Capitolo 4

Logica dei predicati del primo ordine

4.1 Limitazioni della logica proposizionale

Introduciamo la *logica dei predicati del primo ordine*, o semplicemente *logica del primo ordine*, facendo alcune considerazioni sulle limitazioni espressive della logica proposizionale.

Cercheremo di rendere in logica proposizionale il noto sillogismo espresso dalle frasi "Tutti gli uomini sono mortali; Socrate è un uomo; pertanto Socrate è mortale". Secondo quanto visto nel capitolo precedente, possiamo associare ad ogni lettera proposizionale il ruolo di rappresentare la verità o la falsità di un concetto. Nella realtà che stiamo considerando, scegliamo di associare le lettere us e ms alla veridicità dei concetti "Socrate è un uomo" e "Socrate è mortale", rispettivamente. Poiché la prima premessa del sillogismo esprime una relazione di conseguenza fra il fatto che un generico individuo sia un uomo ed il fatto che lo stesso individuo sia mortale, e la seconda premessa esprime un'affermazione categorica, potremmo pensare di rappresentare la realtà di interesse con l'insieme di formule $S = \{us{\rightarrow}ms, us\}$, costituito da un'implicazione (cfr. esempio 3.3.7) e da una formula atomica. In tutti i modelli di S la lettera ms è assegnata a T, o in altre parole $S \models ms$. Ciò vuol dire che l'insieme di formule scelto è in qualche misura ragionevole, in quanto la conclusione del sillogismo è sempre verificata nei suoi modelli.

La limitazione di questo schema sta nel fatto che in S non c'è una vera e propria rappresentazione della prima premessa del sillogismo ("Tutti gli uomini sono mortali"), ma solo una sua istanziazione ad uno specifico individuo ("Se Socrate è un uomo, allora è mortale"). Ciò può essere visto considerando una realtà di interesse in cui esista anche l'individuo Platone, che è un uomo. Chiaramente possiamo considerare un ulteriore sillogismo, simile al precedente, che faccia riferimento all'individuo Platone invece che a Socrate. Secondo il metodo di cui sopra, per rendere in logica proposizionale anche questo secondo sillogismo, dobbiamo fare riferimento a due ulteriori lettere up e mp (associate ai concetti "Platone è un uomo" e "Platone è mortale") e considerare l'insieme di formule $SP = \{us{\rightarrow}ms, us, up{\rightarrow}mp, up\}$, che implica logicamente $ms{\wedge}mp$.

Aggiungendo ulteriori individui, dovremmo aggiungere altrettante formule. In altre parole, nella logica proposizionale non è possibile esprimere asserzioni sulle proprietà di insiemi di oggetti. Ciò è invece possibile nella logica dei predicati del prim'ordine, che analizziamo in questo paragrafo. Il termine "del primo ordine" si riferisce al fatto che in questa logica non si può quantificare sui predicati, come vedremo in seguito.

4.2 Sintassi

Iniziamo con la definizione del linguaggio della logica del primo ordine.

Definizione 4.2.1 (Alfabeto della logica del prim'ordine) *L'alfabeto della logica del primo ordine è dato da:*

- *un insieme* **V** *di* variabili*;*
- *un insieme* **SF** *di* simboli di funzione*, ognuno dei quali ha associato il suo numero di argomenti detto* arità*;*
- *un insieme* **SP** *di* simboli di predicato*, ognuno dei quali ha associato il suo numero di argomenti detto* arità*; assumeremo che* **SP** *contenga il predicato di arità 2 "=" (chiamato "uguaglianza");*

- *i* connettivi logici $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\equiv$*;*
- *i* quantificatori $\forall$ *ed* $\exists$*, denominati rispettivamente quantificatore* universale *e quantificatore* esistenziale*;*
- *i simboli speciali "(", ") " e ", " (virgola).*

Poiché il simbolo di predicato "=" è "obbligatorio", per semplicità di lettura spesso lo ometteremo nella lista dei simboli di interesse. Se f è un simbolo di funzione o di predicato di arità k, scriveremo f/k per riferirci ad esso. I simboli di funzione di arità 0 vengono anche detti *simboli di costante*. Rispetto alla logica proposizionale, gli oggetti più simili alle lettere proposizionali sono i simboli di predicato di arità 0, che verranno appunto denominati lettere proposizionali anche in questo contesto.

Utilizzeremo le seguenti convenzioni relative ai simboli di variabile, di costante, di funzione e di predicato.

- Simboli di variabile saranno sequenze di caratteri alfabetici aventi la lettera iniziale maiuscola ($X, Y, Z, \mathit{MiaVariabile}, \ldots$).
- Tutti gli altri simboli saranno sequenze di caratteri alfabetici aventi la lettera iniziale minuscola (*prodotto*, *divisibile*, *a*, *clku*, ...).

Queste convenzioni saranno sufficienti per identificare, in base alla struttura sintattica, la categoria di appartenenza di ogni simbolo occorrente in una formula.

Esempio 4.2.2 Nel seguito utilizzeremo i seguenti simboli di funzione $\mathit{zero}/0$, $\mathit{succ}/1$, $\mathit{socrate}/0$ e $\mathit{padre}/1$. Il significato intuitivo che vogliamo attribuire alle due costanti $\mathit{zero}/0$ e $\mathit{socrate}/0$ è "il numero naturale 0" e "l'individuo Socrate", rispettivamente. Il significato intuitivo dei simboli di funzione $\mathit{succ}/1$ e $\mathit{padre}/1$ è invece "il successore di un numero naturale" e "il padre di un individuo", rispettivamente.

Useremo anche i simboli di predicato $\mathit{doppio}/2$, $\mathit{somma}/3$, $\mathit{uomo}/1$ e $\mathit{mortale}/1$, il cui significato intuitivo sarà: per $\mathit{doppio}(X, Y)$, il numero naturale Y è il *doppio* del numero naturale X; per $\mathit{somma}(X, Y, Z)$, il numero naturale Z è la somma dei numeri naturali X ed Y; per $\mathit{uomo}(X)$, l'individuo X è un uomo; per $\mathit{mortale}(X)$, l'individuo X è mortale. ◦

A partire dall'alfabeto si può definire il linguaggio della logica del primo ordine. Questo linguaggio ha una struttura sintattica più complessa di quello della logica proposizionale: la sua definizione induttiva deve essere effettuata in due passi.

Al primo passo viene definito un linguaggio intermedio, chiamato linguaggio dei *termini*, che viene usato nella regola base della definizione del linguaggio delle *formule*, ossia del linguaggio della logica del primo ordine.

Definizione 4.2.3 (Termini) *L'insieme dei* termini *è definito induttivamente come segue:*

- *ogni variabile in* **V** *è un termine;*
- *ogni simbolo di costante in* **SF** *è un termine;*
- *se f è un simbolo di funzione ($f \in$* **SF***) di arità $n > 0$ e $t_1, \ldots, t_n$ sono termini, allora anche $f(t_1, \ldots, t_n)$ è un termine.*

Esempio 4.2.4 Sia **SF** $= \{\mathit{zero}/0, \mathit{succ}/1, \mathit{socrate}/0, \mathit{padre}/1\}$. Le seguenti sequenze di simboli sono termini (*MiaVariabile* e X sono variabili).

(1) zero

(2) $\mathit{MiaVariabile}$

(3) $\mathit{succ}(\mathit{zero})$

(4) $\mathit{padre}(\mathit{padre}(\mathit{socrate}))$

(5) $\mathit{padre}(\mathit{succ}(X))$

(6) $\mathit{succ}(\mathit{succ}(\mathit{zero}))$

◦

Definizione 4.2.5 (Formule) *L'insieme delle* formule *(ossia della* logica del primo ordine*) è definito induttivamente come segue:*

- *se p è un simbolo di predicato di arità n e $t_1, \ldots, t_n$ sono termini, $p(t_1, \ldots, t_n)$ è una formula (detta formula* atomica*);*

- *se ϕ e ψ sono formule, lo sono anche:*
  - (ϕ)
  - $\neg\phi$
  - $\phi \lor \psi$
  - $\phi \land \psi$
  - $\phi \rightarrow \psi$
  - $\phi \equiv \psi$
- *se ϕ è una formula e V è una variabile allora anche:*
  - $\forall V\ \phi$
  - $\exists V\ \phi$

  sono formule.

Il primo punto della definizione chiarisce che useremo la notazione *prefissa* per la rappresentazione delle formule atomiche, ovvero il simbolo di predicato precede i termini. Seguendo la sintassi usuale faremo un'eccezione per il predicato di uguaglianza, usando in tal caso la notazione *infissa* e inserendo il simbolo di predicato fra i termini. In altre parole, preferiremo scrivere $X = Y$ al posto di $=(X, Y)$. Si noti che i termini vengono sempre rappresentati in notazione prefissa (cf. definizione 4.2.3).

Per rendere le formule più leggibili faremo un'ulteriore eccezione ed utilizzeremo il simbolo "$\neq$" con il seguente significato: $X \neq Y$ al posto di $\neg(X = Y)$ (a sua volta al posto di $\neg{=}(X, Y)$). È da notare che il simbolo $\neq$ è ridondante (cfr. discussione successiva alla definizione 3.3.3).

Per completare il quadro sullo *zucchero sintattico*, ovvero sulle notazioni che rendono più leggibili le formule senza introdurre elementi nuovi dal punto di vista semantico, accenniamo al fatto che, data una formula $\phi(X)$ in cui la variabile X non è quantificata, utilizzeremo la notazione $\exists! X\ \phi(X)$ come abbreviazione di $\exists X\ (\phi(X) \land (\forall Y\ \phi(Y) \rightarrow X = Y))$.

Esempio 4.2.6 Sia $SP = \{p/0, q/0, s/0\}$. Le sequenze di simboli (1)-(12) dell'esempio 3.2.3 sono formule della logica del prim'ordine. Le sequenze di simboli (13)-(16) dello stesso esempio non sono formule della logica del prim'ordine. ○

Esempio 4.2.7 Sia $\mathbf{SF} = \{zero/0, succ/1, socrate/0, padre/1\}$. Sia inoltre $\mathbf{SP} = \{doppio/2, somma/3, uomo/1, mortale/1\}$. Le seguenti sequenze di simboli sono formule della logica del primo ordine.

(1) $doppio(succ(succ(zero)), X)$

(2) $\exists X\ doppio(succ(succ(zero)), X)$

(3) $\forall X\ doppio(succ(succ(zero)), X)$

(4) $somma(succ(zero), zero, succ(zero))$

(5) $\forall X \forall Y\ somma(X, X, Y) \rightarrow doppio(X, Y)$

(6) $(\forall X \exists Y\ doppio(X,Y)) \land (\forall I \forall J \exists K\ somma(I,J,K))$

(7) $mortale(socrate)$

(8) $mortale(socrate) \land mortale(padre(socrate))$

(9) $\forall X\ mortale(X)$

(10) $(\forall X\ uomo(X) \rightarrow mortale(X)) \land uomo(socrate)$

(11) $\forall X\ uomo(X) \rightarrow uomo(padre(X))$

(12) $\forall X\ uomo(socrate)$

(13) $\forall X \forall Y\ uomo(X)$

(14) $uomo(X)$

(15) $X = socrate$

(16) $X = Y$

(17) $\forall X\ uomo(X) \rightarrow uomo(socrate)$

Le seguenti sequenze di simboli non sono formule della logica del primo ordine.

(18) $succ(zero)$

(19) $mortale(mortale(socrate))$

(20) $padre(mortale(X))$

(21) $\exists socrate\ mortale(socrate)$

(22) $\exists X\ padre(X)$

(23) $X \lor Y$

(24) $zero \land zero$

(25) $X \land zero =$

◦

4.3 Semantica

Per definire la semantica della logica del primo ordine ripercorriamo lo stesso itinerario concettuale che ci ha condotti alla definizione del significato della logica proposizionale.

1. Si definisce la nozione di *interpretazione*.
2. Si definisce come viene valutata una formula per *una particolare interpretazione*.

   Prima di fare ciò è però necessario ricordare che la struttura del linguaggio evidenzia due livelli sintattici, quello dei termini e quello delle formule. Introdurremo quindi due nozioni di valutazione: la valutazione dei termini e la valutazione delle formule. La valutazione delle formule fornirà un valore di verità T o F, esattamente come la valutazione delle formule della logica proposizionale. Ogni termine verrà invece valutato come un elemento di un generico dominio, differente, in generale, da $\{\mathsf{T}, \mathsf{F}\}$. L'introduzione di questo ulteriore dominio è il vero elemento di novità della logica del primo ordine rispetto alla logica proposizionale. In questo modo, infatti, la logica del primo ordine costituisce un sistema per asserire proprietà di insiemi di individui.
3. Si stabilisce il significato di ogni formula senza riferimento a particolari interpretazioni.

Procediamo innanzitutto a definire la nozione di *pre-interpretazione*, che è propedeutica al significato dei termini.

Definizione 4.3.1 (Pre-interpretazione) *Una* pre-interpretazione *è costituita da:*

- *un insieme non vuoto* $\mathbf{D}$ *detto* dominio di interpretazione*;*
- *una funzione che associa ad ogni simbolo di costante un elemento di* $\mathbf{D}$*;*
- *una funzione che associa ad ogni simbolo di funzione di arità* $n > 0$ *una funzione del tipo*

$$\mathbf{D}^n \longrightarrow \mathbf{D}.$$

Ricordiamo che $\mathbf{D}^n$ denota il prodotto cartesiano

$$\underbrace{\mathbf{D} \times \cdots \times \mathbf{D}}_{n \text{ volte}}$$

Notiamo che il dominio di interpretazione può essere finito o infinito.

Sia J una pre-interpretazione con dominio di interpretazione $\mathbf{D}$, k un simbolo di costante dell'alfabeto e f un simbolo di funzione dell'alfabeto. Si farà uso della seguente notazione: l'elemento di $\mathbf{D}$ associato da J a k sarà denotato da $J(k)$, mentre la funzione associata da J a f sarà denotata da $J(f)$.

Esempio 4.3.2 Sia $\mathbf{SF} = \{zero/0, succ/1\}$. La pre-interpretazione $preNAT$ è definita in questo modo:

- il dominio di interpretazione è l'insieme degli interi non negativi;
- $preNAT(zero) = 0$
- $preNAT(succ)$ è la funzione che, per ogni x, restituisce $x + 1$

Si noti che $preNAT$ associa correttamente a $succ/1$ una funzione del tipo $D \longrightarrow D$.

Sia ora $\mathbf{SF} = \{socrate/0, padre/1\}$. La pre-interpretazione $preSILLO$ è definita in questo modo:

- il dominio di interpretazione è $\mathbf{GRECI} = \{\mathsf{Socrate}, \mathsf{Trasibulo}, \mathsf{Aristotele}\}$
- $preSILLO(socrate) = \mathsf{Socrate}$
- $preSILLO(padre)$ è la funzione f del tipo $\mathbf{GRECI} \longrightarrow \mathbf{GRECI}$ definita come $f(\mathsf{Socrate}) = \mathsf{Trasibulo}$, $f(\mathsf{Aristotele}) = \mathsf{Socrate}$, $f(\mathsf{Trasibulo}) = \mathsf{Aristotele}$.

◦

Passiamo ora a dare la nozione di *interpretazione*, propedeutica al significato delle formule.

Definizione 4.3.3 (Interpretazione) *Una* interpretazione *è costituita da:*

- *una pre-interpretazione;*
- *una funzione che associa ad ogni simbolo di predicato di arità n una funzione del tipo*

$$\mathbf{D}^n \longrightarrow \{\mathsf{T}, \mathsf{F}\}.$$

Tale funzione deve assegnare al simbolo di predicato "=" la funzione che associa ad ogni coppia $(a, b) \in \mathbf{D} \times \mathbf{D}$ *il valore* T*se e solo se a è uguale a b.*

Si noti che una funzione del tipo

$$\mathbf{D}^n \longrightarrow \{\mathsf{T}, \mathsf{F}\}$$

può essere anche vista come una relazione su $\mathbf{D}^n$. Infatti, dato un qualsiasi insieme $\mathbf{U}$, una funzione $p : \mathbf{U} \longrightarrow \{\mathsf{T}, \mathsf{F}\}$ può essere rappresentata dal sottoinsieme **U'** di $\mathbf{U}$ degli elementi per i quali p restituisce valore T, ossia:

$$\mathbf{U'} = \{x |\ p(x) = \mathsf{T}\}.$$

D'altra parte un sottoinsieme **U'** di $\mathbf{U}$ può essere rappresentato da una funzione $p : \mathbf{U} \longrightarrow \{\mathsf{T}, \mathsf{F}\}$ che restituisce valore T se e solo se è invocata con elementi di **U'**, ossia: $p(x) = \mathsf{T}$ se $x \in$ **U'**, F altrimenti.

Sia I una interpretazione costruita sulla pre-interpretazione J con dominio di interpretazione $\mathbf{D}$. Si farà uso della seguente notazione:

- se x è un simbolo di costante oppure un simbolo di funzione dell'alfabeto allora $I(x) = J(x)$.
- se p è un simbolo di predicato, la funzione (o relazione) associata da I a p sarà denotata con $I(p)$.

L'ultimo punto della definizione 4.3.3 chiarisce che l'interpretazione di "=" deve rispettare i vincoli intuitivi dell'uguaglianza. Ad esempio, se abbiamo una interpretazione I con dominio di interpretazione $\{a_1, a_2, a_3\}$, allora $I(=)$ deve essere costituita necessariamente dall'insieme $\{(a_1, a_1), (a_2, a_2), (a_3, a_3)\}$.

Esempio 4.3.4 Siano $\mathbf{SF} = \{\mathit{zero}/0, \mathit{succ}/1\}$ e $\mathbf{SP} = \{\mathit{doppio}/2, \mathit{somma}/3\}$. L'interpretazione NAT è definita in questo modo:

- la pre-interpretazione è preNAT (si veda l'esempio 4.3.2)
- $I(\mathit{doppio}) = \{\langle 0, 0\rangle, \langle 1, 2\rangle, \langle 2, 4\rangle, \langle 3, 6\rangle, \ldots\} = \{\langle x, y\rangle |\ y = 2x\}$
- $I(\mathit{somma}) = \{\langle 0, 0, 0\rangle, \langle 0, 1, 1\rangle, \langle 1, 0, 1\rangle, \langle 1, 1, 2\rangle, \ldots\} = \{\langle x, y, z\rangle |\ z = x + y\}$.

Sia ora $\mathbf{SF} = \{\mathit{socrate}/0, \mathit{padre}/1\}$, $\mathbf{SP} = \{\mathit{uomo}/1, \mathit{mortale}/1\}$. L'interpretazione SILLO è definita in questo modo:

- la pre-interpretazione è $\mathit{preSILLO}$ (si veda l'esempio 4.3.2)
- $I(\mathit{uomo}) = \{\mathsf{Socrate}, \mathsf{Trasibulo}\}$
- $I(\mathit{mortale}) = \{\mathsf{Socrate}, \mathsf{Trasibulo}, \mathsf{Aristotele}\} = \mathbf{GRECI}$.

$\circ$

Prima di poter definire la funzione $\mathsf{pre\text{-}eval}$ per la valutazione dei termini, e la funzione eval, per la valutazione delle formule, è necessario introdurre il concetto di *assegnazione di variabili.*

Definizione 4.3.5 (Assegnazione di variabili) *Sia J una pre-interpretazione. Una* assegnazione di variabili *per J è una funzione che associa ad ogni variabile un elemento del dominio di interpretazione di J.*

Esempio 4.3.6 Considerando il primo caso nell'esempio 4.3.2, in cui il dominio di interpretazione è l'insieme degli interi non negativi, la funzione W tale che $W(X) = 3$, $W(Y) = 6$, e $W(Z) = 4$ per ogni variabile $Z \in \mathbf{V}$ diversa da X ed Y è un'assegnazione di variabili per preNAT.

Considerando il secondo caso dello stesso esempio, la funzione U tale che $U(X) = \mathsf{Trasibulo}$, e $U(Z) = \mathsf{Aristotele}$ per ogni variabile $Z \in \mathbf{V}$ diversa da X è un'assegnazione di variabili per $\mathit{preSILLO}$. $\circ$

Definizione 4.3.7 (Valutazione di un termine rispetto ad una pre-interpretazione e ad un'assegnazione) *Siano dati gli insiemi* **V** *e* **SF**, *e sia* **TERM** *l'insieme dei termini risultante. Sia J una pre-interpretazione e S un'assegnazione di variabili per J. Definiamo, in dipendenza da J e da S, la funzione*

$$\mathsf{pre\text{-}eval}^{J,S} : \mathbf{TERM} \longrightarrow \mathbf{D},$$

dove **D** *è il dominio di interpretazione di J. La funzione è definita in modo induttivo, seguendo la struttura induttiva dei termini:*

- *se x è una variabile:*
  $\mathsf{pre\text{-}eval}^{J,S}(x) = S(x)$
- *se c è un simbolo di costante:*
  $\mathsf{pre\text{-}eval}^{J,S}(c) = J(c)$
- *se f è un simbolo di funzione di arità n e $t_1, \ldots, t_n$ sono termini:*
  $\mathsf{pre\text{-}eval}^{J,S}(\ f(t_1, \ldots, t_n)\ ) = J(f)(\mathsf{pre\text{-}eval}^{J,S}(t_1), \ldots, \mathsf{pre\text{-}eval}^{J,S}(t_n))$

Esempio 4.3.8 Riconsideriamo la pre-interpretazione *preNAT* e l'assegnazione W (cfr. esempi 4.3.2 e 4.3.6). Si ha:

- $\mathsf{pre\text{-}eval}^{preNAT,W}(zero) = 0$
- $\mathsf{pre\text{-}eval}^{preNAT,W}(succ(succ(X))) = 5$

Considerando ora la pre-interpretazione *preSILLO* e l'assegnazione U, si ha:

- $\mathsf{pre\text{-}eval}^{preSILLO,U}(padre(padre(Socrate))) = \mathsf{Aristotele}$
- $\mathsf{pre\text{-}eval}^{preSILLO,U}(padre(X)) = \mathsf{Aristotele}$

◦

Nella definizione di valutazione delle formule si farà uso della notazione $S[X/\mathsf{d}]$ per indicare, data un'assegnazione di variabili S, un'assegnazione di variabili uguale ad S eccettuato il fatto che alla variabile X viene assegnato il valore d. Riconsiderando l'esempio 4.3.6 si ha quindi:

- $W[Y/\mathsf{9}](X) = \mathsf{3}$
- $W[Y/\mathsf{9}](Y) = \mathsf{9}$
- $U[Y/\mathsf{Aristotele}](Y) = \mathsf{Aristotele}$

Siamo ora pronti per poter definire la funzione eval per la valutazione delle formule.

Definizione 4.3.9 (Valutazione di una formula rispetto ad una interpretazione e ad un'assegnazione) *Siano dati gli insiemi* **V**, **SF** *e* **SP**, *e sia* **FORM** *l'insieme delle formule risultanti. Sia I una interpretazione avente J come pre-interpretazione, sia inoltre S un'assegnazione di variabili per J; definiamo, in dipendenza da I e da S, la funzione*

$$\mathsf{eval}^{I,S} : \mathbf{FORM} \longrightarrow \{\mathsf{T}, \mathsf{F}\}$$

La funzione è definita in modo induttivo, seguendo la struttura induttiva delle formule:

- *se p è un simbolo di predicato di arità n e $t_1, \ldots t_n$ sono termini:*
  $\mathsf{eval}^{I,S}(p(t_1, \ldots, t_n)) = I(p)(\mathsf{pre\text{-}eval}^{J,S}(t_1), \ldots, \mathsf{pre\text{-}eval}^{J,S}(t_n))$

	Termini	**Formule**
Dominio semantico	**D**	$\{\mathsf{T},\mathsf{F}\}$
Presupposti alla valutazione	pre-interpretazione	interpretazione
Valutazione da dominio sintattico a dominio semantico	pre-eval	eval

Tabella 4.1 Oggetti matematici coinvolti nella semantica della logica del prim'ordine

- *se ϕ e ψ sono formule allora (similmente alla logica proposizionale):*

$$\begin{array}{lllll}
(1) & \mathsf{eval}^{I,S}(\ (\phi)\ ) & = & \mathsf{eval}^{I,S}(\phi) \\
(2) & \mathsf{eval}^{I,S}(\neg\phi) & = & \mathsf{T}\ \textit{se}\ \mathsf{eval}^{I,S}(\phi) = \mathsf{F}; \\
 & \mathsf{eval}^{I,S}(\neg\phi) & = & \mathsf{F}\ \textit{altrimenti} \\
(3) & \mathsf{eval}^{I,S}(\phi\vee\psi) & = & \mathsf{T}\ \textit{se}\ \mathsf{eval}^{I,S}(\phi) = \mathsf{T}\ \textit{oppure}\ \mathsf{eval}^{I,S}(\psi) = \mathsf{T} \\
 & \mathsf{eval}^{I,S}(\phi\vee\psi) & = & \mathsf{F}\ \textit{altrimenti} \\
(4) & \mathsf{eval}^{I,S}(\phi\wedge\psi) & = & \mathsf{T}\ \textit{se}\ \mathsf{eval}^{I,S}(\phi) = \mathsf{eval}^{I,S}(\psi) = \mathsf{T} \\
 & \mathsf{eval}^{I,S}(\phi\wedge\psi) & = & \mathsf{F}\ \textit{altrimenti} \\
(5) & \mathsf{eval}^{I,S}(\phi\rightarrow\psi) & = & \mathsf{eval}^{I,S}(\neg\phi\vee\psi) \\
(6) & \mathsf{eval}^{I,S}(\phi\equiv\psi) & = & \mathsf{eval}^{I,S}(\ (\phi\rightarrow\psi)\wedge(\psi\rightarrow\phi)\ )
\end{array}$$

- *se ϕ è una formula e V è una variabile allora:*

$$\begin{array}{lllll}
(7) & \mathsf{eval}^{I,S}(\exists V\ \phi) & = & \mathsf{T}\ \textit{se esiste}\ \mathsf{d}\in\mathbf{D}\ \textit{tale che}\ \mathsf{eval}^{I,S[V/\mathsf{d}]}(\phi) = \mathsf{T} \\
 & \mathsf{eval}^{I,S}(\exists V\ \phi) & = & \mathsf{F}\ \textit{altrimenti} \\
(8) & \mathsf{eval}^{I,S}(\forall V\ \phi) & = & \mathsf{T}\ \textit{se per ogni}\ \mathsf{d}\in\mathbf{D}\ \textit{vale}\ \mathsf{eval}^{I,S[V/\mathsf{d}]}(\phi) = \mathsf{T} \\
 & \mathsf{eval}^{I,S}(\forall V\ \phi) & = & \mathsf{F}\ \textit{altrimenti}
\end{array}$$

La tabella 4.1 riassume gli oggetti matematici che sono stati definiti per la semantica delle formule della logica del prim'ordine.

Esempio 4.3.10 Riconsideriamo le formule dell'esempio 4.2.7 e, per le formule (1)-(6) l'interpretazione *NAT* e l'assegnazione *W*, per le formule (7)-(17) l'interpretazione *SILLO* e l'assegnazione *U*, (cfr. esempi 4.3.4 e 4.3.6). Si ha:

(1)	$\mathsf{eval}^{NAT,W}(doppio(succ(succ(zero)),X))$	$=$	F
(2)	$\mathsf{eval}^{NAT,W}(\exists X\ doppio(succ(succ(zero)),X))$	$=$	T
(3)	$\mathsf{eval}^{NAT,W}(\forall X\ doppio(succ(succ(zero)),X))$	$=$	F
(4)	$\mathsf{eval}^{NAT,W}(somma(succ(zero),zero,succ(zero)))$	$=$	T
(5)	$\mathsf{eval}^{NAT,W}(\forall X \forall Y\ somma(X,X,Y) \rightarrow doppio(X,Y))$	$=$	T
(6)	$\mathsf{eval}^{NAT,W}((\forall X \exists Y\ doppio(X,Y)) \wedge (\forall I \forall J \exists K\ somma(I,J,K)))$	$=$	T
(7)	$\mathsf{eval}^{SILLO,U}(mortale(socrate))$	$=$	T
(8)	$\mathsf{eval}^{SILLO,U}(mortale(socrate) \wedge mortale(padre(socrate)))$	$=$	T
(9)	$\mathsf{eval}^{SILLO,U}(\forall X\ mortale(X))$	$=$	T
(10)	$\mathsf{eval}^{SILLO,U}((\forall X\ uomo(X) \rightarrow mortale(X)) \wedge uomo(socrate))$	$=$	T
(11)	$\mathsf{eval}^{SILLO,U}(\forall X\ uomo(X) \rightarrow uomo(padre(X)))$	$=$	F
(12)	$\mathsf{eval}^{SILLO,U}(\forall X\ uomo(socrate))$	$=$	T
(13)	$\mathsf{eval}^{SILLO,U}(\forall X \forall Y\ uomo(X))$	$=$	F
(14)	$\mathsf{eval}^{SILLO,U}(uomo(X))$	$=$	T
(15)	$\mathsf{eval}^{SILLO,U}(X = socrate)$	$=$	F
(16)	$\mathsf{eval}^{SILLO,U}(X = Y)$	$=$	F
(17)	$\mathsf{eval}^{SILLO,U}(\forall X\ uomo(X) \rightarrow uomo(socrate))$	$=$	F

$\circ$

Notiamo alcuni aspetti.

- Le convenzioni per disambiguare la sintassi sono le stesse del calcolo proposizionale.
- Nell'ambito della semantica sarebbero inoltre necessarie convenzioni per delimitare il *campo d'azione* delle variabili quantificate: ad esempio nella formula $\forall X\ uomo(X) \wedge \exists X\ mortale(X)$ dovremmo specificare a quale quantificatore fa riferimento la variabile X in $mortale(X)$. Si noti che questo problema è analogo a quello che interviene in linguaggi di programmazione con sottoprogrammi, e che porta alla definizione di campo d'azione di un identificatore. Per ovviare a queste potenziali ambiguità, nel seguito limiteremo la nostra attenzione a formule in cui una variabile viene quantificata al più da un quantificatore.
- Il nome di una variabile quantificata (cfr. ad esempio la formula (2) nell'esempio precedente) non gioca alcun ruolo: sostituendo tutte le sue occorrenze con un altro nome di variabile che non occorre mai nella formula, il significato della formula stessa non cambia.

  Questa situazione è del tutto analoga a quella dei parametri formali nei linguaggi di programmazione.

Esempio 4.3.11 Si consideri la seguente frase italiana:

"tutti i programmi che non sono ricorsivi sono graditi agli studenti".

I simboli di predicato che assumiamo essere in **SP**, insieme con la loro specifica informale, sono i seguenti:

$prog/1$	$prog(X)$:	X è un programma
$ric/1$	$ric(X)$:	X è ricorsivo
$stud/1$	$stud(X)$:	X è uno studente
$grad/2$	$grad(X,Y)$:	X gradisce Y

Non consideriamo simboli di funzione, ovvero $\mathbf{SF} = \emptyset$. Si può facilmente esprimere la frase italiana con il linguaggio della logica del primo ordine, ottenendo la formula:

$$\forall X\ ((\mathit{prog}(X) \wedge \neg \mathit{ric}(X)) \quad \rightarrow \quad (\forall Y\ (\mathit{stud}(Y) \rightarrow \mathit{grad}(Y, X)))) \tag{4.1}$$

Consideriamo adesso la seguente interpretazione Q:

- pre-interpretazione costituita da:
  - dominio di interpretazione $\{\mathsf{ProgA}, \mathsf{ProgB}, \mathsf{Sandra}, \mathsf{Simonetta}\}$
  - irrilevante l'interpretazione per i simboli di funzione, perché non ve ne sono nella formula.
- Interpretazione dei predicati:
  - $Q(\mathit{prog}) = \{\mathsf{ProgA}, \mathsf{ProgB}\}$
  - $Q(\mathit{ric}) = \{\mathsf{ProgB}\}$
  - $Q(\mathit{stud}) = \{\mathsf{Sandra}, \mathsf{Simonetta}\}$
  - $Q(\mathit{grad}) = \{\langle \mathsf{Sandra}, \mathsf{ProgA}\rangle, \langle \mathsf{Simonetta}, \mathsf{ProgA}\rangle, \langle \mathsf{Simonetta}, \mathsf{ProgB}\rangle\}$

Verifichiamo se Q è un modello della formula. Valutiamo la formula con una qualsiasi assegnazione di variabili A.

Abbiamo che $\mathsf{eval}^{Q,A}(4.1) = \mathsf{T}$ se e solo se sono vere tutte le seguenti eguaglianze (cfr. regola (8) della definizione 4.3.9):

1. $\mathsf{eval}^{Q,A[X/\mathsf{ProgA}]}(4.2) = \mathsf{T}$
2. $\mathsf{eval}^{Q,A[X/\mathsf{ProgB}]}(4.2) = \mathsf{T}$
3. $\mathsf{eval}^{Q,A[X/\mathsf{Sandra}]}(4.2) = \mathsf{T}$
4. $\mathsf{eval}^{Q,A[X/\mathsf{Simonetta}]}(4.2) = \mathsf{T}$

dove la formula (4.2) è:

$$(\mathit{prog}(X) \wedge \neg \mathit{ric}(X)) \quad \rightarrow \quad (\forall Y\ (\mathit{stud}(Y) \rightarrow \mathit{grad}(Y, X))) \tag{4.2}$$

È facile vedere che le uguaglianze 2, 3, e 4 sono verificate in quanto Sandra e Simonetta non sono programmi e ProgB pur essendo un programma è anche ricorsivo; in questi casi quindi la sottoformula:

$$\mathit{prog}(X) \wedge \neg \mathit{ric}(X)$$

è valutata F. Dobbiamo quindi concentrarci sulla 1. Perché questa uguaglianza sia anch'essa verificata è necessario che sia verificata la sottoformula (4.3):

$$\forall Y\ (\mathit{stud}(Y) \rightarrow \mathit{grad}(Y, X)) \tag{4.3}$$

Abbiamo che $\mathsf{eval}^{Q,A[X/\mathsf{ProgA}]}(4.3) = \mathsf{T}$ se e solo se sono vere tutte le seguenti eguaglianze:

1. $\mathsf{eval}^{Q,A[X/\mathsf{ProgA}][Y/\mathsf{ProgA}]}(4.4) = \mathsf{T}$
2. $\mathsf{eval}^{Q,A[X/\mathsf{ProgA}][Y/\mathsf{ProgB}]}(4.4) = \mathsf{T}$
3. $\mathsf{eval}^{Q,A[X/\mathsf{ProgA}][Y/\mathsf{Sandra}]}(4.4) = \mathsf{T}$
4. $\mathsf{eval}^{Q,A[X/\mathsf{ProgA}][Y/\mathsf{Simonetta}]}(4.4) = \mathsf{T}$

dove la formula (4.4) è:

$$\mathit{stud}(Y) \quad \rightarrow \quad \mathit{grad}(Y, X) \tag{4.4}$$

È facile vedere che le uguaglianze 1 e 2 sono verificate in quanto ProgA e ProgB non sono studenti. D'altra parte anche le rimanenti due uguaglianze sono verificate in quanto ProgA è gradito da tutti gli studenti. Da ciò si può concludere che Q è un modello della (4.1)). Si noti che il fatto che Simonetta gradisca qualcosa di ricorsivo (ProgB) non compromette questo risultato: la (4.1)) non afferma alcunché riguardo il gradimento dei programmi ricorsivi da parte degli studenti. ◦

Ancora con riferimento alla definizione 4.3.9, notiamo che se Q è un quantificatore, V è una variabile e ϕ è una formula, la valutazione di $\mathsf{eval}^{I,S}(QV\ \phi)$ è indifferente al valore assegnato da S a V. Ad esempio, la formula (9) nell'esempio 4.3.10 verrebbe comunque valutata a T, anche se cambiassimo l'assegnazione U.

Anche qui possiamo trovare un parallelo con i linguaggi di programmazione: se un parametro formale di una procedura ed una variabile globale alla procedura stessa hanno lo stesso identificatore, il valore della variabile globale all'atto dell'invocazione della procedura non ha alcuna influenza sull'esito della procedura.

Quest'ultima considerazione implica che, per una formula ϕ in cui tutte le variabili sono quantificate, l'assegnazione alle variabili non influenza il valore di verità che viene attribuito a ϕ (si noti che il concetto di assegnazione è comunque indispensabile per la valutazione di *sottoformule* di ϕ). Per scorrelare il risultato della valutazione dall'assegnazione alle variabili anche per formule con variabili non quantificate, conveniamo che la valutazione di una formula in cui occorrono variabili non quantificate sia uguale a quella della stessa formula in cui quelle stesse variabili sono quantificate universalmente. Usando tale convenzione, la formula (14) dell'esempio 4.3.10 viene innanzitutto "convertita" nella formula $\forall X\ \mathit{uomo}(X)$, per la quale vale $\mathsf{eval}^{SILLO,U}(\forall X\ \mathit{uomo}(X)) = \mathsf{F}$.

Fatta questa precisazione, abbiamo una nozione di valutazione di formule in una data interpretazione che possiamo far corrispondere a quella della logica proposizionale. Questa astrazione ci permette di ereditare dalla logica proposizionale tutte le definizioni dei concetti appresso elencati, che assumiamo identiche a quelle viste in precedenza (cfr. definizioni 3.3.5, 3.3.8, 3.3.9, 3.3.10, 3.3.12):

- interpretazione che *soddisfa* una formula (o un insieme di formule),
- *modello* di una formula (o di un insieme di formule),
- formula (o insieme di formule) *soddisfacibile* o *insoddisfacibile*,
- *tautologia* o formula *valida*,
- formula (o insieme di formule) che *implica logicamente* una formula,
- *equivalenza logica* tra formule.

Per comodità, enunciamo alcune proprietà che saranno molto usate nel seguito.

Proposizione 4.3.12 *Siano date due formule del prim'ordine ϕ e ψ. Le seguenti tre affermazioni sono equivalenti:*

- $\phi \models \psi$ *(ϕ* implica logicamente *ψ, ovvero ψ è una* conseguenza logica *di ϕ),*
- $\models \phi\rightarrow\psi$ *($\phi\rightarrow\psi$ è* valida*),*
- $\phi\wedge\neg\psi$ *è* insoddisfacibile*.*

Esempio 4.3.13 Facendo riferimento a due formule dell'esempio 4.2.7, dimostreremo che vale la seguente implicazione logica:

$$(\forall X\ \mathit{uomo}(X)\rightarrow\mathit{mortale}(X)) \wedge \mathit{uomo}(\mathit{socrate}) \models \mathit{mortale}(\mathit{socrate}). \tag{4.5}$$

In altre parole, mostreremo che il famoso sillogismo ricordato nel paragrafo 4.1 può essere modellato in maniera efficace tramite il concetto di implicazione logica fra formule del prim'ordine.

Denoteremo le formule di interesse nel seguente modo:

- $(\forall X\ \mathit{uomo}(X)\rightarrow\mathit{mortale}(X))$ con ϕ,
- $\mathit{uomo}(\mathit{socrate})$ con ψ,
- $\mathit{mortale}(\mathit{socrate})$ con γ.

Per raggiungere il nostro scopo dobbiamo dimostrare che $\phi\wedge\psi \models \gamma$, ovvero che ogni modello di $\phi\wedge\psi$ è anche un modello di γ. Poiché i modelli possibili sono infiniti, non possiamo enumerarli, ma dobbiamo ragionare in base alle loro caratteristiche generali.

Sia I un'interpretazione e $\mathbf{D}$ il suo dominio di interpretazione. $\mathbf{D}$ deve contenere un elemento s tale che $I(\mathit{socrate}) = s$. Se I è un modello di ψ, allora $I(\mathit{uomo})$ deve contenere s. Se I è un modello di ϕ, allora $I(\mathit{uomo})$ deve essere contenuto (strettamente o no) in $I(\mathit{mortale})$. Di conseguenza, se I è un modello di $\phi\wedge\psi$, allora $I(\mathit{mortale})$ deve contenere S. Di conseguenza, I è anche un modello di γ. ◦

Esempio 4.3.14 [[22, p. 130]] Facendo riferimento allo stesso insieme di simboli di predicato, desideriamo dimostrare che:

$$\forall X\ uomo(X) \to mortale(X) \models (\forall X\ uomo(X)) \to (\forall X\ mortale(X)).$$

Intuitivamente, questa implicazione logica afferma che, assumendo che tutti gli uomini siano mortali ($\forall X\ uomo(X) \to mortale(X)$, denotato con ϕ), in quelle circostanze in cui tutti gli oggetti siano uomini ($\forall X\ uomo(X)$, denotato con ψ), tutti gli oggetti devono essere mortali ($\forall X\ mortale(X)$, denotato con γ). Dobbiamo quindi dimostrare che ogni modello di ϕ è anche un modello di $\psi \to \gamma$. Poiché tutte le variabili sono quantificate, possiamo evitare di riferirci ad assegnazioni di variabili.

In ogni modello I di ϕ vale $I(uomo) \subseteq I(mortale)$. Tali modelli si dividono in due categorie:

1. quelli, denotati con J, in cui $\mathsf{eval}^J(\neg\psi) = \mathsf{T}$,
2. quelli, denotati con K, in cui $\mathsf{eval}^K(\neg\psi) = \mathsf{F}$, ovvero $\mathsf{eval}^K(\psi) = \mathsf{T}$.

Nei primi vale $\mathsf{eval}^J(\neg\psi \vee \gamma) = \mathsf{T}$, ovvero sono anche modelli di $\psi \to \gamma$. Nei secondi (in cui tutti gli oggetti sono uomini), poiché $K(uomo) \subseteq K(mortale)$, si ha che vale $\mathsf{eval}^K(\gamma) = \mathsf{T}$, ovvero sono anch'essi modelli di $\psi \to \gamma$. ◦

Esempio 4.3.15 Facendo riferimento alla stessa notazione dell'esempio precedente, desideriamo dimostrare che:

$$(\forall X\ uomo(X)) \to (\forall X\ mortale(X)) \not\models \forall X\ uomo(X) \to mortale(X).$$

È sufficiente esibire un modello di $\psi \to \gamma$ che non sia un modello di ϕ. In particolare, mostreremo un modello M di $\neg\psi$ che non è un modello di ϕ.

Come dominio di interpretazione di M scegliamo l'insieme $\{a, b\}$. Come interpretazione dei simboli di predicato scegliamo, rispettivamente, $M(uomo) = \{a\}$ e $M(mortale) = \{b\}$. Vale $\mathsf{eval}^M(\neg\psi) = \mathsf{T}$ (quindi $\mathsf{eval}^M(\psi \to \gamma) = \mathsf{T}$) e $\mathsf{eval}^M(\phi) = \mathsf{F}$, in quanto non vale $M(uomo) \subseteq M(mortale)$. ◦

Esercizio 4.1 Facendo riferimento alla notazione degli esempi precedenti, dimostrare se valgono le seguenti implicazioni logiche oppure no: $\phi \models \psi \wedge \gamma$, $\psi \wedge \gamma \models \phi$. ◦

4.4 Rappresentazione in logica di frasi in italiano

In questo paragrafo mostriamo come si possano rappresentare frasi in italiano attraverso formule della logica del prim'ordine. In particolare analizzeremo le seguenti sei frasi:

1. nessun albero contiene cicli;
2. i programmi scritti in C sono anche programmi scritti in C++;
3. non esiste un linguaggio di programmazione perfetto;
4. c'è un grafo con più di tre archi che è connesso;
5. un programma o non contiene errori o se ne contiene non può essere eseguito;
6. gli studenti bravi scrivono programmi eleganti.

Si noti che l'ultima frase può essere interpretata in due modi:

6a. gli studenti bravi scrivono solamente programmi eleganti;

6b. gli studenti bravi scrivono qualche programma elegante.

Diamo qui di seguito l'espressione, in logica del primo ordine, delle frasi. Il significato inteso dei predicati è specificato informalmente solo quando non è ovvio.

1. $\forall X\ albero(X) \to \neg contiene\_cicli(X)$
2. $\forall X\ programma\_in(X, c) \to programma\_in(X, cpp)$

   [$programma\_in(X, Y)$: il programma X è scritto nel linguaggio Y]

3. $\neg\exists X$ *linguaggio_di_programmazione*$(X)\wedge$*perfetto*(X)

4. $\exists X\ \exists Y$ (*grafo*$(X)\wedge$*numero_archi*$(X,Y)\wedge$*maggiore*$(Y,$ *tre*$)\wedge$*connesso*$(X))$

5. $\forall X$ *programma*$(X)\rightarrow((\neg\exists Y$ *errore*$(Y)\wedge$*contiene*$(X,Y))\vee\neg$*eseguibile*$(X))$
[*contiene*(X,Y): X contiene Y]

6a. $\forall X$ (*studente_bravo*$(X)\rightarrow((\forall Y$ *programma*$(Y)\wedge$*scrive*$(X,Y)\rightarrow$*elegante*$(Y))))$

6b. $\forall X$ (*studente_bravo*$(X)\rightarrow((\exists Y$ *programma*$(Y)\wedge$*scrive*$(X,Y)\wedge$*elegante*$(Y))))$
[*scrive*(X,Y): X scrive Y]

Esercizio 4.2 [Soluzione a pagina 55] Considerando il predicato binario *ama*/2, con il significato intuitivo *ama*(X,Y): X ama Y (Y è amato da X), rappresentare in logica del prim'ordine le seguenti frasi in italiano:

1. c'è qualcuno che ama tutti,
2. c'è qualcuno che è amato da tutti,
3. tutti amano qualcuno,
4. tutti sono amati da qualcuno,
5. Mario non ama nessuno,
6. Sergio ama solo se stesso.

◦

Esercizio 4.3 Denotando con $\phi_1,\ldots,\phi_6$ le formule scritte come soluzione per l'esercizio precedente, considerare le coppie di formule ϕ_i,ϕ_j $(i\neq j)$ e dimostrare se vale $\phi_i\models\phi_j$ oppure no. Ad esempio, dimostrare che vale $\phi_1\models\phi_4$ e $\phi_4\not\models\phi_1$ (ovvero non vale $\phi_4\models\phi_1$). ◦

4.5 Aspetti computazionali

In questo paragrafo ci concentriamo su alcuni aspetti computazionali delle nozioni viste.

4.5.1 Indecidibilità

Grazie al matematico Alonzo Church è noto dal 1935 che il problema della validità di una formula della logica del prim'ordine non è decidibile. Per la proposizione 4.3.12, lo stesso può essere detto del problema della soddisfacibilità e di quello della conseguenza logica. Da allora sono stati effettuati notevoli sforzi per determinare *restrizioni sintattiche* alle regole della definizione 4.2.5 che determinano classi di formule per cui i problemi in questione sono decibibili.

4.5.2 Algoritmi

A causa del risultato di Church, non esistono speranze di progettare algoritmi corretti e completi che siano in grado di risolvere il problema della soddisfacibilità di una formula del prim'ordine.

Molti fra gli algoritmi progettati si basano sul concetto di *refutazione*, ovvero di dimostrazione che una certa formula non è soddisfacibile tramite la creazione di una sequenza di passi che mostrano una contraddizione. Ad esempio, la dimostrazione che vale $\phi\models\psi$ viene ottenuta *refutando* $\phi\wedge\neg\psi$, ovvero cercando di dimostrare che quest'ultima è insoddisfacibile. Tipicamente il tentativo di dimostrare l'insoddisfacibilità si può concludere con un successo (provando così l'implicazione logica $\phi\models\psi$) oppure no (lasciando in tal modo la questione non risolta).

In questa sede non possiamo descrivere in dettaglio gli algoritmi di refutazione, ma ci limitiamo ad introdurre mediante esempi alcuni concetti che sono usati da tali algoritmi.

Clausola: si tratta di una formula del tipo

$$\forall X_1,\ldots,X_l\ \neg A_1\vee\cdots\vee\neg A_n\vee B_1\vee\cdots\vee B_m \tag{4.6}$$

(cfr. paragrafo 3.4.2), ovvero di una formula quantificata universalmente che viene spesso scritta in maniera equivalente come

$$A_1\wedge\cdots\wedge A_n\rightarrow B_1\vee\cdots\vee B_m.$$

Facendo riferimento alla notazione dell'esempio 4.3.13, ϕ, ψ e γ sono clausole.

Conversione in forma clausale: procedimento attraverso cui una formula ψ viene trasformata in un insieme di clausole ϕ tale che ψ è soddisfacibile se e solo se ϕ è soddisfacibile. Il procedimento termina e viene tipicamente svolto in maniera abbastanza efficiente, usando spesso una sequenza di passi come quella che segue.

- Sostituzione di ψ con la sua chiusura universale (cfr. paragrafo 4.3).
- Eliminazione dei connettivi $\rightarrow$ e $\equiv$ (cfr. discussione successiva alla definizione 3.3.3) e associazione del connettivo $\neg$ solo alle formule atomiche.
- Eliminazione dei quantificatori esistenziali utilizzando il processo di *skolemizzazione*, che comporta l'introduzione di simboli di costante e funzione *freschi*, ovvero che non sono mai stati usati altrove nella formula.

  Nel caso di quantificazioni esistenziali che non occorrono nell'ambito di quantificazioni universali, la procedura introduce simboli di costante.

  Ad esempio, la formula $\exists X\ uomo(X)$ viene trasformata in $uomo(c_1)$ e la formula $\exists X\ \forall Y\ ama(X, Y)$ viene trasformata in $\forall Y\ ama(c_1, Y)$.

  Nel caso di quantificazioni esistenziali che occorrono nell'ambito di quantificazioni universali, la procedura introduce simboli di funzione, di opportuna arità.

  Ad esempio, la formula $\forall Y\ \exists X\ ama(Y, X)$ viene trasformata in $\forall Y\ ama(Y, f_1(Y))$.
- Trasformazione della *matrice* (ovvero della formula senza quantificatori) in forma normale congiuntiva (CNF), cfr. paragrafo 3.4.2.
- Eliminazione delle clasole *tautologiche* (ovvero sempre vere) e di quelle *sussunte* (ovvero implicate logicamente) da altre.

Ad esempio, la formula

$$(\forall X\ uomo(X) \rightarrow mortale(X)) \wedge uomo(socrate) \ \wedge\ \neg mortale(socrate)$$

(cfr. esempio 4.3.13) viene semplicemente trasformata nel seguente insieme di clausole:

$$\neg uomo(X) \vee mortale(X) \tag{4.7}$$
$$uomo(socrate) \tag{4.8}$$
$$\neg mortale(socrate) \tag{4.9}$$

Risoluzione: procedimento attraverso cui un insieme di clausole ϕ viene arricchito da altre clausole che sono conseguenze logiche di ϕ. Se si arriva ad aggiungere anche una *clausola vuota*, ovvero una clausola del tipo (4.6) in cui $n = m = 0$, allora si è dimostrato che ϕ è insoddisfacibile. Il procedimento (a causa dell'indecidibilità del problema della soddisfacibilità) non ha garanzia di terminazione, e sfrutta concetti come quelli espressi sinteticamente di seguito.

- L'*unificazione* è la ricerca di una sostituzione di variabili che rende due termini identici, che sia più generale possibile. Ad esempio i termini *socrate* e X vengono resi identici attraverso la sostituzione $X : socrate$.
- Dopo un opportuno procedimento di unificazione, possono esistere coppie di clausole $C = L \vee \gamma$ e $D = \neg L \vee \delta$ tali che in C occorre il letterale L e in D il letterale $\neg L$. Ad esempio, dopo la sostituzione $X : socrate$, la clausola (4.7) diventa

  $$\neg uomo(socrate) \vee mortale(socrate). \tag{4.10}$$

  Quest'ultima e la (4.8) hanno, rispettivamente, il letterale $\neg uomo(socrate)$ e $uomo(socrate)$.

  Tali coppie di clausole possono *risolvere*, ossia dare luogo alla clausola $\gamma \vee \delta$. Si noti che vale $C \wedge D \models \gamma \vee \delta$. Ad esempio, la (4.10) e la (4.8) danno luogo alla clausola

  $$mortale(socrate).$$

  Quest'ultima e la (4.9) danno luogo alla clausola vuota, dimostrando così l'implicazione logica di cui all'esempio 4.3.13.

Altri algoritmi si basano sul tentativo di trovare modelli, tipicamente finiti, che dimostrano la soddisfacibilità di una formula ψ. Si basano su un dominio di interpretazione standardizzato (di cardinalità decisa dall'utente) e, dopo una fase di trasformazione in forma clausale, sull'associazione in maniera standard di simboli di costante ad elementi di tale dominio. A questo punto viene costruita una opportuna formula della logica proposizionale la cui soddisfacibilità implica l'esistenza di una pre-interpretazione (per i simboli di funzione) e di una intepretazione che garantiscono l'esistenza di un modello per ψ.

In questo caso c'è garanzia di terminazione, ma non c'è garanzia di completezza, in quanto possono essere generati solamente modelli con dominio di cardinalità limitata.

4.5.3 Programmi disponibili

Esistono decine di programmi di dominio pubblico che implementano algoritmi basati su risoluzione e su ricerca di modelli finiti, e sono frequenti competizioni fra essi. I programmi differiscono fra loro relativamente alle tecniche di risoluzione (quella illustrata nel paragrafo precedente è la cosiddetta *risoluzione binaria*, ma ne esistono molte altre) e alle strategie di scelta delle clausole da risolvere. Sono tipicamente dotati di varie opzioni che consentono all'utente di intervenire su tali strategie. Esiste un formato standard (chiamato TPLP, acronimo di *thousand of problems for theorem proving*) per la rappresentazione di formule in forma clausale.

4.5.4 Validazione di modelli tramite dimostratori di teoremi

Nel caso della logica proposizionale abbiamo visto come il problema della validazione di modelli sia riducibile al problema della verifica di soddisfacibilità di una formula (cfr. proposizione 3.3.14 e paragrafo 3.5.1). In questo paragrafo vedremo che ciò è possibile anche nel caso della logica del prim'ordine.

In maniera simile al caso proposizionale, vogliamo ottenere una formula del prim'ordine che, se soddisfacibile, abbia come unico modello l'interpretazione che vogliamo verificare. Sia M un'interpretazione il cui dominio di interpretazione è **DOM** e ϕ una formula. Assumiamo che **DOM** sia finito e che l'insieme dei simboli di funzione usato da ϕ sia vuoto, di conseguenza, i simboli di **DOM** non occorrono in ϕ. Definiamo di seguito alcune formule utilizzando **DOM** come insieme di simboli di costante.

$$DCA \doteq \forall X \bigvee_{c \in \mathbf{DOM}} X = c \tag{4.11}$$

$$UNA \doteq \bigwedge_{c,d \in \mathbf{DOM}, c \neq d} c \neq d \tag{4.12}$$

$$LR \doteq \bigwedge_{p/n \in \mathbf{SP}, c_1,\ldots,c_n \in \mathbf{DOM}, p(c_1,\ldots,c_n) = \mathsf{T}} p(c_1, \ldots, c_n) \tag{4.13}$$

$$CWA \doteq \bigwedge_{p/n \in \mathbf{SP}, c_1,\ldots,c_n \in \mathbf{DOM}, p(c_1,\ldots,c_n) = \mathsf{F}} \neg p(c_1, \ldots, c_n) \tag{4.14}$$

La formula (4.11) esprime la cosiddetta *assunzione di dominio chiuso* (in inglese *domain closure assumption*), la quale asserisce che tutti gli oggetti di interesse appartengono all'insieme **DOM**. Di fatto, assumere la DCA rende proposizionale la formula.

La formula (4.12) esprime la cosiddetta *assunzione di nome unico* (in inglese *unique name assumption*), la quale asserisce che gli oggetti di interesse vengono denotati da un solo simbolo di costante.

La formula (4.13) serve per la cosiddetta *rappresentazione di un'interpretazione tramite letterali*, per un certo simbolo di predicato p di arità n, ovvero quanto asserito come vero dall'interpretazione viene fatto corrispondere ad un letterale positivo.

La formula (4.14) esprime la cosiddetta *assunzione di mondo chiuso* (in inglese *closed world assumption*), ovvero quanto asserito come falso dall'interpretazione viene fatto corrispondere ad un letterale negativo.

Esempio 4.5.1 Con riferimento all'esempio 4.3.11, abbiamo che

$$\begin{array}{rcl} DOM & \doteq & \{\mathsf{ProgA}, \mathsf{ProgB}, \mathsf{Sandra}, \mathsf{Simonetta}\} \\ DCA & \doteq & \forall X\ X = \mathsf{ProgA}\ \vee\ X = \mathsf{ProgB}\ \vee\ X = \mathsf{Sandra}\ \vee\ X = \mathsf{Simonetta} \\ UNA & \doteq & \mathsf{ProgA} \neq \mathsf{ProgB}\ \wedge\ \mathsf{ProgA} \neq \mathsf{Sandra}\ \wedge\ \mathsf{ProgA} \neq \mathsf{Simonetta}\ \wedge\ \mathsf{ProgB} \neq \mathsf{Sandra}\ \wedge \\ & & \mathsf{ProgB} \neq \mathsf{Simonetta}\ \wedge\ \mathsf{Sandra} \neq \mathsf{Simonetta} \\ LR & \doteq & prog(\mathsf{ProgA})\ \wedge\ prog(\mathsf{ProgB})\ \wedge\ ric(\mathsf{ProgB})\ \wedge\ stud(\mathsf{Sandra})\ \wedge\ stud(\mathsf{Simonetta})\ \wedge \\ & & grad(\mathsf{Sandra}, \mathsf{ProgA})\ \wedge\ grad(\mathsf{Simonetta}, \mathsf{ProgA})\ \wedge\ grad(\mathsf{Simonetta}, \mathsf{ProgB}) \end{array}$$

$$
\begin{aligned}
CWA \doteq\ & \neg prog(\mathsf{Sandra}) \wedge \neg prog(\mathsf{Simonetta}) \wedge \\
& \neg stud(\mathsf{ProgA}) \wedge \neg stud(\mathsf{ProgB}) \wedge \\
& \neg ric(\mathsf{Sandra}) \wedge \neg ric(\mathsf{Simonetta}) \wedge \neg ric(\mathsf{ProgA}) \wedge \\
& \neg grad(\mathsf{Sandra}, \mathsf{ProgB}) \wedge \neg grad(\mathsf{Sandra}, \mathsf{Sandra}) \wedge \neg grad(\mathsf{Sandra}, \mathsf{Simonetta}) \wedge \\
& \neg grad(\mathsf{Simonetta}, \mathsf{Sandra}) \wedge \neg grad(\mathsf{Simonetta}, \mathsf{Simonetta}) \wedge \\
& \neg grad(\mathsf{ProgA}, \mathsf{ProgA}) \wedge \neg grad(\mathsf{ProgA}, \mathsf{ProgB}) \wedge \neg grad(\mathsf{ProgA}, \mathsf{Sandra}) \wedge \\
& \neg grad(\mathsf{ProgA}, \mathsf{Simonetta}) \wedge \\
& \neg grad(\mathsf{ProgB}, \mathsf{ProgA}) \wedge \neg grad(\mathsf{ProgB}, \mathsf{ProgB}) \wedge \neg grad(\mathsf{ProgB}, \mathsf{Sandra}) \wedge \\
& \neg grad(\mathsf{ProgB}, \mathsf{Simonetta}).
\end{aligned}
$$

Si noti che il numero di letterali di UNA è quadratico rispetto alla dimensione del dominio di interpretazione. Per quanto riguarda CWA, il numero di letterali dipende dall'arità del simbolo di predicato (ad esempio, è quadratico per il predicato $grad/2$). ○

Proposizione 4.5.2 *Data una interpretazione M e una formula ϕ in cui non occorrono simboli di funzione, $M \models \phi$ vale se e solo se la formula*

$$\phi \wedge LR \wedge CWA \wedge DCA \wedge UNA \tag{4.15}$$

è soddisfacibile.

Cofrontando la (4.15)) con la (3.1), LR ha, dal punto di vista concettuale, lo stesso ruolo della seconda congiunzione, mentre CWA ha lo stesso ruolo della terza.

Dal punto di vista computazionale, la verifica di un modello è decidibile ed ha costo polinomiale (più precisamente, in LOGSPAZIO) nella dimensione dell'interpretazione, assumendo cioè come fissata la formula. Se viceversa si assume la formula come parte dell'input, allora la complessità è PSPAZIO-ardua.

4.6 Esercitazione pratica

Lo scopo di questa esercitazione è la verifica di proprietà di alcune formule del prim'ordine utilizzando appositi programmi.

I programmi da usare sono:

- Otter, disponibile alla pagina `http://www-unix.mcs.anl.gov/AR/otter`; si tratta di un dimostratore di teoremi basato sulla risoluzione, che ha una sintassi che permette la rappresentazione di formule del prim'ordine arbitrarie;
- Mace, disponibile alla stessa pagina di Otter e che ammette la stessa sintassi; implementa un ricercatore di modelli basato sull'algoritmo `DPLL` (cfr. paragrafo 3.4.2.2).

Per quanto riguarda Mace, le opzioni più importanti sono:

- `-n`, che permette di stabilire la dimensione del dominio di interpretazione;
- `-p`, che formatta il modello trovato in maniera leggibile,
- `-h`, che stampa tutte le opzioni.

4.6.1 Prima parte

Prendere in considerazione le formule, le interpretazioni e le implicazioni logiche degli esempi 4.2.7, 4.3.4, 4.3.10, 4.3.11, 4.3.14, 4.3.15, e 4.5.1, e degli esercizi 4.1, 4.2 e 4.3.

Tenendo conto della proposizione 4.3.12, vogliamo raggiungere i seguenti obiettivi.

- Per ogni formula f vogliamo sapere se è f soddisfacibile, e in caso affermativo vogliamo un suo modello.
- Per ogni implicazione logica suggerita (cfr., ad es., esercizi 4.2 e 4.3), vogliamo sapere se valga oppure no.
- Vogliamo verificare che sia possibile risolvere il problema della validazione di modello (cfr. ad es., esempio 4.5.1) usando Mace.

Nell'utilizzo di Otter:

- analizzare le formule prodotte durante il meccanismo di conversione in forma clausale;
- analizzare il meccanismo di risoluzione e come viene prodotta la clausola vuota.

Nell'utilizzo di Mace:

- analizzare in che modo vengono fatti corrispondere i simboli di costante agli elementi del dominio di interpretazione;
- analizzare i modelli prodotti.

4.6.2 Seconda parte

Considerare il predicato $g/2$ il cui significato inteso è: $g(X, Y)$: nel grafo g esiste un arco uscente dal nodo X ed entrante nel nodo Y. Scrivere in logica del prim'ordine formule che rappresentano le seguenti frasi in italiano, usando solamente i predicati g e "=":

1. esiste un cappio;
2. ogni nodo ha almeno un successore;
3. ogni nodo ha almeno un predecessore;
4. ogni nodo ha al più un successore;
5. ogni nodo ha al più un predecessore;
6. ogni nodo ha almeno due successori;
7. ogni nodo ha almeno due predecessori;
8. ogni nodo ha al più due successori;
9. ogni nodo ha al più due predecessori;
10. esiste una cricca di tre nodi;
11. è possibile raggiungere ogni nodo da ogni altro nodo, passando per al più un nodo intermedio;
12. è possibile raggiungere ogni nodo da ogni altro nodo, passando per al più due nodi intermedi.

Per ognuna delle formule ϕ ottenute, facendo riferimento ad un universo di interpretazione di tre o quattro elementi, definire due interpretazioni tali che, rispettivamente,

- la prima sia un modello di ϕ,
- la seconda non sia un modello di ϕ.

Tenendo presente la proposizione 4.5.2, verificare, usando Otter e Mace, se le interpretazioni definite siano oppure no modelli delle rispettive formule.

4.6.3 Terza parte

Oltre al predicato $g/2$ (cfr. paragrafo 4.6.2) considerare i predicati $r/1$, $v/1$ e $b/1$ il cui significato inteso è: $r(X)$: il nodo X è colorato di rosso (risp. $v(X)$: verde, $b(X)$: blu).

Scrivere in logica del prim'ordine formule che rappresentano le seguenti frasi in italiano, usando solamente i predicati g, r, v, b e "=":

1. il grafo è 2-colorato da r e v, ovvero ogni nodo è rosso o verde, e nodi adiacenti hanno diverso colore;
2. il grafo è 3-colorato da r, v e b.

Procedere come al paragrafo 4.6.2.

4.6.4 Quarta parte

Riportiamo di seguito alcuni cartelli pubblicitari/informativi visti di recente.

1. Avviso di un negozio:

Si effettuano spedizioni in tutto il mondo Anche in Italia

2. Scritta su una carta telefonica di una famosa ditta di telecomunicazioni.

Con questa carta telefonica puoi chiamare dall'Italia tutto il mondo

3. Cartello in un negozio:

Si cambia la merce solo se accompagnata dallo scontrino Non si cambiano i rossetti

L'obiettivo è:

- tradurre le affermazioni dall'italiano alla logica del prim'ordine, aggiungendo, se necessario, alcune formule che rappresentino informazione implicita;
- utilizzare Otter per verificare le conseguenze delle affermazioni.

In particolare, vogliamo, rispettivamente:

1. verificare che c'è una *ridondanza* nel cartello;
2. verificare che con la carta possiamo chiamare Roma da Roma (cosa che, per inciso, non è possibile nella realtà :-( );
3. verificare se è possibile cambiare un profumo.

4.7 Nota bibliografica

Per una trattazione più estesa della logica matematica si rimanda a testi specifici, come [30]. Un testo accurato ma di facile consultazione è [22]. Per una trattazione estesa dei casi decidibili del problema della soddisfacibilità di una formula del prim'ordine si rimanda a [5]. Le assunzioni DCA, UNA e CWA, con le loro relazioni alla logica del prim'ordine come linguaggio di interrogazione, sono spiegati in [37]. Per una trattazione esaustiva del sistema Otter e dei metodi di risoluzione da esso impiegati, rimandiamo a [36, 43]. Il sito `www.tptp.org` contiene riferimenti a dimostratori di teoremi e molte formule che vengono usate come *benchmark* per stabilire l'efficacia dei medesimi.

4.8 Esercizi proposti

Esercizio 4.4 Si consideri la seguente formula del prim'ordine:

$$\forall X\ s(f(X,g)) \wedge r(h, f(l(X), X)) \vee \exists Y\ s(l(f(g,g)))$$

Quali sono le variabili, i simboli di costante, quelli di funzione e quelli di predicato che occorrono nella formula? Qual è l'arità di questi ultimi?

Esercizio 4.5 Scrivere alcune formule della logica del prim'ordine in cui occorrano i termini dell'esempio 4.2.4.

Esercizio 4.6 Per quale motivo le seguenti sequenze di simboli non sono formule del prim'ordine?

$$\forall X\ s(f(X,g)) \wedge f(h,g)$$

$$\forall X\ s(f(X,g)) \wedge s(s(f(X,g)))$$

Esercizio 4.7 Per ognuna delle formule logiche del paragrafo 4.4 si dia un'interpretazione che è un modello ed una che non lo è.

Esercizio 4.8 [Soluzione a pagina 55] Si consideri la formula ϕ della logica del primo ordine

$$\exists X \ (\ q(X) \ \wedge \ \forall Y \ (p(f(X), Y) \rightarrow q(Y)) \ )$$

Si dica a quale categoria sintattica (predicato, funzione, ecc.) appartiene ciascuno dei simboli della formula. Si faccia poi riferimento al dominio di interpretazione $Z = \{1, 2, 3\}$, e si definisca, su tale dominio, una interpretazione I che renda vera ϕ, ed una interpretazione J che renda falsa ϕ.

Esercizio 4.9 [Soluzione a pagina 56] Si consideri un alfabeto della logica del primo ordine con simboli di costante c, d e simboli di funzione $f/2$ e $g/1$ (cioè f ha arità 2 e g arità 1). Si scriva:

1. un qualunque termine con due occorrenze del simbolo g;
2. un qualunque termine con esattamente una occorrenza di f, esattamente una di g, ed un numero qualunque di occorrenze di c e d;
3. un qualunque termine con almeno una occorrenza di c, almeno una di d e almeno due di f.

Si faccia poi riferimento al dominio di interpretazione $Z = \{1, 2, 3\}$, si definisca una pre-interpretazione J su tale dominio, e si illustri il processo di valutazione dei termini scritti per i punti 1, 2 e 3 in tale pre-interpretazione.

Esercizio 4.10 Si considerino gli insiemi di simboli di funzione e di predicato $\mathbf{SF} = \{padre/1, madre/1\}$, $\mathbf{SP} = \{maschio/1, femmina/1, fratello/2, sorella/2, cugino/2, cugina/2, =/2\}$, con il significato inteso ($fratello(X, Y)$: Y è fratello di X). Il concetto di "fratello" può essere espresso mediante la formula del prim'ordine

$$\begin{array}{lll} \forall X \forall Y \ fratello(X, Y) \equiv & (& padre(X) = padre(Y) \wedge \\ & & madre(X) = madre(Y) \wedge maschio(Y) \quad) \end{array}$$

Si scrivano tre formule della logica del prim'ordine che esprimono i concetti di "sorella", "cugino" e "cugina".

Esercizio 4.11 Si considerino gli insiemi **SF** e **SP** dell'esercizio precedente, e si fornisca una loro interpretazione che rappresenti le relazioni esistenti fra i propri consanguinei (o qualsiasi altro gruppo familiare).

Tale interpretazione è un modello delle formule del prim'ordine di cui all'esercizio precedente?

Esercizio 4.12 Dare un modello, se esiste, della formula della logica del prim'ordine $p(a) \wedge \neg \forall X \ p(X)$.

Esercizio 4.13 Si formuli la frase "Ogni uomo che si rispetti non ruba, o se ruba restituisce il maltolto" in logica del prim'ordine, facendo uso solamente dei seguenti simboli di predicato: *uomo*/1, *rispettabile*/1, *ruba*/2 [*ruba*(X, Y): X ruba Y], *restituisce*/2[(*restituisce*(X, Y): X restituisce Y].

Esercizio 4.14 [Soluzione a pagina 56] Si consideri un alfabeto della logica del primo ordine con i seguenti simboli di predicato (specificati con il significato inteso):

- *stud*/1, *prof*/1, *corso*/1, *fac*/1, con gli ovvi significati;
- *iscr*/2, dove *iscr*(X, Y) significa che lo studente X è iscritto alla facoltà Y;
- *aff*/2, dove *aff*(X, Y) significa che il corso X afferisce alla facoltà Y;
- *rel*/2, dove *rel*(X, Y) significa che il professore X è il relatore della tesi dello studente Y;
- *tit*/2, dove *tit*(X, Y) significa che il professore X è il titolare del corso Y.

Utilizzando i simboli di questo alfabeto, si scrivano due formule in logica del primo ordine che formalizzano le seguenti frasi:

1. Ogni professore ha un corso di cui è titolare.
2. Il relatore della tesi di uno studente deve essere un professore titolare di un corso che afferisce alla facoltà in cui lo studente è iscritto.

Si definisca poi una interpretazione che abbia almeno 3 oggetti che soddisfano il predicato *stud*, che abbia almeno un oggetto che soddisfa il predicato *prof*, e che sia un modello delle due formule scritte.

Esercizio 4.15 Nel paragrafo 4.6.2 e nel successivo abbiamo visto come usare OTTER come strumento per la *verifica di proprietà*.

In effetti possibile utilizzarlo anche come linguaggio di *interrogazione*, ad esempio per conoscere:

1. l'insieme dei cappi;
2. l'insieme dei nodi che hanno almeno un successore;
3. l'insieme dei nodi che hanno almeno un predecessore;

e così via.

Progettare un metodo generale per effettuare interrogazioni e verificarlo in pratica sugli esempi prodotti come soluzione per l'esercitazione del paragrafo 4.6.2.

Esercizio 4.16 Nel paragrafo 4.6.2, ai punti 11 e 12, si chiede di rappresentare in logica del prim'ordine il problema della 2-raggiungibilità e della 3-raggiungibilità in un grafo, rispettivamente. Come vedremo nel capitolo 5, non è possibile scrivere in logica del prim'ordine una formula che rappresenti la *raggiungibilità*, ovvero attraversando un numero arbitrario di nodi intermedi. Dare una spiegazione intuitiva di questo fatto.

4.9 Soluzione di alcuni esercizi

Soluzione esercizio 4.2. Diamo qui di seguito l'espressione, in logica del primo ordine, delle frasi.

1. $\exists X\ \forall Y\ ama(X, Y)$,
2. $\exists X\ \forall Y\ ama(Y, X)$,
3. $\forall Y\ \exists X\ ama(Y, X)$,
4. $\forall Y\ \exists X\ ama(X, Y)$,
5. $\forall X\ \neg ama(mario, X)$, oppure $\neg\exists X\ ama(mario, X)$,
6. assumendo che la frase implichi che Sergio ama se stesso, possiamo utilizzare le seguenti tre formule, fra loro equivalenti:
   - $ama(sergio, sergio) \wedge \forall X\ ama(sergio, X) \rightarrow X = sergio$, oppure
   - $ama(sergio, sergio) \wedge \neg\exists X\ ama(sergio, X) \wedge X \neq sergio$, oppure
   - $ama(sergio, sergio) \wedge \exists! X\ ama(sergio, X)$.

○

Soluzione esercizio 4.8. Per quanto riguarda le categorie sintattiche, X, Y sono variabili, f è un simbolo di funzione di arità 1 q e p sono simboli di predicato, di arità 1 e 2, rispettivamente.

Un'interpretazione I che rende vera ϕ è la seguente:

- pre-interpretazione dei simboli di funzione:
  $preI(f)$ = arbitraria funzione $g : Z \longrightarrow Z$
- interpretazione dei simboli di predicato:
  $I(q) = \{1, 2, 3\}$, $I(p)$ = arbitrario sottoinsieme di $Z \times Z$.

Un'interpretazione J che rende falsa ϕ è la seguente:

- pre-interpretazione dei simboli di funzione:
- $preJ(f)$ = arbitraria funzione $g : Z \longrightarrow Z$
- interpretazione dei simboli di predicato:
  $J(q) = \{\}$, $J(p)$ = arbitrario sottoinsieme di $Z \times Z$.

○

Soluzione esercizio 4.9. I termini richiesti sono definiti nel seguente modo:

1. $g(g(c))$
2. $g(f(c,d))$
3. $f(f(c,d),c)$

Definiamo J nel seguente modo:

- $J(c) = 1$
- $J(d) = 2$
- $J(g) = r : Z \longrightarrow Z$ tale che $r(i) = i$ per ogni $i \in Z$
- $J(f) = s : Z \times Z \longrightarrow Z$ tale che $s(i,j) = j$ per ogni $i, j \in Z$

Non essendoci variabili nei termini di cui ai punti 1,2,3, possiamo fare riferimento alla valutazione di tali termini a prescindere da qualsiasi assegnazione di variabili.

1. $$\begin{array}{ll} \textsf{pre-eval}^J(g(g(c))) & = \\ J(g)(\textsf{pre-eval}^J(g(c))) & = \\ J(g)(J(g)(\textsf{pre-eval}^J(c))) & = \\ J(g)(J(g)(J(c))) & = \\ J(g)(J(g)(1)) & = \\ J(g)(r(1)) & = \\ r(r(1)) & = \\ r(1) & = \\ 1 & \end{array}$$

2. $$\begin{array}{ll} \textsf{pre-eval}^J(g(f(c,d))) & = \\ J(g)(\textsf{pre-eval}^J(f(c,d))) & = \\ J(g)(J(f)(\textsf{pre-eval}^J(c), \textsf{pre-eval}^J(d))) & = \\ J(g)(J(f)(J(c), J(d))) & = \\ J(g)(J(f)(1,2)) & = \\ J(g)(s(1,2)) & = \\ r(s(1,2)) & = \\ r(2) & = \\ 2 & \end{array}$$

3. $$\begin{array}{ll} \textsf{pre-eval}^J(f(f(c,d),c)) & = \\ J(f)(\textsf{pre-eval}^J(f(c,d)), \textsf{pre-eval}^J(c)) & = \\ J(f)(J(f)(\textsf{pre-eval}^J(c), \textsf{pre-eval}^J(d)), \textsf{pre-eval}^J(c)) & = \\ J(f)(J(f)(J(c), J(d)), J(c)) & = \\ J(f)(J(f)(1,2),1) & = \\ J(f)(s(1,2),1) & = \\ s(s(1,2),1) & = \\ s(2,1) & = \\ 1 & \end{array}$$

$\circ$

Soluzione esercizio 4.14. Le formule richieste sono:

1. $\forall X \; \mathit{prof}(X) \rightarrow \exists Y \; \mathit{corso}(Y) \wedge \mathit{tit}(X,Y)$
2. $\forall X \forall Y \forall W \; \mathit{prof}(X) \wedge \mathit{stud}(Y) \wedge \mathit{fac}(W) \wedge \mathit{rel}(X,Y) \wedge \mathit{iscr}(Y,W) \rightarrow$
   $\exists Z \; \mathit{corso}(Z) \wedge \mathit{tit}(X,Z) \wedge \mathit{aff}(Z,W)$

Dominio di interpretazione = $\{\textsf{Maxwell}, \textsf{Rossi}, \textsf{Bianchi}, \textsf{Verdi}, \textsf{Ingegneria}, \textsf{Elettromagnetismo}\}$.
Un'interpretazione J che è un modello delle formule 1. e 2. è:

- $J(\mathit{prof}) = \{\mathsf{Maxwell}\}$
- $J(\mathit{stud}) = \{\mathsf{Rossi}, \mathsf{Bianchi}, \mathsf{Verdi}\}$
- $J(\mathit{corso}) = \{\mathsf{Elettromagnetismo}\}$
- $J(\mathit{fac}) = \{\mathsf{Ingegneria}\}$
- $J(\mathit{iscr}) = \{\langle \mathsf{Rossi}, \mathsf{Ingegneria} \rangle\}$
- $J(\mathit{aff}) = \{\langle \mathsf{Elettromagnetismo}, \mathsf{Ingegneria} \rangle\}$
- $J(\mathit{rel}) = \{\langle \mathsf{Maxwell}, \mathsf{Rossi} \rangle\}$
- $J(\mathit{tit}) = \{\langle \mathsf{Maxwell}, \mathsf{Elettromagnetismo} \rangle\}$

$\circ$