

# METODI MATEMATICI PER L'INFORMATICA

ANNO ACCADEMICO 2011/2012

SOMMARIO. Sintassi e semantica della Logica dei predicati. Strutture, soddisfacibilità e verità logica. Proprietà fondamentali dei quantificatori.

## 1. SINTASSI DELLA LOGICA PREDICATIVA

Un linguaggio del I ordine è una collezione (finita o infinita)  $\mathcal{L}$  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  $v_1, v_2, \dots$ . Queste sono le variabili ufficiali del linguaggio. Useremo  $x, y, z, w, v$  (con pedici) come variabili su variabili (dette anche metavariabili, perché sono usate come variabili per variabili individuali). Costanti e simboli di funzione si possono combinare per costruire nomi più complessi per elementi del dominio.

**Esempio 1.1.** Un linguaggio adeguato per la teoria dei gruppi è il seguente  $\mathcal{L}_G = \{\cdot, e\}$ ; dove  $\cdot$  è un simbolo di funzione binaria per l'operazione di gruppo e  $e$  una costante per l'elemento neutro. Un linguaggio per l'aritmetica è il seguente  $\mathcal{L}_A = \{+, \times, <, 0, 1\}$ , dove  $+$ ,  $\times$  sono simboli di funzioni a due argomenti,  $<$  un simbolo di relazione binario e  $0, 1$  sono costanti. Un linguaggio per la teoria degli insiemi è il seguente  $\mathcal{L}_I = \{\in\}$  dove  $\in$  è un simbolo di relazione binario.

**Definizione 1.2** (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.

**Esempio 1.3.**  $((v_9 + 1) + 1) \times 0$  è un termine nel linguaggio  $\mathcal{L}_A$ .  $e, e \cdot e, e \cdot (e \cdot e), (e \cdot e) \cdot e$  sono termini chiusi nel linguaggio  $\mathcal{L}_G$ . I termini del linguaggio  $\mathcal{L}_I$  sono solo le variabili libere. N.B. Spesso usiamo la notazione infissa (i.e.,  $(x + y)$ ) invece di quella prefissa (i.e.,  $+(x, y)$ ).

Per formulare proposizioni nel linguaggio  $\mathcal{L}$  usiamo i simboli logici seguenti

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

In logica proposizionale le variabili proposizionali erano la minima unità dotata di un valore di verità (0,1). In logica predicativa il loro posto viene preso da espressioni più complesse, dette formule atomiche.

**Definizione 1.4** (Formule Atomiche). Una formula atomica è una formula del tipo  $t = s$  dove  $t, s$  sono termini, o  $R(t_1, \dots, t_k)$  dove  $R$  è un simbolo di relazione di dimensione  $k$  e  $t_1, \dots, t_k$  sono termini.

**Esempio 1.5.**  $0 < (v_1 + 1) \cdot ((v_4 + 1) + v_2)$  è una formula atomica in  $\mathcal{L}_A$ . Le uniche formule atomiche in  $\mathcal{L}_I$  sono di forma  $x \in y$  e  $x = y$  con  $x, y$  variabili.

**Definizione 1.6** (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.

---

Note preparate da Lorenzo Carlucci, [carlucci@di.uniroma1.it](mailto:carlucci@di.uniroma1.it).

Nelle formule  $((\forall v)F)$  e  $((\exists v)F)$ ,  $F$  è detto il *dominio* (o *scope*) del quantificatore e  $v$  la variabile quantificata. Se  $v$  non occorre in  $F$  possiamo identificare le due formule quantificate con  $F$ . Una distinzione fondamentale è quella tra variabili quantificate (dette vincolate o legate) e variabili non quantificate (dette libere).

**Definizione 1.7** (Variabili libere e legate). Una occorrenza di una variabile  $x$  in una formula  $F$  è vincolata se e solo se (i) l'occorrenza di  $x$  è la variabile quantificata di un quantificatore, oppure (ii) l'occorrenza di  $x$  è nel dominio di un quantificatore con variabile quantificata  $x$ . Tutte le altre occorrenze di  $x$  in  $F$  sono dette libere.

Ad ogni formula  $F$  possiamo associare in modo ovvio l'insieme delle sue variabili libere (le variabili che hanno almeno un'occorrenza libera in  $F$ ) e l'insieme delle sue variabili legate (le variabili che hanno almeno un'occorrenza vincolata in  $F$ ). Ovviamente i due insiemi non sono necessariamente disgiunti.

Esercizio: dare una definizione induttiva dell'insieme delle variabili libere e dell'insieme delle variabili vincolate di una formula.

Un *enunciato* è una formula senza variabili libere. In alcuni casi è utile sostituire una occorrenza di una variabile  $x$  in una formula  $F$  con un termine  $t$ . La prossima definizione indica in quali casi questa sostituzione è legittima. Il succo è che se sostituiamo  $t$  per  $x$  in  $F(x)$  nessuna occorrenza di una variabile in  $t$  diventa vincolata in  $F(t)$ .

**Definizione 1.8.** 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  $t$ .

**Esempio 1.9.** Se la formula  $F$  è  $\exists y(x = y + y)$ , allora nessun termine contenente  $y$  è libero per  $x$  in  $F$ .

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

## 2. SEMANTICA DELLA LOGICA PREDICATIVA

Il nostro scopo è quello di definire la nozione di *verità logica* per la logica dei predicati. Questa nozione è analoga a quella di tautologia per la logica proposizionale. A tale scopo dobbiamo definire la nozione di verità di una formula della logica predicativa. Nel caso della logica proposizionale la verità di una formula è fissata una volta che è fissato un assegnamento di valori di verità alle variabili proposizionali. Nel caso della logica predicativa la verità di una proposizione dipende dalla scelta dell'ambiente in cui decidiamo di interpretare i simboli del linguaggio. Un tale ambiente è detto *struttura*.

Si tratta di astrarre dalla normale pratica matematica. In Algebra è abituale definire un gruppo come un insieme  $G$  su cui è definita una operazione  $\cdot$  associativa e tale che esiste un elemento  $e$  di  $G$  che è neutro rispetto a  $\cdot$  e commuta con tutti gli elementi di  $G$ . In Teoria dei Grafi è abituale definire un grafo (semplice)  $G$  come una coppia  $(V, E)$ , dove  $V$  è un insieme (detto insieme dei vertici) e  $E$  è una relazione binaria non riflessiva e simmetrica (i.e., gli archi non hanno orientazione). Gruppi e grafi sono esempi di *strutture*, ossia insiemi su cui sono definite operazioni, relazioni e costanti.

In Algebra ci si può interessare alle proprietà di *singoli gruppi* di particolare interesse, o alle proprietà di *classi di gruppi* di particolare interesse (per esempio i gruppi abeliani, i gruppi ciclici, i gruppi di permutazioni, etc.), o alle proprietà di *tutti i gruppi*. In Teoria dei Grafi ci si può interessare alle verità che valgono in singoli grafi di particolare interesse, o alle verità che valgono di classi di grafi di particolare interesse (bipartiti, completi, Euleriani, etc.), o infine alle verità che valgono per tutti i grafi. In Logica Matematica facciamo un passo di generalizzazione ulteriore e ci interessiamo alla verità *in tutte le strutture*. Per questo motivo diamo le seguenti definizioni in forma molto generale.

Fissiamo un linguaggio  $\mathcal{L} = \{R_i, f_j, c_k : i \in I, j \in J, k \in K\}$  dove  $I, J, K$  sono insiemi (di indici). Una struttura  $\mathbf{A}$  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  $R_i^{\mathbf{A}}$ .
- Per ogni  $f_j$  di dimensione  $k$ , una funzione a  $k$  argomenti su  $A$ , che denotiamo con  $f_j^{\mathbf{A}}$ .
- Per ogni  $c_k \in K$ , un elemento di  $A$ , che denotiamo con  $c_k^{\mathbf{A}}$ .

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  $\mathbf{A}$* , che denotiamo con  $\mathbf{A} \models F$ .

Un *assegnamento*  $\alpha$  in  $\mathbf{A}$  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 in modo univoco ai termini ponendo  $\alpha(c)$  uguale a  $c^{\mathbf{A}}$  e  $\alpha(f(t_1, \dots, t_k))$  uguale a  $f^{\mathbf{A}}(\alpha(t_1), \dots, \alpha(t_k))$ . Indichiamo con  $\alpha \binom{x}{a}$  l'assegnamento che differisce da  $\alpha$  solo perché associa l'elemento  $a$  alla variabile  $x$ .

Definiamo la relazione  $\mathbf{A} \models F[\alpha]$ , che intuitivamente significa: la formula  $F$  è soddisfatta nella struttura  $\mathbf{A}$  relativamente all'assegnamento  $\alpha$ .

**Definizione 2.1** (Soddisfazione). Definiamo la relazione  $\mathbf{A} \models F[\alpha]$  come segue, per induzione sulla complessità di  $F$ .

$\mathbf{A} \models (t = s)[\alpha]$  se e solo se  $\alpha(t)$  e  $\alpha(s)$  sono lo stesso elemento di  $A$ .

$\mathbf{A} \models R(t_1, \dots, t_k)[\alpha]$  se e solo se  $(\alpha(t_1), \dots, \alpha(t_k)) \in R^{\mathbf{A}}$ .

$\mathbf{A} \models \neg G[\alpha]$  se e solo se non vale  $\mathbf{A} \models G[\alpha]$ .

$\mathbf{A} \models \exists v G[\alpha]$  se e solo se esiste  $a \in A$  tale che  $\mathbf{A} \models G[\alpha \binom{v}{a}]$ .

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

**Esempio 2.2.** Sia  $\mathcal{L} = \{0, 1, +, \times, <\}$ . Sia  $F(x)$  la formula  $\exists y(x = y + y)$ . Consideriamo la struttura  $\mathcal{N}$  con dominio  $\mathbf{N}$  che interpreta la costante 0 nel numero 0 la costante 1 nel numero 1, il simbolo di funzione  $+$  con l'addizione, il simbolo di funzione  $\times$  con il prodotto e il simbolo di relazione  $<$  con la normale relazione d'ordine stretto sui numeri naturali.  $\mathcal{N} \models F(x)[\alpha]$  se e solo se esiste  $n \in \mathbf{N}$  tale che  $\mathcal{N} \models (x = y + y)[\alpha \binom{y}{n}]$  se e solo se  $\alpha(x)$  è identico a  $n + n$ . Dunque  $\mathcal{N} \models F(x)[\alpha]$  se e solo se  $\alpha(x)$  è pari.

**Osservazione 2.3.** Il fatto che valga  $\mathbf{A} \models F[\alpha]$  o no dipende soltanto dai valori di  $\alpha$  sulle variabili libere che appaiono in  $F$ . In altre parole, se le variabili libere di  $F$  sono contenute in  $\{x_1, \dots, x_n\}$  e  $\alpha$  e  $\beta$  sono due assegnamenti che coincidono sui valori assegnati alle variabili  $x_1, \dots, x_n$ , allora  $\mathbf{A} \models F[\alpha]$  se e solo se  $\mathbf{A} \models F[\beta]$ . Pertanto possiamo scrivere  $\mathbf{A} \models F[\alpha(x_1), \dots, \alpha(x_n)]$  indicando esplicitamente gli elementi assegnati alle variabili che contano. Da questa osservazione segue anche che se  $F$  è un enunciato, allora  $\mathbf{A} \models F[\alpha]$  vale per tutti gli assegnamenti o per nessuno!

**Esempio 2.4.** Sia  $\mathcal{L} = \{c, f, R\}$  dove  $f$  è un simbolo di funzione a un argomento e  $R$  un simbolo di relazione binaria. Sia  $F$  l'enunciato  $\forall x \forall y (R(x, y) \rightarrow \exists z (R(x, z) \wedge R(z, y)))$  e sia  $G$  l'enunciato  $\forall x (x = c \vee R(f(x), x))$ . Sia  $\mathbf{A}$  la struttura con dominio  $\mathbb{Z}$ , che interpreta  $c$  in 0,  $f$  in  $x \mapsto x - 1$  e  $R$  nella relazione d'ordine  $<$ .  $\mathbf{A} \models F[\alpha]$  se e solo se per ogni intero  $a$  se  $R(\alpha(x), a)$  allora per esiste un intero  $b$  tale che  $a < b$  e  $b < a$ .  $\mathbf{A} \models G[\alpha]$  se e solo se per ogni intero  $a$  per ogni intero  $b$ , se  $a$  non è 0 allora  $a - 1$  è minore di  $a$ .

**Definizione 2.5** (Soddisfacibilità, Validità in una struttura). Se  $\mathbf{A} \models F[\alpha]$  per qualche assegnamento  $\alpha$ , diciamo che  $\alpha$  *soddisfa* l'enunciato  $F$  in  $\mathbf{A}$ , e in tal caso  $F$  è detta *soddisfacibile in  $\mathbf{A}$* . Una formula è *soddisfacibile* se è soddisfacibile in qualche struttura. Diciamo che una formula  $F$  è *vera in una struttura* se è soddisfatta da tutti gli assegnamenti su quella struttura. In questo caso scriviamo  $\mathbf{A} \models F$ .

**Osservazione 2.6.** Una formula  $F$  è vera in una struttura se e solo se l'enunciato

$$\forall x_1 \dots \forall x_n F(x_1, \dots, x_n),$$

è vero nella struttura, dove  $x_1, \dots, x_n$  sono tutte e sole le variabili libere di  $F$

**Definizione 2.7** (Validità). Una formula è *valida* se è vera in tutte le strutture, ossia se è soddisfatta da tutte le strutture e per tutti gli assegnamenti. Diciamo anche che è una *verità logica* e scriviamo  $\models F$ .

**Osservazione 2.8.** Si osserva che  $F$  è valida se e solo se  $\neg F$  non è soddisfacibile.

**Definizione 2.9** (Conseguenza logica, Equivalenza logica). Siano  $F, G$  formule. Diciamo che  $G$  è una *conseguenza (logica)* di  $F$  se per ogni struttura  $\mathbf{A}$ , per ogni assegnamento  $\alpha$  in  $A$ , se  $\mathbf{A} \models F[\alpha]$  allora  $\mathbf{A} \models G[\alpha]$ . In tal caso scriviamo  $F \models G$ . Due formule  $F, G$  sono *logicamente equivalenti* se  $F \models G$  e  $G \models F$ . In tal caso scriviamo  $F \equiv G$ .

**Osservazione 2.10.**  $F \models G$  se e solo se  $\models (F \rightarrow G)$ .  $F \equiv G$  se e solo se  $\models (F \leftrightarrow G)$ .

**Osservazione 2.11.** Se  $F \models G$  allora vale anche la seguente implicazione: per ogni  $\mathbf{A}$ , se  $\mathbf{A} \models F$  allora  $\mathbf{A} \models G$ . Se  $F$  e  $G$  sono enunciati, allora vale anche il viceversa, ossia: se per ogni  $\mathbf{A}$ , se  $\mathbf{A} \models F$  allora  $\mathbf{A} \models G$ , allora  $F \models G$ . Se invece  $F$  e  $G$  sono formule con variabili libere, l'implicazione inversa non vale in generale. Per un controesempio consideriamo  $R(v)$  e  $\forall v R(v)$ . Vale che per ogni  $\mathbf{A}$ , se  $\mathbf{A} \models R(v)$  allora  $\mathbf{A} \models \forall v R(v)$ , ma non vale  $R(v) \models \forall v R(v)$ . Per es., nella struttura con dominio  $\mathbf{N}$  dove  $R$  viene interpretata come la proprietà di *essere un numero pari*, esiste un assegnamento che associa alla variabile  $v$  il numero 4. Un tale assegnamento soddisfa  $R(v)$  ma ovviamente non  $\forall v R(v)$ .

### 3. PROPRIETÀ FONDAMENTALI DEI QUANTIFICATORI

Dimostriamo le seguenti equivalenze logiche fondamentali riguardanti i quantificatori.

$$\begin{aligned} \exists x(F \vee G) &\equiv \exists xF \vee \exists xG \\ \forall x(F \wedge G) &\equiv \forall xF \wedge \forall xG \\ \neg \exists x \neg F &\equiv \forall xF \\ \neg \forall x \neg F &\equiv \exists xF \\ \neg \exists x F &\equiv \forall x \neg F \\ \neg \forall x F &\equiv \exists x \neg F \\ \exists x \exists y F &\equiv \exists y \exists x F \\ \forall x \forall y F &\equiv \forall y \forall x F \end{aligned}$$

Se  $x$  non occorre in  $F$ ,

$$\begin{aligned} F \vee \exists x G &\equiv \exists x(F \vee G), & F \wedge \exists x G &\equiv \exists x(F \wedge G) \\ F \wedge \forall x G &\equiv \forall x(F \wedge G), & F \vee \forall x G &\equiv \forall x(F \vee G) \end{aligned}$$

Dimostriamo la prima equivalenza. Cominciamo col dimostrare il primo verso, ossia

$$\models \exists x(F \vee G) \rightarrow \exists xF \vee \exists xG$$

Dobbiamo dimostrare che, per ogni struttura  $\mathbf{A}$ , per ogni assegnamento  $\alpha$  in  $A$ , se  $\mathbf{A} \models \exists x(F \vee G)[\alpha]$  allora allora  $\mathbf{A} \models (\exists xF \vee \exists xG)[\alpha]$ . Sia  $\alpha$  un assegnamento tale che  $\mathbf{A} \models \exists x \exists x(F \vee G)[\alpha]$ . Per definizione esiste  $a \in A$  tale che  $\mathbf{A} \models (F \vee G)[\alpha(\frac{x}{a})]$ . Per definizione  $\mathbf{A} \models (F \vee G)[\alpha(\frac{x}{a})]$  è vero se e solo se  $\mathbf{A} \models F[\alpha(\frac{x}{a})]$  oppure  $\mathbf{A} \models G[\alpha(\frac{x}{a})]$ . Ma allora esiste un  $a \in A$  tale che  $\mathbf{A} \models F[\alpha(\frac{x}{a})]$  oppure esiste un  $a \in A$  tale che  $\mathbf{A} \models G[\alpha(\frac{x}{a})]$ . Dunque per definizione  $\mathbf{A} \models (\exists xF)[\alpha]$  oppure  $\mathbf{A} \models (\exists xG)[\alpha]$ . Dunque  $\mathbf{A} \models (\exists xF \vee \exists xG)[\alpha]$

Dimostriamo ora il secondo verso, ossia

$$\models \exists x(F \vee G) \rightarrow \exists xF \vee \exists xG.$$

Dobbiamo dimostrare che, per ogni struttura  $\mathbf{A}$ , per ogni assegnamento  $\alpha$ , se  $\mathbf{A} \models (\exists xF \vee \exists xG)[\alpha]$  allora vale  $\mathbf{A} \models \exists x(F \vee G)[\alpha]$ . Supponiamo l'antecedente e dimostriamo il conseguente. Sia  $\alpha$  un assegnamento tale che  $\mathbf{A} \models (\exists xF \vee \exists xG)[\alpha]$ . Per definizione questo significa che  $\mathbf{A} \models (\exists xF)[\alpha]$  oppure  $\mathbf{A} \models (\exists xG)[\alpha]$ . Dunque o esiste  $a \in A$  tale che  $\mathbf{A} \models F[\alpha(\frac{x}{a})]$  oppure esiste  $a \in A$  tale che  $\mathbf{A} \models G[\alpha(\frac{x}{a})]$ . Dunque in ogni caso esiste un  $a \in A$  tale che  $\mathbf{A} \models (F \vee G)[\alpha(\frac{x}{a})]$ . Per definizione questo significa che  $\mathbf{A} \models (\exists x)(F \vee G)[\alpha]$ .

Tutte le altre equivalenze si dimostrano in modo analogo applicando la definizione di soddisfacibilità.

Come dimostrare che una certa formula non è una verità logica? Occorre esibire una struttura e un assegnamento che non la soddisfa. Facciamo un esempio. Dimostriamo che il seguente enunciato non è una verità logica.

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

Esibiamo una struttura che soddisfa l'antecedente ma non il conseguente. Consideriamo la struttura  $\mathcal{N}$  con dominio  $\mathbf{N}$  in cui il simbolo  $R$  è interpretato come la relazione d'ordine tra i naturali,  $<$ .

$$\mathcal{N} \models \forall x \exists y R(x, y)$$

perché per ogni  $n \in \mathbf{N}$  esiste  $m \in \mathbf{N}$  tale che  $\mathcal{N} \models R(x, y) \left[ \binom{x}{n} \binom{y}{m} \right]$ , ossia per ogni  $n \in \mathbf{N}$  esiste  $m \in \mathbf{N}$  tale che  $n < m$ . D'altra parte però  $\mathcal{N}$  non soddisfa il conseguente, perché è falso che per esiste  $n \in \mathbf{N}$  tale che per ogni  $m \in \mathbf{N}$  vale  $n > m$ .

A volte per dimostrare che un enunciato è una verità logica è comodo ragionare per assurdo e ragionare su una struttura che verifica la negazione dell'enunciato. Facciamo un esempio. Dimostriamo che

$$\models \neg \exists x \forall y (S(x, y) \leftrightarrow \neg S(x, x))$$

Supponiamo per assurdo che l'enunciato non è una verità logica. Allora esiste una struttura  $\mathbf{A}$  tale che

$$\mathbf{A} \models \exists x \forall y (S(x, y) \leftrightarrow \neg S(x, x)).$$

Questo è vero se e solo se esiste  $a \in A$  tale che per ogni  $b \in A$   $(a, b) \in S^{\mathbf{A}}$  (i.e.,  $a$  e  $b$  stanno nella relazione con cui  $\mathbf{A}$  interpreta il simbolo  $S$ ) se e solo se  $(a, a) \notin S^{\mathbf{A}}$ . Sia  $a_0$  un tale  $a \in A$ . Allora

$$\text{per ogni } b \in A \text{ vale che } (a_0, b) \in S^{\mathbf{A}} \text{ se e solo se } (a_0, a_0) \notin S^{\mathbf{A}}.$$

Chiediamoci se valga  $(a_0, a_0) \in S^{\mathbf{A}}$  o meno. Se  $(a_0, a_0) \in S^{\mathbf{A}}$  allora  $(a_0, a_0) \notin S^{\mathbf{A}}$  (perché l'implicazione vale per ogni  $b \in A$ , incluso  $a_0$ ). Se invece  $(a_0, a_0) \notin S^{\mathbf{A}}$ , allora  $(a_0, a_0) \in S^{\mathbf{A}}$ . L'esistenza di  $a_0$  è dunque contraddittoria. Concludiamo che l'enunciato di partenza è una verità logica.