Intelligenza Artificiale II

Logica del primo ordine

Elementi di base

Marco Piastra

Limiti del linguaggio proposizionale

- *Esempio*:

  "Ogni essere umano è mortale"
  "Socrate è un essere umano"

  "Socrate è mortale"

  Il legame logico (intuitivo) è evidente

(Essere umano) implica (essere mortale) $A \to B$
Socrate è un essere umano C

Socrate è mortale D

$$\frac{A \to B \quad C}{D}$$

Schema del *modus ponens*

$$\frac{\varphi \to \psi \quad \varphi}{\psi}$$

Nella traduzione logico-proposizionale A, B, C e D non hanno alcun legame

- Altri esempi 'intraducibili':

  "Quando capra e cavolo stanno sulla stessa riva, la capra mangia il cavolo"

  "Il cacciatore sente una brezza quando si trova sul ciglio di una trappola"

Logica del primo ordine

- **Logica proposizionale**

  Rappresentazione del mondo come sulla base di **fatti** atomici (proposizioni)

- **Logica del primo ordine**

  Una base più complessa e articolata per la rappresentazione

  Oggetti

  persone, numeri, costruzioni, colori, storie, percorsi, pezzi di legno, …

  Proprietà (di oggetti) e **relazioni** (tra oggetti)

  rosso, grande, primo, fratello di, maggiore di, compreso tra, …

  Funzioni (di oggetti)

  successore, padre di, …

Linguaggio del primo ordine

Costanti, predicati, funzioni

- Simboli del linguaggio L_{PO}

  Costanti individuali (singoli oggetti)

  Esempi: *Socrate*, *Sfera1*, *MickeyMouse*, *Amelia*, ...

  In generale: a, b, c, …

  Predicati

  Esempi: *Red*/1, *Large*/1, *GreaterThan*/2, =/2

  Ogni predicato ha un numero di argomenti prestabilito

  In generale: P/n, Q/m, R/p, …

  Funzioni

  Esempi: *sqrt*/1, *colorOf*/1, *greatestCommonDivisor*/2

  Ogni funzione ha un numero di argomenti prestabilito

  In generale: f/n, g/m, h/p, …

Convenzionalmente, per il predicato binario = si usa la forma infissa (p.es. '$a = b$')

Linguaggio del primo ordine

Altri simboli

- Ulteriori simboli del linguaggio L_{PO}

  Variabili

  Convenzionalmente indicate come x, y, z, …

  Connettivi

  Come in logica proposizionale: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$

  Parentesi e virgola

  ( ) ,

  Quantificatori

  Universale: $\forall$

  Esistenziale: $\exists$

  Si applicano **sempre** ad una ed una sola variabile
  Esempi: $\forall x$, $\exists y$, $\forall x \forall y$

Si dice linguaggio **del primo ordine** in quanto
i quantificatori agiscono solo sugli **oggetti**,
vale a dire sulle variabili x, y, z …, e non sulle **relazioni** e **funzioni**
(*In una logica del secondo ordine si hanno formule del tipo*: $\exists F\ F(a,b)$)

Esempi di formule

- "Essere fratelli significa essere parenti"

  $\forall x \forall y\ (Fratello(x, y) \rightarrow Parente(x, y))$

- "La relazione di parentela è simmetrica"

  $\forall x \forall y\ (Parente(x ,y) \leftrightarrow Parente(y, x))$

- "Una madre è un genitore di sesso femminile"

  $\forall x \forall y\ (Madre(x, y) \leftrightarrow (Genitore(x, y) \wedge Femmina(x)))$

- "Un cugino è figlio di un fratello o una sorella di uno dei genitori"

  $\forall x \forall y\ (Cugino(x, y) \leftrightarrow \exists z \exists w\ (Genitore(z, x) \wedge Genitore(w, y) \wedge (Fratello(z, w) \vee Sorella(z, w))))$

- "Ciascuno ha una madre"

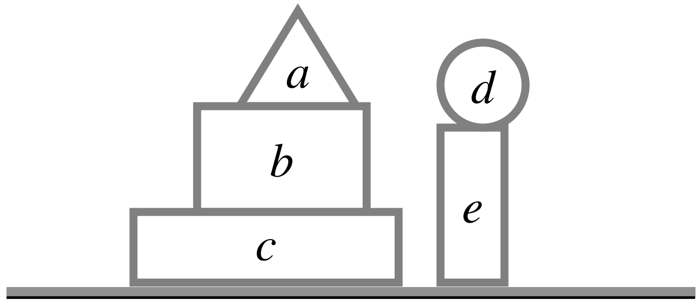
  $\forall x \exists y\ Madre(y, x)$

  Fare attenzione all'ordine dei quantificatori

  $\exists y \forall x\ Madre(y, x)$

  "Esiste una madre di tutti"

Descrivere un mondo



- Insiemi (esempi)

  $Pyramid(a)$
  $Parallelepiped(b) \land Parallelepiped(c) \land Parallelepiped(e)$
  $Sphere(d)$
  $Ontable(c) \land Ontable(e)$
  $Clear(a) \land Clear(d)$

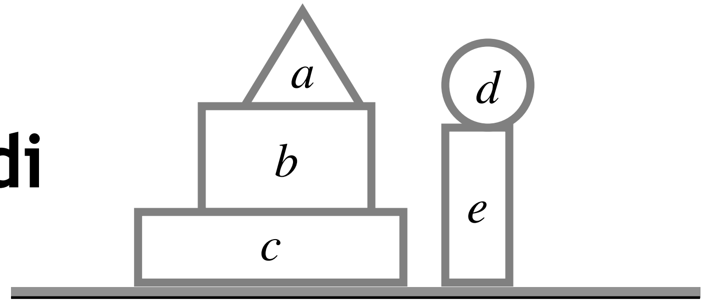
- Relazioni (esempio)

  $Above(a,b) \land Above(b,c) \land Above(a,c) \land Above(d,e)$

- Definizioni

  $\forall x \forall y\ (On(x,y) \leftrightarrow (Above(x,y) \land \neg \exists z\ (Above(x,z) \land Above(z,y))))$
  $\forall x\ (Ontable(x) \leftrightarrow \neg \exists z\ Above(x,z))$
  $\forall x\ (Clear(x) \leftrightarrow \neg \exists z\ Above(z,x))$

Assiomatizzare una famiglia di mondi



- Assiomi Ax_{bw} (Cook & Liu, 2002)
  1) $\forall x\ \neg Above(x,x)$
  2) $\forall x \forall y \forall z\ ((Above(x,y) \wedge Above(y,z)) \rightarrow Above(x,z))$
  3) $\forall x \forall y \forall z\ ((Above(x,y) \wedge Above(x,z)) \rightarrow ((y=z) \vee Above(y,z) \vee Above(z,y))$
  4) $\forall x \forall y \forall z\ ((Above(y,x) \wedge Above(z,x)) \rightarrow ((y=z) \vee Above(y,z) \vee Above(z,y))$
  5) $\forall x\ (Ontable(x) \vee \exists y\ (Above(x,y) \wedge Ontable(y)))$
  6) $\forall x\ (Clear(x) \vee \exists y\ (Above(y,x) \wedge Clear(y)))$
  7) $\forall x \forall y\ (Above(x,y) \rightarrow (\exists z\ On(x,z) \wedge \exists w\ On(w,y)))$

Termini e formule atomiche

- **Termini**

  Ogni *variabile* o *costante individuale* è un **termine**

  Se f è un *simbolo funzionale* a n argomenti e $t_1, \ldots, t_n$ sono **termini**, allora $f(t_1, \ldots, t_n)$ è un **termine**

  Un termine **chiuso** (*ground*) non contiene variabili

- **Formula atomica**

  Se P è un *simbolo predicativo* a *n* argomenti e $t_1, \ldots, t_n$ sono **termini**, allora $P(t_1, \ldots, t_n)$ è una **formula atomica**

Esempi:

$$\underbrace{\underbrace{Sorella}_{\text{predicato}}(\underbrace{\underbrace{Amelia}_{\text{costante}}}_{\text{termine}}, \underbrace{\underbrace{Alba}_{\text{costante}}}_{\text{termine}})}_{\text{formula atomica}}$$

$$\underbrace{\underbrace{shapeOf(\underset{\text{costante}}{b})}_{\text{termine}} \underset{\text{predicato}}{=} \underbrace{shapeOf(\underset{\text{variabile}}{x})}_{\text{termine}}}_{\text{formula atomica}}$$

Regole di buona formazione

- Formule ben formate di L_{PO}

Ogni *formula atomica* è una fbf

$$\varphi \in \text{fbf}(L_{PO}) \;\Rightarrow\; (\neg\varphi) \in \text{fbf}(L_{PO})$$
$$\varphi, \psi \in \text{fbf}(L_{PO}) \;\Rightarrow\; (\varphi \to \psi) \in \text{fbf}(L_{PO})$$
$$\varphi \in \text{fbf}(L_{PO}) \;\Rightarrow\; (\forall x\, \varphi) \in \text{fbf}(L_{PO})$$

$$\varphi, \psi \in \text{fbf}(L_{PO}), \qquad (\varphi \vee \psi) \;\Leftrightarrow\; ((\neg\varphi) \to \psi)$$
$$\varphi, \psi \in \text{fbf}(L_{PO}), \qquad (\varphi \wedge \psi) \;\Leftrightarrow\; (\neg(\varphi \to (\neg\psi)))$$
$$\varphi, \psi \in \text{fbf}(L_{PO}), \qquad (\varphi \leftrightarrow \psi) \;\Leftrightarrow\; ((\varphi \to \psi) \wedge (\psi \to \varphi))$$
$$\varphi \in \text{fbf}(L_{PO}) \qquad (\exists x\, \varphi) \Leftrightarrow (\neg\forall x\, \neg\varphi)$$

Formule aperte, enunciati

- Variabili **libere** e **vincolate**

  una variabile (in una fbf) è **vincolata** se si trova nel raggio di azione di un **quantificatore** per quella variabile

  una variabile è **libera** se non è *vincolata*

  esempi di variabile vincolata:
  $\forall x\, P(x)$
  $\exists x\, (P(x) \rightarrow (A(x) \wedge B(x))$

  esempi di variabile libera:
  $P(x)$
  $\exists y\, (P(y) \rightarrow (A(x,y) \wedge B(y)))$

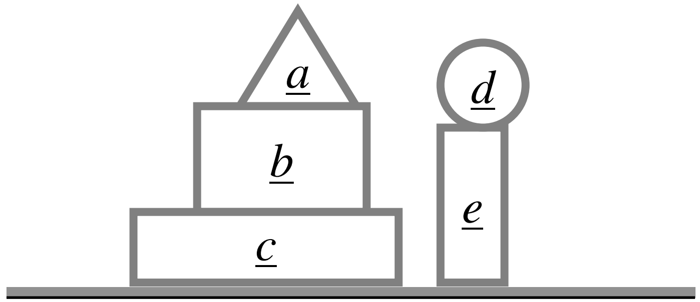
- **Formule aperte** e **chiuse**

  Una fbf è **aperta** se in essa vi è almeno una variabile libera

  Una fbf è **chiusa** (anche **enunciato** - *sentence*) in caso contrario

  solo le fbf *chiuse*, cioè gli *enunciati*, hanno un valore di verità (in quanto rappresentano delle *affermazioni* ...)

Semantica estensionale



- **Universo del discorso U**

  Un insieme di oggetti di base

  $\{\underline{a}, \underline{b}, \underline{c}, \underline{d}, \underline{e}, \underline{nil}\}$

  Le costanti individuali indicano oggetti di **U**

  Esempi: $v(a) = \underline{a},\ v(b) = \underline{b}$

- **Relazioni**

  Insiemi di tuple formate a partire da **U**

  $\underline{Ontable} : \{\underline{c}, \underline{e}\}$

  $\underline{Above} : \{<\underline{a}, \underline{b}>, <\underline{a}, \underline{c}>, <\underline{b}, \underline{c}>, <\underline{d}, \underline{e}>\}$

  $\underline{On} : \{<\underline{a}, \underline{b}>, <\underline{b}, \underline{c}>, <\underline{d}, \underline{e}>\}$

  $\underline{Between} : \{<\underline{b}, \underline{a}, \underline{c}>\}$

- **Funzioni**

  Insiemi di tuple formate a partire da **U**, tali

  $\underline{on} : \{<\underline{a}, \underline{b}>, <\underline{b}, \underline{c}>, <\underline{c}, nil>, <\underline{d}, \underline{e}>, <\underline{e}, nil>\}$

Si usa a per indicare la costante ed $\underline{a}$ per indicare l'oggetto (*solo per comodità)

- Predicati e funzioni indicano relazioni e funzioni in **U**

  $v(Above/2) : \{<\underline{a}, \underline{b}>, <\underline{a}, \underline{c}>, <\underline{b}, \underline{c}>, <\underline{d}, \underline{e}>\}$
  $v(on/2) : \{<\underline{a}, \underline{b}>, <\underline{b}, \underline{c}>, <\underline{c}, nil>, <\underline{d}, \underline{e}>, <\underline{e}, nil>\}$

Strutture, interpretazioni e assegnazioni

- Una **struttura** $<\mathbf{U}, v>$ per L_{PO} contiene:

  Un insieme di oggetti $\mathbf{U}$ (l'universo del discorso)

  Un'**interpretazione** v che associa

  ad ogni **costante** c un **oggetto** di $\mathbf{U}$
  $v(c) \in \mathbf{U}$

  ad ogni **predicato** P a n argomenti una **relazione** n-aria in $\mathbf{U}^n$
  $v(P) \subseteq \mathbf{U}^n$

  ad ogni **funzione** f a n argomenti una **funzione** da $\mathbf{U}^n$ a $\mathbf{U}$
  $v(f) \subseteq \mathbf{U}^n \rightarrow \mathbf{U}$

  Un'interpretazione (ovviamente) non fissa il valore delle variabili

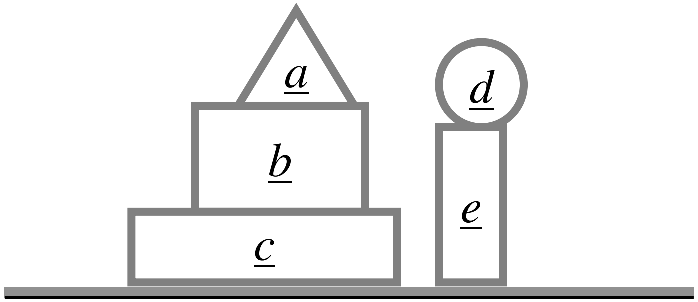
- Assegnazione

  Data una struttura $<\mathbf{U}, v>$, un'**assegnazione** (*valuation*) s

  è una *funzione* che associa ad ogni **variabile** x un **oggetto** di $\mathbf{U}$
  $s(x) \in \mathbf{U}$

  La combinazione di una $<\mathbf{U}, v>$ e di una s determina univocamente gli oggetti associati a ciascun elemento di L_{PO}

Il mondo dei blocchi



- Linguaggio

  simboli predicativi: $Ontable(.)$, $Above(..)$, $On(..)$, $Between(\ldots)$

  variabili: $x, y, z, \ldots$

  costanti individuali: a, b, c, d, e

- Interpretazione

  Universo del discorso
  $\mathbf{U} = \{\underline{a}, \underline{b}, \underline{c}, \underline{d}, \underline{e}\}$

  Predicati

  $v(Ontable) = \{\underline{c}, \underline{e}\}$
  $v(Above) = \{<\underline{a}, \underline{b}>, <\underline{a}, \underline{c}>, <\underline{b}, \underline{c}>, <\underline{d}, \underline{e}>\}$
  $v(On) = \{<\underline{a}, \underline{b}>, <\underline{b}, \underline{c}>, <\underline{d}, \underline{e}>\}$
  $\mathrm{v}(Between) = \{<\underline{b}, \underline{a}, \underline{c}>\}$

  Costanti individuali

  $v(a) = \underline{a}$, $v(b) = \underline{b}$, $v(c) = \underline{c}$, $v(d) = \underline{d}$, $v(e) = \underline{e}$

Si usa a per indicare la costante
ed $\underline{a}$ per indicare l'oggetto
(*solo per comodità)

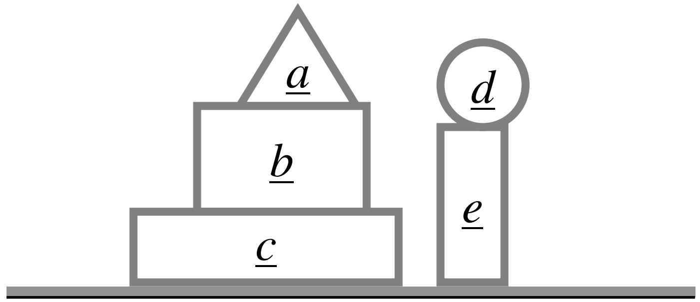
- Assegnazione

  Esempio: $s = \{(x{:}\underline{a}), (y{:}\underline{b}), (z{:}\underline{a}) \ldots\}$ (per _tutte_ le variabili del linguaggio)

Soddisfacimento (forma intuitiva)

- Una fbf φ è soddisfatta da $<\mathbf{U}, v>$ $[s]$
  sse φ afferma una cosa vera in $<\mathbf{U}, v>$ $[s]$

- Nel mondo dei blocchi



$Pyramid(a)$

è vera perchè: $(v(a) = \underline{a}) \in v(Pyramid(.)) = \{\underline{a}\}$

$Parallelepiped(d)$

non è vera perchè: $(v(d) = \underline{d}) \notin v(Parallelepiped(.)) = \{\underline{b}, \underline{c}, \underline{e}\}$

$\neg\exists x\ (Parallelepiped(x) \wedge Sphere(x))$

è vera perché: $((v(Parallelepiped(.)) = \{\underline{b}, \underline{c}, \underline{e}\}) \cap (v(Sphere(.)) = \{\underline{d}\})) \equiv \varnothing$

$\forall x\ (Pyramid(x) \vee Parallelepiped(x) \vee Sphere(x))$

è vera perché:
$((v(Pyramid(.)) = \{\underline{a}\}) \cup (v(Parallelepiped(.)) = \{\underline{b}, \underline{c}, \underline{e}\}) \cup (v(Sphere(.)) = \{\underline{d}\})) \equiv \mathbf{U}$

Soddisfacimento

- Data una struttura $<\mathbf{U}, v>$ un'assegnazione s

  Se φ è una formula atomica, $<\mathbf{U}, v> [s] \models \varphi$ sse

  se φ ha la forma $P(t_1, \ldots, t_n)$ allora $<v(t_1)\ [s], \ldots, v(t_n)\ [s]> \in v(P)\ [s]$

  Se φ e ψ sono fbf qualsiasi

  $<\mathbf{U}, v> [s] \models (\neg\varphi)$ sse $<\mathbf{U}, v> [s] \not\models \varphi$

  $<\mathbf{U}, v> [s] \models (\varphi \land \psi)$ sse $<\mathbf{U}, v> [s] \models \varphi$ e $<\mathbf{U}, v> [s] \models \psi$

  $<\mathbf{U}, v> [s] \models (\varphi \lor \psi)$ sse $<\mathbf{U}, v> [s] \models \varphi$ o $<\mathbf{U}, v> [s] \models \psi$

  $<\mathbf{U}, v> [s] \models (\varphi \to \psi)$ allora non $<\mathbf{U}, v> [s] \models \varphi$ e $<\mathbf{U}, v> [s] \not\models \psi$

  Formule con quantificatori

  $<\mathbf{U}, v> [s] \models \forall x\, \varphi$ sse per ogni $\underline{d} \in \mathbf{U}$ si ha $<\mathbf{U}, v> [s](x{:}\underline{d}) \models \varphi$

  $<\mathbf{U}, v> [s] \models \exists x\, \varphi$ sse esiste un $\underline{d} \in \mathbf{U}$ per cui si ha $<\mathbf{U}, v> [s](x{:}\underline{d}) \models \varphi$

Modelli

- Validità in un'interpretazione, modello

  Una fbf φ tale per cui si ha $<\mathbf{U}, v> [s] \models \varphi$ per qualsiasi assegnazione s è detta valida in $<\mathbf{U}, v>$

  Si dice anche che $<\mathbf{U}, v>$ è un modello di φ

  si scrive $<\mathbf{U}, v> \models \varphi$ (si elimina il riferimento a s)

  Una struttura $<\mathbf{U}, v>$ è detta modello di un *insieme di fbf* Γ sse è un modello di tutte le fbf in Γ

  si scrive allora $<\mathbf{U}, v> \models \Gamma$

- Verità

  Un enunciato ψ si dice vero in $<\mathbf{U}, v>$ se è valido in $<\mathbf{U}, v>$

  per un enunciato, basta l'esistenza di un'assegnazione s per cui $<\mathbf{U}, v> [s] \models \psi$

Validità

- Validità e verità logiche

  Una fbf (aperta o chiusa) è **valida** (o **logicamente valida**) se è **valida** in qualsiasi $<\mathbf{U}, v>$

  Esempi:

  $(P(x) \lor \neg P(x))$ (tautologia come formula aperta)

  Un enunciato ψ è **vero** (o **logicamente vero**) se è **vero** in qualsiasi $<\mathbf{U}, v>$

  si scrive allora $\models \psi$ (si elimina il riferimento a $<\mathbf{U}, v>$)

  Esempi:

  $\forall x\, (P(x) \lor \neg P(x))$ (generalizzazione di una tautologia)

  $\forall x \forall y\, (G(x,y) \to (H(x,y) \to G(x,y)))$ (generalizzazione di assioma – vedi oltre)

- Inconsistenza

  Una fbf (aperta o chiusa) è **inconsistente** se non è soddisfacibile

  Un enunciato ψ è **inconsistente** se non ha un *modello*

  Esempi:

  $\forall x\, (P(x) \land \neg P(x))$ (generalizzazione di una contraddizione)

Conseguenza logica

- Definizione

  Dato un insieme di fbf Γ ed una fbf φ di L_{PO} si ha

  $$\Gamma \models \varphi$$

  sse tutte le $<\mathbf{U}, v> [s]$ che soddisfano Γ soddisfano anche φ

- Osservazioni

  La definizione si estende a tutte le possibili $<\mathbf{U}, v> [s]$

  Quindi, a tutti i possibili insiemi $\mathbf{U}$, alle relazioni e funzioni in $\mathbf{U}$ ed alle associazioni di oggetti di $\mathbf{U}$ a variabili e costanti

  Il calcolo diretto della conseguenza logica in L_{PO} è impossibile anche nelle forme più semplici

FAQ 1

- Funzioni o predicati (i.e. relazioni)?

  I due oggetti semantici sono molto simili, si può fare a meno delle funzioni?

- Le funzioni (come oggetti semantici) si possono *rappresentare* anche tramite predicati

  ad esempio, la validità dell'enunciato:

  $$\forall x \forall y \forall z \, ((\varphi(x,y) \wedge \varphi(x,z)) \rightarrow (y = z))$$

  indica che l'interpretazione di $\varphi(..)$ (in generale, una relazione $v(\varphi) \subseteq \mathbf{U}^2$) è anche una funzione $\mathbf{U} \rightarrow \mathbf{U}$

  La presenza delle funzioni arricchisce il linguaggio in modo sostanziale: a differenza dei predicati, le funzioni si possono nidificare (nei termini)

  Di conseguenza, viene grandemente aumentata la portata del calcolo logico-simbolico

  (*con un corrispondente aumento della complessità di calcolo* ...)

Linguaggio e funzioni

- Ricchezza espressiva

  Basta una sola funzione in L_{PO} per creare un'infinità (numerabile) di termini

  Ad esempio, la sola funzione $s/1$ (=*successore*) è sufficiente per definire l'aritmetica (*la teoria delle proprietà dei numeri naturali*)

- Assiomi ricorsivi

  Esempio: (*postulati di Peano* – secondo Mendelson, 1972)

  S1: $\forall x \forall y \forall z\, ((x = y) \rightarrow ((x = z) \rightarrow (y = z)))$

  S2: $\forall x \forall y\, ((x = y) \rightarrow (s(x) = s(y)))$

  S3: $\neg \exists x\, (s(x) = 0)$

  S4: $\forall x \forall y\, ((s(x) = s(y)) \rightarrow (x = y))$

  S5: $\forall x\, (x + 0 = x)$

  S6: $\forall x \forall y\, ((x + s(y)) = s(x + y))$

  S7: $\forall x\, (x \cdot 1 = x)$

  S8: $\forall x \forall y\, ((x \cdot s(y)) = ((x \cdot y) + x))$

  S9: *Per qualsiasi fbf* $\varphi(x)$: $\varphi(0) \rightarrow (\forall x\, (\varphi(x) \rightarrow \varphi(s(x))) \rightarrow \forall x\, \varphi(x)$

  (~*principio di induzione matematica*)

Esempio

- Il mondo delle liste di oggetti $[a, b, c, \ldots]$

  $cons(s, x)$
  funzione, associa ad un oggetto (es. a) ed una lista (es. $[b, c]$) la lista ottenuta inserendo l'oggetto all'inizio (es. $[a, b, c]$)_

  $Append(x,y,z)$
  predicato, associa alle liste x e y la concatenazione z

  nil
  costante, indica la lista vuota.

  Notazione abbreviata:

  $[] \Leftrightarrow nil$
  $[a] \Leftrightarrow cons(a,nil)$
  $[a,b] \Leftrightarrow cons(a,cons(b,nil))$
  $[a|[b,c]] \Leftrightarrow cons(a,[b,c])$

Assiomi ($\mathbf{AL}$)

$\forall x\ Append(nil,x,x)$
$\forall x \forall y \forall z\ (Append(x,y,z) \rightarrow \forall s\ Append(cons(s,x),y,cons(s,z)))$

Esempi (conseguenze logiche)

$\mathbf{AL} + \exists z\ Append([a],[b,c],z)$	$\models Append([a],[b,c],[a,b,c])$	$= [z/[a,b,c]]$
$\mathbf{AL} + \exists x \exists y\ Append(x,y,[a,b])$	$\models Append([a],[b],[a,b])$	$= [x/[a], x/[b]]$
	$\models Append(nil,[a,b],[a,b])$	$= [x/nil, y/[a,b]]$
	$\models Append([a,b],nil,[a,b])$	$= [x/[a,b], y/nil]$