LOGICA MATEMATICA PER INFORMATICA

A.A. 10/11, SETTIMANA N. 1

SOMMARIO. Introduciamo il linguaggio e la sintassi e la semantica della Logica del I Ordine. Introduciamo i concetti di teoria, teoria completa, teoria decidibile. Dimostriamo il Lemma di Craig.

1. LINGUAGGIO, TERMINI, FORMULE

Linguaggio $\mathcal{L}$. Una collezione (finita o infinita) di simboli. I simboli sono di tre tipi.

- Simboli di relazioni, ciascuno con la sua molteplicità.
- Simboli di funzioni, ciascuno con la sua molteplicità.
- Costanti.

Inoltre assumiamo sempre un insieme numerabile di variabili $v_1, v_2, \ldots$. Costanti e funzioni si possono combinare per costruire nomi più complessi per elementi del dominio.

Definizione 1.1 (Termini)**.** I termini sono ottenuti partendo dalle variabili e dalle costanti e chiudendo sotto applicazione di simboli di funzione. Un termine che non contiene variabili è un termine chiuso.

Logica su $\mathcal{L}$. Per creare enunciati nel linguaggio $\mathcal{L}$ usiamo i simboli logici

- Connettivi $\wedge$, $\vee$, $\rightarrow$, $\neg$.
- Quantificatori $\exists$, $\forall$.
- Il simbolo di identità $=$.

Definizione 1.2 (Formule Atomiche)**.** Una formula atomica è una formula del tipo $t = s$ dove t, s sono termini, o $R(t_1, \ldots, t_k)$ dove R è un simbolo di relazione di dimensione k e $t_1, \ldots, t_k$ sono termini.

Definizione 1.3 (Formule)**.** Le formule sono ottenute partendo dalle formule atomiche e chiudendo sotto connettivi proposizionali e quantificatori universali ed esistenziali. Le formule (non atomiche) sono dunque del tipo

$$(F \wedge G), (F \vee G), (\neg F), (F \rightarrow G), ((\forall v)F), ((\exists v)F),$$

dove F e G sono formule (atomiche o non atomiche) e v è una variabile.

Nelle formule $((\forall v)F)$ e $((\exists v)F)$, F è detto il *dominio* (o *scope*) del quantificatore. Se v non occorre in F possiamo identificare le due formule con F.

Definizione 1.4 (Variabili libere e legate)**.** Associamo a ogni termine e a ogni formula l'insieme delle sue variabili libere e legate. Il termine c non ha variabili. Il termine v ha v come unica variabile libera e non ha variabili legate. Le variabili libere di $f(t_1, \ldots, t_k)$ sono l'unione delle variabili libere dei t_i, analogamente per le legate. Per i connettivi proposizionali è ovvio. Le variabili libere di $\forall v F$ sono le variabili libere di F meno v, e le variabili legate sono le variabili legate di F più v. Analogamente per $\exists v F$.

Note preparate da Lorenzo Carlucci, carlucci@di.uniroma1.it.

Un *enunciato* è una formula senza variabili libere. Un termine t è *libero per* una variabile v in una formula F se nessuna occorrenza libera di v in F è nel dominio di un quantificatore $\forall y$ o $\exists y$ con y una variabile in t. Il succo è che se sostituiamo t per x in $F(x)$ nessuna occorrenza di una variabile in t diventa vincolata in $F(t)$.

Se F è una formula e $x_1, \ldots, x_n$ sono variabili *distinte*, indichiamo con $F(x_1, \ldots, x_n)$ il fatto che le variabili libere di F sono *contenute* nell'insieme $\{x_1, \ldots, x_n\}$. Analogamente per un termine.

Definizione 1.5 (Sostituzione in termini)**.** Siano $v_1, \ldots, v_k$ variabili distinte. Sia $(t_1, \ldots, t_k)$ una sequenza di termini. Sia s un termine. Definiamo la sostituzione simultanea delle variabili $\vec{v}$ con i termini $\vec{t}$ in s.

- Se s è una costante allora $s[v_1, \ldots, v_k/t_1, \ldots, t_k]$ è s.
- Se s è una variabile diversa da $v_1, \ldots, v_k$ allora $s[v_1, \ldots, v_k/t_1, \ldots, t_k]$ è s.
- Se s è la variabile v_i, $i \in [1, k]$ allora $s[v_1, \ldots, v_k/t_1, \ldots, t_k]$ è t_i.
- Se s è $f(s_1, \ldots, s_\ell)$ dove f è un simbolo di funzione di dimensione ℓ e $s_1, \ldots, s_\ell$ sono termini, allora $s[v_1, \ldots, v_k/t_1, \ldots, t_k]$ è $f(s_1[v_1, \ldots, v_k/t_1, \ldots, t_k], \ldots, s_\ell[v_1, \ldots, v_k/t_1, \ldots, t_k])$.

Definizione 1.6 (Sostituzione in formule)**.** Siano $v_1, \ldots, v_k$ variabili distinte. Sia F una formula le cui variabili libere sono tutte tra $v_1, \ldots, v_k$. Sia $(t_1, \ldots, t_k)$ una sequenza di termini chiusi. Definiamo la sostituzione simultanea delle variabili $\vec{v}$ con i termini $\vec{t}$ in una formula F. I casi proposizionali sono banali.

- Se F è del tipo $\forall v G$, dove G ha una variabile libera v e v non è tra $v_1, \ldots, v_k$, allora $F[v_1, \ldots, v_k/a_1, \ldots, a_k]$ è definita come $\forall v G[v_1, \ldots, v_k/a_1, \ldots, a_k]$.
- Se F è del tipo $\forall v_i G$, dove G ha una variabile libera v_i e $i \in [1, k]$, allora $F[v_1, \ldots, v_k/a_1, \ldots, a_k]$ è definita come $\forall v_i G[v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k/t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_k]$.

2. Strutture, soddisfazione, verità, validità

Fissiamo un linguaggio $\mathcal{L} = \{R_i, f_j, c_k \ : \ i \in I, j \in J, k \in K\}$. Una struttura $\mathfrak{A}$ per il linguaggio $\mathcal{L}$ consiste di

- Un insieme A, detto dominio.
- Per ogni R_i di dimensione k, una relazione di dimensione k su A, che denotiamo con $R_i^{\mathfrak{A}}$.
- Per ogni f_j di dimensione k, una funzione a k argomenti su A, che denotiamo con $f_j^{\mathfrak{A}}$.
- Per ogni $k \in K$, un elemento di A, che denotiamo con $c_k^{\mathfrak{A}}$.

Si ricorda che una relazione di dimensione k su un insieme A è un insieme di sequenze ordinate di dimensione k di elementi di A, ossia un sottinsieme del prodotto cartesiano A^k (l'insieme di tutte le k-ple ordinate di elementi di A).

Definiamo la relazione di validità di una formula F in una struttura $\mathfrak{A}$, che denotiamo con $\mathfrak{A} \models F$.

Un *assegnamento* α in $\mathfrak{A}$ è una mappa che associa ad ogni variabile un elemento di A, i.e.,

$$\alpha : \{v_n \ : \ n \in \mathbf{N}\} \longrightarrow A$$

Un assegnamento si estende ai termini ponendo $\alpha(c)$ uguale a $c^{\mathfrak{A}}$ e $\alpha(f(t_1, \ldots, t_k))$ uguale a $f^{\mathfrak{A}}(\alpha(t_1), \ldots, \alpha(t_k))$. Indichiamo con $\alpha\frac{a}{x}$ l'assegnamento che differisce da α solo perché associa a a x.

Definizione 2.1 (Soddisfazione)**.** Definiamo la relazione $\mathfrak{A} \models F[\alpha]$ come segue.

$\mathfrak{A} \models (t = s)[\alpha]$ se e solo se $\alpha(t)$ è uguale $\alpha(s)$.
$\mathfrak{A} \models R(t_1, \ldots, t_k)[\alpha]$ se e solo se $(\alpha(t_1), \ldots, \alpha(t_k)) \in R^{\mathfrak{A}}$.
$\mathfrak{A} \models \neg G[\alpha]$ se e solo se non vale $\mathfrak{A} \models G[\alpha]$.
$\mathfrak{A} \models \exists v G[\alpha]$ se e solo se esiste $a \in A$ tale che $\mathfrak{A} \models G[\alpha\frac{v}{a}]$.

$\mathfrak{A} \models \forall v G[\alpha]$ se e solo se per ogni $a \in A$ vale $\mathfrak{A} \models G[\alpha\frac{v}{a}]$.

Il fatto che valga $\mathfrak{A} \models F[\alpha]$ o no dipende soltanto dai valori di α sulle variabili libere che appaiono in F. Pertanto, se queste sono incluse in $\{x_1, \ldots, x_n\}$, possiamo scrivere $\mathfrak{A} \models F[\alpha(x_1), \ldots, \alpha(x_n)]$ indicando esplicitamente gli elementi assegnati alle variabili che contano. Osserviamo anche che se F è un enunciato, allora $\mathfrak{A} \models F[\alpha]$ vale per tutti gli assegnamenti o per nessuno! Se $\mathfrak{A} \models F[\alpha]$ per qualche assegnamento α, diciamo che α *soddisfa* l'enunciato F in $\mathfrak{A}$, e in tal caso F è detta *soddisfacibile*. Diciamo che una formula $F(x_1, \ldots, x_n)$ è *vera* in una struttura se è soddisfatta da tutti gli assegnamenti su quella struttura. Questo equivale a dire che l'enunciato $\forall x_1 \ldots \forall x_n F(x_1, \ldots, x_n)$ (dove $x_1, \ldots, x_n$ sono tutte e sole le variabili libere di F) è vero nella struttura. Una formula è *(logicamente) valida* se è vera in tutte le strutture, ossia se è soddisfatta da tutte le strutture e per tutti gli assegnamenti. Si osserva che F è valida se e solo se $\neg F$ non è soddisfacibile. Diciamo che $\mathfrak{A} \models X[\alpha]$ se per ogni $F \in X$, $\mathfrak{A} \models F[\alpha]$. Diciamo che $\mathfrak{A} \models X$ se per ogni α vale $\mathfrak{A} \models X[\alpha]$. Diciamo che F è una *conseguenza (logica)* di X se per ogni $\mathfrak{A}$, per ogni assegnamento α in $\mathfrak{A}$, se $\mathfrak{A} \models X[\alpha]$ allora $\mathfrak{A} \models F[\alpha]$. In tal caso scriviamo $X \models F$. Due formule F, F' sono *logicamente equivalenti* se è valida la doppia implicazione $F \leftrightarrow F'$ (che è equivalente a $(F \to F') \wedge (F' \to F)$). Questo equivale a dire che F è una conseguenza di F' e che F' è una conseguenza di F.

3. Teorie, Decidibilità e Completezza

Una *teoria* in un linguaggio $\mathcal{L}$ è un insieme T di enunciati di $\mathcal{L}$. Un *modello* di una teoria T è una struttura per $\mathcal{L}$ che soddisfa tutti gli elementi di T.

Una teoria è *soddisfacibile* se ha un modello. Una teoria T è *completa* se, per ogni enunciato E nel linguaggio della teoria, o E vale in tutti i modelli di T oppure $\neg E$ vale in tutti i modelli di T, i.e.,

$$T \models E \text{ oppure } T \models \neg E.$$

Una teoria T è *decidibile* se esiste un algoritmo di decisione per il seguente problema:

Dato un enunciato E in $\mathcal{L}$, decidere se E vale in tutti i modelli di T.

Nota Bene: dato che abbiamo definito una teoria come un insieme di enunciati, l'espressione T è decidibile può anche significare che esiste un algoritmo per decidere l'appartenenza di un enunciato all'insieme di enunciati T. Manteniamo l'ambiguità ma chiariamo di volta in volta quale significato usiamo.

David Hilbert propose all'inizio del XX secolo il seguente problema come *il problema principale della Logica Matematica*. **Problema della Decisione (PD)**: *Esiste un algoritmo per il seguente problema?*

(Validità) Dato un enunciato E del I ordine, decidere se E è logicamente valido.

Dato che E è valido se e solo se $\neg E$ non è soddisfacibile, si osserva che il problema è anche equivalente al seguente.

(Soddisfacibilità) Dato un enunciato E del I ordine, decidere se E ha un modello.

Per capire l'importanza del **PD** (!), dobbiamo citare il Teorema di Completezza (che dimostreremo). Per il Teorema di Completezza l'insieme degli E che valgono in tutti i modelli di T coincide con l'insieme degli E che sono formalmente dimostrabili da premesse in T con le regole di deduzione del calcolo dei predicati del I ordine (chiamiamo questi enunciati i *teoremi* di T). Per tali enunciati esiste una successione finita di passi di ragionamento formali che portano da certe premesse scelte in T (in numero finito) alla conclusione E.

Dal Teorema di Completezza segue che un algoritmo per il problema della Validità è anche un algoritmo per il problema seguente

(Dimostrabilità) Dato un enunciato E del I ordine, decidere se E è dimostrabile dagli assiomi della logica del I ordine.

Perché il **PD** era così importante per Hilbert? Perché una soluzione positiva al **PD** darebbe un metodo uniforme per meccanizzare l'attività matematica.

L'attività del matematico può in generale descriversi così: scelto un insieme di enunciati, $\mathcal{S}$ come assiomi, ci interessano gli enunciati che possiamo dimostrare a partire dagli assiomi scelti. In altre parole ci interessa l'insieme degli enunciati che sono veri in tutte le strutture che soddisfano gli assiomi, ossia l'insieme delle conseguenze logiche degli assiomi. Ancora per il Teorema di Completezza, questo insieme coincide con l'insieme degli enunciati (formalmente) dimostrabili a partire dagli assiomi scelti.

Se $\mathcal{S}$ è un insieme finito, sia $\{S_1, \ldots, S_t\}$, allora, per ogni enunciato E, vale che

$$E \text{ è dimostrabile dagli assiomi } \mathcal{S} \text{ se e solo se } (S_1 \wedge \cdots \wedge S_t \to E) \text{ è una formula valida.}$$

Per questa equivalenza serve il cosidetto Teorema di Deduzione, che dice che se B si dimostra assumendo A, allora $A \to B$ è valida.

Se abbiamo un algoritmo per il Problema della Validità, allora abbiamo un algoritmo per decidere se un enunciato E è dimostrabile dagli assiomi $\mathcal{S}$. Questo algoritmo è inoltre uniforme, nel senso che non dipende dal tipo di matematica che è formalizzata in $\mathcal{S}$!

Purtroppo (?), il Problema della Decisione non ha una soluzione algoritmica, i.e., il Problema della Validità, il Problema della Soddisfacibilità e il Problema della Dimostrabilità sono indecidibili.

Non sempre è possibile individuare un numero finito di assiomi per catturare una teoria matematica. In alcuni casi è necessario usare un numero infinito di assiomi. Una richiesta ragionevole è che deve essere possibile riconoscere algoritmicamente se una certa formula è un assioma o no. In altre parole, richiedere che l'insieme degli assiomi sia decidibile.

Quando si studiano gli enunciati veri in una particolare struttura matematica, come per esempio i naturali (con la loro struttura moltiplicativa e additiva) o i reali (con la loro struttura di campo), è naturale sperare di poter formulare una teoria completa, ossia un insieme $\mathcal{S}$ di enunciati (assiomi) tali che, per ogni enunciato E, o E è dimostrabile da $\mathcal{S}$ oppure $\neg E$ è dimostrabile da $\mathcal{S}$.

La completezza e la decidibilità di una teoria sono strettamente legate. Sia T un insieme di enunciati.

Osservazione 3.1. Se l'insieme delle conseguenze di T è algoritmicamente enumerabile, completa, e non contraddittoria, allora è decidibile. L'algoritmo di decisione enumera le conseguenze di T. Dato che T è completa, nell'enumerazione apparirà o E o $\neg E$. Dato che T è non contraddittoria, se appare $\neg E$ significa che E non è un teorema.

Ricordiamo che un insieme è algoritmicamente enumerabile se esiste un programma che produce una lista di tutti e soli gli elementi dell'insieme. Intuitivamente un insieme è enumerabile se esiste un algoritmo in grado di riconoscere tutti e soli gli elementi dell'insieme, ma che diverge sui non-elementi. Si dice che un tale insieme ha una procedura di decisione parziale. Più tecnicamente, un insieme è algoritmicamente enumerabile se è il dominio di una funzione calcolabile. Si ricorda che esistono insiemi algoritmicamente enumerabili ma non ricorsivi. E.g., supponendo fissata una enumerazione di tutti i programmi (in un certo sistema di programmazione universale) l'insieme degli n tali che il programma numero n su input n termina è algoritmicamente enumerabile ma non decidibile (indecidibilità del Problema della Fermata).

Come facciamo ad assicurarci che l'insieme dei teoremi di T è algoritmicamente enumerabile?

Osservazione 3.2. Se T è finita, allora l'insieme dei teoremi di T è algoritmicamente enumerabile. Possiamo applicare meccanicamente regole di inferenza agli elementi di T per ottenere una lista dei teoremi.

Osservazione 3.3. Se l'appartenenza in T è decidibile, allora l'insieme dei teoremi di T è algoritmicamente enumerabile. Se T è un insieme decidibile (i.e., esiste un algoritmo che, data una stringa di simboli, decide se è in T o no) allora si può decidere algoritmicamente se una stringa è una dimostrazione con premesse in T. Si possono allora enumerare le dimostrazioni formali con premesse in T e per ciascuna identificare la conclusione. Questi sono tutti e soli i teoremi di T.

Cosa succede se T è un insieme soltanto algoritmicamente enumerabile (ma non necessariamente decidibile)? Si dimostra che anche in questo caso, l'insieme dei teoremi è algoritmicamente enumerabile.

La dimostrazione è istruttiva.

Proposizione 3.4 (Lemma di Craig)**.** *Sia $\mathcal{S}$ un insieme algoritmicamente enumerabile di enunciati. Esiste un insieme decidibile di enunciati $\mathcal{S}'$ tale che l'insieme delle conseguenze di $\mathcal{S}$ coincide con l'insieme delle conseguenze di $\mathcal{S}'$.*

Dimostrazione. Per ipotesi $\mathcal{S}$ è algoritmicamente enumerabile. Fissiamo un programma che enumera $\mathcal{S}$. Questo programma produce, su input $i \in \mathbf{N}$, un enunciato S_i in $\mathcal{S}$ e

$$S_1, S_2, \ldots, S_i, \ldots$$

è una enumerazione di tutti e soli gli enunciati in $\mathcal{S}$.

Definiamo $\mathcal{S}'$ come segue. Definiamo

$$S'_1 = S_1, S_2 = (S_2 \wedge S_2), S'_3 = (S_3 \wedge (S_3 \wedge S_3)), \ldots$$

In generale, S'_i è il risultato di una congiunzione iterata i volte dell'enunciato S_i.

L'insieme $\mathcal{S}'$ è l'insieme che cerchiamo!

Dimostriamo che le conseguenze di $\mathcal{S}'$ coincidono con quelle di $\mathcal{S}$. A tale fine basta dimostrare i due punti seguenti.

(a) Per ogni i, S_i è una conseguenza di $\mathcal{S}'$.
(b) Per ogni i, S'_i è una conseguenza di $\mathcal{S}$.

Per il punto (a), basta osservare che $S'_i = (S_i \wedge (S_i \wedge \ldots))$ è una conseguenza di $\mathcal{S}'$ e che $(A \wedge B) \to A$ è una verità logica.

Per il punto (b), basta osservare che S_i è una conseguenza di $\mathcal{S}$ e che $A \to (A \wedge A)$ è una verità logica e che $A, B \models (A \wedge B)$. Dunque S'_i è deducibile da S_i.

Resta da dimostrare che $\mathcal{S}'$ è decidibile. Data una formula F, come decidere se F è in $\mathcal{S}'$? Se F è S_1, siamo a posto. Altrimenti, verifichiamo se F è di forma $(G \wedge (G \wedge \ldots))$ per qualche formula G. Si vede chiaramente che questo controllo è algoritmico. Se F non è della forma desiderata, allora non è in $\mathcal{S}'$. Altrimenti, sia i il numero di volte che G è ripetuto in F. Allora F è in $\mathcal{S}'$ se e solo se G è uguale a S_i. Per verificare se questo è il caso, basta produrre l'i-esimo elemento nell'enumerazione di $\mathcal{S}$ e confrontarlo con G. □