

Appunti per il corso di Logica (Prima Parte del corso di Metodi Formali e Logica)

Giovanna D'Agostino

PROGRAMMA

- PRIMA PARTE: Espressività dei formalismi logici, con particolare attenzione alle applicazioni.
- SECONDA PARTE:
 1. INFORMATICA: Calcoli Logici (Risoluzione, Calcolo dei Sequenti).
 2. MATEMATICA: approfondimenti.

1 BREVE PANORAMICA DELLA PRIMA PARTE DEL CORSO

Data una classe di strutture, possiamo considerare proprietà relative a questa classe.

Esempio 1. Sia $\mathcal{L} = \{r, R, P_1, P_2, \dots\}$ il linguaggio dei grafi diretti etichettati. Nella classe delle strutture per $\mathcal{L}$ possiamo considerare le proprietà seguenti (le proprietà binarie sono relative alla relazione di adiacenza del grafo, quelle unarie sono relative alle etichette).

1. riflessività, transitività, simmetria;
2. planarità;
3. dal nodo r parte un cammino di tre passi che porta ad un nodo etichettato P_1 ;
4. dal nodo r parte un cammino infinito;
5. dal nodo r parte un cammino infinito dove vale infinite volte P_1 ;
6. il grafo è un albero;

7. etc

Il linguaggio descritto è particolarmente interessante per le applicazioni informatiche ed in particolare per la verifica dei programmi.

Esempio 2. *Consideriamo il seguente programma:*

1. $v := 0, w := 1;$
2. *while* $w \neq 0$ *do* $v := v + w \pmod{4}, w := w + 2 \pmod{3};$
3. *while* $w < 3$ *do* $w := w + v, v := v + v \pmod{5};$
4. *goto* 2;

Possiamo rappresentare le possibili computazioni del programma con il grafo etichettato

$$\{v = 0, w = 1\} \rightarrow \{v = 1, w = 0\} \rightarrow \{v = 2, w = 1\} \rightarrow \{v = 4, w = 3\} \rightarrow \{v = 3, w = 2\} \rightarrow \dots$$

Le proprietà atomiche possono essere, ad esempio:

$$P_1 := v_1 + v_2 > 0, \quad P_2 := v_1 \times v_2 > 0.$$

Possiamo chiederci se queste proprietà valgono nei primi stati, se da un certo punto in poi non varranno più etc.

Nella verifica dei programmi, una volta modellate tutte le possibili computazioni del programma con un grafo etichettato, si cerca di formalizzare le proprietà da verificare tramite una formula di un determinato linguaggio formale. A quel punto la verifica della proprietà diviene equivalente alla verità della formula sul grafo.

Rimane il problema di stabilire quale sia il linguaggio formale più adatto ad esprimere una (data classe di) proprietà. Tale linguaggio deve essere il più possibile espressivo e al contempo mantenere buone proprietà di complessità, nel senso che verificare la verità di una formula in una data struttura non deve essere troppo dispendioso.

Logiche note e meno note:

1. Logica Proposizionale: $P_1 \wedge P_2, P_1 \vee P_2$. Esprime proprietà locali degli stati.
2. Logica al Prim'ordine $\forall x P_1(x), \forall x (P_1(x) \rightarrow P_2(x)), \exists x (x R y \wedge P_1(x) \wedge P_2(x))$.
Sembra in grado di esprimere proprietà più globali delle strutture.
3. Logica Modale: $\Diamond P_1 \wedge \neg \Box (P_2)$ è a metà fra le due precedenti
4. Logica Monadica al Second'ordine: rispetto alla logica al prim'ordine, permette la quantificazione su sottoinsiemi del dominio:

$$\exists X (\exists x X(x) \wedge \forall y (X(y) \rightarrow \exists z (X(z) \wedge R(y, z)))).$$

5. μ -Calcolo : logica Modale più punti fissi di operatori monotoni: $\mu X \Diamond (X \vee P_1)$.

Nel corso introdurremo delle tecniche per stabilire se, data una Logica e una proprietà, esiste o meno una formula della logica che esprima la proprietà. Per quanto riguarda la logica al prim'ordine, se siamo interessati (come spesso avviene nelle applicazioni) ai modelli finiti, uno strumento adatto è quello della teoria dei giochi. Introdurremo brevemente la teoria dei giochi a due a somma zero con informazione perfetta accennando a qualche risultato generale su questi giochi. Considereremo poi i Giochi di Ehrenfeucht-Fraïssé per la logica al prim'ordine e la corrispondenza con il problema dell'espressività.

Tornando alla verifica dei Programmi, studieremo delle logiche adatte a tale verifica, partendo dalla debole Logica Modale, fino ad arrivare alla più espressiva Logica di Punto fisso nota come μ -Calcolo. Nel far questo introdurremo il concetto di bisimulazione fra grafi e lo utilizzeremo per contraddistinguere la logica Modale ed il μ -Calcolo rispetto alle Logiche Classiche (al prim'ordine e al Second'Ordine). Termineremo questa parte del programma con l'introduzione di nuovi Giochi logici, i giochi di Parità, e ne discuteremo i legami con il μ -calcolo.