Logica per la Programmazione

Lezione 6

- Logica del Primo Ordine
  - Motivazioni
  - Sintassi di Termini e Formule
  - Formule aperte e chiuse

Limiti del Calcolo Proposizionale

- Nella formalizzazione di enunciati dichiarativi, gli **enunciati atomici** non hanno struttura (sono rappresentati da variabili proposizionali)
- **Es: "Alberto va al cinema con Bruno o va al teatro con Carlo** "
  Introduciamo 4 proposizioni atomiche:
  - AC $\equiv$ Alberto va al cinema
  - BC $\equiv$ Bruno va al cinema
  - AT $\equiv$ Alberto va al teatro
  - CT $\equiv$ Carlo va al teatro
- Formula proposizionale: $(\mathbf{AC} \wedge \mathbf{BC}) \vee (\mathbf{AT} \wedge \mathbf{CT})$
- Tuttavia, "Alberto", "Bruno", ... "cinema" ..., gli individui del nostro discorso e le relazioni tra di essi ("andare al") scompaiono...

Limiti del Calcolo Proposizionale (2)

Le formule proposizionali possono descrivere **relazioni logiche tra un numero finito di enunciati**, ma

- Vorremmo esprimere proprietà di un' **infinità di individui**:
  - "tutti i numeri pari maggiori di due non sono primi"
    In CP(?) ("4 non è primo") $\wedge$ ("6 non è primo") $\wedge$ ... **NO!**
  - "esiste almeno un numero naturale maggiore di due che non è primo"
    In CP(?) ("3 non è primo") $\vee$ ("4 non è primo") $\vee$ ... **NO!**
- Vorremmo poter esprimere **proprietà "generali"** come
  "se x è pari allora x+1 è dispari"
  ... e riconoscere che da esse derivano proprietà specifiche come
  "se 4 è pari allora 5 è dispari"

Limiti del Calcolo Proposizionale (3)

Anche se descriviamo proprietà di un numero finito di enunciati vorremmo descriverli in maniera **compatta**

- **"tutti gli studenti di LPP vanno al cinema"**
  In CP(?) $S_1 \wedge S_2 \wedge S_3 \wedge S_4 \ldots \wedge S_{150}$
- **"tutti gli studenti di LPP tranne uno vanno al cinema"**
  In CP(?)

$$\begin{gathered}
(\neg S_1 \wedge S_2 \wedge S_3 \wedge S_4 \ldots \wedge S_{150}) \vee \\
(S_1 \wedge \neg S_2 \wedge S_3 \wedge S_4 \ldots \wedge S_{150}) \vee \\
(S_1 \wedge S_2 \wedge \neg S_3 \wedge S_4 \ldots \wedge S_{150}) \vee \\
(S_1 \wedge S_2 \wedge S_3 \wedge \neg S_4 \ldots \wedge S_{150}) \vee \\
\ldots \\
(S_1 \wedge S_2 \wedge S_3 \wedge S_4 \ldots \wedge \neg S_{150})
\end{gathered}$$

Verso la Logica del Primo Ordine

- La **Logica del Primo Ordine** (LPO) estende (include) il **Calcolo Proposizionale**
- Con le formule di LPO
  - si possono denotare/rappresentare esplicitamente gli elementi del **dominio di interesse** (gli individui, usando i **termini**)
  - si possono esprimere **proprietà** di individui e **relazioni** tra due o più individui (usando i **predicati**)
  - si può **quantificare** una formula, dicendo che vale per almeno un individuo, o per tutti gli individui (usando i **quantificatori**)

La Logica del Primo Ordine

- Presenteremo **Sintassi**, **Semantica** e **Sistema di Dimostrazioni**
- Useremo **formule** della LPO per formalizzare enunciati dichiarativi
- La **semantica** di una formula di LPO è sempre un valore booleano assegnato in base ad una **interpretazione**, come per il Calcolo Proposizionale, ma determinato in modo molto più complesso...
- Come per il Calcolo Proposizionale, ci interessano le formule che sono "sempre vere" (**formule valide** analoghe alle **tautologie**)
- **Non esistono tabelle di verità**: per vedere se una formula è valida occorre dimostrarlo (e non sempre è possibile trovare una dimostrazione)

Espressività della Logica del Primo Ordine

Esempi (li analizzeremo meglio in seguito):

- Tutti i numeri pari maggiori di due non sono primi:

$$(\forall x \,.\, pari(x) \wedge (x > 2) \Rightarrow \neg primo(x))$$

- Esiste almeno un numero naturale maggiore di due che non è primo

$$(\exists x \,.\, (x > 2) \wedge \neg primo(x))$$

- Se x è pari allora x+1 è dispari (*) $(\forall x.pari(x) \Rightarrow dispari(x+1))$
- (*) implica "se 4 è pari allora 5 è dispari"

$$(\forall x.pari(x) \Rightarrow dispari(x+1)) \Rightarrow (pari(4) \Rightarrow dispari(5))$$

Espressività della Logica del Primo Ordine (2)

- Tutti gli studenti di LPP vanno al cinema:

$$(\forall x.studente(x) \wedge frequenta(x, LPP) \Rightarrow vaAlCinema(x))$$

- Tutti gli studenti di LPP tranne uno vanno al cinema:

$$\Big(\exists x.studente(x) \wedge frequenta(x, LPP) \wedge \neg vaAlCinema(x) \wedge$$

$$(\forall y.studente(y) \wedge frequenta(y, LPP) \wedge \neg(y = x) \Rightarrow vaAlCinema(y))\Big)$$

Logica del Primo Ordine: commenti

Nella Logica del Primo Ordine ci sono due categorie sintattiche e semantiche differenti

- i **termini** che sono passati come argomenti dei predicati e denotano elementi del dominio di riferimento
- le **formule** costruite a partire dai predicati e che assumono valore booleano

La Sintassi della Logica del Primo Ordine: l'Alfabeto

Un **alfabeto** del primo ordine comprende:

- Un insieme $\mathcal{V}$ di **simboli di variabile**
- Un insieme $\mathcal{C}$ di **simboli di costante**
- Un insieme $\mathcal{F}$ di **simboli di funzione**, ognuno con la sua **arietà** (o **numero di argomenti**)
- Un insieme $\mathcal{P}$ di **simboli di predicato**, ognuno con la sua **arietà** (eventualmente 0)
- I simboli $\neg, \wedge, \vee, \Rightarrow, \equiv, \ \Leftarrow$ (**connettivi logici**)
- I simboli $\forall, \exists$ (**quantificatori**)
- I simboli ( ) [parentesi] , [virgola] . [punto]

La Sintassi della Logica del Primo Ordine: la Grammatica

Estende la grammatica del Calcolo Proposizionale con nuove produzioni

$$
\begin{array}{lll}
\mathit{Fbf} & ::= & (\mathit{Fbf} \equiv \mathit{Fbf}) \mid (\mathit{Fbf} \wedge \mathit{Fbf}) \mid (\mathit{Fbf} \vee \mathit{Fbf}) \mid \\
 & & (\mathit{Fbf} \Rightarrow \mathit{Fbf}) \mid (\mathit{Fbf} \Leftarrow \mathit{Fbf}) \mid (\neg \mathit{Fbf}) \mid \\
 & & \mathbf{T} \mid \mathbf{F} \mid \mathit{Pred} \mid \\
 & & (\forall \mathit{Var}.\mathit{Fbf}) \mid (\exists \mathit{Var}.\mathit{Fbf}) \\
\mathit{Pred} & := & \mathit{PIde} \mid \mathit{PIde}(\mathit{Term}\{, \mathit{Term}\}) \\
\mathit{Term} & ::= & \mathit{Const} \mid \mathit{Var} \mid \mathit{FIde}(\mathit{Term}\{, \mathit{Term}\})
\end{array}
$$

dove

- $\mathit{Var} \in \mathcal{V}$ è un **simbolo di variabile**,
- $\mathit{Const} \in \mathcal{C}$ è un **simbolo di costante**,
- $\mathit{FIde} \in \mathcal{F}$ è un **simbolo di funzione**, e
- $\mathit{PIde} \in \mathcal{P}$ è un **simbolo di predicato**.

Sintassi della Logica del Primo Ordine: i Termini

I **termini** denotano "elementi del dominio di interesse" ("individui").
Sono definiti dalla categoria sintattica *Term*:

$$Term \quad ::= \quad Const \mid Var \mid FIde(Term\{, Term\})$$

- Quindi i termini sono definiti induttivamente, a partire da un alfabeto $\mathcal{V}$, $\mathcal{C}$, $\mathcal{F}$ e $\mathcal{P}$, nel modo seguente:
  - Ogni costante $c \in \mathcal{C}$ è un termine
  - Ogni variabile $x \in \mathcal{V}$ è un termine
  - Se $f \in \mathcal{F}$ è un simbolo di funzione con arietà n e $t_1, \ldots, t_n$ sono termini, allora $f(t_1, \ldots, t_n)$ è un termine

Termini: Esempi

- a, con $a \in \mathcal{C}$
- x, con $x \in \mathcal{V}$
- $g(a)$, con $g \in \mathcal{F}$ di arietà 1
- $f(x, g(a))$, con $f \in \mathcal{F}$ di arietà 2
- **Notazione**: I ben noti simboli di funzione binari a volte sono rappresentati con notazione **infissa** invece che **prefissa**
- Esempio: Siano dati $+, * \in \mathcal{F}$ con arietà 2:
  - allora $\boxed{x + (1 * z)}$ è un termine, in notazione infissa
  - Usando la notazione prefissa, lo stesso termine sarebbe $\boxed{+(x, *(1, z))}$

Sintassi della Logica del Primo Ordine: le Formule (1)

Le **formule** rappresentano enunciati dichiarativi. Sono definite induttivamente come segue, fissato un alfabeto $\mathcal{V}$, $\mathcal{C}$, $\mathcal{F}$ e $\mathcal{P}$:

- Se $p \in \mathcal{P}$ è un simbolo di predicato allora
  - se ha arietà $n > 0$ e $t_1, \ldots, t_n$ sono termini allora $p(t_1, \ldots, t_n)$ è una formula.
  - se ha arietà 0 allora p è una formula. Corrisponde a una variabile proposizionale nel CP, e lo scriviamo p invece di $p()$

  Queste sono le **formule atomiche**, corrispondenti alla categoria sintattica *Pred*:

$$Pred \quad := \quad PIde \mid PIde(Term\{, Term\})$$

- A volte usiamo notazione infissa per simboli di arietà 2.
- Esempio: $x{=}y$ o $z{\leq}f(x)$ con $=, \leq, \in \mathcal{P}$ con arietà 2

Sintassi della Logica del Primo Ordine: le Formule (2)

La rimanente categoria sintattica è

$$\begin{array}{lll} \mathit{Fbf} & ::= & (\mathit{Fbf} \equiv \mathit{Fbf}) \mid (\mathit{Fbf} \wedge \mathit{Fbf}) \mid (\mathit{Fbf} \vee \mathit{Fbf}) \mid \\ & & (\mathit{Fbf} \Rightarrow \mathit{Fbf}) \mid (\mathit{Fbf} \Leftarrow \mathit{Fbf}) \mid (\neg \mathit{Fbf}) \mid \\ & & \mathbf{T} \mid \mathbf{F} \mid \mathit{Pred} \mid \\ & & (\forall \mathit{Var}.\mathit{Fbf}) \mid (\exists \mathit{Var}.\mathit{Fbf}) \end{array}$$

e definiscono le formule induttivamente come segue

- **T** e **F** sono formule
- Ogni formula atomica della categoria sintattica *Pred* è una formula
- Se P è una formula allora $\neg P$ è una formula
- Se P e Q sono formule allora $P \wedge Q, P \vee Q, P \Rightarrow Q, P \equiv Q, P \Leftarrow Q$ sono formule
- Se P è una formula e $x \in \mathcal{V}$, allora $(\forall x.P)$ e $(\exists x.P)$

Sintassi delle Formule: Esempi

- Tutti i numeri pari maggiori di due non sono primi

$$(\forall x. \mathit{pari}(x) \wedge x > 2 \Rightarrow \neg \mathit{primo}(x))$$

- Se x è pari allora il successore di x è dispari

$$(\forall x. \mathit{pari}(x) \Rightarrow \mathit{dispari}(x+1))$$

- "Se x è pari allora $x + 1$ è dispari" implica "se 4 è pari allora 5 è dispari"

$$(\forall x. \mathit{pari}(x) \Rightarrow \mathit{dispari}(x+1)) \Rightarrow (\mathit{pari}(4) \Rightarrow \mathit{dispari}(5))$$

Simboli di Variabile? Simboli di Predicato? Simboli di Funzione? Simboli di Costante?

Occorrenze di Variabili Libere o Legate

- In una formula quantificata come $(\forall x.P)$ o $(\exists y.P)$ la sottoformula P è detta la **portata** del quantificatore.
- Una occorrenza di variabile x è **legata** se compare nella portata di un quantificatore $\forall x$ o $\exists x$, altrimenti è detta **libera**.
- Esempio:

$$(\forall y \,.\, z = y \wedge (x = y \vee (\exists x \,.\, x = z \vee z = y)))$$

  - Portata di $\forall y$?
  - Portata di $\exists x$?
  - Occorrenze di variabili legate?
  - Occorrenze di variabili libere?

Formule Aperte e Chiuse

- Il nome di una variabile legata può essere cambiato grazie alle leggi di **ridenominazione**:

  $$(\forall x.P) \equiv (\forall y.P[y/x]) \quad \text{se } y \text{ non compare in } P \quad (\textit{Ridenom.})$$
  $$(\exists x.P) \equiv (\exists y.P[y/x]) \quad \text{se } y \text{ non compare in } P \quad (\textit{Ridenom.})$$

  Attenzione: qui $P[y/x]$ rappresenta la formula P in cui TUTTE le occorrenze di x sono sostituite da y
- Una formula che contiene occorrenze di variabili libere è detta **aperta**
- Spesso scriveremo $P(x)$ per indicare che x è libera nella formula P
- Una formula senza variabili libere è detta **chiusa**. Considereremo principalmente formule chiuse.

Due esempi da Analisi

- Si definisce il predicato *debolmente crescente* come segue

$$DC(f) \equiv \forall x.(\forall y.(x \leq y \Rightarrow f(x) \leq f(y)))$$

- Si definisce il predicato $lim_{x \to x_0} f(x) = c$ come segue

$$\forall \epsilon > 0.(\exists \delta > 0.(\forall x.(|x - x_0| < \delta \wedge \neg(x = x_0) \Rightarrow |f(x) - c| < \epsilon)))$$

- In realtà la formula precedente non appartiene alla sintassi della logica del primo ordine, a causa della presenza di $\forall \epsilon > 0.$ e $\exists \delta > 0.$
- La formulazione corretta è la seguente

$$Lim(f, x_0, c) \equiv$$

$$\forall \epsilon.(\epsilon > 0 \Rightarrow \exists \delta.(\delta > 0 \wedge \forall x.(|x - x_0| < \delta \wedge \neg(x = x_0) \Rightarrow |f(x) - c| < \epsilon)))$$