anno accademico 2010–11 in poi gli argomenti trattati sono rimasti sostanzialmente invariati. Oltre alla correzione degli errori riscontrati, l'aggiunta di qualche esercizio e (si spera) il miglioramento in alcuni casi dell'esposizione, queste sono le modifiche principali, in ordine cronologico:

- a partire dall'anno accademico 2016–17 è cambiata la definizione 8.23 di p-grado (la nuova definizione è equivalente, ma si spera più facilmente comprensibile, di quella utilizzata in precedenza) e alcuni dettagli nella presentazione della deduzione naturale con uguaglianza (sezione 12.5);
- a partire dall'anno accademico 2017–18 sono cambiati la notazione nella dimostrazione del lemma 3.32, i dettagli nella dimostrazione di una direzione del lemma 8.26 e l'ordine delle sezioni nel capitolo 7:
- a partire dall'anno accademico 2020–21 il capitolo 8 è stato separato dal capitolo 7; inoltre sono stati introdotti i lemmi 5.19 e 12.14, relativi alla composizione delle deduzioni naturali.

Tutti i cambiamenti comportano variazioni nella numerazione di capitoli, sezioni, definizioni, teoremi, ecc, e nella numerazione delle pagine.

Le dispense sono state scritte in LaTeX con la classe `amsbook` dell'American Mathematical Society. Per i tableaux e le deduzioni naturali ho utilizzato rispettivamente i pacchetti `forest` di Sašo Živanović e `bussproofs` di Samuel R. Buss.

Ringrazio quanti hanno contribuito o contribuiranno, con le loro segnalazioni di errori, imprecisioni ed omissioni, a migliorare queste dispense. In particolare la prof. Giovanna D'Agostino ha utilizzato queste dispense nell'anno accademico 2011–12, quando è stata titolare del corso, ed ha fornito alcuni utili suggerimenti. Lo studente Stefano Rocco, che ha frequentato il corso nell'anno accademico 2015–16, ha segnalato diversi errori di battitura e fornito parecchi suggerimenti.

Alberto Marcone

Introduzione

Sia la logica che l'informatica si occupano di problemi legati alla formalizzazione, elaborazione e comunicazione della conoscenza. Alcune delle radici dell'informatica, che è una disciplina relativamente nuova, sono proprio nella logica. Da un lato la logica si è occupata, soprattutto a partire dal lavoro di Alan Turing[1] intorno al 1935, di studiare il concetto astratto di computer (prima ancora che i computer venissero effettivamente costruiti), dall'altro l'informatica ha l'esigenza di un linguaggio formale e sintatticamente preciso. La logica è la disciplina che da sempre si è occupata dello studio delle leggi del ragionamento. A partire dalla fine del XIX secolo, e con sempre maggior successo fino ad oggi, questo studio si è sviluppato attraverso la costruzione e lo studio con strumenti matematici dei linguaggi formali.

Per capire le problematiche che affronteremo iniziamo con il considerare tre semplici esempi di "deduzione", in cui da due affermazioni di partenza se ne deduce una terza. Nella colonna di destra abbiamo riportato le traduzioni in un opportuno linguaggio formale delle affermazioni della colonna di sinistra: rendere precise queste traduzioni è tra gli scopi di queste dispense. Alle questioni relative alla traduzione delle affermazioni da un linguaggio naturale (come l'italiano) ai linguaggi formali della logica è dedicato il capitolo 9 di queste dispense.

Ogni cane è un mammifero	$\forall x(c(x) \to m(x))$
Esistono cani non bianchi	$\exists x(c(x) \wedge \neg b(x))$
Non ogni mammifero è bianco	$\neg\forall x(m(x) \to b(x))$

Alcuni studenti sono biondi	$\exists x(s(x) \wedge b(x))$
Alcuni biondi hanno gli occhi azzurri	$\exists x(b(x) \wedge a(x))$
Qualche studente biondo ha gli occhi azzurri	$\exists x(s(x) \wedge b(x) \wedge a(x))$

Ogni computer è una mucca	$\forall x(c(x) \to m(x))$
Esistono computer non buffi	$\exists x(c(x) \wedge \neg b(x))$
Non ogni mucca è buffa	$\neg\forall x(m(x) \to b(x))$

Il primo e il terzo esempio, pur parlando di argomenti assai diversi, e facendo affermazioni più o meno sensate, hanno la stessa forma, e infatti le formule che compaiono nella colonna di destra risultano essere identiche. Queste due deduzioni sono (in un senso che renderemo preciso più avanti) argomenti corretti, e ciò dipende solo dalla loro forma, cioè dalle formule della colonna di destra.

Il secondo esempio, pur facendo asserzioni presumibilmente vere e deducendone una conclusione presumibilmente vera, non è un argomento corretto: possiamo concepire una situazione in cui tutti gli studenti biondi hanno gli occhi neri, pur esistendo studenti biondi e persone bionde con gli occhi azzurri. Invece, per quanto possa apparire assurda la situazione descritta, **se** siamo in una situazione in cui ogni

[1] Alan Turing (1912-1954) è stato un matematico, logico e crittografo britannico, considerato uno dei padri dell'informatica e uno dei più grandi matematici del XX secolo.

computer è una mucca ed esistono computer non buffi, **allora** necessariamente non tutte le mucche sono buffe.

Gli esempi precedenti hanno lo scopo di evidenziare come il nostro interesse si concentri sulla correttezza degli argomenti indipendentemente dal modo in cui le asserzioni contenute in essi vengono "interpretate", ovvero tenendo conto di tutte le possibili interpretazioni di queste asserzioni. Perciò dal punto di vista della logica il primo e il terzo esempio sono del tutto equivalenti.

In questo corso ci concentreremo quindi sulla manipolazione di formule di linguaggi formali, allo scopo di individuare precisamente cosa significa affermare che a partire da un insieme di formule possiamo dedurre logicamente una formula e come possiamo fare per riconoscere se ciò avviene o meno.

Il nostro oggetto di studio saranno quindi le formule e gli insiemi di formule. Questo comporta un'analisi del linguaggio formale che utilizziamo.

L'analisi di un linguaggio formale ha essenzialmente due aspetti:

- la SINTASSI: le "regole" per la costruzione di un linguaggio formale e delle sue espressioni (*termini* e *formule*);
- la SEMANTICA: il "significato" dei termini e delle formule di un linguaggio formale, analizzato attraverso le nozioni di *interpretazione* e di *soddisfazione*, che conducono a quelle di *equivalenza logica*, *conseguenza logica* (che rende matematicamente precisa la nozione intuitiva di "deduzione corretta" di cui abbiamo parlato in precedenza), *soddisfacibilità* e *validità*.

Per applicare questa analisi del linguaggio formale introdurremo

- i CALCOLI LOGICI: gli "algoritmi" che ci permettono di stabilire le proprietà semantiche di certe formule lavorando solo con la loro sintassi.

Inizieremo lo studio di sintassi e semantica in un caso piuttosto semplice, ma che permette di acquisire familiarità con le idee principali della materia, quello della *logica proposizionale*. Ad essa è dedicata la prima parte di queste dispense.

- Nel capitolo 1 definiremo la sintassi della logica proposizionale, cioè le regole che ci permettono di costruire le formule in questa logica.
- Il secondo capitolo è dedicato alla definizione della semantica della logica proposizionale, che conduce alle importanti nozioni di *conseguenza logica* e di *formula valida*. Nell'ultima sezione di questo capitolo vedremo i primi esempi di *traduzione* di asserzioni da un linguaggio naturale (quale l'italiano) ad un linguaggio formale (quale appunto la logica proposizionale).
- Nel capitolo 3 vengono presentate alcune procedure che permettono di *trasformare* (in modo puramente sintattico) una formula della logica proposizionale in una formula ad essa logicamente equivalente (una nozione semantica) che abbia una forma (sintattica) particolarmente semplice. Visto che il corso è indirizzato a studenti di informatica, particolare attenzione sarà dedicata agli aspetti algoritmici di queste trasformazioni.
- Nel quarto capitolo viene introdotto un primo calcolo logico: il metodo dei *tableaux* è un algoritmo relativamente efficiente per stabilire se una formula della logica proposizionale è valida.
- Nel capitolo 5 viene introdotto (sempre nel caso proposizionale) un altro calcolo logico: la *deduzione naturale*, che formalizza i metodi di dimostrazione utilizzati dagli esseri umani per fare deduzioni corrette.

La seconda parte delle dispense è invece dedicata alla *logica predicativa* (detta anche logica del prim'ordine). Rivedremo quanto visto nel caso più semplice della logica proposizionale, concentrandoci in particolare sugli aspetti più complessi che caratterizzano la logica predicativa.

- Il sesto capitolo è dedicato alla sintassi della logica predicativa. Diverse sono le differenze con il caso proposizionale: a partire da *variabili*, simboli di costante e di funzione si arriva dapprima alla nozione di *termine*, e solo a partire da essa, con l'uso dei simboli di relazione e dei *quantificatori*, si definiscono le formule.
- Nel settimo capitolo viene sviluppata la semantica della logica predicativa, assai più complessa di quella vista nel contesto proposizionale, basata sulle nozioni di *interpretazione* e di stato. Anch'essa conduce alle nozioni di conseguenza logica e di validità che sono da considerarsi le principali di questo corso.
- Nel capitolo 8 viene descritto un algoritmo che *trasforma*, in modo puramente sintattico, una formula predicativa in una formula logicamente equivalente che abbia una certa struttura sintattica.
- Come già indicato, nel capitolo 9 viene affrontata la traduzione di asserzioni da un linguaggio naturale ad un linguaggio formale (in questa sede la logica predicativa). Il lettore potrà apprezzare la maggior espressività della logica predicativa rispetto alla logica proposizionale.
- Il capitolo 10 affronta molto brevemente lo studio (di cui si occupa un importante settore della logica matematica moderna, la teoria dei modelli) delle relazioni che possono intercorrere tra diverse interpretazioni per lo stesso linguaggio.
- Nell'undicesimo capitolo viene presentato il metodo dei tableaux nel caso predicativo. Saranno messe in risalto le notevoli, ed inevitabili, differenze con il caso proposizionale, ed in particolare la possibilità che il metodo non termini.
- Infine, nel capitolo 12 la deduzione naturale viene estesa al caso predicativo.

Gli esercizi inseriti nel testo sono parte integrante delle dispense e si invita il lettore a tentare sistematicamente di svolgerli. Gli esercizi o le parti di esercizio contrassegnati con ($\star$) sono più impegnativi, e spesso richiedono una certa "maturità matematica".

La durata limitata del corso non permette un'introduzione esaustiva a tutti i numerosi aspetti della logica matematica di interesse per l'informatica: ulteriori approfondimenti sono lasciati al corso più avanzato offerto nel Corso di Laurea Magistrale.

Alcuni dei testi utilizzati nella preparazione delle dispense, e che possono essere utilmente consultati da chi desiderasse approfondire gli argomenti affrontati, sono i seguenti:

- M. Ben-Ari, *Mathematical Logic for Computer Science*, Springer;
- A. Asperti e A. Ciabattoni, *Logica a Informatica*, McGraw-Hill;
- C. Toffalori e P. Cintioli, *Logica Matematica*, McGraw-Hill;
- M. Huth and M. Ryan, *Logic in Computer Science*, Cambridge UP.

I primi tre testi si pongono approssimativamente al livello di queste dispense, mentre l'ultimo è più avanzato.

Segnalo inoltre il sito web `http://www.logicmatters.net/tyl/` (in inglese) che contiene una guida alla letteratura logica.

Parte 1

La logica proposizionale

CAPITOLO 1

Sintassi della logica proposizionale

Nella logica proposizionale partiamo da alcune *affermazioni* che non possono venir scomposte in affermazioni più semplici e di cui supponiamo di poter stabilire verità o falsità. Affermazioni di questo tipo sono ad esempio: "la terra è una sfera", "oggi piove", "Roma è la capitale della Francia", "Claudio e Maria sono amici".

A partire da questi *atomi* costruiamo delle affermazioni più complesse, quali "oggi piove oppure Claudio e Maria sono amici", "la terra non è una sfera", "se Roma è la capitale della Francia allora oggi piove", "se oggi piove allora Claudio e Maria sono amici e Roma non è la capitale della Francia". La costruzione di affermazioni più complesse avviene per mezzo di *connettivi* quali "e", "non", "se ... allora ...", "oppure". La verità o falsità di un'affermazione complessa dipende dalla verità o falsità degli atomi da cui essa è stata costruita. Ad esempio l'affermazione "oggi piove e Claudio e Maria sono amici" è vera se e solo se entrambe le affermazioni atomiche "oggi piove" e "Claudio e Maria sono amici" sono vere.

Il lettore si sarà accorto che un'analisi di questo tipo si applica solo a frasi del linguaggio naturale (nel nostro caso l'italiano) di un certo tipo. Non ha senso stabilire se frasi interrogative come "oggi piove?" sono vere o false, e quindi queste frasi non rientrano nel nostro oggetto di analisi. Lo stesso si può dire per frasi imperative come "vai a comprare il latte!".

Inoltre, nel contesto descritto qui sopra, affermazioni a prima vista complesse come "ogni amico di Claudio è amico anche di Bruna" non possono essere scomposte in affermazioni più semplici da cui esse siano state costruite usando i connettivi. Dal punto di vista della logica proposizionale questa affermazione è dunque un atomo. Vedremo però nella seconda parte del corso come la logica predicativa sia in grado di analizzare assai meglio affermazioni di questo tipo.

1. Formule proposizionali

Per approfondire la nostra analisi dobbiamo definire in modo matematicamente preciso il concetto di affermazione che finora abbiamo discusso con riferimento al linguaggio naturale. Nell'ambito della logica matematica un'affermazione è rappresentata da una formula, nel nostro caso della logica proposizionale.

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 $\rightarrow$, che sono chiamati rispettivamente *negazione*, *congiunzione*, *disgiunzione* e *implicazione*; il primo è un *connettivo unario*, mentre i restanti tre sono *connettivi binari*;
- le *parentesi* sono i due simboli (e).

NOTAZIONE 1.2. Nella prima parte (corrispondente ai primi cinque capitoli) di queste dispense le lettere $p, q, r, s, \dots$ rappresentano sempre lettere proposizionali del linguaggio che stiamo considerando.

Una formula proposizionale è una stringa di lettere proposizionali, connettivi proposizionali e parentesi, costruita secondo le regole specificate dalla prossima definizione.

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 $(F \rightarrow G)$ è una formula.

La formula $(\neg F)$ viene letta "non F", la formula $(F \wedge G)$ viene letta "F e G" oppure "F et G", la formula $(F \vee G)$ viene letta "F oppure G" o anche "F vel G" ("vel" è una delle due forme di "oppure" in latino, vedi nota 2.5), la formula $(F \rightarrow G)$ viene letta "se F allora G" oppure "F implica G".

ESEMPIO 1.4. La definizione 1.3 ci permette di riconoscere se una certa stringa di simboli è una formula. E' facile verificare che $\wedge p \rightarrow$ non è una formula (non è un elemento di $\mathcal{P}$ e non è stata costruita utilizzando nessuna delle altre quattro condizioni). D'altra parte $((\neg p) \vee q)$ è una formula: per verificarlo dettagliatamente si osserva che sono formule p, $(\neg p)$, q ed infine $((\neg p) \vee q)$ stessa (abbiamo utilizzato nell'ordine la prima, la seconda, la prima e la quarta condizione della definizione 1.3).

ESERCIZIO 1.5. Quali delle seguenti stringhe di simboli sono formule? $(\vee pq)$, $(\neg(p \rightarrow (q \wedge p)))$, $((p \wedge q) \rightarrow (r \vee))$, $((p \wedge (\neg q)) \vee (q \rightarrow r))$, $(p\neg r)$, $(\wedge p)$, $p \rightarrow q$.

NOTAZIONE 1.6. Nel seguito le lettere maiuscole F, G, H, ..., eventualmente fornite di pedice, indicheranno sempre formule (proposizionali nei primi cinque capitoli, predicative nei capitoli seguenti). Con le lettere maiuscole T e S (e anche con T', T'') indicheremo invece insiemi di formule. Con T, F indicheremo l'insieme $T \cup \{F\}$ e similmente T, F, G e T, S significano rispettivamente $T \cup \{F, G\}$ e $T \cup S$.

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 $(F \rightarrow G)$ per formule F e G; in questo caso F è detto l'*antecedente* di H, mentre G è il *conseguente* di H.

Il seguente lemma, la cui dimostrazione è immediata, ci permette di classificare le formule in cinque insiemi disgiunti.

LEMMA 1.8. *Ogni formula proposizionale è di uno e uno solo dei seguenti cinque tipi:*

- *una lettera proposizionale;*
- *una negazione;*
- *una congiunzione;*
- *una disgiunzione;*
- *un'implicazione.*

ESERCIZIO 1.9. Di che tipo sono la formula $((\neg p) \vee q)$ e le stringhe dell'esercizio 1.5 che sono formule?

La definizione 1.3 è ricorsiva, e questo ci permette di dimostrare proprietà delle formule ragionando per induzione. Una dimostrazione di questo tipo è giustificata dal seguente teorema e viene detta *per induzione sulla complessità delle formule.*

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 $\mathcal{A}((F \to G))$.*

Allora $\mathcal{A}(F)$ vale per ogni formula F.

ESERCIZIO 1.11. Dimostrare per induzione sulla complessità delle formule che ogni formula contiene un uguale numero di (e di).

Similmente è possibile dare definizioni procedendo *per ricorsione sulla complessità delle formule.* Anziché dare una descrizione generale di questo procedimento, forniamo un esempio del suo utilizzo.

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.

ESERCIZIO 1.14. ($\star$) Dimostrare per induzione sulla complessità delle formule che il grado di F è uguale al numero di connettivi che compaiono in F.

2. Usare meno parentesi: precedenze tra connettivi

Il lettore che ha svolto con cura l'esercizio 1.5 si sarà accorto che la nostra definizione di formula comporta l'uso di molte parentesi, più di quelle a cui siamo solitamente abituati. Si considerino ad esempio le stringhe $(\neg((\neg p) \to (q \wedge (\neg r))))$, $(\neg((\neg p) \to (q \wedge (\neg r)))$ e $p \wedge q$: secondo la nostra definizione la prima è una formula, la seconda no (perché una parentesi aperta non si "chiude") e la terza neppure, a causa della mancanza di parentesi. La differenza tra la prima e la seconda stringa è però difficile da cogliere, almeno all'occhio umano, proprio a causa dell'alto numero di parentesi. La terza stringa è invece di facile lettura, e l'abbiamo probabilmente già incontrata come formalizzazione di "p e q". Secondo le nostre regole la formula che si legge "p e q" è $(p \wedge q)$, appesantita dalle parentesi esterne.

Per semplificare la lettura introdurremo ora alcune regole di precedenza tra i connettivi. Per capire ciò che intendiamo fare richiamiamo le regole sull'uso delle parentesi in aritmetica, con cui siamo talmente familiari da non accorgerci neppure più del loro utilizzo. In quel contesto la scrittura $a \cdot b + c$ va intesa come $(a \cdot b) + c$ e non come $a \cdot (b + c)$: questo perché alla moltiplicazione è stata assegnata (per convenzione arbitraria, ma universalmente accettata) precedenza sull'addizione. Similmente $a \cdot b^2$ significa $a \cdot (b^2)$ e non $(a \cdot b)^2$: l'operazione di elevamento a potenza ha precedenza sulla moltiplicazione. Naturalmente ci sono dei casi in cui l'uso delle parentesi è necessario, proprio per sovvertire l'ordine di precedenza delle operazioni aritmetiche fissato: è il caso di $((a + b) \cdot c)^2$, in cui vanno eseguite nell'ordine addizione, prodotto e elevamento a potenza.

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

(1) si omettono le parentesi più esterne;
(2) $\neg$ ha la precedenza su $\wedge$, $\vee$ e $\to$, così che $\neg F \vee G$ abbrevia $((\neg F) \vee G)$;
(3) $\wedge$ e $\vee$ hanno la precedenza su $\to$, così che $F \wedge G \to H$ e $F \to G \vee H$ abbreviano rispettivamente $((F \wedge G) \to H)$ e $(F \to (G \vee H))$;
(4) ulteriori parentesi eventualmente omesse in formule costruite con più di una $\wedge$ o $\vee$ si appoggiano a sinistra, così che $F \wedge G \wedge H$ e $F \vee G \vee H$ abbreviano rispettivamente $((F \wedge G) \wedge H)$ e $((F \vee G) \vee H)$.

Esempio 1.16. Consideriamo nuovamente la formula $(\neg((\neg p) \to (q \wedge (\neg r))))$: utilizzando la convenzione 1.15 essa viene abbreviata da $\neg(\neg p \to q \wedge \neg r)$, che è assai più leggibile.

Esercizio 1.17. Stabilite qual è la differenza tra le formule (scritte adottando la convenzione 1.15) $\neg p \to \neg q \wedge r$ e $\neg(p \to \neg q) \wedge r$ analizzando i passaggi attraverso cui sono state costruite a partire dalle lettere proposizionali.

Nota 1.18. Osserviamo che $p \wedge q \vee r$ non è un'abbreviazione corretta per una formula. Infatti la convenzione 1.15 non assegna una precedenza tra $\wedge$ e $\vee$. Sarà quindi necessario scrivere $(p \wedge q) \vee r$ oppure $p \wedge (q \vee r)$.

Similmente non è stato specificato un modo per interpretare $p \to q \to r$.

Torneremo su queste espressioni nell'esempio 2.7.

D'ora in poi utilizzeremo sistematicamente, e senza più richiamarla esplicitamente, la convenzione 1.15 per scrivere le formule.

3. Sottoformule

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, mentre $p \to q$ non lo è. Infatti, inserendo alcune delle parentesi omesse in base alla convenzione 1.15, F è $p \to (q \vee \neg r)$. In effetti $q \vee \neg r$ è una delle formule utilizzate nella costruzione di F, mentre $p \to q$ non lo è.

Esercizio 1.21. Elencate tutte le sottoformule della F dell'esempio precedente (sono sei, di cui cinque proprie).

La definizione di sottoformula può anche venir data tramite una ricorsione sulla complessità delle formule.

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.

Esercizio 1.23. Dimostrare per induzione sulla complessità delle formule (teorema 1.10) che le definizioni 1.19 e 1.22 coincidono.

CAPITOLO 2

Semantica della logica proposizionale

Nel capitolo 1 abbiamo introdotto la nozione di formula, ora il nostro obiettivo è stabilire quando una formula risulta essere vera. Anche in questo caso è utile ricorrere all'analogia con l'aritmetica: l'uguaglianza $a + b^2 = c \cdot a$ non è né vera né falsa, almeno sino a quando non assegnamo dei valori a a, b e c. Se $a = 8$, $b = 4$ e $c = 3$ essa risulterà essere vera, mentre per $a = 3$, $b = 4$ e $c = 7$ essa risulterà falsa. In modo analogo la verità o falsità delle formule proposizionali dipende dalla verità o falsità delle lettere proposizionali che sono state utilizzate nella loro costruzione.

1. Valutazioni e interpretazioni

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.

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 $\bar{v}(F) = \begin{cases} \mathbb{V}, & \text{se } \bar{v}(G) = \mathbb{F}; \\ \mathbb{F}, & \text{se } \bar{v}(G) = \mathbb{V}. \end{cases}$
- se F è $G \wedge H$ allora $\bar{v}(F) = \begin{cases} \mathbb{V}, & \text{se } \bar{v}(G) = \mathbb{V} \text{ e } \bar{v}(H) = \mathbb{V}; \\ \mathbb{F}, & \text{altrimenti.} \end{cases}$
- se F è $G \vee H$ allora $\bar{v}(F) = \begin{cases} \mathbb{V}, & \text{se } \bar{v}(G) = \mathbb{V} \text{ oppure } \bar{v}(H) = \mathbb{V}; \\ \mathbb{F}, & \text{altrimenti.} \end{cases}$
- se F è $G \to H$ allora $\bar{v}(F) = \begin{cases} \mathbb{V}, & \text{se } \bar{v}(G) = \mathbb{F} \text{ oppure } \bar{v}(H) = \mathbb{V}; \\ \mathbb{F}, & \text{altrimenti.} \end{cases}$

La definizione 2.3 va vista come un'usuale definizione matematica in cui viene dato per acquisito il senso di "se", "e" e "oppure".

Nota 2.4. La seguente tabella riassume alcune parti della definizione precedente. In ogni riga sono indicati i valori di verità di $G \wedge H$, $G \vee H$ e $G \to H$ in corrispondenza dei valori di verità di G e H indicati nelle prime due colonne.

G	H	$G \wedge H$	$G \vee H$	$G \to H$
$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$
$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$

Nota 2.5. Notiamo che $G \vee H$ è vera se e solo se G è vera oppure H è vera. Qui intendiamo "oppure" nel senso inclusivo del "vel" latino (dalla cui iniziale proviene il simbolo $\vee$) per cui è ammesso che entrambe le asserzioni siano vere. In latino esiste anche l'oppure esclusivo "aut" (di cui in italiano resta traccia nell'espressione "aut aut"), per affermare che una e una sola delle due alternative è vera. È possibile introdurre un ulteriore connettivo logico per designare l'aut, ma è più semplice osservare che "G aut H" può venir scritto[1] come $(G \vee H) \wedge \neg(G \wedge H)$.

Nota 2.6. La nostra intenzione è che $G \rightarrow H$ sia vera se e solo se è vera l'affermazione "se G allora H". Intendiamo "se ... allora ..." nel senso dell'implicazione materiale: se G è vero deve esserlo anche H, se G è falso allora l'implicazione è vera, indipendentemente dalla verità di H. Perciò $G \rightarrow H$ è vero se e solo se G è falso oppure H è vero. Questo significa che perché $G \rightarrow H$ sia falso è necessario che G sia vero e H falso.

Esempio 2.7. Consideriamo la valutazione v definita da $v(p) = \mathbb{F}$, $v(q) = \mathbb{V}$, $v(r) = \mathbb{V}$ e supponiamo di voler calcolare $\bar{v}(F)$ dove F è $\neg(\neg p \rightarrow q \wedge \neg r)$. Applicando le varie parti della definizione di interpretazione troveremo: $\bar{v}(\neg r) = \mathbb{F}$, $\bar{v}(q \wedge \neg r) = \mathbb{F}$, $\bar{v}(\neg p) = \mathbb{V}$, $\bar{v}(\neg p \rightarrow q \wedge \neg r) = \mathbb{F}$, ed infine $\bar{v}(F) = \mathbb{V}$. La formula F risulta dunque essere vera nell'interpretazione $\bar{v}$.

Notiamo anche che $\bar{v}(p \rightarrow (q \rightarrow p)) = \mathbb{V}$, $\bar{v}((p \rightarrow q) \rightarrow p) = \mathbb{F}$, $\bar{v}((p \wedge q) \vee r) = \mathbb{V}$ e $\bar{v}(p \wedge (q \vee r)) = \mathbb{F}$. Questo mostra che le scritture $p \rightarrow q \rightarrow p$ e $p \wedge q \vee r$ sono ambigue (infatti secondo la convenzione 1.15 queste stringhe non abbreviano delle formule) ed è necessario usare le parentesi.

Il prossimo lemma è la formulazione matematicamente precisa di un'osservazione intuitiva che abbiamo utilizzato implicitamente nell'esempio 2.7: il valore di verità di una formula secondo l'interpretazione $\bar{v}$ dipende solo dai valori assunti da v sulle lettere proposizionali che compaiono nella formula.

Lemma 2.8. *Sia F una formula e sia $\mathcal{P}' \subseteq \mathcal{P}$ 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 $p \in \mathcal{P}'$. Allora $\bar{v}_1(F) = \bar{v}_2(F)$.*

Dimostrazione. Per induzione sulla complessità di F. □

Notazione 2.9. Per semplificare la notazione d'ora in poi scriveremo v anche per l'interpretazione, confondendola così con la valutazione da cui siamo partiti per definirla.

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 $(p \rightarrow \neg q) \vee \neg(r \wedge q)$ e $\neg(\neg p \rightarrow q) \wedge r$.

2. Equivalenza e conseguenza logica

Le formule $p \vee q$ e $q \vee p$ hanno intuitivamente lo stesso significato. Per rendere precisa questa osservazione notiamo che per qualunque interpretazione v si ha $v(p \vee q) = v(q \vee p)$.

---

[1] L'espressione "può venir scritto" viene resa precisa come segue. Sia $\oplus$ un nuovo connettivo binario per l'aut, con la regola $\bar{v}(G \oplus H) = \mathbb{V}$ se e solo se $\bar{v}(G) = \mathbb{V}$ e $\bar{v}(H) = \mathbb{F}$ oppure $\bar{v}(G) = \mathbb{F}$ e $\bar{v}(H) = \mathbb{V}$. Allora $G \oplus H \equiv (G \vee H) \wedge \neg(G \wedge H)$, dove il simbolo $\equiv$ di equivalenza logica verrà introdotto nella definizione 2.12.

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 $F \equiv H$).

ESEMPIO 2.14. Siano F e G le formule $p \to \neg p$ e $\neg p$. Sia v una interpretazione qualsiasi. Ci sono due possibilità: se $v(p) = \mathbb{V}$ allora si verifica che $v(F) = v(G) = \mathbb{F}$; se $v(p) = \mathbb{F}$ invece $v(F) = v(G) = \mathbb{V}$. In ogni caso $v(F) = v(G)$, e perciò $F \equiv G$.

Siano ora F e G le formule $p \wedge q$ e p. F e G non sono logicamente equivalenti: basta considerare un'interpretazione con $v(p) = \mathbb{V}$ e $v(q) = \mathbb{F}$. Possiamo però notare che se un'interpretazione soddisfa F allora soddisfa anche G: la verità di F ha come conseguenza la verità di G. Esiste dunque un legame tra le due formule.

DEFINIZIONE 2.15. Siano F e G due formule. Diciamo che G è *conseguenza logica* di F (in simboli $F \vDash G$) 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 $F \vDash F$, e da $F \vDash G$ e $G \vDash H$ segue $F \vDash H$.

ESEMPIO 2.17. Siano F e G le formule $(p \to q) \wedge p$ e $p \vee q$ rispettivamente.

$F \vDash G$: infatti se v è un'interpretazione che soddisfa F allora in particolare deve essere $v(p) = \mathbb{V}$, da cui segue che v soddisfa G.

$G \nvDash F$: si consideri ad esempio v tale che $v(p) = \mathbb{F}$ e $v(q) = \mathbb{V}$; v soddisfa G ma non soddisfa F.

Notiamo che abbiamo dimostrato anche che $F \not\equiv G$.

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

NOTA 2.18. A differenza di $\wedge$ e $\vee$, i simboli $\equiv$ e $\vDash$ **non** sono simboli logici. Perciò $F \equiv G$ e $F \vDash G$ **non** sono formule nel senso della definizione 1.3. Le espressioni $F \equiv G$ e $F \vDash G$ sono abbreviazioni per certe affermazioni che noi facciamo parlando delle formule della logica, esprimendoci nel *metalinguaggio* (il linguaggio che utilizziamo per parlare dei linguaggi logici).

Possiamo affermare che gran parte delle presenti dispense riguarda lo studio delle relazioni $\vDash$ e $\equiv$, le cui definizioni sono quindi le più importanti del corso. Esse sono strettamente collegate tra loro, come evidenziato dal prossimo lemma.

LEMMA 2.19. *Due formule F e G sono logicamente equivalenti se e solo se $F \vDash G$ e $G \vDash F$.*

DIMOSTRAZIONE. Se $F \vDash G$ e $G \vDash F$ consideriamo un'interpretazione qualsiasi v. Se $v(F) = \mathbb{V}$ allora $F \vDash G$ implica $v(G) = \mathbb{V}$. Se $v(F) = \mathbb{F}$ allora non può essere $v(G) = \mathbb{V}$ (altrimenti, dato che $G \vDash F$, si avrebbe $v(F) = \mathbb{V}$) e quindi vale $v(G) = \mathbb{F}$.

La direzione inversa è lasciata come facile esercizio per il lettore. □

Il prossimo lemma raccoglie alcune equivalenze e conseguenze logiche elementari, che seguono immediatamente dalle definizioni.

LEMMA 2.20. *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* $\neg F \models F \rightarrow G$.

Dimostrazione. Immediata dalle definizioni 2.3, 2.15 e 2.12. Si invita il lettore a svolgere in dettaglio almeno alcune di queste dimostrazioni. □

Ciascuna delle affermazioni del lemma 2.20 è uno *schema* di equivalenze e conseguenze logiche e non una singola equivalenza o conseguenza logica. Ad esempio (2) significa che, tra le altre, valgono le equivalenze logiche $p \wedge q \equiv q \wedge p$, $p \wedge r \equiv r \wedge p$ e $(p \rightarrow q \vee \neg r) \wedge (\neg q \vee s) \equiv (\neg q \vee s) \wedge (p \rightarrow q \vee \neg r)$. Quasi tutte le nostre affermazioni sulle conseguenze ed equivalenze logiche saranno schemi di questo tipo.

Nota 2.21. L'implicazione non compare in nessuna delle equivalenze logiche del lemma 2.20, ed in effetti questo connettivo non ha alcune delle proprietà degli altri connettivi binari. Ad esempio, salvo casi particolari, $F \rightarrow G \not\equiv G \rightarrow F$ (si veda però il lemma 2.24.5 qui sotto) e $F \rightarrow F \not\equiv F$ (ma notate che un caso particolare del lemma 2.20.8 è $F \models F \rightarrow F$).

Il seguente lemma è molto utile per dimostrare l'equivalenza logica di formule complesse. Esso asserisce che sostituendo all'interno di una formula una sottoformula con una formula ad essa equivalente, si ottiene una formula equivalente a quella di partenza.

Lemma 2.22. *Se F è una sottoformula di una formula H e $F \equiv G$ allora $H \equiv H'$, dove H' è la formula ottenuta da H rimpiazzando la sottoformula F con G.*

Dimostrazione. Fissiamo le formule logicamente equivalenti F e G. La dimostrazione è per induzione sulla complessità della formula H di cui F è sottoformula.

Il caso più semplice possibile è quello in cui F coincide con H: in questo caso H' è G e il risultato segue da $F \equiv G$.

Se H è una negazione $\neg H_0$ e F è una sottoformula propria di H allora F è una sottoformula di H_0. Per ipotesi induttiva $H_0 \equiv H_0'$ e quindi $\neg H_0 \equiv \neg H_0'$, cioè $H \equiv H'$.

Se H è del tipo $H_0 * H_1$ con $*$ uno di $\wedge$, $\vee$ e $\rightarrow$ e F è una sottoformula propria di H allora F è una sottoformula di H_0 oppure F è una sottoformula di H_1. Nel primo caso H' è $H_0' * H_1$ e per ipotesi induttiva $H_0 \equiv H_0'$ da cui è facile ricavare che, qualunque sia $*$, si ha $H_0 * H_1 \equiv H_0' * H_1$, cioè $H \equiv H'$. Il secondo caso è analogo. □

Esempio 2.23. Utilizzando il lemma 2.22, la transitività di $\equiv$, e alcune delle equivalenze logiche del lemma 2.20 è facile dimostrare che

$$F \wedge G \rightarrow (\neg F \vee H) \wedge (\neg\neg H \vee \neg F) \equiv \neg\neg G \wedge F \rightarrow H \vee \neg F.$$

Il seguente lemma contiene alcuni importanti ed utili schemi di equivalenze logiche. Le prime due equivalenze sono note come leggi di De Morgan, la terza e la quarta permettono di eliminare l'implicazione in favore di disgiunzione e congiunzione, la quinta asserisce l'equivalenza tra un'implicazione e la sua contrapositiva, la sesta e la settima esprimono l'associatività di congiunzione e disgiunzione (invece in generale non si ha $(F \rightarrow G) \rightarrow H \equiv F \rightarrow (G \rightarrow H)$, come mostrato nella seconda parte dell'esempio 2.7), mentre le ultime quattro esprimono la distributività della congiunzione rispetto alla disgiunzione e viceversa.

Lemma 2.24. *Se F, G e H sono formule allora:*
(1) $\neg(F \wedge G) \equiv \neg F \vee \neg G$;
(2) $\neg(F \vee G) \equiv \neg F \wedge \neg G$;
(3) $F \to G \equiv \neg F \vee G$;
(4) $\neg(F \to G) \equiv F \wedge \neg G$;
(5) $F \to G \equiv \neg G \to \neg F$;
(6) $(F \wedge G) \wedge H \equiv F \wedge (G \wedge H)$;
(7) $(F \vee G) \vee H \equiv F \vee (G \vee H)$;
(8) $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$;
(9) $(G \vee H) \wedge F \equiv (G \wedge F) \vee (H \wedge F)$;
(10) $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$;
(11) $(G \wedge H) \vee F \equiv (G \vee F) \wedge (H \vee F)$.

Dimostrazione. Spesso utilizzeremo il lemma 2.19 (senza citarlo ogni volta esplicitamente), cioè dimostreremo un'equivalenza logica dimostrando due conseguenze logiche. In tutta la dimostrazione v è un'interpretazione arbitraria.

(1) Se v soddisfa $\neg(F \wedge G)$ consideriamo due possibilità. Se $v(F) = \mathbb{F}$, allora $v(\neg F) = \mathbb{V}$ e quindi $v(\neg F \vee \neg G) = \mathbb{V}$. Se invece $v(F) = \mathbb{V}$, dall'ipotesi che $v(\neg(F \wedge G)) = \mathbb{V}$, si ottiene che $v(G) = \mathbb{F}$, quindi che $v(\neg G) = \mathbb{V}$ e dunque che $v(\neg F \vee \neg G) = \mathbb{V}$. In ogni caso $v(\neg F \vee \neg G) = \mathbb{V}$. Quindi $\neg(F \wedge G) \models \neg F \vee \neg G$.

    Se v soddisfa $\neg F \vee \neg G$ allora $v(F) = \mathbb{F}$ oppure $v(G) = \mathbb{F}$. In ogni caso $v(F \wedge G) = \mathbb{F}$, cioè $v(\neg(F \wedge G)) = \mathbb{V}$. Perciò $\neg F \vee \neg G \models \neg(F \wedge G)$.
(2) Tanto $v(\neg(F \vee G)) = \mathbb{V}$ che $v(\neg F \wedge \neg G) = \mathbb{V}$ sono equivalenti a $v(F) = \mathbb{F}$ e $v(G) = \mathbb{F}$.
(3) Se v soddisfa $F \to G$ consideriamo due possibilità. Se $v(F) = \mathbb{V}$ allora $v(G) = \mathbb{V}$ e quindi $v(\neg F \vee G) = \mathbb{V}$. Se invece $v(F) = \mathbb{F}$, allora $v(\neg F) = \mathbb{V}$ e quindi $v(\neg F \vee G) = \mathbb{V}$. In ogni caso v soddisfa $\neg F \vee G$, così che $F \to G \models \neg F \vee G$.

    Se v soddisfa $\neg F \vee G$ è immediato constatare che sia nel caso in cui $v(\neg F) = \mathbb{V}$, sia in quello in cui $v(G) = \mathbb{V}$ si ha che $v(F \to G) = \mathbb{V}$. Quindi $\neg F \vee G \models F \to G$.
(4) Questa equivalenza si può ottenere, grazie al lemma 2.22 e alla transitività di $\equiv$, utilizzando nell'ordine la (3), la (2) e il lemma 2.20.1:
$$\neg(F \to G) \equiv \neg(\neg F \vee G) \equiv \neg\neg F \wedge \neg G \equiv F \wedge \neg G.$$
(5) Questa equivalenza si può ottenere, grazie al lemma 2.22 e alla transitività di $\equiv$, utilizzando nell'ordine la (3), le parti (1) e (3) del lemma 2.20, ed infine nuovamente la (3):
$$F \to G \equiv \neg F \vee G \equiv \neg F \vee \neg\neg G \equiv \neg\neg G \vee \neg F \equiv \neg G \to \neg F.$$
(6) e (7) sono immediate, usando l'associatività di "e" e "oppure" del linguaggio naturale.
(8) Se $v(F \wedge (G \vee H)) = \mathbb{V}$ allora $v(F) = \mathbb{V}$ e $v(G \vee H) = \mathbb{V}$. Se $v(G) = \mathbb{V}$ allora $v(F \wedge G) = \mathbb{V}$, mentre se $v(H) = \mathbb{V}$ allora $v(F \wedge H) = \mathbb{V}$. In ogni caso $v((F \wedge G) \vee (F \wedge H)) = \mathbb{V}$. Perciò $F \wedge (G \vee H) \models (F \wedge G) \vee (F \wedge H)$.

    Sia $v((F \wedge G) \vee (F \wedge H)) = \mathbb{V}$. Se $v(F \wedge G) = \mathbb{V}$ allora $v(F) = \mathbb{V}$ e $v(G) = \mathbb{V}$, mentre se $v(F \wedge H) = \mathbb{V}$ allora $v(F) = \mathbb{V}$ e $v(H) = \mathbb{V}$. In ogni caso $v(F) = \mathbb{V}$ e $v(G \vee H) = \mathbb{V}$. Quindi $v(F \wedge (G \vee H)) = \mathbb{V}$ e $(F \wedge G) \vee (F \wedge H) \models F \wedge (G \vee H)$.
(9) segue da (8) utilizzando il lemma 2.20.2 e il lemma 2.22.

(10) Se $v(F \vee (G \wedge H)) = \mathbb{V}$ allora $v(F) = \mathbb{V}$ oppure $v(G \wedge H) = \mathbb{V}$. Se vale $v(F) = \mathbb{V}$ allora $v(F \vee G) = \mathbb{V}$ e $v(F \vee H) = \mathbb{V}$, così che $v((F \vee G) \wedge (F \vee H)) = \mathbb{V}$. Se invece $v(G \wedge H) = \mathbb{V}$ allora $v(G) = \mathbb{V}$ e $v(H) = \mathbb{V}$, e quindi $v((F \vee G) \wedge (F \vee H)) = \mathbb{V}$ anche in questo caso. Abbiamo mostrato che $F \vee (G \wedge H) \vDash (F \vee G) \wedge (F \vee H)$.

Se $v((F \vee G) \wedge (F \vee H)) = \mathbb{V}$ allora $v(F \vee G) = \mathbb{V}$ e $v(F \vee H) = \mathbb{V}$. Se $v(F) = \mathbb{V}$ allora ovviamente $v(F \vee (G \wedge H)) = \mathbb{V}$. Se invece $v(F) = \mathbb{F}$ allora abbiamo tanto $v(G) = \mathbb{V}$ che $v(H) = \mathbb{V}$: quindi $v(G \wedge H) = \mathbb{V}$ e di conseguenza $v(F \vee (G \wedge H)) = \mathbb{V}$. Perciò $(F \vee G) \wedge (F \vee H) \vDash F \vee (G \wedge H)$.

(11) segue da (10) utilizzando il lemma 2.20.3 e il lemma 2.22. □

Esercizio 2.25. Dimostrare (4) e (5) del lemma 2.24 usando le interpretazioni.

Esercizio 2.26. Utilizzando i lemmi 2.20 e 2.24 (nonchè il lemma 2.22) dimostrare le seguenti equivalenze logiche:

$$F \to (G \to H) \equiv F \wedge G \to H;$$
$$(F \to G) \to H \equiv (F \vee H) \wedge (G \to H);$$
$$F \to G \wedge H \equiv (F \to G) \wedge (F \to H);$$
$$F \vee G \to H \equiv (F \to H) \wedge (G \to H);$$
$$(F \to G) \wedge (H \vee \neg F) \equiv \neg(H \to F) \vee (F \to (G \wedge (F \to H)));$$
$$\neg((F \vee G) \wedge (\neg G \vee H)) \equiv F \vee G \to G \wedge \neg H.$$

In alcune circostanze è utile estendere la nozione di conseguenza logica ad insiemi di formule.

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 $T = \{p, q\}$, $G = p \wedge q$.

Notazione 2.29. Se $T = \{F_1, \dots, F_n\}$ è un insieme finito di formule spesso scriviamo $F_1, \dots, F_n \vDash G$ al posto di $\{F_1, \dots, F_n\} \vDash G$.

Lemma 2.30. *Se $F_1, \dots, F_n$ e G sono formule, allora $F_1, \dots, F_n \vDash G$ se e solo se $F_1 \wedge \dots \wedge F_n \vDash G$.*

Dimostrazione. Immediata dalle definizioni. □

Esercizio 2.31. Siano T e S insiemi di formule, F, G e H formule. Dimostrare le seguenti proprietà:

(a) $F, G \vDash F \wedge G$;
(b) $F, F \to G \vDash G$;
(c) $F \to G, \neg G \vDash \neg F$;
(d) Se $T \vDash F$ e $S \supseteq T$ allora $S \vDash F$;
(e) Se $T \vDash F$ e $S, F \vDash G$ allora $T \cup S \vDash G$;
(f) $T, F \vDash G$ se e solo se $T \vDash F \to G$;
(g) $T, \neg F \vDash G$ se e solo se $T \vDash F \vee G$;
(h) Se $T \vDash F$ e $T \vDash F \to G$ allora $T \vDash G$;
(i) Se $T \vDash F$ e $T, G \vDash H$ allora $T, F \to G \vDash H$;
(j) $T \vDash F$ e $T \vDash G$ se e solo se $T \vDash F \wedge G$;
(k) Se $T \vDash F$ oppure $T \vDash G$ allora $T \vDash F \vee G$;
(l) Se $T, F \vDash H$ e $T, G \vDash H$ allora $T, F \vee G \vDash H$.

Esercizio 2.32. Trovate F, G e H tali che $H \vDash F \vee G$ ma $H \nvDash F$ e $H \nvDash G$, mostrando così che l'implicazione inversa a quella dell'esercizio 2.31.k non vale. Similmente, mostrate che l'implicazione inversa a quella dell'esercizio 2.31.l non vale trovando F', G' e H' tali che $F' \vee G' \vDash H'$ e $F' \nvDash H'$.

3. Validità e soddisfacibilità

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).

Esempio 2.34. La formula $p \vee \neg p$ è valida. La formula $p \wedge \neg p$ è insoddisfacibile. La formula $p \to \neg p$ è soddisfacibile ma non valida.

Più in generale, per ogni formula F la formula $F \vee \neg F$ è valida, mentre $F \wedge \neg F$ è insoddisfacibile.

Esercizio 2.35. Dimostrate che ogni lettera proposizionale è soddisfacibile e non valida.

Nota 2.36. Osserviamo che se $F \equiv G$ e F è valida (soddisfacibile, insoddisfacibile), allora G è valida (soddisfacibile, insoddisfacibile).

Un importante legame tra validità e insoddisfacibilità è contenuto nel prossimo teorema.

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.

(b) Si può ragionare come in (a), oppure osservare che per (a) $\neg F$ è valida se e solo se $\neg\neg F$ è insoddisfacibile: per il lemma 2.20.1 $\neg\neg F \equiv F$ e per la nota 2.36 l'ultima condizione è equivalente alla insoddisfacibilità di F. □

Esempio 2.38. Verifichiamo che la formula $(p \to q) \wedge (q \to \neg p)$, che indicheremo con F, è soddisfacibile.

A questo scopo ci basta trovare un'interpretazione v che soddisfa F. Ponendo $v(p) = \mathbb{F}$ e $v(q) = \mathbb{V}$ è facile verificare che vale $v(F) = \mathbb{V}$.

Notiamo che anche l'interpretazione v' tale che $v'(p) = \mathbb{F}$ e $v'(q) = \mathbb{F}$ soddisfa F: comunque questo ulteriore esempio è superfluo, dato che una sola interpretazione è sufficiente a mostrare la soddisfacibilità di F.

Esempio 2.39. Verifichiamo che la formula $(p \to q) \vee (p \wedge (q \to \neg q))$, che indicheremo con G, è valida.

A questo scopo consideriamo un'interpretazione arbitraria v. Se $v(p \to q) = \mathbb{V}$ allora v soddisfa G. Se v non soddisfa $p \to q$ allora deve essere $v(p) = \mathbb{V}$ e $v(q) = \mathbb{F}$. Allora v soddisfa $q \to \neg q$ e quindi anche $p \wedge (q \to \neg q)$. Ma allora anche in questo caso $v(G) = \mathbb{V}$. Abbiamo dunque dimostrato che qualunque interpretazione soddisfa G, cioè che G è valida.

Per il teorema 2.37 abbiamo anche dimostrato che $\neg G$ è insoddisfacibile.

Gli esempi 2.38 e 2.39 mostrano una differenza essenziale tra le dimostrazioni di soddisfacibilità e quelle di validità: nelle prime è sufficiente costruire **una singola** interpretazione che soddisfa la formula in esame, nelle seconde si devono considerare **tutte** le possibili interpretazioni e, ragionando astrattamente, mostrare che la formula è soddisfatta da ognuna di esse. Da questo punto di vista le dimostrazioni di insoddisfacibilità di una formula sono simili a quelle di validità: in questo caso si tratta di dimostrare che la formula non è soddisfatta in tutte le interpretazioni.

La discussione appena fatta è molto simile a quella svolta dopo l'esempio 2.17 a proposito delle dimostrazioni di conseguenza logica e di non conseguenza logica. In effetti esiste uno stretto legame tra conseguenza logica e validità/soddisfacibilità, come evidenziato dal prossimo lemma.

LEMMA 2.40. *Siano F e G formule.*

(a) *$F \models G$ se e solo se $F \to G$ è valida.*
(b) *$F \not\models G$ se e solo se $F \wedge \neg G$ è soddisfacibile.*

DIMOSTRAZIONE. (a) Supponiamo $F \models G$ e dimostriamo la validità di $F \to G$. Sia v un'interpretazione qualunque. Se $v(F) = \mathbb{F}$ allora $v(F \to G) = \mathbb{V}$. Se invece $v(F) = \mathbb{V}$ dalla nostra ipotesi segue $v(G) = \mathbb{V}$ e quindi v soddisfa $F \to G$ anche in questo caso.

Per l'implicazione inversa supponiamo che $F \to G$ sia valida e fissiamo un'interpretazione v che soddisfa F. Visto che $v(F \to G) = \mathbb{V}$ deve essere $v(G) = \mathbb{V}$: abbiamo dunque dimostrato che $F \models G$.

(b) Per (a) abbiamo che $F \not\models G$ se e solo se $F \to G$ non è valida. Per il teorema 2.37 quest'ultima condizione è equivalente alla soddisfacibilità di $\neg(F \to G)$. Dato che per il lemma 2.24.4 $\neg(F \to G) \equiv F \wedge \neg G$ la dimostrazione è completa. $\square$

ESERCIZIO 2.41. Alcune delle seguenti affermazioni riguardanti le formule F e G sono corrette, altre no. Dimostrate le prime e trovate un controesempio alle seconde.

(a) Se sia F che G sono soddisfacibili allora $F \wedge G$ è soddisfacibile;
(b) se almeno una tra F e G è soddisfacibile allora $F \vee G$ è soddisfacibile;
(c) $G \not\models \neg F$ se e solo se $F \wedge G$ è soddisfacibile.
(d) $F \models \neg G$ se e solo se $F \not\models G$;
(e) se F non è valido allora $\neg F$ è valido.

Consideriamo ora validità e soddisfacibilità per insiemi di formule.

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).

Nel caso in cui T è finito ci si può ricondurre al caso di una singola formula.

LEMMA 2.43. *Se $T = \{F_1, \dots, F_n\}$ è un insieme finito di formule allora la validità (soddisfacibilità, insoddisfacibilità) di T è equivalente alla validità (soddisfacibilità, insoddisfacibilità) della formula $F_1 \wedge \cdots \wedge F_n$.*

DIMOSTRAZIONE. Immediata dalle definizioni. $\square$

NOTA 2.44. Notiamo che un insieme di formule è valido se e solo se tutti i suoi elementi sono validi. La proprietà analoga non vale però per soddisfacibilità e insoddisfacibilità: l'insieme $\{p, \neg p\}$ è insoddisfacibile, pur essendo ognuno dei suoi elementi soddisfacibile. Se però un insieme di formule è soddisfacibile allora tutti i suoi elementi sono soddisfacibili. D'altra parte se una formula è insoddisfacibile, allora tutti gli insiemi che la contengono sono insoddisfacibili.

ESERCIZIO 2.45. ($\star$) Dimostrate che l'insieme

$$T = \{ p_i \rightarrow \neg p_{i+1} \,:\, i \in \mathbb{N} \} \cup \{ \neg p_i \rightarrow p_{i+1} \,:\, i \in \mathbb{N} \}$$

è soddisfacibile e che l'insieme T, p_1, p_4 (si veda la notazione 1.6) è insoddisfacibile.

NOTAZIONE 2.46. $\vDash F$ sta ad indicare $\emptyset \vDash F$, dove $\emptyset$ è l'insieme vuoto.

LEMMA 2.47. *Sia F una formula e T un insieme di formule.*

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

DIMOSTRAZIONE. (a) $\vDash F$ significa che per ogni interpretazione che soddisfa tutti gli elementi di $\emptyset$ soddisfa anche F. Ma qualunque interpretazione soddisfa tutti gli elementi di $\emptyset$ (perché non ce ne sono), e perciò questa condizione è equivalente alla validità di F.

(b) $T, \neg F$ insoddisfacibile significa che nessuna interpretazione soddisfa sia le formule di T che $\neg F$, cioè che se un'interpretazione v soddisfa tutte le formule di T deve essere $v(\neg F) = \mathbb{F}$. Questo è equivalente a affermare che se un'interpretazione soddisfa tutte le formule di T allora soddisfa anche F, cioè $T \vDash F$. □

4. Una procedura di decisione: le tavole di verità

Iniziamo specificando cosa intendiamo per procedura di decisione nel nostro contesto.

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 convertita in una procedura di decisione per l'altro. Ad esempio se abbiamo una procedura di decisione per la validità, per testare la soddisfacibilità di F è sufficiente applicare la procedura di decisione a $\neg F$ e ribaltare la risposta. Grazie al lemma 2.40 un tale algoritmo può essere anche applicato anche per verificare se una formula è conseguenza logica di un'altra. Il lemma 2.43 ci permette di utilizzare una procedura di decisione di questo tipo anche per insiemi finiti di formule.

Nella logica proposizionale esiste una procedura di decisione piuttosto semplice e naturale per la validità e per la soddisfacibilità. Il metodo delle tavole di verità si basa sul lemma 2.8: per stabilire se un'interpretazione v soddisfa una formula F basta conoscere i valori assunti da v sulle lettere proposizionali che compaiono in F. Dato che in F compaiono solo un numero finito di lettere proposizionali, per stabilire se F è valida è sufficiente considerare tutte le possibili combinazioni di valori di verità assegnati dalle valutazioni a queste lettere.

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 colonna 2^{n-1} $\mathbb{V}$ sono seguiti da 2^{n-1} $\mathbb{F}$, nella seconda colonna si alternano 4 blocchi di lunghezza 2^{n-2} di $\mathbb{V}$ e di $\mathbb{F}$, e così via fino alla colonna n-esima in cui il valore di verità cambia ad ogni riga). La colonna corrispondente a F contiene $\mathbb{V}$ o $\mathbb{F}$ a seconda se l'interpretazione generata dalla valutazione che compare in quella riga soddisfa o meno F.

Se la colonna di F contiene solo $\mathbb{V}$ allora F è valida, se contiene almeno un $\mathbb{V}$ allora F è soddisfacibile, se contiene solo $\mathbb{F}$ allora F è insoddisfacibile.

In pratica è comodo avere a disposizione colonne supplementari, usualmente posizionate tra quelle delle lettere proposizionali e quella di F, in cui calcolare i valori di verità di opportune sottoformule di F.

Esempio 2.50. Usiamo l'algoritmo delle tavole di verità per verificare che la formula $(p \to q) \vee (p \to \neg q)$ dell'esempio 2.38 è valida.

p	q	$p \to q$	$\neg q$	$p \to \neg q$	$(p \to q) \vee (p \to \neg q)$
$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$
$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$
$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$

Come si nota l'ultima colonna è composta interamente da $\mathbb{V}$.

Esempio 2.51. Usiamo l'algoritmo delle tavole di verità per stabilire se la formula (che per comodità indichiamo con F) $(\neg p \vee q) \wedge (q \to \neg r \wedge \neg p) \wedge (p \vee r)$ è soddisfacibile.

p	q	r	$\neg p \vee q$	$\neg r \wedge \neg p$	$q \to \neg r \wedge \neg p$	$p \vee r$	F
$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{F}$
$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{F}$
$\mathbb{V}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$
$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$

Nella riga corrispondente alla valutazione $v(p) = \mathbb{F}$, $v(q) = \mathbb{F}$, $v(r) = \mathbb{V}$ nell'ultima colonna compare $\mathbb{V}$. Perciò esiste un'interpretazione che soddisfa F e la formula è soddisfacibile.

L'algoritmo delle tavole di verità è piuttosto inefficiente: per stabilire se una formula con n lettere proposizionali è valida è necessario compilare 2^n righe. Esistono metodi di decisione per la validità delle formule proposizionali che molto spesso sono più efficienti (ad esempio quello cui è dedicato il capitolo 4), ma non è stato trovato un algoritmo che sia più efficiente in ogni caso. L'esistenza di un metodo di decisione siffatto è equivalente ad una risposta positiva all'importante problema $P = NP$, che viene discusso nel corso di Fondamenti dell'Informatica.

Esempio 2.52. Usiamo l'algoritmo delle tavole di verità per verificare che la formula $p \wedge \neg q \to p \wedge q$ è conseguenza logica di $\neg p$. Per il lemma 2.40 basta verificare che $\neg p \to (p \wedge \neg q \to p \wedge q)$ è valida. Una piccola semplificazione consiste nel calcolare le tavole di verità delle due formule e verificare che quando $\neg p$ è soddisfatta lo è anche $p \wedge \neg q \to p \wedge q$:

p	q	$\neg p$	$p \wedge \neg q$	$p \wedge q$	$p \wedge \neg q \to p \wedge q$
$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$
$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$

Un'ulteriore semplificazione consiste nel non calcolare il valore di verità di $p \wedge \neg q \rightarrow p \wedge q$ quando si è verificato che $v(\neg p) = \mathbb{F}$:

p	q	$\neg p$	$p \wedge \neg q$	$p \wedge q$	$p \wedge \neg q \rightarrow p \wedge q$
$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$			
$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$			
$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$

ESEMPIO 2.53. Usiamo l'algoritmo delle tavole di verità per verificare che le formule $p \rightarrow (q \wedge \neg q)$ e $\neg p$ sono logicamente equivalenti. Per i lemmi 2.19 e 2.40 basta dimostrare che $(p \rightarrow q \wedge \neg q) \rightarrow \neg p$ e $\neg p \rightarrow (p \rightarrow q \wedge \neg q)$ sono entrambe valide. Una strada più breve è calcolare le tavole di verità delle due formule e verificare che $p \rightarrow q \wedge \neg q$ e $\neg p$ hanno sempre lo stesso valore di verità:

p	q	$q \wedge \neg q$	$p \rightarrow q \wedge \neg q$	$\neg p$
$\mathbb{V}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{V}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{V}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{V}$	$\mathbb{V}$

ESERCIZIO 2.54. Dimostrare con le tavole di verità che $p \rightarrow (q \rightarrow r)$ e $p \wedge q \rightarrow r$ sono logicamente equivalenti.

ESERCIZIO 2.55. Stabilite se le seguenti formule sono valide, soddisfacibili, insoddisfacibili (usate sia le tavole di verità che le definizioni):

$$(p \rightarrow q) \wedge \neg q \rightarrow \neg p;$$
$$(p \rightarrow q) \rightarrow (p \rightarrow \neg q);$$
$$(p \vee q \rightarrow r) \vee p \vee q;$$
$$(p \vee q) \wedge (p \rightarrow r \wedge q) \wedge (q \rightarrow \neg r \wedge p);$$
$$(p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r)).$$

ESERCIZIO 2.56. Stabilite se le seguenti conseguenze ed equivalenze logiche sono corrette (usate sia le tavole di verità che le definizioni):

$$p \rightarrow q \models \neg p \rightarrow \neg q;$$
$$(p \rightarrow q) \wedge \neg q \models \neg p;$$
$$(p \vee q) \wedge (\neg p \rightarrow \neg q) \equiv q;$$
$$(p \vee q) \wedge (\neg p \rightarrow \neg q) \equiv p.$$

5. Traduzioni dal linguaggio naturale

In questa sezione ci occuperemo di tradurre frasi del linguaggio naturale (nel nostro caso, l'italiano) in formule della logica proposizionale e viceversa.

ESEMPIO 2.57. Consideriamo un linguaggio proposizionale in cui p significa "Paola è contenta", q significa "Paola dipinge un quadro" e r significa "Renzo è contento". La formula proposizionale $p \wedge q \rightarrow \neg r$ viene interpretata come "se Paola è contenta e dipinge un quadro allora Renzo non è contento". La formula $p \rightarrow q$ viene interpretata come "se Paola è contenta allora dipinge un quadro", ma anche come "Paola è contenta soltanto se dipinge un quadro".

Queste due diverse traduzioni della stessa formula mostrano che l'italiano, come qualunque altro linguaggio naturale, consente di aggiungere sfumature che il linguaggio formale non è in grado di esprimere (un ulteriore esempio è fornito dal fatto che dal punto di vista logico non c'è differenza tra "e" e "ma"). D'altro canto il linguaggio formale permette una maggior precisione, ed evita le ambiguità insite nei linguaggi naturali (e che per certi versi ne costituiscono la ricchezza, permettendo ad esempio il linguaggio poetico).

La traduzione dal linguaggio formale al linguaggio naturale non presenta in genere difficoltà, mentre la direzione inversa è spesso più delicata.

Esempio 2.58. Consideriamo un linguaggio proposizionale in cui p significa "Pietro sarà eletto leader del partito", r significa "Raffaella si dimetterà", m significa "Mario si dimetterà" e v significa "vinceremo le elezioni". La frase "vinceremo le elezioni, se Pietro sarà eletto leader del partito" può venir tradotta dalla formula proposizionale $p \to v$. La frase "solo se Pietro sarà eletto leader del partito vinceremo le elezioni" viene tradotta da $v \to p$ (la frase è equivalente ad affermare che se Pietro non verrà eletto leader del partito le elezioni saranno certamente perse, e per il Lemma 2.24.5 $v \to p \equiv \neg p \to \neg v$). La frase "se Pietro non sarà eletto leader del partito, allora Raffaella o Mario si dimetteranno e non vinceremo le elezioni" ha come traduzione $\neg p \to (r \vee m) \wedge \neg v$.

Esempio 2.59. Consideriamo un linguaggio proposizionale in cui p significa "x è primo" e d significa "x è dispari". "Una condizione sufficiente perché x sia primo è che x sia dispari" viene tradotto come $d \to p$. "Una condizione necessaria perché x sia primo è che x sia dispari" viene tradotto come $p \to d$. "Una condizione necessaria e sufficiente perché x sia primo è che x sia dispari" viene tradotto come $(d \to p) \wedge (p \to d)$.

Esercizio 2.60. Introducendo opportuni linguaggi proposizionali, traducete le frasi seguenti:

(a) "Se l'algoritmo termina abbiamo un risultato, e se abbiamo un risultato lo stampiamo", "se l'algoritmo termina abbiamo un risultato e lo stampiamo", "non è possibile che l'algoritmo non termini", "non è possibile che l'algoritmo termini ma non dia un risultato".

(b) "Patrizia va al cinema solo se Roberta ci va", "se Roberta va al cinema, anche Patrizia ci va", "al massimo una tra Roberta e Patrizia va al cinema".

(c) "Se il Signor Rossi è felice, la Signora Rossi è felice", "se il Signor Rossi è infelice, la Signora Rossi è infelice".

La logica proposizionale ha degli evidenti limiti espressivi, già notati all'inizio del capitolo 1. Malgrado questo è possibile utilizzarla per analizzare alcuni tipi di ragionamento e stabilirne la correttezza (o meno).

Esempio 2.61. Supponiamo di sapere che:

- se Paolo ha la barba allora Carlo non è biondo oppure Roberta non studia Informatica;
- se Roberta studia Informatica allora Sara ha gli occhi neri;
- se Sara ha gli occhi neri e Carlo è biondo allora Paolo ha la barba;
- Carlo è biondo.

Possiamo dedurre che Roberta non studia Informatica?

Se indichiamo "Paolo ha la barba", "Carlo è biondo", "Roberta studia Informatica" e "Sara ha gli occhi neri" rispettivamente con p, q, r e s allora dobbiamo verificare se

$$p \to \neg q \vee \neg r, r \to s, s \wedge q \to p, q \models \neg r.$$

Il lettore può stabilire se questa conseguenza logica sussista, utilizzando a sua scelta il ragionamento diretto basato sulle definizioni oppure il metodo delle tavole di verità (che richiede 16 righe).

CAPITOLO 3

Forma normale congiuntiva e disgiuntiva

In questo capitolo ci occuperemo di trasformare una formula proposizionale in una formula ad essa logicamente equivalente che abbia una forma particolarmente semplice dal punto di vista sintattico, per cui è particolarmente facile verificare la soddisfacibilità o la validità. Questa trasformazione è di tipo algoritmico, e presenteremo uno dei molti algoritmi che la effettuano. L'algoritmo che introdurremo, come la maggior parte di quelli alternativi, è non deterministico (ad ogni passo abbiamo diverse scelte possibili), ma gode della proprietà della terminazione forte (termina qualunque sia la successione delle scelte).

1. Definizione di forma normale congiuntiva e disgiuntiva

Iniziamo con il definire una classe di formule piuttosto semplici.

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 nel seguente lemma.

Lemma 3.3. *Un insieme di letterali è soddisfacibile se e solo se non contiene nessuna coppia complementare.*

Dimostrazione. Iniziamo con il dimostrare la direzione da sinistra a destra: è ovvio che un insieme soddisfacibile di formule non può contenere coppie complementari, dato che non può esistere un'interpretazione che renda vera sia una formula che il suo complemento.

Viceversa sia T un insieme di letterali che non contiene nessuna coppia complementare. Definiamo un'interpretazione v ponendo

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

Verifichiamo che v soddisfa tutte le formule (che sono letterali) di T. Se $p \in T$ è immediato dalla definizione di v che $v(p) = \mathbb{V}$. Se invece $\neg p \in T$ non può essere $p \in T$ (perché T non contiene coppie complementari) e quindi $v(p) = \mathbb{F}$, cioè $v(\neg p) = \mathbb{V}$. □

Nota 3.4. Notiamo che la direzione da destra a sinistra del lemma 3.3 è falsa se a "insieme di letterali" sostituiamo "insieme di formule": $\{p \wedge \neg p\}$ non contiene coppie complementari, ma è insoddisfacibile. La direzione da sinistra a destra è invece vera anche per insiemi di formule (nella dimostrazione di quella direzione non si è fatto alcun uso del fatto che gli elementi dell'insieme fossero tutti letterali).

Esercizio 3.5. Dimostrate che una disgiunzione di letterali è valida se e solo se tra i disgiunti vi è una coppia complementare.

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.

Esempio 3.7. La formula $(p \vee \neg q \vee r) \wedge (p \vee \neg s) \wedge (\neg t \vee \neg w)$ è in forma normale congiuntiva con $m = 3$ e $h_1 = 3$, $h_2 = h_3 = 2$.

La formula $\neg p \vee (\neg q \wedge \neg r \wedge s) \vee (p \wedge \neg s)$ è in forma normale disgiuntiva con $m = 3$ e $h_1 = 1$, $h_2 = 3$, $h_3 = 2$.

La formula $p \wedge \neg r$ è sia in forma normale congiuntiva che in forma normale disgiuntiva (nel primo caso $m = 2$ e $h_1 = h_2 = 1$, nel secondo caso $m = 1$ e $h_1 = 2$).

La formula $p \vee \neg\neg q$ non è né in forma normale congiuntiva né in forma normale disgiuntiva (perché $\neg\neg q$ non è un letterale).

La formula $p \wedge \neg(q \vee r)$ non è né in forma normale congiuntiva né in forma normale disgiuntiva.

Ogni formula che contiene qualche occorrenza di $\rightarrow$ non è né in forma normale congiuntiva né in forma normale disgiuntiva.

Nota 3.8. Una delle ragioni dell'importanza delle forme normali congiuntiva e disgiuntiva è la semplicità con cui (basandosi sul fatto che il controllo della soddisfazione dei letterali è immediato) si può verificare se un'interpretazione soddisfa o meno una formula di questo tipo. Nel caso della forma normale disgiuntiva basta verificare se esiste un disgiunto i cui letterali sono tutti soddisfatti dall'interpretazione. Nel caso della forma normale congiuntiva basta verificare se in ogni congiunto esiste almeno un letterale soddisfatto dall'interpretazione.

Esercizio 3.9. Dimostrate che una formula in forma normale disgiuntiva è soddisfacibile se e solo se almeno uno dei disgiunti non contiene una coppia complementare di letterali. (Usate il lemma 3.3.)

Dimostrate che una formula in forma normale congiuntiva è valida se e solo se ogni congiunto contiene almeno una coppia complementare di letterali. (Usate l'esercizio 3.5.)

L'obiettivo di questo capitolo è dimostrare il seguente teorema.

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 che il teorema verrà dimostrato attraverso la descrizione di due algoritmi che preso come input F, forniscono come output rispettivamente G_1 e G_2.

Le formule G_1 e G_2 la cui esistenza è asserita dal teorema 3.10 non sono uniche. Ad esempio p e $(\neg q \vee p) \wedge (q \vee p)$ sono due formule in forma normale congiuntiva logicamente equivalenti alla formula $q \vee \neg q \rightarrow p$.

Nota 3.11. Una volta dimostrato il teorema 3.10 avremo nuove procedure di decisione per la validità e la soddisfacibilità delle formule proposizionali. Per la validità, data una formula possiamo trasformarla in una logicamente equivalente e in forma normale congiuntiva usando l'algoritmo 3.18 e poi utilizzare la seconda parte dell'esercizio 3.9 per "leggere" se la formula è valida. Per la soddisfacibilità, utilizziamo l'algoritmo 3.22 per trasformare l'input in una formula logicamente equivalente in forma normale disgiuntiva, e poi utilizziamo la prima parte dell'esercizio 3.9.

2. Doppie negazioni, α-formule e β-formule

Esistono diversi algoritmi per la trasformazione in forma normale congiuntiva e disgiuntiva delle formule proposizionali e quindi per la dimostrazione del teorema 3.10. Gli algoritmi che utilizzeremo sono dovuti a Melvin Fitting[1] e si basano su una classificazione delle formule proposizionali che ci sarà utile anche in altri capitoli.

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.

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

β-formula	ridotti	
$F \vee G$	F	G
$\neg(F \wedge G)$	$\neg F$	$\neg G$
$F \to G$	$\neg F$	G

LEMMA 3.14. *Ogni doppia negazione è logicamente equivalente al suo ridotto. Ogni α-formula è logicamente equivalente alla congiunzione dei suoi ridotti. Ogni β-formula è logicamente equivalente alla disgiunzione dei suoi ridotti.*

DIMOSTRAZIONE. L'affermazione riguardo alle doppie negazioni è il lemma 2.20.1. Per le α e β-formule contenute nelle prime righe delle due tabelle l'affermazione è ovvia, mentre per quelle delle altre righe l'equivalenza logica è contenuta nel lemma 2.24. $\square$

LEMMA 3.15. *Una formula proposizionale è di uno e uno solo dei tipi seguenti:*
- *un letterale,*
- *una doppia negazione,*
- *una α-formula,*
- *una β-formula.*

DIMOSTRAZIONE. Per il lemma 1.8 ogni formula F è una lettera proposizionale, una negazione, una congiunzione, una disgiunzione o un'implicazione.
- Se F è una lettera proposizionale allora è un letterale;
- Se F è una negazione $\neg G$ dobbiamo considerare di che tipo è G:
  - se G è una lettera proposizionale allora F è un letterale;
  - se G è una negazione allora F è una doppia negazione;
  - se G è una congiunzione allora F è una β-formula;
  - se G è una disgiunzione oppure un'implicazione allora F è una α-formula.

[1]Melvin Fitting (1942–) è un logico statunitense.

- Se F è una congiunzione allora è una α-formula.
- Se F è una disgiunzione oppure un'implicazione allora è una β-formula.

□

3. Gli algoritmi di Fitting

Conviene ricordare alcune equivalenze logiche dimostrate nel capitolo 2 (lemma 2.24), che saranno utili per giustificare i passaggi degli algoritmi di Fitting:

LEMMA 3.16. *Siano F, G e H formule proposizionali qualunque. Valgono le seguenti equivalenze logiche:*

(1) $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$;
(2) $(G \vee H) \wedge F \equiv (G \wedge F) \vee (H \wedge F)$;
(3) $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$;
(4) $(G \wedge H) \vee F \equiv (G \vee F) \wedge (H \vee F)$.

Per descrivere gli algoritmi di Fitting useremo la seguente notazione.

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 $G_1, \dots, G_n$ 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 $(((G_1 \wedge G_2) \wedge \dots G_{n-1}) \wedge G_n))$. Quest'ultima formula è chiamata la *congiunzione generalizzata* di $G_1, \dots, G_n$ e in questo capitolo la scriviamo $\langle G_1, \dots, G_n \rangle$.

Analogamente, a partire da commutatività e associatività di $\vee$, si definisce la *disgiunzione generalizzata* di $G_1, \dots, G_n$, che in questo capitolo indichiamo con $[G_1, \dots, G_n]$.

Usando congiunzioni e disgiunzioni generalizzate, una formula in forma normale congiuntiva ha la forma

$$\langle [G_{1,1}, \dots, G_{1,h_1}], \dots, [G_{m,1}, \dots, G_{m,h_m}] \rangle$$

dove ogni $G_{i,j}$ è un letterale. Similmente, una formula in forma normale disgiuntiva ha la forma

$$[\langle G_{1,1}, \dots, G_{1,h_1} \rangle, \dots, \langle G_{m,1}, \dots, G_{m,h_m} \rangle]$$

con ogni $G_{i,j}$ letterale.

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 sceglie uno, che indichiamo con G. Per il lemma 3.15 ci sono tre possibilità:

(1) G è una doppia negazione con ridotto H: in questo caso si sostituisce G con H nel congiunto in cui appariva G; gli altri congiunti restano immutati.
(2) G è una β-formula e i suoi ridotti sono G_1 e G_2: in questo caso si sostituisce G con G_1, G_2 nel congiunto in cui appariva G; gli altri congiunti restano immutati.
(3) G è una α-formula e i suoi ridotti sono G_1 e G_2: in questo caso si sostituisce il congiunto in cui appariva G con due nuovi congiunti; nel primo congiunto G è sostituito da G_1, nel secondo congiunto G è sostituito da G_2 (e in entrambi i casi gli altri disgiunti restano immutati); gli altri congiunti restano immutati.

Osserviamo che la congiunzione generalizzata di disgiunzioni generalizzate di formule così ottenuta è sempre logicamente equivalente a quella precedente: nel primo e nel secondo caso per il lemma 3.14, nel terzo per il lemma 3.14 e per (3) e (4) del lemma 3.16. Quindi la formula ottenuta ad ogni passo di esecuzione dell'algoritmo è logicamente equivalente alla F di partenza.

Esempio 3.19. Applichiamo l'algoritmo 3.18 alla formula

$$(r \wedge \neg s) \vee \neg(p \to \neg q).$$

Sulla sinistra abbiamo la formula (congiunzione generalizzata di disgiunzioni generalizzate di formule) a cui siamo arrivati ad ogni passo dell'applicazione dell'algoritmo, nella colonna centrale indichiamo la formula (che non è un letterale) su cui agiamo per effettuare il passo successivo, nella colonna di destra il tipo di formula (α, β o doppia negazione, indicata da $\neg\neg$).

$$\begin{array}{c|c|c}
\langle[(r \wedge \neg s) \vee \neg(p \to \neg q)]\rangle & (r \wedge \neg s) \vee \neg(p \to \neg q) & \beta \\
\langle[r \wedge \neg s, \neg(p \to \neg q)]\rangle & r \wedge \neg s & \alpha \\
\langle[r, \neg(p \to \neg q)], [\neg s, \neg(p \to \neg q)]\rangle & \neg(p \to \neg q) & \alpha \\
\langle[r, p], [r, \neg\neg q], [\neg s, \neg(p \to \neg q)]\rangle & \neg\neg q & \neg\neg \\
\langle[r, p], [r, q], [\neg s, \neg(p \to \neg q)]\rangle & \neg(p \to \neg q) & \alpha \\
\langle[r, p], [r, q], [\neg s, p], [\neg s, \neg\neg q]\rangle & \neg\neg q & \neg\neg \\
\langle[r, p], [r, q], [\neg s, p], [\neg s, q]\rangle & &
\end{array}$$

La formula di partenza è quindi equivalente alla formula in forma normale congiuntiva $\langle[r, p], [r, q], [\neg s, p], [\neg s, q]\rangle$, cioè a

$$(r \vee p) \wedge (r \vee q) \wedge (\neg s \vee p) \wedge (\neg s \vee q).$$

Esempio 3.20. Applichiamo l'algoritmo 3.18 alla formula

$$(p \to \neg q) \vee \neg(r \wedge s \to \neg(\neg r \vee t)).$$

Questa volta omettiamo l'indicazione della formula su cui si effettua la riduzione.

$$\begin{array}{c|c}
\langle[(p \to \neg q) \vee \neg(r \wedge s \to \neg(\neg r \vee t))]\rangle & \beta \\
\langle[p \to \neg q, \neg(r \wedge s \to \neg(\neg r \vee t))]\rangle & \beta \\
\langle[\neg p, \neg q, \neg(r \wedge s \to \neg(\neg r \vee t))]\rangle & \alpha \\
\langle[\neg p, \neg q, r \wedge s], [\neg p, \neg q, \neg\neg(\neg r \vee t)]\rangle & \neg\neg \\
\langle[\neg p, \neg q, r \wedge s], [\neg p, \neg q, \neg r \vee t]\rangle & \alpha \\
\langle[\neg p, \neg q, r], [\neg p, \neg q, s], [\neg p, \neg q, \neg r \vee t]\rangle & \beta \\
\langle[\neg p, \neg q, r], [\neg p, \neg q, s], [\neg p, \neg q, \neg r, t]\rangle &
\end{array}$$

La formula in forma normale congiuntiva ottenuta è

$$(\neg p \vee \neg q \vee r) \wedge (\neg p \vee \neg q \vee s) \wedge (\neg p \vee \neg q \vee \neg r \vee t).$$

Esercizio 3.21. Applicate l'algoritmo 3.18 alla formula

$$\neg(\neg p \to q) \vee (r \wedge \neg s) \to \neg t.$$

L'algoritmo di Fitting per la trasformazione in forma normale disgiuntiva è duale a quello per la forma normale congiuntiva: congiunzioni generalizzate e disgiunzioni generalizzate sono scambiate e il ruolo delle α-formule e delle β-formule è invertito.

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 letterali la formula è in forma normale disgiuntiva e l'algoritmo si arresta. Se esistono elementi di queste congiunzioni generalizzate che non sono letterali se ne sceglie uno, che indichiamo con G. Per il lemma 3.15 ci sono tre possibilità:

(1) G è una doppia negazione con ridotto H: in questo caso si sostituisce G con H nel disgiunto in cui appariva G; gli altri disgiunti restano immutati.
(2) G è una α-formula e i suoi ridotti sono G_1 e G_2: in questo caso si sostituisce G con G_1, G_2 nel disgiunto in cui appariva G; gli altri disgiunti restano immutati.
(3) G è una β-formula e i suoi ridotti sono G_1 e G_2: in questo caso si sostituisce il disgiunto in cui appariva G con due nuovi disgiunti; nel primo disgiunto G è sostituito da G_1, nel secondo disgiunto G è sostituito da G_2 (e in entrambi i casi gli altri congiunti restano immutati); gli altri disgiunti restano immutati.

Osserviamo che la disgiunzione generalizzata di congiunzioni generalizzate di formule così ottenuta è sempre logicamente equivalente a quella precedente: nel primo e nel secondo caso per il lemma 3.14, nel terzo per il lemma 3.14 e per (1) e (2) del lemma 3.16. Quindi la formula ottenuta ad ogni passo di esecuzione dell'algoritmo è logicamente equivalente alla F di partenza.

Esempio 3.23. Applichiamo l'algoritmo 3.22 alla formula dell'esempio 3.19.

$$
\begin{array}{c|c|c}
[\langle (r \wedge \neg s) \vee \neg(p \to \neg q)\rangle] & (r \wedge \neg s) \vee \neg(p \to \neg q) & \beta \\
[\langle r \wedge \neg s\rangle, \langle \neg(p \to \neg q)\rangle] & r \wedge \neg s & \alpha \\
[\langle r, \neg s\rangle, \langle \neg(p \to \neg q)\rangle] & \neg(p \to \neg q) & \alpha \\
[\langle r, \neg s\rangle, \langle p, \neg\neg q\rangle] & \neg\neg q & \neg\neg \\
[\langle r, \neg s\rangle, \langle p, q\rangle] & &
\end{array}
$$

La formula di partenza è quindi equivalente alla formula in forma normale disgiuntiva $[\langle r, \neg s\rangle, \langle p, q\rangle]$, cioè a $(r \wedge \neg s) \vee (p \wedge q)$.

Esempio 3.24. Applichiamo l'algoritmo 3.22 alla formula dell'esempio 3.20.

$$
\begin{array}{c|c}
[\langle (p \to \neg q) \vee \neg(r \wedge s \to \neg(\neg r \vee t))\rangle] & \beta \\
[\langle p \to \neg q\rangle, \langle \neg(r \wedge s \to \neg(\neg r \vee t))\rangle] & \beta \\
[\langle \neg p\rangle, \langle \neg q\rangle, \langle \neg(r \wedge s \to \neg(\neg r \vee t))\rangle] & \alpha \\
[\langle \neg p\rangle, \langle \neg q\rangle, \langle r \wedge s, \neg\neg(\neg r \vee t)\rangle] & \neg\neg \\
[\langle \neg p\rangle, \langle \neg q\rangle, \langle r \wedge s, \neg r \vee t\rangle] & \alpha \\
[\langle \neg p\rangle, \langle \neg q\rangle, \langle r, s, \neg r \vee t\rangle] & \beta \\
[\langle \neg p\rangle, \langle \neg q\rangle, \langle r, s, \neg r\rangle, \langle r, s, t\rangle] &
\end{array}
$$

La formula in forma normale disgiuntiva ottenuta è

$$\neg p \vee \neg q \vee (r \wedge s \wedge \neg r) \vee (r \wedge s \wedge t).$$

Esercizio 3.25. Applicate l'algoritmo 3.22 alla formula dell'esercizio 3.21.

Esercizio 3.26. Applicate gli algoritmi 3.18 e 3.22 alle formule:

$$
\begin{array}{cc}
(p \to q) \to (q \to \neg r); & \neg(p \wedge q \wedge (r \to s)); \\
\neg((p \to q) \wedge (q \to p)); & \neg(p \to \neg q) \to (p \wedge q); \\
(\neg q \to p) \vee \neg(s \wedge q \to \neg p); & \neg(\neg(p \to q) \wedge (r \vee s \to \neg t)).
\end{array}
$$

Entrambi gli algoritmi di Fitting (3.18 e 3.22) sono non deterministici.

Esempio 3.27. La seguente tabella presenta due diverse applicazioni dell'algoritmo 3.18 alla formula $(p \wedge q) \vee \neg(r \wedge s)$:

$$\begin{array}{c|c}
\langle[(p \wedge q) \vee \neg(r \wedge s)]\rangle & \langle[(p \wedge q) \vee \neg(r \wedge s)]\rangle \\
\langle[p \wedge q, \neg(r \wedge s)]\rangle & \langle[p \wedge q, \neg(r \wedge s)]\rangle \\
\langle[p \wedge q, \neg r, \neg s]\rangle & \langle[p, \neg(r \wedge s)], [q, \neg(r \wedge s)]\rangle \\
\langle[p, \neg r, \neg s], [q, \neg r, \neg s]\rangle & \langle[p, \neg r, \neg s], [q, \neg(r \wedge s)]\rangle \\
 & \langle[p, \neg r, \neg s], [q, \neg r, \neg s]\rangle
\end{array}$$

La forma normale congiuntiva ottenuta è la stessa in entrambi i casi ma nella prima colonna al secondo passo si è operato sulla β-formula $\neg(r \wedge s)$, mentre nella seconda colonna si è operato sulla α-formula $p \wedge q$. Questo ha portato ad ottenere il risultato finale in tre e in quattro passi rispettivamente.

Esercizio 3.28. Trovate due diverse applicazioni dell'algoritmo 3.22 alla formula

$$(p \rightarrow \neg q) \vee \neg(r \rightarrow s).$$

Nota 3.29. Per ridurre i tempi di esecuzione dell'algoritmo 3.18 per la forma normale congiuntiva è opportuno operare su β-formule anziché su α-formule ogniqualvolta ciò sia possibile. Dualmente, per ridurre i tempi di esecuzione dell'algoritmo 3.22 per la forma normale disgiuntiva è opportuno operare, ove possibile, su α-formule invece che su β-formule.

Nota 3.30. Durante l'esecuzione degli algoritmi 3.18 e 3.22 con carta e penna può essere utile prendere qualche scorciatoia, che permette di diminuire il numero dei passaggi. Le più comode sono le seguenti:

- quando un ridotto è una doppia negazione scrivere immediatamente la formula senza $\neg\neg$, saltando quindi il passaggio successivo in cui si applicherebbe la regola della doppia negazione;
- nell'algoritmo 3.18 si può agire simultaneamente su formule che compaiono in due congiunti diversi; dualmente, nell'algoritmo 3.22 si può agire simultaneamente su formule che compaiono in due disgiunti diversi.

4. Terminazione forte degli algoritmi di Fitting

Abbiamo descritto gli algoritmi di Fitting 3.18 e 3.22, e abbiamo osservato che se arrivano a una formula in forma normale congiuntiva (risp. disgiuntiva), e quindi si fermano, allora la formula finale è logicamente equivalente alla formula da cui sono partiti. Ciò che non abbiamo ancora dimostrato è che essi si fermano sempre. Notiamo che questo non è immediatamente evidente dalla definizione degli algoritmi. Infatti, esaminando il caso dell'algoritmo 3.22, osserviamo che quando si sceglie una β-formula la disgiunzione generalizzata che si ottiene ha un disgiunto in più di quella precedente ed è quindi, in un certo senso, una formula più complessa. Ad esempio da $[\langle p \rightarrow q, r\rangle]$ si passa a $[\langle \neg p, r\rangle, \langle q, r\rangle]$: sviluppando queste formule e calcolandone il grado secondo la Definizione 1.12 osserviamo che siamo passati da una formula di grado 2 ad una di grado 4.

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

Per il lemma 3.15 le clausole della definizione comprendono tutte le possibilità.

LEMMA 3.32. *Gli algoritmi 3.18 e 3.22 godono della proprietà della terminazione forte, cioè terminano qualunque sia la formula su cui si decide di operare ad ogni singolo passo.*

DIMOSTRAZIONE. Faremo in dettaglio la dimostrazione della terminazione forte per l'algoritmo 3.18: la dimostrazione per l'algoritmo 3.22 si ottiene da questa con ovvi cambiamenti.

Al passo s di esecuzione dell'algoritmo ci troviamo di fronte ad una congiunzione generalizzata di disgiunzioni generalizzate della forma

$$\langle [F_{1,1}, \dots, F_{1,h_1}], \dots, [F_{m,1}, \dots, F_{m,h_m}] \rangle$$

e definiamo il numero naturale

$$R(s) = 2^{\mathrm{rg}(F_{1,1})+\cdots+\mathrm{rg}(F_{1,h_1})} + \cdots + 2^{\mathrm{rg}(F_{m,1})+\cdots+\mathrm{rg}(F_{m,h_m})}.$$

Dimostreremo ora che qualunque sia la regola dell'algoritmo 3.18 utilizzata per passare dal passo s al passo $s+1$, si avrà $R(s+1) < R(s)$.

Se si agisce su una doppia negazione o su una β-formula l'esponente di uno degli addendi che compongono $R(s+1)$ è inferiore di 1 rispetto al corrispondente esponente nella somma che definisce $R(s)$. Dato che gli altri addendi sono invariati si ha $R(s+1) < R(s)$.

Se invece si agisce su una α-formula F con ridotti F_1 e F_2 la situazione è più delicata, perché in $R(s+1)$ c'è un addendo in più che in $R(s)$. Utilizzeremo la seguente disuguaglianza, che vale per ogni $a, b > 0$: $2^a + 2^b < 2^{a+b+1}$ (per dimostrarla supponete $a \geq b$ e osservate che $2^a + 2^b \leq 2^a + 2^a = 2^{a+1} < 2^{a+b+1}$). Per la disuguaglianza si ha $2^{\mathrm{rg}(F_1)} + 2^{\mathrm{rg}(F_2)} < 2^{\mathrm{rg}(F)}$ e quindi, moltiplicando entrambi i membri della disuguaglianza per 2^n, anche $2^{\mathrm{rg}(F_1)+n} + 2^{\mathrm{rg}(F_2)+n} < 2^{\mathrm{rg}(F)+n}$ per ogni numero naturale n. Quindi i due addendi nella definizione di $R(s+1)$ hanno somma minore dell'addendo di $R(s)$ che hanno rimpiazzato. Perciò, anche in questo caso, $R(s+1) < R(s)$.

In un'esecuzione dell'algoritmo 3.18 (in cui il passo 0 è quello iniziale) si ha quindi $R(0) > R(1) > \cdots$. Dato che gli $R(s)$ sono numeri naturali e non esiste una successione discendente infinita di numeri naturali, non possono esserci infiniti passi, cioè l'algoritmo termina. □

NOTA 3.33. Notiamo esplicitamente che gli algoritmi di Fitting **non** prevedono all'interno di una disgiunzione o congiunzione generalizzata la sostituzione di una formula con un'altra logicamente equivalente ad essa. Se introducessimo questa possibilità, l'algoritmo non godrebbe più della proprietà della terminazione forte e sarebbe quindi assai meno utile.

La dimostrazione del lemma 3.32 fornisce un limite superiore al numero di passi necessario all'algoritmo 3.18 per terminare: dato che la prima congiunzione generalizzata di disgiunzioni generalizzate è $\langle [F] \rangle$ si ha $R(0) = 2^{\mathrm{rg}(F)}$ e certamente l'algoritmo termina entro $2^{\mathrm{rg}(F)}$ passi. Questo limite superiore è molto grossolano: se F è la formula dell'esempio 3.19 si ha $\mathrm{rg}(F) = 8$ e quindi $2^{\mathrm{rg}(F)} = 256$, ma l'algoritmo termina in 6 passi.

ESERCIZIO 3.34. Calcolate $R(s)$ per i vari passi dell'esecuzione dell'algoritmo 3.18 nell'esempio 3.19.

Possiamo completare il nostro lavoro sulle trasformazioni in forma normale congiuntiva e disgiuntiva con la:

DIMOSTRAZIONE DEL TEOREMA 3.10. Data una formula F applichiamo a partire da F gli algoritmi 3.18 e 3.22. Per il lemma 3.32 essi terminano, producendo due formule G_1 e G_2, la prima in forma normale congiuntiva, la seconda in forma normale disgiuntiva. Queste formule sono logicamente equivalenti a F. □

CAPITOLO 4

Il metodo dei tableaux: caso proposizionale

Il metodo dei tableaux proposizionali è un calcolo logico che fornisce una procedura di decisione relativamente efficiente per la soddisfacibilità delle formule proposizionali. Il principio che ispira questo algoritmo è piuttosto semplice: cerchiamo sistematicamente un'interpretazione che soddisfi la formula in esame. Se la ricerca ha successo la formula sarà soddisfacibile, altrimenti la sistematicità della ricerca ci assicura che essa è insoddisfacibile. Come gli algoritmi del capitolo 3, l'algoritmo dei tableaux è non deterministico e possiede la proprietà della terminazione forte.

1. Esempi preliminari

Partiamo da una formula F e, supponendo l'esistenza di un'interpretazione v che la soddisfi, esaminiamo quali altre proprietà deve avere v: ad esempio se F è la congiunzione $G \wedge H$ dovremo avere $v(G) = \mathbb{V}$ e $v(H) = \mathbb{V}$. A questo punto esaminiamo quali conseguenze hanno queste prime conseguenze e così via, passando ad esaminare formule via via più semplici fino a raggiungere i letterali (definizione 3.1). Riconosceremo la non esistenza di un'interpretazione con le caratteristiche richieste se ci troveremo a richiedere che gli elementi di una coppia complementare di letterali siano entrambi veri.

Prima di descrivere l'algoritmo dei tableaux esaminiamo in dettaglio un paio di esempi.

Esempio 4.1. Sia F la formula $(p \vee \neg q) \wedge \neg p$. Se v soddisfa F deve essere $v(p \vee \neg q) = \mathbb{V}$ e $v(\neg p) = \mathbb{V}$. Dalla prima di queste proprietà possiamo dedurre che $v(p) = \mathbb{V}$ oppure $v(\neg q) = \mathbb{V}$. La prima possibilità ci conduce a dover soddisfare la coppia complementare di letterali $\{p, \neg p\}$, che è impossibile. Resta la seconda possibilità, che ci chiede di soddisfare l'insieme di letterali $\{\neg q, \neg p\}$ che non contiene nessuna coppia complementare: ciò è possibile per il lemma 3.3. Dall'insieme soddisfacibile di letterali possiamo "leggere" un'interpretazione che soddisfa F: $v(p) = \mathbb{F}$ e $v(q) = \mathbb{F}$.

Esempio 4.2. Sia F la formula $(p \rightarrow \neg q) \wedge (p \wedge q)$. Se v soddisfa F deve essere $v(p \rightarrow \neg q) = \mathbb{V}$ e $v(p \wedge q) = \mathbb{V}$. La seconda condizione implica $v(p) = \mathbb{V}$ e $v(q) = \mathbb{V}$. La prima condizione è soddisfatta se e solo se $v(\neg p) = \mathbb{V}$ oppure $v(\neg q) = \mathbb{V}$. Nel primo caso dovremmo soddisfare l'insieme $\{\neg p, p, q\}$, nel secondo caso l'insieme $\{\neg q, p, q\}$. Entrambi questi insiemi contengono una coppia complementare e sono quindi insoddisfacibili per il lemma 3.3. Perciò F è insoddisfacibile.

Se abbiamo a che fare con formule più complesse il filo del ragionamento svolto nei due esempi precedenti può diventare piuttosto difficile da seguire. Il metodo dei tableaux rappresenta questi ragionamenti in una forma più facilmente leggibile attraverso una struttura ad albero.

La radice dell'albero è etichettata con la formula che si intende studiare, gli altri nodi con insiemi di formule. Inoltre una foglia (o nodo terminale) dell'albero è etichettata con un insieme di letterali.

Esempio 4.3. Disegniamo il tableau relativo alla formula studiata nell'esempio 4.1:

$$
\begin{tikzcd}[column sep=tiny]
& (p \lor \neg q) \land \neg p \arrow[d, dash] & \\
& p \lor \neg q, \neg p \arrow[dl, dash] \arrow[dr, dash] & \\
p, \neg p & & \neg q, \neg p
\end{tikzcd}
$$

e quello relativo alla formula studiata nell'esempio 4.2:

$$
\begin{tikzcd}[column sep=tiny]
& (p \to \neg q) \land (p \land q) \arrow[d, dash] & \\
& p \to \neg q, p \land q \arrow[d, dash] & \\
& p \to \neg q, p, q \arrow[dl, dash] \arrow[dr, dash] & \\
\neg p, p, q & & \neg q, p, q
\end{tikzcd}
$$

Il primo tableau contiene una foglia (quella di destra) etichettata con un insieme di letterali che non contiene coppie complementari: questo testimonia la soddisfacibilità della formula originaria. Il secondo tableau contiene solo foglie etichettate con insiemi di letterali che contengono coppie complementari: ciò implica che la formula di partenza è insoddisfacibile.

Esempio 4.4. La costruzione del tableau non è in generale unica. Ecco un altro tableau per la formula dell'esempio 4.2:

$$
\begin{tikzcd}[column sep=tiny]
& (p \to \neg q) \land (p \land q) \arrow[d, dash] & \\
& p \to \neg q, p \land q \arrow[dl, dash] \arrow[dr, dash] & \\
\neg p, p \land q \arrow[d, dash] & & \neg q, p \land q \arrow[d, dash] \\
\neg p, p, q & & \neg q, p, q
\end{tikzcd}
$$

Il nuovo tableau corrisponde ad un inversione nell'ordine del ragionamento: si è prima considerato il significato della soddisfazione di $p \to \neg q$ e solo successivamente, per ognuna delle due possibilità, il significato di $p \land q$. Notiamo che anche in questo tableau tutte le foglie sono etichettate con insiemi di letterali che contengono coppie complementari.

Gli alberi costruiti sinora evidenziano una caratteristica dei tableaux: alcuni nodi hanno un solo figlio mentre altri ne hanno due. I primi corrispondono all'analisi di formule come $p \land q$, i secondi all'analisi di $p \lor \neg q$ o $p \to \neg q$. Per distinguere questi casi ricorreremo alla distinzione tra doppie negazioni, α-formule e β-formule utilizzata già nel capitolo precedente (si ricordino in particolare le definizioni 3.12 e 3.13 e i lemmi 3.14 e 3.15).

2. L'algoritmo

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 $\mathcal{T}_0, \dots, \mathcal{T}_k$: 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 a $\mathcal{T}_i$. L'albero $\mathcal{T}_k$ (il numero k ovviamente dipende da F) è il tableau per F. Se n è un nodo di qualche $\mathcal{T}_i$ indichiamo con $E(n)$ l'etichetta di n (che, per quanto detto prima, è la stessa per tutti i $\mathcal{T}_i$ cui appartiene n).

All'inizio della costruzione $\mathcal{T}_0$ consiste di un solo nodo (la *radice* dell'albero) etichettato con $\{F\}$. Allo stadio i cerchiamo una foglia n dell'albero $\mathcal{T}_i$ 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. Altrimenti fissiamo n e scegliamo una formula $G \in E(n)$ che non è un letterale. Per il lemma 3.15 ci sono tre possibilità:

(1) se G è una doppia negazione con ridotto G_1 aggiungiamo un nodo n' sotto n e poniamo $E(n') = (E(n) \setminus \{G\}) \cup \{G_1\}$;
(2) se G è una α-formula con ridotti G_1 e G_2 aggiungiamo un nodo n' sotto n e poniamo $E(n') = (E(n) \setminus \{G\}) \cup \{G_1, G_2\}$;
(3) se G è una β-formula con ridotti G_1 e G_2 aggiungiamo due nodi tra loro inconfrontabili n_1 e n_2 sotto n e poniamo $E(n_1) = (E(n) \setminus \{G\}) \cup \{G_1\}$ e $E(n_2) = (E(n) \setminus \{G\}) \cup \{G_2\}$.

Notiamo che in ogni caso n non è una foglia di $\mathcal{T}_{i+1}$.

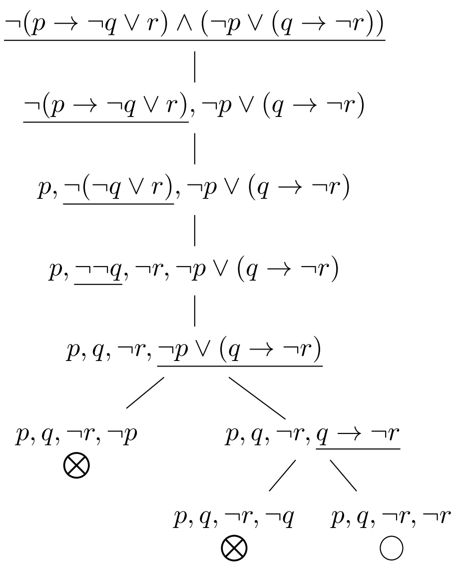
L'algoritmo che abbiamo appena descritto è non deterministico perché ad ogni passo scegliamo una foglia n non etichettata solo con letterali e, all'interno di $E(n)$, scegliamo una formula che non sia un letterale.

In pratica gli alberi $\mathcal{T}_0, \dots \mathcal{T}_k$ non sono rappresentati separatamente: si deve piuttosto pensare che il tableau "cresce" verso la sua forma finale.

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 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 $E(n)$.

Esempio 4.8. Costruiamo un tableau per $\neg(p \to \neg q \vee r) \wedge (\neg p \vee (q \to \neg r))$.



In ogni nodo abbiamo sottolineato la formula su cui abbiamo agito in quel nodo.

Notate che l'interpretazione definita da $v(p) = \mathbb{V}$, $v(q) = \mathbb{V}$, $v(r) = \mathbb{F}$ soddisfa sia i letterali che etichettano l'unica foglia marcata con $\bigcirc$ che la formula da cui siamo partiti.

Esercizio 4.9. Costruite tableaux per $\neg\big((q \to \neg p) \land (r \lor q) \to \neg(p \land \neg r)\big)$ e per $\neg(p \land q \to \neg r) \lor (p \land \neg(q \land r))$. Controllate anche, con le tavole di verità o ragionando dalle definizioni, se queste due formule sono soddisfacibili e, se è il caso, trovate un'interpretazione che le soddisfi.

3. Terminazione forte dei tableaux

Il primo risultato che vogliamo dimostrare relativamente al metodo dei tableaux è la sua terminazione forte. La dimostrazione si basa sul seguente lemma, che è un caso particolare del lemma di König[1]. Ricordiamo che un albero binario è un albero tale che ogni nodo ha al più due figli. Gli alberi costruiti dal metodo dei tableaux sono sempre binari.

Lemma 4.10. *Se un albero binario è infinito allora ha un ramo infinito.*

Dimostrazione. Sia $\mathcal{T}$ un albero binario e supponiamo che $\mathcal{T}$ sia infinito. Vogliamo dimostrare che $\mathcal{T}$ ha un ramo infinito. Diciamo che un nodo n di $\mathcal{T}$ è buono se il sottoalbero radicato in n è infinito[2].

Per ipotesi la radice di $\mathcal{T}$ è buona (il sottoalbero radicato in essa è $\mathcal{T}$). Sia n un nodo buono: il sottoalbero radicato in n consiste di n e dell'unione dei sottoalberi radicati nei figli di n. Dato che n ha al più due figli, se entrambi questi ultimi fossero finiti (o addirittura n non avesse figli) il sottoalbero radicato in n sarebbe finito, contro la bontà di n. Perciò almeno uno dei figli di n deve essere buono.

Nel paragrafo precedente abbiamo dimostrato che la radice di $\mathcal{T}$ è buona e che ogni nodo buono ha un figlio buono. Usando questi due fatti è facile costruire un ramo in $\mathcal{T}$ che consiste di nodi buoni ed è infinito. $\square$

Notiamo che l'inverso del lemma 4.10 (ovvero l'affermazione che ogni albero con un ramo infinito è infinito) è ovvio.

Possiamo ora dimostrare il teorema che ci interessa.

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 ad ogni nodo n di $\mathcal{T}$ un numero naturale $W(n)$ in modo tale che se n' è un figlio di n si abbia $W(n') < W(n)$. Se $\mathcal{T}$ avesse un ramo infinito i valori di W lungo questo ramo sarebbero una successione infinita decrescente di numeri naturali, che è assurdo.

Per definire W useremo il rango rg (definizione 3.31): se n è un nodo di $\mathcal{T}$, $W(n)$ è la somma dei ranghi delle formule in $E(n)$:

$$W(n) = \sum_{G \in E(n)} \mathrm{rg}(G).$$

---

[1]Dénes Kőnig (1884-1944) è stato un matematico ungherese, autore del primo libro sulla teoria dei grafi.

[2]il sottoalbero radicato in n è l'albero che ha n come radice e consiste di tutti i nodi che sono discendenti di n.

Sia ora n' un figlio di n. Dobbiamo verificare che si ha sempre $W(n') < W(n)$, cioè $W(n') \leq W(n) - 1$.

Se n' è stato ottenuto agendo su una doppia negazione G con ridotto G_1 allora

$$E(n') = (E(n) \setminus \{G\}) \cup \{G_1\}$$

e quindi

$$W(n') = W(n) - \mathrm{rg}(G) + \mathrm{rg}(G_1) = W(n) - (\mathrm{rg}(G_1) + 1) + \mathrm{rg}(G_1) = W(n) - 1.$$

Se n' è stato ottenuto agendo su una α-formula $G \in E(n)$, con ridotti G_1 e G_2 allora

$$E(n') = (E(n) \setminus \{G\}) \cup \{G_1, G_2\}$$

e quindi

$$\begin{aligned} W(n') &= W(n) - \mathrm{rg}(G) + \mathrm{rg}(G_1) + \mathrm{rg}(G_2) = \\ &= W(n) - (\mathrm{rg}(G_1) + \mathrm{rg}(G_2) + 1) + \mathrm{rg}(G_1) + \mathrm{rg}(G_2) = W(n) - 1. \end{aligned}$$

Se invece n' è stato ottenuto agendo su una β-formula $G \in E(n)$, con ridotti G_1 e G_2 allora

$$\begin{aligned} &E(n') = (E(n) \setminus \{G\}) \cup \{G_1\} \quad \text{oppure} \\ &E(n') = (E(n) \setminus \{G\}) \cup \{G_2\}. \end{aligned}$$

Nel primo caso abbiamo

$$\begin{aligned} &W(n') = W(n) - \mathrm{rg}(G) + \mathrm{rg}(G_1) = \\ &= W(n) - (\mathrm{rg}(G_1) + \mathrm{rg}(G_2) + 1) + \mathrm{rg}(G_1) = W(n) - \mathrm{rg}(G_2) - 1 \leq W(n) - 1. \end{aligned}$$

Il secondo caso è del tutto analogo.

La dimostrazione del teorema è così completa. □

Esercizio 4.12. Costruite un tableau per la formula $\neg(p \to q) \to (\neg p \lor q)$ e per ogni nodo n calcolate $W(n)$.

La dimostrazione del teorema 4.11 fornisce un limite superiore all'altezza di un tableau completo: se F è la formula che etichetta la radice l'altezza è minore o uguale a $\mathrm{rg}(F)$.

Esercizio 4.13. Sia F la formula dell'esempio 4.8. Calcolate $\mathrm{rg}(F)$ e confrontatelo con l'altezza del tableau per F.

Nota 4.14. Come notato per gli algoritmi di Fitting nella nota 3.33, anche l'algoritmo di costruzione dei tableaux **non** prevede la sostituzione di una formula con un'altra logicamente equivalente ad essa. Se introducessimo questa possibilità l'algoritmo non godrebbe della proprietà della terminazione forte e sarebbe assai meno utile.

4. Correttezza e completezza del metodo dei tableaux

Dobbiamo ora dimostrare che il metodo dei tableaux è effettivamente una procedura di decisione per la soddisfacibilità delle formule proposizionali. Iniziamo con alcune definizioni che ci permettono di leggere l'output del nostro algoritmo.

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.

Il teorema principale che dimostreremo in questa sezione è il seguente.

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.*

DIMOSTRAZIONE. Immediata dal teorema 4.16. □

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 in un tableau per F tutte le foglie sono etichettate con insiemi di letterali che non contengono coppie complementari. Ciò non è vero come mostra il tableau per la formula non valida $p \vee q$, che ha due foglie etichettate con un solo letterale e quindi prive di coppie complementari.

I teoremi precedenti ci permettono anche di concludere che la chiusura o apertura del tableau dipende solo dalla formula iniziale, e non dalle scelte fatte durante la costruzione del tableau stesso.

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 direzioni di questo teorema. Quella da sinistra a destra è chiamata teorema di completezza: ci dice che se F è insoddisfacibile l'algoritmo dei tableaux riesce a scoprirlo, chiudendo tutti i rami di qualsiasi tableau per F. In altre parole, se l'algoritmo non riesce ad ottenere un tableau chiuso per F significa che non è proprio possibile farlo perché F è soddisfacibile. La direzione opposta è invece chiamata teorema di correttezza e asserisce che se il metodo dei tableaux dichiara una formula insoddisfacibile perché il tableau per essa è chiuso, la formula è veramente insoddisfacibile.

La distinzione tra completezza e correttezza è utile per tutte le procedure di decisione, anche in logiche diverse (e più complesse) di quella proposizionale. Più avanti dimostreremo o enunceremo teoremi di correttezza e completezza anche per la deduzione naturale proposizionale (sezione 5.4), il metodo dei tableaux predicativi (sezioni 11.4 e 11.6), la deduzione naturale predicativa (sezione 12.3) e la deduzione naturale con uguaglianza (sezione 12.5).

Usualmente è molto più facile dimostrare la correttezza di una procedura di decisione che provarne la completezza. Infatti è probabile che una procedura di decisione consista di passaggi chiaramente corretti, così che la correttezza sia facile da verificare. Più delicato è dimostrare che i passaggi ammessi sono sufficienti a garantire la completezza, cioè che non si è "dimenticato nulla". Il seguente esempio, pur paradossale, può essere utile a chiarire questo punto. Consideriamo la procedura di decisione che dichiara ogni formula soddisfacibile: essa è corretta perché non sbaglia mai nel dichiarare una formula insoddisfacibile (non lo fa mai!), ma è ovviamente ben lontana dall'essere completa.

Terremo dunque distinte le due direzioni del teorema 4.16, e le enunceremo e dimostreremo separatamente, iniziando dalla più semplice.

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 $E(n) = \{F\}$) mostra che F è insoddisfacibile.

La dimostrazione di $(\star)$ è per induzione sull'altezza di n in $\mathcal{T}$[3].

Se l'altezza di n in $\mathcal{T}$ è 0 significa che n è una foglia del tableau. Dato che $\mathcal{T}$ è chiuso $E(n)$ è un insieme di letterali che contiene una coppia complementare. Per il lemma 3.3 $E(n)$ è insoddisfacibile.

Consideriamo ora il caso in cui l'altezza di n in $\mathcal{T}$ è maggiore di 0. Allora n ha uno o due successori in $\mathcal{T}$, che sono stati ottenuti agendo su qualche $G \in E(n)$ e ci sono tre possibilità.

(1) Se G è una doppia negazione con ridotto G_1, n ha un solo successore n' e si ha $E(n') = (E(n) \setminus \{G\}) \cup \{G_1\}$. Il nodo n' ha altezza minore di n in $\mathcal{T}$. Per ipotesi induttiva (cioè applicando $(\star)$ a n') $E(n')$ è insoddisfacibile. Sia ora v un'interpretazione qualsiasi: deve essere $v(H) = \mathbb{F}$ per qualche formula $H \in E(n')$ (che dipende in generale da v).
  (i) se $H \in E(n) \setminus \{G\}$ allora $H \in E(n)$ e quindi v non soddisfa $E(n)$;
  (ii) se H è G_1 allora $v(H) = \mathbb{F}$ implica $v(G) = \mathbb{F}$ (ricordate che $G \equiv G_1$ per il lemma 3.14) ed anche in questo caso v non soddisfa $E(n)$.
  Dato che un'interpretazione arbitraria non soddisfa $E(n)$ concludiamo che $E(n)$ è insoddisfacibile.

(2) se G è una α-formula con ridotti G_1 e G_2, n ha un solo successore n' e si ha $E(n') = (E(n) \setminus \{G\}) \cup \{G_1, G_2\}$. Come nel caso precedente deduciamo che $E(n')$ è insoddisfacibile: fissato v si ha $v(H) = \mathbb{F}$ per qualche $H \in E(n')$.
  (i) se $H \in E(n) \setminus \{G\}$ allora v non soddisfa $E(n)$;
  (ii) se H è G_1 oppure G_2 allora da $v(H) = \mathbb{F}$ segue $v(G_1 \wedge G_2) = \mathbb{F}$ e quindi $v(G) = \mathbb{F}$ perché $G \equiv G_1 \wedge G_2$ per il lemma 3.14. Dunque v non soddisfa $E(n)$.
  Abbiamo dunque dimostrato che $E(n)$ è insoddisfacibile.

(3) se G è una β-formula con ridotti G_1 e G_2, n ha due successori n_1 e n_2 e si ha $E(n_i) = (E(n) \setminus \{G\}) \cup \{G_i\}$ (per $i = 1, 2$). Il ragionamento del punto (1) può venir ripetuto per ognuno dei due successori ottenendo che sia $E(n_1)$ che $E(n_2)$ sono insoddisfacibili. Fissiamo nuovamente un'interpretazione v:
  (i) se $v(G) = \mathbb{F}$ allora chiaramente v non soddisfa $E(n)$;
  (ii) se $v(G) = \mathbb{V}$ allora, dato che $G \equiv G_1 \vee G_2$ per il lemma 3.14, $v(G_1) = \mathbb{V}$ oppure $v(G_2) = \mathbb{V}$. Supponiamo dapprima che $v(G_1) = \mathbb{V}$. Visto che $E(n_1)$ è insoddisfacibile, v non soddisfa $E(n_1)$ e deve essere $v(H) = \mathbb{F}$ per qualche $H \in E(n) \setminus \{G\}$. Quindi v non soddisfa $E(n)$ neppure in questo caso. Se $v(G_2) = \mathbb{V}$ il ragionamento è analogo utilizzando l'insoddisfacibilità di $E(n_2)$.
  Perciò $E(n)$ è insoddisfacibile.

[3] l'altezza di una foglia è 0, l'altezza di un nodo con successori è il massimo dell'altezza dei suoi successori aumentato di 1.

Abbiamo dunque dimostrato ($\star$) e quindi il teorema di correttezza. $\square$

Sottolineiamo che la dimostrazione precedente non procede per induzione sull'altezza del tableau. Infatti il tableau $\mathcal{T}$ è fissato e l'induzione riguarda i suoi nodi.

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);
(b) dimostreremo che ogni insieme di Hintikka è soddisfacibile (lemma 4.27);
(c) dimostreremo che $\bigcup_{n \in r} E(n)$ è un insieme di Hintikka (lemma 4.28).

Dato che $F \in \bigcup_{n \in r} E(n)$ (perché la radice del tableau appartiene a r) questi passi sono sufficienti a completare la dimostrazione: infatti qualunque interpretazione che soddisfa l'insieme di Hintikka $\bigcup_{n \in r} E(n)$ soddisfa in particolare F. $\square$

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 $\mathcal{H}$).

La definizione 4.23 è basata sull'idea che un insieme di Hintikka $\mathcal{H}$ consiste di formule che supponiamo essere vere in una qualche interpretazione v. La verità di ogni $F \in \mathcal{H}$ che non è un letterale deve essere "giustificata" dalla verità di altre formule (tutti o alcuni dei ridotti di F, a seconda del tipo di formula) di $\mathcal{H}$.

Esempio 4.24. L'insieme

$$\{p \lor \neg r \to q, \neg(p \lor \neg r), \neg p, \neg\neg r, r\}$$

è un insieme di Hintikka. Infatti non contiene coppie complementari di letterali, contiene uno dei ridotti della β-formula $p \lor \neg r \to q$, entrambi i ridotti della α-formula $\neg(p \lor \neg r)$ e il ridotto della doppia negazione $\neg\neg r$.

Esercizio 4.25. Siano F_1 e F_2 le formule $\neg(p \to q \lor \neg r)$ e $\neg p \lor (r \to q)$.

(i) Definite insiemi di Hintikka $\mathcal{H}_1$ e $\mathcal{H}_2$ con $F_i \in \mathcal{H}_i$.
(ii) $\mathcal{H}_1 \cup \mathcal{H}_2$ è un insieme di Hintikka?
(iii) ($\star$) Esiste un insieme di Hintikka che contiene sia F_1 che F_2?

Esercizio 4.26. Verificare che l'insieme di formule che compaiono nelle etichette del ramo aperto del tableau dell'esempio 4.8 è un insieme di Hintikka.

Il prossimo lemma estende una direzione del lemma 3.3. Infatti un insieme di letterali privo di coppie complementari è un insieme di Hintikka.

Lemma 4.27 (Lemma di Hintikka). *Ogni insieme di Hintikka è soddisfacibile.*

Dimostrazione. Sia $\mathcal{H}$ un insieme di Hintikka. Definiamo un'interpretazione v esattamente come nella dimostrazione del lemma 3.3, e cioè ponendo:

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

[4] Jaakko Hintikka (1929-2015) è stato un logico e filosofo finlandese.

Dimostreremo che per ogni $G \in \mathcal{H}$ si ha $v(G) = \mathbb{V}$: questo implica che $\mathcal{H}$ è soddisfacibile. La dimostrazione è per induzione sul rango di G (definizione 3.31). Se $\mathrm{rg}(G) = 1$ allora G è un letterale:

- se $G \in \mathcal{H}$ è una lettera proposizionale allora $v(G) = \mathbb{V}$ per definizione di v.
- se $G \in \mathcal{H}$ è la negazione di una lettera proposizionale p allora $p \notin \mathcal{H}$ per (1) nella definizione di insieme di Hintikka. Quindi $v(p) = \mathbb{F}$ e perciò $v(G) = \mathbb{V}$.

Se $\mathrm{rg}(G) > 1$ allora G è una doppia negazione, una α-formula o una β-formula:

- Se $G \in \mathcal{H}$ è una doppia negazione con ridotto G_1 allora $G_1 \in \mathcal{H}$ per (2) nella definizione di insieme di Hintikka. Poiché $\mathrm{rg}(G_1) = \mathrm{rg}(G) - 1$ l'ipotesi induttiva ci dice che $v(G_1) = \mathbb{V}$. Dato che per il lemma 3.14 $G \equiv G_1$ abbiamo $v(G) = \mathbb{V}$.
- Se $G \in \mathcal{H}$ è una α-formula con ridotti G_1 e G_2 allora $G_1, G_2 \in \mathcal{H}$ per (3) nella definizione di insieme di Hintikka. Per ipotesi induttiva, dato che $\mathrm{rg}(G_1) < \mathrm{rg}(G)$ e $\mathrm{rg}(G_2) < \mathrm{rg}(G)$, possiamo assumere $v(G_1) = \mathbb{V}$ e $v(G_2) = \mathbb{V}$. Poiché per il lemma 3.14 $G \equiv G_1 \wedge G_2$ abbiamo $v(G) = \mathbb{V}$.
- Se $G \in \mathcal{H}$ è una β-formula con ridotti G_1 e G_2 allora $G_1 \in \mathcal{H}$ oppure $G_2 \in \mathcal{H}$ per (4) nella definizione di insieme di Hintikka. Se $G_1 \in \mathcal{H}$ per ipotesi induttiva, dato che $\mathrm{rg}(G_1) < \mathrm{rg}(G)$, $v(G_1) = \mathbb{V}$ e, dato che per il lemma 3.14 $G \equiv G_1 \vee G_2$, abbiamo $v(G) = \mathbb{V}$. Il caso in cui $G_2 \in \mathcal{H}$ è del tutto analogo. □

Per qualunque insieme di formule $\mathcal{H}$ è possibile definire v come abbiamo fatto nella dimostrazione precedente, ma per dimostrare che v soddisfa $\mathcal{H}$ abbiamo usato ripetutamente l'ipotesi che $\mathcal{H}$ fosse insieme di Hintikka.

Lemma 4.28. *Se r è un ramo aperto di un tableau allora $\mathcal{H} = \bigcup_{n \in r} E(n)$ è un insieme di Hintikka.*

Dimostrazione. Sia f la foglia di r. Dobbiamo verificare che $\mathcal{H}$ soddisfa le quattro proprietà della definizione di insieme di Hintikka.

(1) Per verificare che $\mathcal{H}$ non contiene una coppia complementare di letterali notiamo che l'algoritmo dei tableaux non elimina mai i letterali dalle etichette dei nodi. Perciò se un letterale G appartene a qualche $E(n)$ con $n \in r$ allora $G \in E(f)$. Questo significa che se $\mathcal{H}$ contiene due letterali complementari G e H si ha $G, H \in E(f)$ e r non è aperto, contro la nostra ipotesi.

(2) Se una doppia negazione G con ridotto G_1 appartiene a $\mathcal{H}$ allora sia n il nodo di r più vicino a f tra quelli tali che $G \in E(n)$. Non può essere $n = f$, perché $E(f)$ è un insieme di letterali, e quindi n ha un successore $n' \in r$. Dato che $G \notin E(n')$ deve essere che G è la formula su cui si agisce in n. Perciò $G_1 \in E(n')$ e quindi $G_1 \in \mathcal{H}$ come volevamo.

(3) Se una α-formula G con ridotti G_1 e G_2 appartiene a $\mathcal{H}$ allora si può ripetere il ragionamento svolto nel caso della doppia negazione per arrivare a concludere che $G_1, G_2 \in \mathcal{H}$.

(4) se una β-formula G con ridotti G_1 e G_2 appartiene a $\mathcal{H}$ allora come sopra sia $n \neq f$ il nodo di r più vicino a f tra quelli tali che $G \in E(n)$. Anche in questo caso deve essere che G è la formula su cui si agisce in n e quindi n ha due successori. Esattamente uno di questi successori appartiene a r: indichiamolo con n'. Si ha $G_1 \in E(n')$ oppure $G_2 \in E(n')$ e quindi $G_1 \in \mathcal{H}$ oppure $G_2 \in \mathcal{H}$, come volevamo. □

La dimostrazione del teorema di completezza fornisce informazioni su come trovare un'interpretazione che soddisfi una formula il cui tableau è aperto.

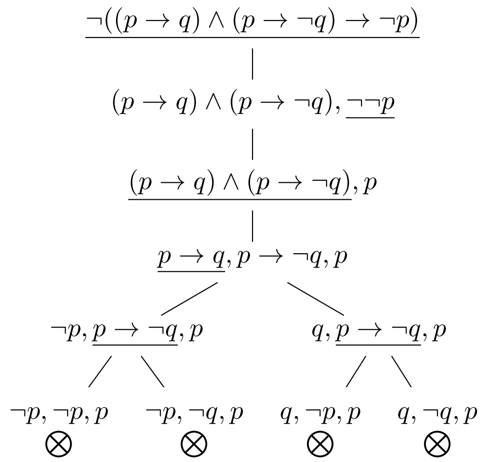
Lemma 4.29. *Se una foglia f di un tableau per F non contiene coppie complementari allora l'interpretazione v definita da*

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

soddisfa F.

Un esempio di applicazione di questo lemma è contenuto nell'esempio 4.8.

Esempio 4.30. Utilizziamo il metodo dei tableaux per dimostrare la validità della formula $(p \to q) \wedge (p \to \neg q) \to \neg p$. Per il teorema 4.18 dobbiamo verificare che un tableau per la negazione di questa formula è chiuso. In ogni nodo sottolineiamo la formula su cui agiamo in quel nodo.



5. Semplificare i tableaux

Per scrivere più concisamente i tableaux è utile introdurre alcune abbreviazioni e piccole varianti dell'algoritmo 4.5.

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).

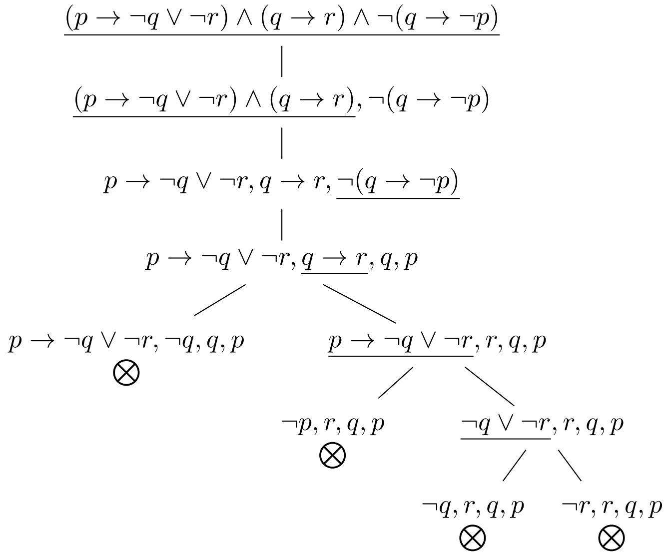
Nel tableau dell'esempio 4.30 la precedente convenzione avrebbe permesso di non scrivere le due foglie più a sinistra, chiudendo il ramo di sinistra dopo il nodo etichettato con $\neg p, p \to \neg q, p$.

Nota 4.33. La convenzione 4.31 è solo formale (si tratta solamente di un modo per scrivere più rapidamente i tableaux). La convenzione 4.32 riguarda invece un mutamento sostanziale ed è necessario verificare che i teoremi di correttezza e completezza valgono anche per questi tipi di tableaux. La completezza non presenta nessun problema, perché la dimostrazione assume l'esistenza di rami aperti, mentre la convenzione indica un modo per chiudere rami. Risulta invece necessario dimostrare nuovamente il teorema di correttezza: la dimostrazione è però una semplice modifica di quella precedente, utilizzando la seconda parte della nota 3.4.

Esempio 4.34. Utilizziamo le convenzioni 4.31 e 4.32 per costruire un tableau per la formula

$$(p \to \neg q \vee \neg r) \wedge (q \to r) \wedge \neg(q \to \neg p)$$

In ogni nodo sottolineiamo la formula su cui agiamo in quel nodo.



La convenzione 4.31 è stata applicata nel nodo ottenuto agendo su $\neg(q \to \neg p)$ (in cui sarebbe dovuto comparire $\neg\neg p$), la convenzione 4.32 nel nodo chiuso più in alto, in cui compare la formula $p \to \neg q \vee \neg r$ che non è un letterale.

Dato che tutti i rami sono chiusi il teorema di correttezza ci permette di concludere che la formula di partenza è insoddisfacibile.

Esempio 4.35. Nel seguente tableaux la semplificazione apportata dalla convenzione 4.32 è particolarmente evidente:

$$\begin{array}{c} (p \vee q \to q \wedge \neg r) \wedge \neg(p \vee q \to q \wedge \neg r) \\ | \\ p \vee q \to q \wedge \neg r, \neg(p \vee q \to q \wedge \neg r) \\ \otimes \end{array}$$

Esercizio 4.36. Sviluppate completamente un tableau per la formula

$$(p \vee q \to q \wedge \neg r) \wedge \neg(p \vee q \to q \wedge \neg r)$$

dell'esempio precedente utilizzando l'algoritmo 4.5 senza usare la convenzione 4.32.

Come già notato, l'algoritmo 4.5 è non deterministico: la seguente osservazione ci indica come cercare di ottenere il tableau più semplice possibile.

Nota 4.37. Per semplificare il più possibile il tableau costruito dall'algoritmo 4.5 è opportuno agire su doppie negazioni o α-formule ogniqualvolta ciò sia possibile. In questo modo si dilaziona il più possibile la creazione di biforcazioni nell'albero e si riduce al minimo il numero dei nodi.

Esercizio 4.38. Studiate con il metodo del tableaux le formule degli esempi 2.38, 2.39, 2.50 e 2.51 e dell'esercizio 2.55 (verificate che il risultato ottenuto con i tableaux coincida con quello ottenuto in precedenza).

6. I tableaux e la conseguenza logica

Abbiamo presentato il metodo dei tableaux come un metodo per studiare la validità o la soddisfacibilità di una singola formula. Usando il lemma 2.43 possiamo usare i tableaux per studiare la validità o la soddisfacibilità di un insieme finito di formule $\{F_1, \dots, F_n\}$. Nel caso della soddisfacibilità si dovrebbe costruire un tableau per $F_1 \wedge \dots \wedge F_n$. I primi $n-1$ passi del tableau conducono ad un nodo etichettato $F_1, \dots, F_n$: questa osservazione giustifica il seguente algoritmo.

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 ci conducono a costruire un tableau per $\neg(F_1 \wedge \dots \wedge F_n)$. Dopo $n-1$ passi del tableau otteniamo un albero con n foglie etichettate rispettivamente con $\neg F_1, \dots, \neg F_n$. Questo tableau è chiuso se e solo se sono chiusi gli n tableaux per $\neg F_1, \dots, \neg F_n$, e quindi in pratica dobbiamo studiare la validità di ogni F_i. Questo non è sorprendente, dato che, come osservato nella nota 2.44, $\{F_1, \dots, F_n\}$ è valido se e solo se ognuna delle formule $F_1, \dots, F_n$ è valida.

Il lemma 2.40 ci permette di usare i tableaux per stabilire se vale una conseguenza logica. In questo caso per stabilire che $F \models G$ bisogna costruire un tableau chiuso per $\neg(F \to G)$ (la negazione delle formula di cui si vuole stabilire la validità), che ha il primo nodo sotto la radice etichettato da $F, \neg G$. Generalizzando questa osservazione al caso in cui a sinistra del simbolo di conseguenza logica compare un insieme finito di formule (definizione 2.27) ed usando il lemma 2.30 si ottiene il seguente algoritmo.

Algoritmo 4.40. 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 $F_1, \dots, F_n \models G$, se il tableau è 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 come nel lemma 4.29).

Esercizio 4.41. Studiate con il metodo del tableaux le conseguenze logiche degli esempi 2.17, 2.52, 2.53 e 2.61 e degli esercizi 2.26, 2.54 e 2.56 (verificate che il risultato ottenuto con i tableaux coincida con quello ottenuto in precedenza). Per i problemi che riguardano un'equivalenza logica bisogna studiare le due conseguenze logiche, usando il lemma 2.19.

CAPITOLO 5

La deduzione naturale: caso proposizionale

La deduzione naturale è un calcolo logico introdotto da Gentzen nel 1935. Il suo scopo è quello di riprodurre in modo formale alcuni ragionamenti che siamo abituati ad effettuare nella nostra vita quotidiana per trarre conclusioni e fare previsioni a partire da ciò che sappiamo. Nell'attività scientifica, ed in modo particolarmente spiccato in quella matematica e informatica, questi ragionamenti sono sottoposti a criteri di rigore stringenti. In questo contesto i ragionamenti assumono la forma di *deduzioni* in cui si dimostra che un'affermazione è conseguenza di alcune ipotesi. La deduzione naturale si propone quindi come un metodo per ottenere conseguenze logiche. Essa segue quindi un approccio diverso da quello utilizzato dal metodo dei tableaux, che si concentra sulle dimostrazioni di insoddisfacibilità (nella sezione 4.6 abbiamo descritto come utilizzare i tableaux per studiare la conseguenza logica, ma ciò avviene riconducendo la conseguenza logica all'insoddisfacibilità).

Sottolineiamo da subito che la costruzione di deduzioni naturali, a differenza dell'utilizzo del metodo dei tableaux, non è un procedimento algoritmico ma richiede l'utilizzo di strategie che partono da un'analisi di ciò che si vuole dedurre e degli strumenti (le regole) a nostra disposizione. Facendo un paragone con l'analisi matematica, il metodo dei tableaux è analogo alle procedure di differenziazione di una funzione, mentre la deduzione naturale è simile all'integrazione. Per questa ragione dedicheremo un certo spazio (nella sezione 5.6) agli esempi di costruzione di deduzioni naturali, cercando di avviare il lettore a questa "arte".

1. Caratteristiche di un sistema deduttivo

La deduzione naturale è un *sistema deduttivo*. Un sistema deduttivo, visto astrattamente, è una relazione che può sussistere tra un insieme di formule e una formula: se la relazione vale diciamo che la formula viene dedotta dall'insieme. Per introdurre le regole della deduzione naturale partiamo dall'analisi di alcune caratteristiche che sono desiderabili in un sistema deduttivo arbitrario. Fissato dunque un sistema deduttivo, indichiamo con $T \vdash F$ il fatto che dall'insieme di formule T deduciamo la formula F[1].

La prima condizione che vorremmo veder soddisfatta da $\vdash$ è che $T \vdash F$ implichi $T \vDash F$: ciò che deduciamo deve essere vero. Questa proprietà è la *correttezza* del sistema deduttivo rappresentato da $\vdash$ ed è garantita se i singoli passi della nostra deduzione sono corretti. Verrà dimostrata per la deduzione naturale proposizionale nel teorema 5.17.

Idealmente desideriamo anche che valga l'implicazione inversa: $T \vDash F$ 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". Ci limiteremo ad enunciare questo risultato per la deduzione naturale proposizionale nel teorema 5.18.

[1] La scelta (tradizionale) del simbolo $\vdash$ ha lo scopo di richiamare, pur differenziandosene, il simbolo $\vDash$ usato per la conseguenza logica: $\vDash$ è un concetto semantico, mentre qui abbiamo a che fare con una nozione che utilizza le proprietà sintattiche delle formule coinvolte.

Una proprietà fondamentale che il sistema deduttivo dovrebbe possedere è quella della *componibilità delle deduzioni*. Spesso per dimostrare un teorema F iniziamo con il dimostrare un "lemma" intermedio G, per poi utilizzare G nella dimostrazione di F. Utilizzando il nostro simbolismo, esprimiamo questa proprietà come:

$$\texttt{se } T \vdash G \texttt{ e } T', G \vdash F \texttt{ allora } T, T' \vdash F.$$

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: lo verificheremo nel lemma 5.19.

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.

Iniziamo da $\wedge$. 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:

$$F \wedge G \vdash F \qquad (\wedge e.1)$$
$$F \wedge G \vdash G \qquad (\wedge e.2)$$
$$F, G \vdash F \wedge G \qquad (\wedge i)$$

Le lettere e e i sono state scelte perché nelle prime due righe stiamo *eliminando* il connettivo $\wedge$, mentre nell'ultima lo stiamo *introducendo*. La correttezza delle regole di eliminazione è garantita dal lemma 2.20.7, quella della regola di introduzione dall'esercizio 2.31.a.

Nel caso di $\vee$ è evidente (sulla base del lemma 2.20.6) come possiamo introdurre il connettivo:

$$F \vdash F \vee G \qquad (\vee i.1)$$
$$G \vdash F \vee G \qquad (\vee i.2)$$

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.

Nel caso di $\rightarrow$ la regola di eliminazione è nota come *modus ponens*[2], e rappresenta il modo tipico di utilizzare un'implicazione:

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

Questa regola è giustificata dall'esercizio 2.31.b. 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.

Esempio 5.1. 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 quindi ottenuto F a partire da una delle nostre ipotesi. Per $(\rightarrow e)$ abbiamo $F, F \rightarrow G \vdash G$, e siamo quindi giunti a G partendo da una delle ipotesi e da una conclusione già ottenuta. Per $(\wedge e.2)$ abbiamo $F \wedge (G \rightarrow H) \vdash G \rightarrow H$ e da $(\rightarrow e)$ otteniamo $G, G \rightarrow H \vdash H$.

[2]modus ponens abbrevia la locuzione latina *modus ponendo ponens*, che letteralmente significa "modo che pone con l'aver posto", ovvero "modo che afferma".

Abbiamo quindi ottenuto sia G che H, e possiamo usare $(\wedge i)$ per ottenere $G \wedge H$.

Il ragionamento dell'esempio precedente non è facilissimo da seguire quando viene scritto per esteso come sopra, ed è possibile che a qualche lettore sia venuto naturale prendere carta e penna per "disegnarlo". La deduzione naturale fornisce proprio un modo di "disegnare" i ragionamenti e rappresentarli in maniera chiara. Nell'esempio 5.5 "disegneremo" il ragionamento dell'esempio 5.1.

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.

Definizione 5.2. La formula $\perp$ 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 $v(\perp) = \mathbb{F}$.

Nota 5.3. La definizione precedente 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. L'esercizio 2.35 si applica quindi solo alle lettere proposizionali diverse da $\perp$.

Esercizio 5.4. Dimostrare che per qualunque formula F e insieme di formule T:

(1) $\perp \models F$ e $F \models \neg\perp$;
(2) $\perp \equiv F \wedge \neg F$;
(3) $\neg\perp \equiv F \vee \neg F$;
(4) $F \wedge \perp \equiv \perp$;
(5) $F \wedge \neg\perp \equiv F$;
(6) $F \vee \perp \equiv F$;
(7) $F \vee \neg\perp \equiv \neg\perp$
(8) $F \to \perp \equiv \neg F$;
(9) $F \to \neg\perp \equiv \neg\perp$;
(10) $\perp \to F \equiv \neg\perp$;
(11) $\neg\perp \to F \equiv F$;
(12) $T \models \perp$ se e solo se T è insoddisfacibile;
(13) $\neg\perp \models F$ se e solo se F è valida.

Riprendendo il discorso sulla negazione, otteniamo la regola:

$$F, \neg F \vdash \perp \qquad (\neg e)$$

Questa regola, la cui correttezza deriva dall'esercizio 5.4.2, può anche essere vista come la regola di introduzione di $\perp$, $(\perp i)$.

Un'ultima regola che riguarda la negazione è l'eliminazione della doppia negazione:

$$\neg\neg F \vdash F \qquad (\neg\neg e)$$

Essa è giustificata dalla prima equivalenza logica del lemma 2.20.

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 $\to$. Per dimostrare l'implicazione $F \to G$ 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 $F \to G$. La correttezza di questo principio è verificata osservando che se F è falsa, $F \to G$ è certamente vera, mentre se F è vera la nostra deduzione di partenza (che supponiamo corretta) ci assicura che anche G è vera e quindi anche $F \to G$ lo è. Utilizzando la nostra simbologia:

$$\texttt{se}\ F \vdash G\ \texttt{allora}\ \vdash F \to G \qquad (\to i)$$

dove la deduzione finale non ha alcuna ipotesi.

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

$$\texttt{se}\ F \vdash \bot\ \texttt{allora}\ \vdash \neg F \qquad (\neg i)$$

Questa regola può anche essere viste come l'eliminazione di $\bot$, $(\bot e)$.

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:

$$\texttt{se}\ F \vdash H\ \texttt{e}\ G \vdash H\ \texttt{allora}\ F \vee G \vdash H \qquad (\vee e)$$

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).

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

$$\texttt{se}\ \neg F \vdash \bot\ \texttt{allora}\ \vdash F \qquad (RAA)$$

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.

2. La deduzione naturale proposizionale

In ogni deduzione naturale le regole che abbiamo analizzato nella sezione precedente vengono combinate (usando la regola di taglio, ma non solo: si veda lo scaricamento delle ipotesi descritto più avanti) tra di loro in modo da ottenere un albero (a differenza del metodo dei tableaux, nel caso della deduzione naturale è più naturale che l'albero abbia la radice in basso). Ogni nodo dell'albero è etichettato con una singola formula (anziché con un insieme di formule come nel caso dei tableaux). La radice dell'albero è etichettata con il risultato della deduzione, mentre le foglie sono etichettate con le ipotesi della deduzione. Quando abbiamo a che fare con la deduzione naturale usiamo $\rhd$ come simbolo deduttivo (al posto di $\vdash$): perciò $T \rhd F$ significa 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 corrisponde ad aggiungere un nodo sotto uno o più nodi preesistenti. Così ad esempio $(\wedge i)$ è rappresentata dall'albero

$$\frac{F \qquad G}{F \wedge G}$$

(la linea orizzontale è il modo tradizionale di rappresentare una regola deduttiva). Similmente è facile rappresentare ($\wedge e$.1), ($\wedge e$.2), ($\vee i$.1), ($\vee i$.2), ($\to e$), ($\neg e$) e ($\neg\neg e$), rispettivamente come

$$\frac{F \wedge G}{F} \qquad \frac{F \wedge G}{G} \qquad \frac{F}{F \vee G} \qquad \frac{G}{F \vee G}$$

$$\frac{F \qquad F \to G}{G} \qquad \frac{F \qquad \neg F}{\bot} \qquad \frac{\neg\neg F}{F}$$

Più delicato è il caso delle regole "condizionali", cioè ($\vee e$), ($\to i$) e ($\neg i$), in cui è necessario rappresentare un cambiamento delle ipotesi della deduzione. Posticipiamo un poco la loro discussione, per permettere al lettore di familiarizzare con la costruzione di deduzioni naturali in cui non compaiono le regole condizionali.

Esempio 5.5. Utilizzando le regola introdotte sinora, rappresentiamo con un albero di deduzione naturale il ragionamento svolto nell'Esempio 5.1:

$$\dfrac{\dfrac{\dfrac{F \wedge (G \to H)}{F} \qquad F \to G}{G} \qquad \dfrac{\dfrac{\dfrac{F \wedge (G \to H)}{F} \qquad F \to G}{G} \qquad \dfrac{F \wedge (G \to H)}{G \to H}}{H}}{G \wedge H}$$

Abbiamo così ottenuto $F \wedge (G \to H), F \to G \rhd G \wedge H$.

Notiamo come in questa deduzione naturale ognuna delle nostre ipotesi compaia in più di una foglia. Ciò è perfettamente accettabile: se assumiamo $F \wedge (G \to H)$ possiamo sfruttare questa ipotesi ripetutamente nel corso della dimostrazione[3].

Esercizio 5.6. Scrivere accanto ad ogni linea orizzontale della deduzione naturale dell'esempio precedente la regola utilizzata.

Nota 5.7. Notiamo come la deduzione naturale dell'esempio 5.5 sia in realtà uno schema. Essa vale qualunque siano le formule F, G e H e quindi abbiamo mostrato ad esempio che $\neg p \wedge (q \to (r \to s)), \neg p \to q \rhd q \wedge (r \to s)$. 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.

Esercizio 5.8. Costruire una deduzione naturale che mostri che

$$F \wedge (F \to \neg\neg G) \rhd G.$$

Per trattare le regole condizionali, è opportuno considerare alcune delle formule coinvolte come delle "ipotesi ausiliarie". Ad esempio in ($\to i$), il nostro obiettivo è dedurre $F \to G$, 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 $F \to G$. Per indicare lo scaricamento di F metteremo questa formula tra parentesi quadre. La deduzione naturale che corrisponde a ($\to i$) è dunque

$$\dfrac{\begin{array}{c} T, [F] \\ \triangledown \\ G \end{array}}{F \to G}$$

[3] La logica lineare, la cui trattazione esula da questo corso, invece considera le ipotesi come risorse che vengono consumate e non possono essere riutilizzate.

In questa deduzione il simbolo ∇ indica l'esistenza di una deduzione naturale che mostra $T, F \rhd G$. Le etichette di foglie che sono messe tra [ ] non sono ipotesi della deduzione naturale. In questo modo abbiamo effettivamente ottenuto $T \rhd F \to G$.

Similmente ($\vee e$) e ($\neg i$) sono rappresentate dalle deduzioni naturali

$$\dfrac{T, F \vee G \qquad \begin{matrix} T', [F] \\ \nabla \\ H \end{matrix} \qquad \begin{matrix} T'', [G] \\ \nabla \\ H \end{matrix}}{H}$$

e

$$\begin{matrix} T, [F] \\ \nabla \\ \dfrac{\perp}{\neg F} \end{matrix}$$

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

Esempio 5.9. Vogliamo dimostrare in deduzione naturale l'inverso della regola della doppia negazione, cioè che $F \rhd \neg\neg F$. Consideriamo la deduzione naturale

$$\dfrac{\dfrac{F \qquad [\neg F]^1}{\perp}}{\neg\neg F}\,1$$

La prima riga della deduzione naturale è un'applicazione di ($\neg e$): da F e $\neg F$ abbiamo dedotto $\perp$. Successivamente abbiamo un'applicazione di ($\neg i$): dato che da $\neg F$ (più l'altra ipotesi F) abbiamo dedotto $\perp$, possiamo concludere $\neg\neg F$ eliminando 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: $F \rhd \neg\neg F$.

Esempio 5.10. Come primo esempio di applicazione della regola ($\vee e$) consideriamo la seguente deduzione naturale, che mostra che $\neg F \wedge \neg G \rhd \neg(F \vee G)$:

$$\dfrac{\dfrac{[F \vee G]^2 \qquad \dfrac{[F]^1 \qquad \dfrac{\neg F \wedge \neg G}{\neg F}}{\perp} \qquad \dfrac{[G]^1 \qquad \dfrac{\neg F \wedge \neg G}{\neg G}}{\perp}}{\perp}\,1}{\neg(F \vee G)}\,2$$

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 $\neg F \wedge \neg G, F \vee G \rhd \perp$). 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 $\neg F \wedge \neg G, F \rhd \perp$ e $\neg F \wedge \neg G, G \rhd \perp$). 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.

Esempio 5.11. Consideriamo ora la seguente deduzione naturale, che mostra che $\neg F \lor \neg G \rhd \neg(F \land G)$:

$$\dfrac{\dfrac{\neg F \lor \neg G \qquad \dfrac{[\neg F]^1 \qquad \dfrac{[F \land G]^2}{F}}{\bot} \qquad \dfrac{[\neg G]^1 \qquad \dfrac{[F \land G]^2}{G}}{\bot}}{\bot}\,{\scriptstyle 1}}{\neg(F \land G)}\,{\scriptstyle 2}$$

In questo caso consigliamo al lettore di percorrere autonomamente a parole il ragionamento rappresentato dalla deduzione naturale.

Osserviamo che nell'applicazione della regola $(\neg i)$ al termine della deduzione naturale, abbiamo scaricato due occorrenze di $F \land G$: questo è utile quando, come in questo caso, un'ipotesi ausiliaria viene utilizzata più di una volta.

Una seconda osservazione è che la deduzione naturale potrebbe essere modificata invertendo l'ordine di applicazione di $(\lor e)$ e di $(\neg i)$. Otterremmo così un'altra deduzione naturale per $\neg F \lor \neg G \rhd \neg(F \land G)$:

$$\dfrac{\neg F \lor \neg G \qquad \dfrac{\dfrac{[\neg F]^3 \qquad \dfrac{[F \land G]^1}{F}}{\bot}}{\neg(F \land G)}\,{\scriptstyle 1} \qquad \dfrac{\dfrac{[\neg G]^3 \qquad \dfrac{[F \land G]^2}{G}}{\bot}}{\neg(F \land G)}\,{\scriptstyle 2}}{\neg(F \land G)}\,{\scriptstyle 3}$$

Esempio 5.12. La regola di riduzione all'assurdo (RAA), che avevamo dedotto astrattamente nelle sezione precedente, è rappresentata da

$$\begin{array}{c} T, [\neg F] \\ \triangledown \\ \dfrac{\bot}{F} \end{array}$$

Una deduzione naturale che, utilizzando $(\neg i)$ e $(\neg\neg e)$, giustifica questa regola è

$$\begin{array}{c} T, [\neg F]^1 \\ \triangledown \\ \dfrac{\dfrac{\bot}{\neg\neg F}\,{\scriptstyle 1}}{F} \end{array}$$

Nella nostra presentazione della deduzione naturale (RAA) è quindi una *regola derivata*.

Esempio 5.13. La deduzione naturale

$$\dfrac{F \lor \neg G \qquad [F]^2 \qquad \dfrac{\dfrac{\dfrac{G \qquad [\neg G]^2}{\bot}}{\neg\neg F}\,{\scriptstyle 1}}{F}}{F}\,{\scriptstyle 2}$$

mostra $F \lor \neg G, G \rhd F$. Le applicazioni (contrassegnate dagli indici 1 e 2) delle regole $(\neg i)$ e $(\lor e)$ hanno caratteristiche che vale la pena di mettere in evidenza.

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.

Nell'uso di $(\vee e)$ notiamo che una delle deduzioni ausiliarie utilizzate consiste del solo nodo F, che svolge simultaneamente il ruolo di radice e di foglia. Ciò significa che questa deduzione naturale mostra $F \rhd F$. Nel suo utilizzo all'interno di $(\vee e)$, ciò implica che F è sia la conclusione (visto che anche l'altra deduzione ha la stessa conclusione) che una delle ipotesi ausiliarie da scaricare.

Esempio 5.14. L'utilizzo consecutivo di $(\neg i)$ e $(\neg\neg e)$ nell'esempio precedente può essere isolato considerando la deduzione naturale

$$\dfrac{\dfrac{\bot}{\neg\neg F}\,1}{F}$$

Otteniamo perciò la regola

$$\dfrac{\bot}{F}$$

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*. La correttezza di questa regola è giustificata dall'Esercizio 5.4.1. In alcune presentazioni della deduzione naturale ex-falso è una regola di base, mentre per noi è derivabile dalle altre regole, ed è quindi una regola derivata.

Esempio 5.15. Un altro esempio di regola derivata è la regola (MT) di *modus tollens*[4] ("toglie" la verità di una proposizione "togliendo" quella di un'altra), che corrisponde alla conseguenza logica dell'esercizio 2.31.c:

$$\dfrac{F \to G \qquad \neg G}{\neg F}$$

Una deduzione naturale che, utilizzando $(\neg i)$, giustifica questa regola è

$$\dfrac{\dfrac{\dfrac{[F]^1 \qquad F \to G}{G} \qquad \neg G}{\bot}}{\neg F}\,1$$

(MT) è una regola che simultaneamente elimina l'implicazione e una negazione per introdurre una nuova negazione.

Esempio 5.16. Mostriamo ora come sia possibile ricavare il principio del terzo escluso (TE), cioè otteniamo $\rhd F \vee \neg F$ per ogni formula F:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[F]^1}{F \vee \neg F} \qquad [\neg(F \vee \neg F)]^2}{\bot}}{\neg F}\,1}{F \vee \neg F} \qquad [\neg(F \vee \neg F)]^2}{\bot}}{F \vee \neg F}\,2$$

Si noti che nel passaggio etichettato con 2 abbiamo utilizzato la regola (RAA).

[4] modus tollens abbrevia la locuzione latina *modus tollendo tollens*, che letteralmente significa "modo che toglie con l'aver tolto", ovvero "modo che toglie".

3. 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.

$$(\wedge i)\ \ \dfrac{\begin{array}{c}T\\ \triangledown\\ F\end{array}\qquad \begin{array}{c}T'\\ \triangledown\\ G\end{array}}{F\wedge G}\qquad\qquad (\wedge e.1)\ \ \dfrac{\begin{array}{c}T\\ \triangledown\\ F\wedge G\end{array}}{F}\qquad\qquad (\wedge e.2)\ \ \dfrac{\begin{array}{c}T\\ \triangledown\\ F\wedge G\end{array}}{G}$$

$$(\vee i.1)\ \ \dfrac{\begin{array}{c}T\\ \triangledown\\ F\end{array}}{F\vee G}\qquad\qquad (\vee i.2)\ \ \dfrac{\begin{array}{c}T\\ \triangledown\\ G\end{array}}{F\vee G}\qquad\qquad (\vee e)\ \ \dfrac{\begin{array}{c}T\\ \triangledown\\ F\vee G\end{array}\qquad \begin{array}{c}T',[F]\\ \triangledown\\ H\end{array}\qquad \begin{array}{c}T'',[G]\\ \triangledown\\ H\end{array}}{H}$$

$$(\rightarrow i)\ \ \dfrac{\begin{array}{c}T,[F]\\ \triangledown\\ G\end{array}}{F\rightarrow G}\qquad\qquad (\rightarrow e)\ \ \dfrac{\begin{array}{c}T\\ \triangledown\\ F\end{array}\qquad \begin{array}{c}T'\\ \triangledown\\ F\rightarrow G\end{array}}{G}$$

$$(\neg i)\ \ \dfrac{\begin{array}{c}T,[F]\\ \triangledown\\ \bot\end{array}}{\neg F}\qquad\qquad (\neg e)\ \ \dfrac{\begin{array}{c}T\\ \triangledown\\ F\end{array}\qquad \begin{array}{c}T'\\ \triangledown\\ \neg F\end{array}}{\bot}\qquad\qquad (\neg\neg e)\ \ \dfrac{\begin{array}{c}T\\ \triangledown\\ \neg\neg F\end{array}}{F}$$

Ci limitiamo ad un esempio per chiarire il significato di queste regole: la regola $(\vee e)$ va letta come "se da T deduco $F\vee G$, da T', F deduco H e da T'', G deduco H, allora da T, T', T'' posso dedurre H" (ricordate che in base alla convenzione 1.6 T, T', T'' significa $T\cup T'\cup T''$).

Negli esempi 5.12, 5.14, 5.15 e 5.16, a partire dalle regole riportate sopra abbiamo ottenuto le seguenti regole derivate:

$$(RAA)\ \ \dfrac{\begin{array}{c}T,[\neg F]\\ \triangledown\\ \bot\end{array}}{F}\qquad\qquad (\textit{ex-falso})\ \ \dfrac{\begin{array}{c}T\\ \triangledown\\ \bot\end{array}}{F}$$

$$(MT)\ \ \dfrac{\begin{array}{c}T\\ \triangledown\\ F\rightarrow G\end{array}\qquad \begin{array}{c}T'\\ \triangledown\\ \neg G\end{array}}{\neg F}\qquad\qquad (TE)\quad [F\vee\neg F]$$

La regola (TE) afferma che ipotesi della forma $F\vee\neg F$ sono immediatamente considerate scaricate. Dato che lo scaricamento di $F\vee\neg F$ non avviene in corrispondenza di una regola successiva, quando useremo (TE) non indicheremo nessun numero ad esponente.

Useremo liberamente queste quattro regole derivate nelle nostre deduzioni naturali.

4. Correttezza e completezza della deduzione naturale proposizionale

In questa sezione dimostriamo che la deduzione naturale è un sistema deduttivo corretto (nel senso formulato all'inizio della sezione 5.1) e enunciamo, senza dimostrarla, la sua completezza.

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

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 $T \vDash F$.

Il caso base è quello in cui l'altezza dell'albero è 0. Ciò significa che la deduzione naturale consiste di un solo nodo etichettato con una formula F. La deduzione naturale si presenta quindi come

$$F$$

e mostra che $F \rhd F$. Dato che $F \vDash F$, la conclusione è immediata.

Quando l'albero ha altezza $n+1$ concentriamo la nostra attenzione sull'ultima regola utilizzata nella costruzione dell'albero, cioè quella con cui si è giunti alla conclusione della deduzione naturale. Per ognuna delle possibili undici regole utilizzate (naturalmente non dobbiamo considerare le regole derivate) è necessaria una dimostrazione. Molte di queste dimostrazioni sono banali (spesso abbiamo già menzionato la conseguenza o equivalenza logica da utilizzare nella sezione 5.1, presentando le regole nel contesto di un sistema deduttivo arbitrario). Ci limitiamo quindi ad alcuni casi (comprendendo tutte le regole condizionali, che sono quelle più delicate), lasciando al lettore il compito di verificare gli altri.

($\wedge i$) Se la deduzione naturale è della forma

$$\frac{\begin{array}{cc} T & T' \\ \triangledown & \triangledown \\ F & G \end{array}}{F \wedge G}$$

abbiamo $T, T' \rhd F \wedge G$ e vogliamo dimostrare $T, T' \vDash F \wedge G$. Le due deduzioni naturali sopra la linea orizzontale mostrano che $T \rhd F$ e $T' \rhd G$ e hanno entrambe altezza $\leq n$. Per ipotesi induttiva abbiamo dunque $T \vDash F$ e $T' \vDash G$. Da questo è facile ottenere (usando l'esercizio 2.31.a) che $T, T' \vDash F \wedge G$.

($\vee e$) In questo caso la deduzione naturale è della forma

$$\frac{\begin{array}{ccc} T & T', [F] & T'', [G] \\ \triangledown & \triangledown & \triangledown \\ F \vee G & H & H \end{array}}{H}$$

e mostra $T, T', T'' \rhd H$. Le deduzioni naturali sopra la linea orizzontale hanno altezza $\leq n$ e mostrano che $T \rhd F \vee G$, che $T', F \rhd H$ e che $T'', G \rhd H$ (notiamo che in queste deduzioni F e G non sono ancora state scaricate). L'ipotesi induttiva ci garantisce che $T \vDash F \vee G$, che $T', F \vDash H$ e che $T'', G \vDash H$: da queste ipotesi dobbiamo ottenere $T, T', T'' \vDash H$. Se v è un'interpretazione che soddisfa T, T', T'' allora, dato che $T \vDash F \vee G$, soddisfa anche $F \vee G$. Se $v(F) = \mathbb{V}$ allora $v(H) = \mathbb{V}$ discende da $T', F \vDash H$, altrimenti vale $v(G) = \mathbb{V}$ e $v(H) = \mathbb{V}$ è una conseguenza di $T'', G \vDash H$.

($\to i$) In questo caso abbiamo una deduzione naturale della forma

$$\frac{\begin{array}{c} T, [F] \\ \triangledown \\ G \end{array}}{F \to G}$$

che mostra $T \rhd F \to G$. Il nostro obiettivo è quindi mostrare che $T \models F \to G$. La deduzione naturale sopra la linea orizzontale (in cui F non è ancora stata scaricata) mostra che $T, F \rhd G$ e ha altezza $\leq n$. L'ipotesi induttiva ci garantisce che $T, F \models G$. Se v è un'interpretazione che soddisfa T consideriamo due casi: se $v(F) = \mathbb{F}$ allora $v(F \to G) = \mathbb{V}$, mentre se $v(F) = \mathbb{V}$ possiamo usare $T, F \models G$ per ottenere $v(G) = \mathbb{V}$ e quindi nuovamente $v(F \to G) = \mathbb{V}$. Abbiamo sostanzialmente imitato la dimostrazione di una direzione del lemma 2.40.a per ottenere $T \models F \to G$.

($\neg i$) Stiamo considerando una deduzione naturale della forma

$$\dfrac{\begin{array}{c} T, [F] \\ \triangledown \\ \bot \end{array}}{\neg F}$$

che mostra $T \rhd \neg F$. La deduzione naturale sopra la linea orizzontale (in cui F non è ancora stata scaricata) mostra che $T, F \rhd \bot$ e ha altezza $\leq n$. L'ipotesi induttiva ci garantisce che $T, F \models \bot$. Per l'esercizio 5.4.12 ciò significa che l'insieme T, F è insoddisfacibile. Perciò se v è un'interpretazione che soddisfa T non può verificarsi $v(F) = \mathbb{V}$. Quindi $v(F) = \mathbb{F}$, cioè $v(\neg F) = \mathbb{V}$. Abbiamo quindi dimostrato $T \models \neg F$, come volevamo.

($\neg e$) Stiamo considerando una deduzione naturale della forma

$$\dfrac{\begin{array}{c} T \\ \triangledown \\ F \end{array} \qquad \begin{array}{c} T' \\ \triangledown \\ \neg F \end{array}}{\bot}$$

che mostra $T, T' \rhd \bot$. Dobbiamo dunque mostrare $T, T' \models \bot$, che per l'esercizio 5.4.12 significa che T, T' è insoddisfacibile. Le deduzioni naturali sopra la linea orizzontale mostrano che $T \rhd F$ e $T' \rhd \neg F$, ed hanno altezza $\leq n$. L'ipotesi induttiva ci garantisce che $T \models F$ e $T' \models \neg F$. Dunque ogni interpretazione che soddisfa T, T' dovrebbe soddisfare sia F che $\neg F$. Dato che nessuna interpretazione può soddisfare una formula e la sua negazione, nessuna interpretazione soddisfa T, T', cioè T, T' è insoddisfacibile, come volevamo. $\square$

Il teorema di correttezza implica che gli esempi 5.11 e 5.10 mostrano che $\neg F \wedge \neg G \models \neg(F \vee G)$ e che $\neg F \vee \neg G \models \neg(F \wedge G)$. Essi forniscono dunque una dimostrazione attraverso un sistema deduttivo di una conseguenza logica per ognuna delle leggi di De Morgan (lemma 2.24.1-2). Nell'esempio 5.27 e nell'esercizio 5.28 otterremo con la deduzione naturale anche le altre conseguenze logiche.

L'inverso del teorema di correttezza è il teorema di completezza, la cui dimostrazione esula dai limiti di questo corso.

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

Lemma 5.19. *Siano T e T' insiemi di formule proposizionali e F e G formule proposizionali. Se $T \rhd G$ e $T', G \rhd F$ allora $T, T' \rhd F$.*

Dimostrazione. E' sufficiente combinare le due deduzioni nel seguente modo:

$$\dfrac{\begin{array}{c} T \\ \triangledown \\ G \end{array} \qquad T'}{\begin{array}{c} \triangledown \\ F \end{array}}$$

□

5. La deduzione naturale e la logica intuizionistica

La deduzione naturale si presta ad introdurre la distinzione tra *logica classica* (quella che abbiamo utilizzato sin qui e che continueremo ad utilizzare) e *logica intuizionistica*. Nella logica intuizionistica si ritiene ad esempio che una dimostrazione di $F \vee G$ debba sempre comprendere l'indicazione di quale tra F e G viene dimostrata. Ciò significa che si può assumere $F \vee \neg F$ solo nei casi in cui si possa decidere quale tra F e $\neg F$ sia vera. Quindi il principio del terzo escluso (che asserisce che $F \vee \neg F$ è vera per ogni F) non è intuizionisticamente accettabile. L'approccio intuizionistico è particolarmente interessante se la deducibilità viene interpretata come dimostrabilità: infatti è possibile che né F né $\neg F$ siano dimostrabili. Per la logica intuizionistica è necessario introdurre una semantica diversa da quella presentata nel capitolo 2.

Solitamente i matematici tendono ad accettare il principio del terzo escluso, sebbene a volte il suo uso abbia conseguenze che possono lasciare qualche dubbio. Considerate ad esempio la seguente dimostrazione di un fatto che riguarda i numeri reali.

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 $a = \sqrt{2}^{\sqrt{2}}$.

In entrambi i casi a è irrazionale e dobbiamo verificare che a^b è razionale. Nel primo caso ciò è immediato per ipotesi, mentre nel secondo abbiamo $a^b = \left(\sqrt{2}^{\sqrt{2}}\right)^{\sqrt{2}} = \left(\sqrt{2}\right)^{\sqrt{2}\cdot\sqrt{2}} = \left(\sqrt{2}\right)^2 = 2$, che è razionale. □

Questa dimostrazione usa il terzo escluso ($\sqrt{2}^{\sqrt{2}}$ è razionale oppure $\sqrt{2}^{\sqrt{2}}$ non è razionale) ed è classicamente corretta, ma può lasciare perplessi: abbiamo dimostrato l'esistenza di a e b con certe proprietà, ma non siamo in grado di esibire questi due numeri (perché non sappiamo quale sia a)[5].

Per ottenere una versione della deduzione naturale che sia in accordo con i principi intuizionistici è certamente necessario rinunciare a (TE). Però (TE) è una regola derivata. Nell'esempio 5.16 (TE) è stata derivata utilizzando (RAA) (che è intuizionisticamente inaccettabile: se da $\neg F$ deduciamo una contraddizione allora sappiamo che $\neg F$ non è dimostrabile, che non implica che F sia dimostrabile, dato che il terzo escluso non vale). A sua volta (RAA) è stata ottenuta nell'esempio 5.12 a partire da $(\neg\neg e)$. La regola di base che è in contrasto con l'intuizionismo è dunque $(\neg\neg e)$.

[5]Nel caso specifico del teorema 5.20 è possibile dare una dimostrazione che esibisce i due numeri: basta prendere $a = e$ e $b = \ln(2)$: si può dimostrare (utilizzando un po' di teoria dei numeri avanzata, di cui la nostra dimostrazione non ha bisogno) che entrambi questi numeri sono irrazionali, mentre $a^b = 2$ per definizione di logaritmo.

Eliminando la regola $(\neg\neg e)$ si ottiene una versione più debole della deduzione naturale. Per ottenerne una che corrisponda alla logica intuizionistica è necessario inserire tra le regole di base (*ex-falso*), che è intuizionisticamente accettabile ma nell'esempio 5.14 era stato ottenuto utilizzando $(\neg\neg e)$. Nella deduzione naturale intuizionistica rimane la regola derivata (MT) (nell'esempio 5.15 non si è fatto uso di principi non intuizionistici), mentre ovviamente (RAA) e (TE) scompaiono.

In questo corso non abbiamo spazio per sviluppare ulteriormente queste problematiche, e continueremo quindi a lavorare nell'ambito della logica classica, limitandoci di tanto in tanto ad evidenziare l'uso delle regole non intuizionistiche.

6. Esempi di deduzione naturale proposizionale

In questa sezione presentiamo numerosi esempi di deduzione naturale, spesso ottenendo alcune delle conseguenze logiche della sezione 2.2 (nella maggior parte dei casi sono una parte di un'equivalenza logica). Nei primi esempi cerchiamo di spiegare al lettore come costruire le deduzioni naturali desiderate, mentre più avanti presenteremo semplicemente le deduzioni naturali. Suggeriamo però di studiare a fondo queste deduzioni naturali per capire il ragionamento che viene formalizzato. Solo in questo modo sarà possibile imparare a costruire autonomamente deduzioni naturali. E' anche consigliabile provare a costruire alcune di queste deduzioni naturali prima di guardare le soluzioni: anche quando non si avesse successo ciò permette di comprendere ed apprezzare meglio la soluzione proposta.

Esempio 5.21. Vogliamo costruire una deduzione naturale che mostri

$$\rhd (F \to (G \to H)) \to (F \wedge G \to H).$$

(La validità della formula che dobbiamo dedurre segue dalla prima equivalenza logica dell'esercizio 2.26.) La conclusione della deduzione naturale (che non ha ipotesi) è un'implicazione, e quindi pare naturale usare $(\to i)$ per ottenerla (questa non è l'unica scelta possibile: possiamo anche immaginare che l'ultima regola sia $(\wedge e)$, oppure (RAA), anche se nel nostro caso specifico è difficile trovare una deduzione con queste caratteristiche). Quindi possiamo usare l'antecedente $F \to (G \to H)$ come ipotesi ausiliaria, con l'obiettivo di ottenere il conseguente $F \wedge G \to H$. Dato che quest'ultimo è nuovamente un'implicazione, pensiamo di usare un'altra volta $(\to i)$: possiamo dunque ipotizzare $F \wedge G$ con lo scopo di ottenere H. In altre parole è sufficiente mostrare che $F \to (G \to H), F \wedge G \rhd H$. Questo obiettivo è più facile da raggiungere di quello iniziale, e porta alla deduzione naturale

$$\dfrac{\dfrac{F \wedge G}{G} \qquad \dfrac{\dfrac{F \wedge G}{F} \qquad F \to (G \to H)}{G \to H}}{H}$$

Utilizzando ora due volte $(\to i)$ otteniamo la deduzione naturale cercata:

$$\dfrac{\dfrac{\dfrac{\dfrac{[F \wedge G]^1}{G} \qquad \dfrac{\dfrac{[F \wedge G]^1}{F} \qquad [F \to (G \to H)]^2}{G \to H}}{H}}{F \wedge G \to H}\,1}{(F \to (G \to H)) \to (F \wedge G \to H)}\,2$$

Esercizio 5.22. Mostrare che $\rhd (F \wedge G \to H) \to (F \to (G \to H))$.

Esempio 5.23. Mostriamo ora che $F \to H, G \to H \rhd F \vee G \to H$. Anche in questo caso la conclusione è un'implicazione e possiamo supporre di voler usare $(\to i)$: assumiamo dunque $F \vee G$ con l'obiettivo di ottenere H. In altre parole vogliamo ottenere $F \to H, G \to H, F \lor G \rhd H$. Per sfruttare l'ipotesi ausiliaria $F \lor G$ è probabilmente necessario ricorrere a $(\lor e)$. Dobbiamo quindi usare separatamente F e G per ottenere H, cioè mostrare da un lato che $F \to H, G \to H, F \rhd H$ e dall'altro che $F \to H, G \to H, G \rhd H$. Queste ultime deduzioni sono entrambe immediate e conducono alla deduzione naturale

$$\dfrac{\dfrac{[F \lor G]^2 \qquad \dfrac{[F]^1 \qquad F \to H}{H} \qquad \dfrac{[G]^1 \qquad G \to H}{H}}{H}\,{\scriptstyle 1}}{F \lor G \to H}\,{\scriptstyle 2}$$

Esempio 5.24. Vogliamo mostrare che $F \lor (G \land H) \rhd (F \lor G) \land (F \lor H)$. Dato che in questo caso la conclusione è una congiunzione supponiamo che l'ultima regola applicata sia $(\land i)$. E' dunque sufficiente mostrare che $F \lor (G \land H) \rhd F \lor G$ e che $F \lor (G \land H) \rhd F \lor H$. Per farlo è opportuno usare $(\lor e)$. La deduzione naturale che si ottiene è la seguente:

$$\dfrac{\dfrac{F \lor (G \land H) \qquad \dfrac{[F]^1}{F \lor G} \qquad \dfrac{\dfrac{[G \land H]^1}{G}}{F \lor G}}{F \lor G}\,{\scriptstyle 1} \qquad \dfrac{F \lor (G \land H) \qquad \dfrac{[F]^2}{F \lor H} \qquad \dfrac{\dfrac{[G \land H]^2}{H}}{F \lor H}}{F \lor H}\,{\scriptstyle 2}}{(F \lor G) \land (F \lor H)}$$

Esempio 5.25. Cerchiamo ora una deduzione naturale che mostri $(F \lor G) \land (F \lor H) \rhd F \lor (G \land H)$. Un modo per farlo è ripercorre i passi della dimostrazione della conseguenza logica corrispondente nel lemma 2.24.10, distinguendo il caso in cui F è vera da quello in cui è falsa ed utilizzando quindi (TE). Nel primo caso la conclusione è immediata, mentre nel secondo è necessario un po' più di lavoro per arrivare a concludere che $G \land H$ è vero. L'idea è dunque quella di ottenere una deduzione naturale della forma

$$\dfrac{[F \lor \neg F] \qquad \dfrac{[F]^1}{F \lor (G \land H)} \qquad \dfrac{\begin{array}{c}(F \lor G) \land (F \lor H), [\neg F]^1 \\ \triangledown \\ G \land H\end{array}}{F \lor (G \land H)}}{F \lor (G \land H)}\,{\scriptstyle 1}$$

A questo punto è necessario mostrare che $(F \lor G) \land (F \lor H), \neg F \rhd G \land H$. Per farlo basta dimostrare $(F \lor G) \land (F \lor H), \neg F \rhd G$ e $(F \lor G) \land (F \lor H), \neg F \rhd H$ per poi utilizzare $(\land i)$. Ecco una deduzione naturale che mostra $(F \lor G) \land (F \lor H), \neg F \rhd G$ (la deduzione di H dalle stesse ipotesi è analoga):

$$\dfrac{\dfrac{(F \lor G) \land (F \lor H)}{F \lor G} \qquad \dfrac{\dfrac{[F]^1 \qquad \neg F}{\bot}}{G} \qquad [G]^1}{G}\,{\scriptstyle 1}$$

Si noti l'utilizzo di (*ex-falso*) in questa deduzione.

Proponiamo ora una deduzione naturale diversa dalla precedente per mostrare che $(F \lor G) \land (F \lor H) \rhd F \lor (G \land H)$:

$$\dfrac{\dfrac{(F \lor G) \land (F \lor H)}{F \lor G} \qquad \dfrac{[F]^2}{F \lor (G \land H)} \qquad \dfrac{\dfrac{(F \lor G) \land (F \lor H)}{(F \lor H)} \qquad \dfrac{[F]^1}{F \lor (G \land H)} \qquad \dfrac{\dfrac{[G]^2 \qquad [H]^1}{G \land H}}{F \lor (G \land H)}}{F \lor (G \land H)}\,{\scriptstyle 1}}{F \lor (G \land H)}\,{\scriptstyle 2}$$

E' quindi evidente che deduzioni naturali diverse possono avere le stesse ipotesi e conclusione. In questo caso è utile ripercorrere le due deduzioni naturali per comprendere come esse formalizzino dimostrazioni essenzialmente differenti dello stesso fatto. Questa differenza è ancor più significativa perché la prima deduzione naturale usa (TE), che è intuizionisticamente inaccettabile, mentre la seconda usa solo regole ammesse dalla logica intuizionistica.

Il teorema di correttezza 5.17 implica che i due esempi precedenti mostrano rispettivamente che $F \vee (G \wedge H) \models (F \vee G) \wedge (F \vee H)$ e che $(F \vee G) \wedge (F \vee H) \models F \vee (G \wedge H)$, e quindi che $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$. Essi forniscono dunque una dimostrazione attraverso la deduzione naturale del lemma 2.24.10.

Esercizio 5.26. Dimostrare con la deduzione naturale le due conseguenze logiche contenute nel lemma 2.24.11.

Completiamo ora la dimostrazione delle leggi di De Morgan (lemma 2.24.1–2) per mezzo della deduzione naturale (per le altre direzioni si vedano gli esempi 5.11 e 5.10).

Esempio 5.27. Per dimostrare $\neg(F \wedge G) \rhd \neg F \vee \neg G$ formalizziamo il ragionamento utilizzato nella dimostrazione del lemma 2.24.1, che si basava sul considerare separatamente il caso in cui vale $\neg F$ da quello in cui vale F:

$$
\begin{prooftree}
\AxiomC{$[F \vee \neg F]$}
\AxiomC{$\neg(F \wedge G)$}
\AxiomC{$[F]^2$}
\AxiomC{$[G]^1$}
\BinaryInfC{$F \wedge G$}
\BinaryInfC{$\bot$}
\RightLabel{1}
\UnaryInfC{$\neg G$}
\UnaryInfC{$\neg F \vee \neg G$}
\AxiomC{$[\neg F]^2$}
\UnaryInfC{$\neg F \vee \neg G$}
\RightLabel{2}
\TrinaryInfC{$\neg F \vee \neg G$}
\end{prooftree}
$$

In questo caso è noto che l'uso di una regola non intuizionistica come (TE) non può essere evitato.

Esercizio 5.28. Dimostrare che $\neg(F \vee G) \rhd \neg F \wedge \neg G$.

Esempio 5.29. In questo esempio dimostriamo attraverso la deduzione naturale l'equivalenza logica del lemma 2.24.3. Anche in questo caso seguiamo le idee delle dimostrazioni fatte a suo tempo. Iniziamo con il dimostrare che $F \to G \rhd \neg F \vee G$:

$$
\begin{prooftree}
\AxiomC{$[F \vee \neg F]$}
\AxiomC{$[F]^1$}
\AxiomC{$F \to G$}
\BinaryInfC{G}
\UnaryInfC{$\neg F \vee G$}
\AxiomC{$[\neg F]^1$}
\UnaryInfC{$\neg F \vee G$}
\RightLabel{1}
\TrinaryInfC{$\neg F \vee G$}
\end{prooftree}
$$

Per dimostrare invece che $\neg F \vee G \rhd F \to G$ basta costruire la deduzione naturale

$$
\begin{prooftree}
\AxiomC{$\neg F \vee G$}
\AxiomC{$[F]^1$}
\AxiomC{$[\neg F]^3$}
\BinaryInfC{$\bot$}
\UnaryInfC{G}
\RightLabel{1}
\UnaryInfC{$F \to G$}
\AxiomC{$[G]^3$}
\RightLabel{2}
\UnaryInfC{$F \to G$}
\RightLabel{3}
\TrinaryInfC{$F \to G$}
\end{prooftree}
$$

Si noti come in quest'ultima deduzione naturale abbiamo utilizzato $(\to i)$ senza effettuare alcuno scaricamento. Infatti $(\to i)$ ci permette di ottenere $G \rhd F \to G$.

Una deduzione naturale analoga alla precedente, in cui però l'ordine di applicazione di $(\to i)$ e $(\vee e)$ viene invertito, è la seguente:

$$
\begin{prooftree}
\AxiomC{$\neg F \vee G$}
\AxiomC{$[F]^2$}
\AxiomC{$[\neg F]^1$}
\BinaryInfC{$\bot$}
\UnaryInfC{G}
\AxiomC{$[G]^1$}
\RightLabel{1}
\TrinaryInfC{G}
\RightLabel{2}
\UnaryInfC{$F \to G$}
\end{prooftree}
$$

Esempio 5.30. Consideriamo ora l'equivalenza logica del lemma 2.24.4. Iniziamo con il dimostrare che $F \land \neg G \rhd \neg(F \to G)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{F \land \neg G}{F} \qquad [F \to G]^1}{G} \qquad \dfrac{F \land \neg G}{\neg G}}{\bot}}{\neg(F \to G)}\,{\scriptstyle 1}$$

Ecco una deduzione naturale alternativa per lo stesso risultato:

$$\dfrac{\dfrac{\dfrac{F \land \neg G}{F} \qquad \dfrac{[F \to G]^1 \qquad \dfrac{F \land \neg G}{\neg G}}{\neg F}}{\bot}}{\neg(F \to G)}\,{\scriptstyle 1}$$

Si noti l'uso di (MT) per ottenere $\neg F$.

Occupiamoci ora di dimostrare che $\neg(F \to G) \rhd F \land \neg G$. A questo scopo è sufficiente dimostrare che $\neg(F \to G) \rhd F$ e che $\neg(F \to G) \rhd \neg G$, per poi combinare le due deduzioni naturali con un'applicazione di $(\land i)$. Nella prima deduzione naturale usiamo $\neg F \rhd F \to G$, già utilizzata nelle ultime due deduzioni dell'esempio 5.29, e (RAA), nella seconda $(\to i)$ senza scaricamento. La deduzione naturale completa risulta essere

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[F]^1 \qquad [\neg F]^2}{\bot}}{G}}{F \to G}\,{\scriptstyle 1} \qquad \neg(F \to G)}{\bot}}{F}\,{\scriptstyle 2} \qquad \dfrac{\dfrac{\dfrac{[G]^4}{F \to G}\,{\scriptstyle 3} \qquad \neg(F \to G)}{\bot}}{\neg G}\,{\scriptstyle 4}}{F \land \neg G}$$

Esempio 5.31. Dimostriamo ora che $F \to G \rhd \neg G \to \neg F$ (la conseguenza logica corrispondente è contenuta nel lemma 2.24.5), utilizzando (MT):

$$\dfrac{\dfrac{F \to G \qquad [\neg G]^1}{\neg F}}{\neg G \to \neg F}\,{\scriptstyle 1}$$

Esercizio 5.32. Dimostrare che $\neg G \to \neg F \rhd F \to G$.

Esempio 5.33. In questo esempio dimostriamo che $F \lor G, \neg F \lor H \rhd G \lor H$.

$$\dfrac{F \lor G \qquad \dfrac{\neg F \lor H \qquad \dfrac{\dfrac{[F]^2 \qquad [\neg F]^1}{\bot}}{G \lor H} \qquad \dfrac{[H]^1}{G \lor H}}{G \lor H}\,{\scriptstyle 1} \qquad \dfrac{[G]^2}{G \lor H}}{G \lor H}\,{\scriptstyle 2}$$

Esercizio 5.34. Costruire una deduzione naturale diversa dalla precedente che dimostri $F \lor G, \neg F \lor H \rhd G \lor H$ utilizzando (TE) e partendo da $[F \lor \neg F]$. Notare come questa deduzione naturale sia più complessa della precedente.

Esercizio 5.35. Usare la deduzione naturale per dimostrare le equivalenze logiche del lemma 2.24.6-7 (associatività di congiunzione e disgiunzione).

Esercizio 5.36. Usare la deduzione naturale per dimostrare le conseguenze ed equivalenze logiche dell'esercizio 5.4.

Esempio 5.37. Una deduzione che mostra che $F \to \neg G \lor \neg H, G \to H \rhd G \to \neg F$ è la seguente:

$$\dfrac{\dfrac{\dfrac{\dfrac{[F]^2 \qquad F \to \neg G \vee \neg H}{\neg G \vee \neg H} \qquad \dfrac{[G]^3 \qquad [\neg G]^1}{\bot} \qquad \dfrac{\dfrac{[G]^3 \qquad G \to H}{H} \qquad [\neg H]^1}{\bot}}{\bot}\,{\scriptstyle 1}}{\neg F}\,{\scriptstyle 2}}{G \to \neg F}\,{\scriptstyle 3}$$

Esercizio 5.38. Usare la deduzione naturale per dimostrare la conseguenza logica dell'esercizio 2.61.

Parte 2

La logica predicativa

CAPITOLO 6

Sintassi della logica predicativa

In questo capitolo definiremo la sintassi della logica predicativa. Nel farlo faremo spesso riferimento ad una semantica "informale" per gli elementi sintattici di questa logica (qualcosa di simile, ma in forma meno evidente, era già accaduto nel capitolo 1). Quest'uso della semantica nel definire la sintassi ha lo scopo di motivare le definizioni che verranno date, che altrimenti risulterebbero di difficile comprensione. La definizione della sintassi resta comunque indipendente dalla semantica.

Le formule predicative sono asserzioni riguardo alle relazioni che intercorrono tra certi oggetti. Esse si differenziano dalle formule proposizionali proprio per la possibilità di riferirsi ad oggetti.

Per poter parlare di oggetti useremo espressioni che non sono formule: i termini. A differenza delle formule, anche dopo lo sviluppo della semantica i termini non saranno né veri né falsi. Come già fatto nel capitolo 2, ci può aiutare considerare la situazione dell'aritmetica: $(1 + 1) \cdot 1$, che è un termine in un opportuno linguaggio (quello dell'esempio 6.3 qui sotto), designa un elemento (il numero 2, se tutti i simboli sono interpretati in modo naturale), e non può essere né vero né falso. $0 < (1 + 1) \cdot 1$ è invece una formula del linguaggio dell'esempio 6.3 e risulta essere vera se tutti i simboli sono interpretati in modo naturale. Formule più complicate nello stesso linguaggio sono $\forall x \, \exists y \, x < y$ e $\exists y \forall x \, x < y$, che sono rispettivamente vera e falsa (sempre se tutti i simboli sono interpretati in modo naturale nell'ambito dell'insieme dei numeri naturali $\mathbb{N}$). Per stabilire se la formula $1 < x$ sia vera o falsa è necessario specificare, oltre all'interpretazione di 1 e di $<$, anche il significato che si attribuisce a x. Notate che x era presente anche in alcune formule precedenti, ma in quei casi non era necessario interpretarlo per capire se la formula fosse vera o falsa: questa distinzione viene resa precisa dalle nozioni di variabile libera e legata (definizione 6.38).

1. Linguaggi predicativi

Come nel caso delle formule proposizionali, anche la costruzione dei termini e delle formule predicative avviene per ricorsione (definizioni 6.9 e 6.20). Prima di poter dare queste definizioni dobbiamo fissare gli ingredienti fondamentali che usiamo per costruire termini e formule.

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 $x, y, z, \dots$;
- 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 ")".

Questi elementi saranno sempre a nostra disposizione. Oltre a questi vi saranno altri simboli, scelti di volta in volta a seconda dell'argomento su cui si intende fare delle affermazioni. Questi elementi variabili (ma le variabili non sono tra essi!) identificano un linguaggio.

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*), ciascuno fornito della propria *arietà*, che è un numero naturale ≥ 1.

Intuitivamente la arietà di un simbolo di funzione indica a quanti elementi esso si può applicare. Simile è il significato della arietà di un simbolo di relazione. Se un simbolo di funzione o di relazione ha arietà n diciamo anche che è n-ario.

Notiamo che gli insiemi dei simboli di costante e di funzione possono essere vuoti, mentre ogni linguaggio contiene sempre almeno un simbolo di relazione. Questa è l'unica restrizione che poniamo sui tre insiemi: ognuno di essi può essere finito o infinito, quelli dei simboli di funzione e relazione possono contenere solo simboli di arietà 1 (detti usualmente unari), oppure qualche simbolo di arietà 2 (binario) e qualcuno di arietà 3 (ternario), oppure ancora simboli di infinite arietà diverse.

Esempio 6.3. Il linguaggio $\mathcal{L}_{\mathrm{arit}}$, adatto a fare affermazioni sull'aritmetica, consiste dei simboli di costante 0 e 1, dei simboli di funzione binari $+$ e $\cdot$, dei simboli di relazione binari $<$, $>$ e $=$.

Esempio 6.4. Il linguaggio $\mathcal{L}_{\mathbb{R}}$, adatto a fare affermazioni sui numeri reali, consiste dei simboli di costante 0, 1, e e π, dei simboli di funzione unari di elevazione all'n-esima potenza ^{n} (uno per ogni $n \in \mathbb{N}$, così che abbiamo infiniti simboli di funzione), sin e cos, dei simboli di funzione binari $+$ e $\cdot$, dei simboli di relazione binari $<$, $>$ e $=$.

Esempio 6.5. Il linguaggio $\mathcal{L}_{\mathrm{seq}}$, adatto a fare affermazioni sulle stringhe di numeri naturali, consiste dei simboli di costante $\emptyset$ e $\langle n \rangle$ (uno per ogni n, così che abbiamo infiniti simboli di costante), del simbolo di funzione binario $^\frown$, dei simboli di relazione binari $\subset$ e $=$. [$\emptyset$ rappresenta la stringa vuota, $\langle n \rangle$ quella il cui unico elemento è n, $^\frown$ la concatenazione di due stringhe, $\subset$ l'essere segmento iniziale.]

Esempio 6.6. Il linguaggio $\mathcal{L}_{\mathrm{set}}$, adatto a fare affermazioni sugli insiemi, consiste del simbolo di costante $\emptyset$, dei simboli di funzione unari $\{\}$, $\bigcup$ e $\mathcal{P}$, dei simboli di funzione binari $\cup$, $\cap$ e $\setminus$, dei simboli di relazione binari $\in$, $\subseteq$ e $=$.

Esempio 6.7. Il linguaggio $\mathcal{L}_{\mathrm{fam}}$, adatto a fare affermazioni sulle relazioni di parentela, consiste di un simbolo di costante per ogni membro della famiglia, dei simboli di funzione unari p ("il padre di"), e m ("la madre di"), dei simboli di relazione binari f_1 ("sono fratelli") e f_2 ("è figlio di").

Esempio 6.8. Il linguaggio $\mathcal{L}_{\mathrm{gr}}$, adatto a fare affermazioni sui vertici di un grafo, consiste dei simboli di relazione binari $=$ e E (che intercorre tra due vertici quando sono adiacenti).

2. Termini

I termini di un linguaggio fissato sono costruiti utilizzando solo alcuni dei suoi elementi: le variabili e i simboli di costante e di funzione, ed inoltre la virgola e le parentesi.

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 $\mathcal{L}$.

Un termine è *chiuso* se in esso non compare nessuna variabile (per definire i termini chiusi si può ripetere la definizione precedente omettendo la prima condizione).

Se un linguaggio non contiene né simboli di costante né simboli di funzione allora i suoi termini sono solamente le variabili. Se un linguaggio non contiene simboli di costante allora non esiste alcun termine chiuso.

Esempio 6.10. Alcuni termini del linguaggio $\mathcal{L}_{\mathrm{arit}}$ dell'esempio 6.3 sono i seguenti: x, 1, $+(x, 0)$ (il secondo è chiuso, gli altri no). Per quest'ultimo è più comune scrivere $x+0$, e convenzioni analoghe si usano anche per $\cdot$, così che $(x \cdot y)+(0 \cdot (y+x))$ è anch'esso un termine (non chiuso), che, con l'usuale convenzione che dà precedenza al prodotto rispetto alla somma, può venir scritto come $x \cdot y + 0 \cdot (y + x)$.

Esempio 6.11. Alcuni termini del linguaggio $\mathcal{L}_{\mathbb{R}}$ dell'esempio 6.4 sono e, $\cos(\pi)$, $\sin(\cos(x+e))$, $\sin(x \cdot y^7)+\cos(1)^3$ (anche in questo caso usiamo le usuali convenzioni di scrittura delle funzioni utilizzate in analisi): i primi due sono chiusi.

Esempio 6.12. Alcuni termini del linguaggio $\mathcal{L}_{\mathrm{seq}}$ dell'esempio 6.5 sono $\langle 3 \rangle {}^\frown \emptyset$ e $(x {}^\frown \langle 6 \rangle) {}^\frown z$: il primo è chiuso.

Esempio 6.13. Alcuni termini del linguaggio $\mathcal{L}_{\mathrm{fam}}$ dell'esempio 6.7 sono $p(x)$ e $m(p(a))$, dove a è un simbolo di costante che corrisponde a qualche membro della famiglia: il secondo è chiuso.

Esempio 6.14. Dato che il linguaggio $\mathcal{L}_{\mathrm{gr}}$ dell'esempio 6.8 non ha né simboli di costante né simboli di funzione, i suoi unici termini sono le variabili.

Esempio 6.15. Sia $\mathcal{L}_0$ un linguaggio privo di simboli di costante e con un unico simbolo di funzione f, che è unario. I termini di $\mathcal{L}_0$ sono tutte le stringhe di simboli del tipo v, $f(v)$, $f(f(v))$, $f(f(f(v)))$, $\dots$, dove v è una qualsiasi variabile (per ogni v ci sono infiniti di questi termini!). Se quando f compare k volte scriviamo $f^{(k)}(v)$ (e quindi in particolare $f^{(0)}(v)$ è v), abbiamo che tutti i termini di $\mathcal{L}_0$ sono della forma $f^{(k)}(v)$ con $k \in \mathbb{N}$ e v variabile.

Nel linguaggio $\mathcal{L}_0$ non ci sono termini chiusi.

Esempio 6.16. Sia $\mathcal{L}_1$ il linguaggio ottenuto aggiungendo a $\mathcal{L}_0$ dell'esempio precedente un unico simbolo di costante c. I termini chiusi di $\mathcal{L}_1$ sono quelli della forma $f^{(k)}(c)$ con $k \in \mathbb{N}$.

La definizione di termine è ricorsiva e questo fa sì che spesso ragioneremo induttivamente sui termini. Una dimostrazione di questo tipo è giustificata dal seguente teorema (analogo al teorema 1.10) e viene detta *per induzione sulla complessità dei termini.*

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 $\mathcal{A}(f(t_1, \dots, t_n))$.*

Allora $\mathcal{A}(t)$ vale per ogni termine t di $\mathcal{L}$.

Similmente è possibile dare definizioni procedendo *per ricorsione sulla complessità dei termini*: basta definire il risultato dell'operazione su variabili e simboli di costante e, supponendo di aver già definito il risultato dell'operazione su $t_1, \dots, t_n$, definirlo anche su $f(t_1, \dots, t_n)$.

Per ricorsione possiamo definire un'operazione fondamentale sui termini: quella di sostituzione. L'idea è che se s è un termine, x una variabile e t un termine la sostituzione di x con t in s (indicata da $s\{x/t\}$), è il termine ottenuto rimpiazzando ogni occorrenza di x in s con t.

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 $f(s_1, \dots, s_n)$ allora $s\{x/t\}$ è $f(s_1\{x/t\}, \dots, s_n\{x/t\})$.

Esempio 6.19. Consideriamo un linguaggio contenente i simboli di costante a e b e i simboli di funzione 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.

3. Formule predicative

Le formule di un linguaggio predicativo fissato sono costruite a partire dai termini per mezzo dei simboli di relazione e dei simboli logici:

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 $(F \to G)$ sono formule di $\mathcal{L}$;
- se F è una formula di $\mathcal{L}$ e x è una variabile, allora $(\forall x\, F)$ e $(\exists x\, F)$ sono formule di $\mathcal{L}$ (che vengono dette rispettivamente la *quantificazione universale* e la *quantificazione esistenziale di F rispetto a x*).

Le formule $(\neg F)$, $(F \wedge G)$, $(F \vee G)$ e $(F \to G)$ vengono lette come nel caso proposizionale. La formula $(\forall x\, F)$ viene letta "per ogni x, F" e la formula $(\exists x\, F)$ viene letta "esiste x tale che F".

Nota 6.21. Notate che se un linguaggio fosse privo di simboli di relazione non avrebbe formule atomiche e di conseguenza non avrebbe formule: questa è la ragione per cui nella definizione 6.2 abbiamo richiesto che l'insieme dei simboli di relazione sia non vuoto.

Esempio 6.22. Sia $\mathcal{L}_0$ un linguaggio privo di simboli di costante, con un simbolo di funzione unario f (si veda l'esempio 6.15), con un simbolo di relazione unario p ed un simbolo di relazione binario r. Le formule atomiche di $\mathcal{L}_0$ sono della forma $p(f^{(k)}(v))$ e $r(f^{(k)}(v), f^{(h)}(u))$ con $k, h \in \mathbb{N}$ e v e u variabili. In particolare tra di esse ci sono $p(x)$, $r(f(y), x)$ e $p(f(f(x)))$.

Notate invece che $p(p(x))$, $r(p(x), y)$, $f(p(x))$ e $r(x)$ **non** sono né termini né formule di $\mathcal{L}_0$.

Alcune formule di $\mathcal{L}_0$ sono $(\neg p(x))$, $(p(x) \wedge (r(f(y), x) \vee p(f(f(x)))))$, $(\forall x\, p(x))$, $(\exists y\, r(x, z))$ e $(((\forall x\, p(x)) \to (\neg p(x))) \to (((\neg p(x)) \wedge ((\exists y\, r(f(y), x)) \vee p(z))) \wedge p(x)))$.

Per indicare formule e insiemi di formule continueremo ad utilizzare la convenzione 1.6.

Il seguente lemma è l'analogo del lemma 1.8.

Lemma 6.23. *Ogni formula è di uno e uno solo dei seguenti sette tipi:*

- *una formula atomica;*
- *una* negazione*;*
- *una* congiunzione*;*
- *una* disgiunzione*;*
- *un'*implicazione*;*
- *una* quantificazione universale*, cioè una formula del tipo* $(\forall x\, F)$ *per una variabile* x *e una formula* F*;*
- *una* quantificazione esistenziale*, cioè una formula del tipo* $(\exists x\, F)$ *per una variabile* x *e una formula* F*.*

Esempio 6.24. Le formule dell'ultimo paragrafo dell'esempio 6.22 sono rispettivamente una negazione, una congiunzione, una quantificazione universale, una quantificazione esistenziale e un'implicazione.

L'ultima formula dell'esempio 6.22, per quanto formalmente corretta, è probabilmente al limite della leggibilità umana. Per migliorare la leggibilità delle formule adotteremo alcune convenzioni, che estendono quelle adottate nel caso proposizionale (convenzione 1.15).

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 $\to$;
- ulteriori parentesi eventualmente omesse in formule costruite con più di una $\wedge$ o $\vee$ si appoggiano a sinistra.

Con queste convenzioni l'ultima formula dell'esempio 6.22 diventa

$$(\forall x\, p(x) \to \neg p(x)) \to \neg p(x) \wedge (\exists y\, r(f(y), x) \vee p(z)) \wedge p(x).$$

Esercizio 6.26. Stabilite qual è la differenza tra le formule

$$\forall x\, p(x) \to \exists y\, r(x, y) \wedge \neg \exists u\, r(f(u), z)$$
$$\forall x\, p(x) \to \exists y\, (r(x, y) \wedge \neg \exists u\, r(f(u), z))$$
$$\forall x\, (p(x) \to \exists y\, r(x, y) \wedge \neg \exists u\, r(f(u), z))$$

analizzando i passaggi attraverso cui sono state costruite a partire da formule atomiche. In particolare indicate per ognuna delle tre formule a quale tipo (secondo la classificazione del lemma 6.23) appartiene.

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

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

$\mathcal{L}(F)$ consiste dei simboli di costante a e b, dei simboli di funzione f (unario) e g (binario), e dei simboli di relazione p (unario), r (binario) e q (ternario).

Esercizio 6.29. Dire di che tipo devono essere i simboli a, b, p, q, f e g (se di costante, di funzione, di relazione e di quale arietà) affinché la stringa di simboli seguente sia una formula:

$$\forall x(p(g(a, f(x))) \vee q(g(x, y), f(g(a, b)))).$$

(Questo è equivalente a stabilire qual è il linguaggio della formula in questione.)

Esercizio 6.30. Spiegare perché le stringhe di simboli

$$\forall x(p(x) \rightarrow p(x, a)), \quad \exists x(q(x) \wedge r(q(a), a)) \quad \text{e} \quad g(s(x) \rightarrow t(x))$$

non sono formule di nessun linguaggio.

Esercizio 6.31. Indicare quale tra le seguenti stringhe di simboli è una formula atomica del linguaggio dell'esempio 6.28:

$$q(a); \quad p(y); \quad p(g(b)); \quad \neg r(x, a); \quad q(x, p(a), b); \quad p(g(f(a), g(x, f(x))));$$
$$q(f(a), f(f(x)), f(g(f(z), g(a, b)))); \quad r(a, r(a, a)); \quad r(a, g(a, a)); \quad g(a, g(a, a)).$$

Esercizio 6.32. Indicare quale tra le seguenti stringhe di simboli è una formula del linguaggio dell'esempio 6.28:

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

Esercizio 6.33. Indicare in quale tra i seguenti linguaggi la stringa di simboli $\exists x\, p(g(f(x)))$ è una formula:

(a) p, g e f sono simboli di funzione unari;
(b) p è un simbolo di relazione unario e g e f sono simboli di funzione unari;
(c) p e g sono simboli di relazione unari e f è un simbolo di funzione unario;
(d) p, g e f sono simboli di relazione unari.

La definizione di formula, come quella di termine, è ricorsiva e questo ci permette di ragionare *per induzione sulla complessità delle formule* (si ricordi il teorema 1.10 nel caso proposizionale).

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 $\mathcal{A}(F \rightarrow G)$;*
- *se vale $\mathcal{A}(F)$ per una formula F allora valgono anche $\mathcal{A}(\forall x\, F)$ e $\mathcal{A}(\exists x\, F)$ per ogni $x \in \mathrm{Var}$.*

Allora $\mathcal{A}(F)$ vale per ogni formula F.

Similmente è possibile dare definizioni procedendo *per ricorsione sulla complessità delle formule*. Come primo esempio di quest'ultimo procedimento consideriamo la seguente definizione, che estende quella analoga nel caso proposizionale (definizione 1.12):

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

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

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

Esercizio 6.37. ($\star$) Dimostrare per induzione sulla complessità delle formule che il grado di F è il numero di connettivi e quantificatori che compaiono in F.

4. Variabili libere e enunciati

Il ruolo della variabile x nelle formule $p(x)$ e $\forall x\, p(x)$ (ovviamente siamo in un linguaggio in cui p è un simbolo di relazione unario) è ben diverso. Infatti 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. Per catturare questa differenza diamo la seguente definizione per ricorsione sulla complessità di una formula.

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;
- se F è $\forall x\, G$ oppure $\exists x\, G$, allora nessuna occorrenza di x in F è libera;
- se F è $\forall y\, G$ oppure $\exists y\, G$ dove y è una variabile diversa da x, allora 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). Le *variabili libere* di una formula F sono quelle che hanno almeno un'occorrenza libera in F. Una formula priva di variabili libere è chiamata *enunciato* o *formula chiusa*.

Osserviamo che asserire che una variabile x non è libera nella formula F significa o che x compare solamente legata in F, oppure che x non compare affatto in F.

Esempio 6.39. Nella formula

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

le occorrenze libere delle variabili sono sottolineate. Pertanto le variabili libere di questa formula sono y, x e z e la formula non è un enunciato. Notiamo che x e y hanno sia occorrenze libere che occorrenze legate in questa formula.

L'idea della definizione 6.38 è che le variabili libere di una formula vanno interpretate in un modo che è ancora da stabilire: la loro presenza impedisce quindi di stabilire se la formula in questione sia vera o falsa. Un enunciato invece esprime direttamente una proprietà di ciò di cui si sta parlando e pertanto, una volta stabilito come interpretare i simboli di costante, funzione e relazione, risulterà essere vero o falso.

Il nostro interesse principale è rivolto agli enunciati, ma il nostro studio deve necessariamente coinvolgere anche le formule che non sono enunciati. Infatti per costruire un enunciato applicando la definizione ricorsiva della definizione 6.20 è quasi sempre necessario (salvo quando l'enunciato non contiene variabili) passare attraverso formule che non sono enunciati. Anticipando la terminologia della definizione 6.46, il motivo è che le sottoformule di enunciati non sono necessariamente enunciati. L'unica eccezione a questa situazione si avrà nel capitolo 11, dove il metodo dei tableaux sarà presentato in modo da utilizzare solamente enunciati.

Esempio 6.40. Nel linguaggio $\mathcal{L}_{\mathrm{arit}}$ dell'esempio 6.3 con i vari simboli interpretati in modo naturale, non ha senso chiedersi se la formula atomica con variabili libere $x+1=1$ sia vera o falsa. Invece gli enunciati $1+1=1$, $\forall x\, x+1=1$ e $\exists x\, x+1=1$ sono falsi i primi due e vero il terzo.

Esercizio 6.41. Nel linguaggio dell'esempio 6.28 stabilire quali sono le variabili libere nelle seguenti formule e quali di esse sono enunciati:

$$p(a);\quad p(x)\wedge\neg r(y,a);\quad \exists x\, r(y,y);\quad \forall x\, p(x)\rightarrow\exists y\,\neg q(f(x),y,f(y));$$
$$\forall x\,\exists y\, r(x,f(y))\rightarrow r(x,y);\quad \forall x(\exists y\, r(x,f(y))\rightarrow r(x,y));$$
$$\neg r(f(a),a);\quad \forall z\,(p(z)\rightarrow\exists y(\exists x\, q(x,y,z)\vee q(z,y,x)));$$
$$\forall x(p(x)\rightarrow\exists y\,\neg q(f(x),y,f(y)));\quad \forall z\,\exists u\,\exists y(q(z,u,g(u,y))\vee r(u,g(z,u)));$$
$$\forall z\,\exists x\,\exists y(q(z,u,g(u,y))\vee r(u,g(z,u)));\quad \forall z(\exists y\, q(z,u,g(u,y))\vee\exists u\, r(u,g(z,u))).$$

Fate lo stesso con la formula dell'esempio 6.28 e quelle degli esercizi 6.29, 6.31, 6.32 e 6.33.

Esercizio 6.42. Indichiamo con $\mathrm{lib}(F)$ l'insieme delle variabili libere della formula F. Dimostrare che:

- $\mathrm{lib}(\neg F)=\mathrm{lib}(F)$;
- $\mathrm{lib}(F\wedge G)=\mathrm{lib}(F\vee G)=\mathrm{lib}(F\rightarrow G)=\mathrm{lib}(F)\cup\mathrm{lib}(G)$;
- $\mathrm{lib}(\forall x\, F)=\mathrm{lib}(\exists x\, F)=\mathrm{lib}(F)\setminus\{x\}$.

Queste condizioni permettono di dare una definizione alternativa delle variabili libere di una formula, senza specificare quale occorrenze siano libere.

Alcune classi di formule hanno una certa importanza e quindi hanno dei nomi specifici per designarle. Due esempi li abbiamo già incontrati: le formule atomiche e gli enunciati.

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 formule proposizionali studiate nei capitoli precedenti di queste dispense, mentre i letterali predicativi coincidono con i letterali della definizione 3.1.

Definizione 6.44. 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, mentre l'enunciato $\exists x_1\dots\exists x_n\, F$ 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 che il corollario 7.31 mostrerà che tutte le chiusure universali di una formula F sono tra loro logicamente equivalenti (nel senso che definiremo nel capitolo 7), indichiamo con $\forall(F)$ una qualunque chiusura universale di F. Lo stesso vale per le chiusure esistenziali, ed in questo caso la notazione sarà $\exists(F)$.

5. Sottoformule

Questa sezione è quasi una ripetizione della sezione 1.3: le idee della definizione di sottoformula sono esattamente le stesse nel caso proposizionale e in quello predicativo.

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

$$\forall x(p(x) \to \exists y\, p(y) \lor r(x,y)),$$

$\exists y\, p(y) \lor r(x,y)$ è una sottoformula di F, mentre $p(x) \to \exists y\, p(y)$ non lo è. Infatti inserendo alcune delle parentesi omesse in base alla convenzione 6.25 F è

$$\forall x(p(x) \to ((\exists y\, p(y)) \lor r(x,y))).$$

In effetti $\exists y\, p(y) \lor r(x,y)$ è una delle formule utilizzate nella costruzione di F, mentre $p(x) \to \exists y\, p(y)$ non lo è.

Esercizio 6.48. Elencate tutte le sottoformule della F dell'esempio precedente (sono sette, di cui sei proprie).

Per dare una definizione precisa di sottoformula possiamo procedere per induzione sulla complessità delle formule.

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 $G \land H$, $G \lor H$ oppure $G \to H$ allora le sottoformule di F sono le sottoformule di G, le sottoformule di H e F stessa.

6. Sostituzioni in formule

Abbiamo visto come effettuare sostituzioni nei termini (definizione 6.18), e ora ci proponiamo di effettuare sostituzioni nelle formule. Nel caso delle formule atomiche il procedimento è semplice, e si basa proprio sulle sostituzioni in termini.

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

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

Quando la formula in cui vogliamo effettuare la sostituzione non è atomica (ed in particolare quando non è priva di quantificatori) occorre un po' più di cautela. Si possono infatti presentare due diversi problemi.

Il primo problema è esemplificato dal caso in cui F è $p(x) \land \exists x\, q(x)$ e la sostituzione è $\{x/a\}$, dove a è un simbolo di costante. Sostituendo sistematicamente x

con a in tutta F otterremmo $p(a) \wedge \exists a\, q(a)$, che non è neppure una formula (perché a non è una variabile). Il nostro obiettivo è in realtà asserire riguardo ad a ciò che la F asserisce riguardo a x, e quindi ottenere $p(a) \wedge \exists x\, q(x)$. A questo scopo basta stabilire cha vanno sostituite solo le occorrenze libere delle variabili nel dominio della sostituzione.

Il secondo problema è esemplificato dal caso in cui F è $\exists y\, r(x, y)$ (in questo caso l'unica occorrenza di x è libera e perciò se sostituiamo un termine al posto di x non abbiamo il problema rilevato precedentemente). Se $\exists y\, r(x, y)$ è vera per ogni x (ad esempio parliamo di numeri naturali e $r(x, y)$ è $x < y$), ci aspettiamo che ogni sostituzione in F di x con qualche termine produca una formula vera; invece se la sostituzione è $\{x/y\}$ si ottiene $\exists y\, r(y, y)$, che è falsa nell'interpretazione descritta sopra.

In questo caso la sostituzione conduce a una formula (addirittura ad un enunciato), che ha un significato completamente diverso da quello che avevamo in mente. La formula ottenuta dopo la sostituzione dovrebbe condurre ad una formula che asserisca riguardo a y ciò che F asserisce riguardo a x, ed in particolare ad una formula in cui y è libera. La situazione è analoga a quella che si incontra in analisi, con gli integrali definiti: $\int_0^1 f(x, y)\, dy$ è una funzione di x, mentre $\int_0^1 f(y, y)\, dy$ è un numero.

Questo secondo problema è più delicato e ci porta a stabilire che non tutte le sostituzioni si possano effettuare.

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 questo accade se t è un termine chiuso, ed ancora più in particolare se t è un simbolo di costante.

Se F è priva di quantificatori ogni termine è libero per la sostituzione al posto di ogni occorrenza libera di ogni variabile in F.

Esempio 6.54. Se F è $\exists y\, r(x, y) \vee \forall z\, \neg r(z, x)$, x ha due occorrenze libere in F. Il termine $f(y)$ non è libero per la sostituzione al posto della prima occorrenza libera di x, ma è libero per la sostituzione al posto della seconda occorrenza libera di x. Al contrario il termine $g(z, w)$ è libero per la sostituzione al posto della prima occorrenza libera di x, ma non è libero per la sostituzione al posto della seconda occorrenza libera di x. I termini w, $f(w)$ e $g(v, a)$ sono liberi per la sostituzione al posto di entrambe le occorrenze libere di x in F.

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 (dimostrato nella sezione 7.4 di queste dispense) che collega la nozione sintattica di sostituzione con la semantica che introdurremo nel prossimo capitolo.

Esempio 6.56. Qualunque termine t è libero per la sostituzione all'unica occorrenza libera di x nella formula F

$$\exists y\, \forall x (p(x) \to r(x, y)) \to p(x).$$

Perciò $\{x/t\}$ è una sostituzione ammissibile in F e $F\{x/t\}$ è

$$\exists y\,\forall x(p(x) \to r(x,y)) \to p(t).$$

Esempio 6.57. Se F è $\forall z\, r(x,z) \wedge \exists y\, r(x, f(y)) \to \neg\exists x\, r(x,x)$ 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

$$\forall z\, r(c,z) \wedge \exists y\, r(c, f(y)) \to \neg\exists x\, r(x,x),$$
$$\forall z\, r(f(x),z) \wedge \exists y\, r(f(x), f(y)) \to \neg\exists x\, r(x,x),$$
$$\forall z\, r(w,z) \wedge \exists y\, r(w, f(y)) \to \neg\exists x\, r(x,x).$$

Esercizio 6.58. In ognuna delle situazioni seguenti stabilire se la sostituzione $\{x/t\}$ è ammissibile in F. Se la risposta è positiva trovare $F\{x/t\}$.

- t è y, F è $\exists y\,\forall x\, r(x,y) \vee \exists z\, r(x,z)$;
- t è $g(y,a)$, F è $\forall x\,\exists y\, r(x,y) \to q(x, f(x,y), y)$;
- t è $g(y,a)$, F è $\exists y(r(x,y) \to q(x, f(x), y))$;
- t contiene solo variabili che non compaiono (né libere, né legate) in F;
- t è $f(x)$, F qualsiasi.

Anche le occorrenze legate di una variabile possono venire sostituite, ma solo con un'altra variabile (perché altrimenti si otterrebbe una stringa di simboli che non è una formula).

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 stessa funzione di z.

Esempio 6.60. Se F è la formula $r(x,w) \wedge \forall z\,\exists x\, r(x,z)$ allora $F_x(y)$ è definita ed è $r(x,w) \wedge \forall z\,\exists y\, r(y,z)$. Notate che in questo caso non sono definite né $F_x(z)$ né $F_x(w)$.

Una delle ragioni per cui le varianti sono utili è che passando ad una variante è possibile rendere qualsiasi sostituzione ammissibile.

Esempio 6.61. Se F è $\exists y\, r(x,y)$ la sostituzione $\{x/f(y)\}$ non è ammissibile. Dato che $F_y(z)$ è $\exists z\, r(x,z)$, la medesima sostituzione è ammissibile in $F_y(z)$ e $F_y(z)\{x/f(y)\}$ è $\exists z\, r(f(y),z)$. Quest'ultima formula asserisce di $f(y)$ ciò che F asserisce di x.

Esercizio 6.62. Sia F la formula $\forall x\,\forall y\,\exists z\, q(v, g(x,y), g(z,w))$. Applicate ripetutamente l'operazione di variante a partire da F fino a trovare una formula per cui la sostituzione $\{v/h(x,y,z)\}$ sia ammissibile.

7. Linguaggi con uguaglianza

Nei linguaggi degli esempi 6.3, 6.4, 6.5, 6.6 e 6.8 è stato naturale inserire un simbolo di relazione binario che abbiamo denotato con $=$, e il cui significato implicito è "essere uguale a". Dal punto di vista sintattico, $=$ è un simbolo di relazione binario che viene considerato esattamente come un qualsiasi altro simbolo di relazione binario. Quando però passeremo allo studio della semantica (vedere la sezione 7.5), al simbolo $=$ verrà attribuito un ruolo particolare: ad esempio la nostra comprensione del significato dell'uguaglianza ci dice che è equivalente scrivere $x = y$ oppure $y = x$.

Il simbolo $=$ ha quindi un ruolo del tutto particolare, e d'ora in poi lo useremo solo quando svolge questo ruolo. In particolare $=$ sarà sempre un simbolo di relazione binario. Ci uniformiamo inoltre (come abbiamo già fatto nel paragrafo precedente) al modo usuale di scrivere formule che contengono $=$.

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 esprimono alcune delle proprietà fondamentali dell'uguaglianza. Il ruolo di questo insieme di enunciati verrà chiarito dal lemma 7.67 e dal teorema 10.33.

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) $\forall x_1 \dots \forall x_n\, \forall y_1 \dots \forall y_n (x_1 = y_1 \wedge \cdots \wedge x_n = y_n \to f(x_1, \dots, x_n) = f(y_1, \dots, y_n))$ per tutti i simboli di funzione f di $\mathcal{L}$, dove n è l'arietà di f.
(e5) $\forall x_1 \dots \forall x_n\, \forall y_1 \dots \forall y_n (x_1 = y_1 \wedge \cdots \wedge x_n = y_n \wedge p(x_1, \dots, x_n) \to p(y_1, \dots, y_n))$ per tutti i simboli di relazione p di $\mathcal{L}$, dove n è l'arietà di p.

Notate che (e1), (e2) e (e3) sono singoli enunciati che appartengono a $\mathsf{Eq}_{\mathcal{L}}$ per ogni linguaggio $\mathcal{L}$. Invece (e4) e (e5) rappresentano insiemi di enunciati (sono chiamati schemi di enunciati). A seconda di quanti simboli di funzione e di relazione appartengono a $\mathcal{L}$, (e4) può essere vuoto (per linguaggi privi di simboli di funzione), ed entrambi possono essere finiti o infiniti.

Dal punto di vista della sintassi non molto altro può venire detto, e rimandiamo alla sezione 7.5 per la semantica dei linguaggi con uguaglianza, alla sezione 9.3 per l'uso dell'uguaglianza nelle traduzioni dal linguaggio naturale, alla sezione 10.3 per il ruolo di $\mathsf{Eq}_{\mathcal{L}}$ nello studio della logica con uguaglianza, alla sezione 11.8 per i tableaux nella logica con uguaglianza, e alla sezione 12.5 per le deduzioni naturali nella logica con uguaglianza.

CAPITOLO 7

Semantica della logica predicativa

In questo capitolo svilupperemo la semantica della logica predicativa in analogia a quanto fatto nel capitolo 2 per la semantica della logica proposizionale. La nozione di interpretazione è piuttosto diversa nei due contesti, ma —una volta che essa sia stata stabilita— molte definizioni (conseguenza ed equivalenza logica, validità, ecc.) saranno identiche a quelle del capitolo 2.

1. Interpretazioni e soddisfazione

Per interpretare una formula predicativa dobbiamo prima di tutto interpretare gli elementi del suo linguaggio. Ciò avverrà in relazione ad un dominio, o universo, che consiste degli oggetti di cui vogliamo parlare. I simboli di costante, funzione e relazione, vanno interpretati con riferimento a questo dominio.

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 formule. Iniziamo con i termini chiusi.

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 è sufficiente a stabilire quale elemento del dominio associare al termine. A questo scopo affianchiamo all'interpretazione un modo di interpretare le variabili.

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

ESERCIZIO 7.4. Dimostrate che se t è un termine chiuso e σ uno stato di un'interpretazione I, si ha $\sigma(t) = t^I$.

ESEMPIO 7.5. Sia $\mathcal{L}$ 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 $\mathcal{L}$ ponendo

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

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 σ del termine $f(x)$ è $\sigma(f(x)) = f^I(\sigma(x)) = f^I(0) = 1$.

Il prossimo lemma (che ricorda il lemma 2.8) asserisce che il valore assunto da un termine non dipende né dall'interpretazione dei simboli di costante e di funzione che non occorrono nel termine, né dall'interpretazione dei simboli di relazione, né dal valore dello stato su variabili che non occorrono nel termine.

Lemma 7.6. *Sia t un termine di un linguaggio $\mathcal{L}$, e siano I e I' interpretazioni per $\mathcal{L}$ che hanno lo stesso dominio e coincidono sulle interpretazioni dei simboli di costante e di funzione che occorrono in t. Siano inoltre σ e σ' stati rispettivamente di I e I' che coincidono sulle variabili che occorrono in t. Allora $\sigma(t) = \sigma'(t)$.*

Dimostrazione. La dimostrazione è per induzione sulla complessità del termine t (teorema 6.17). Se t è una variabile o una costante allora per ipotesi $\sigma(t) = \sigma'(t)$. Se $t = f(t_1, \dots, t_n)$ allora per ipotesi si ha che le funzioni f^I e $f^{I'}$ coincidono e, usando l'ipotesi induttiva, abbiamo $\sigma(t) = f^I(\sigma(t_1), \dots, \sigma(t_n)) = f^{I'}(\sigma'(t_1), \dots, \sigma'(t_n)) = \sigma'(t)$. □

Per arrivare a interpretare le formule è opportuno introdurre la seguente notazione per gli stati.

Notazione 7.7. Se σ è uno stato dell'interpretazione I, x una variabile e $d \in D^I$ indichiamo con $\sigma[x/d]$ (da leggersi σ *perturbato mandando* x *in* d) lo stato che coincide con σ su tutte le variabili diverse da x, e assegna a x il valore d.

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 $I, \sigma \vDash H$;
- $I, \sigma \vDash G \vee H$ se e solo se $I, \sigma \vDash G$ oppure $I, \sigma \vDash H$;
- $I, \sigma \vDash G \rightarrow H$ se e solo se $I, \sigma \nvDash G$ oppure $I, \sigma \vDash H$;
- $I, \sigma \vDash \forall x\, G$ se e solo se per ogni $d \in D$ si ha che $I, \sigma[x/d] \vDash G$;
- $I, \sigma \vDash \exists x\, G$ se e solo se esiste $d_0 \in D$ tale che $I, \sigma[x/d_0] \vDash G$.

Definizione 7.9. Diciamo che I *soddisfa* F, e scriviamo $I \vDash F$ se $I, \sigma \vDash F$ 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 $I, \sigma \vDash T$, 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 proposizionali e quelle predicative, la seconda, terza, quarta e quinta clausola della definizione 7.8 ricalcano le clausole analoghe della definizione 2.3. La prima clausola usa le interpretazioni dei termini e l'interpretazione dei simboli predicativi per sostituire la valutazione che attribuiva valori di verità alle lettere proposizionali, mentre completamente nuove sono solo le ultime due clausole che riguardano i quantificatori.

Il simbolo $\models$ è stato usato nel caso proposizionale (e lo sarà anche nel caso predicativo) per indicare la conseguenza logica: per distinguere tra i due significati attribuiti a questo simbolo si veda la nota 7.30.

Esempio 7.11. Siano $\mathcal{L}$, I e σ come nell'esempio 7.5. Sia F la formula

$$(q(f(y)) \to \neg p(c)) \vee \exists z(r(x,z) \wedge p(z) \wedge r(y,z)).$$

Per verificare se $I, \sigma \models F$ iniziamo con lo stabilire se $I, \sigma \models q(f(y)) \to \neg p(c)$. A questo scopo calcoliamo $\sigma(f(y)) = f^I(\sigma(y)) = f^I(1) = 2$; dato che $2 \in q^I$ si ha $I, \sigma \models q(f(y))$ e dobbiamo verificare se $I, \sigma \models \neg p(c)$: dato che $\sigma(c) = c^I = 1$ e $1 \in p^I$ si ha $I, \sigma \models p(c)$ e quindi $I, \sigma \not\models \neg p(c)$. Perciò $I, \sigma \not\models q(f(y)) \to \neg p(c)$ e l'unica possibilità che $I, \sigma \models F$ è che $I, \sigma \models \exists z(r(x,z) \wedge p(z) \wedge r(y,z))$. Per verificare se ciò avviene dobbiamo cercare uno stato della forma $\sigma[z/d]$ con $d \in D^I$ tale che $I, \sigma[z/d] \models r(x,z) \wedge p(z) \wedge r(y,z)$. Si ha che $I, \sigma[z/0] \models r(x,z)$, ma $I, \sigma[z/0] \not\models p(z)$ e $I, \sigma[z/0] \not\models r(y,z)$, così che 0 non è l'elemento adatto. Dato che $I, \sigma[z/1] \not\models r(x,z)$ neppure 1 va bene. Invece $I, \sigma[z/2] \models r(x,z)$, $I, \sigma[z/2] \models p(z)$ e $I, \sigma[z/2] \models r(y,z)$, così che $I, \sigma[z/2] \models r(x,z) \wedge p(z) \wedge r(y,z)$. Perciò $I, \sigma \models \exists z(r(x,z) \wedge p(z) \wedge r(y,z))$ e quindi $I, \sigma \models F$.

Nell'esempio precedente il valore assunto da σ su variabili (come w) che non occorrono in F è del tutto irrilevante. Anche il fatto che $\sigma(z) = 2$ non è mai stato utilizzato, malgrado z occorra in F. Ciò avviene perché z non è libera in F e pertanto abbiamo utilizzato il valore di uno stato su z solo per stati della forma $\sigma[z/d]$. Queste osservazioni conducono ad un lemma che corrisponde ai lemmi 7.6 (per quanto riguarda i termini) e 2.8 (nel caso proposizionale) e che asserisce che la soddisfazione di una formula non dipende né dall'interpretazione dei simboli di costante, di funzione e di relazione che non occorrono nella formula né dal valore dello stato su variabili che non sono libere nella formula.

Lemma 7.12. *Sia F una formula di un linguaggio $\mathcal{L}$, e siano I e I' interpretazioni per $\mathcal{L}$ che hanno lo stesso dominio e coincidono sui simboli di costante, di funzione e di relazione che occorrono in F. Siano inoltre σ e σ' stati rispettivamente di I e I' che coincidono sulle variabili libere di F. Allora $I, \sigma \models F$ se e solo se $I', \sigma' \models F$.*

Dimostrazione. La dimostrazione è per induzione sulla complessità della formula F (teorema 6.34).

Se F è una formula atomica basta applicare la definizione 7.8 e il lemma 7.6.

Se F è del tipo $\neg G$, $G \wedge H$, $G \vee H$ oppure $G \to H$ basta applicare la definizione 7.8 e l'ipotesi induttiva.

Se F è del tipo $\forall x\, G$ e si ha $I, \sigma \models F$ allora per ogni $d \in D^I$ $I, \sigma[x/d] \models G$. Per dimostrare che $I', \sigma' \models F$ fissiamo $d \in D^{I'} = D^I$ e consideriamo lo stato $\sigma'[x/d]$: quest'ultimo coincide con $\sigma[x/d]$ sulle variabili libere di G (che, per l'esercizio 6.42, sono le variabili libere di F più eventualmente x) e per ipotesi induttiva si ha $I', \sigma'[x/d] \models G$. Quindi $I', \sigma' \models F$. In maniera simmetrica si dimostra che da $I', \sigma' \models F$ segue $I, \sigma \models F$.

Se F è del tipo $\exists x\, G$ il ragionamento è simile al caso del quantificatore universale. $\square$

Corollario 7.13. *Se F è un enunciato di $\mathcal{L}$, I un'interpretazione per $\mathcal{L}$ e σ e σ' due stati di I allora $I, \sigma \vDash F$ se e solo se $I, \sigma' \vDash F$. Quindi $I \vDash F$ se e solo se $I, \sigma \vDash F$ per qualche stato σ di I.*

Il corollario precedente asserisce che 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 $I \vDash \forall x\, F$, dove F è una formula in cui x è l'unica variabile libera, dobbiamo considerare per ogni $d \in D^I$ se $I, \sigma[x/d] \vDash F$; in questo caso però per il lemma 7.12 possiamo scegliere σ arbitrariamente: l'unica proprietà di $\sigma[x/d]$ che ci interessa è che manda x in d.

Esercizio 7.14. Siano $\mathcal{L}$, I e σ come nell'esempio 7.5. Sia J l'interpretazione per $\mathcal{L}$ definita da

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

Sia G l'enunciato

$$\forall x(\exists y\, r(x,y) \to \neg q(x) \vee \forall z(p(f(z)) \to r(z,x))).$$

Stabilite se $I \vDash G$, se $J \vDash G$ e se $J, \sigma \vDash F$, dove F è la formula dell'esempio 7.11 (notate che σ è anche uno stato di J).

Esempio 7.15. Un grafo (nozione definita nel corso di Matematica Discreta) $G = (V,E)$ può essere visto come un'interpretazione I per il linguaggio $\mathcal{L}_{\mathrm{gr}}$ dell'esempio 6.8. Il dominio di I è l'insieme V dei vertici del grafo, $=^I = \{\,(x,y) : x = y\,\}$ e infine $E^I = \{\,(i,j) : ij \in E\,\}$ rappresenta l'insieme dei lati del grafo.

Dato che abbiamo a che fare con un grafo avremo

$$I \vDash \forall x\, \forall y(E(x,y) \to x \neq y)$$

e se il grafo è non orientato varrà anche

$$I \vDash \forall x\, \forall y(E(x,y) \to E(y,x)).$$

In realtà $G = (V,E)$ è un grafo non orientato precisamente quando l'interpretazione I definita sopra soddisfa questi due enunciati.

Inoltre G è aciclico se e soltanto se $I \vDash T$ dove T è l'insieme infinito di enunciati F_n per $n \geq 3$ e F_n è

$$\neg\exists x_1\, \exists x_2 \dots \exists x_n \Big(\bigwedge_{1 \leq i < n} E(x_i, x_{i+1}) \wedge E(x_n, x_1) \wedge \bigwedge_{1 \leq i < j \leq n} x_i \neq x_j \Big).$$

Infatti F_n è soddisfatto da I se e solo se G non contiene cicli elementari di lunghezza n.

Esercizio 7.16. Per ognuno degli enunciati seguenti nel linguaggio dell'esempio 7.5 definite un'interpretazione che lo renda vero ed una che lo renda falso:

$$\forall x\, \exists y\, r(x,y) \wedge \neg\forall x\, p(x); \quad \forall x\, p(x) \vee \forall x\, \neg p(x); \quad p(c) \to \neg p(c); \quad p(c) \to p(a);$$
$$(\exists x\, p(x) \to p(c)) \wedge \neg p(c); \quad (\exists x\, p(x) \to p(c)) \wedge p(f(c)); \quad \exists x\, \neg q(x) \wedge \forall x\, q(f(x))$$
$$\forall x(p(x) \vee q(x)) \to \forall x\, p(x) \vee \forall x\, q(x); \quad \forall x\, \exists y\, r(x,y) \to \exists y\, \forall x\, r(x,y).$$

Esercizio 7.17. Per ognuna delle formule seguenti F, trovate un'interpretazione I per il linguaggio dell'esempio 7.5 e due stati σ_1 e σ_2 di I tali che $I, \sigma_1 \vDash F$ e $I, \sigma_2 \nvDash F$.

$$p(x); \quad \exists y\, r(x,y); \quad r(x,y) \to r(y,x); \quad \exists x\, \neg r(x, f(x)) \wedge r(x, f(x)).$$

Esercizio 7.18. Siano F e G gli enunciati $\forall x(r(x,a) \to r(x,b))$ e $\exists x(r(x,a) \land r(x,b))$. Definite interpretazioni I_1, I_2, I_3 e I_4 per il linguaggio $\mathcal{L}(F,G)$, tutte con dominio $\{0,1\}$, tali che:

$$\begin{array}{lcl} I_1 \models F & \text{e} & I_1 \models G; \\ I_2 \models F & \text{e} & I_2 \not\models G; \\ I_3 \not\models F & \text{e} & I_3 \models G; \\ I_4 \not\models F & \text{e} & I_4 \not\models G. \end{array}$$

Esercizio 7.19. Siano F e G gli enunciati

$$\forall x(p(x) \to \neg r(x,x)) \quad \text{e} \quad \neg\exists x(p(x) \land \forall y\, r(y,x)).$$

Definite interpretazioni I_1, I_2 e I_3 per $\mathcal{L}(F,G)$, tutte con dominio $\{0,1\}$, tali che:

$$\begin{array}{lcl} I_1 \models F & \text{e} & I_1 \models G; \\ I_2 \not\models F & \text{e} & I_2 \models G; \\ I_3 \not\models F & \text{e} & I_3 \not\models G. \end{array}$$

(Si veda l'esempio 7.32.)

Esercizio 7.20. Sia $\mathcal{L} = \{r\}$ dove r è un simbolo di relazione binario. Siano I e J le interpretazioni per $\mathcal{L}$ definite da: $D^I = \{0\}$, $r^I = \{(0,0)\}$, $D^J = \{1,2\}$, $r^J = \{(1,2),(2,1)\}$. Trovate un enunciato vero in I e falso in J. Trovate un enunciato vero in J e falso in I.

Esempio 7.21. Sia H l'enunciato

$$\forall x\, \forall y(r(x,y) \to \neg r(y,x)) \land \forall x\, \neg r(x,x) \land \forall x\, \exists y\, r(x,y).$$

Sia I un'interpretazione che soddisfa H. Se I avesse cardinalità 1 (cioè D^I avesse un solo elemento, chiamiamolo 0), dato che $I \models \forall x\, \exists y\, r(x,y)$, dovremmo avere $(0,0) \in r^I$, contro $I \models \forall x\, \neg r(x,x)$. Se I avesse cardinalità 2 (cioè D^I avesse esattamente due elementi, diciamo 0 e 1), ragionando in modo analogo avremmo $(0,1) \in r^I$ e $(1,0) \in r^I$, contro $I \models \forall x\, \forall y(r(x,y) \to \neg r(y,x))$. Abbiamo quindi dimostrato che H non ha modelli di cardinalità 1 o 2.

Esercizio 7.22. Sia H l'enunciato dell'esempio 7.21. Definite un modello di cardinalità 3 per H.

Esercizio 7.23. Siano F e G le formule

$$\forall x(p(x,x) \to p(f(x),x)) \quad \text{e} \quad \exists x\, \neg p(x,x) \land \forall x\, p(x,f(x)).$$

Definite interpretazioni I_1, I_2 e I_3 per $\mathcal{L}(F,G)$, tutte con dominio $D = \{0,1\}$, tali che:

$$\begin{array}{lcl} I_1 \models F & \text{e} & I_1 \models G; \\ I_2 \models F & \text{e} & I_2 \not\models G; \\ I_3 \not\models F & \text{e} & I_3 \not\models G. \end{array}$$

Definite un'interpretazione J per $\mathcal{L}(F,G)$ con dominio $D' = \{0,1,2\}$ tale che $J \not\models F$ e $J \models G$.

Lemma 7.24. *Sia F una formula e I un'interpretazione per $\mathcal{L}(F)$. Se G è una chiusura universale di F, allora $I \models G$ se e solo se $I \models F$ (cioè $I, \sigma \models F$ per ogni stato σ di I). Se H è una chiusura esistenziale di F allora $I \models H$ se e solo se esiste uno stato σ di I tale che $I, \sigma \models F$.*

Dimostrazione. Per dimostrare la prima parte del lemma è sufficiente mostrare che $I \vDash \forall x\, F$ se e solo se $I \vDash F$ (si può poi ripetere questo passaggio fino a ottenere G). Si ha

$$\begin{aligned} I \vDash \forall x\, F \text{ se e solo se } & I, \sigma \vDash \forall x\, F \text{ per ogni stato } \sigma \\ \text{se e solo se } & I, \sigma[x/d] \vDash F \text{ per ogni stato } \sigma \text{ e ogni } d \in D^I \\ \text{se e solo se } & I, \tau \vDash F \text{ per ogni stato } \tau \\ \text{se e solo se } & I \vDash F, \end{aligned}$$

dove l'equivalenza tra la seconda e la terza riga è basata sul fatto che ogni stato τ è della forma $\sigma[x/d]$ per opportuni σ e d (ad esempio si può prendere $\sigma = \tau$ e $d = \tau(x)$).

Per dimostrare la seconda parte del lemma osserviamo che esiste uno stato σ tale che $I, \sigma \vDash F$ se e solo se esiste uno stato τ tale che $I, \tau \vDash \exists x\, F$ (è immediato dalla clausola della definizione 7.8 che riguarda le quantificazioni esistenziali). Ripetendo questo passaggio si ottiene che esiste uno stato σ tale che $I, \sigma \vDash F$ se e solo se esiste uno stato τ tale che $I, \tau \vDash H$. Dato che H è un enunciato, quest'ultima condizione è equivalente a $I \vDash H$. □

Esercizio 7.25. ($\star$) Siano I un'interpretazione, σ uno stato di I e F una formula. Una sola direzione dell'equivalenza "$I, \sigma \vDash F$ se e solo se $I, \sigma \vDash \forall x\, F$" è vera. Quale? Perché?

Esercizio 7.26. ($\star$) Dimostrate che l'enunciato

$$F = \neg p(a) \wedge \forall x (p(f(x)) \to p(x)) \wedge \exists x\, p(x)$$

non è vero in nessuna interpretazione I in cui ogni elemento del dominio è l'interpretazione di un termine chiuso di $\mathcal{L}(F)$ (cioè tale che per ogni $d \in D^I$ esiste un termine chiuso t con $d = t^I$).

Considerate l'interpretazione I che ha come dominio l'insieme dei termini chiusi di $\mathcal{L}(F)$ ($D^I = \{\, f^{(n)}(a) : n \in \mathbb{N} \,\}$) e che interpreta i simboli del linguaggio nel modo seguente: $a^I = a$, $f^I(a) = a, f^I(f^{(n)}(a)) = f^{(n+1)}(a)$, per ogni $n \geq 1$, $p^I = \{\, f^{(n)}(a) : n > 0 \,\}$. Dimostrate che $I \vDash F$. Perché questo risultato non è in contraddizione con quanto dimostrato prima?

2. Equivalenza e conseguenza logica

Come preannunciato, le definizioni di equivalenza e conseguenza logica sono analoghe alle corrispondenti definizioni nel caso proposizionale (definizioni 2.12, 2.15 e 2.27)

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 7.29. 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 7.28 e 7.29). È sempre possibile capire con quale delle due nozioni si ha a che fare semplicemente esaminando ciò che compare a sinistra di $\models$ (a destra c'è sempre una formula): nel primo caso un'interpretazione eventualmente affiancata da uno stato, nel secondo caso una formula o un insieme di formule.

Molte proprietà dell'equivalenza e della conseguenza logica viste nel caso proposizionale valgono anche nel caso predicativo, spesso con le stesse dimostrazioni: ad esempio l'equivalenza logica è una relazione di equivalenza e la conseguenza logica è riflessiva e transitiva.

Il Lemma 7.24 ha la seguente conseguenza espressa in termini di equivalenza logica.

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.

Sia dunque I un'interpretazione tale che $I \models F$ (dato che F è un enunciato possiamo non utilizzare lo stato): vogliamo dimostrare che $I \models G$. Procediamo per assurdo e supponiamo $I \not\models G$, cioè $I \models \exists x(p(x) \wedge \forall y\, r(y,x))$. Esiste dunque $d_0 \in D^I$ tale che $d_0 \in p^I$ e per ogni $d \in D^I$, $(d, d_0) \in r^I$. In particolare $(d_0, d_0) \in r^I$ e quindi, per uno stato σ qualunque, $I, \sigma[x/d_0] \not\models p(x) \to \neg r(x,x)$. Dato che $I \models F$, questo non è possibile.

L'interpretazione I_2 costruita nell'esercizio 7.19 mostra che $G \not\models F$ (e quindi $F \not\equiv G$).

L'esempio precedente evidenzia nuovamente come per mostrare che $F \not\models G$ sia sufficiente trovare **una** interpretazione in cui F è vera, ma G è falsa. Invece per mostrare che $F \models G$ è necessario considerare **tutte** le interpretazioni del linguaggio in considerazione (e quindi bisogna ragionare in modo più astratto).

Esercizio 7.33. Dimostrate:

$$\forall x\, \forall y\, F \equiv \forall y\, \forall x\, F;$$
$$\exists x\, \exists y\, F \equiv \exists y\, \exists x\, F;$$
$$\exists y\, \forall x\, F \models \forall x\, \exists y\, F;$$
$$\forall x\, \exists y\, r(x,y) \not\models \exists y\, \forall x\, r(x,y).$$

Nota 7.34. Notate che l'ultima parte dell'esercizio precedente mostra che $\forall x\, \exists y\, F \models \exists y\, \forall x\, F$ non vale in generale. Quindi, salvo che per formule F particolari (ad esempio quelle in cui compare libera al più una delle variabili x e y), $\forall x\, \exists y\, F \not\equiv \exists y\, \forall x\, F$ e non è possibile scambiare l'ordine dei quantificatori esistenziali e universali. Esempi che possono essere utili a ricordare questo fatto sono i seguenti:

- per ogni numero ne esiste uno più grande, ma non esiste un numero più grande di ogni altro numero;
- ogni persona ha una madre, ma non esiste una madre di tutte le persone.

Esercizio 7.35. Dimostrate:

$$p(c) \wedge \neg p(f(c)) \models \neg \forall x(p(x) \to p(f(x)));$$
$$\forall y\big(p(y) \to \neg q(f(y))\big), \exists x\big(q(x) \wedge p(f(x))\big) \models \exists x\big(q(x) \wedge \neg q(f(f(x)))\big);$$
$$\exists x(p(x) \wedge q(f(x))) \not\models \exists x(p(x) \wedge q(x));$$
$$\exists x(p(x) \wedge q(f(x))) \models \exists x\, p(x) \wedge \exists x\, q(x);$$
$$\forall x(\neg p(x) \vee \exists y\, q(x,y)), \exists z\, p(f(z)) \models \neg \forall y\, \forall z\, \neg q(f(z), y).$$

Esercizio 7.36. Stabilite se:

$$\forall x\,(\exists y\, r(y,x) \to \forall y\, r(x,y)) \land \neg r(a,a) \models \neg r(f(a),a);$$
$$\forall x\,(\exists y\, r(y,x) \to \forall y\, r(x,y)) \land r(a,a) \models r(f(a),a).$$

I prossimi due esercizi chiedono di provare affermazioni che sono identiche ad alcune proprietà dimostrate nel caso proposizionale. Le dimostrazioni nel caso predicativo sono sostanzialmente quelle già viste.

Esercizio 7.37. Dimostrate che due formule F e G sono logicamente equivalenti se e solo se $F \models G$ e $G \models F$ (vedere il lemma 2.19).

Esercizio 7.38. Verificate che i lemmi 2.20, 2.24 e 2.30 e l'esercizio 2.31 valgono anche se le formule coinvolte sono formule predicative.

Diverso è il caso del lemma 2.22 sulla sostituzione di sottoformule, la cui dimostrazione procedeva per induzione sulla complessità delle formule: ora è necessario considerare anche il caso dei quantificatori utilizzando il teorema 6.34.

Lemma 7.39. *Se F è una sottoformula di una formula H e $F \equiv G$ allora $H \equiv H'$ dove H' è la formula ottenuta da H rimpiazzando la sottoformula F con G.*

Dimostrazione. La dimostrazione è per induzione sulla complessità delle formule H di cui F è sottoformula.

Quando H è F oppure H non è F ed è una negazione, una congiunzione, una disgiunzione o un'implicazione si può ripetere esattamente la dimostrazione del lemma 2.22.

Se H non è F ed è della forma $\forall x\, H_0$ oppure $\exists x\, H_0$, per ipotesi induttiva si ha $H_0 \equiv H_0'$. Per ogni interpretazione I, stato σ e $d \in D^I$ si ha che $I, \sigma[x/d] \models H_0$ se e solo se $I, \sigma[x/d] \models H_0'$. Questo significa che $H \equiv H'$. □

Esempio 7.40. Combinando i lemmi 2.20, 2.24 (validi anche nel caso predicativo per l'esercizio 7.38) e 7.39 è immediato dimostrare che

$$F \land \exists x\, G \to \neg(\neg F \lor \forall y\, H) \equiv \neg\exists x\, G \lor \neg F \lor (\neg\forall y\, H \land F).$$

Lemma 7.41. *Se x non è una variabile libera della formula F allora*

$$F \equiv \forall x\, F \equiv \exists x\, F.$$

Dimostrazione. Immediata dalla definizione 7.8 e dal lemma 7.12. □

Nota 7.42. Per F arbitraria non è vero né che $F \models \forall x\, F$ (si veda l'esercizio 7.25) né che $\exists x\, F \models F$.

3. Validità e soddisfacibilità

Le definizioni di validità e soddisfacibilità sono analoghe alle corrispondenti definizioni nel caso proposizionale (definizioni 2.33 e 2.42).

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 $I, \sigma \not\models F$.

Se T è un insieme di formule diciamo che

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

Come nel caso proposizionale sia per singole formule che per insiemi di formule essere insoddisfacibile è equivalente a non essere soddisfacibile.

Esempio 7.44. La formula $p(a) \vee \neg p(a)$ è valida. La formula $p(a) \wedge \neg p(a)$ è insoddisfacibile. La formula $p(a)$ è soddisfacibile ma non valida. $\forall x\, p(x) \rightarrow p(a)$ e $p(a) \rightarrow \exists x\, p(x)$ sono valide, $\exists x\, p(x) \wedge \forall x\, \neg p(x)$ è insoddisfacibile. $p(a) \wedge \exists x\, \neg p(x)$ è soddisfacibile ma non valida.

Più in generale, per ogni F 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.

Nota 7.45. Come già nel caso proposizionale un insieme di formule è valido se e solo se tutti i suoi elementi sono validi. La proprietà analoga non è vera però per soddisfacibilità e insoddisfacibilità.

Esercizio 7.46. Dimostrate che ogni formula atomica è soddisfacibile e non valida.

Esempio 7.47. Verifichiamo che l'enunciato

$$\exists x\, p(x) \wedge \forall x(p(x) \rightarrow \forall y\, q(x, y)) \rightarrow \exists x\, q(x, x).$$

è valido.

A questo scopo consideriamo una qualunque interpretazione I per il linguaggio $\{p, q\}$ con p simbolo di relazione unario e q simbolo di relazione binario. Se $I \models \exists x\, p(x) \wedge \forall x(p(x) \rightarrow \forall y\, q(x, y))$ esiste $d_0 \in D^I$ tale che $d_0 \in p^I$ e per qualunque stato σ di I si ha $I, \sigma[x/d_0] \models p(x) \rightarrow \forall y\, q(x, y)$. Questi due fatti implicano che per qualunque $d \in D^I$ si ha $(d_0, d) \in q^I$ e quindi in particolare che $(d_0, d_0) \in q^I$. Perciò $I \models \exists x\, q(x, x)$.

Abbiamo quindi verificato che se $I \models \exists x\, p(x) \wedge \forall x(p(x) \rightarrow \forall y\, q(x, y))$ allora $I \models \exists x\, q(x, x)$, cioè che per qualunque I si ha

$$I \models \exists x\, p(x) \wedge \forall x(p(x) \rightarrow \forall y\, q(x, y)) \rightarrow \exists x\, q(x, x).$$

Molte proprietà viste nel caso proposizionale si trasferiscono a quello predicativo senza alcuna difficoltà. Ad esempio l'enunciato del seguente teorema è uguale a quello del teorema 2.37 (ma ora stiamo parlando di formule predicative, e non proposizionali).

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:

$$\forall x(p(x) \rightarrow q(f(x))) \wedge \exists x\, p(x) \rightarrow \exists x\, q(x)$$

$$\exists x\, \forall y\, r(x, y) \rightarrow \forall y\, \exists x\, r(x, y);$$

$$(\star)\ \exists x(p(f(x)) \rightarrow p(x));$$

$$(\star)\ \neg\exists x\, \forall y((r(x, y) \rightarrow \neg r(y, y)) \wedge (\neg r(y, y) \rightarrow r(x, y))).$$

Esercizio 7.51. Dimostrate che l'enunciato
$$\forall x\big(p(x) \to (\neg q(f(x), x) \land \forall y\, q(x, y))\big) \land \exists x\big(p(x) \land p(f(x))\big)$$
è insoddisfacibile.

Esercizio 7.52. ($\star$) Dimostrate che l'insieme
$$T = \{\exists x\, \neg p(x)\} \cup \{\, p(z) \;:\; z \in \mathrm{Var}\,\}$$
(ricordate che Var è l'insieme di tutte le variabili) è soddisfacibile.

Esercizio 7.53. Dimostrate che l'enunciato
$$\forall x\, (\forall y(r(x, y) \to p(y)) \land \exists z \neg p(z) \to \exists z \neg r(x, z))$$
è valido.

Esercizio 7.54. Siano F e G gli enunciati $\forall x(p(x) \to \neg p(f(x)))$ e $\exists x\, p(f(x))$. Per ognuno dei quattro insiemi $\{F, G\}$, $\{F, \neg G\}$, $\{\neg F, G\}$ e $\{\neg F, \neg G\}$ costruite se possibile un'interpretazione con dominio $\mathbb{N}$. Nell'unico caso in cui ciò non è possibile dimostrate che l'insieme è insoddisfacibile.

Esercizio 7.55. L'enunciato $\forall y\, \exists x(q(x, y) \to q(y, y))$ è valido? Se la risposta è positiva, dimostratelo, se la risposta è negativa, definite un'interpretazione in cui esso non è soddisfatto.

Esercizio 7.56. Considerate il linguaggio $\mathcal{L} = \{p, r, a\}$, dove p è un simbolo di relazione unario, r un simbolo di relazione binario e a un simbolo di costante. Sia F l'enunciato
$$\big(\exists x\, \exists y\, r(x, y) \land \forall x(r(x, x) \to p(x))\big) \to p(a).$$
(a) Dimostrate che F è vero in ogni interpretazione il cui dominio ha un solo elemento.
(b) F è valido?

Esercizio 7.57. ($\star$) Siano F una formula, a un simbolo di costante che non compare in F e x una variabile.
(a) Dimostrate che se $F\{x/a\}$ è valida allora $\forall x\, F$ è valida.
(b) È vero che per ogni interpretazione I e stato σ vale $I, \sigma \vDash F\{x/a\}$ se e solo se $I, \sigma \vDash \forall x\, F$?
(c) Dimostrate con un controesempio che in (a) l'ipotesi che a non compaia in F è necessaria.
(d) Dal fatto che a occorra in F segue la falsità di (a)?

4. Il lemma di sostituzione

In questa sezione affrontiamo il problema di stabilire come la relazione (semantica) di soddisfazione interagisce con l'operazione (sintattica) di sostituzione.

Iniziamo, come al solito, a considerare il caso dei termini.

Lemma 7.58 (Lemma di Sostituzione per termini). *Siano σ uno stato di un'interpretazione I, x una variabile e s e t due termini. Allora*
$$\sigma(s\{x/t\}) = \sigma[x/\sigma(t)](s).$$

Dimostrazione. Per induzione sulla complessità di s, ricordando la definizione 6.18 di sostituzione in un termine.

Se s è un simbolo di costante c, si ha $\sigma(s\{x/t\}) = \sigma(s) = c^I = \sigma[x/\sigma(t)](s)$.

Se s è una variabile y diversa da x si ha $\sigma(s\{x/t\}) = \sigma(s) = \sigma(y) = \sigma[x/\sigma(t)](s)$.

Se s è x si ha $\sigma(s\{x/t\}) = \sigma(t) = \sigma[x/\sigma(t)](s)$.

Se s è $f(s_1, \dots, s_n)$ allora

$$\begin{aligned}\sigma(s\{x/t\}) &= \sigma(f(s_1, \dots, s_n)\{x/t\}) \\ &= \sigma(f(s_1\{x/t\}, \dots, s_n\{x/t\})) \\ &= f^I(\sigma(s_1\{x/t\}), \dots, (\sigma(s_n\{x/t\}))) \\ &= f^I(\sigma[x/\sigma(t)](s_1), \dots, \sigma[x/\sigma(t)](s_n)) \\ &= \sigma[x/\sigma(t)](f(s_1, \dots, s_n)) = \sigma[x/\sigma(t)](s),\end{aligned}$$

dove nel passaggio dalla terza alla quarta riga si è usata l'ipotesi induttiva, mentre nei passaggi tra prima e seconda riga e tra la quarta e la quinta si è utilizzata la definizione 6.18. □

Passiamo ora al caso delle formule.

Lemma 7.59 (Lemma di Sostituzione per formule). *Siano σ uno stato di un'interpretazione I, x una variabile, t un termine e F una formula. Se la sostituzione $\{x/t\}$ è ammissibile in F, allora*

$$I, \sigma \vDash F\{x/t\} \quad \textit{se e solo se} \quad I, \sigma[x/\sigma(t)] \vDash F.$$

Dimostrazione. Per induzione sulla complessità di F, ricordando le definizioni 6.50 e 6.55 di sostituzione in una formula.

Se F è la formula atomica $p(s_1, \dots, s_n)$ allora

$$\begin{aligned}I, \sigma \vDash F\{x/t\} &\text{ se e solo se } I, \sigma \vDash p(s_1, \dots, s_n)\{x/t\} \\ &\text{ se e solo se } I, \sigma \vDash p(s_1\{x/t\}, \dots, s_n\{x/t\}) \\ &\text{ se e solo se } (\sigma(s_1\{x/t\}), \dots, \sigma(s_n\{x/t\})) \in p^I \\ &\text{ se e solo se } (\sigma[x/\sigma(t)](s_1), \dots, \sigma[x/\sigma(t)](s_n)) \in p^I \\ &\text{ se e solo se } I, \sigma[x/\sigma(t)] \vDash p(s_1, \dots, s_n) \\ &\text{ se e solo se } I, \sigma[x/\sigma(t)] \vDash F,\end{aligned}$$

dove nel passaggio dalla terza alla quarta riga abbiamo utilizzato il lemma 7.58.

Se F è $G \vee H$ si ha

$$\begin{aligned}I, \sigma \vDash F\{x/t\} &\text{ se e solo se } I, \sigma \vDash (G \vee H)\{x/t\} \\ &\text{ se e solo se } I, \sigma \vDash (G\{x/t\} \vee H\{x/t\}) \\ &\text{ se e solo se } I, \sigma \vDash G\{x/t\} \text{ oppure } I, \sigma \vDash H\{x/t\} \\ &\text{ se e solo se } I, \sigma[x/\sigma(t)] \vDash G \text{ oppure } I, \sigma[x/\sigma(t)] \vDash H \\ &\text{ se e solo se } I, \sigma[x/\sigma(t)] \vDash G \vee H \\ &\text{ se e solo se } I, \sigma[x/\sigma(t)] \vDash F,\end{aligned}$$

dove nel passaggio dalla terza alla quarta riga abbiamo utilizzato l'ipotesi induttiva.

In modo del tutto analogo si tratta il caso di negazioni, congiunzioni e implicazioni.

Se F è $\forall y\, G$ dobbiamo distinguere due casi.

(i) Se x non è libero in F (che comprende il caso in cui y è x) si ha che $F\{x/t\}$ è F. Inoltre $I, \sigma \vDash F$ è equivalente a $I, \sigma[x/\sigma(t)] \vDash F$ per il lemma 7.12 e quindi abbiamo l'equivalenza cercata.

(ii) Se x è libero in F le variabili y e x sono certamente distinte e quindi $F\{x/t\}$ è $\forall y\, G\{x/t\}$. Inoltre, dato che la sostituzione $\{x/t\}$ è ammissibile in F, y non occorre in t. Allora

$$\begin{aligned} I,\sigma \models F\{x/t\} &\text{ se e solo se per ogni } d \in D^I,\ I,\sigma[y/d] \models G\{x/t\} \\ &\text{ se e solo se per ogni } d \in D^I,\ I,\sigma[y/d][x/\sigma[y/d](t)] \models G \\ &\text{ se e solo se per ogni } d \in D^I,\ I,\sigma[y/d][x/\sigma(t)] \models G \\ &\text{ se e solo se per ogni } d \in D^I,\ I,\sigma[x/\sigma(t)][y/d] \models G \\ &\text{ se e solo se } I,\sigma[x/\sigma(t)] \models F, \end{aligned}$$

dove nel passaggio dalla prima alla seconda riga abbiamo utilizzato l'ipotesi induttiva, in quello dalla seconda alla terza riga il fatto che y non occorre in t e il lemma 7.6, e in quello dalla terza alla quarta riga che y e x sono distinte (e quindi $\sigma[y/d][x/\sigma(t)]$ e $\sigma[x/\sigma(t)][y/d]$ sono lo stesso stato).

Se F è una quantificazione esistenziale procediamo in modo analogo: il caso (i) è identico, mentre nel caso (ii) si tratta di sostituire "per ogni" con "esiste". □

Esercizio 7.60. Fate i passi induttivi che sono stati omessi nella dimostrazione del lemma 7.59.

Notiamo che se nelle dimostrazioni dei lemmi 7.58 e 7.59 si tentasse di procedere per induzione sulla complessità del termine t ci si bloccherebbe ben presto.

Il lemma di sostituzione ha diverse conseguenze immediate che ci saranno utili in seguito.

Lemma 7.61. *Se y non è libera in F e la sostituzione $\{x/y\}$ è ammissibile in F allora*

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

Dimostrazione. Entrambe le affermazioni discendono facilmente dal fatto che per ogni interpretazione I, stato σ di I e $d \in D^I$ si ha

$$I,\sigma[y/d] \models F\{x/y\} \quad \text{se e solo se} \quad I,\sigma[x/d] \models F.$$

Per verificare questo osserviamo che se x e y sono la stessa variabile non c'è nulla da dimostrare, mentre se x e y sono distinte abbiamo

$$\begin{aligned} I,\sigma[y/d] \models F\{x/y\} &\text{ se e solo se } I,\sigma[y/d][x/\sigma[y/d](y)] \models F \\ &\text{ se e solo se } I,\sigma[y/d][x/d] \models F \\ &\text{ se e solo se } I,\sigma[x/d] \models F, \end{aligned}$$

dove nella prima riga abbiamo utilizzato il lemma di sostituzione e nel passaggio dalla seconda alla terza riga il fatto che y non è libera in F (lemma 7.12). □

Il lemma 7.61 ci permette di giustificare quanto affermato dopo la definizione 6.59 a proposito delle varianti.

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 che in quella della deduzione naturale predicativa.

Lemma 7.63. *Se F è una formula, x una variabile, t un termine e $\{x/t\}$ una sostituzione ammissibile in F, allora $\forall x\, F \models F\{x/t\}$ e $F\{x/t\} \models \exists x\, F$.*

DIMOSTRAZIONE. Se $I, \sigma \vDash \forall x\, F$ allora si ha $I, \sigma[x/d] \vDash F$ per ogni $d \in D^I$ e in particolare questo vale per $d = \sigma(t)$. Per il Lemma di Sostituzione 7.59 questo significa che $I, \sigma \vDash F\{x/t\}$.

Se $I, \sigma \vDash F\{x/t\}$ allora per il Lemma di Sostituzione 7.59 $I, \sigma[x/\sigma(t)] \vDash F$. Quindi si ha $I, \sigma \vDash \exists x\, F$. $\square$

Il seguente corollario va confrontato con il lemma 7.41, che arriva ad una conclusione più forte nel caso in cui x non sia libera in F.

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 questo simbolo di relazione binario abbiamo assegnato un ruolo particolare. Questo ruolo si riflette nel fatto che per linguaggi con uguaglianza vogliamo restringere la nostra attenzione a interpretazioni in cui $=$ è interpretato “correttamente”.

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 $=$.

NOTAZIONE 7.66. Se scriviamo $I, \sigma \vDash_= F$ intendiamo dire che $I, \sigma \vDash F$ e che I è un'interpretazione normale.

Il prossimo lemma stabilisce un legame tra le interpretazioni normali e l'insieme di assiomi dell'uguaglianza $\mathsf{Eq}_{\mathcal{L}}$ introdotto nella definizione 6.64. Esso afferma che $\mathsf{Eq}_{\mathcal{L}}$ esprime alcune delle proprietà fondamentali dei linguaggi con uguaglianza.

LEMMA 7.67. *Se I è un'interpretazione normale per $\mathcal{L} \cup \{=\}$ allora $I \vDash_= \mathsf{Eq}_{\mathcal{L}}$.*

DIMOSTRAZIONE. Immediata, ispezionando la definizione 6.64. $\square$

ESEMPIO 7.68. L'inverso del lemma 7.67 non è vero, cioè non è vero che se un'interpretazione I è un modello degli assiomi dell'uguaglianza allora I è normale (si veda però il teorema 10.33).

L'interpretazione I per il linguaggio con uguaglianza $\mathcal{L} = \{p, =\}$ definita da $D^I = \{A, B, C\}$, $p^I = \{A, C\}$, $=^I = \{(A, A), (B, B), (C, C), (A, C), (C, A)\}$ non è normale, ma è facile verificare (fatelo!) che $I \vDash \mathsf{Eq}_{\mathcal{L}}$.

Usando le interpretazioni normali è possibile definire una *semantica per la logica con uguaglianza*: essa è del tutto analoga alla semantica definita nelle sezioni precedenti per linguaggi arbitrari (per distinguerla quest'ultima viene talvolta chiamata *semantica per la logica pura*), ma con la restrizione che vengono prese in considerazione solamente le interpretazioni normali.

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 $T \vDash_= G$) se per ogni interpretazione normale I per $\mathcal{L}$ ed ogni stato σ di I tale che $I, \sigma \vDash_= T$ si ha anche $I, \sigma \vDash_= G$. Se $T = \{F\}$ allora scriviamo $F \vDash_= G$ 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 $F \vDash_= G$ e $G \vDash_= F$.

Esempio 7.70. Se T e F sono un insieme di formule ed una formula del linguaggio con uguaglianza $\mathcal{L}$ e si ha $T \models F$, allora $T \models_= F$. Infatti per stabilire che $T \models F$ bisogna prendere in considerazione tutte le interpretazioni per $\mathcal{L}$, e tra di esse ci sono anche tutte quelle normali.

Se invece sappiamo che $T \models_= F$ non possiamo concludere che $T \models F$. Ad esempio $\models_= a = a$ (perché in ogni interpretazione normale I si ha $(a^I, a^I) \in =^I$), ma $\not\models a = a$ (è facile costruire un'interpretazione non normale in cui $(a^I, a^I) \notin =^I$).

Esercizio 7.71. Dimostrare che $p(a), \neg p(b) \models_= a \neq b$ e che $p(a), \neg p(b) \not\models a \neq b$.

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 $I, \sigma \not\models_= F$.

Esercizio 7.73. Rispondete alle seguenti domande, fornendo dei controesempi se la risposta è negativa:

(a) Se F è valida nella logica con uguaglianza allora F è valida?
(b) Se F è valida allora F è valida nella logica con uguaglianza?
(c) Se F è soddisfacibile nella logica con uguaglianza allora F è soddisfacibile?
(d) Se F è soddisfacibile allora F è soddisfacibile nella logica con uguaglianza?
(e) Se F è insoddisfacibile nella logica con uguaglianza allora F è insoddisfacibile?
(f) Se F è insoddisfacibile allora F è insoddisfacibile nella logica con uguaglianza?

Esempio 7.74. Sia H_1 l'enunciato $\forall x \forall y\, x = y$. Si verifica facilmente che $I \models_= H_1$ se e solo se D^I ha *esattamente* un elemento (ricordate che D^I è sempre non vuoto).

Siano F_2 e G_2 gli enunciati $\exists x \exists y\, x \neq y$ e $\forall x \forall y \forall z(x = y \vee x = z \vee y = z)$. In questo caso si ha che $I \models_= F_2$ se e solo se D^I ha *almeno* due elementi, mentre $I \models_= G_2$ se e solo se D^I ha *al più* due elementi. Perciò $I \models_= F_2 \wedge G_2$ se e solo se D^I ha *esattamente* due elementi. Se H_2 è $\exists x \exists y(x \neq y \wedge \forall z(z = x \vee z = y))$ si ha $H_2 \equiv_= F_2 \wedge G_2$ e quindi $I \models_= H_2$ se e solo se D^I ha esattamente due elementi.

Più in generale, per ogni $n > 0$ si possono scrivere enunciati (sempre più lunghi al crescere di n) F_n, G_n e H_n tali che

- $I \models_= F_n$ se e solo se D^I ha almeno n elementi;
- $I \models_= G_n$ se e solo se D^I ha al più n elementi;
- $I \models_= F_n \wedge G_n$ se e solo se $I \models_= H_n$ se e solo se D^I ha esattamente n elementi.

Esercizio 7.75. Nella notazione dell'esempio 7.74 scrivete gli enunciati F_3, G_3 e H_3.

L'esempio 7.74 mostra come nella logica con uguaglianza sia possibile in un certo senso "contare" gli elementi del dominio. Questo resta possibile fino a che si ha a che fare con interpretazioni con dominio di cardinalità finita e fissata. Infatti (ma per dimostrarlo servono metodi che esulano dall'ambito di questo corso) non esiste un enunciato F tale che $I \models_= F$ se e solo se D^I è finito.

Esempio 7.76. Sia $\mathcal{L}$ un linguaggio con uguaglianza che contiene un simbolo di relazione unario p. La formula $\exists x(p(x) \wedge \forall y(p(y) \rightarrow y = x))$ asserisce che esiste esattamente un elemento che soddisfa p. La formula $\exists x\, \exists y(x \neq y \wedge p(x) \wedge p(y))$ asserisce che esistono almeno due elementi che soddisfano p.

Esercizio 7.77. Nel linguaggio dell'esempio precedente, scrivete una formula che asserisce che esistono esattamente due elementi che soddisfano p.

Esercizio 7.78. Nel linguaggio dell'esempio 7.76, per ogni numero naturale $n > 1$, scrivete una formula F_n tale che

$$I \models_{=} F_n \quad \text{se e solo se} \quad \begin{array}{c} p^I \text{ ha almeno } n \text{ elementi e} \\ D^I \setminus p^I \text{ ha almeno } n \text{ elementi.} \end{array}$$

Esercizio 7.79. Nel linguaggio con uguaglianza che contiene un simbolo funzionale unario f e un simbolo relazionale binario r scrivete un enunciato F che traduce "ogni elemento nell'immagine di f è in relazione r con se stesso, ma nulla è in relazione r con la sua immagine secondo f". Dimostrate che

$$F \models_{=} \forall x\, f(x) \neq x.$$

Esercizio 7.80. Nel linguaggio con uguaglianza che contiene i simboli relazionali unari p e q scrivete enunciati F, G e H tali che per ogni interpretazione normale I si abbia che:

(i) $I \models_{=} F$ se e solo se esistono almeno due elementi del dominio che non soddisfano né p né q;
(ii) $I \models_{=} G$ se e solo se esiste al più un elemento del dominio che soddisfa q;
(iii) $I \models_{=} H$ se e solo se esistono esattamente tre elementi del dominio che soddisfano uno solo tra p e q.

Definite un'interpretazione normale che soddisfa i tre enunciati e ha dominio di cardinalità minima possibile.

CAPITOLO 8

Trasformazione in forma prenessa

In questo capitolo ci occupiamo di trasformare una formula in una ad essa logicamente equivalente in cui i quantificatori compaiono in una posizione particolare.

DEFINIZIONE 8.1. Una formula F si dice *in forma prenessa* se è priva di quantificatori oppure è della forma $Q_1x_1 \dots Q_kx_k\, G$, dove G è una formula priva di quantificatori e $Q_1, \dots, Q_k$ sono quantificatori. In questo caso $Q_1x_1 \dots Q_kx_k$ (che non è una formula) si dice il *prefisso* di F, mentre G è chiamata la *matrice* di F.

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

Il nostro obiettivo è dunque dimostrare il seguente teorema.

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.

ESEMPIO 8.4. Il teorema 8.3 non asserisce l'unicità di K. Ad esempio vedremo che la formula $\forall x\, p(x) \land \exists y\, q(y)$ è logicamente equivalente alle formule in forma prenessa $\forall x\, \exists y(p(x) \land q(y))$ e $\exists y\, \forall x(p(x) \land q(y))$ (entrambe le equivalenze logiche si ottengono applicando due volte il lemma 8.7).

L'algoritmo che descriveremo per dimostrare il teorema 8.3 si basa sulle equivalenze logiche che dimostreremo nella prossima sezione, e che ci permetteranno di "far uscire" i quantificatori dai connettivi.

1. Alcune equivalenze logiche notevoli

Iniziamo con l'esaminare l'interazione tra negazione e quantificatori.

LEMMA 8.5. *Per ogni formula F si ha*

$$\neg\forall x\, F \equiv \exists x\, \neg F,$$
$$\neg\exists x\, F \equiv \forall x\, \neg F.$$

DIMOSTRAZIONE. La dimostrazione consiste nello stabilire quattro conseguenze logiche, due per ogni equivalenza logica da dimostrare.

$\boxed{\neg\forall x\, F \models \exists x\, \neg F}$ Se $I, \sigma \models \neg\forall x\, F$ allora $I, \sigma \not\models \forall x\, F$, e quindi non è vero che per ogni $d \in D^I$ si ha $I, \sigma[x/d] \models F$. Esiste dunque $d_0 \in D^I$ tale che $I, \sigma[x/d_0] \not\models F$, e quindi tale che $I, \sigma[x/d_0] \models \neg F$. Di conseguenza $I, \sigma \models \exists x\, \neg F$.

$\boxed{\exists x\, \neg F \models \neg\forall x\, F}$ Se $I, \sigma \models \exists x\, \neg F$ esiste $d_0 \in D^I$ tale che $I, \sigma[x/d_0] \models \neg F$, cioè $I, \sigma[x/d_0] \not\models F$. Perciò $I, \sigma \not\models \forall x\, F$, e quindi si ha che $I, \sigma \models \neg\forall xF$.

$\boxed{\neg\exists x\, F \models \forall x\, \neg F}$ Se $I, \sigma \models \neg\exists x\, F$ allora $I, \sigma \not\models \exists x\, F$. Qualunque sia $d \in D^I$, si ha allora che $I, \sigma[x/d] \not\models F$, cioè $I, \sigma[x/d] \models \neg F$. Di conseguenza $I, \sigma \models \forall x\, \neg F$.

$\boxed{\forall x \, \neg F \models \neg \exists x \, F}$ Se $I, \sigma \models \forall x \, \neg F$ allora per ogni $d \in D^I$ si ha $I, \sigma[x/d] \not\models F$. Non esiste dunque alcun $d \in D^I$ tale che $I, \sigma[x/d] \models F$ e quindi $I, \sigma \not\models \exists x \, F$. Di conseguenza $I, \sigma \models \neg \exists x \, F$. □

Esercizio 8.6. Date una dimostrazione alternativa della seconda equivalenza logica del lemma 8.5, utilizzando la prima equivalenza logica dello stesso lemma, e i lemmi 2.20.1 e 7.39. Iniziate da $\neg \exists x \, F \equiv \neg \exists x \, \neg\neg F$.

Passiamo ora a considerare il caso della congiunzione e della disgiunzione.

Lemma 8.7. *Siano $*$ uno di $\wedge$ e $\vee$ e Q uno di $\forall$ e $\exists$. Per ogni formula F, ogni variabile x e ogni formula G in cui x non è libera, si ha*

$$Qx \, F * G \equiv Qx(F * G),$$
$$G * Qx \, F \equiv Qx(G * F).$$

Dimostrazione. Ognuna delle equivalenze logiche contenute nella seconda riga segue da quella corrispondente contenuta nella prima riga usando il lemma 2.20.2 o 2.20.3 e il lemma 7.39. Restano quindi da dimostrare le quattro equivalenze logiche (e quindi le otto conseguenze logiche) contenute nella prima riga. Un utile esercizio consiste nel dimostrare autonomamente queste conseguenze logiche.

$\boxed{\forall x \, F \wedge G \models \forall x(F \wedge G)}$ Se $I, \sigma \models \forall x \, F \wedge G$, allora $I, \sigma \models \forall x \, F$ e $I, \sigma \models G$. Da $I, \sigma \models \forall x \, F$ si ha che per ogni $d \in D^I$, $I, \sigma[x/d] \models F$. Inoltre, da $I, \sigma \models G$, dato che x non è libera in G, si ha anche che $I, \sigma[x/d] \models G$. Di conseguenza $I, \sigma[x/d] \models F \wedge G$. Poiché questo è vero per ogni $d \in D^I$, segue che $I, \sigma \models \forall x(F \wedge G)$.

$\boxed{\forall x(F \wedge G) \models \forall x \, F \wedge G}$ Se $I, \sigma \models \forall x(F \wedge G)$, allora per ogni $d \in D^I$ si ha che $I, \sigma[x/d] \models F$ e $I, \sigma[x/d] \models G$. Dal fatto che per ogni $d \in D^I$ si abbia $I, \sigma[x/d] \models F$, segue che $I, \sigma \models \forall x \, F$. D'altra parte quando $d = \sigma(x)$, $\sigma[x/d]$ è σ, e perciò da $I, \sigma[x/d] \models G$, segue che $I, \sigma \models G$. Di conseguenza $I, \sigma \models \forall x \, F \wedge G$.

$\boxed{\forall x \, F \vee G \models \forall x(F \vee G)}$ Se $I, \sigma \models \forall x \, F \vee G$, allora $I, \sigma \models \forall x \, F$ oppure $I, \sigma \models G$. Se $I, \sigma \models \forall x \, F$, allora per ogni $d \in D^I$, $I, \sigma[x/d] \models F$ da cui segue che per ogni $d \in D^I$, $I, \sigma[x/d] \models F \vee G$ e quindi che $I, \sigma \models \forall x(F \vee G)$. Se invece $I, \sigma \models G$, allora per ogni $d \in D^I$, $I, \sigma[x/d] \models G$, in quanto x non occorre libera in G. Ne segue anche in questo caso, che per ogni $d \in D^I$, $I, \sigma[x/d] \models F \vee G$ e quindi che $I, \sigma \models \forall x(F \vee G)$.

$\boxed{\forall x(F \vee G) \models \forall x \, F \vee G}$ Se $I, \sigma \models \forall x(F \vee G)$ allora dimostreremo che $I, \sigma \models \forall x \, F \vee G$, sia se $I, \sigma \models G$, che se $I, \sigma \not\models G$. Se $I, \sigma \models G$, ovviamente $I, \sigma \models \forall x \, F \vee G$. Se invece $I, \sigma \not\models G$, per ogni $d \in D^I$, $I, \sigma[x/d] \not\models G$, in quanto x non occorre libera in G. Poiché per ogni $d \in D^I$, $I, \sigma[x/d] \models F \vee G$, ne segue che per ogni $d \in D^I$, $I, \sigma[x/d] \models F$. Quindi $I, \sigma \models \forall x \, F$ da cui segue che $I, \sigma \models \forall x \, F \vee G$.

$\boxed{\exists x \, F \wedge G \models \exists x(F \wedge G)}$ Se $I, \sigma \models \exists x \, F \wedge G$, allora $I, \sigma \models \exists x \, F$ e $I, \sigma \models G$. Da $I, \sigma \models \exists x \, F$, segue che esiste $d_0 \in D^I$ tale che $I, \sigma[x/d_0] \models F$. D'altra parte dato che x non occorre libera in G, si ha $I, \sigma[x/d_0] \models G$. Di conseguenza $I, \sigma[x/d_0] \models F \wedge G$, e quindi $I, \sigma \models \exists x(F \wedge G)$.

$\boxed{\exists x(F \wedge G) \models \exists x \, F \wedge G}$ Se $I, \sigma \models \exists x(F \wedge G)$ allora esiste $d_0 \in D^I$ tale che $I, \sigma[x/d_0] \models F \wedge G$. Perciò $I, \sigma[x/d_0] \models F$ e $I, \sigma[x/d_0] \models G$. Da $I, \sigma[x/d_0] \models F$ segue che $I, \sigma \models \exists x \, F$. Dato che x non è libera in G da $I, \sigma[x/d_0] \models G$, segue che $I, \sigma \models G$. Di conseguenza $I, \sigma \models \exists x \, F \wedge G$.

$\boxed{\exists x \, F \vee G \models \exists x(F \vee G)}$ Se $I, \sigma \models \exists x \, F \vee G$, allora $I, \sigma \models \exists x \, F$ oppure $I, \sigma \models G$. Nel primo caso esiste $d_0 \in D^I$, tale che $I, \sigma[x/d_0] \models F$, e quindi $I, \sigma[x/d_0] \models F \vee G$, da cui $I, \sigma \models \exists x(F \vee G)$. Nel secondo caso si ha $I, \sigma \models F \vee G$; posto $d_0 = \sigma(x)$, σ è $\sigma[x/d_0]$ e quindi $I, \sigma[x/d_0] \models F \vee G$, da cui segue che $I, \sigma \models \exists x(F \vee G)$.

Poiché la conclusione segue in entrambi i casi possibili, possiamo concludere che $I, \sigma \vDash \exists x(F \vee G)$.

$\boxed{\exists x(F \vee G) \vDash \exists x\, F \vee G}$ Se $I, \sigma \vDash \exists x(F \vee G)$ sia $d_0 \in D^I$ tale che $I, \sigma[x/d_0] \vDash F \vee G$, cioè tale che $I, \sigma[x/d_0] \vDash F$ oppure $I, \sigma[x/d_0] \vDash G$. Nel primo caso $I, \sigma \vDash \exists x\, F$ e quindi $I, \sigma \vDash \exists x\, F \vee G$. Nel secondo caso, dato che x non è libera in G, si ha $I, \sigma \vDash G$, da cui segue $I, \sigma \vDash \exists x\, F \vee G$. Poiché la conclusione segue in entrambi i casi possibili, possiamo concludere che $I, \sigma \vDash \exists x\, F \vee G$. □

Esercizio 8.8. Se x non è libera in G giustificate la seguente catena di equivalenze logiche:

$$\begin{aligned}\exists x\, F \vee G &\equiv \neg\neg(\exists x\, F \vee G) \equiv \neg(\neg\exists x\, F \wedge \neg G) \equiv \neg(\forall x\, \neg F \wedge \neg G) \equiv \\ &\equiv \neg\forall x(\neg F \wedge \neg G) \equiv \neg\forall x\, \neg(F \vee G) \equiv \neg\neg\exists x(F \vee G) \equiv \exists x(F \vee G).\end{aligned}$$

In questo modo abbiamo dimostrato una delle equivalenze logiche riguardanti $\vee$ del lemma 8.7 sulla base (tra l'altro) di una delle equivalenze logiche riguardanti $\wedge$ dello stesso lemma. Analogamente dimostrate $\forall x\, F \vee G \equiv \forall x(F \vee G)$ attraverso una catena di equivalenze logiche.

Passiamo ora a considerare il caso dell'implicazione.

Lemma 8.9. *Per ogni formula F, ogni variabile x e ogni formula G in cui x non è libera, si ha*

$$\begin{aligned} &\forall x\, F \to G \equiv \exists x(F \to G), &\qquad G \to \forall x\, F \equiv \forall x(G \to F),\\ &\exists x\, F \to G \equiv \forall x(F \to G), &\qquad G \to \exists x\, F \equiv \exists x(G \to F).\end{aligned}$$

Dimostrazione. Questo lemma può essere dimostrato considerando interpretazioni e stati come fatto per i lemmi 8.5 e 8.7. Ne diamo invece una dimostrazione attraverso la costruzione di catene di equivalenze logiche:

$$\begin{aligned}\forall x\, F \to G &\equiv \neg\forall x\, F \vee G \equiv \exists x\, \neg F \vee G \equiv \exists x(\neg F \vee G) \equiv \exists x(F \to G);\\ G \to \forall x\, F &\equiv \neg G \vee \forall x\, F \equiv \forall x(\neg G \vee F) \equiv \forall x(G \to F);\\ \exists x\, F \to G &\equiv \neg\exists x\, F \vee G \equiv \forall x\, \neg F \vee G \equiv \forall x(\neg F \vee G) \equiv \forall x(F \to G);\\ G \to \exists x\, F &\equiv \neg G \vee \exists x\, F \equiv \exists x(\neg G \vee F) \equiv \exists x(G \to F).\end{aligned}$$

Abbiamo utilizzato i lemmi 7.39, 2.24.3, 8.5 e 8.7. □

Esercizio 8.10. Dimostrate il lemma 8.9 utilizzando interpretazioni e stati (dovete dimostrare otto conseguenze logiche).

Esempio 8.11. Se F e G sono $p(x)$ e $q(x)$ allora $\forall x\, F \wedge G \nvDash \forall x(F \wedge G)$: un'interpretazione e uno stato che lo mostrano sono ad esempio $D^I = \{0,1\}$, $p^I = \{0,1\}$, $q^I = \{0\}$, $\sigma(x) = 0$.

Esercizio 8.12. Mostrate che $\forall x(F \wedge G) \vDash \forall x\, F$ e quindi che $\forall x(F \wedge G) \vDash \forall x\, F \wedge G$. Mostrate anche che $\exists x\, F \vee G \vDash \exists x(F \vee G)$.

Esercizio 8.13. Siano F e G le formule dell'esempio 8.11. Dimostrate:

$$\begin{aligned} &\exists x(F \wedge G) \nvDash \exists x\, F \wedge G, &\qquad \exists x\, F \wedge G \nvDash \exists x(F \wedge G),\\ &\forall x(F \vee G) \nvDash \forall x\, F \vee G, &\qquad \forall x\, F \vee G \nvDash \forall x(F \vee G),\\ &\exists x(F \vee G) \nvDash \exists x\, F \vee G. \end{aligned}$$

In qualche caso è possibile trovare una formula logicamente equivalente che contenga un quantificatore in meno di quella di partenza.

LEMMA 8.14. *Per ogni formula F e G si ha*

$$\begin{aligned}\forall x\, F \wedge \forall x\, G &\equiv \forall x(F \wedge G),\\ \exists x\, F \vee \exists x\, G &\equiv \exists x(F \vee G),\\ \forall x\, F \to \exists x\, G &\equiv \exists x(F \to G).\end{aligned}$$

DIMOSTRAZIONE. La prima equivalenza logica può essere dimostrata direttamente, mentre per la seconda stabiliamo le due conseguenze logiche.

$\boxed{\forall x\, F \wedge \forall x\, G \equiv \forall x(F \wedge G)}$ $I, \sigma \models \forall x(F \wedge G)$ se e solo se per ogni $d \in D^I$ si ha $I, \sigma[x/d] \models F$ e $I, \sigma[x/d] \models G$, se e solo se $I, \sigma \models \forall x\, F$ e $I, \sigma \models \forall x\, G$, se e solo se $I, \sigma \models \forall x\, F \wedge \forall x\, G$.

$\boxed{\exists x\, F \vee \exists x\, G \models \exists x(F \vee G)}$ Se $I, \sigma \models \exists x\, F \vee \exists x\, G$ allora $I, \sigma \models \exists x\, F$ oppure $I, \sigma \models \exists x\, G$. Nel primo caso esiste $d_0 \in D^I$ tale che $I, \sigma[x/d_0] \models F$, mentre nel secondo esiste $d_0 \in D^I$ tale che $I, \sigma[x/d_0] \models G$. In ogni caso $I, \sigma[x/d_0] \models F \vee G$ e quindi $I, \sigma \models \exists x(F \vee G)$.

$\boxed{\exists x(F \vee G) \models \exists x\, F \vee \exists x\, G}$ Se $I, \sigma \models \exists x(F \vee G)$ allora esiste $d_0 \in D^I$ tale che $I, \sigma[x/d_0] \models F \vee G$ e quindi $I, \sigma[x/d_0] \models F$ oppure $I, \sigma[x/d_0] \models G$. Nel primo caso $I, \sigma \models \exists x\, F$, nel secondo caso $I, \sigma \models \exists x\, G$; in ogni caso $I, \sigma \models \exists x\, F \vee \exists x\, G$.

Per dimostrare la terza equivalenza logica osserviamo che

$$\begin{aligned}\forall x\, F \to \exists x\, G &\equiv \neg\forall x\, F \vee \exists x\, G\\ &\equiv \exists x\, \neg F \vee \exists x\, G\\ &\equiv \exists x(\neg F \vee G)\\ &\equiv \exists x(F \to G),\end{aligned}$$

dove i passaggi sono giustificati dalla seconda equivalenza logica, dai lemmi 7.39, 8.5 e 2.24.3. □

ESERCIZIO 8.15. Dimostrate l'equivalenza logica nella terza riga del lemma 8.14 utilizzando interpretazioni e stati.

ESERCIZIO 8.16. Dimostrate l'equivalenza logica nella seconda riga del lemma 8.14 tramite una catena di equivalenze logiche che sfruttino, tra l'altro, la prima equivalenza logica dello stesso lemma. Si consiglia di partire da $\exists x(F \vee G) \equiv \neg\neg\exists x(F \vee G)$.

ESERCIZIO 8.17. Dimostrate le seguenti conseguenze logiche:

$$\begin{aligned}\exists x(F \wedge G) &\models \exists x\, F \wedge \exists x\, G;\\ \forall x\, F \vee \forall x\, G &\models \forall x(F \vee G);\\ \exists x\, F \to \forall x\, G &\models \forall x(F \to G).\end{aligned}$$

Quando F è $p(x)$ e G è $q(x)$ dimostrate che le conseguenze logiche inverse sono false, e quindi le formule non sono logicamente equivalenti.

NOTA 8.18. La seconda parte dell'esercizio 8.17 mostra che

$$\begin{aligned}\exists x\, F \wedge \exists x\, G &\not\equiv \exists x(F \wedge G),\\ \forall x\, F \vee \forall x\, G &\not\equiv \forall x(F \vee G),\\ \exists x\, F \to \forall x\, G &\not\equiv \forall x(F \to G).\end{aligned}$$

Per memorizzare le equivalenze logiche del lemma 8.14 è utile notare che $\forall$ e $\exists$ possono essere considerati come estensioni rispettivamente di $\wedge$ e $\vee$ a tutti gli elementi del dominio. Infatti $\forall x\, p(x)$ asserisce che p è vera per questo elemento del dominio **e** quest'altro, **e** quest'altro.... D'altra parte $\exists x\, p(x)$ asserisce che p è vera per questo elemento del dominio **oppure** per quest'altro, **oppure** per quest'altro.... Inoltre è utile ricordare che $\rightarrow$ è equivalente a una disgiunzione (lemma 2.24.3) e quindi si comporta analogamente a $\vee$.

Esempio 8.19. Utilizziamo i lemmi 8.5, 8.7 e 8.9 (nonché il lemma 7.39) per trasformare in forma prenessa la formula $\neg(\forall x\, p(x) \rightarrow \neg(\exists y\, r(y,y) \vee \forall z\, q(z)))$. La formula che compare in una riga è logicamente equivalente a quella che compare nella riga precedente in virtù del lemma il cui numero compare sulla destra.

$$\begin{array}{ll}
\neg(\forall x\, p(x) \rightarrow \neg(\exists y\, r(y,y) \vee \forall z\, q(z))) & \\
\neg(\forall x\, p(x) \rightarrow \neg\exists y(r(y,y) \vee \forall z\, q(z))) & 8.7 \\
\neg(\forall x\, p(x) \rightarrow \neg\exists y\, \forall z(r(y,y) \vee q(z))) & 8.7 \\
\neg(\forall x\, p(x) \rightarrow \forall y\, \neg\forall z(r(y,y) \vee q(z))) & 8.5 \\
\neg(\forall x\, p(x) \rightarrow \forall y\, \exists z\, \neg(r(y,y) \vee q(z))) & 8.5 \\
\neg\forall y(\forall x\, p(x) \rightarrow \exists z\, \neg(r(y,y) \vee q(z))) & 8.9 \\
\neg\forall y\, \exists z(\forall x\, p(x) \rightarrow \neg(r(y,y) \vee q(z))) & 8.9 \\
\neg\forall y\, \exists z\, \exists x(p(x) \rightarrow \neg(r(y,y) \vee q(z))) & 8.9 \\
\exists y\, \neg\exists z\, \exists x(p(x) \rightarrow \neg(r(y,y) \vee q(z))) & 8.5 \\
\exists y\, \forall z\, \neg\exists x(p(x) \rightarrow \neg(r(y,y) \vee q(z))) & 8.5 \\
\exists y\, \forall z\, \forall x\, \neg(p(x) \rightarrow \neg(r(y,y) \vee q(z))) & 8.5.
\end{array}$$

L'ultima formula è in forma prenessa ed è logicamente equivalente a quella di partenza. Notate anche come nell'applicazione delle varie regole abbiamo fatto diverse scelte arbitrarie: ad esempio il primo passo si poteva utilizzare il lemma 8.7 per ottenere $\neg(\forall x\, p(x) \rightarrow \neg\forall z(\exists y\, r(y,y) \vee q(z)))$ oppure il lemma 8.9 per ottenere $\neg\exists x(p(x) \rightarrow \neg(\exists y\, r(y,y) \vee \forall z\, q(z)))$.

I lemmi 8.7 e 8.9 non possono essere applicati alla formula $q(x) \wedge \forall x\, p(x)$ per ottenere una sua forma prenessa, perché x è libera in $q(x)$. Per superare questo problema è necessario cambiare la variabile quantificata.

Lemma 8.20. *Siano $*$ uno di $\wedge$ e $\vee$ e Q uno di $\forall$ e $\exists$. Per ogni formula F, ogni variabile x e ogni formula G, se y è una variabile che non è libera né in F né in G e la sostituzione $\{x/y\}$ è ammissibile in F, si ha*

$$\begin{aligned}
Qx\, F * G &\equiv Qy(F\{x/y\} * G), \\
G * Qx\, F &\equiv Qy(G * F\{x/y\}), \\
\forall x\, F \rightarrow G &\equiv \exists y(F\{x/y\} \rightarrow G), \\
G \rightarrow \forall x\, F &\equiv \forall y(G \rightarrow F\{x/y\}), \\
\exists x\, F \rightarrow G &\equiv \forall y(F\{x/y\} \rightarrow G), \\
G \rightarrow \exists x\, F &\equiv \exists y(G \rightarrow F\{x/y\}).
\end{aligned}$$

Dimostrazione. Dimostriamo quanto asserito nella prima riga:

$$Qx\, F * G \equiv QyF\{x/y\} * G \equiv Qy(F\{x/y\} * G),$$

dove la prima equivalenza logica è giustificata dal lemma 7.61 (e dal lemma 7.39) e la seconda dal lemma 8.7 (visto che y non è libera in G).

L'asserzione della seconda riga è dimostrata similmente (oppure usando la prima riga, una delle equivalenze del lemma 2.20 e il lemma 7.39), mentre le asserzioni delle altre quattro righe hanno dimostrazioni analoghe, utilizzando il lemma 8.9 al posto del lemma 8.7. $\square$

ESEMPIO 8.21. Utilizziamo il lemma 8.20 (nonché il lemma 7.39) per trasformare in forma prenessa la formula $\forall x\, \exists y\, r(x,y) \to \exists x\, \forall y\, r(y,x)$. La formula che compare in una riga è logicamente equivalente a quella che compare nella riga precedente.

$$\begin{gathered}
\forall x\, \exists y\, r(x,y) \to \exists x\, \forall y\, r(y,x)\\
\exists x(\exists y\, r(x,y) \to \exists x\, \forall y\, r(y,x))\\
\exists x\, \forall y(r(x,y) \to \exists x\, \forall y\, r(y,x))\\
\exists x\, \forall y\, \exists z(r(x,y) \to \forall y\, r(y,z))\\
\exists x\, \forall y\, \exists z\, \forall w(r(x,y) \to r(w,z)).
\end{gathered}$$

I primi due passaggi sono giustificati dal lemma 8.9, mentre gli ultimi due sfruttano il lemma 8.20 (visto che sia x che y sono libere in $r(x,y)$). L'ultima formula è in forma prenessa ed è logicamente equivalente a quella di partenza.

NOTA 8.22. Notate come anche nell'esempio 8.21 l'ordine di applicazione delle varie regole è arbitrario: ad esempio nel primo passo era possibile operare su $\exists x$ anziché su $\forall x$ (si veda l'esercizio 8.32). Quello che **non** è possibile fare è 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 $\exists y\, \forall x\, r(x,y)$ (si veda la nota 7.34).

2. Il p-grado

Per descrivere l'algoritmo che dimostra il teorema 8.3 e che abbiamo già utilizzato negli esempi precedenti ci sarà utile la seguente nozione, che misura quanti connettivi devono essere superati dai quantificatori perché questi ultimi siano portati nel prefisso della forma prenessa.

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 $\to$;
- $\mathfrak{p}(F) = \mathfrak{p}(G)$ se F è $Qx\, G$ con Q uno di $\forall$ e $\exists$.

ESEMPIO 8.24. Calcoliamo $\mathfrak{p}(H)$ dove H è:

$$\forall x(p(x) \to \exists y(r(x,y) \wedge \neg\forall z\, \neg r(y,z))).$$

Applicando la quarta condizione della definizione di p-grado si ha $\mathfrak{p}(H) = \mathfrak{p}(H_1)$ dove H_1 è $p(x) \to \exists y(r(x,y) \wedge \neg\forall z\, \neg r(y,z))$. Se H_2 e H_3 sono l'antecedente e il conseguente dell'implicazione si ha $\mathfrak{p}(H_1) = \mathfrak{p}(H_2) + \mathfrak{p}(H_3) + \mathfrak{q}(H_1) = \mathfrak{p}(H_3) + 2$, dove abbiamo utilizzato la terza e la prima condizione della definizione di p-grado e il calcolo della funzione $\mathfrak{q}$. Ora abbiamo $\mathfrak{p}(H_3) = \mathfrak{p}(r(x,y) \wedge \neg\forall z\, \neg r(y,z)) = \mathfrak{p}(r(x,y)) + \mathfrak{p}(\neg\forall z\, \neg r(y,z)) + \mathfrak{q}(r(x,y) \wedge \neg\forall z\, \neg r(y,z)) = \mathfrak{p}(\neg\forall z\, \neg r(y,z)) + 1$. Infine abbiamo $\mathfrak{p}(\neg\forall z\, \neg r(y,z)) = \mathfrak{p}(\forall z\, \neg r(y,z)) + \mathfrak{q}(\neg\forall z\, \neg r(y,z)) = \mathfrak{p}(\forall z\, \neg r(y,z)) + 1$ e $\mathfrak{p}(\forall z\, \neg r(y,z)) = \mathfrak{p}(\neg r(y,z)) = \mathfrak{p}(r(y,z)) + \mathfrak{q}(\neg r(y,z)) = 0$.

Inserendo a ritroso i valori ottenuti si ottiene $\mathfrak{p}(H) = 4$.

ESERCIZIO 8.25. Calcolate

$$\mathfrak{p}(\forall x\, p(x) \to \exists y\, \forall z(r(z,y) \wedge \forall u\, \neg r(y,u))).$$

L'osservazione cruciale riguardo al p-grado è contenuta nel seguente lemma.

LEMMA 8.26. *Una formula F è in forma prenessa se e solo se $\mathfrak{p}(F) = 0$.*

Dimostrazione. Osserviamo dapprima che se G è priva di quantificatori allora $\mathfrak{p}(G) = 0$. Questo si vede con una semplice induzione sulla complessità di G, utilizzando le prime tre condizioni nella definizione di p-grado e il fatto che $\mathfrak{q}(G') = 0$ per ogni sottoformula G' di G.

Supponiamo ora che F sia $Q_1 x_1 \dots Q_k x_k\, G$ con G priva di quantificatori e $Q_1, \dots, Q_k$ quantificatori. La quarta condizione nella definizione di p-grado implica che $\mathfrak{p}(F) = \mathfrak{p}(G)$ e quindi $\mathfrak{p}(F) = 0$ segue da quanto osservato sopra.

Dimostriamo ora per induzione sulla complessità di F che se $\mathfrak{p}(F) = 0$ allora F è in forma prenessa. Se F è una formula atomica allora F è in forma prenessa e non c'è nulla da dimostrare. Se F è della forma $\neg G$ oppure $G * H$ con $*$ uno di $\wedge$, $\vee$ e $\rightarrow$ allora, per la seconda e terza condizione della definizione 8.23, deve essere $\mathfrak{q}(F) = 0$. Questo significa che F è priva di quantificatori e quindi in forma prenessa. Se invece F è $Qx\, G$ con Q uno di $\forall$ e $\exists$, da $\mathfrak{p}(F) = 0$ segue che $\mathfrak{p}(G) = 0$. Per ipotesi induttiva questo implica che G è in forma prenessa, da cui segue che anche F è in forma prenessa. □

Lemma 8.27. *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 $\exists$, e $*$ è uno di $\wedge$, $\vee$ e $\rightarrow$. Allora H è in forma prenessa.*

Dimostrazione. Per induzione sulla complessità di G dimostriamo che ogni sottoformula G di H soddisfa $\mathfrak{p}(G) = 0$. Da questo segue in particolare che $\mathfrak{p}(H) = 0$ e quindi, per il lemma 8.26, che H è in forma prenessa.

Se G è atomica allora $\mathfrak{p}(G) = 0$ per definizione di p-grado.

Se G è del tipo $\neg G'$ allora per ipotesi induttiva $\mathfrak{p}(G') = 0$ e per l'ipotesi del lemma G' non è del tipo $Qx\, F$. Allora deve essere anche $\mathfrak{q}(G') = 0$. Infatti G' è atomica (e quindi non contiene quantificatori) oppure una negazione, congiunzione, disgiunzione o implicazione: in questi ultimi quattro casi $\mathfrak{q}(G') > 0$ implica $\mathfrak{p}(G') > 0$. Dato che $\mathfrak{q}(G) = \mathfrak{q}(G')$ si ha $\mathfrak{p}(G) = \mathfrak{p}(G') + \mathfrak{q}(G) = 0$.

Se G è del tipo $G' * G''$ con $*$ uno di $\wedge$, $\vee$ e $\rightarrow$ il ragionamento è analogo.

Se infine G è del tipo $Qx\, G'$ si ha $\mathfrak{p}(G') = 0$ per ipotesi induttiva e quindi $\mathfrak{p}(G) = \mathfrak{p}(G') = 0$. □

Nelle edizioni precedenti al 2016-17 delle dispense il p-grado veniva definito in modo diverso, come descritto nel prossimo esercizio.

Esercizio 8.28. Sia H una formula. Ad ogni occorrenza di un quantificatore in H associamo un numero naturale, detto il *$\bar{p}$-grado* di quell'occorrenza: esso è il numero di sottoformule di H (inclusa eventualmente H stessa) che contengono quell'occorrenza del quantificatore e che sono negazioni, congiunzioni, disgiunzioni oppure implicazioni (cioè non sono quantificazioni). $\bar{\mathfrak{p}}(H)$ è la somma dei $\bar{p}$-gradi di tutte le occorrenze di quantificatori in H.

Dimostrare per induzione sulla complessità di H che $\bar{\mathfrak{p}}(H) = \mathfrak{p}(H)$ per ogni formula H.

3. L'algoritmo per la forma prenessa

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 H'' sostituendo L con una formula ad essa logicamente equivalente: nel primo caso questa formula è fornita dal lemma 8.5; nel secondo e nel terzo caso se x non è libera in G dal lemma 8.7 o dal lemma 8.9, mentre se x è libera in G si sceglie una variabile y che non compare in H' e si utilizza il lemma 8.20.

La formula H'' così ottenuta è logicamente equivalente a H' (e quindi a H) per il lemma utilizzato nella sostituzione e per il lemma 7.39.

ESEMPIO 8.30. Utilizziamo l'algoritmo 8.29 per ottenere tre diverse formule in forma prenessa logicamente equivalenti alla formula $\exists x\, p(x) \to \forall x\, \exists y\, q(x,y)$.

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

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

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

Vedremo più avanti (commento dopo la dimostrazione del lemma 8.33) come non sia un caso che le diverse esecuzioni dell'algoritmo 8.29 a partire dalla stessa formula terminino dopo lo stesso numero di passi.

ESERCIZIO 8.31. Utilizzate l'algoritmo 8.29 per trasformare in forma prenessa le seguenti formule:

$$\begin{gathered}(\forall x\, \exists y\, r(x,y) \to \exists x\, \forall y\, r(y,x)) \land \forall z\, p(w,z);\\ \exists y\, \forall x\, \left(r(x,y) \land p(f(x,y))\right) \lor \neg \left(\forall x\, \exists y\, s(x,y) \to \neg\exists x\, p(x)\right);\\ (\exists y\, q(y) \lor \exists y\, \neg r(y,y) \to \forall x\, r(c,x) \land \forall x\, p(x)) \lor \neg\exists x\, r(c,x).\end{gathered}$$

ESERCIZIO 8.32. Dimostrate che la formula dell'esempio 8.21 è logicamente equivalente alla formula in forma prenessa $\exists x\, \exists z\, \forall y\, \forall w(r(z,y) \to r(w,x))$.

Abbiamo descritto l'algoritmo per la forma prenessa 8.29 e abbiamo osservato che se arriva a una formula in forma prenessa, e quindi si arresta, allora la formula finale è logicamente equivalente alla formula da cui sono partiti. Ciò che non abbiamo ancora dimostrato, ma stiamo per fare, è che esso si arresta sempre.

LEMMA 8.33. *L'algoritmo 8.29 gode della proprietà della terminazione forte, cioè termina qualunque sia la formula su cui si decide di operare ad ogni singolo passo.*

DIMOSTRAZIONE. Dimostriamo che ogni passo dell'algoritmo abbassa di 1 il p-grado della formula ottenuta, cioè che $\mathfrak{p}(H'') = \mathfrak{p}(H') - 1$. A questo scopo fissiamo la formula L a cui applichiamo una delle equivalenze logiche dei lemmi 8.5, 8.7, 8.9, e 8.20 e procediamo per induzione sulla complessità di H'.

Il caso base è quello in cui H' coincide con L. Ad esempio osserviamo che $\mathfrak{p}(\neg\forall x\, F) = \mathfrak{p}(\forall x\, F) + \mathfrak{q}(\neg\forall x\, F) = \mathfrak{p}(F) + \mathfrak{q}(F) + 1$ mentre $\mathfrak{p}(\exists x\, \neg F) = \mathfrak{p}(\neg F) = \mathfrak{p}(F) + \mathfrak{q}(F)$. Similmente, $\mathfrak{p}(G \to \exists x\, F) = \mathfrak{p}(G) + \mathfrak{p}(\exists x\, F) + \mathfrak{q}(G \to \exists x\, F) = \mathfrak{p}(G) + \mathfrak{p}(F) + \mathfrak{q}(G) + \mathfrak{q}(F) + 1$ mentre $\mathfrak{p}(\exists y(G \to F)) = \mathfrak{p}(G \to F) = \mathfrak{p}(G) + \mathfrak{p}(F) + \mathfrak{q}(G \to F) = \mathfrak{p}(G) + \mathfrak{p}(F) + \mathfrak{q}(G) + \mathfrak{q}(F)$.

Come esempio di passo induttivo consideriamo il caso in cui H' è del tipo $G_1 * G_2$ con $*$ uno di $\land$, $\lor$ e $\to$. Supponiamo che L sia una sottoformula di G_1 e la formula H'' ottenuta sia $G_1' * G_2$: l'ipotesi induttiva ci assicura che $\mathfrak{p}(G_1') = \mathfrak{p}(G_1) - 1$. Dato che le equivalenze logiche che applichiamo non cambiano il numero di quantificatori, otteniamo $\mathfrak{p}(H'') = \mathfrak{p}(G_1') + \mathfrak{p}(G_2) + \mathfrak{q}(G_1' * G_2) = \mathfrak{p}(G_1) - 1 + \mathfrak{p}(G_2) + \mathfrak{q}(G_1 * G_2) = \mathfrak{p}(H') - 1$. Gli altri passi induttivi sono simili.

Perciò l'algoritmo non può compiere infiniti passi ma deve terminare. □

La dimostrazione del lemma 8.33 e il lemma 8.26 ci permettono di calcolare esattamente il numero di passi necessario all'algoritmo 8.29 per terminare: esso è il p-grado della formula che vogliamo trasformare in forma prenessa.

Possiamo riassumere il nostro lavoro sulla trasformazione in forma prenessa con la

Dimostrazione del teorema 8.3. Data una formula H applichiamo a partire da H l'algoritmo 8.29. Per il lemma 8.33 esso termina, producendo una formule K in forma prenessa che è logicamente equivalente a H. □

4. Usare il minimo numero di quantificatori

L'algoritmo 8.29 ci fornisce un metodo per trasformare ogni formula in una formula in forma prenessa ad essa logicamente equivalente. Utilizzando il Lemma 8.14 è però possibile migliorare i suoi risultati, in particolare ottenendo formule in forma prenessa logicamente equivalenti a quella di partenza con un prefisso più breve (cioè un minor numero di quantificatori) di quelle ottenute con l'algoritmo 8.29.

Esempio 8.34. Utilizziamo il lemma 8.14 per trasformare in forma prenessa utilizzando meno quantificatori possibili la formula $\exists x\, \forall y\, r(x,y) \vee \exists x\, \forall y\, q(x,y)$.

$$\begin{aligned}
\exists x\, \forall y\, r(x,y) \vee \exists x\, \forall y\, q(x,y) &\equiv \exists x(\forall y\, r(x,y) \vee \forall y\, q(x,y)) \\
&\equiv \exists x\, \forall y(r(x,y) \vee \forall y\, q(x,y)) \\
&\equiv \exists x\, \forall y\, \forall z(r(x,y) \vee q(x,z)).
\end{aligned}$$

La prima equivalenza logica è giustificata dal lemma 8.14, mentre successivamente non è più stato possibile utilizzare questo lemma e si sono sfruttati i lemmi 8.7 e 8.20.

Ripetiamo la stessa operazione con $\exists x\, \forall y\, r(x,y) \wedge \exists x\, \forall y\, q(x,y)$

$$\begin{aligned}
\exists x\, \forall y\, r(x,y) \wedge \exists x\, \forall y\, q(x,y) &\equiv \exists x(\forall y\, r(x,y) \wedge \exists x\, \forall y\, q(x,y)) \\
&\equiv \exists x\, \exists z(\forall y\, r(x,y) \wedge \forall y\, q(z,y)) \\
&\equiv \exists x\, \exists z\, \forall y(r(x,y) \wedge q(z,y)).
\end{aligned}$$

In questo caso si sono sfruttati i lemmi 8.7 e 8.20 per le prime due equivalenze, mentre il lemma 8.14 giustifica l'ultima equivalenza logica.

La seconda trasformazione dell'esempio 8.34 evidenzia come per poter sfruttare il lemma 8.14 sia necessario intervenire in un ordine opportuno sui vari quantificatori: se dopo la prima equivalenza si fosse operato sul quantificatore $\forall y$ del primo congiunto, l'opportunità di usare successivamente il lemma 8.14 sarebbe sfumata e la formula in forma prenessa finale avrebbe avuto un prefisso di quattro quantificatori. Comunque questo non significa che vi sia unicità della forma prenessa con minimo numero di quantificatori: basta considerare l'esempio 8.4.

Il lemma 8.14 si limita a considerare il caso in cui la variabile su cui si effettua la quantificazione sia la stessa in entrambe le sottoformule della formula in considerazione. Esso può venir agevolmente generalizzato al caso in cui le due variabili siano diverse.

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:*

$$\begin{aligned}\forall x\, F \wedge \forall y\, G &\equiv \forall z(F\{x/z\} \wedge G\{y/z\}),\\ \exists x\, F \vee \exists y\, G &\equiv \exists z(F\{x/z\} \vee G\{y/z\}),\\ \forall x\, F \to \exists y\, G &\equiv \exists z(F\{x/z\} \to G\{y/z\}).\end{aligned}$$

DIMOSTRAZIONE. Le tre equivalenze logiche si ottengono in modo simile. A titolo di esempio consideriamo la terza equivalenza logica:

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

dove la prima equivalenza è giustificata dal lemma 7.61 (e dal lemma 7.39), mentre la seconda equivalenza sfrutta l'equivalenza corrispondente del lemma 8.14. □

Spesso nelle applicazioni del lemma 8.35 la variabile z può essere scelta come x oppure y.

ESEMPIO 8.36. Consideriamo nuovamente la formula dell'esempio 8.19 ed utilizziamo il lemma 8.35 per trasformarla in forma prenessa usando il minimo numero di quantificatori. Per diminuire la lunghezza dell'esempio effettuiamo più passaggi simultaneamente:

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

Nel passaggio dalla terza alla quarta riga abbiamo utilizzato il lemma 8.35, scegliendo come variabile x.

ESERCIZIO 8.37. Mettete le formule dell'esempio 8.21 e dell'esercizio 8.31 in forma prenessa, usando il minimo numero di quantificatori possibili.

Combinando gli algoritmi descritti in questo capitolo con l'algoritmo 3.18 per la trasformazione in forma normale congiuntiva è possibile trasformare qualunque formula in una formula logicamente equivalente in forma prenessa con matrice in forma normale congiuntiva. Questa "forma normale" per formule predicative è utile in diverse applicazioni, che purtroppo non abbiamo il tempo di descrivere in queste dispense.

CAPITOLO 9

Traduzioni dal linguaggio naturale

In questo capitolo ci occuperemo di tradurre frasi del linguaggio naturale (nel nostro caso, l'italiano) in formule (in realtà enunciati) e viceversa. Il problema analogo per la logica proposizionale è stato trattato nella sezione 2.5: la ricchezza espressiva della logica predicativa ci permette di tradurre in modo più preciso molte frasi del linguaggio naturale.

Il metodo migliore per familiarizzarsi con queste traduzioni è naturalmente la pratica. Per questa ragione questo capitolo (così come la sezione 2.5) consiste prevalentemente di esempi ed esercizi.

Anche nel caso predicativo la traduzione dal linguaggio formale al linguaggio naturale non presenta in genere difficoltà, mentre la direzione inversa è spesso più delicata.

1. Traduzioni di frasi

Esempio 9.1. Sia $\{a, b, v, d\}$ un linguaggio in cui a e b sono simboli di costante (da interpretarsi rispettivamente come "Andrea" e "Bruna") e v e d sono simboli di relazione unari ($v(x)$ da interpretarsi come "x va alla festa", $d(x)$ come "x si diverte").

La formula $v(b) \wedge d(b) \to \neg v(a)$ viene interpretata come "se Bruna va alla festa e si diverte allora Andrea non va alla festa".

La frase "Andrea va alla festa e Bruna no, oppure Andrea va alla festa e Bruna si diverte" viene tradotta nella formula $(v(a) \wedge \neg v(b)) \vee (v(a) \wedge d(b))$.

La formula $\exists x(v(x) \wedge \neg d(x))$ viene interpretata come "qualcuno va alla festa e non si diverte" oppure "c'è qualcuno che non si diverte, pur andando alla festa".

La frase "tutti quelli che vanno alla festa si divertono" viene tradotta nella formula $\forall x(v(x) \to d(x))$, che traduce anche "chi va alla festa si diverte".

Nota 9.2. Per tradurre in un linguaggio proposizionale la prima frase dell'esempio 9.1 avremmo scelto alcune lettere proposizionali (ad esempio p per "Bruna va alla festa", q per "Bruna si diverte" e r per "Andrea va alla festa") e la traduzione sarebbe stata $p \wedge q \to \neg r$. In questa formula proposizionale non c'è traccia né del fatto che Bruna è l'oggetto delle prime due affermazioni atomiche né del fatto che Andrea non fà ciò che fà Bruna (divertirsi).

L'espressività della logica predicativa emerge quindi anche se non si usano quantificatori, ma è ancora più evidente nel caso di espressioni che riguardano la totalità degli oggetti considerati. Come notato nell'introduzione del Capitolo 1, l'unico modo per tradurre "tutti quelli che vanno alla festa si divertono" nella logica proposizionale è quello di introdurre una lettera proposizionale che corrisponda a questa affermazione. La traduzione predicativa $\forall x(v(x) \to d(x))$ è invece molto più espressiva.

Esempio 9.3. Utilizziamo il linguaggio $\{m, d, c, s\}$ dove m e d sono simboli di relazione unari ($m(x)$ sta per "x è un malato", $d(x)$ sta per "x è un dottore") e c e s sono simboli di relazione binari ($c(x, y)$ sta per "x cura y" e $s(x, y)$ sta per "x stima y").

"Ogni malato non stima se stesso" è tradotto da

$$\forall x(m(x) \to \neg s(x,x)).$$

"Ci sono dottori che curano se stessi" è tradotto da

$$\exists x(d(x) \wedge c(x,x)).$$

"Qualche malato stima tutti i dottori che lo curano" è tradotto da

$$\exists x(m(x) \wedge \forall y(d(y) \wedge c(y,x) \to s(x,y))).$$

"Tutti i malati stimano almeno un dottore che li cura" è tradotto da

$$\forall x(m(x) \to \exists y(d(y) \wedge c(y,x) \wedge s(x,y))).$$

Notate come l'ultima formula sia logicamente equivalente (per il lemma 8.9) a

$$\forall x\, \exists y(m(x) \to d(y) \wedge c(y,x) \wedge s(x,y)),$$

che può quindi essere considerata un'altra traduzione corretta della frase in esame.

Nota 9.4. Questi esempi evidenziano come in queste traduzioni $\forall$ spesso preceda un'implicazione (e l'antecedente dell'implicazione restringe l'ambito degli elementi a cui si applica $\forall$, nel caso della prima frase dell'esempio 9.3 ai malati), mentre $\exists$ spesso preceda una congiunzione (l'elemento di cui si asserisce l'esistenza ha spesso diverse proprietà, nel caso della seconda frase oltre ad essere un dottore ha la caratteristica di curare se stesso).

Se in una traduzione dal linguaggio naturale si ottiene una quantificazione universale di una congiunzione, o una quantificazione esistenziale di un'implicazione, è bene controllare accuratamente la propria soluzione.

Esercizio 9.5. Consideriamo il linguaggio $\{s,c,f,u\}$, dove s, c e f sono simboli di relazione unari (da interpretarsi come "x è uno studente", "x è un computer" e "x è funzionante") mentre u è un simbolo di relazione binario (da interpretarsi come "x utilizza y"). Considerate le seguenti frasi:

(i) Un computer non è utilizzato da nessuno studente.
(ii) Ogni computer funzionante è utilizzato da almeno uno studente.
(iii) Non tutti i computer sono funzionanti.

Quale dei seguenti enunciati è una traduzione di (i)?

$$\exists x(c(x) \wedge \forall y(\neg s(y) \wedge \neg u(y,x)));$$
$$\exists x(c(x) \to \forall y(s(y) \to \neg u(y,x)));$$
$$\exists x(c(x) \wedge \forall y(s(y) \to \neg u(y,x))).$$

Qual è il significato degli altri enunciati? Notate che uno di essi è soddisfatto in ogni interpretazione I tale che $c^I \neq D^I$, mentre se l'altro è soddisfatto in un'interpretazione J allora deve essere $s^J = \emptyset$. Traducete (ii) e (iii).

Per ora abbiamo considerato solo linguaggi privi di simboli di funzione. Ecco un linguaggio con simboli di funzione.

Esempio 9.6. Sia $\{c,p,r,s\}$ un linguaggio in cui c è un simbolo di costante, p è un simbolo di funzione unario, r e s sono simboli di relazione binari. Interpretiamo c come "Claudio", $p(x)$ come "il padre di x", $r(x,y)$ come "x è parente di y" e $s(x,y)$ come "x stima y".

"Tutti i parenti di Claudio stimati da Claudio, sono stimati anche dal padre di Claudio" è tradotta da

$$\forall x(r(x,c) \wedge s(c,x) \to s(p(c),x)).$$

"Claudio stima se stesso e tutti quelli che stimano suo padre, ma non stima suo padre" è tradotta da

$$s(c,c) \wedge \forall x(s(x,p(c)) \to s(c,x)) \wedge \neg s(c,p(c)).$$

"Claudio stima solo quelli che stimano il loro nonno paterno" è tradotta da

$$\forall x(s(c,x) \to s(x,p(p(x)))).$$

Esempio 9.7. Sia $\{b,f,p,c,g,a\}$ un linguaggio dove b e f sono simboli di costante, p è un simbolo di funzione unario, c e g sono simboli di relazione unari, e a è un simbolo di relazione binario. Interpretiamo b come "Bobi", f come "Fifi", $p(x)$ come "il padrone di x", $c(x)$ come "x è un cane", $g(x)$ come "x è un gatto", $a(x,y)$ come "x ama y".

"Bobi non ama il padrone di Fifi" è tradotta da

$$\neg a(b,p(f)).$$

"Tutti i cani e i gatti amano i loro padroni" è tradotta da

$$\forall x(c(x) \vee g(x) \to a(x,p(x)))$$

oppure da

$$\forall x(c(x) \to a(x,p(x))) \wedge \forall x(g(x) \to a(x,p(x))).$$

Notiamo come nel primo enunciato "e" sia stato tradotto da $\vee$. Infatti l'enunciato

$$\forall x(c(x) \wedge g(x) \to a(x,p(x)))$$

corrisponde all'affermazione "tutti coloro che sono sia cane che gatto amano i loro padroni", che ha un significato ben diverso.

"Tutti i cani non amano i padroni di un gatto" è tradotta da

$$\forall x(c(x) \to \forall y(g(y) \to \neg a(x,p(y)))),$$

oppure da

$$\forall x\, \forall y(c(x) \wedge g(y) \to \neg a(x,p(y)))$$

(i due enunciati sono logicamente equivalenti).

Esercizio 9.8. Formalizzate "se ogni sorella di Gianni litiga con almeno una sorella di Fabio, il miglior amico di Gianni litiga con il miglior amico di Fabio" utilizzando il linguaggio $\{g,f,s,m,l\}$ dove g,f sono costanti (che denotano rispettivamente "Gianni" e "Fabio"), m è un simbolo di funzione unario ($m(x)$ sta per "il miglior amico di x"), s e l sono simboli di relazione binari ($s(x,y)$ sta per "x è sorella di y" e $l(x,y)$ sta per "x litiga con y").

Esempio 9.9. Utilizziamo il linguaggio dell'esempio 6.4. Ecco alcune traduzioni:

(a) "su qualche numero reale seno e coseno coincidono"

$$\exists x\ \sin(x) = \cos(x);$$

(b) "esistono due numeri reali distinti tali che il seno del primo è minore del coseno del secondo"

$$\exists x\, \exists y(x \neq y \wedge \sin(x) < \cos(y)).$$

Esercizio 9.10. Traducete le frasi seguenti utilizzando il linguaggio dell'esempio 6.5:

(a) "la stringa vuota è segmento iniziale di ogni stringa";
(b) "ogni stringa è segmento iniziale della sua concatenazione con un'altra stringa";
(c) "ogni stringa è segmento iniziale della sua concatenazione con un'altra stringa o con se stessa";

(d) “esistono due stringhe che non sono l’una segmento iniziale dell’altra”;
(e) “qualche stringa ha tutte le stringhe come segmento iniziale”.

ESERCIZIO 9.11. Traducete le frasi seguenti utilizzando il linguaggio dell’esempio 6.6:

(a) “un sottoinsieme di un insieme appartiene all’insieme delle parti di quell’insieme”;
(b) “esiste un insieme i cui elementi sono anche suoi sottoinsiemi”.

ESERCIZIO 9.12. Introducendo opportuni linguaggi, traducete le frasi seguenti:

(a) “Se tutti gli uomini sono mortali e Socrate è un uomo, allora Socrate è mortale”.
(b) “Se tutti i gatti sono animali e Fifi è un gatto, allora Fifi è un animale” (confrontate questa traduzione con quella di (a)).
(c) “Se ogni amico di Mario è amico di Luca e Pietro non è amico di Mario, allora Pietro non è amico di Luca”.
(d) “Nessun ladro è onesto”.
(e) “Se tutti i filosofi intelligenti sono curiosi e solo i tedeschi sono filosofi intelligenti, allora, se ci sono filosofi intelligenti, qualche tedesco è curioso”.
(f) “Il cervello di un delfino è più grande di quello di un topo”.
[Suggerimento: utilizzate il linguaggio contenente il simbolo di funzione unario c (“il cervello di x”), il simbolo di relazione binario g (“x è più grande di y”) e i simboli di relazione unari d e t (“x è un delfino” e “x è un topo”).]
(g) “Se Carlo è più basso di Luca, allora almeno un amico di Carlo è più basso di tutti gli amici di Luca”.
(h) “Se tutti gli studenti sono persone serie, tutti gli studenti sono studiosi e tutte le persone serie e studiose non fanno tardi la sera, allora se esiste qualcuno che fa tardi la sera, non tutti sono studenti”.

ESERCIZIO 9.13. Utilizzando il linguaggio $\{u, c, d, a\}$ dove u è un simbolo di funzione unario ($u(x)$ sta per “l’ultimo cd di x”), c è un simbolo di relazione unario ($c(x)$ sta per “x è un cantante”) e d, a sono simboli di relazione binari ($d(x, y)$ sta per “x è un cd di y” e $a(x, y)$ sta per “x acquista y”), formalizzate le frasi seguenti:

(a) “qualcuno acquista tutti i cd di qualche cantante”;
(b) “ogni cd di un cantante è sempre acquistato da qualcuno”;
(c) “l’ultimo cd di un cantante è sempre acquistato da qualcuno”.

ESERCIZIO 9.14. Introducendo un linguaggio opportuno, traducete le frasi seguenti (che conducono a formule prive di quantificatori e con variabili libere): “x^2 è pari, se x è pari”, “una condizione sufficiente affinché x sia dispari é che x sia primo”, “una condizione necessaria affinché x^2 sia pari è che x non sia primo”.
[Suggerimento: utilizzate i simboli di relazione $p(x)$ (“x è pari”), $pr(x)$ (“x è primo”), il simbolo funzionale $f(x)$ (“il quadrato di x”) e considerate “dispari” come la negazione di “pari”].

ESERCIZIO 9.15. Considerate il linguaggio $\{b, d, c, t, a\}$, dove b e d sono simboli di costante che rappresentano Barbara e Donatella, c e t sono simboli di relazione unari ($c(x)$ sta per “x ama il cinema”, $t(x)$ sta per “x ama il teatro”), mentre a è un simbolo di relazione binario ($a(x, y)$ sta per “x è amico di y”). Formalizzate nel linguaggio le seguenti frasi:

(a) Chi è amico di qualcuno che ama il cinema, ama il cinema.
(b) Chi ama il teatro, è amico di qualcuno che ama il teatro.
(c) Barbara è amica di Donatella e ama il teatro, ma non il cinema.

ESERCIZIO 9.16. ($\star$) Sia $\mathcal{L} = \{r\}$ dove r è un simbolo di relazione binario. Formalizzate in $\mathcal{L}$ le seguenti proprietà:

(a) r è transitiva;
(b) r non è riflessiva su alcun punto.

Siano F_a e F_b gli enunciati corrispondenti. Dimostrate che l'enunciato

$$\forall x\,\exists y\, r(x,y) \wedge F_a \wedge F_b$$

non è vero in nessuna interpretazione con dominio finito, ma è soddisfacibile.

2. Traduzioni di argomenti

La nozione di conseguenza logica ci permette di analizzare la correttezza dei ragionamenti, come discusso nell'introduzione.

Esempio 9.17. Consideriamo le frasi seguenti:

(i) Tutti gli attori ed i giornalisti invitati alla festa sono in ritardo.
(ii) Qualcuno è puntuale.
(iii) Qualche invitato non è né attore né giornalista.

Utilizziamo il linguaggio $\{a, g, i, p\}$, dove $a(x)$ sta per "x è un attore", $g(x)$ sta per "x è un giornalista", $i(x)$ sta per "x è invitato alla festa" e $p(x)$ sta per "x è puntuale". Le frasi precedenti vengono tradotte rispettivamente da

$$\forall x\,((a(x) \vee g(x)) \wedge i(x) \rightarrow \neg p(x))\,; \quad \exists x\, p(x); \quad \exists x(i(x) \wedge \neg a(x) \wedge \neg g(x)).$$

(Notate che in (i) "e" è stato tradotto da $\vee$: vedere l'esempio 9.7.)

Indicando con F, G e H queste tre traduzioni abbiamo che $F, G \nvDash H$ (trovate un'interpretazione che lo mostri!) e quindi possiamo dire che la frase in (iii) **non** segue logicamente dalle frasi in (i) e (ii), e quindi il ragionamento che deduce (iii) a partire da (i) e (ii) non è corretto.

Esercizio 9.18. Se nell'esempio precedente (ii) venisse sostituita da "qualche invitato è puntuale" il ragionamento diventerebbe corretto?

Esercizio 9.19. Esaminate alla luce di ciò che avete imparato sinora gli esempi di pagina 1, stabilendo rigorosamente quali dei tre argomenti sono corretti.

Esercizio 9.20. Utilizzando il linguaggio $\{s, p, v, g\}$ dove i quattro simboli sono simboli di relazione predicati unari, da interpretarsi come come "x è stupido", "x è presuntuoso", "x è vanitoso", e "x è simpatico", ed inoltre considerando "antipatico" come la negazione di "simpatico", formalizzate le frasi seguenti:

(i) tutti gli stupidi sono presuntuosi o vanitosi;
(ii) i presuntuosi sono antipatici;
(iii) le persone simpatiche non sono vanitose;
(iv) tutti gli stupidi sono antipatici.

Possiamo dire che la frase in (iv) segue logicamente dalle precedenti?

Esercizio 9.21. Formalizzate in un linguaggio opportuno il seguente argomento: "alcuni studenti apprezzano tutti i professori, ogni studente non apprezza i ciarlatani, dunque nessun professore è un ciarlatano". Verificate la correttezza dell'argomento.

Esercizio 9.22. ($\star$) Sia r un simbolo di relazione binario e sia F l'enunciato $\forall x(\exists y\, r(x,y) \rightarrow r(x,x))$.

(a) Dimostrate che F è vero in ogni interpretazione I in cui r^I è una relazione simmetrica e transitiva.
(b) Scrivete un enunciato G che esprima il fatto che r è simmetrica e transitiva. Stabilite se $G \vDash F$.

Esercizio 9.23. Formalizzate in un opportuno linguaggio le frasi seguenti:

(a) Tutti coloro che scendono dall'aereo tranne i membri dell'equipaggio sono perquisiti da almeno un poliziotto.
(b) Alcuni ladri scendono dall'aereo e sono perquisiti solo da ladri.
(c) Nessun ladro è membro dell'equipaggio.
(d) Alcuni poliziotti sono ladri.

Siano F_a, F_b, F_c e F_d gli enunciati corrispondenti: stabilite se $F_a \wedge F_b \wedge F_c \vDash F_d$.

3. Traduzioni con uguaglianza

La presenza dell'uguaglianza in un linguaggio ci permette di estendere le frasi che possiamo tradurre.

Esempio 9.24. Consideriamo il linguaggio con uguaglianza $\{a, b, m, c, =\}$ dove a e b sono simboli di costante, m è un simbolo di funzione unario e c un simbolo di relazione binario. Supponiamo che a e b rappresentino Anna e Barbara, $m(x)$ "la madre di x" e $c(x, y)$ significhi "x conosce y". Ecco alcune traduzioni in questo linguaggio:

- Barbara non è madre di nessuno;
$$\forall x\, m(x) \neq b$$
- Anna è la madre di qualcuno;
$$\exists x\, m(x) = a$$
- Barbara è l'unica figlia di Anna;
$$m(b) = a \wedge \forall x(x \neq b \rightarrow m(x) \neq a) \quad \text{oppure} \quad m(b) = a \wedge \forall x(m(x) = a \rightarrow x = b)$$
- Anna ha almeno due figli;
$$\exists x\, \exists y(m(x) = a \wedge m(y) = a \wedge x \neq y)$$
- Anna ha al più due figli;
$$\forall x\, \forall y\, \forall z(m(x) = a \wedge m(y) = a \wedge m(z) = a \rightarrow x = y \vee y = z \vee x = z)$$
- Anna ha esattamente due figli;
si può usare la congiunzione delle due formule precedenti, oppure
$$\exists x\, \exists y(m(x) = a \wedge m(y) = a \wedge x \neq y \wedge \forall z(m(z) = a \rightarrow z = x \vee z = y))$$
- i figli di Anna conoscono Barbara;
$$\forall x(m(x) = a \rightarrow c(x, b))$$
- ogni figlio di Anna conosce un figlio di Barbara;
$$\forall x(m(x) = a \rightarrow \exists y(m(y) = b \wedge c(x, y)))$$
- un figlio di Barbara è conosciuto da tutti i figli di Anna;
$$\exists y(m(y) = b \wedge \forall x(m(x) = a \rightarrow c(x, y)))$$

Esercizio 9.25. Nel linguaggio dell'esempio 9.24 traducete le seguenti frasi:
(a) Anna e Barbara hanno la stessa madre;
(b) Anna è l'unica a non conoscere la madre di Barbara;
(c) due figli di Barbara conoscono Anna;
(d) la madre di Anna ha un altro figlio.

Esercizio 9.26. Nel linguaggio con uguaglianza $\{a, f, r, =\}$, dove a è un simbolo di costante, f un simbolo funzionale unario, r un simbolo di relazione binario, traducete le frasi seguenti:
(a) "f è una funzione suriettiva ma non iniettiva";

(b) "ogni elemento del dominio è in relazione r con la sua immagine secondo f";
(c) "a non appartiene all'immagine di f".

Esercizio 9.27. Nel linguaggio con uguaglianza $\{2, \times, |, =\}$ dove 2 è un simbolo di costante (che sta per il numero naturale 2), $\times$ è un simbolo funzionale binario ($x \times y$ sta per "il prodotto di x ed y"), e $|$ è un simbolo di relazione binario ($x|y$ sta per "x divide y"), traducete le frasi seguenti:
(a) "x è pari";
(b) "x è il quadrato di qualche numero";
(c) "x è un numero composto";
(d) "x è un numero primo";
(e) "4 è un numero primo";
(f) "se un numero primo divide un prodotto, allora divide uno dei fattori";
(g) "il quadrato di ogni numero pari diverso da due non è primo";
(h) "ogni numero che dividendo un prodotto divide uno dei fattori, è primo".

Esercizio 9.28. Nel linguaggio con uguaglianza $\{a, b, d, m, p, =\}$, dove a e b sono simboli di costante da interpretarsi come "Andrea" e "Barbara", d e m sono simboli di funzione unari da interpretarsi come "il dentista di x", e "la madre di x" e p è un simbolo di relazione binario da interpretarsi come "x è parente di y", traducete le frasi:
(a) La madre di Andrea è parente del dentista di Barbara, che è anche il dentista di Andrea;
(b) Il dentista di Andrea è il dentista di tutti i parenti di Barbara, ad eccezione della madre di Barbara, che ha un altro dentista.

Esercizio 9.29. Nel linguaggio dell'esercizio 9.13 cui avete aggiunto il simbolo $=$, traducete "qualcuno acquista l'ultimo cd di qualche cantante, ma non gli altri cd dello stesso cantante".

4. Traduzioni in linguaggi multisorta

Se consideriamo le interpretazioni di alcuni degli esempi precedenti possiamo notare che il linguaggio utilizzato ci obbliga a definire l'interpretazione di simboli di funzione anche per elementi del dominio in cui non pare che ciò abbia senso. Un'interpretazione I del linguaggio dell'esempio 9.7 contiene la definizione di $p^I(d)$ (che significa "il padrone di d") per ogni $d \in D^I$. Dato che in D^I sono compresi non solo cani e gatti, ma anche esseri umani (come presumibilmente è $p^I(b^I)$), ciò può lasciare perplessi.

Un modo di risolvere questo problema è quello di definire in maniera arbitraria $p^I(d)$ quando d non ha padrone. In questo caso è opportuno considerare solamente formule la cui verità dipende solo dal valore assunto da p su elementi del dominio che denotano individui con padrone (di questo tipo sono le formule ottenute nell'esempio 9.7).

Un secondo approccio è quello dei linguaggi multisorta, in cui gli elementi del dominio di un'interpretazione sono divisi in diverse sorte (dette anche specie o tipi). Quindi anche i simboli del linguaggio devono essere associati a queste sorte.

Ci limitiamo ad introdurre i linguaggi multisorta attraverso un esempio.

Esempio 9.30. Nel caso dell'esempio 9.7 è naturale considerare un linguaggio con due sorte: ANIMALI e UMANI.

Ogni variabile appartiene a una e una sola di queste sorte ed è comodo usare ad esempio variabili minuscole per gli ANIMALI e variabile maiuscole per gli UMANI. Anche i simboli di costante appartengono ad una sorte (b e f appartengono alla sorte ANIMALI). Per ogni simbolo funzionale è necessario specificare tra quali sorte esso agisce. Ad esempio p associa ad un oggetto di sorta ANIMALI un oggetto della sorta UMANI. In questo modo ogni termine appartiene ad una sorta e ad esempio $p(b)$ appartiene alla sorta UMANI. Quindi $p(p(b))$ non è un termine del linguaggio multisorta (perché p non si applica al termine $p(b)$ che appartiene alla sorta UMANI), e perciò non deve essere interpretato. Similmente è necessario stabilire per ogni simbolo di relazione ai termini di quali sorte può essere applicato per formare una formula atomica. Ad esempio possiamo stabilire che c e g sono simboli di relazione per la sorta ANIMALI mentre a è un simbolo di relazione che collega un termine della sorta ANIMALI con uno della sorta UMANI.

In questo linguaggio (ed utilizzando anche l'uguaglianza) "tutti i cani amano chi non è padrone di un gatto" viene tradotto da

$$\forall x(c(x) \to \forall Y(\neg\exists z(g(z) \wedge p(z) = Y) \to a(x, Y))).$$

Notate l'uso della variabile maiuscola per riferirsi ad un essere umano.

ESERCIZIO 9.31. Descrivete un linguaggio multisorta per il linguaggio dell'esercizio 9.13, in cui non ha senso parlare del "cd di un cd".

ESERCIZIO 9.32. Formalizzate in un linguaggio multisorta la frase: "Se ogni scrittore è apprezzato da almeno un lettore, ed ogni lettore legge i libri degli scrittori che apprezza, allora ogni libro è letto da qualcuno".

Un linguaggio multisorta può comunque essere ricondotto ad un linguaggio predicativo che rispetta la definizione 6.2. Anche in questo caso ci limitiamo a spiegare il procedimento attraverso un esempio.

ESEMPIO 9.33. Consideriamo il linguaggio multisorta definito nell'esempio 9.30. Per ricondurlo ad un linguaggio predicativo introduciamo due nuovi simboli di relazione unari an e u, dove $an(x)$ e $u(x)$ sono interpretati rispettivamente come "x è un animale" e "x è un umano". La formula

$$\forall X\, \exists y\, p(y) = X$$

diventa allora un'abbreviazione per

$$\forall x(u(x) \to \exists y(an(y) \wedge p(y) = x)).$$

Nella semantica di questo linguaggio si considerano solo interpretazioni che soddisfano l'enunciato

$$\forall x((an(x) \vee u(x)) \wedge \neg(an(x) \wedge u(x)))$$

che esprime che ogni individuo appartiene ad esattamente una sorta.

CAPITOLO 10

Interpretazioni elementarmente equivalenti

In questo capitolo studieremo una importante relazione che può intercorrere tra diverse interpretazioni per lo stesso linguaggio. Nella prima sezione definiremo la nozione di equivalenza elementare e individueremo una condizione sufficiente per essa (corollario 10.14), nella seconda sezione introdurremo il concetto di congruenza e nella terza sezione useremo questi due strumenti per ottenere alcuni risultati relativi alla logica con uguaglianza.

1. Equivalenza elementare e omomorfismi forti

Due interpretazioni per lo stesso linguaggio sono simili se soddisfano gli stessi enunciati.

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 $I \vDash F$ se e solo se $J \vDash F$. 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 $I \vDash F$ implica $J \vDash F$ allora $I \equiv_{\mathcal{L}} J$.
[Suggerimento: se F è un enunciato anche $\neg F$ è un enunciato.]

L'essere elementarmente equivalenti è una relazione d'equivalenza tra le interpretazioni dello stesso linguaggio, come è immediato verificare.

Un'altra relazione che può intercorrere tra due interpretazioni per lo stesso linguaggio è l'esistenza di una funzione che "rispetti" le interpretazioni dei vari simboli del linguaggio. Questa idea è catturata dalla seguente definizione.

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 $(\varphi(d_1), \dots, \varphi(d_n)) \in p^J$.

φ è un *omomorfismo forte* se la terza condizione è sostituita da

- per ogni simbolo di relazione n-ario p di $\mathcal{L}$ ed ogni $d_1, \dots, d_n \in D^I$ si ha che $(d_1, \dots, d_n) \in p^I$ se e solo se $(\varphi(d_1), \dots, \varphi(d_n)) \in p^J$.

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

NOTA 10.4. Per dimostrare che un omomorfismo φ è un omomorfismo forte è sufficiente verificare che per ogni simbolo di relazione n-ario p di $\mathcal{L}$ ed ogni $d_1, \dots, d_n \in D^I$ tali che $(d_1, \dots, d_n) \notin p^I$ si ha $(\varphi(d_1), \dots, \varphi(d_n)) \notin p^J$.

Due interpretazioni isomorfe sono sostanzialmente la stessa interpretazione: ciò che cambia è solo il nome degli elementi del dominio, secondo la corrispondenza descritta dall'isomorfismo. Sembra naturale congetturare che se $I \cong J$ allora $I \equiv_{\mathcal{L}}$

J. Nel corollario 10.14 dimostreremo una versione più forte (ottenuta indebolendo l'ipotesi) di questa congettura.

L'essere isomorfi è una relazione d'equivalenza tra le interpretazioni di un linguaggio fissato (si vede facilmente che l'identità è un isomorfismo di un'interpretazione con se stessa, che la funzione inversa di un isomorfismo è un isomorfismo e che la composizione di due isomorfismi è un isomorfismo).

Esempio 10.5. Sia $\mathcal{L}$ il linguaggio che consiste del solo simbolo di relazione unario p. Definiamo tre interpretazioni I, J e K per $\mathcal{L}$ ponendo:

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

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

Esercizio 10.6. Sia $\mathcal{L}'$ il linguaggio ottenuto da $\mathcal{L}$ dell'esempio precedente aggiungendo un simbolo di costante c. Estendiamo le interpretazioni I, J e K dell'esempio precedente a interpretazioni per $\mathcal{L}'$ ponendo $c^{I'} = A$, $c^{J'} = 0$ e $c^{K'} = 31$. Dimostrate:

(a) non esiste un omomorfismo di I' in J';
(b) esiste un omomorfismo di J' in I', ma non esiste un omomorfismo forte di J' in I';
(c) esiste un omomorfismo forte di K' in I';
(d) esiste un omomorfismo di I' in K', ma non esiste un omomorfismo forte di I' in K';
(e) non esiste un omomorfismo di K' in J';
(f) esiste un omomorfismo di J' in K', ma non esiste un omomorfismo forte di J' in K'.

Esempio 10.7. Supponiamo che $\mathcal{L}$ sia un linguaggio con uguaglianza e I e J siano interpretazioni normali per $\mathcal{L}$. Sia φ un omomorfismo forte di I in J. L'ultima condizione nella definizione di omomorfismo forte, applicata al simbolo di relazione $=$, fa sì che per ogni $d_1, d_2 \in D^I$ tali che $d_1 = d_2$ si abbia $\varphi(d_1) = \varphi(d_2)$. Perciò ogni omomorfismo forte di I in J è un'iniezione.

Definizione 10.8. Siano I e J due interpretazioni per lo stesso linguaggio $\mathcal{L}$ e sia $\varphi : D^I \to D^J$ una funzione. Allo stato σ di I corrisponde uno stato $\varphi \circ \sigma$ di J, ottenuto componendo σ e φ, 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 $\sigma : \mathrm{Var} \to D^I$, si ha che $\varphi \circ \sigma : \mathrm{Var} \to D^J$.

Lemma 10.9. *Siano I e J due interpretazioni per un linguaggio $\mathcal{L}$, σ uno stato di I. Se φ è un omomorfismo di I in J e t è un termine di $\mathcal{L}$ allora $\varphi(\sigma(t)) = (\varphi \circ \sigma)(t)$.*

Dimostrazione. Procediamo per induzione sulla complessità del termine t.

Se t è una variabile allora $\varphi(\sigma(t)) = (\varphi \circ \sigma)(t)$ per definizione.

Se t è un simbolo di costante c allora $\varphi(\sigma(t)) = \varphi(c^I) = c^J = (\varphi \circ \sigma)(t)$ per la prima condizione nella definizione di omomorfismo.

Se t è $f(t_1, \ldots, t_n)$ allora

$$\begin{aligned}\varphi(\sigma(t)) &= \varphi(f^I(\sigma(t_1), \ldots, \sigma(t_n)))\\ &= f^J(\varphi(\sigma(t_1)), \ldots, \varphi(\sigma(t_n)))\\ &= f^J((\varphi \circ \sigma)(t_1), \ldots, (\varphi \circ \sigma)(t_n))\\ &= (\varphi \circ \sigma)(t),\end{aligned}$$

dove nel passaggio dalla prima alla seconda riga abbiamo usato la seconda condizione nella definizione di omomorfismo e nel passaggio dalla seconda alla terza riga l'ipotesi induttiva. □

LEMMA 10.10. *Siano I e J due interpretazioni per un linguaggio $\mathcal{L}$, σ uno stato di I. Se φ è un omomorfismo forte di I in J e F è una formula priva di quantificatori allora $I, \sigma \vDash F$ se e solo se $J, \varphi \circ \sigma \vDash F$.*

DIMOSTRAZIONE. Procediamo per induzione sulla complessità della formula priva di quantificatori F.

Se F è la formula atomica $p(t_1, \ldots, t_n)$ abbiamo

$$\begin{aligned}I, \sigma \vDash F &\text{ se e solo se } (\sigma(t_1), \ldots, \sigma(t_n)) \in p^I\\ &\text{ se e solo se } (\varphi(\sigma(t_1)), \ldots, \varphi(\sigma(t_n))) \in p^J\\ &\text{ se e solo se } ((\varphi \circ \sigma)(t_1), \ldots, (\varphi \circ \sigma)(t_n)) \in p^J\\ &\text{ se e solo se } J, \varphi \circ \sigma \vDash F,\end{aligned}$$

dove nel passaggio dalla prima alla seconda riga abbiamo usato la terza condizione nella definizione di omomorfismo forte e nel passaggio dalla seconda alla terza riga quanto appena dimostrato sui termini.

Dato che ci interessano solo le formule prive di quantificatori, i passi induttivi da considerare sono quelli in cui F è una negazione, una congiunzione, una disgiunzione o un'implicazione. Questi passi sono tutti immediati, utilizzando la definizione 7.8 e l'ipotesi induttiva. □

ESEMPIO 10.11. Il lemma 10.10 non può essere esteso alle formule che contengono quantificatori: se K e I sono le interpretazioni dell'esempio 10.5 si ha che $I \vDash \exists x\, \neg p(x)$ mentre $K \nvDash \exists x\, \neg p(x)$ (dato che abbiamo a che fare con un enunciato non è necessario menzionare lo stato), malgrado l'esistenza di un omomorfismo forte di K in I. (Notate che abbiamo dimostrato che $I \not\equiv_{\mathcal{L}} K$.)

Per estendere il lemma 10.10 a tutte le formule è necessario rafforzare l'ipotesi sull'omomorfismo. A questo scopo ci sarà utile il seguente lemma.

LEMMA 10.12. *Siano I e J due interpretazioni per un linguaggio $\mathcal{L}$, $\varphi : D^I \to D^J$ una funzione qualsiasi e σ uno stato di I. Per ogni $x \in \mathrm{Var}$ e ogni $d \in D^I$ gli stati $\varphi \circ (\sigma[x/d])$ e $(\varphi \circ \sigma)[x/\varphi(d)]$ di J coincidono.*

DIMOSTRAZIONE. Basta dimostrare che per ogni $v \in \mathrm{Var}$ si ha $\varphi \circ (\sigma[x/d])(v) = (\varphi \circ \sigma)[x/\varphi(d)](v)$. L'unico caso in cui ciò non segue immediatamente dalla definizione 10.8 è quando v è x. In questo caso, applicando le definizioni, si ottiene che entrambi i membri sono uguali a $\varphi(d)$. □

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 già stati considerati nella dimostrazione del lemma 10.10, in cui l'ipotesi su φ era più debole.

Per quanto riguarda i quantificatori cominciamo a considerare il caso in cui F è $\exists x\, G$. Supponiamo che $I, \sigma \models F$. Esiste allora $d_0 \in D^I$ tale che $I, \sigma[x/d_0] \models G$. Per ipotesi induttiva si ha $J, \varphi \circ (\sigma[x/d_0]) \models G$. Per il lemma 10.12 $\varphi \circ (\sigma[x/d_0])$ è $(\varphi \circ \sigma)[x/\varphi(d_0)]$, e quindi $J, (\varphi \circ \sigma)[x/\varphi(d_0)] \models G$. Dunque $J, \varphi \circ \sigma \models F$. Notate che in questa direzione la suriettività di φ non è stata utilizzata.

Viceversa supponiamo che $J, \varphi \circ \sigma \models F$. Esiste allora $d_0' \in D^J$ tale che $J, (\varphi \circ \sigma)[x/d_0'] \models G$. Poiché φ è suriettiva, esiste $d_0 \in D^I$ tale che $\varphi(d_0) = d_0'$. Per il lemma 10.12 $(\varphi \circ \sigma)[x/d_0']$ è $\varphi \circ (\sigma[x/d_0])$, e quindi $J, \varphi \circ (\sigma[x/d_0]) \models G$. Dall'ipotesi induttiva segue che $I, \sigma[x/d_0] \models G$, e quindi che $I, \sigma \models F$.

Supponiamo ora che F sia $\forall x\, G$. Se $I, \sigma \models F$ allora per ogni $d \in D^I$ si ha $I, \sigma[x/d] \models G$ e, per ipotesi induttiva, $J, (\varphi \circ \sigma)[x/\varphi(d)] \models G$ per ogni $d \in D^I$. Dato che la suriettività di φ significa che ogni $d' \in D^J$ è della forma $\varphi(d)$ per qualche $d \in D^I$, abbiamo $J, (\varphi \circ \sigma)[x/d'] \models G$ per ogni $d' \in D^J$. Ma allora $J, \varphi \circ \sigma \models F$.

Viceversa supponiamo che $J, \varphi \circ \sigma \models F$, cioè $J, (\varphi \circ \sigma)[x/d'] \models G$ per ogni $d' \in D^J$. In particolare $J, (\varphi \circ \sigma)[x/\varphi(d)] \models G$ per ogni $d \in D^I$ (qui la suriettività di φ non è necessaria) e, per ipotesi induttiva, abbiamo che $I, \sigma[x/d] \models G$ per ogni $d \in D^I$. Allora $I, \sigma \models F$. □

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. □

Esempio 10.15. Sia $\mathcal{L} = \{p\}$ con p simbolo di relazione unario, e sia I_0 l'interpretazione per $\mathcal{L}$ definita da $D^{I_0} = \{0, 1\}$, $p^{I_0} = \{0\}$.

Se I è una qualsiasi interpretazione per $\mathcal{L}$ possiamo definire $\varphi : D^I \to D^{I_0}$ ponendo

$$\varphi(d) = \begin{cases} 0 & \text{se } d \in p^I, \\ 1 & \text{se } d \notin p^I. \end{cases}$$

Si verifica facilmente che φ è un omomorfismo forte. Se $p^I \neq \emptyset$ e $p^I \neq D^I$, allora φ è suriettivo e possiamo concludere che $I \equiv_{\mathcal{L}} I_0$.

Quindi le interpretazioni I_1 e I_2 per $\mathcal{L}$ definite da $D^{I_1} = D^{I_2} = \{A, B, C\}$, $p^{I_1} = \{A, B\}$, $p^{I_2} = \{A\}$ sono entrambe elementarmente equivalenti a I_0. Dato che $\equiv_{\mathcal{L}}$ è una relazione d'equivalenza (e in particolare è transitiva) possiamo concludere che $I_1 \equiv_{\mathcal{L}} I_2$.

Dimostriamo ora che non esiste un omomorfismo forte suriettivo di I_1 in I_2. Sia φ un omomorfismo forte di I_1 in I_2: dato che $A, B \in p^{I_1}$ deve essere $\varphi(A), \varphi(B) \in p^{I_2}$ e quindi $\varphi(A) = \varphi(B) = A$. Dato che $\varphi(C)$ è uno solo tra B e C, φ non può essere suriettivo.

In maniera analoga si dimostra che non esiste un omomorfismo forte suriettivo di I_2 in I_1: partiamo dal fatto che $B, C \notin p^{I_2}$ per arrivare a $\varphi(B) = \varphi(C) = C$.

Nota 10.16. L'esempio precedente mostra che è possibile che $I \equiv_{\mathcal{L}} J$ anche quando non esiste alcun omomorfismo forte suriettivo di I in J o di J in I. Quindi il corollario 10.14 esprime una condizione **sufficiente** ma **non necessaria** affinché due interpretazioni siano elementarmente equivalenti. Perciò per provare che $I \not\equiv_{\mathcal{L}} J$ non basta mostrare che non ci sono omomorfismi forti suriettivi tra le due interpretazioni: è necessario trovare un enunciato F tale che $I \models F$ e $J \not\models F$ (o viceversa), come è stato fatto nell'esempio 10.11.

Esercizio 10.17. Siano I, J e K le seguenti interpretazioni per il linguaggio $\mathcal{L} = \{p, r\}$, dove p è un simbolo di relazione unario e r un simbolo di relazione binario:

$$
\begin{array}{lll}
D^I = \{A, B\} & p^I = \{A\} & r^I = \{(A, B), (B, A)\}; \\
D^J = \mathbb{N} & p^J = \{\, n : n \text{ è dispari} \,\} & r^J = \{\, (n, m) : |n - m| \text{ è dispari} \,\}; \\
D^K = \mathbb{Z} & p^K = \{\, n : n \text{ è pari} \,\} & r^K = \{\, (n, m) : |n - m| = 1 \,\}.
\end{array}
$$

(i) Dimostrate che $I \equiv_{\mathcal{L}} J$;
(ii) dimostrate che $J \not\equiv_{\mathcal{L}} K$;
(iii) è vero che $I \equiv_{\mathcal{L}} K$?

Esercizio 10.18. In ognuno dei seguenti casi, dite se le interpretazioni I e J sono elementarmente equivalenti, isomorfe, o se esiste un omomorfismo suriettivo di I in J o di J in I (nei primi cinque esempi le interpretazioni sono definite indicando prima il dominio e poi le interpretazioni dei simboli di relazione e di funzione, così che (a), (b) e (c) si riferiscono ad un linguaggio con un simbolo di relazione binario, (d) ad un linguaggio con un simbolo di relazione unario, (e) ad un linguaggio con un simbolo di funzione unario ed un simbolo di relazione unario, ed (f) ad un linguaggio con un simbolo di funzione unario ed un simbolo di relazione binario).

(a) $I = (\mathbb{N}, \leq)$, $J = (\mathbb{Z}, \leq)$;
(b) $I = (\mathbb{Q}, <)$, $J = (\mathbb{Z}, <)$;
(c) $I = (\mathbb{Z}, \leq)$, $J = (\mathbb{Z}, \geq)$;
(d) $I = (\mathbb{R}, \{\, x : x > 0 \,\})$, $J = (\mathbb{Z}, \{\, z : z < 0 \,\})$;
(e) $I = (\mathbb{N}, \{\, n : n \text{ è pari} \,\}, S)$ [dove $S(n) = n + 1$], $J = (\mathbb{Z}, \{\, z : z < 0 \,\}, -)$;
(f) $D^I = \{A, B\}$, $f^I = id$, $r^I = \{(A, B)\}$, $D^J = \{A, B\}$, $f^J(A) = B$, $f^J(B) = A$, $r^J = \{(A, B)\}$.

Esercizio 10.19. Siano $\mathcal{L} = \{f, r\}$ e $\mathcal{L}' = \{r\}$ linguaggi in cui f è un simbolo di funzione unario e r è un simbolo di relazione binario. Siano I e J le seguenti interpretazioni per il linguaggio $\mathcal{L}$:

$$D^I = D^J = \mathbb{Z}, \qquad f^I(n) = f^J(n) = n + 1,$$
$$r^I = \{\, (n, n+1) : n \in \mathbb{Z} \,\}, \qquad r^J = \{\, (n, n-1) : n \in \mathbb{Z} \,\}.$$

Siano inoltre I' e J' le rispettive restrizioni al linguaggio $\mathcal{L}'$. Stabilite se $I \equiv_{\mathcal{L}} J$ e se $I' \equiv_{\mathcal{L}'} J'$.

2. Relazioni di congruenza

In questa sezione introduciamo la nozione di relazione di congruenza su un'interpretazione e la utilizziamo per definire l'interpretazione quoziente. Il corollario 10.31 asserisce che quest'ultima risulta essere elementarmente equivalente all'interpretazione di partenza.

Definizione 10.20. Sia I un'interpretazione per $\mathcal{L}$. 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 di $\mathcal{L}$ e per ogni $d_1, d'_1, \ldots, d_n, d'_n \in D^I$ tali che $d_1 \sim d'_1, \ldots, d_n \sim d'_n$ vale $f^I(d_1, \ldots, d_n) \sim f^I(d'_1, \ldots, d'_n)$;
- per ogni simbolo di relazione n-ario p di $\mathcal{L}$ e per ogni $d_1, d'_1, \ldots, d_n, d'_n \in D^I$ tali che $d_1 \sim d'_1, \ldots, d_n \sim d'_n$ vale
$$(d_1, \ldots, d_n) \in p^I \text{ se e solo se } (d'_1, \ldots, d'_n) \in p^I.$$

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

ESEMPIO 10.21. L'esempio più semplice di relazione di congruenza è dato dall'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.

ESEMPIO 10.22. Sia $\mathcal{L}$ il linguaggio con un simbolo di costante 0, due simboli di funzione binari $+$ e $\cdot$ ed un simbolo di relazione binario r. Sia I l'interpretazione per $\mathcal{L}$ ottenuta ponendo $D^I = \mathbb{Z}$, definendo 0^I, $+^I$ e $\cdot^I$ in modo naturale, ed infine definendo $r^I = \{\, (n, m) \,:\, n - m - 1 \text{ è un multiplo di } 3 \,\}$. In altre parole,
$$(n, m) \in r^I \text{ se e solo se } n \equiv m + 1 \pmod 3.$$
Definiamo una relazione $\sim$ su $\mathbb{Z}$ ponendo $n \sim m$ se e solo se $n - m$ è un multiplo di 3, cioè
$$n \sim m \text{ se e solo se } n \equiv m \pmod 3.$$
Dal corso di Matematica Discreta sappiamo che $\sim$ è una relazione di equivalenza. Per mostrare che $\sim$ è una relazione di congruenza su I basta verificare:
$$\begin{aligned} \text{se } n \equiv n' \text{ e } m \equiv m' &\text{ allora } n + m \equiv n' + m', \\ \text{se } n \equiv n' \text{ e } m \equiv m' &\text{ allora } n \cdot m \equiv n' \cdot m', \\ \text{se } n \equiv m + 1, n \equiv n' \text{ e } m \equiv m' &\text{ allora } n' \equiv m' + 1, \end{aligned}$$
dove tutte le congruenze sono modulo 3. Questi fatti discendono da quanto visto a Matematica Discreta a proposito delle classi di congruenza modulo 3.

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, $c^{I/\sim} = [c^I]$;
- per ogni simbolo di funzione n-ario f e ogni $d_1, \dots, d_n \in D^I$,
$$f^{I/\sim}([d_1], \dots, [d_n]) = [f^I(d_1, \dots, d_n)];$$
- per ogni simbolo di relazione n-ario p,
$$p^{I/\sim} = \left\{\, ([d_1], \dots, [d_n]) \,:\, (d_1, \dots, d_n) \in p^I \,\right\}.$$

Il dominio di $I/\!\sim$ è quindi l'insieme delle classi d'equivalenza di $\sim$, ed è essenziale per definire l'interpretazione quoziente che $\sim$ sia una relazione d'equivalenza. Le altre condizioni nella definizione di congruenza assicurano che le definizioni di $f^{I/\sim}$ e $p^{I/\sim}$ siano "ben date", cioè non dipendano dai rappresentanti delle classi di equivalenza. Infatti se $[d_1] = [d'_1]$, $\dots$, $[d_n] = [d'_n]$ è necessario che $[f^I(d_1, \dots, d_n)] = [f^I(d'_1, \dots, d'_n)]$ e che $(d_1, \dots, d_n) \in p^I$ se e solo se $(d'_1, \dots, d'_n) \in p^I$.

ESEMPIO 10.24. Riprendiamo l'esempio 10.21. In questo caso ogni classe d'equivalenza ha un solo elemento e la funzione che manda $d \in D^I$ in $[d] \in D^{I/\sim}$ è una biiezione. Si verifica facilmente che è un isomorfismo e perciò $D^{I/\sim}$ è isomorfo a I. Questo esempio non è quindi molto interessante, perché non abbiamo ottenuto un'interpretazione veramente diversa da quella di partenza.

ESEMPIO 10.25. Riprendiamo l'esempio 10.22. In questo caso $D^{I/\sim}$ ha tre elementi, che possiamo indicare con [0], [1] e [2]. Inoltre si ha
$$0^{I/\sim} = [0], \quad +^{I/\sim}([n], [m]) = [n + m], \quad \cdot^{I/\sim}([n], [m]) = [n \cdot m],$$
$$r^{I/\sim} = \{([1], [0]), ([2], [1]), ([0], [2])\}.$$

Ad esempio $+^{I/\sim}([2],[2]) = [1]$.

ESEMPIO 10.26. Sia $\mathcal{L} = \{f, r\}$ un linguaggio in cui f è un simbolo di funzione unario e r è un simbolo di relazione binario. Sia I l'interpretazione per $\mathcal{L}$ definita da

$$D^I = \{0,1,2,3,4,5,6\}; \qquad r^I = \{(1,3),(2,0),(2,2),(2,5),(3,2),(4,3),(6,3)\}$$
$$f^I(0) = 3; f^I(1) = 4; f^I(2) = 1; f^I(3) = 3; f^I(4) = 6; f^I(5) = 3; f^I(6) = 4.$$

Vogliamo definire una relazione di congruenza $\sim$ su I che abbia quattro classi d'equivalenza. Dato che 2 è l'unico elemento di D^I che compare insieme a 0 negli elementi di r^I, non può essere in relazione di congruenza con nessun altro elemento di D^I. Analogamente 3 è l'unico elemento collegato a 1 da r^I e non è congruente a nessun altro elemento. Notiamo anche che 1, 4 e 6 sono in relazione r^I con 3, mentre 0 e 5 non lo sono. Perciò le quattro classi d'equivalenza rispetto a $\sim$ non possono che essere $\{0,5\}$, $\{1,4,6\}$, $\{2\}$ e $\{3\}$. Inoltre $\sim$ verifica anche la condizione che riguarda f, perché $f^I(0) \sim f^I(5)$ e $f^I(1) \sim f^I(4) \sim f^I(6)$.

Si ha inoltre che $D^{I/\sim} = \{[0],[1],[2],[3]\}$, $f^{I/\sim}([0]) = [3]$, $f^{I/\sim}([1]) = [1]$, $f^{I/\sim}([2]) = [1]$, $f^{I/\sim}([3]) = [3]$, $r^{I/\sim} = \{([1],[3]),([2],[0]),([2],[2]),([3],[2])\}$.

ESERCIZIO 10.27. Sia $\mathcal{L} = \{a, f, r\}$ un linguaggio dove a è un simbolo di costante, f è un simbolo di funzione unario ed r è un simbolo di relazione binario. Sia I l'interpretazione per $\mathcal{L}$ definita da:

$$D^I = \mathbb{Z}, \qquad a^I = 19, \qquad f^I(z) = z - 4,$$
$$r^I = \{\, (z, z') \,:\, \text{la divisione } (z - z') \div 4 \text{ ha resto } 2 \,\}.$$

Sia inoltre $\sim$ la relazione su $\mathbb{Z}$ definita da

$$z \sim z' \iff 4 \text{ divide } z - z'.$$

Dimostrate che $\sim$ è una relazione di congruenza su I e descrivete l'interpretazione quoziente $I/\!\sim$.

ESERCIZIO 10.28. Sia I l'interpretazione dell'esempio 7.68. Dimostrate che la relazione $=^I$ è una relazione di congruenza su I e descrivete $I/\!=^I$.

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 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 $\pi(c^I) = [c^I] = c^{I/\sim}$;
- per ogni simbolo di funzione n-ario f di $\mathcal{L}$ ed ogni $d_1, \dots, d_n \in D^I$ si ha
$$\begin{aligned}\pi(f^I(d_1,\dots,d_n)) &= [f^I(d_1,\dots,d_n)]\\ &= f^{I/\sim}([d_1],\dots,[d_n])\\ &= f^{I/\sim}(\pi(d_1),\dots,\pi(d_n));\end{aligned}$$
- per ogni simbolo di relazione n-ario p di $\mathcal{L}$ ed ogni $d_1, \dots, d_n \in D^I$ si ha $(d_1,\dots,d_n) \in p^I$ se e solo se $([d_1],\dots,[d_n]) \in p^{I/\sim}$ se e solo se $(\pi(d_1),\dots,\pi(d_n)) \in p^{I/\sim}$. □

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

In questa sezione applicheremo quanto ottenuto nelle sezioni precedenti alla logica con uguaglianza. In particolare caratterizzeremo, nel teorema 10.33, le interpretazioni per un linguaggio con uguaglianza $\mathcal{L}$ che soddisfano gli assiomi dell'uguaglianza $\mathsf{Eq}_{\mathcal{L}}$ e troveremo dei collegamenti tra la semantica per la logica con uguaglianza e la semantica per la logica pura.

Lemma 10.32. *Se I è un'interpretazione per un linguaggio con uguaglianza $\mathcal{L}$ tale che $I \vDash \mathsf{Eq}_{\mathcal{L}}$, allora $=^I$ è una relazione di congruenza su I, e $I/\!=^I$ è un'interpretazione normale.*

Dimostrazione. Il fatto che I soddisfa gli enunciati di $\mathsf{Eq}_{\mathcal{L}}$ (definizione 6.64) assicura che le condizioni della definizione di relazione di congruenza sono soddisfatte da $=^I$. Ad esempio, $I \vDash$ (e2) significa che $=^I$ è simmetrica (una delle condizioni necessarie per essere relazione d'equivalenza), e $I \vDash$ (e5) significa che $=^I$ soddisfa la terza condizione della definizione 10.20.

Per verificare che l'interpretazione di $=$ in $I/\!=^I$ è l'identità osservate che $([d],[d']) \in =^{I/=^I}$ se e solo se $(d,d') \in =^I$ se e solo se $[d]=[d']$. □

Teorema 10.33. *Sia I è un'interpretazione per un linguaggio con uguaglianza $\mathcal{L}$. Allora $I \vDash \mathsf{Eq}_{\mathcal{L}}$ se e solo se I è elementarmente equivalente ad un'interpretazione normale.*

Dimostrazione. Se $I \vDash \mathsf{Eq}_{\mathcal{L}}$ allora per il corollario 10.31 si ha $I \equiv_{\mathcal{L}} I/\!=^I$. Per il lemma 10.32 $I/\!=^I$ è un'interpretazione normale.

Viceversa se I è elementarmente equivalente ad un'interpretazione normale J allora $J \vDash \mathsf{Eq}_{\mathcal{L}}$ per il lemma 7.67, e quindi $I \vDash \mathsf{Eq}_{\mathcal{L}}$. □

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*

$$T \vDash_{=} F \quad \textit{se e solo se} \quad T, \mathsf{Eq}_{\mathcal{L}} \vDash F.$$

Dimostrazione. Supponiamo che $T \vDash_{=} F$. Per mostrare che $T, \mathsf{Eq}_{\mathcal{L}} \vDash F$ fissiamo un'interpretazione (non necessariamente normale) I per $\mathcal{L}$ e uno stato σ di I tale che $I, \sigma \vDash T, \mathsf{Eq}_{\mathcal{L}}$. Dato che $I \vDash \mathsf{Eq}_{\mathcal{L}}$, per il lemma 10.32 $=^I$ è una relazione di congruenza su I e $I/\!=^I$ è un'interpretazione normale. Il corollario 10.31 implica che $I/\!=^I, \pi \circ \sigma \vDash T$, da cui per l'ipotesi si ottiene che $I/\!=^I, \pi \circ \sigma \vDash F$. Usando nuovamente il corollario 10.31 si ha che $I, \sigma \vDash F$.

Ora supponiamo che $T, \mathsf{Eq}_{\mathcal{L}} \vDash F$. Per dimostrare che $T \vDash_{=} F$ sia I un'interpretazione normale per $\mathcal{L}$ e σ uno stato di I tale che $I, \sigma \vDash T$: per il lemma 7.67 $I \vDash \mathsf{Eq}_{\mathcal{L}}$ e allora $I, \sigma \vDash T, \mathsf{Eq}_{\mathcal{L}}$. Per l'ipotesi $I, \sigma \vDash F$. □

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 $\nvDash_{=} \neg F$ se e solo se (per il teorema 10.34) $\mathsf{Eq}_{\mathcal{L}} \nvDash \neg F$ se e solo se $\mathsf{Eq}_{\mathcal{L}}, F$ è soddisfacibile. □

Esercizio 10.36. Dimostrate il teorema 10.35 usando le interpretazioni, in maniera analoga a quanto fatto nella dimostrazione del teorema 10.34.

CAPITOLO 11

Il metodo dei tableaux: caso predicativo

In questo capitolo il metodo dei tableaux studiato nel capitolo 4 verrà esteso al caso della logica predicativa. L'idea che guida il metodo dei tableaux è la stessa: cerchiamo sistematicamente un'interpretazione che soddisfi l'enunciato di partenza. La maggior complessità della nozione di interpretazione nel caso predicativo rende necessari diversi cambiamenti all'algoritmo. Introdurremo nuove regole per la costruzione di nodi nel caso in cui si agisca su formule con quantificatori e il metodo avrà proprietà piuttosto diverse da quelle dimostrate nel caso della logica proposizionale: l'algoritmo predicativo non ha la proprietà di terminazione ed è solo una procedura di semidecisione per la validità (si veda la nota 11.47).

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

La seguente definizione è la ovvia generalizzazione al caso predicativo della definizione 3.2. Ricordiamo che nella definizione 6.43 abbiamo definito i letterali nella logica predicativa.

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 allora è insoddisfacibile.*

La nostra descrizione del metodo dei tableaux nel caso proposizionale si basava sulla distinzione tra doppie negazioni, α-formule e β-formule. Nel caso predicativo la definizione di questi tipi di formule è la stessa che nel caso proposizionale (definizioni 3.12 e 3.13), ma per poter classificare tutte le formule predicative è necessario introdurre nuovi tipi di formule.

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.

γ-formula	istanza
$\forall x\, F$	$F\{x/a\}$
$\neg\exists x\, F$	$\neg F\{x/a\}$

δ-formula	istanza
$\exists x\, F$	$F\{x/a\}$
$\neg\forall x\, F$	$\neg F\{x/a\}$

Diciamo che $F\{x/a\}$ o $\neg F\{x/a\}$ è l'*istanza della γ- o δ-formula relativa ad a.* Notiamo che la sostituzione effettuata per ottenere un'istanza è sempre ammissibile per la nota 6.53.

Nota 11.5. Notiamo che se G è una γ- o δ-formula che è un enunciato allora tutte le sue istanze sono enunciati. Similmente, i ridotti di doppie negazioni, α-formule e β-formule che sono enunciati sono a loro volta enunciati.

I prossimi risultati stabiliscono dei legami tra γ e δ-formule e loro istanze. Essi corrispondono al lemma 3.14 nel caso proposizionale, ma in questo caso non otteniamo equivalenze logiche, ma risultati più deboli.

Lemma 11.6. *Se G è una γ-formula e G_1 una sua istanza allora $G \vDash G_1$.*

Dimostrazione. Per le formule del tipo $\forall x\, F$ questo è un caso particolare del lemma 7.63. Per le formule del tipo $\neg\exists x\, F$ basta usare la seconda equivalenza logica del lemma 8.5 per ricondurci al caso precedente. □

Esercizio 11.7. Dimostrate che se G è una δ-formula e G_1 una sua istanza allora $G_1 \vDash G$.

Il risultato dell'esercizio precedente non è però utile nella costruzione dei tableau, perché permette di passare da una formula più semplice (l'istanza) ad una più complessa (la δ-formula), mentre a noi interessa fare il contrario.

Il prossimo lemma corrisponde al lemma 3.15 nel caso proposizionale.

Lemma 11.8. *Una formula è di uno e uno solo dei tipi seguenti:*

- *un letterale,*
- *una doppia negazione,*
- *una α-formula,*
- *una β-formula,*
- *una γ-formula,*
- *una δ-formula.*

Dimostrazione. La dimostrazione è analoga a quella del lemma 3.15, basandosi questa volta sul lemma 6.23. La lasciamo come esercizio per il lettore. □

2. Esempi preliminari

Come già fatto nel caso proposizionale, prima di descrivere l'algoritmo dei tableaux esaminiamo in dettaglio alcuni esempi in cui partiamo dalle regole per i tableaux proposizionali già introdotte e le estendiamo alle γ- e δ-formule, evidenziando alcune importanti novità che è necessario introdurre.

Esempio 11.9. Consideriamo l'enunciato $\exists x(p(x) \vee q(x)) \to \exists x\, p(x) \vee \exists x\, q(x)$, che indichiamo con F: è valido per il lemma 8.14 (e l'esercizio 7.49). Per dimostrare la validità di F con il metodo dei tableaux partiamo da $\neg F$, che è una α-formula, e al passo successivo abbiamo un'altra α-formula su cui agire. Il tableau comincia quindi in questo modo:

$$\begin{array}{c} \neg F \\ | \\ \exists x(p(x) \vee q(x)), \neg(\exists x\, p(x) \vee \exists x\, q(x)) \\ | \\ \exists x(p(x) \vee q(x)), \neg\exists x\, p(x), \neg\exists x\, q(x) \end{array}$$

A questo punto non possiamo più basarci su ciò che facevamo nel caso proposizionale perché abbiamo due γ-formule e una δ-formula. La δ-formula $\exists x(p(x) \vee q(x))$ asserisce l'esistenza di un elemento del dominio con certe caratteristiche. Introduciamo una costante che rappresenti questo elemento. Se a è la costante sostituiamo la δ-formula con $p(a) \vee q(a)$, che è l'istanza relativa ad a di $\exists x(p(x) \vee q(x))$. Otteniamo una β-formula e sappiamo come fare il passo successivo:

$$\exists x(p(x) \vee q(x)), \neg\exists x\, p(x), \neg\exists x\, q(x)$$
$$|$$
$$p(a) \vee q(a), \neg\exists x\, p(x), \neg\exists x\, q(x)$$
$$\swarrow \qquad \searrow$$
$$p(a), \neg\exists x\, p(x), \neg\exists x\, q(x) \qquad q(a), \neg\exists x\, p(x), \neg\exists x\, q(x)$$

Siamo nuovamente giunti ad un punto in cui non possiamo applicare le regole proposizionali. La γ-formula $\neg\exists x\, p(x)$ è logicamente equivalente a $\forall x\, \neg p(x)$ (lemma 8.5) e quindi asserisce che per ogni elemento del dominio, ed in particolare per quello rappresentato da a, non vale p. Dunque possiamo sostituire nel ramo di sinistra $\neg\exists x\, p(x)$ con la sua istanza $\neg p(a)$ e ottenere un nodo la cui etichetta

$$p(a), \neg p(a), \neg\exists x\, q(x)$$

contiene una coppia complementare di letterali.

Similmente nel ramo di destra possiamo agire sulla γ-formula $\neg\exists x\, q(x)$ ed ottenere un nodo con etichetta

$$q(a), \neg\exists x\, p(x), \neg q(a)$$

Tutte le foglie del tableau contengono dunque una coppia complementare di letterali e quindi il tableau è chiuso, come volevamo.

Esercizio 11.10. Usate le idee dell'esempio 11.9 per mostrare che l'enunciato $\forall x(p(x) \rightarrow q(x)) \rightarrow (\forall x\, p(x) \rightarrow \forall x\, q(x))$ è valido.

Esempio 11.11. Consideriamo l'enunciato $\exists x\, p(x) \wedge \exists x\, q(x) \rightarrow \exists x(p(x) \wedge q(x))$, che indichiamo con G. G è soddisfacibile ma non valido (vedere esercizio 8.17), e quindi ci aspettiamo che il tableau per $\neg G$ sia aperto. Ecco un primo tentativo di tableau per $\neg G$, in cui abbiamo utilizzato le idee dell'esempio 11.9:

$$\neg G$$
$$|$$
$$\exists x\, p(x) \wedge \exists x\, q(x), \neg\exists x(p(x) \wedge q(x))$$
$$|$$
$$\exists x\, p(x), \exists x\, q(x), \neg\exists x(p(x) \wedge q(x))$$
$$|$$
$$p(a), \exists x\, q(x), \neg\exists x(p(x) \wedge q(x))$$
$$|$$
$$p(a), q(a), \neg\exists x(p(x) \wedge q(x))$$
$$|$$
$$p(a), q(a), \neg(p(a) \wedge q(a))$$
$$\swarrow \qquad \searrow$$
$$p(a), q(a), \neg p(a) \qquad p(a), q(a), \neg q(a)$$
$$\otimes \qquad\qquad\qquad \otimes$$

Questo tableau è chiuso e quindi qualcosa non ha funzionato. Il problema è nell'aver utilizzato la stessa costante per istanziare sia $\exists x\, p(x)$ che $\exists x\, q(x)$: non c'è nessuna ragione per cui l'elemento che soddisfa $p(x)$ coincida con quello che soddisfa $q(x)$! La soluzione è imporre che una δ-formula esistenziale sia sempre istanziata relativamente ad una costante **nuova**, cioè che non compare nel tableau sviluppato fino a quel momento. Le ultime righe del nostro tentativo di tableau diventano quindi:

$$p(a), \exists x\, q(x), \neg\exists x(p(x) \land q(x))$$
$$|$$
$$p(a), q(b), \neg\exists x(p(x) \land q(x))$$
$$|$$
$$p(a), q(b), \neg(p(a) \land q(a))$$
$$\swarrow \qquad \searrow$$
$$\begin{array}{cc} p(a), q(b), \neg p(a) & p(a), q(b), \neg q(a) \\ \otimes & \bigcirc \end{array}$$

C'è però ancora qualcosa di insoddisfacente in questo tentativo di tableau. La γ-formula $\neg\exists x(p(x) \land q(x))$ (che indichiamo con H) asserisce che $p(x) \land q(x)$ non vale per nessun elemento del dominio. Questa informazione è però stata utilizzata solo per a, mentre sarebbe utile poterla sfruttare anche per b. La soluzione è che le γ-formule non vanno mai cancellate dalle etichette dei nodi, per poter essere eventualmente usate più volte. Otteniamo così la seguente parte finale del tableau:

$$p(a), q(b), H$$
$$|$$
$$p(a), q(b), H, \neg(p(a) \land q(a))$$
$$\swarrow \qquad \searrow$$
$$\begin{array}{cc} p(a), q(b), H, \neg p(a) & p(a), q(b), H, \neg q(a) \\ \otimes & | \\ & p(a), q(b), H, \neg q(a), \neg(p(b) \land q(b)) \end{array}$$
$$\swarrow \qquad \searrow$$
$$\begin{array}{cc} p(a), q(b), H, \neg q(a), \neg p(b) & p(a), q(b), H, \neg q(a), \neg q(b) \\ \bigcirc & \otimes \end{array}$$

Il nodo aperto ci suggerisce un'interpretazione che soddisfa $\neg G$: $D^I = \{a, b\}$, $p^I = \{a\}$, $q^I = \{b\}$ (notiamo che non è necessario interpretare i simboli di costante a e b, che non facevano parte del linguaggio di G).

Esempio 11.12. Consideriamo ora l'enunciato F definito da $F_1 \land F_2 \land F_3$ dove F_1, F_2 e F_3 sono rispettivamente

$$\forall x\, \exists y\, r(x, y);$$
$$\forall x\, \forall y\, \forall z(r(x, y) \land r(y, z) \to r(x, z));$$
$$\forall x\, \neg r(x, x).$$

Supponiamo di voler mostrare che F è soddisfacibile, e a questo scopo costruiamo un tableau per F.

Dopo alcuni passaggi che riguardano solo α-formule otteniamo un nodo etichettato da F_1, F_2, F_3, che sono tutte γ-formule. Non abbiamo nessuna costante su cui istanziare le γ-formule, e quindi introduciamo una nuova costante, come fatto nel caso delle δ-formule. Se a_1 è questa costante e consideriamo l'istanza di F_1 relativa ad a_1 otteniamo un nodo etichettato da

$$F_1, F_2, F_3, \exists y\, r(a_1, y)$$

(notare che non abbiamo cancellato la γ-formula utilizzata).

Se ora agiamo sulla δ-formula $\exists y\, r(a_1, y)$ (che possiamo cancellare) introducendo la nuova costante a_2 il nodo successivo sarà etichettato da

$$F_1, F_2, F_3, r(a_1, a_2).$$

Ora F_1 può essere istanziata su a_2 e otteniamo un nodo etichettato da

$$F_1, F_2, F_3, r(a_1, a_2), \exists y\, r(a_2, y)$$

che conduce al nodo etichettato da

$$F_1, F_2, F_3, r(a_1, a_2), r(a_2, a_3).$$

Proseguendo in questo tableau è chiaro che la costruzione non terminerà mai, ma otterremo un ramo infinito le cui etichette conterranno i letterali del tipo $r(a_i, a_{i+1})$. Ciò suggerisce un'interpretazione che soddisfa F ed ha dominio infinito: $D^I = \mathbb{N}$, $r^I = \{ (i, j) : i < j \}$.

In effetti F non ha interpretazioni con dominio finito, come dimostrato nell'esercizio 9.16, e quindi il ramo infinito di questo tableau è inevitabile.

Nota 11.13. L'esempio 11.12 mostra che il metodo dei tableaux predicativi non gode della proprietà della terminazione forte e quindi non è una procedura di decisione per la soddisfacibilità (o la validità) degli enunciati predicativi: è possibile che la sua esecuzione si prolunghi all'infinito.

Esempio 11.14. Sia F l'enunciato dell'esempio 11.12 e indichiamo con F' l'enunciato $F \wedge p(a_1) \wedge \neg p(a_1)$ (in questo caso a_1 è un simbolo di costante del linguaggio). È evidente che F' è insoddisfacibile. È altrettanto evidente che potremmo ripetere la costruzione del tableau dell'esempio 11.12 aggiungendo semplicemente in ogni etichetta l'enunciato $p(a_1) \wedge \neg p(a_1)$, senza mai agire su di esso. Otterremo dunque un tableau con un ramo infinito, che suggerisce soddisfacibilità dell'enunciato di partenza.

Nota 11.15. Il comportamento dell'esempio 11.14 è dovuto al fatto che le γ-formule non vengono cancellate dalle etichette e quindi si può continuare ad agire su di esse e "dimenticarsi" di qualche altra formula (nel nostro esempio la α-formula $p(a) \wedge \neg p(a)$) mancando così delle opportunità di chiudere il tableau. Notiamo che questo non avveniva nel caso proposizionale, perché ogni formula utilizzata veniva cancellata dall'etichetta e si aveva la proprietà della terminazione forte.

La soluzione a questo problema verrà ottenuta procedendo ad una costruzione *sistematica* del tableau, in cui non ci si potrà "dimenticare" di qualche formula.

Nella prossima sezione descriviamo l'algoritmo dei tableaux predicativi **senza** tener conto del problema evidenziato dall'esempio 11.14. Il tableau quindi non verrà costruito in maniera sistematica, ma questo è sufficiente a dimostrare la correttezza del nostro metodo nella sezione 4. Nella sezione 5 descriveremo come costruire sistematicamente i tableaux, e questo ci porterà ad un risultato di completezza di cui accenneremo la dimostrazione nella sezione 6.

3. L'algoritmo

Diverse parti della descrizione dell'algoritmo per i tableaux predicativi sono identiche a quelle corrispondenti nell'algoritmo nel caso proposizionale (algoritmo 4.5): bisogna dunque prestare attenzione soprattutto alle parti nuove, cioè quelle che riguardano γ- e δ-formule. Un'altra differenza è che il fatto che le γ-formule non vengano mai eliminate dalle etichette ha come conseguenza che non possiamo aspettarci di arrivare a foglie la cui etichetta è un insieme di letterali: una foglia verrà chiusa quando la sua etichetta contiene una coppia complementare di letterali, anche se contiene anche formule che non sono letterali (in realtà facevamo questo anche a livello proposizionale, in base alla convenzione 4.32).

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 $\mathcal{T}_0, \dots, \mathcal{T}_i, \dots$: 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.

All'inizio della costruzione $\mathcal{T}_0$ consiste di un solo nodo (la *radice* dell'albero) etichettato con $\{F\}$. Allo stadio i cerchiamo una foglia n dell'albero $\mathcal{T}_i$ tale che $E(n)$ non contiene una coppia complementare di letterali e contiene qualche enunciato che non è un letterale. Se una tale foglia non esiste la costruzione del tableau è terminata e l'algoritmo si arresta. Altrimenti fissiamo n e scegliamo un enunciato $G \in E(n)$ che non è un letterale. Per il lemma 11.8 ci sono cinque possibilità:

(1) se G è una doppia negazione con ridotto G_1 aggiungiamo un nodo n' sotto n e poniamo $E(n') = (E(n) \setminus \{G\}) \cup \{G_1\}$;
(2) se G è una α-formula con ridotti G_1 e G_2 aggiungiamo un nodo n' sotto n e poniamo $E(n') = (E(n) \setminus \{G\}) \cup \{G_1, G_2\}$;
(3) se G è una β-formula con ridotti G_1 e G_2 aggiungiamo due nodi tra loro inconfrontabili n_1 e n_2 sotto n e poniamo $E(n_1) = (E(n) \setminus \{G\}) \cup \{G_1\}$ e $E(n_2) = (E(n) \setminus \{G\}) \cup \{G_2\}$;
(4) se G è una γ-formula scegliamo un'istanza G_1 di G, aggiungiamo un nodo n' sotto n e poniamo $E(n') = E(n) \cup \{G_1\}$;
(5) se G è una δ-formula fissiamo una costante a che **non** compare in $E(n)$ e sia G_1 l'istanza di G relativa ad a: aggiungiamo un nodo n' sotto n e poniamo $E(n') = (E(n) \setminus \{G\}) \cup \{G_1\}$.

Notiamo che in ogni caso n non è una foglia di $\mathcal{T}_{i+1}$ e che, per la nota 11.5, le etichette dei nuovi nodi contengono solo enunciati.

L'algoritmo che abbiamo appena descritto è non deterministico perché ad ogni passo scegliamo una foglia n con certe caratteristiche e, all'interno di $E(n)$, scegliamo un enunciato che non sia un letterale. Inoltre in alcuni casi è necessario scegliere anche un'istanza della γ-formula considerata. Questo implica che qualche enunciato che non è un letterale può non essere mai scelto (come accade negli esempi 11.12 e 11.14), oppure che qualche istanza di una γ-formula non appartenga a nessun $E(n)$. Nel caso proposizionale ciò non poteva avvenire, come dimostrato dal teorema 4.11.

Come nel caso proposizionale, gli alberi $\mathcal{T}_0, \dots, \mathcal{T}_i, \dots$ non sono rappresentati separatamente: si deve piuttosto pensare che il tableau "cresce" verso la sua forma finale, che usualmente è l'unica che vediamo.

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 $E(n)$.

NOTA 11.20. È importante notare che l'algoritmo 11.16 **non** gode della proprietà della terminazione forte, come già evidenziato nell'esempio 11.12 e nella nota successiva. È quindi possibile che un tableau non abbia foglie, o solo alcuni dei suoi rami terminino con una foglia mentre altri siano infiniti.

Per semplificare la costruzione dei tableau adottiamo da subito la convenzione 4.31 (ricordiamo che la convenzione 4.32 è già stata adottata), che riportiamo qui:

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 convenzione 11.21. Notiamo che la γ-formula $\forall x\, \neg p(x)$ non viene mai cancellata.

$$\begin{gathered}
\underline{\neg(\forall x\, \neg p(x) \to \neg\exists x\, p(x))} \\
| \\
\forall x\, \neg p(x), \underline{\exists x\, p(x)} \\
| \\
\underline{\forall x\, \neg p(x)}, p(a) \\
| \\
\forall x\, \neg p(x), \neg p(a), p(a) \\
\otimes
\end{gathered}$$

Notiamo che per chiudere il tableau abbiamo scelto in modo opportuno l'istanza della γ-formula $\forall x\, \neg p(x)$ quando abbiamo agito su di essa. Quando si agisce su una γ-formula è sempre meglio, se possibile, utilizzare un'istanza relativa ad un simbolo di costante che compare nell'etichetta del nodo.

Ecco un altro tableau per lo stesso enunciato: qui abbiamo agito sulla γ-formula prima di aver eliminato la δ-formula, e questo porta alla creazione di un simbolo di costante "inutile" (in questo caso a) e al dover agire due volte sulla γ-formula (una volta per creare l'istanza relativa a a, che non ci porterà a chiudere alcuna foglia, ed una volta per creare l'istanza relativa a b).

$$\begin{gathered}
\underline{\neg(\forall x\, \neg p(x) \to \neg\exists x\, p(x))} \\
| \\
\underline{\forall x\, \neg p(x)}, \exists x\, p(x) \\
| \\
\forall x\, \neg p(x), \neg p(a), \underline{\exists x\, p(x)} \\
| \\
\underline{\forall x\, \neg p(x)}, \neg p(a), p(b) \\
| \\
\forall x\, \neg p(x), \neg p(b), \neg p(a), p(b) \\
\otimes
\end{gathered}$$

NOTA 11.23. Per semplificare il più possibile il tableau costruito dall'algoritmo 11.16 è opportuno agire sugli enunciati disponibili nel seguente ordine: doppie negazioni o α-formule, β-formule, δ-formule, γ-formule. Questa indicazione nasce innanzitutto dall'opportunità di applicare prima le regole proposizionali, che sono più semplici. La ragione per preferire doppie negazioni e α-formule alle β-formule è stata già discussa nella nota 4.37. Agendo sulle δ-formule prima che sulle γ-formule si introducono prima i nuovi simboli di costante su cui si potranno successivamente istanziare le γ-formule.

Esempio 11.24. Costruiamo un tableau per l'enunciato

$$\forall x(p(x) \to q(x)) \land \forall x(\neg q(x) \to p(x)) \land \exists x\, \neg q(x).$$

Indichiamo con F e G gli enunciati $\forall x(p(x) \to q(x))$ e $\forall x(\neg q(x) \to p(x))$. In ogni passaggio sottolineiamo l'enunciato su cui agiamo.

$$\begin{array}{ccc}
 & \underline{F \land G \land \exists x\, \neg q(x)} & \\
 & | & \\
 & \underline{F \land G}, \exists x\, \neg q(x) & \\
 & | & \\
 & F, G, \underline{\exists x\, \neg q(x)} & \\
 & | & \\
 & \underline{F}, G, \neg q(a) & \\
 & | & \\
 & F, \underline{p(a) \to q(a)}, G, \neg q(a) & \\
 & \swarrow \quad \searrow & \\
F, \neg p(a), \underline{G}, \neg q(a) & & F, q(a), G, \neg q(a) \\
| & & \otimes \\
F, \neg p(a), G, \underline{\neg q(a) \to p(a)}, \neg q(a) & & \\
\swarrow \quad \searrow & & \\
F, \neg p(a), G, q(a), \neg q(a) \qquad F, \neg p(a), G, p(a), \neg q(a) & & \\
\otimes \qquad\qquad\qquad \otimes & &
\end{array}$$

Esercizio 11.25. Costruite un tableau per

$$\exists x\, p(x) \land \forall x\, (\exists y\, r(x, y) \to \neg p(x)) \land \forall x\, r(x, a).$$

4. La correttezza dei tableaux predicativi

In questa sezione dimostreremo l'analogo per i tableaux predicativi del teorema 4.21 per i tableaux proposizionali.

Abbiamo già a disposizione, oltre a quanto utilizzato a livello proposizionale, il lemma 11.6 che collega una γ-formula con le sue istanze. Invece, come già osservato, l'esercizio 11.7 non rispecchia ciò che avviene nella costruzione dei tableaux. Abbiamo dunque bisogno di un risultato che colleghi una δ-formula alla sua istanza che utilizziamo nei tableaux. Notiamo che questo collegamento riguarda la soddisfacibilità e non la conseguenza logica.

Lemma 11.26. *Siano T un insieme di formule, G una δ-formula e G_1 un'istanza di G relativa ad un simbolo di costante che non compare in T, G. Se T, G è soddisfacibile allora T, G_1 è soddisfacibile.*

Dimostrazione. Iniziamo a considerare il caso in cui G sia del tipo $\exists x\, F$ e quindi G_1 sia $F\{x/a\}$. Sia $\mathcal{L} = \mathcal{L}(T, G)$ il linguaggio di T, G: per ipotesi a non vi appartiene. Sia $\mathcal{L}' = \mathcal{L} \cup \{a\}$.

Dato che T, G è soddisfacibile possiamo fissare un'interpretazione I per $\mathcal{L}$ e uno stato σ di I tali che $I, \sigma \models T, G$. Dato che $I, \sigma \models \exists x\, F$ esiste $d_0 \in D^I$ tale che $I, \sigma[x/d_0] \models F$. Definiamo un'interpretazione I' per $\mathcal{L}'$ che coincide con I su $\mathcal{L}$

(quindi $D^{I'} = D^I$, e i simboli di costante e relazione diversi da a sono interpretati in I' come in I) e interpreta a come d_0 (cioè $a^{I'} = d_0$). Notiamo che, dato che $D^{I'} = D^I$, σ è uno stato anche per I'.

Dato che a non compare in T, per il lemma 7.12 si ha $I', \sigma \vDash T$. Inoltre $I', \sigma[x/a^{I'}] \vDash F$ per la nostra definizione di $a^{I'}$ e perché a non compare in G e quindi neppure nella sua sottoformula F. Per il Lemma di Sostituzione 7.59 (ricordando che $\sigma(a) = a^{I'}$) abbiamo $I', \sigma \vDash F\{x/a\}$. Quindi I' e σ mostrano la soddisfacibilità di $T, F\{x/a\}$.

Se G è del tipo $\neg\forall x\, F$ usiamo la prima equivalenza logica del lemma 8.5 per ricondurci al caso precedente. □

Nota 11.27. L'ipotesi su a del lemma 11.26 è necessaria. Ad esempio se $T = \{\neg p(a)\}$ e G è $\exists x\, p(x)$ allora a compare in T, $\{\neg p(a), \exists x\, p(x)\}$ è soddisfacibile, ma $\{\neg p(a), p(a)\}$ è insoddisfacibile. Un esempio in cui a compare in G si ottiene considerando $T = \{\forall x\, \neg r(x,x)\}$ e prendendo $\exists x\, r(a,x)$ come G.

Abbiamo ora tutti gli strumenti per dimostrare il teorema di correttezza, dando per prima cosa le definizioni di tableau chiuso e aperto.

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 contiene coppie complementari.

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.

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

Dato che $\mathcal{T}$ è chiuso, e in particolare non contiene rami infiniti, per il lemma 4.10 $\mathcal{T}$ è finito. Possiamo quindi effettuare la dimostrazione di $(\star)$ per induzione sull'altezza del nodo n in $\mathcal{T}$ come nel caso proposizionale.

Se l'altezza di n in $\mathcal{T}$ è 0 significa che n è una foglia del tableau. Dato che $\mathcal{T}$ è chiuso $E(n)$ contiene una coppia complementare di letterali. Per il lemma 11.3 $E(n)$ è insoddisfacibile.

Se l'altezza di n in $\mathcal{T}$ è maggiore di 0, allora n ha uno o due successori in $\mathcal{T}$, che sono stati ottenuti agendo su qualche $G \in E(n)$ e ci sono cinque possibilità. Nei primi tre casi, cioè quando G è una doppia negazione, una α-formula o una β-formula, la dimostrazione del caso proposizionale può essere ripetuta tale e quale, cambiando solamente la notazione utilizzata per le interpretazioni. Restano dunque solo i casi delle γ- e δ-formule.

(4) se G è una γ-formula, n ha un solo successore n' e si ha $E(n') = E(n) \cup \{G_1\}$ dove G_1 è un'istanza di G. Come negli altri casi, usando l'ipotesi induttiva, si ottiene che $E(n')$ è insoddisfacibile. Supponiamo per assurdo che $I \vDash E(n)$ (dato che gli elementi di $E(n)$ sono enunciati non è necessario menzionare lo stato). Dato che $G \vDash G_1$ (lemma 11.6) e $I \vDash G$ (perché $G \in$

$E(n)$) si ha anche $I \vDash G_1$. Quindi $I \vDash E(n')$, contro l'insoddisfacibilità di $E(n')$.

(5) se G è una δ-formula, n ha un solo successore n' e si ha $E(n') = (E(n) \setminus \{G\}) \cup \{G_1\}$ dove G_1 è un'istanza di G relativa ad un simbolo di costante a che non compare in $E(n)$. Sfruttando l'ipotesi induttiva si ottiene che $E(n')$ è insoddisfacibile. Sia $T = E(n) \setminus \{G\}$. Il simbolo di costante a soddisfa le ipotesi del lemma 11.26 e quindi se $E(n) = T, G$ fosse soddisfacibile lo sarebbe anche $E(n') = T, G_1$. Quindi $E(n)$ è insoddisfacibile.

Abbiamo dunque dimostrato ($\star$) e quindi il teorema. $\square$

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

$$\exists x\, r(a,x) \wedge \forall x(\exists y\, r(y,x) \to \neg p(x)) \to \exists x\, \neg p(x).$$

Indichiamo con F, G e H gli enunciati $\forall x(\exists y\, r(y,x) \to \neg p(x))$, $\neg \exists x\, \neg p(x)$ e $\neg \exists y\, r(y,b)$. In ogni passaggio sottolineiamo l'enunciato su cui agiamo.

$$\underline{\neg(\exists x\, r(a,x) \wedge F \to \exists x\, \neg p(x))}$$
$$|$$
$$\underline{\exists x\, r(a,x) \wedge F}, G$$
$$|$$
$$\underline{\exists x\, r(a,x)}, F, G$$
$$|$$
$$r(a,b), \underline{F}, G$$
$$|$$
$$r(a,b), F, \underline{\exists y\, r(y,b) \to \neg p(b)}, G$$
$$\swarrow \qquad \searrow$$
$$r(a,b), F, \underline{H}, G \qquad\qquad r(a,b), F, \neg p(b), \underline{G}$$
$$| \qquad\qquad\qquad\qquad |$$
$$r(a,b), F, H, \neg r(a,b), G \qquad r(a,b), F, \neg p(b), G, p(b)$$
$$\otimes \qquad\qquad\qquad\qquad \otimes$$

Notate come nel passaggio in cui abbiamo agito su F avremmo potuto aggiungere l'istanza di F relativa a a (anziché, come abbiamo fatto, quella relativa a b). La scelta di b è però apparsa migliore perché il nostro obiettivo è chiudere il tableau: in particolare la presenza di $r(a,b)$ nell'etichetta del nodo ha suggerito di creare un'istanza in cui b compaia come secondo argomento di r. Considerazioni analoghe possono essere fatte per i nodi in cui abbiamo agito su H e su G.

Esempio 11.32. Usiamo il metodo dei tableaux per mostrare la validità di

$$\exists x\, \forall y\, r(y,x) \wedge \forall x\, \forall y(r(y,x) \to r(x,y)) \to \exists x\, r(x,a).$$

Indichiamo con F, G, H e K gli enunciati $\forall x\, \forall y(r(y,x) \to r(x,y))$, $\neg \exists x\, r(x,a)$, $\forall y\, r(y,b)$ e $\forall y(r(y,b) \to r(b,y))$. Come al solito sottolineiamo l'enunciato su cui agiamo.

$$
\begin{array}{c}
\underline{\neg(\exists x\,\forall y\, r(y,x) \wedge F \to \exists x\, r(x,a))}\\
|\\
\underline{\exists x\,\forall y\, r(y,x) \wedge F}, G\\
|\\
\underline{\exists x\,\forall y\, r(y,x)}, F, G\\
|\\
\underline{H}, F, G\\
|\\
H, r(a,b), \underline{F}, G\\
|\\
H, r(a,b), F, \underline{K}, G\\
|\\
H, r(a,b), F, K, \underline{r(a,b) \to r(b,a)}, G
\end{array}
$$

$$
\begin{array}{cc}
\swarrow & \searrow\\
H, r(a,b), F, K, \neg r(a,b), G & H, r(a,b), F, K, r(b,a), \underline{G}\\
\otimes & |\\
 & H, r(a,b), F, K, r(b,a), G, \neg r(b,a)\\
 & \otimes
\end{array}
$$

Anche in questo caso siamo riusciti ad ottenere una chiusura piuttosto rapida del tableau scegliendo opportunamente i simboli di costante su cui istanziare le γ-formule.

Gli enunciati di cui abbiamo dimostrato la validità negli esempi 11.31 e 11.32 verranno rivisitati dal punto di vista della deduzione naturale negli esempi 12.49 e 12.50.

Esercizio 11.33. Studiate con il metodo del tableaux gli enunciati degli esempi 7.44 e 7.47 e degli esercizi 7.16, 7.21, 7.50, 7.51 e 7.53 che non contengono simboli di funzione. Verificate che il risultato ottenuto con i tableaux coincida con quello ottenuto in precedenza.

5. La costruzione sistematica dei tableaux

Come evidenziato dall'esempio 11.14, l'algoritmo 11.16 non è sufficiente per dimostrare il teorema di completezza per i tableaux predicativi: se F è insoddisfacibile è possibile che un tableau per F non sia chiuso. Come già suggerito nella nota 11.15, la soluzione a questo problema è rendere l'algoritmo 11.16 "sistematico", eliminando alcuni aspetti di non determinismo che esso contiene.

In pratica si tratta di fare ogni possibile tentativo per chiudere il tableau, non "dimenticando" nessun enunciato e considerando ogni possibile istanza delle γ-formule che si incontrano (può essere utile riesaminare l'esempio 11.11, ed in particolare le considerazioni che hanno portato alla versione finale del tableau).

In sostanza vogliamo fare in modo che se r è un ramo aperto di un tableau e $S = \bigcup_{n \in r} E(n)$:

(a) se $G \in S$ non è un letterale allora in qualche nodo di r si agisce su G;
(b) se $G \in S$ è una γ-formula e c è un simbolo di costante che compare in qualche enunciato di S allora l'istanza di G relativa a c appartiene a S.

Per raggiungere questi obiettivi:

(a) agiremo sempre prima sugli enunciati che vengono eliminati (doppie negazioni, α-, β- e δ-formule), lasciando per ultime le γ-formule: questo ci assicura che nessuna doppia negazione, α-, β- o δ-formula venga "dimenticata" e che prima o poi si arriverà anche ad agire sulle γ-formule;

(b) affiancheremo a $E(n)$ una funzione C_n che assegna ad ogni γ-formula $G \in E(n)$ un insieme di simboli di costante $C_n(G)$: l'idea è che $c \in C_n(G)$ significa che l'istanza di G relativa a c deve essere ancora considerata. Quando agiamo su G (e in realtà agiremo su tutte le γ-formule di $E(n)$ simultaneamente) aggiungeremo all'etichetta del nodo che stiamo creando tutte le istanze di G relative a elementi di $C_n(G)$.

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 $\gamma(n) = \{\, G \in E(n) : G \text{ è } \gamma\text{-formula} \,\}$ un insieme finito di simboli di costante. Il tableau è costruito per stadi $\mathcal{T}_0, \dots, \mathcal{T}_i, \dots$: 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.

All'inizio della costruzione $\mathcal{T}_0$ consiste di un solo nodo (la *radice* dell'albero) etichettato con $\{F\}$ e, se F è una γ-formula, con la funzione che associa a F l'insieme dei simboli di costante che compaiono in F o, se non ce ne sono, l'insieme $\{c\}$ dove c è un simbolo di costante qualsiasi.

Allo stadio i diciamo che una foglia n dell'albero $\mathcal{T}_i$ è

foglia finale: se $E(n)$ contiene una coppia complementare di letterali, oppure se $E(n)$ contiene solo letterali e γ-formule G tali che $C_n(G) = \emptyset$;

γ-nodo: se $E(n)$ contiene solo letterali e γ-formule ma non è una foglia finale (questo significa che esiste qualche $G \in \gamma(n)$ per cui $C_n(G) \neq \emptyset$);

nodo ordinario: se non è né una foglia finale né un γ-nodo (questo significa che $E(n)$ contiene qualche doppia negazione, α-, β- o δ-formula).

Cerchiamo una foglia n dell'albero $\mathcal{T}_i$ che sia un nodo ordinario. Se esiste fissiamo n e scegliamo un enunciato $G \in E(n)$ che sia una doppia negazione, oppure una α-, β- o δ-formula:

(1) se G è una doppia negazione con ridotto G_1 aggiungiamo un nodo n' sotto n e poniamo $E(n') = (E(n) \setminus \{G\}) \cup \{G_1\}$; per ogni $H \in \gamma(n)$ poniamo $C_{n'}(H) = C_n(H)$; se G_1 è una γ-formula definiamo $C_{n'}(G_1)$ come l'insieme dei simboli di costante che compaiono in $E(n)$ (che coincidono con quelli che compaiono in $E(n')$) o, se non ce ne sono, l'insieme $\{c\}$ dove c è un simbolo di costante qualsiasi;

(2) se G è una α-formula con ridotti G_1 e G_2 aggiungiamo un nodo n' sotto n e poniamo $E(n') = (E(n) \setminus \{G\}) \cup \{G_1, G_2\}$; per ogni $H \in \gamma(n)$ poniamo $C_{n'}(H) = C_n(H)$; se G_j ($j = 1$ o $j = 2$) è una γ-formula definiamo $C_{n'}(G_j)$ come l'insieme dei simboli di costante che compaiono in $E(n)$ (che coincidono con quelli che compaiono in $E(n')$) o, se non ce ne sono, l'insieme $\{c\}$ dove c è un simbolo di costante qualsiasi;

(3) se G è una β-formula con ridotti G_1 e G_2 aggiungiamo due nodi tra loro inconfrontabili n_1 e n_2 sotto n e poniamo per $j = 1$ e $j = 2$, $E(n_j) = (E(n) \setminus \{G\}) \cup \{G_j\}$; per ogni $H \in \gamma(n)$ poniamo $C_{n_j}(H) = C_n(H)$; se G_j è una γ-formula definiamo $C_{n_j}(G_j)$ come l'insieme dei simboli di costante che compaiono in $E(n)$ (che coincidono con quelli che compaiono in $E(n_j)$) o, se non ce ne sono, l'insieme $\{c\}$ dove c è un simbolo di costante qualsiasi;

(4) se G è una δ-formula sia a una costante che non compare in $E(n)$ e sia G_1 l'istanza di G relativa ad a: aggiungiamo un nodo n' sotto n e poniamo $E(n') = (E(n) \setminus \{G\}) \cup \{G_1\}$; per ogni $H \in \gamma(n)$ poniamo $C_{n'}(H) =$

$C_n(H) \cup \{a\}$; se G_1 è una γ-formula definiamo $C_{n'}(G_1)$ come l'insieme dei simboli di costante che compaiono in $E(n)$ con l'aggiunta di a.

Se non esistono nodi ordinari cerchiamo una foglia n di $\mathcal{T}_i$ che sia un γ-nodo. Sia S_n l'insieme di tutte le istanze di γ-formule $G \in \gamma(n)$ relative a simboli di costante in $C_n(G)$ (notiamo che la definizione di γ-nodo implica che S_n non è vuoto). Aggiungiamo un nodo n' sotto n e poniamo $E(n') = E(n) \cup S_n$; per ogni $H \in \gamma(n)$ poniamo $C_{n'}(H) = \emptyset$; se $H \in S_n \setminus E(n)$ è una γ-formula definiamo $C_{n'}(H)$ come l'insieme dei simboli di costante che compaiono in $E(n)$ con l'aggiunta di tutti quelli utilizzati per ottenere le istanze di S_n.

Se infine nessuna foglia di $\mathcal{T}_i$ è un nodo ordinario o un γ-nodo arrestiamo la costruzione del tableau.

Esempio 11.35. Descriviamo un tableau sistematico per $\forall x\, \exists y\, r(x,y)$ (che indichiamo con F). Si tratta dell'enunciato all'origine dell'infinità del tableau dell'esempio 11.12.

F è una γ-formula e quindi alla radice n_1 del tableau possiamo porre $C_{n_1}(F) = \{a_1\}$. Dato che n_1 è un γ-nodo il nodo successivo m_1 è etichettato da F, $\exists y\, r(a_1,y)$ (S_{n_1} ha un solo elemento) e abbiamo $C_{m_1}(F) = \emptyset$.

Ora abbiamo un nodo ordinario e agiamo sulla δ-formula $\exists y\, r(a_1,y)$: il nodo successivo n_2 è etichettato da F, $r(a_1,a_2)$ e abbiamo $C_{n_2}(F) = \{a_2\}$.

n_2 è nuovamente un γ-nodo: il nodo successivo m_2 è etichettato da F, $r(a_1,a_2)$, $\exists y\, r(a_2,y)$ (S_{n_2} ha nuovamente un solo elemento) e abbiamo $C_{m_2}(F) = \emptyset$.

m_2 è un nodo ordinario in cui si agisce su una δ-formula ed è seguito da un γ-nodo n_3, poi da un altro nodo ordinario m_4 e così via. Quando si agisce su m_i si introduce il simbolo di costante a_{i+1} e l'istanza di F relativa a a_{i+1} verrà introdotta agendo su n_{i+1} (e quindi nell'etichetta di m_{i+1}). Inoltre $C_{n_i}(F) = \{a_i\}$ e $C_{m_i}(F) = \emptyset$. Nelle etichette dei nodi si aggiungono uno dopo l'altro gli enunciati atomici della forma $r(a_i, a_{i+1})$.

Il tableau suggerisce un'interpretazione con dominio infinito che soddisfa F: $D^I = \mathbb{N}$, $r^I = \{\, (i, i+1) : i \in \mathbb{N}\,\}$. Notiamo però che F è soddisfatta anche da interpretazioni con dominio finito (addirittura da un'interpretazione con dominio di un solo elemento!) che il tableau non è riuscito a individuare.

La dimostrazione del teorema di correttezza 11.29 può essere adattata facilmente ai tableaux sistematici (l'unico caso in cui c'è una reale differenza è quello delle γ-formule, in cui si può comunque sfruttare il lemma 11.6) ottenendo che anche il metodo dei tableaux sistematici è corretto.

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);
(b) dimostrazione che ogni insieme di Hintikka è soddisfacibile (lemma 11.43);
(c) dimostrazione che $\bigcup_{n \in r} E(n)$ è un insieme di Hintikka (lemma 11.45).

Dato che $F \in \bigcup_{n \in r} E(n)$ (perché la radice del tableau appartiene a r) questi passi sono sufficienti a completare la dimostrazione.

Nel seguito diamo la definizione di insieme di Hintikka, ed enunciamo precisamente i lemmi che corrispondono al secondo e al terzo passo, limitandoci a dare solamente alcuni cenni relativamente alle loro dimostrazioni. □

La definizione di insieme di Hintikka riprende quella del caso proposizionale (definizione 4.23): aggiungiamo le condizioni riguardanti γ- e δ-formule ed inoltre ci assicuriamo che vi siano simboli di costante.

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 $H_2 \in \mathcal{H}$ (è possibile che entrambi siano in $\mathcal{H}$);
(5) se $F \in \mathcal{H}$ è una γ-formula e H è un'istanza di F relativa ad un simbolo di costante che compare in $\mathcal{H}$ allora $H \in \mathcal{H}$;
(6) se $F \in \mathcal{H}$ è una δ-formula allora esiste una sua istanza H tale che $H \in \mathcal{H}$.

La definizione 11.38 è dettata dalla stessa idea descritta dopo la definizione 4.23: la verità di ogni $F \in \mathcal{H}$ che non è un letterale deve essere "giustificata" dalla verità di altri enunciati di $\mathcal{H}$. Il primo esempio evidenzia la necessità della condizione (0).

ESEMPIO 11.39. L'insieme $\mathcal{H} = \{\forall x(p(x) \wedge \neg p(x))\}$ soddisfa alle condizioni (1)–(6) ma non è un insieme di Hintikka perché non contiene nessun simbolo di costante. D'altronde $\mathcal{H}$ è insoddisfacibile e quindi perché valga il lemma 11.43 non deve essere un insieme di Hintikka.

ESEMPIO 11.40. L'insieme

$$\{\forall x(\neg r(x,a) \to \exists y\, r(x,y)), \neg r(a,a) \to \exists y\, r(a,y), \exists y\, r(a,y),\\ r(a,b), \neg r(b,a) \to \exists y\, r(b,y), \neg\neg r(b,a), r(b,a)\}$$

è un insieme di Hintikka. Infatti non contiene coppie complementari di letterali, contiene il ridotto della doppia negazione $\neg\neg r(b,a)$, uno dei ridotti di ciascuna delle β-formule $\neg r(a,a) \to \exists y\, r(y,a)$ e $\neg r(b,a) \to \exists y\, r(y,b)$, le istanze relative a a e b della γ-formula $\forall x(\neg r(x,a) \to \exists y\, r(y,x))$ ed una istanza (quella relativa a b) della δ-formula $\exists y\, r(y,a)$.

ESERCIZIO 11.41. Siano F_1 e F_2 gli enunciati $\forall x(\neg p(x) \wedge r(x,x))$ e $\exists x(p(x) \vee \neg r(x,x))$.

(i) Definite insiemi di Hintikka $\mathcal{H}_1$ e $\mathcal{H}_2$ con $F_i \in \mathcal{H}_i$.
(ii) $\mathcal{H}_1 \cup \mathcal{H}_2$ è un insieme di Hintikka?
(iii) ($\star$) Esiste un insieme di Hintikka che contiene sia F_1 che F_2?

ESERCIZIO 11.42. Verificare che l'insieme di enunciati che compaiono nelle etichette del ramo aperto del tableau ottenuto al termine dell'esempio 11.11 è un insieme di Hintikka.

LEMMA 11.43 (Lemma di Hintikka). *Ogni insieme di Hintikka è soddisfacibile.*

CENNO DI DIMOSTRAZIONE. Sia $\mathcal{H}$ un insieme di Hintikka e sia $\mathcal{C}$ l'insieme dei simboli di costante che compaiono in $\mathcal{H}$. Per definizione di insieme di Hintikka si ha $\mathcal{C} \neq \emptyset$.

Definiamo un'interpretazione I per il linguaggio di $\mathcal{H}$ ponendo: $D^I = \mathcal{C}$, $c^I = c$ per ogni $c \in \mathcal{C}$, $p^I = \{ (c_1, \dots, c_n) \in \mathcal{C}^n : p(c_1, \dots, c_n) \in \mathcal{H} \}$ per ogni simbolo di relazione n-ario p che compare in $\mathcal{H}$ (ricordiamo che stiamo supponendo che il linguaggio di $\mathcal{H}$ non contenga simboli di funzione).

Si dimostra poi, estendendo prima la definizione di rango di una formula e poi i ragionamenti svolti per dimostrare il lemma 4.27 (che è la versione proposizionale di questo lemma), che per ogni $F \in \mathcal{H}$ si ha $I \models F$. Questo implica che $\mathcal{H}$ è soddisfacibile. □

Esempio 11.44. Applicando la dimostrazione del lemma 11.43 all'insieme di Hintikka $\mathcal{H}$ costruito nell'esempio 11.40 si ottiene l'interpretazione I definita da $D^I = \{a, b\}$, $a^I = a$, $b^I = b$, $r^I = \{(a, b), (b, a)\}$.

Lemma 11.45. *Se r è un ramo aperto (finito o infinito) di un tableau sistematico allora $\mathcal{H} = \bigcup_{n \in r} E(n)$ è un insieme di Hintikka.*

Cenno di dimostrazione. Bisogna verificare che $\mathcal{H}$ soddisfa le sette proprietà della definizione di insieme di Hintikka. La dimostrazione è anche in questo caso una generalizzazione di quella del caso proposizionale (lemma 4.28). Ci sono però diverse difficolta in più, principalmente dovute al fatto che r può essere infinito (ed in quel caso anche $\mathcal{H}$ sarà infinito).

Preliminarmente si osserva che in $\mathcal{H}$ compare almeno un simbolo di costante. Successivamente è necessario dimostrare che si agisce su ogni enunciato di $\mathcal{H}$ che non è un letterale. Inoltre il caso delle γ-formule richiede particolare attenzione: bisogna infatti verificare che **tutte** le loro istanze relative a simboli di costante che compaiono in $\mathcal{H}$ appartengono a $\mathcal{H}$. □

La dimostrazione del teorema di completezza accennata sopra fornisce informazioni su come trovare un'interpretazione che soddisfi un enunciato il cui tableau è aperto.

Lemma 11.46. *Sia r un ramo di un tableau sistematico per l'enunciato F che è aperto. Siano $\mathcal{H} = \bigcup_{n \in r} E(n)$ e $\mathcal{C}$ l'insieme dei simboli di costante che compaiono nelle etichette dei nodi di r (si può dimostrare che $\mathcal{C} \neq \emptyset$).*

Sia I l'interpretazione definita da: $D^I = \mathcal{C}$, $c^I = c$ per ogni $c \in \mathcal{C}$ che compare in F, $p^I = \{ (c_1, \dots, c_n) \in \mathcal{C}^n : p(c_1, \dots, c_n) \in \mathcal{H} \}$ per ogni simbolo di relazione n-ario p che compare in F. Allora $I \models F$.

Un esempio di applicazione di questo lemma è contenuto nell'esempio 11.35. Anche l'esempio 11.11 contiene un esempio di questo tipo: in questo caso il tableau ottenuto al termine di quell'esempio non soddisfa esattamente alla nostra definizione di tableau sistematico (perché le istanze di H relative ad a e b non vengono introdotte simultaneamente), ma è stato costruito senza dimenticare nessun enunciato e soddisfa al teorema di completezza.

Nota 11.47. I teoremi di correttezza e completezza mostrano che il metodo dei tableaux sistematici è una procedura di semidecisione per l'insoddisfacibilità (e quindi per il lemma 7.48 anche per la validità) degli enunciati della logica predicativa. Questo significa che se l'enunciato F è insoddisfacibile l'algoritmo 11.34 si fermerà fornendoci la risposta corretta; se invece F è soddisfacibile l'algoritmo può arrestarsi e dare la risposta corretta, oppure proseguire la sua esecuzione all'infinito. In pratica, quindi, il metodo dei tableaux predicativi è utile soprattutto per dimostrare l'insoddisfacibilità di un enunciato.

Nel 1936 Alonzo Church[1], basandosi sul lavoro precedente di Alan Turing, ha dimostrato il seguente teorema.

[1] Alonzo Church (1903-1995) è stato un matematico e logico statunitense.

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 tavole di verità ed il metodo dei tableaux).

NOTA 11.49. Come già osservato nel caso proposizionale (nota 4.14), anche nel caso predicativo l'algoritmo di costruzione dei tableaux **non** prevede la sostituzione di un enunciato con un altro logicamente equivalente ad esso. Se introducessimo questa possibilità, il teorema di completezza non sarebbe più vero, perché un enunciato insoddisfacibile potrebbe avere tableaux con rami infiniti, e quindi aperti.

7. Tableaux per la conseguenza logica

Questa sezione ripercorre ciò che è stato detto nella sezione 4.6.

Abbiamo presentato il metodo dei tableaux come un metodo per studiare la validità o la soddisfacibilità di un singolo enunciato. Usando la parte dell'esercizio 7.49 riguardante il lemma 2.43 possiamo usare i tableaux per studiare la validità o la soddisfacibilità di un insieme finito di enunciati $\{F_1, \dots, F_n\}$. Come nel caso proposizionale si ottiene il seguente algoritmo.

ALGORITMO 11.50. Per stabilire se un insieme finito $T = \{F_1, \dots, F_n\}$ 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 usare i tableaux per stabilire la sussistenza della relazione di conseguenza logica. Come nel caso proposizionale si ottiene il seguente algoritmo.

ALGORITMO 11.51. Per stabilire se $F_1, \dots, F_n \vDash G$ costruiamo un tableau con radice etichettata da $\{F_1, \dots, F_n, \neg G\}$. Se il tableau è chiuso abbiamo $F_1, \dots, F_n \vDash G$. Se il tableau è aperto e sistematico allora $F_1, \dots, F_n \nvDash G$, e un ramo aperto del tableau permette di definire un'interpretazione che soddisfa $F_1, \dots, F_n$ ma non G.

ESEMPIO 11.52. Usiamo il metodo dei tableaux per stabilire che

$$\forall x\, p(x), \forall x(\forall z\, r(x,z) \to \neg p(x)) \vDash \forall y\, \exists x\, \neg r(y,x).$$

Seguendo le indicazioni dell'algoritmo 11.51 (e le convenzioni 11.19 e 11.21) otteniamo il seguente tableau, in cui indichiamo con F, G e H le γ-formule $\forall x\, p(x)$, $\forall x(\forall z\, r(x,z) \to \neg p(x))$ e $\neg\exists x\, \neg r(a,x)$ (come al solito in ogni passaggio sottolineiamo l'enunciato su cui agiamo):

$$
\begin{array}{c}
F, G, \underline{\neg\forall y\,\exists x\,\neg r(y,x)} \\
| \\
F, \underline{G}, H \\
| \\
F, G, \underline{\forall z\, r(a,z) \to \neg p(a)}, H \\
\swarrow \qquad \searrow \\
\begin{array}{cc}
\begin{array}{c}
F, G, \underline{\neg\forall z\, r(a,z)}, H \\
| \\
F, G, \neg r(a,b), \underline{H} \\
| \\
F, G, \neg r(a,b), H, r(a,b) \\
\otimes
\end{array}
&
\begin{array}{c}
\underline{F}, G, \neg p(a), H \\
| \\
F, p(a), G, \neg p(a), H \\
\otimes \\
\\
\\
\end{array}
\end{array}
\end{array}
$$

Esercizio 11.53. Studiate con il metodo del tableaux le conseguenze logiche dei lemmi 8.5, 8.7, 8.9 e 8.14, degli esempi 7.32 e 8.11 e degli esercizi 7.33 e 8.17 (verificate che il risultato ottenuto con i tableaux coincida con quello ottenuto in precedenza). Per i problemi che riguardano generiche F e G utilizzate enunciati atomici specifici. Quando avete a che fare con un'equivalenza logica usate l'esercizio 7.37.

8. Tableaux per la logica con uguaglianza

Affrontiamo ora lo studio della logica con uguaglianza attraverso i tableaux. Vogliamo cioè utilizzare i tableaux per stabilire se un enunciato G è valida nella logica con uguaglianza o, più in generale, se $F_1, \dots, F_n \models_= G$ (quando $n = 0$ abbiamo appunto la validità di G).

Possiamo sfruttare il teorema 10.34, che ci dice che $F_1, \dots, F_n \models_= G$ è equivalente a $F_1, \dots, F_n, \mathsf{Eq}_\mathcal{L} \models G$, dove $\mathcal{L} = \mathcal{L}(F_1, \dots, F_n, G)$. Dato che $\mathcal{L}$ contiene solo un numero finito di simboli di relazione (e nessun simbolo di funzione, per la convenzione 11.1), l'insieme degli assiomi dell'uguaglianza $\mathsf{Eq}_\mathcal{L}$ è finito, e quindi possiamo applicare l'algoritmo 11.51, ottenendo il seguente algoritmo.

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

Esempio 11.55. Utilizziamo i tableaux per dimostrare $p(a), \neg p(b) \models_= a \neq b$. Sia $\mathcal{L}$ il linguaggio di questi enunciati: oltre all'uguaglianza, vi compaiono i simboli di costante a e b e il simbolo di relazione unario p. Nel tableau che costruiremo avremo bisogno di un solo elemento di $\mathsf{Eq}_\mathcal{L}$, e precisamente dell'unico enunciato di tipo (e5) di $\mathsf{Eq}_\mathcal{L}$ (si ricordi la definizione 6.64): $\forall x\, \forall y (x = y \wedge p(x) \to p(y))$, che indichiamo con E_5. Indichiamo con E_5' l'enunciato $\forall y (a = y \wedge p(a) \to p(y))$, che è un'istanza di E_5. Ecco il tableau, da cui omettiamo gli altri elementi di $\mathsf{Eq}_\mathcal{L}$ (come al solito sottolineiamo l'enunciato su cui agiamo):

$$p(a), \neg p(b), a = b, \underline{E_5}$$

$$|$$

$$p(a), \neg p(b), a = b, E_5, \underline{E_5'}$$

$$|$$

$$p(a), \neg p(b), a = b, E_5, E_5', \underline{a = b \wedge p(a) \to p(b)}$$

$$p(a), \neg p(b), a = b, E_5, E_5', \underline{\neg(a = b \wedge p(a))} \qquad p(a), \neg p(b), a = b, E_5, E_5', p(b)$$
$$\otimes$$

$$p(a), \neg p(b), a = b, E_5, E_5', a \neq b \qquad p(a), \neg p(b), a = b, E_5, E_5', \neg p(a)$$
$$\otimes \qquad \otimes$$

Esercizio 11.56. Mostrare utilizzando i tableaux che

$$\forall x\, r(a, x), \exists x\, \exists y\, \neg r(y, x) \models_= \exists x\, a \neq x.$$

Le conseguenze logiche con uguaglianza dell'esempio 11.55 e dell'esercizio 11.56 verranno considerate dal punto di vista della deduzione naturale negli esempi 12.55 e 12.62.

9. Tableaux per linguaggi con simboli di funzione

Supponiamo di voler utilizzare il metodo dei tableaux per enunciati di un linguaggio che comprenda anche dei simboli di funzione. Se non ci preoccupiamo di avere un tableau sistematico, la modifica che dobbiamo fare all'algoritmo 11.16 riguarda solo le γ-formule. Nella nuova situazione quando agiamo su una γ-formula possiamo aggiungere un'istanza relativa ad un termine chiuso (ricordare la definizione 6.9) che contenga anche simboli di funzione. Per prima cosa è quindi necessario estendere la nozione di istanza di una γ-formula a termini chiusi arbitrari.

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 costante (il lemma 7.63 vale per termini arbitrari).

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 del teorema 11.29. Infatti nel caso (4) di quella dimostrazione si utilizzava il lemma 11.6, che (come osservato sopra) vale anche per istanze relative a termini chiusi arbitrari.

Esempio 11.59. Utilizziamo i tableaux per mostrare che

$$\forall x(p(x) \to r(x, f(x))), \exists y\, p(g(y)) \models \exists x\, \exists y\, r(g(x), f(y)).$$

Indichiamo con F, G e H gli enunciati $\forall x(p(x) \to r(x, f(x)))$, $\neg\exists x\, \exists y\, r(g(x), f(y))$ e $\neg\exists y\, r(g(a), f(y))$. Come al solito sottolineiamo l'enunciato su cui agiamo.

$$
\begin{array}{c}
F, \underline{\exists y\, p(g(y))}, G \\
| \\
\underline{F}, p(g(a)), G \\
| \\
F, \underline{p(g(a)) \to r(g(a), f(g(a)))}, p(g(a)), G \\
\swarrow \qquad \searrow \\
\begin{array}{cc}
\begin{array}{c} F, \neg p(g(a)), p(g(a)), G \\ \otimes \end{array}
&
\begin{array}{c}
F, r(g(a), f(g(a))), p(g(a)), \underline{G} \\
| \\
F, r(g(a), f(g(a))), p(g(a)), G, \underline{H} \\
| \\
F, r(g(a), f(g(a))), p(g(a)), G, H, \neg r(g(a), f(g(a))) \\
\otimes
\end{array}
\end{array}
\end{array}
$$

Notiamo come nell'esempio precedente le azioni sulle δ-formule non presentano differenze rispetto all'algoritmo 11.16: infatti abbiamo istanziato $\exists y\, p(g(y))$ sul nuovo simbolo di costante a. Quando abbiamo agito sulle γ-formule abbiamo utilizzato la novità dell'algoritmo 11.58: abbiamo infatti istanziato F, G e H rispettivamente su $g(a)$, a e $g(a)$. Ancor più che nel caso in cui non erano presenti simboli di funzione, bisogna scegliere con cura i termini chiusi su cui istanziare le γ-formule per riuscire a chiudere in meno passaggi possibili il tableau.

Esercizio 11.60. Dimostrare con il metodo dei tableaux che l'insieme di enunciati
$$\{\forall x(p(x) \lor q(x)), \forall x(q(f(x)) \to p(x)), \neg\forall x(\neg p(x) \to p(f(x)))\}$$
è insoddisfacibile.

Se vogliamo costruire tableaux sistematici per ottenere la completezza dell'algoritmo la presenza di simboli di funzione causa qualche problema in più. Infatti in un linguaggio con simboli di funzione ed almeno un simbolo di costante l'insieme dei termini chiusi è infinito (l'esempio 6.16 descrive il caso più semplice possibile di questa situazione). Nell'algoritmo 11.34, nel caso in cui si trattavano le γ-formule, si aggiungeva all'etichetta del nodo l'insieme di tutte le istanze di γ-formule $G \in E(n)$ relative a simboli di costante in $C_n(G)$. Dato che ora sarebbe necessario considerare ogni possibile istanza relativa a termini chiusi di ogni γ-formula, ciò condurrebbe ad avere nodi etichettati con insiemi infiniti di enunciati, che non è accettabile. Per superare questo ostacolo è necessario aggiungere ad ogni passo solo un numero finito di istanze di γ-formule, pur garantendo che ogni istanza venga prima o poi aggiunta. A tal fine è opportuno ordinare i termini chiusi e seguire l'ordine nella scelta delle istanze da aggiungere. Ciò comporta che i tableaux aperti per enunciati di linguaggi con simboli di funzione sono sempre infiniti (a meno che la formula sia talmente semplice da condurre ad un tableau privo di γ-formule).

In questa sede omettiamo gli ulteriori dettagli necessari alla definizione rigorosa dei tableaux sistematici per linguaggi con simboli di funzione.

CAPITOLO 12

La deduzione naturale: caso predicativo

In questo capitolo estendiamo alla logica predicativa il sistema deduttivo introdotto per la logica proposizionale nel capitolo 5. Continua ad essere utile la formula $\perp$ che rappresenta il falso: nel contesto predicativo si tratta di un enunciato atomico che non è soddisfatto da nessuna interpretazione in nessuno stato.

Scriveremo sempre $T \rhd F$ per indicare 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 sempre la stessa e tutte le regole della deduzione naturale proposizionale fanno parte del nuovo sistema. Dobbiamo quindi solamente aggiungere regole di introduzione ed eliminazione per i quantificatori.

1. La deduzione naturale e i quantificatori

Ogni quantificatore ha una regola di introduzione ed una di eliminazione, esattamente come le avevano i connettivi proposizionali. Vedremo come ognuna di queste regole richieda che certe condizioni siano soddisfatte dalle variabili e dai termini coinvolti. In questa sezione cercheremo di spiegare intuitivamente le ragioni per introdurre queste condizioni, che peraltro risulteranno essenziali nella dimostrazione del teorema di correttezza 12.11. A questo proposito può essere utile anche lo svolgimento dell'esercizio 12.12.

Per eliminare $\forall$ dobbiamo capire cosa possiamo dedurre se abbiamo già dedotto $\forall x\, F$. In questo caso sappiamo che F vale per qualunque oggetto vogliamo considerare: abbiamo cioè la possibilità di passare dall'universale al particolare. Dato che gli oggetti sono rappresentati dai termini, è naturale poter dedurre $F\{x/t\}$ per qualunque termine t, a condizione che la sostituzione di x con t sia ammissibile in F. Nella notazione della sezione 5.1 possiamo quindi scrivere $\forall x\, F \vdash F\{x/t\}$, se la sostituzione $\{x/t\}$ è ammissibile in F. In termini di deduzione naturale questa regola corrisponde all'albero

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

La correttezza di questa regola è espressa dal lemma 11.6.

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

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

La correttezza di questa regola è giustificata dall'esercizio 11.7.

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

Esempio 12.1. Mostriamo che $\forall x(p(x) \to q(x)), p(f(a)) \rhd \exists x\, q(f(x))$:

$$\dfrac{\dfrac{p(f(a)) \qquad \dfrac{\forall x(p(x) \to q(x))}{p(f(a)) \to q(f(a))}}{q(f(a))}}{\exists x\, q(f(x))}$$

Abbiamo iniziato la deduzione naturale utilizzando $(\forall e)$ (sulla formula $p(x) \to q(x)$ e sul termine $f(a)$), abbiamo poi sfruttato $(\to e)$, per concludere con un'applicazione di $(\exists i)$ (alla formula $q(f(x))$ e al termine a). Il ragionamento rappresentato è in effetti quello "naturale" in un caso del genere: se sappiamo che ogni elemento con la proprietà p possiede anche la proprietà q, e sappiamo che $f(a)$ ha la proprietà p possiamo dedurre che $f(a)$ ha la proprietà q e quindi che esiste un elemento nell'immagine di f con quest'ultima proprietà.

Notiamo anche che da $\forall x(p(x) \to q(x))$ avremmo potuto dedurre $p(a) \to q(a)$ invece di $p(f(a)) \to q(f(a))$. Se avessimo fatto così la deduzione naturale non avrebbe però potuto proseguire, vista l'impossibilità di utilizzare $(\to e)$ o qualsiasi altra regola. Analogamente, da $q(f(a))$ si poteva usare $(\exists i)$ per dedurre $\exists x\, q(x)$ (formula $q(x)$ e termine $f(a)$), che non è però la conclusione desiderata.

Consideriamo ora l'introduzione di $\forall$. 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 mostrarlo aprendo una specifica 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". Dobbiamo però 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, estremamente simile alla prima parte di quella dell'esempio 12.1:

$$\dfrac{c(x) \qquad \dfrac{\forall x(c(x) \to m(x))}{c(x) \to m(x)}}{m(x)}$$

Se interpretiamo $c(x)$ come "x è un cane" e $m(x)$ come "x è un mammifero" siamo giunti alla conclusione 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.

Riassumendo: se abbiamo $T \vdash F$ e x non è libera in nessuna formula di T (nel corso della dimostrazione del teorema 12.11 si vedrà come questa condizione garantisca la correttezza della regola), allora possiamo dedurre $\forall x\, F$. Otteniamo dunque l'albero

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

Ecco un esempio in cui viene utilizzata questa regola.

Esempio 12.2. Mostriamo che $\forall x(p(x) \to q(x)), \forall x\, p(x) \rhd \forall x\, q(x)$:

$$\dfrac{\dfrac{\dfrac{\forall x\, p(x)}{p(x)} \qquad \dfrac{\forall x(p(x) \to q(x))}{p(x) \to q(x)}}{q(x)}}{\forall x\, q(x)}$$

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 x generico, ottenendo che esso soddisfa la proprietà q; la genericità di x ci assicura che ogni elemento ha quest'ultima proprietà.

Rimane da considerare la regola di eliminazione di $\exists$. In questo caso ci può aiutare l'osservazione (già fatta nella nota 8.18) 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 si assegna un nome generico (per esempio la variabile x stessa) all'oggetto di cui si conosce l'esistenza, e 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 e se x non è libera in nessuna formula di $T \setminus \{F\}$ e neppure in G (nel corso della dimostrazione del teorema 12.11 si vedrà come queste condizioni garantiscano la correttezza della regola), allora possiamo dedurre G da $T, \exists x\, F$. Otteniamo dunque l'albero

$$\dfrac{\exists x\, F \qquad \begin{array}{c} T, [F] \\ \triangledown \\ G \end{array}}{G} \qquad x \text{ non libera in } T \setminus \{F\}, G$$

Ecco un esempio in cui viene utilizzata la regola di eliminazione di $\exists$.

Esempio 12.3. Mostriamo che $\forall x(p(x) \to q(x)), \exists x\, p(x) \rhd \exists x\, q(x)$:

$$\dfrac{\exists x\, p(x) \qquad \dfrac{\dfrac{[p(x)]^1 \qquad \dfrac{\forall x(p(x) \to q(x))}{p(x) \to q(x)}}{q(x)}}{\exists x\, q(x)}}{\exists x\, q(x)}\,1$$

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 perciò esiste un elemento con quest'ultima proprietà. Quest'ultima affermazione non riguarda x e perciò essa non dipende dall'aver usato quel nome per designare un elemento con la proprietà p.

Notiamo che in questa deduzione naturale è necessario che $(\exists i)$ venga utilizzato prima di $(\exists e)$: se avessimo invertito l'ordine di applicazione di queste due regole avremmo applicato $(\exists e)$ alla formula $q(x)$ (in cui x è libera), violando la condizione per l'applicabilità di $(\exists e)$ (e in effetti $\forall x(p(x) \to q(x)), \exists x\, p(x) \not\models q(x)$).

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 12.4. Supponiamo che F sia una formula qualunque e di voler mostrare che $\forall x(p(x) \to q(x)), F \wedge \forall x\, p(x) \rhd \forall x\, q(x)$. Con una piccola modifica alla deduzione naturale dell'esempio 12.2 si ottiene

$$\dfrac{\dfrac{\dfrac{\dfrac{F \wedge \forall x\, p(x)}{\forall x\, p(x)}}{p(x)} \qquad \dfrac{\forall x(p(x) \to q(x))}{p(x) \to q(x)}}{q(x)}}{\forall x\, q(x)}$$

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

Un esempio del tutto analogo può essere fatto per $(\exists e)$ modificando l'esempio 12.3 in modo che $F \wedge \forall x(p(x) \to q(x)), \exists x\, p(x) \rhd \exists x\, q(x)$. Anche in questo caso se x è libera in F la deduzione naturale

$$\dfrac{\exists x\, p(x) \qquad \dfrac{\dfrac{[p(x)]^1 \qquad \dfrac{\dfrac{F \wedge \forall x(p(x) \to q(x))}{\forall x(p(x) \to q(x))}}{p(x) \to q(x)}}{q(x)}}{\exists x\, q(x)}}{\exists x\, q(x)}\ {}_1$$

non è accettabile.

La soluzione naturale a queste difficoltà è quella di utilizzare per indicare l'elemento generico una variabile diversa da x (ciò è giustificato dalle equivalenze logiche del lemma 7.61). Si ottengono quindi gli alberi

$$\dfrac{\dfrac{\dfrac{\dfrac{F \wedge \forall x\, p(x)}{\forall x\, p(x)}}{p(y)} \qquad \dfrac{\forall x(p(x) \to q(x))}{p(y) \to q(y)}}{q(y)}}{\forall x\, q(x)}$$

e

$$\dfrac{\exists x\, p(x) \qquad \dfrac{\dfrac{[p(y)]^1 \qquad \dfrac{\dfrac{F \wedge \forall x(p(x) \to q(x))}{\forall x(p(x) \to q(x))}}{p(y) \to q(y)}}{q(y)}}{\exists x\, q(x)}}{\exists x\, q(x)}\ {}_1$$

Notiamo però che nel primo caso deduciamo $\forall x\, q(x)$ da $q(y)$, mentre nel secondo caso utilizziamo $p(y)$ come ipotesi ausiliaria che scarichiamo in presenza di $\exists x\, p(x)$. In entrambi i casi non stiamo applicando le regole introdotte in precedenza.

Per poter enunciare le generalizzazioni di $(\forall i)$ e $(\exists e)$, che indicheremo con $(\forall i)^g$ 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

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

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

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

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.

In realtà $(\forall i)^g$ e $(\exists e)^g$ si possono ottenere da $(\forall i)$ e $(\exists e)$ e dalle altre regole. Per farlo è necessario partire da un'osservazione.

Lemma 12.5. *Siano F una formula ed x e y due variabili distinte. Supponiamo che y non sia libera in F e che la sostituzione $\{x/y\}$ sia ammissibile in F. Allora la sostituzione $\{y/x\}$ è ammissibile in $F\{x/y\}$ e $F\{x/y\}\{y/x\}$ coincide con F.*

Dimostrazione. L'effetto della sostituzione $\{x/y\}$ su F è quello di rimpiazzare tutte le occorrenze libere di x con y. Dato che y non è libera in F, quelle ottenute in questo modo sono tutte le occorrenze libere di y in $F\{x/y\}$. In particolare nessuna occorrenza libera di y in $F\{x/y\}$ è nell'ambito d'azione di un quantificatore su x, e perciò $\{y/x\}$ è ammissibile in $F\{x/y\}$. Se effettuiamo questa sostituzione rimpiazziamo tutte le occorrenze di y con x, e quindi ritorniamo alla situazione di partenza, ovvero a F. □

Esercizio 12.6. Sia F la formula $r(x, y)$. La sostituzione $\{x/y\}$ è ammissibile in F? La sostituzione $\{y/x\}$ è ammissibile in $F\{x/y\}$? Scrivete $F\{x/y\}\{y/x\}$. Quale ipotesi del lemma 12.5 è stata violata?

Ripetete l'esercizio quando F è $\forall x\, r(x, y) \land p(x)$.

Esempio 12.7. In questo esempio dimostriamo $(\forall i)^g$. Quando x e y coincidono abbiamo $(\forall i)$ e non c'è nulla da dimostrare. Supponiamo quindi che x e y siano variabili distinte, che $\{x/y\}$ sia ammissibile in F, che $T \rhd F\{x/y\}$, e che y non sia libera in nessuna formula di $T, \forall x\, F$. Notiamo che y non è libera in F: quindi $F\{x/y\}\{y/x\}$ è F per il lemma 12.5. Mostriamo $T \rhd \forall x\, F$ come segue

$$\dfrac{\dfrac{\begin{array}{c}T\\ \triangledown\\ F\{x/y\}\end{array}}{\forall y\, F\{x/y\}} \qquad \dfrac{\dfrac{\dfrac{[\forall y\, F\{x/y\}]^1}{F\{x/y\}\{y/x\}}}{\forall x\, F}}{\forall y\, F\{x/y\} \to \forall x\, F}\,{}^{1}}{\forall x\, F}$$

In questa deduzione abbiamo applicato $(\forall i)$ due volte. Sulla sinistra abbiamo usato la variabile y, ma senza cambiare la formula, mentre sulla destra sfruttiamo che $F\{x/y\}\{y/x\}$ è F e il fatto che x non è libera in $\forall y\, F\{x/y\}$ (perché tutte le occorrenze libere di x sono state rimpiazzate da y).

Notiamo che nel corso di questa deduzione naturale abbiamo anche mostrato, nelle ipotesi del lemma 12.5, $\forall y\, F\{x/y\} \rhd \forall x\, F$.

Esercizio 12.8. Spiegare perché la deduzione dell'esempio 12.7 non può essere (in generale) rimpiazzata dalla seguente:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}T\\ \triangledown\\ F\{x/y\}\end{array}}{\forall y\, F\{x/y\}}}{F\{x/y\}\{y/x\}}}{\forall x\, F}$$

Esempio 12.9. In questo esempio dimostriamo $(\exists e)^g$. Se le variabili x e y coincidono abbiamo $(\exists e)$ e non c'è nulla da dimostrare. Supponiamo quindi che x e y siano variabili distinte, che $\{x/y\}$ sia ammissibile in F, che $T, F\{x/y\} \rhd G$ e che y non sia libera in nessuna formula di $T \setminus \{F\{x/y\}\}, \exists x\, F, G$. Nuovamente abbiamo che y non è libera in F e quindi $F\{x/y\}\{y/x\}$ è F per il lemma 12.5. Mostriamo che $T, \exists x\, F \rhd G$ come segue

$$\dfrac{\dfrac{\exists x\, F \qquad \dfrac{[F\{x/y\}\{y/x\}]^1}{\exists y\, F\{x/y\}}}{\exists y\, F\{x/y\}}\,{}^{1} \qquad \begin{array}{c}T, [F\{x/y\}]^2\\ \triangledown\\ G\end{array}}{G}\,{}^{2}$$

In questa deduzione abbiamo applicato $(\exists e)$ due volte. Nella prima occasione scriviamo $F\{x/y\}\{y/x\}$ al posto di F, ed è immediato che x non è libera in $\exists y\, F\{x/y\}$. Nella seconda sfruttiamo l'ipotesi che y non è libera in $T \setminus \{F\{x/y\}\}, G$.

Notiamo che nel corso di questa deduzione naturale abbiamo anche mostrato, nelle ipotesi del lemma 12.5, $\exists x\, F \rhd \exists y\, F\{x/y\}$.

Esempio 12.10. Per mostrare che

$$\exists x\, p(x), \exists x\, \neg p(x), \forall x\, \forall y (p(x) \wedge \neg p(y) \to r(x,y)) \rhd \exists x\, \exists y\, r(x,y).$$

$(\exists e)^g$ risulta estremamente utile, perché non avrebbe senso usare la stessa variabile x per indicare sia l'elemento che soddisfa $p(x)$ che quello che soddisfa $\neg p(x)$:

$$\dfrac{\exists x\, p(x) \qquad \dfrac{\exists x\, \neg p(x) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[p(x)]^2 \qquad [\neg p(y)]^1}{p(x) \wedge \neg p(y)} \qquad \dfrac{\dfrac{\forall x\, \forall y (p(x) \wedge \neg p(y) \to r(x,y))}{\forall y (p(x) \wedge \neg p(y) \to r(x,y))}}{p(x) \wedge \neg p(y) \to r(x,y)}}{r(x,y)}}{\exists y\, r(x,y)}}{\exists x\, \exists y\, r(x,y)}}{\exists x\, \exists y\, r(x,y)}\,{}^{1}}{\exists x\, \exists y\, r(x,y)}\,{}^{2}$$

2. Le regole della deduzione naturale predicativa

Per favorire la consultazione, in questa sezione (in analogia a quanto fatto nella sezione 5.3) riassumiamo tutte le regole del sistema di deduzione naturale predicativa. Esse comprendono le regole proposizionali:

$$(\wedge i)\quad \frac{\begin{array}{c} T \\ \triangledown \\ F \end{array} \qquad \begin{array}{c} T' \\ \triangledown \\ G \end{array}}{F \wedge G} \qquad\qquad (\wedge e.1)\quad \frac{\begin{array}{c} T \\ \triangledown \\ F \wedge G \end{array}}{F} \qquad\qquad (\wedge e.2)\quad \frac{\begin{array}{c} T \\ \triangledown \\ F \wedge G \end{array}}{G}$$

$$(\vee i.1)\quad \frac{\begin{array}{c} T \\ \triangledown \\ F \end{array}}{F \vee G} \qquad (\vee i.2)\quad \frac{\begin{array}{c} T \\ \triangledown \\ G \end{array}}{F \vee G} \qquad (\vee e)\quad \frac{\begin{array}{c} T \\ \triangledown \\ F \vee G \end{array} \qquad \begin{array}{c} T', [F] \\ \triangledown \\ H \end{array} \qquad \begin{array}{c} T'', [G] \\ \triangledown \\ H \end{array}}{H}$$

$$(\rightarrow i)\quad \frac{\begin{array}{c} T, [F] \\ \triangledown \\ G \end{array}}{F \rightarrow G} \qquad\qquad (\rightarrow e)\quad \frac{\begin{array}{c} T \\ \triangledown \\ F \end{array} \qquad \begin{array}{c} T' \\ \triangledown \\ F \rightarrow G \end{array}}{G}$$

$$(\neg i)\quad \frac{\begin{array}{c} T, [F] \\ \triangledown \\ \bot \end{array}}{\neg F} \qquad\qquad (\neg e)\quad \frac{\begin{array}{c} T \\ \triangledown \\ F \end{array} \qquad \begin{array}{c} T' \\ \triangledown \\ \neg F \end{array}}{\bot} \qquad\qquad (\neg\neg e)\quad \frac{\begin{array}{c} T \\ \triangledown \\ \neg\neg F \end{array}}{F}$$

Poi abbiamo le regole per i quantificatori, che sono caratterizzate dall'essere applicabili solo se le condizioni indicate alla loro destra sono soddisfatte:

$$(\forall i)\quad \frac{\begin{array}{c} T \\ \triangledown \\ F \end{array}}{\forall x\, F} \quad x \text{ non libera in } T \qquad\qquad (\forall e)\quad \frac{\begin{array}{c} T \\ \triangledown \\ \forall x\, F \end{array}}{F\{x/t\}} \quad \{x/t\} \text{ ammissibile in } F$$

$$(\exists i)\quad \frac{\begin{array}{c} T \\ \triangledown \\ F\{x/t\} \end{array}}{\exists x\, F} \quad \{x/t\} \text{ ammissibile in } F$$

$$(\exists e)\quad \frac{\begin{array}{c} T' \\ \triangledown \\ \exists x\, F \end{array} \qquad \begin{array}{c} T, [F] \\ \triangledown \\ G \end{array}}{G} \quad x \text{ non libera in } T \setminus \{F\}, G$$

Infine, ricordiamo anche le regole derivate ottenute nel capitolo 5 e quelle ottenute negli esempi 12.7 e 12.9:

$$(RAA)\quad \frac{\begin{array}{c} T, [\neg F] \\ \triangledown \\ \bot \end{array}}{F} \qquad\qquad (\textit{ex-falso})\quad \frac{\begin{array}{c} T \\ \triangledown \\ \bot \end{array}}{F}$$

$$(MT) \quad \dfrac{\begin{matrix} T \\ \triangledown \\ F \to G \end{matrix} \qquad \begin{matrix} T' \\ \triangledown \\ \neg G \end{matrix}}{\neg F} \qquad\qquad (TE) \quad [F \vee \neg F]$$

$$(\forall i)^g \quad \dfrac{\begin{matrix} T \\ \triangledown \\ F\{x/y\} \end{matrix}}{\forall x\, F} \qquad \begin{matrix} \{x/y\} \text{ ammissibile in } F \\ y \text{ non libera in } T, \forall x\, F \end{matrix}$$

$$(\exists e)^g \quad \dfrac{\begin{matrix} T' \\ \triangledown \\ \exists x\, F \end{matrix} \qquad \begin{matrix} T, [F\{x/y\}] \\ \triangledown \\ G \end{matrix}}{G} \qquad \begin{matrix} \{x/y\} \text{ ammissibile in } F \\ y \text{ non libera in } T \setminus \{F\{x/y\}\}, \exists x\, F, G \end{matrix}$$

3. Correttezza e completezza della deduzione naturale predicativa

In questa sezione estendiamo al caso predicativo quanto fatto nella sezione 5.4. Dimostriamo quindi che la deduzione naturale predicativa è un sistema deduttivo corretto e enunciamo, senza dimostrarla, la sua completezza.

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

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 $T \vDash F$.

Il caso base ed i passi induttivi relativi alle undici regole proposizionali sono già stati svolti nella dimostrazione del teorema 5.17. In questa sede sviluppiamo quindi solamente i passi induttivi che riguardano le quattro regole per i quantificatori.

($\forall i$) Se la deduzione naturale è della forma

$$\dfrac{\begin{matrix} T \\ \triangledown \\ F \end{matrix}}{\forall x\, F}$$

dove x non è libera in T, abbiamo $T \rhd \forall x\, F$ e vogliamo dimostrare $T \vDash \forall x\, F$. La deduzione naturale sopra la linea orizzontale mostra che $T \rhd F$. Per ipotesi induttiva abbiamo dunque $T \vDash F$. Per dimostrare $T \vDash \forall x\, F$ consideriamo un'interpretazione I ed uno stato σ tali che $I, \sigma \vDash T$. Per mostrare $I, \sigma \vDash \forall x\, F$ fissiamo $d \in D^I$. Dato che x non è libera in T abbiamo anche $I, \sigma[x/d] \vDash T$ (stiamo utilizzando il lemma 7.12) e la nostra ipotesi ci assicura che $I, \sigma[x/d] \vDash F$. Dato che questo vale per ogni $d \in D^I$ otteniamo $I, \sigma \vDash \forall x\, F$.

($\forall e$) Se la deduzione naturale è della forma

$$\dfrac{\begin{matrix} T \\ \triangledown \\ \forall x\, F \end{matrix}}{F\{x/t\}}$$

dove la sostituzione $\{x/t\}$ è ammissibile in F, abbiamo $T \rhd F\{x/t\}$ e vogliamo dimostrare $T \vDash F\{x/t\}$. La deduzione naturale sopra la linea orizzontale mostra che $T \rhd \forall x\, F$ e per ipotesi induttiva abbiamo $T \vDash \forall x\, F$. Dato che per il lemma 7.63 si ha $\forall x\, F \vDash F\{x/t\}$, la conclusione è immediata.

($\exists i$) Se la deduzione naturale è della forma

$$\dfrac{\begin{array}{c} T \\ \triangledown \\ F\{x/t\} \end{array}}{\exists x\, F}$$

dove la sostituzione $\{x/t\}$ è ammissibile in F, abbiamo $T \rhd \exists x\, F$ e vogliamo dimostrare $T \vDash \exists x\, F$. La deduzione naturale sopra la linea orizzontale mostra che $T \rhd F\{x/t\}$ e dunque per ipotesi induttiva abbiamo $T \vDash F\{x/t\}$. La conclusione è immediata perché per il lemma 7.63 si ha $F\{x/t\} \vDash \exists x\, F$.

($\exists e$) Se la deduzione naturale è della forma

$$\dfrac{\begin{array}{c} T' \\ \triangledown \\ \exists x\, F \end{array} \qquad \begin{array}{c} T, [F] \\ \triangledown \\ G \end{array}}{G}$$

con x non libera in $T \setminus \{F\}, G$, abbiamo $T, T' \rhd G$ e vogliamo dimostrare $T, T' \vDash G$. Le deduzioni naturali sopra la linea orizzontale mostrano che $T' \rhd \exists x\, F$ e che $T, F \rhd G$. Per ipotesi induttiva abbiamo dunque sia $T' \vDash \exists x\, F$ che $T, F \vDash G$. Per dimostrare $T, T' \vDash G$ consideriamo un'interpretazione I ed uno stato σ tali che $I, \sigma \vDash T, T'$. La nostra prima ipotesi ci permette di dedurre che $I, \sigma \vDash \exists x\, F$, e quindi esiste $d_0 \in D^I$ tale che $I, \sigma[x/d_0] \vDash F$. Dato che x non è libera in $T \setminus \{F\}$ si ha anche $I, \sigma[x/d_0] \vDash T$ (stiamo nuovamente utilizzando il lemma 7.12). Sfruttando la seconda ipotesi otteniamo $I, \sigma[x/d_0] \vDash G$. Visto che x non è libera in G, una nuova applicazione del lemma 7.12 ci permette di concludere che $I, \sigma \vDash G$. □

Esercizio 12.12. I due alberi qui sotto pretendono di mostrare che

$$\exists x\, p(x), \forall x(p(x) \to q(x)) \rhd \forall x\, q(x).$$

Dato che $\exists x\, p(x), \forall x(p(x) \to q(x)) \nvDash \forall x\, q(x)$ (definite un'interpretazione che giustifichi questa affermazione) questo contraddice il teorema 12.11. Individuate il punto scorretto in ognuna delle presunte deduzioni naturali.

$$\dfrac{\exists x\, p(x) \qquad \dfrac{\dfrac{[p(x)]^1 \qquad \dfrac{\forall x(p(x) \to q(x))}{p(x) \to q(x)}}{q(x)}}{\forall x\, q(x)}}{\forall x\, q(x)}\,1$$

$$\dfrac{\dfrac{\exists x\, p(x) \qquad \dfrac{[p(x)]^1 \qquad \dfrac{\forall x(p(x) \to q(x))}{p(x) \to q(x)}}{q(x)}}{q(x)}\,1}{\forall x\, q(x)}$$

L'inverso del teorema di correttezza è il teorema di completezza, che asserisce che tutto ciò che è vero è dimostrabile per mezzo della deduzione naturale, che è quindi "completa". Come già nel caso proposizionale, ci limitiamo ad enunciare questo importante risultato.

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:

$$\begin{array}{c} \begin{array}{cc} \begin{array}{c} T \\ \triangledown \\ G \end{array} & T' \end{array} \\ \hline \triangledown \\ F \end{array}$$

Nel caso predicativo, se la deduzione $T', G \rhd F$ utilizza delle applicazioni di $(\forall i)$ e $(\exists e)$ relative a variabili che occorrono libere in T questa deduzione non è corretta. Un caso particolare di questo problema è all'origine dell'esercizio 12.8 e per risolverlo applichiamo la stessa strategia utilizzata nell'esempio 12.7, utilizzando $(\to i)$ e $(\to e)$.

Lemma 12.14. *Siano T e T' insiemi di formule predicative e F e G formule predicative. Se $T \rhd G$ e $T', G \rhd F$ allora $T, T' \rhd F$.*

Dimostrazione. E' sufficiente combinare le due deduzioni nel seguente modo:

$$\dfrac{\begin{array}{c} T \\ \triangledown \\ G \end{array} \qquad \dfrac{\begin{array}{c} T', [G]^1 \\ \triangledown \\ F \end{array}}{G \to F}\,{}^{1}}{F}$$

□

4. Esempi di deduzione naturale predicativa

In questa sezione presentiamo numerosi esempi di deduzione naturale, ottenendo tra l'altro le conseguenze logiche della sezione 8.1 (nella maggior parte dei casi sono una parte di un'equivalenza logica). Come già nella sezione 5.6, si consiglia di tentare di costruire alcune di queste deduzioni naturali senza guardare le soluzioni: anche qualora si incontrino delle difficoltà si avrà comunque una miglior comprensione della soluzione proposta.

Iniziamo dal lemma 8.5 relativo all'interazione tra negazione e quantificatori.

Esempio 12.15. Mostriamo ora che $\exists x\, \neg F \rhd \neg\forall x\, F$

$$\dfrac{\dfrac{\exists x\, \neg F \qquad \dfrac{[\neg F]^1 \qquad \dfrac{[\forall x\, F]^2}{F}}{\bot}}{\bot}\,{}^{1}}{\neg\forall x\, F}\,{}^{2}$$

In questa deduzione naturale vengono usate entrambe le regole di eliminazione dei quantificatori, e per l'applicazione di $(\exists e)$ è cruciale che x non sia libera né in $\bot$ né in $\forall x\, F$. Notiamo inoltre che sarebbe stato possibile invertire l'ordine di applicazione di $(\exists e)$ e $(\neg i)$, ottenendo una deduzione naturale lievemente diversa.

La deduzione naturale riproduce la dimostrazione della conseguenza logica corrispondente nel lemma 8.5: se $\exists x\, \neg F$ fissiamo un tale x e troviamo una contraddizione a partire da $\forall x\, F$, così da concludere che $\neg\forall x\, F$.

Esempio 12.16. Per mostrare che $\neg\forall x\, F \rhd \exists x\, \neg F$ utilizziamo la deduzione naturale

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\neg F]^1}{\exists x\, \neg F} \qquad [\neg\exists x\, \neg F]^2}{\bot}}{F}\,{}^{1}}{\forall x\, F} \qquad \neg\forall x\, F}{\bot}}{\exists x\, \neg F}\,{}^{2}$$

In questa deduzione naturale (RAA) viene utilizzata due volte, mentre c'è un'applicazione di ognuna delle due regole di introduzione dei quantificatori. Quando viene utilizzata ($\forall i$) l'ipotesi $\neg F$ è già stata scaricata e l'unica ipotesi attiva è $\neg\exists x\,\neg F$, in cui la variabile x non è libera.

La dimostrazione riprodotta da questa deduzione naturale è per assurdo: se $\neg\exists x\,\neg F$ allora $\neg F$ non sarebbe mai vera, quindi F sarebbe sempre vera e avremmo $\forall x\, F$ che contraddice la nostra ipotesi.

Esempio 12.17. Per mostrare che $\neg\exists x\, F \rhd \forall x\,\neg F$ utilizziamo la seguente deduzione naturale

$$\dfrac{\dfrac{\dfrac{\dfrac{[F]^1}{\exists x\, F} \qquad \neg\exists x\, F}{\bot}}{\neg F}{\scriptstyle 1}}{\forall x\,\neg F}$$

L'idea della dimostrazione è che, volendo ottenere $\forall x\,\neg F$, supponiamo che per qualche x valga F e raggiungiamo una contraddizione, in modo da essere certi che vale $\neg F$. Dato che x è generico, abbiamo $\forall x\,\neg F$.

Esempio 12.18. Per mostrare che $\forall x\,\neg F \rhd \neg\exists x\, F$ possiamo procedere in questo modo

$$\dfrac{\dfrac{[\exists x\, F]^2 \qquad \dfrac{[F]^1 \qquad \dfrac{\forall x\,\neg F}{\neg F}}{\bot}}{\bot}{\scriptstyle 1}}{\neg\exists x\, F}{\scriptstyle 2}$$

Esercizio 12.19. Descrivere a parole la dimostrazione formalizzata dalla deduzione naturale precedente.

Esercizio 12.20. Scrivere accanto ad ogni linea orizzontale delle deduzioni naturali dei quattro esempi precedenti la regola utilizzata.

Passiamo ora alle conseguenze logiche del lemma 8.7.

Esempio 12.21. Mostrare che $\forall x\, F \land G \rhd \forall x(F \land G)$, supponendo che x non sia libera in G, è piuttosto semplice

$$\dfrac{\dfrac{\dfrac{\dfrac{\forall x\, F \land G}{\forall x\, F}}{F} \qquad \dfrac{\forall x\, F \land G}{G}}{F \land G}}{\forall x(F \land G)}$$

Notiamo come l'ipotesi su x e G sia stata necessaria per poter utilizzare ($\forall i$): dato che x non è libera in G non lo è neppure in $\forall x\, F \land G$.

Esercizio 12.22. Mostrare che $\forall x(F \land G) \rhd \forall x\, F \land G$ (non è necessaria l'ipotesi che x non sia libera in G).

Esempio 12.23. Per dimostrare che $\forall x\, F \lor G \rhd \forall x(F \lor G)$, supponendo che x non sia libera in G, procediamo come segue

$$\dfrac{\dfrac{\forall x\, F \lor G \qquad \dfrac{\dfrac{[\forall x\, F]^1}{F}}{F \lor G} \qquad \dfrac{[G]^1}{F \lor G}}{F \lor G}{\scriptstyle 1}}{\forall x(F \lor G)}$$

Anche in questo caso l'ipotesi che x non sia libera in G, e quindi in $\forall x\, F \vee G$, viene utilizzata per poter applicare $(\forall i)$.

Esercizio 12.24. Scrivete una deduzione naturale diversa dalla precedente che mostri $\forall x\, F \vee G \rhd \forall x (F \vee G)$. Per esempio potete scambiare l'ordine di utilizzo di $(\vee e)$ e $(\forall i)$.

Esempio 12.25. Per mostrare che $\forall x (F \vee G) \rhd \forall x\, F \vee G$, dove x non è libera in G, utilizziamo (TE): se vale G la conclusione è immediata, altrimenti dobbiamo ricorrere all'ipotesi e ottenere $F \vee G$: a partire da entrambi i disgiunti otteniamo F (nel secondo caso usiamo (*ex-falso*)).

$$
\dfrac{[G \vee \neg G] \qquad \dfrac{[G]^2}{\forall x\, F \vee G} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall x (F \vee G)}{F \vee G} \qquad [F]^1 \qquad \dfrac{\dfrac{[G]^1 \qquad [\neg G]^2}{\bot}}{F}}{F}{\scriptstyle 1}}{\forall x\, F}}{\forall x\, F \vee G}}{\forall x\, F \vee G}{\scriptstyle 2}
$$

Per poter applicare $(\forall i)$ è necessario che x non sia libera in $\neg G$, che è equivalente all'ipotesi che x non sia libera in G.

Esempio 12.26. Mostriamo ora, supponendo che x non sia libera in G, che vale $\exists x\, F \wedge G \rhd \exists x (F \wedge G)$:

$$
\dfrac{\dfrac{\exists x\, F \wedge G}{\exists x\, F} \qquad \dfrac{\dfrac{[F]^1 \qquad \dfrac{\exists x\, F \wedge G}{G}}{F \wedge G}}{\exists x (F \wedge G)}}{\exists x (F \wedge G)}{\scriptstyle 1}
$$

In questo caso l'ipotesi che x non sia libera in G, e quindi in $\exists x\, F \wedge G$, è necessaria per utilizzare $(\exists e)$.

Esercizio 12.27. Spiegare perché nella deduzione naturale precedente non è possibile scambiare l'ordine di utilizzo di $(\exists i)$ e $(\exists e)$.

Esempio 12.28. La seguente deduzione naturale mostra che $\exists x (F \wedge G) \rhd \exists x\, F \wedge G$, supponendo sempre che x non sia libera in G:

$$
\dfrac{\exists x (F \wedge G) \qquad \dfrac{\dfrac{\dfrac{[F \wedge G]^1}{F}}{\exists x\, F} \qquad \dfrac{[F \wedge G]^1}{G}}{\exists x\, F \wedge G}}{\exists x\, F \wedge G}{\scriptstyle 1}
$$

In questo caso l'utilizzo di $(\exists e)$ richiede che x non sia libera in $\exists x\, F \wedge G$, e questo segue dall'ipotesi su x e G.

Esercizio 12.29. Scrivere un'altra deduzione naturale che mostri $\exists x (F \wedge G) \rhd \exists x\, F \wedge G$ e che utilizzi $(\wedge i)$ come ultima regola.

Esempio 12.30. Per dimostrare che $\exists x\, F \vee G \rhd \exists x (F \vee G)$ utilizziamo prima $(\vee e)$ e poi $(\exists e)$.

$$
\dfrac{\exists x\, F \vee G \qquad \dfrac{[\exists x\, F]^2 \qquad \dfrac{\dfrac{[F]^1}{F \vee G}}{\exists x (F \vee G)}}{\exists x (F \vee G)}{\scriptstyle 1} \qquad \dfrac{\dfrac{[G]^2}{F \vee G}}{\exists x (F \vee G)}}{\exists x (F \vee G)}{\scriptstyle 2}
$$

Notiamo che questa deduzione naturale non richiede che x non sia libera in G. Anche nella dimostrazione della conseguenza logica corrispondente nel lemma 8.7 non avevamo sfruttato questa ipotesi.

Esempio 12.31. L'ipotesi che x non sia libera in G, e quindi in $\exists x\, F \lor G$, è invece necessaria per poter applicare $(\exists e)$ nella seguente deduzione naturale che mostra $\exists x(F \lor G) \rhd \exists x\, F \lor G$. Rispetto all'esempio precedente l'ordine di utilizzo di $(\exists e)$ e $(\lor e)$ è invertito.

$$\dfrac{\exists x(F \lor G) \qquad \dfrac{[F \lor G]^2 \qquad \dfrac{\dfrac{[F]^1}{\exists x\, F}}{\exists x\, F \lor G} \qquad \dfrac{[G]^1}{\exists x\, F \lor G}}{\exists x\, F \lor G}\,{}^1}{\exists x\, F \lor G}\,{}^2$$

Consideriamo ora le conseguenze logiche contenute nel lemma 8.9, svolgendo in dettaglio quelle più delicate e lasciando come utile esercizio per il lettore le altre.

Esempio 12.32. Per dimostrare che $\forall x\, F \to G \rhd \exists x(F \to G)$ la deduzione naturale inizia distinguendo il caso in cui vale $\forall x\, F$ da quello in cui ciò non accade e quindi usando la regola (TE). Il primo caso utilizza $(\to i)$ senza scaricamento, mentre nel secondo utilizziamo $\neg\forall x\, F \rhd \exists x\, \neg F$ (esempio 12.16) per poi usare $(\exists e)$ e dedurre $F \to G$ dall'ipotesi ausiliaria $\neg F$ (esempio 5.29).

$$\dfrac{[\forall x\, F \lor \neg\forall x\, F] \qquad \dfrac{\dfrac{\dfrac{[\forall x\, F]^2 \qquad \forall x\, F \to G}{G}}{F \to G}}{\exists x(F \to G)} \qquad \dfrac{\begin{matrix}[\neg\forall x\, F]^2 \\ \triangledown \\ \exists x\, \neg F\end{matrix} \qquad \dfrac{\begin{matrix}[\neg F]^1 \\ \triangledown \\ F \to G\end{matrix}}{\exists x(F \to G)}}{\exists x(F \to G)}\,{}^1}{\exists x(F \to G)}\,{}^2$$

In questo caso non è necessario supporre che x non sia libera in G.

Esempio 12.33. Mostriamo che $\exists x(F \to G) \rhd \forall x\, F \to G$, quando x non è libera in G, e quindi in $\forall x\, F \to G$ (questa ipotesi è necessaria per l'applicazione di $(\exists e)$):

$$\dfrac{\exists x(F \to G) \qquad \dfrac{\dfrac{\dfrac{[\forall x\, F]^1}{F} \qquad [F \to G]^2}{G}}{\forall x\, F \to G}\,{}^1}{\forall x\, F \to G}\,{}^2$$

Esercizio 12.34. Scrivete una deduzione naturale diversa dalla precedente che mostri $\exists x(F \to G) \rhd \forall x\, F \to G$. Per esempio potete scambiare l'ordine di utilizzo di $(\to i)$ e $(\exists e)$.

Esercizio 12.35. Mostrare che se x non è libera in G allora:

$$G \to \forall x\, F \rhd \forall x(G \to F),$$
$$\forall x(G \to F) \rhd G \to \forall x\, F,$$
$$\exists x\, F \to G \rhd \forall x(F \to G).$$

Esempio 12.36. Per ottenere $\forall x(F \to G) \rhd \exists x\, F \to G$ quando x non è libera in G, partiamo dall'ipotesi che $\exists x\, F$ con lo scopo di dedurre G:

$$\dfrac{\dfrac{[\exists x\, F]^2 \qquad \dfrac{[F]^1 \qquad \dfrac{\forall x(F \to G)}{F \to G}}{G}}{G}\,{}^1}{\exists x\, F \to G}\,{}^2$$

Esempio 12.37. Per ottenere $G \to \exists x\, F \rhd \exists x(G \to F)$ utilizziamo (TE) distinguendo a seconda se G vale o meno. Nel primo caso si usa l'ipotesi e, nel corso della derivazione che porta a $(\exists e)$, si utilizza $({\to}i)$ senza scaricamento. Nel secondo caso si usa l'esempio 5.29. Notiamo che questa deduzione naturale non richiede l'ipotesi che x non sia libera in G.

$$\dfrac{[G \lor \neg G] \qquad \dfrac{\dfrac{[G]^2 \qquad G \to \exists x\, F}{\exists x\, F} \qquad \dfrac{\dfrac{[F]^1}{G \to F}}{\exists x(G \to F)}}{\exists x(G \to F)}\,{\scriptstyle 1} \qquad \dfrac{\begin{array}{c}[\neg G]^2\\ \triangledown\\ G \to F\end{array}}{\exists x(G \to F)}}{\exists x(G \to F)}\,{\scriptstyle 2}$$

Esercizio 12.38. Dimostrare $G \to \exists x\, F \rhd \exists x(G \to F)$ applicando (TE) alla formula $\exists x\, F$.

Esercizio 12.39. Dimostrare che $\exists x(G \to F) \rhd G \to \exists x\, F$ quando x non è libera in G.

Consideriamo ora i risultati contenuti nel lemma 8.14.

Esercizio 12.40. Mostrare $\forall x\, F \land \forall x\, G \rhd \forall x(F \land G)$ e $\forall x(F \land G) \rhd \forall x\, F \land \forall x\, G$.

Esercizio 12.41. Mostrare $\exists x\, F \lor \exists x\, G \rhd \exists x(F \lor G)$ e $\exists x(F \lor G) \rhd \exists x\, F \lor \exists x\, G$.

Esempio 12.42. Per mostrare $\forall x\, F \to \exists x\, G \rhd \exists x(F \to G)$ usiamo (TE) scaricando $\exists x\, G \lor \neg \exists x\, G$. Ciò significa che basta mostrare $\exists x\, G \rhd \exists x(F \to G)$ e $\forall x\, F \to \exists x\, G, \neg \exists x\, G \rhd \exists x(F \to G)$. Lasciamo la prima di queste deduzioni naturali per esercizio, e ci concentriamo sulla seconda. Utilizziamo (MT) e il risultato dell'esempio 12.16 per dedurre dalle ipotesi $\exists x\, \neg F$; dall'ipotesi ausiliaria $\neg F$ deduciamo $F \to G$ utilizzando l'esempio 5.29 e quindi $\exists x(F \to G)$.

$$\dfrac{\begin{array}{c}\dfrac{\neg \exists x\, G \qquad \forall x\, F \to \exists x\, G}{\neg \forall x\, F}\\ \triangledown\\ \exists x\, \neg F\end{array} \qquad \dfrac{\begin{array}{c}[\neg F]^1\\ \triangledown\\ F \to G\end{array}}{\exists x(F \to G)}}{\exists x(F \to G)}\,{\scriptstyle 1}$$

Un altro approccio consiste nell'applicare (TE) per scaricare $\forall x\, F \lor \neg \forall x\, F$. In questo caso basta mostrare $\forall x\, F \to \exists x\, G, \forall x\, F \rhd \exists x(F \to G)$ e $\neg \forall x\, F \rhd \exists x(F \to G)$. La prima di queste deduzioni è

$$\dfrac{\dfrac{\forall x\, F \qquad \forall x\, F \to \exists x\, G}{\exists x\, G} \qquad \dfrac{\dfrac{[G]^1}{F \to G}}{\exists x(F \to G)}}{\exists x(F \to G)}\,{\scriptstyle 1}$$

Nella seconda utilizziamo nuovamente il risultato dell'esempio 12.16:

$$\dfrac{\begin{array}{c}\neg \forall x\, F\\ \triangledown\\ \exists x\, \neg F\end{array} \qquad \dfrac{\dfrac{\dfrac{\dfrac{[F]^1 \qquad [\neg F]^2}{\bot}}{G}}{F \to G}\,{\scriptstyle 1}}{\exists x(F \to G)}}{\exists x(F \to G)}\,{\scriptstyle 2}$$

Esempio 12.43. Mostriamo che $\exists x(F \to G) \rhd \forall x\, F \to \exists x\, G$:

$$\dfrac{\dfrac{\exists x(F \to G) \qquad \dfrac{\dfrac{\dfrac{[\forall x\, F]^2}{F} \qquad [F \to G]^1}{G}}{\exists x\, G}}{\exists x\, G}\;{}^{1}}{\forall x\, F \to \exists x\, G}\;{}^{2}$$

Si notino le similitudini tra questa deduzione naturale e quella dell'esempio 12.33.

Consideriamo le conseguenze logiche dell'esercizio 8.17.

Esercizio 12.44. Dimostrate che $\exists x(F \wedge G) \rhd \exists x\, F \wedge \exists x\, G$ e che $\forall x\, F \vee \forall x\, G \rhd \forall x(F \vee G)$.

Esempio 12.45. Mostriamo che $\exists x\, F \to \forall x\, G \rhd \forall x(F \to G)$ attraverso al seguente deduzione naturale:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[F]^1}{\exists x\, F} \qquad \exists x\, F \to \forall x\, G}{\forall x\, G}}{G}}{F \to G}\;{}^{1}}{\forall x(F \to G)}$$

Notiamo come in questa deduzione naturale abbiamo ottenuto ad un certo punto $\exists x\, F$. La formula esistenziale è stata sfruttata attraverso $(\to e)$, senza quindi ricorrere a $(\exists e)$.

Esercizio 12.46. Mostrare che $\exists x\, \forall y\, F \rhd \forall y\, \exists x\, F$.

Esempio 12.47. Mostriamo ora che $\rhd \exists x(p(x) \to \forall y\, p(y))$, utilizzando nella nostra deduzione naturale quella dell'esempio 12.16. Si noti anche l'applicazione di $(\exists e)^g$ contrassegnata dal numero 2.

$$\dfrac{[\forall y\, p(y) \vee \neg\forall y\, p(y)] \qquad \dfrac{\dfrac{[\forall y\, p(y)]^3}{p(x) \to \forall y\, p(y)}}{\exists x(p(x) \to \forall y\, p(y))} \qquad \dfrac{\begin{array}{c}[\neg\forall y\, p(y)]^3 \\ \triangledown \\ \exists y\, \neg p(y)\end{array} \qquad \dfrac{\dfrac{\dfrac{\dfrac{[p(x)]^1 \qquad [\neg p(x)]^2}{\bot}}{\forall y\, p(y)}}{p(x) \to \forall y\, p(y)}\;{}^{1}}{\exists x(p(x) \to \forall y\, p(y))}}{\exists x(p(x) \to \forall y\, p(y))}\;{}^{2}}{\exists x(p(x) \to \forall y\, p(y))}\;{}^{3}$$

Il risultato che abbiamo ottenuto è a prima vista contrario all'intuizione. Sia F l'enunciato che abbiamo dedotto senza ipotesi e interpretiamo $p(x)$ come "x è uno studente che supera lo scritto". Allora F asserisce che esiste uno studente tale che, se lui supera lo scritto, allora tutti superano lo scritto. Per il teorema di correttezza 12.11 da $\rhd F$ segue la validità di F, e quindi abbiamo dimostrato che uno studente con questa proprietà esiste. In effetti ciò è vero, ma prima dello scritto è impossibile individuare un tale studente (in modo che lui solo debba studiare!). Infatti se tutti gli studenti superano lo scritto, allora qualunque studente soddisfa l'implicazione, mentre se qualche studente viene bocciato, allora come x scegliamo uno studente che non ha superato lo scritto. Stiamo usando in maniera essenziale la logica classica (si veda la discussione della sezione 5.5), come evidenziato dall'uso di (TE) e di (RAA) (che era stata utilizzata nell'esempio 12.16) nella deduzione naturale.[1]

[1] Questo enunciato è noto nella letteratura inglese come il drinker paradox: per ulteriori dettagli si veda https://en.wikipedia.org/wiki/Drinker_paradox.

Esercizio 12.48. Dimostrare che $\exists x\, r(x) \vee \exists x\, \neg s(f(x)) \rhd \exists x(s(x) \to r(x))$.

Risolviamo ora con la deduzione naturale gli esempi 11.31 e 11.32. che richiedono di dimostrare la validità di enunciati della forma $F \wedge G \to H$. Per il lemma 2.40 la loro validità è equivalente a $F \wedge G \vDash H$ e quindi (per il lemma 2.30) a $F, G \vDash H$. Le deduzioni naturali che otterremo mostreranno (per il teorema di correttezza) precisamente queste conseguenze logiche.

Esempio 12.49. Mostriamo $\exists x\, r(a,x), \forall x(\exists y\, r(y,x) \to \neg p(x)) \rhd \exists x\, \neg p(x)$ attraverso la seguente deduzione naturale:

$$\dfrac{\exists x\, r(a,x) \qquad \dfrac{\dfrac{\dfrac{[r(a,x)]^1}{\exists y\, r(y,x)} \qquad \dfrac{\forall x(\exists y\, r(y,x) \to \neg p(x))}{\exists y\, r(y,x) \to \neg p(x)}}{\neg p(x)}}{\exists x\, \neg p(x)}}{\exists x\, \neg p(x)}\ {\scriptstyle 1}$$

Esempio 12.50. Mostriamo $\exists x\, \forall y\, r(y,x), \forall x\, \forall y(r(y,x) \to r(x,y)) \rhd \exists x\, r(x,a)$:

$$\dfrac{\exists x\, \forall y\, r(y,x) \qquad \dfrac{\dfrac{\dfrac{[\forall y\, r(y,x)]^1}{r(a,x)} \qquad \dfrac{\dfrac{\forall x\, \forall y(r(y,x) \to r(x,y))}{\forall y(r(y,x) \to r(x,y))}}{r(a,x) \to r(x,a)}}{r(x,a)}}{\exists x\, r(x,a)}}{\exists x\, r(x,a)}\ {\scriptstyle 1}$$

Esempio 12.51. Mostriamo $\forall x\, \exists y\, r(x,f(y)), \forall z\, \forall w(r(z,w) \to p(z)) \rhd \forall y\, p(y)$:

$$\dfrac{\dfrac{\dfrac{\forall x\, \exists y\, r(x,f(y))}{\exists y\, r(x,f(y))} \qquad \dfrac{[r(x,f(y))]^1 \qquad \dfrac{\dfrac{\forall z\, \forall w(r(z,w) \to p(z))}{\forall w(r(x,w) \to p(x))}}{r(x,f(y)) \to p(x)}}{p(x)}}{p(x)}\ {\scriptstyle 1}}{\forall y\, p(y)}$$

Si noti che l'applicazione di $(\exists e)$ riguarda la variabile y, e quindi il fatto che x sia libera in $p(x)$ non viola le condizioni. Inoltre nell'ultimo passaggio abbiamo utilizzato la regola derivata $(\forall i)^g$.

Esempio 12.52. Mostriamo $\exists x\, r(x,f(x)), \forall x(\exists y\, r(y,x) \to \neg p(x)) \rhd \exists z\, \neg p(f(z))$:

$$\dfrac{\exists x\, r(x,f(x)) \qquad \dfrac{\dfrac{\dfrac{[r(x,f(x))]^1}{\exists y\, r(y,f(x))} \qquad \dfrac{\forall x(\exists y\, r(y,x) \to \neg p(x))}{\exists y\, r(y,f(x)) \to \neg p(f(x))}}{\neg p(f(x))}}{\exists z\, \neg p(f(z))}}{\exists z\, \neg p(f(z))}\ {\scriptstyle 1}$$

In questo esempio è interessante la prima applicazione di $(\exists i)$ (quella che conduce a $\exists y\, r(y,f(x))$): qui F è $r(y,f(x))$ e la sostituzione è $\{y/x\}$. Da $r(x,f(x))$ sarebbe stato possibile dedurre anche $\exists y\, r(y,f(y))$ (se F è $r(y,f(y))$), $\exists y\, r(x,f(y))$ (se F è $r(x,f(y))$) oppure $\exists y\, r(x,y)$ (se F è $r(x,y)$ e la sostituzione è $\{y/f(x)\}$), ma nessuna di queste formule avrebbe consentito la prosecuzione della deduzione.

Esempio 12.53. Mostriamo che $\forall x\, r(c,x), \forall x(p(x) \to \neg r(x,f(x))) \rhd \exists x\, \neg p(x)$:

$$\cfrac{\cfrac{\cfrac{\cfrac{\forall x\, r(c,x)}{r(c,f(c))} \qquad \cfrac{[p(c)]^1 \qquad \cfrac{\forall x(p(x)\to\neg r(x,f(x)))}{p(c)\to\neg r(c,f(c))}}{\neg r(c,f(c))}}{\bot}}{\neg p(c)}\,{}^{1}}{\exists x\,\neg p(x)}$$

Costruire questa deduzione richiede individuare quale termine t non può soddisfare p, in modo da far partire la deduzione da $p(t)$. In questo caso t è c.

5. Deduzione naturale per la logica con uguaglianza

Nel caso della logica con uguaglianza possiamo procedere analogamente a quanto fatto per il metodo dei tableaux nella sezione 11.8. Combinando il teorema 10.34 con i teoremi di correttezza e completezza si ottiene che $T \models_{=} F$ è equivalente a $T, \mathsf{Eq}_{\mathcal{L}} \rhd F$.

Si può però rispettare maggiormente lo spirito della deduzione naturale e trattare il simbolo $=$ in modo analogo ai simboli logici: aggiungiamo regole per introdurre e eliminare $=$. La regola $(=i)$ consiste semplicemente nel considerare come scaricate tutte le formule del tipo $x = x$, dove x è una variabile. Per eliminare $=$ invece osserviamo che se sappiamo che $s = t$ (dove s e t sono termini), da una formula del tipo $F\{x/s\}$ possiamo dedurre $F\{x/t\}$. Per far ciò è naturalmente necessario che le due sostituzioni siano ammissibili in F.

Otteniamo quindi le due regole

$$(=i) \qquad [x = x] \qquad\qquad\qquad (=e) \quad \cfrac{\begin{array}{c} T \\ \triangledown \\ \| \\ s = t \end{array} \qquad \begin{array}{c} T' \\ \triangledown \\ \| \\ F\{x/s\} \end{array}}{F\{x/t\}} \quad \begin{array}{l} \{x/s\} \text{ e } \{x/t\} \\ \text{ammissibili in } F \end{array}$$

Come nel caso di (TE), quando utilizziamo $(=i)$ lo scaricamento di $x = x$ non avviene in corrispondenza di una regola successiva e quindi non indichiamo nessun numero ad esponente.

Se in una deduzione naturale utilizziamo le regole per l'uguaglianza che abbiamo appena introdotto scriviamo $\rhd_{=}$ al posto di $\rhd$.

Esempio 12.54. Iniziamo col mostrare che $\rhd_{=} t = t$ per qualunque termine t:

$$\cfrac{\cfrac{[x = x]}{\forall x\, x = x}}{t = t}$$

L'esempio precedente mostra che $t = t$ può essere considerata come un'ipotesi scaricata per qualunque termine t, e quindi possiamo considerare la seguente regola derivata, che generalizza l'introduzione dell'uguaglianza:

$$(=i)^g \qquad [t = t]$$

Il prossimo esempio è una rivisitazione dell'esempio 11.55.

Esempio 12.55. Mostriamo che $p(a), \neg p(b) \rhd_{=} a \neq b$:

$$\cfrac{\cfrac{\cfrac{[a = b]^1 \qquad p(a)}{p(b)} \qquad \neg p(b)}{\bot}}{a \neq b}\,{}^{1}$$

Abbiamo ipotizzato che $a = b$ con l'intento di raggiungere una contraddizione. Abbiamo poi applicato $(=e)$ alla formula $p(x)$, sostituendo alla variabile x i termini a e b.

Verifichiamo che le regole appena introdotte ci permettono di dedurre tutte gli assiomi dell'uguaglianza (si ricordi la definizione 6.64).

LEMMA 12.56. *Se G è un enunciato che appartiene a $\mathsf{Eq}_{\mathcal{L}}$ allora $\rhd_= G$.*

DIMOSTRAZIONE. Notiamo che $\rhd_= (e1)$ è già stato ottenuto in un passaggio intermedio della deduzione utilizzata nell'esempio 12.54, dove da $(=i)$, applicando $(\forall i)$, si è ottenuto $\forall x\, x = x$.

Per mostrare $\rhd_= (e2)$ iniziamo ad ottenere $x = y \rhd_= y = x$, dove x e y sono variabili. A questo scopo scegliamo una variabile v diversa da x. Applichiamo $(=e)$ quando F è la formula $v = x$, così che $F\{v/x\}$ sia $x = x$ (che può essere scaricata per $(=i)$) e $F\{v/y\}$ sia $y = x$. Otteniamo

$$\dfrac{x = y \qquad [x = x]}{y = x}$$

A partire da questa deduzione naturale possiamo ottenere $\rhd_= (e2)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{[x = y]^1 \qquad [x = x]}{y = x}}{x = y \to y = x}\,1}{\forall y(x = y \to y = x)}}{\forall x\, \forall y(x = y \to y = x)}$$

Per mostrare che $\rhd_= (e3)$ partiamo similmente da $x = y, y = z \rhd_= x = z$. Scegliamo una variabile v diversa da x. Applichiamo $(=e)$ quando F è la formula $x = v$, così che $F\{v/y\}$ sia $x = y$ e $F\{v/z\}$ sia $x = z$. Otteniamo

$$\dfrac{y = z \qquad x = y}{x = z}$$

Ora otteniamo la deduzione naturale cercata nel seguente modo:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[x = y \land y = z]^1}{y = z} \qquad \dfrac{[x = y \land y = z]^1}{x = y}}{x = z}}{x = y \land y = z \to x = z}\,1}{\forall z(x = y \land y = z \to x = z)}}{\forall y\, \forall z(x = y \land y = z \to x = z)}}{\forall x\, \forall y\, \forall z(x = y \land y = z \to x = z)}$$

Per ottenere $\rhd_= (e4)$ è facile generalizzare al caso n-ario quanto sviluppiamo in dettaglio quando f è un simbolo di funzione unario. In questo caso è sufficiente mostrare che $x = y \rhd_= f(x) = f(y)$, per poi procedere come nei casi precedenti con $(\to i)$ e $(\forall i)$. Scegliamo una variabile v diversa da x e applichiamo $(=e)$ alla formula $f(x) = f(v)$, così che $F\{v/x\}$ sia $f(x) = f(x)$ (che può essere scaricata per $(=i)^g$) e $F\{v/y\}$ sia $f(x) = f(y)$. Otteniamo

$$\dfrac{x = y \qquad [f(x) = f(x)]}{f(x) = f(y)}$$

Per ottenere $\rhd_= (e5)$ per un simbolo di relazione n-ario è sufficiente applicare n volte $(=e)$ e poi utilizzare $(\to i)$ e $(\forall i)$ come negli altri casi. $\square$

ESERCIZIO 12.57. Mostrare, utilizzando l'esempio 12.54 e la dimostrazione precedente, che $s = t \rhd_= t = s$ e $s = t, t = r \rhd_= s = r$, dove s, t e r sono termini qualsiasi.

Dimostriamo ora che il nostro sistema di deduzione naturale per la logica con uguaglianza è corretto per la conseguenza logica nella logica con uguaglianza introdotta nella definizione 7.69.

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 naturale sia della forma, permessa da $(=i)$,

$$[x = x]$$

In questo caso, dato che tutte le ipotesi sono scaricate, dobbiamo mostrare che $x = x$ è valida nella logica con uguaglianza. Questo è ovvio perché in qualunque interpretazione normale I e stato σ, si ha $(\sigma(x), \sigma(x)) \in =^I$.

Il restante caso base ed i passi induttivi relativi alle regole relative ai connettivi e ai quantificatori sono identici a quelli svolti nelle dimostrazioni dei teoremi 5.17 e 12.11.

Resta quindi da considerare solo il passo induttivo che riguarda la regola $(=e)$. Se la deduzione naturale è della forma

$$\dfrac{\begin{array}{c} T \\ \nabla_{=} \\ s = t \end{array} \qquad \begin{array}{c} T' \\ \nabla_{=} \\ F\{x/s\} \end{array}}{F\{x/t\}}$$

dove $\{x/s\}$ e $\{x/t\}$ sono sostituzioni ammissibili in F, abbiamo $T, T' \rhd_= F\{x/t\}$ e dobbiamo mostrare che $T, T' \models_= F\{x/t\}$. Le deduzioni naturali sopra la linea orizzontale mostrano che $T \rhd_= s = t$ e che $T' \rhd_= F\{x/s\}$. Per ipotesi induttiva abbiamo dunque sia $T \models_= s = t$ che $T' \models_= F\{x/s\}$. Fissati un'interpretazione normale I ed uno stato σ tali che $I, \sigma \models T, T'$ abbiamo dunque $I, \sigma \models s = t$ e $I, \sigma \models F\{x/s\}$. Dato che I è normale si ha che $\sigma(s)$ e $\sigma(t)$ sono lo stesso elemento di D^I. Per il lemma di sostituzione 7.59 da $I, \sigma \models F\{x/s\}$ otteniamo $I, \sigma[x/\sigma(s)] \models F$, che è lo stesso che $I, \sigma[x/\sigma(t)] \models F$ e conduce (sempre per il lemma di sostituzione) a $I, \sigma \models F\{x/t\}$. $\square$

Sfruttando la completezza della deduzione naturale e i risultati della sezione 10.3 è facile mostrare che la deduzione naturale per la logica con uguaglianza è completa per la conseguenza logica nella logica con uguaglianza.

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 testimonia $T \rhd_= F$: tutte le volte che nella prima deduzione naturale abbiamo una foglia etichettata con $G \in \mathsf{Eq}_{\mathcal{L}}$ possiamo attaccare sopra di essa una deduzione naturale che mostra $\rhd_= G$. $\square$

Nota 12.60. Combinando il teorema 12.58 con la dimostrazione del teorema 12.59 si ottiene il seguente risultato: se $T \rhd_= F$ allora esiste una deduzione naturale di F con ipotesi in T tale che le regole $(=i)$ e $(=e)$ sono utilizzate solamente in sottodeduzioni senza ipotesi che hanno come conclusione assiomi dell'uguaglianza.

Concludiamo con alcuni esempi di deduzione naturale per la logica con uguaglianza.

Esempio 12.61. Mostriamo che

$$\forall y\, r(f(y), f(y)), \neg\exists x\, r(x, f(x)) \rhd_{=} \forall x\, f(x) \neq x :$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[f(x)=x]^1 \quad \dfrac{\forall y\, r(f(y),f(y))}{r(f(x),f(x))}}{r(x,f(x))}}{\exists x\, r(x,f(x))} \qquad \neg\exists x\, r(x,f(x))}{\bot}}{f(x)\neq x}{}^{1}}{\forall x\, f(x)\neq x}$$

In questa deduzione naturale abbiamo applicato $(=e)$ alla formula $r(z, f(x))$, sostituendo alla variabile z i termini $f(x)$ e x.

Per dimostrare $\forall x\, f(x) \neq x$ abbiamo fissato x e supposto che $f(x) = x$ alla ricerca di una contraddizione. Alla fine della deduzione, quando l'ipotesi ausiliaria era stata scaricata, abbiamo usato $(\forall i)$.

Il prossimo esempio riprende l'esercizio 11.56.

Esempio 12.62. Mostriamo che $\forall x\, r(a, x), \exists x\, \exists y\, \neg r(y, x) \rhd_{=} \exists x\, a \neq x$:

$$\dfrac{\exists x\, \exists y\, \neg r(y,x) \qquad \dfrac{[\exists y\, \neg r(y,x)]^3 \qquad \dfrac{\dfrac{\dfrac{\dfrac{[a=y]^1 \quad \dfrac{\forall x\, r(a,x)}{r(a,x)}}{r(y,x)} \qquad [\neg r(y,x)]^2}{\bot}}{a\neq y}{}^{1}}{\exists x\, a\neq x}}{\exists x\, a\neq x}{}^{2}}{\exists x\, a\neq x}{}^{3}$$

In questa deduzione naturale abbiamo applicato $(=e)$ alla formula $r(z, x)$, sostituendo alla variabile z i termini a e y.

Si noti inoltre l'utilizzo di due $(\exists e)$ consecutivi: la prima applicazione della regola produce una formula esistenziale, a cui la regola viene applicata una seconda volta.

Esempio 12.63. Dimostriamo che $\forall x\, \neg r(x, x), \forall x\, \exists y\, r(x, f(y)) \rhd_{=} \forall x\, \exists y\, x \neq f(y)$.

$$\dfrac{\dfrac{\dfrac{\forall x\, \exists y\, r(x,f(y))}{\exists y\, r(x,f(y))} \qquad \dfrac{\dfrac{\dfrac{\dfrac{[x=f(y)]^1 \quad [r(x,f(y))]^2}{r(f(y),f(y))} \qquad \dfrac{\forall x\, \neg r(x,x)}{\neg r(f(y),f(y))}}{\bot}}{x\neq f(y)}{}^{1}}{\exists y\, x\neq f(y)}}{\exists y\, x\neq f(y)}{}^{2}}{\forall x\, \exists y\, x\neq f(y)}$$

Lasciamo al lettore la ricostruzione del ragionamento rappresentato da questa deduzione naturale.

Esercizio 12.64. Mostrare che

$$\forall x\, r(x, f(x)), \forall x\, \forall y (r(x, y) \to \neg r(y, x)) \rhd_{=} \forall x\, x \neq f(f(x)).$$