IX

TIPI DI DATO

58.- Tipo di dato astratto

Un tipo di dato è individuato da certi oggetti e da certe operazioni su di essi. È evidente che tale nozione intuitiva è completamente specificabile in modo preciso tramite quella di algebra.

Abbiamo per esempio già visto tra i primi esempi di algebra eterogenea il tipo di dato "pila su naturali". La stessa algebra $(N,+,\cdot,\emptyset,1)$ è un tipo di dato (il più antico).

L'approccio algebrico consente però in modo naturale la formalizzazione di un concetto di estremo interesse programmativo: quello di tipo di dato astratto. L'aggettivo astratto indica che gli oggetti su cui si vuole operare non sono definiti poichè ciò che interessa sono certe condizioni sulle operazioni che su di essi si intendono effettuare. In termini algebrici questo significa che un tipo di dato astratto non è un'algebra bensì una teoria algebrica i cui modelli risultano possibili realizzazioni concrete del tipo astratto.

Specificare algebricamente un tipo di dato astratto significa quindi tradurre in teoria algebrica una qualche teoria formale o informale che descriva il tipo considerato.

Consideriamo i seguenti esempi:

I) Pile su elementi di un prefissato insieme

Le pile sono oggetti costituiti a partire da una pila vuota empty e una operazione di "push" tale che

$$P \text{ è pila, } a \text{ è elemento} \Longrightarrow \text{push}(a,P) \text{ è pila}$$

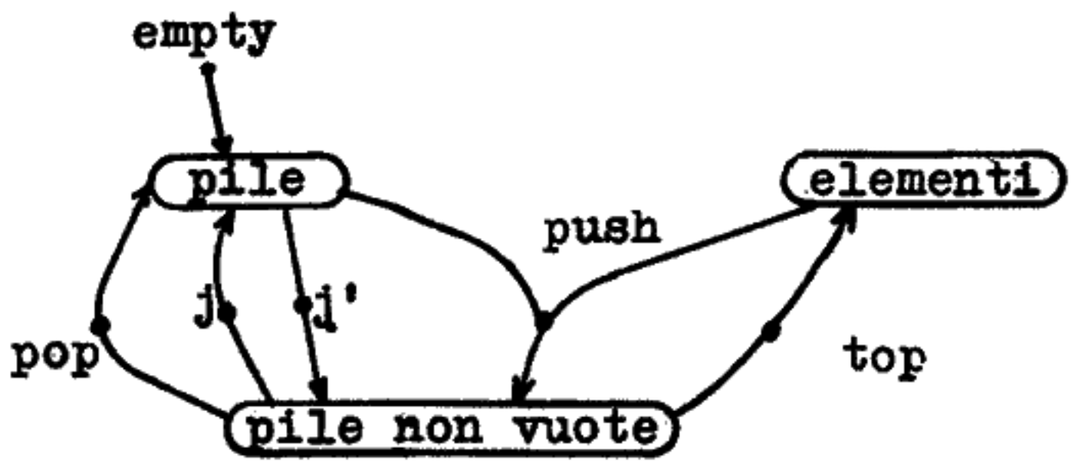
$$P' = \text{push}(a,P) \Longrightarrow \text{top}(P') = a$$

$$P' = \text{push}(a,P) \Longrightarrow \text{pop}(P') = P$$

(top e pop sono operazioni che agiscono su pile non vuote.)

Una specifica algebrica è la seguente

Segnatura



P : variabile di pile
Q : " " pile non vuote
e : " " elementi

Equazioni

j'(jP)= P Assioma che impone l'iniettività dell'immersione j .

pop (push (e,P)) = P

top (push (e,P)) = e

II) Liste su un prefissato insieme A

Le liste sono oggetti costituiti a partire dagli elementi di A detti atomi, utilizzando una operazione di costruzione **cons** secondo le regole seguenti :

o) Liste e atomi costituiscono insieme le espressioni

i) a è atomo, l è lista ⟹ cons (a,l) è lista

l è lista, l' è lista ⟹ cons (l,l') è lista

ii) Vi è un solo oggetto che è sia atomo che lista indicato con **Nil** e altre due operazioni **car** e **cdr** definite su tutte le liste tranne **Nil** come segue

l'= cons(e,l) ⟹ car (l') = e

l'= cons(e,l) ⟹ cdr (l') = l

iii) Vi sono due operazioni booleane

eq binaria su atomi:

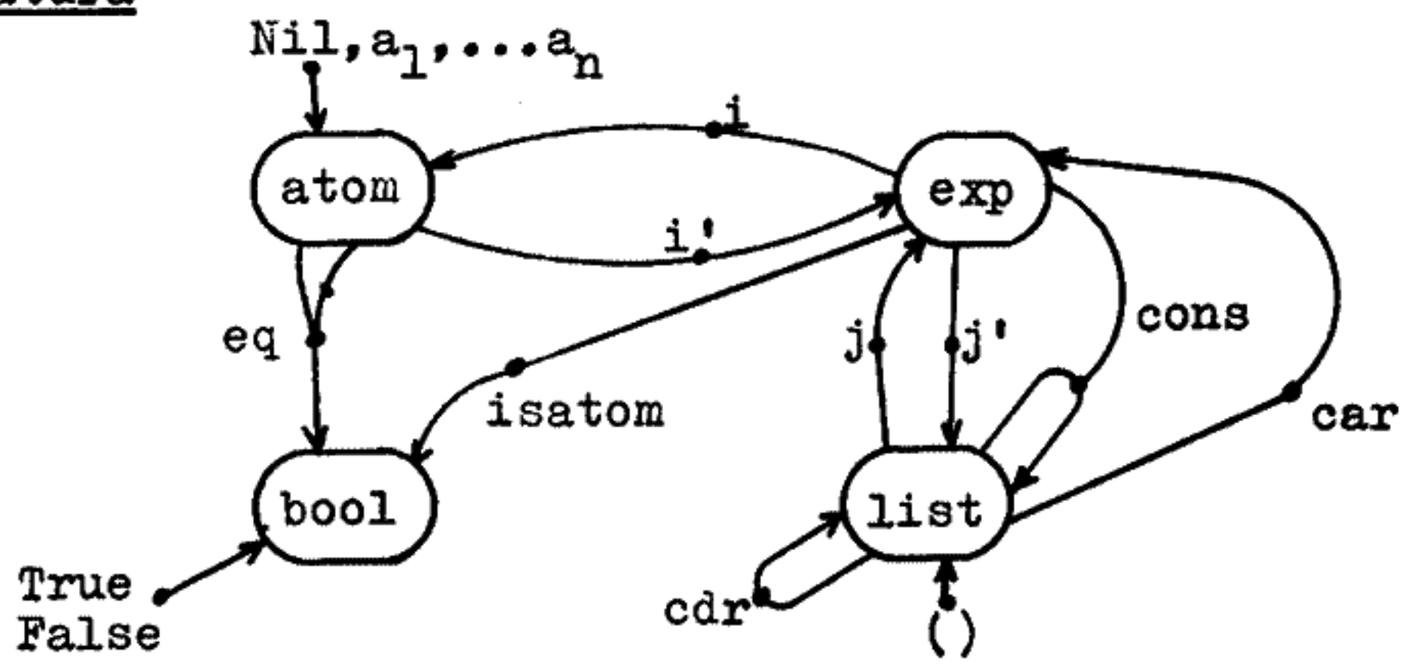
eq(a,a')= True $\iff$ a = a'

isatom unaria su espressioni:

isatom(a)= True $\iff$ a è atomo

Specifica algebrica:

Segnatura



$a_1, a_2, \ldots$ costanti, una per ogni elemento di A

η - variabile di exp

l - " " list

a - " " atom

Abbreviazioni:

$$\text{cons}(\eta, l) = (\eta\, l)$$

$$(\eta_1(\eta_2\eta_3)) = (\eta_1\eta_2\eta_3)$$

Equazioni:

$$\text{car}((\eta\, l)) = \eta$$

$$\text{cdr}((\eta\, l)) = l$$

$$\text{car}(\,()\,) = j()$$

$$\text{cdr}(\,()\,) = ()$$

$$i\ \text{Nil} = j()$$

$$\begin{cases} j'(jl) = l \\ i'(ia) = a \end{cases}$$ iniettività delle immersioni di liste e atomi in espressioni

isatom(ia)= True

isatom((ηl))= False

eq(a,a)= True

eq(a_i,Nil)= False $i= 1,2,...$

eq(a_i,a_j)= False $i,j= 1,2,...$, $i \neq j$

59.- Specifiche gerarchiche

Supponiamo ora di volere specificare il tipo di dato astratto Pila su naturali ove i naturali sono l'usuale algebra

$$\underline{\underline{N}} = (N, B, +, \cdot, =, \emptyset, 1)$$

e poi a partire da tali pile definire altre strutture astratte di dati. Tale modo di procedere è tipico delle cosidette specifiche gerarchiche. Un modo semplice di sviluppare tale procedimento per via algebrica si ottiene tramite la nozione di **algebra basata** che è una variante del concetto di algebra di dati (cfr. [35]).

DEFINIZIONE

Sia $\underline{\underline{B}}$ una algebra di segnatura Σ e sia Σ' una segnatura che estende Σ , cioè avente tra le sorte quelle di Σ e tra gli operatori quelli di Σ

— Una Σ'-algebra $\underline{\underline{A}}$ dicesi **basata** su $\underline{\underline{B}}$ se

$$\underline{\underline{A}}|_{\Sigma} \cong \underline{\underline{B}}$$

cioè se il sistema di domini e operazioni di $\underline{\underline{A}}$ relativa a Σ è una Σ-algebra isomorfa a $\underline{\underline{B}}$.

$\underline{\underline{A}}$ dicesi **sovrabasata** su $\underline{\underline{B}}$ se $\underline{\underline{B}}$ è isomorfa ad una sottoalgebra di $\underline{\underline{A}}|_{\Sigma}$

— Sia Σ_B la segnatura in cui tutti gli elementi dei domini di $\underline{\underline{B}}$ sono aggiunti come costanti (ciascuno nella sorta opportuna) dicesi diagramma di $\underline{\underline{B}}$ il seguente insieme di equazioni δ_B

$$\delta_B = \{ t_1 = t_2 \mid t_1, t_2 \in T_{\Sigma_B} , \underline{\underline{B}} \models t_1 = t_2 \}$$

Una teoria algebrica che includa δ_B dicesi B-basata.

È evidente che una specifica gerarchica è descrivibile algebricamente tramite teorie algebriche basate. Quindi se $\mathcal{E}$ è la teoria delle pile su elementi generici e δ_N è il diagramma di $\underline{\underline{N}}$ (ove ad N è assegnata la sorta elementi) $\mathcal{E} \cup \delta_N$ sarà la teoria delle pile su N.

Vale inoltre per le teorie basate il seguente teorema fondamentale

TEOREMA (sulle teorie basate; cfr. per la prova [27])

Dato un insieme $\mathcal{E}$ di Σ'-equazioni $\mathcal{E}$ dicesi

— B-consistente sse $\mathcal{E} \cup \delta_B \not\vdash a = a'$
con a, a' elementi distinti di $\underline{\underline{B}}$

— B-completo sse $\forall\, t \in T_{\underline{\underline{\Sigma_B}}\mid \Sigma}$ $\mathcal{E} \cup \delta_B \vdash t = a$
per qualche $a \in \underline{\underline{B}}$

Con tali notazioni valgono i fatti seguenti:

i) Se $\mathcal{E}$ è B-consistente e B-completa allora la classe delle Σ'_B-algebre B-basate che verificano $\mathcal{E}$ ha $T_{\Sigma'_B / \overline{(\mathcal{E} \cup \delta_B)^*}}$ come algebra iniziale.

ii) Se $\mathcal{E}$ è B-consistente la classe delle algebre B-sovrabasate che verificano $\mathcal{E}$ ha $T_{\Sigma'_B / \overline{(\mathcal{E} \cup \delta_B)^*}}$ come algebra iniziale.

iii) Se $\mathcal{E}$ è B-consistente e B-completa la classe

delle algebre minimali B-basate che verificano
$\mathcal{L}$ ha algebra finale.

60.- <u>Specifiche per inizialità</u>

Un altro caso di specifica astratta è il se= guente.

Definiamo matematicamente una Σ_o algebra che rappresenta il tipo di dato Lista ove Σ_o è la segnatura prima considerata a proposito delle liste, con l'esclusione di i' e j' :

$$\underline{\underline{L}} = (A, B, E, L, cons^L, car^L, cdr^L, isatom^L, eq^L, a_1^L, \ldots, j^L, i^L, ()^L, Nil^L)$$

ove

$A = \{a_1, \ldots\} \cup \{Nil\}$ (Atomi)

$B = \{True, False\}$ (Booleani)

$L_o = \{()\}$

$L_{i+1} = L_i \times L_i \cup A \times L_i$

$L = \bigcup_{i \in N} L_i$ (liste)

$E = A \cup L$ (espressioni)

$Nil^L = Nil$, $()^L = ()$, $a_i^L = a_i$ per i= 1,..

$cons^L$= operazione di accoppiamento cartesiano

car^L = prima proiezione di coppia cartesiana

cdr^L = seconda proiezione di coppia cartesiana

$isatom^L$, eq^L : usuali test booleani

i, j : immersioni di atomi e liste in espressioni

Possiamo affermare che ogni algebra isomorfa ad $\underline{\underline{L}}$ rappresenta bene il tipo di dato Liste e in tal senso la classe di isomorfismo di $\underline{\underline{L}}$ rappresenta un tipo di

dato astratto.

Notiamo subito la differenza tra la nozione prima vista di tipo di dato astratto e questa qui considerata. La prima si identifica con una teoria e quindi con la classe di tutte le algebre che la soddisfano, mentre la seconda si riferisce alla classe di isomorfismo di un algebra. Per rimarcare tale differenza nel secondo caso parleremo di tipo di dato astratto categorico.

Se $\underline{\underline{I}}$ è l'algebra iniziale relativa alla teoria prima considerata sulle liste,si può verificare facil= mente che $\underline{\underline{L}}$ è isomorfa a $\underline{\underline{I}}|_{\Sigma_o}$ ove Σ_o è la segna= tura di $\underline{\underline{I}}$ privata dagli operatori i' e j'.

In tal senso $\underline{\underline{I}}|_{\Sigma_o}$ dicesi specifica via iniziali= tà dell'algebra $\underline{\underline{L}}$ e quindi del tipo di dato astratto individuato da $\underline{\underline{L}}$.

Le dimostrazioni di isomorfismo tra $\underline{\underline{I}}|_{\Sigma_o}$ e $\underline{\underline{L}}$ co=tituiscono le dimostrazioni di correttezza della spe= cifica.

È evidente che la fondamentale differenza tra la definizione prima data di $\underline{\underline{L}}$ e quelle di $\underline{\underline{I}}$ è che la prima fa uso di nozioni matematiche usuali, mentre la seconda è data tramite una teoria algebrica ovvero in un linguaggio formale in cui si utilizzano solo le variabi= li, gli operatori e l'uguaglianza e si dispone di una ben precisa nozione di derivabilità (il calcolo equazio= nale) e di soddisfacibilità.

Notiamo che per specificare tramite inizialità una data Σ-algebra può essere utile introdurre degli ope= ratori non presenti in Σ che agevolano la caratteriz= zazione assiomatica. Per esempio se si vuole specificare per inizialità la Σ-algebra $\underline{\underline{N}}$ dei naturali con la somma $\underline{\underline{N}}$= (N,+) ($\Sigma = \{+\}$) è del tutto usuale conside= rare i seguenti assiomi nella segnatura $\Sigma' \supset \Sigma$ in cui

oltre la somma + , sono indicati lo zero e il succes= sore ' :

$$\mathcal{E} = \begin{cases} x + 0 = x \\ x + y' = (x + y)' \end{cases}$$

E evidente che

$$^{T}\Sigma/\bar{\mathcal{E}}^{*}\big|_{\Sigma} = \underline{\underline{N}}$$

Tali operatori aggiunti nella specifica e poi non con= siderati a costruzione avvenuta diconsi **operatori na= scosti.**

Nel caso della specifica per inizialità delle liste, i' e j' erano operatori nascosti, ma in tal ca= so la loro presenza nella specifica per inizialità po= teva essere evitata infatti se $\mathcal{E}_0$ sono gli assiomi delle liste senza quelli relativi a i'e j' si può verificare facilmente che

$$^{\underline{\underline{T}}}\Sigma_0/\bar{\mathcal{E}}^{*} = \underline{\underline{L}}$$

(Σ_0 segnatura di $\mathcal{E}_0$) .

61.- Estensioni di tipi di dato

La tecnica di specifica tramite inizialità può essere ovviamente estesa utilizzando le teorie basate secondo la proposizione i) del teorema fondamentale delle teorie basate(cfr. 59).

Inoltre tramite teorie basate si può precisare il concetto di "estensione" di tipo di dato:

consideriamo per esempio il tipo di dato astratto "vettori sui naturali" esso può essere identificato con la classe di isomorfismo dell'algebra

$$\underline{\underline{V}} = (V, N, -(-), -[-/-], \underline{0}, \emptyset)$$

ove N sono i "naturali"

$V = \{f \mid f: N \to N\}$

$-(-) = V \times N \to N$ **v(i)** individua il numero di posizione **i** nel vettore **v**

$-[-/-] : V \times N \times N \to V$ $(v[n/i])(j) = \begin{cases} v(j) & \text{se } i \neq j \\ n & \text{se } i = j \end{cases}$

$\underline{O}(i) = \emptyset \quad \forall\, i \in N$

Se volessimo specificare assiomaticamente tale algebra sarebbe naturale imporre i seguenti assiomi (la segnatura Σ' è evidente)

$$\underline{O}(i) = \emptyset \qquad \forall\, i \in N$$

$$(v[n/i])(i) = n \qquad \forall\, i \in N$$

$$(v[n/i])(j) = v(j) \qquad \forall\, i, j \in N\ ;\ i \neq j$$

Se ora consideriamo l'algebra iniziale $\underline{\underline{I}}$ di ta= le teoria basata ci si convince però che $\underline{\underline{I}}$ non è iso= morfa a $\underline{\underline{V}}$ poichè dagli assiomi di sopra non deriva

$$(\underline{O}\,[n/n])[m/m] = (\underline{O}\,[m/m])[n/n]$$

e quindi i due membri di tale equazione designano ele= menti diversi di $\underline{\underline{I}}$.

In effetti l'algebra $\underline{\underline{V}}$ è isomorfa all'algebra

$\underline{\underline{T}}_{\Sigma'/\theta}$ ove θ è la congruenza definita come segue.

Sia Σ la segnatura relativa alla base di $\underline{\underline{I}}$ contenente solo la sorta dei naturali e l'operatore nullario per lo zero ($\Sigma \subset \Sigma'$).

Diciamo **contesto basato** un termine **W(x)** di sorta in Σ con una variabile libera, poniamo quindi in $\underline{\underline{T}}_{\Sigma'}$

$$t_1 \theta t_2 \iff [W(t_1)]_\tau = [W(t_2)]_\tau$$

per ogni contesto basato $W(x)$ ove $W(x)$ è ottenuta sostituendo t al posto di x in $W(x)$, e τ è un assegnamento qualsiasi nell'algebra $\underline{\underline{I}}$ (qualsiasi poichè in $W(t_1)$ e $W(t_2)$ non vi sono variabili) .

Ovvero due termini sono considerati equivalenti se sono "osservabilmente" equivalenti sulla base, cioè se in $\underline{\underline{I}}$ (e quindi in tutte le algebre basate che verificano gli assiomi dati) il loro comportamento sulla base è identico (dal punto di vista algebrico).

Si dimostra che tale algebra $\underline{\underline{T}}_{\Sigma/\theta}$ è in effetti l'algebra finale nella classe di tutte le algebre minimali basate sui naturali. La proposizione iii) del teorema sulle teorie basate fornisce quindi un criterio per la specifica algebrica, via finalità delle estensioni astratte di tipo di dato.

62.- Implementazioni di tipi di dato

Un altro fenomeno interessante sui tipi di dato, esprimibile per via algebrica, è l'implementazione.

Data un Algebra $\underline{\underline{A}}$ diciamo che una algebra $\underline{\underline{D}}$ è un algebra derivata da $\underline{\underline{A}}$ se i domini di $\underline{\underline{D}}$ sono sottoinsiemi dei domini di $\underline{\underline{A}}$ e le operazioni di $\underline{\underline{D}}$ sono del tipo

$$\lambda x_1 \ldots x_n \; t(x_1 \ldots x_n)$$

ove $t(x_1 \ldots x_n)$ è un termine sull'algebra $\underline{\underline{A}}$ di variabili $x_1 \ldots x_n$.

Una implementazione di un tipo di dato astratto $\underline{\underline{C}}$ su $\underline{\underline{A}}$ è un algebra $\underline{\underline{D}}$ derivata da $\underline{\underline{A}}$ tale che $\underline{\underline{C}} \cong \underline{\underline{D}}$.
La dimostrazione dell'isomorfismo fornisce la correttezza dell'implementazione.

ESEMPIO di implementazione

Sia $\underline{\underline{\mathcal{S}}}$ il seguente tipo di dati

$\underline{\underline{\mathcal{S}}} = (\mathcal{S}, B, \text{delete}, \text{insert}, \text{set}, \text{has}, \text{eq}, \text{empty})$

ove

$B = \{\text{True}, \text{False}\}$

empty= l'insieme vuoto

$\mathcal{S}_0 = \{\text{empty}\}$

$\mathcal{S}_{i+1} = \mathcal{P}(\mathcal{S}_i) \cup \mathcal{S}_i$

$\mathcal{S} = \bigcup_{i \in N} \mathcal{S}_i$

$\begin{cases} \text{delete} : \mathcal{S} \times \mathcal{S} \to \mathcal{S} \\ \text{delete}(x,y) = y - \{x\} \end{cases}$

$\begin{cases} \text{insert} : \mathcal{S} \times \mathcal{S} \to \mathcal{S} \\ \text{insert}(x,y) = y \cup \{x\} \end{cases}$

$\begin{cases} \text{eq} : \mathcal{S} \times \mathcal{S} \to B \\ \text{eq}(x,y) = T \iff x = y \end{cases}$

$\begin{cases} \text{has} : \mathcal{S} \times \mathcal{S} \to B \\ \text{has}(x,y) \iff x \in y \end{cases}$

$\begin{cases} \text{set} : \mathcal{S} \to \mathcal{S} \\ \text{set}(x) = \{x\} \end{cases}$

Implementiamo in $\underline{\underline{\mathcal{S}}}$ il tipo di dati naturali, zero, successore $(N, 0, S)$

Basta definire $\underline{\underline{D}} = (D_N, 0^D, S^D)$ ponendo:

1) o) $\text{empty} \in D_N$

   i) $x \in D_N \Rightarrow x \cup \{x\} \in D_N$

2) $0^D = \text{empty}$

3) $S^D = \lambda x.\ \text{insert}(x, x)$

È evidente che $\underline{D}$ è isomorfa a $(N, 0, S)$ infatti $\underline{D}$ non è altro che il classico modello insiemistico di Von Neumann dei numeri naturali.

Una implementazione di un tipo di dato astratto $\mathfrak{E}$ su un tipo di dato $\underline{A}$ è un algebra $\underline{D}$ derivata da $\underline{A}$ che verifica $\mathfrak{E}$. La dimostrazione che $\underline{D} \models \mathfrak{E}$ costituisce la dimostrazione di correttezza dell'implementazione.

63.- Problemi di specifica formale

La problematica di specifica formale dei tipi di dato ha in sè tutti gli aspetti tipici presenti in generale nella specifica formale del software :

i) definizione formale di strutture e teorie che descrivano astrattamente nozioni programmative;
ii) traduzione di specifiche date, da un certo linguaggio in
iii) verifica di correttezza di una certa specifica nei confronti di un altra.

Diamo alcune veloci esemplificazioni di tale problematica.

Sia data una specifica (in linguaggio del 1º O) del seguente tipo di dato astratto <u>coda</u>

predicati: coda(-), elem(-), top(-,-), pop(-,-), push(-,-,-)

costanti: empty, a, b,....

assiomi:

$$\forall x\ (\text{coda}(x) \vee \text{elem}(x))$$

$$\text{elem}(a)$$

$$\text{elem}(b)$$

$$\vdots$$

$$\text{coda}(\text{empty})$$

$\forall x\ y\ z\ (push(x,y,z) \longleftrightarrow coda(x) \wedge elem(y) \wedge coda(z))$
$\forall\ x\ y\ \ (pop(x,y) \longleftrightarrow coda(x) \wedge \sim x=empty \wedge coda(y)\ )$
$\forall\ x\ y\ \ (top(x,y) \longleftrightarrow coda(x) \wedge \sim x=empty \wedge elem(y)\ )$
$\forall\ x\ ((coda(x) \wedge \sim x = empty) \longrightarrow (\exists\,!y(top(x,y) \wedge \exists\,!z(pop(x,z)))$
$\forall\ x\ y\ (push(empty,x,y) \longleftrightarrow top(y,x) \wedge pop(y,empty))$
$\forall\ x\ y\ z\ u\ v\ (pop(x,y) \wedge push(x,z,u) \wedge push(y,z,v) \longrightarrow pop(u,v))$
$\forall\ x\ y\ z\ u\ (top(x,y)\ \ push(x,z,u) \longrightarrow top(u,p))$
$\forall\ x\ y\ (coda(x) \wedge coda(y) \rightarrow (x=y) \longleftrightarrow (x=empty \wedge y=empty) \vee \exists uv(top(x,y) \wedge top(y;u) \acute{\wedge} pop(x,v) \wedge pop(y;v))$

PROBLEMI

I) **Specificare algebricamente tale teoria e dimostra= re la correttezza della specifica fornita, ovvero dimostrare una corrispondenza biunivoca tra i mo= delli delle due teorie.**

II) **Definire un modello della teoria algebrica data in i) verificando la correttezza di tali definizioni** cioé che i modelli verificano la teoria.

III) **Specificare algebricamente per inizialità alge= bre date in ii) e verificare la correttezza delle** specifiche.

IV) **Implementare** le **algebre definite in ii) sull'alge= bra delle liste e verificare la correttezza delle** implimentazioni.

V) **Definire per inizialità il tipo di dati naturali**

$$(N, B, +, \cdot, \emptyset, 1, \equiv, True, False)$$

(≡ predicato di uguaglianza tra numeri)

VI) **Definire per inizialità il tipo di dati stringhe su** $A = \{a, b\}$

$$\mathcal{J} = (A^*, \{True, False\}, \lambda, \equiv, \cup, a, b)$$

(≡ : uguaglianza tra stringhe)

VII) **1) Specificare algebricamente l'usuale tipo di**

dato file (su certi elementi A) con le opera=
zioni di get, put, reset , read , endoffile.

2) Specificare algebricamente un tipo di dato
matrici booleane 2x3 con l'operazione di somma
(componente per componente) estrazione di un
elemento da matrice, estrazione di riga ed
estrazione di colonna.