Logica dei predicati

-

Linguaggio predicativo (formalizzazione)

- Introduciamo il linguaggio della logica predicativa in forma rigorosa.

- Il **linguaggio predicativo** e` un'estensione del linguaggio proposizionale.

- Seguiremo lo stesso approccio:

  - Vocabolario

  - Sintassi

  - Semantica

Vocabolario

- Introduciamo innanzitutto i simboli per gli *individui*.

- Questi simboli vengono chiamati “termini” o ‘termini singolari”.

- L’insieme dei *termini singolari* e` dato dall’unione dei tre insiemi seguenti.

1.

Un insieme di **costanti individuali** (nomi):

{m, n, o, ... }

2.

Un insieme di **variabili individuali**:

{x, y, w, … }

3.

Un insieme di **nomi arbitrari**:

{ a, b, c, … }

→ La loro funzione sara` piu` chiara in seguito, nel calcolo.

Per ora si puo` pensare a questi nomi come a meta` tra le costanti e le variabili, come una sorta di nome di un individuo particolare (come i nomi) ma non specificato (come le variabili).

- Come nel linguaggio proposizionale, potremmo evitare ambiguita` utilizzando gli indici:

  - **costanti individuali:** 'c_1','c_2','c_3', ...
  - **nomi arbitrari:** 'a_1', 'a_2', 'a_3', ...
  - **-variabili individuali:** 'v_1','v_2 ','v_3', ...

→ Di nuovo, per praticita` lo evitiamo.

- I termini che abbiamo introdotto sono tutti semplici, mai composti.
- Non abbiamo quindi bisogno di una sintassi che ne governi la formazione.

- Pero` nel linguaggio naturale abbiamo termini singolari piu` complessi, come le descrizioni definite.

  Ad esempio:

  “il re di Francia”, “la macchina preferita di Federico”, “2 + 3”, ...

- Per avere espressioni di questo tipo dovremmo introdurre simboli di funzioni, che si applicano a termini singolari e restituiscono ancora termini singolari.

  Ad esempio. ‘f( )’, applicata a ‘c’, darebbe ‘f(c)’ che sarebbe un nome complesso.

  Potremmo avere anche iterazioni ‘f(f(c))’

- Per semplicita`, nei evitiamo nomi complessi di questo tipo.

- Abbiamo poi simboli di predicati.

  Abbiamo due gruppi di simboli:

  1.

  Un insieme di predicati a n posti (con n possibilmente anche 0).

  {P( ), Q( ), R( ), ...}

- Per precisione, potremmo esplicitare il numero di posti (l’arieta`).

$\{P^1(\ ),\ Q^2(\ ,\ ),\ R^1(\ ),\ \ldots\}$

2.

Un simbolo di predicato a due posti per l’identita`.

$\{=\}$

- Abbiamo poi l’insieme dei connettivi logici

{&, v ,-, →, ↔ }

Sono i simboli che gia` conosciamo inclusi nel linguaggio proposizionale.

- Abbiamo poi simboli per i quantificatori:

$\{\forall, \exists\}$

- Infine abbiamo i simboli ausiliari, ovvero le parentesi:

  $\{(\ , \ )\}$

- Il vocabolario di L1, o linguaggio del primo ordine, o linguaggio predicativo, e` dato dall’unione degli insiemi descritti finora, ovvero:

  termini singolari, simboli di predicati, connettivi, quantificatori, parentesi.

- Nient’altro e` un simbolo di L1.

Sintassi di L1

- Come nel caso del calcolo proposizionale

  una **formula** è una sequenza lineare finita (non vuota) di simboli del vocabolario di L1 qualsiasi tipo.

- Questo ci serve piu` che altro ad escludere guazzabugli di simboli, ma quel che ci interessa e` sopratutto quali sequenze, ovvero quali formule, sono **ben formate**.

Formule ben formate di L1

- Per prima cosa definiamo le formule **atomiche**

- Nel calcolo proposizionale, le formule atomiche sono solo le lettere enunciative.

- Ma ora possiamo andare piu` in profondita`.

Formule atomiche

- Se:

  $A^n(\ )$ è un qualsiasi simbolo predicativo a n posti (con n che puo` essere 0).

  e t_1 , ... , t_n sono n (occorrenze di) **termini individuali,** ovvero costanti o variabili (non necessariamente diversi).

  (I nomi arbitrari possono essere assimilati, fino al calcolo, alle variabili)

  Allora $A^n(t_1, \dots, t_n)$ è una **formula atomica** (ben formata).

- Nota che stiamo di nuovo usando meta-variabili, sia per termini (t) che per predicati (A).

- Le meta-variabili non fanno parte di L1.

  (Potremmo utilizzare di nuovo lettere greche, o di altri alfabeti, per evitare eventuali confusioni)

- Nota anche che i posti del predicato (n) devono essere uguale al numero di occorrenze di termini (n) (in modo da non lasciare posti vuoti).

- Nel caso in cui n sia 0, il predicato e` una lettera enunciativa.

  Per semplicita`, noi considereremo solitamente n maggiore a 0.

- Esempi di formule atomiche ben formate:

(* l’arieta` dei predicati puo` essere omessa per semplicita`)

$P^1(m)$

$R^2(x,n)$

… *

$W(y,o,m,x)$

$x=n$

$y=y$

- Nota bene:

  Le fbf atomiche non contengono connettivi o quantificatori.

- Una volta definite le formule atomiche, possiamo definire le **formule ben formate (fbf)** in generale.

- La definizione e` induttiva, simile, per forma, a quella gia` usata per il linguaggio proposizionale.

(1) *(**Base** induttiva)*

Qualsiasi **fbf atomica** è una **fbf**;

*(Clausole/**passi** induttivi)*

(2) se A è un **fbf**, allora -A è un **fbf**;

(3) Se A e B sono **fbf**, allora $(A \rightarrow B)$, $(A \,\&\, B)$, $(A \vee B)$, $(A \leftrightarrow B)$ sono **fbf**;

→ Queste clausole sono analoghe a quelle del linguaggio proposizionale.

Ora abbiamo pero’ due clausole in piu`, per i due quantificatori.

(4)

- Se A e` una **fbf** che **non** contiene (occorrenze di) $\forall v$ o di $\exists v$,

- allora $\forall v A$ e $\exists v A$ sono **fbf**.

Oppure, seguendo il Varzi:

(4 *bis*)

- se A e` una **fbf** che contiene la **costante** c,
- in A **non** occorre la variabile v,
- $A_{v/t}$ e` la formula ottenuta sostituendo almeno una occorrenza di t con v,

- allora $\forall v A_{v/t}$ e $\exists v A_{v/t}$ sono **fbf**.

- Ad esempio:

Se A e`: P(m,o)

in cui occorre, in particolare, la costante: m

sostituendo m con la variabile x (che non c'e` in A), otteniamo:

P(x,o)   (che sarebbe $A_{v/t}$)

Mettendo davanti un quantificatore, otteniamo una fbf:

$\forall x P(x,o)$ oppure $\exists x P(x,o)$

- Si *estrae la costante* (uno o più occorrenze) e si *inserisce la singola variabile* (in una o più occorrenze).

- La fbf A di partenza potrebbe gia ` contenere un quantificatore. Ad esempio:

- Se A e`: $\exists xP(x,o)$

  in cui occorre il termine: o

  sostituendo o con la variabile y, otteniamo:

  $\exists xP(x,y)$ (che sarebbe $A_{v/t}$)

  → *Nota che qua x occorre gia` in A, quindi dobbiamo usare una variabile diversa, y.*

Mettendo poi un quantificatore davanti, otteniamo una fbf:

$\forall y\exists xP(x,y)$ oppure $\exists y\exists xP(x,y)$

- Lo scopo della complessita` della clausola 4 (sia nella prima che nella versione bis di Varzi) è quello di assicurarsi che due o più quantificatori che usano la stessa variabile non si applichino mai a parti sovrapposte di una stessa formula.

- Altrimenti i due quantificatori entrerebbero in conflitto.

- Cioe` vogliamo evitare di avere come fbf formule come:

$\forall x \exists x\ P(x,x)$

o

$\forall y \exists y\ (P(y) \rightarrow y=y)$

- Che non sapremmo come leggere

- Possiamo invece permettere formule come

$\forall xP(x)\ \&\ \exists xQ(x,x)$

Si verifichi che rispettano le clausole!

- E, per la clausola 4 (ma non 4 *bis*) persino:

$\forall x\exists y\ (P(y) \rightarrow y=y)$

Dove “∀x” ha un’occorrenza vacua, perche` “x” non occorre. Questo e` strano ma non un problema)

- *Clausola di chiusura:*

  (5)

  Nient’altro e` una **fbf** di L1.

Esempi di fbf

- Supponiamo di voler dimostrare che $\forall x\ (F(x) \rightarrow G(x))$ è una fbf.

a. $F(x)$ e $G(x)$ sono fbf Per 1.

b. $F(x) \rightarrow G(x)$ e` una fbf per 3

c. $\forall x\ (F(x) \rightarrow G(x))$ e` una fbf per 4.

(per 4bis, la fbf in cui c'e` la costante sostituita con x potrebbe essere $(F(c) \rightarrow G(c))$

Esempi di fbf

- Supponiamo di voler dimostrare che $\forall x \; \forall x \; (F(x) \rightarrow G(x))$ **non** è una fbf.

a. $F(x)$ e $G(x)$ sono fbf Per 1.

b. $F(x) \rightarrow G(x)$ sono fbf per 3

c. $\forall x \; (F(x) \rightarrow G(x))$ per 4.

(dove la fbf in cui c'e` la costante sostituita con x e` $(F(c) \rightarrow G(c))$

ma

d. $\forall x \; \forall x \; (F(x) \rightarrow G(x))$ **non** puo` esser ottenuta per 4 (ne` per 4bis).

- Per esserlo, ci servirebbe una fbf in cui $\forall x$ (per 4) o x (per4 bis) non occorre, a cui aggiungere, $\forall x$
- ma nel candidato $\forall x\ (F(x) \rightarrow G(x))$, x c’e` gia`.

- Quindi non c’e` modo di applicare 4 o 4bis, per ottenere $\forall x\ \forall x\ (F(x) \rightarrow G(x))$.

Alcune nozioni sintattiche

Alcune nozioni sintattiche si ottengono dagli opportuni, e immediati, adattamenti dal linguaggio proposizionale.

Ad esempio:

- **sottoformula**,
- **ambito**,
- **operatore principale**.

- In particolare, possiamo estendere queste nozioni ai quantificatori.

- Importante e` l'**ambito** di una occorrenza di un quantificatore.

  (che e`, la piu` piccola fbf che contiene tale occorrenza).

- L'ambito di un quantificatore e` importante per definire la nozione di **variabile libera** e quella di **enunciato**.

Variabili libere

Le occorrenze di una variabile in una fbf che sono contenute nell’ambito di un quantificatore sono dette **vincolate**.

L’idea e` che tali variabile e` “controllata” dal quantificatore.

- Segue, dalla definizione di fbf, che ogni occorrenza di una variabile puo` essere vincolata al massimo da un solo quantificatore.

- Si noti anche che o tutte le occorrenze di una variabile v nell’ambito di un quantificatore sono vincolate o nessuna lo e`.

- Una occorrenza di una variabile che non è vincolata da alcun quantificatore e` detta **libera**.

- Pero` in una fbf una variabile puo` avere occorrenze libere e occorrenze vincolate (in sottoformule e quindi ambiti diversi).

- Ad esempio:

  $P(x)\ \&\ \forall x\ (F(x))$

  La prima occorrenza di x e` libera, la seconda vincolata.

Enunciati

Tra le fbf possiamo distinguere quelle che non contengono nessuna occorrenza libera di variabili.

Ovvero o non hanno variabili o, se le hanno, occorrono solo vincolate.

- Chiamiamo queste fbf **formule chiuse** o **enunciati**.

Ad esempio, sono **enunciati**:

$P(t,r) \& Q(m)$

$Q(s)$

$R(t) \rightarrow \forall x\text{-}R(x)$

Non sono invece enunciati, le seguenti fbf:

$R(y) \rightarrow -\forall x\, R(x)$

$R(x) \rightarrow \forall x\, R(x)$

$P(x,m)$

Gli enunciati di I1 sono importanti perche` sono quelli che corrispondono agli enunciati dichiarativi del linguaggio naturale, in quanto si possono considerare come facenti una affermazione completa.

- Gli enunciati di I1 (e non le fbf in generale) sono quindi i candidati migliori ad essere valutati come veri o falsi e a comparire nelle nostre formalizzazioni.

Alcune formalizzazioni degne di nota

Descrizioni definite

Si traduca, in L1,

“l’attuale re diFrancia e` calvo”

$\exists x Kx \ \& \ Cx \ \& \ \forall y(Ky \rightarrow y=x)$

Questa formalizzazione e` molto famosa ed e` stata data da Bertrand Russell nell'articolo "On denoting".

Mostra come formalizzare le descrizioni definite e perche` affermazioni che contengono dessrizioni definite di oggetti non esistenti (come l'attuale re di Francia) siano da ritenersi false.

Perche1` almeno un congiunto e` falso (non esiste un x tale che x e` l'attuale re di Francia).

Affermazioni numeriche

Nel linguaggio predicativo L1 possiamo anche fare affermazioni numeriche.

- Questo mostra che in questo linguaggio logico e` possibile descrivere nozioni aritmetiche.

Anzitutto, in L1 possiamo considerare un simbolo per l’identita` = che puo` essere preso come avente certe caratteristiche fisse.

(= e` una relazione:

simmetrica, riflessiva, transitiva)

simmetrica: se x=y, allora y=x

riflessiva: x=x

transitiva: se x=y e y=z, allora x=z

Queste caratteristiche rendono l’identita` una relazione di equivalenza.

Anzi la piu` piccola relazione di equiavlenza.

Usando l’identita`, possiamo esprimere affermazioni numeriche come segue.

Ad esempio:

Esistono **almeno** tre oggetti diversi.

$\exists x \exists y \exists z\ (-(x=y)\ \&\ -(x=z)\ \&\ -(y=z)\ )$

Esistono **almeno** due oggetti diversi.

$\exists x \exists y \, (-(x=y)\,)$

Esistono **al massimo** due oggetti diversi:

$\forall x \forall y \forall z \, ((x=y) \vee (x=z) \vee (y=z))$

- Ci sono **esattamente 2** oggetti:

  → Questo significa che ci sono **almeno** 2 e **al massimo** 2 oggetti.

- È quindi sufficiente unire formule come quelle sopra:

$$\exists x \exists y \text{-}(x=y) \ \& \ \forall x \forall y \forall z \ ((x=y) \vee (y=z) \vee (z=x))$$

- Queste affermazioni possono essere semplificate manipolando
- i simboli, utilizzando equivalenze e regole (che pero` dobbiamo ancora introdurre).

Potremmo cosi` ottenere, ad esempio:

$\exists x \exists y$ -(x=y) & $\forall z$ ((z=x) v (z=y))