Specifiche Logico-Formali

Lezioni per il corso "Metodi per il trattamento dell'Informazione"
Versione 3.1
Corso di Laurea in Scienze dell'Informazione - a.a. 94-95

Egidio Astesiano - Gianna Reggio

In queste note useremo le seguenti notazioni.
Siano X, Y e Z insiemi

- $(X \to Y)$ indica l'*insieme delle funzioni totali* da X in Y;
- date due funzioni $f: X \to Y$ e $g: Y \to Z$, $g \circ f$ indica la *funzione composta* definita da $g \circ f(x) = g(f(x))$ per ogni $x \in X$;
- $Id^X: X \to X$ indica la funzione identità su X;
- data una funzione $f: X \to Y$ ed un insieme $K \subseteq X$, $f_{|K}$ indica la *restrizione* di f a K;
- una famiglia indiciata di insiemi $F = \{F_i\}_{i \in I}$ si dice *disgiunta* se gli insiemi F_i sono disgiunti;
- data una famiglia $F = \{F_i\}_{i \in I}$ quando ciò non causa ambiguità indichiamo con F anche l'insieme $\cup_{i \in I} F_i$;
- $\mathcal{P}(X)$ indica l'insieme delle parti di X;
- se R è una relazione di equivalenza su X, X/R indica l'insieme quoziente di X modulo R e la classe di un elemento $x \in X$ si indica con $[x]_R$.

1 Algebre con predicati (strutture del primo ordine)

In questa sezione presentiamo le strutture che servono da modelli semantici per le logiche del primo ordine e sono quindi dette anche *strutture del primo ordine* (first-order structures). Si tratta sostanzialmente di una variante delle algebre (eterogenee) in cui si aggiungono esplicitamente dei predicati (funzioni a valori di verità, ma senza l'introduzione esplicita della sort dei valori di verità).

1.1 Segnature ed algebre con predicati

Def. 1.1 Una *segnatura con predicati* Σ è una tripla

$$(S, OP, PR)$$

dove S è un insieme, dei simboli di *tipo*, $OP = \{OP_{w,s}\}_{w \in S^*, s \in S}$ una famiglia di insiemi, dei simboli di *operazione*, e $PR = \{PR_w\}_{w \in S^*}$ una famiglia di insiemi, dei simboli di *predicato*.

Un simbolo di tipo si dice anche *tipo* o *sort* e spesso, a meno di ambiguità si omette "simbolo di" davanti ad operazione e predicato.

Se $op \in OP_{w,s}$, diremo che op è un'operazione di *arità* w e *tipo* (o *sort*) s; la coppia (w, s) è detta *funzionalità* di op; se $pr \in PR_w$, diremo che pr è un predicato di *arità* w.

Se OP e PR sono disgiunte allora Σ si dice anche *senza overloading*. □

Da ora in poi Σ indica la segnatura con predicati (S, OP, PR).

Def. 1.2 Date due segnature con predicati Σ e Σ', un *morfismo* fra di esse è una tripla (f, g, h), con

$f: S \to S'$, $\quad g = \{g_{w,s}\}_{w \in S^*, s \in S}$ $\quad$ ed $\quad h = \{h_w\}_{w \in S^*}$

tali che $g_{w,s}: OP_{w,s} \to OP'_{f(w),f(s)}$ ed $h_w: PR_w \to PR'_{f(w)}$, dove si è posto $f(w) = f(s_1) \dots f(s_n)$ se $w = s_1 \dots s_n$.

Un morfismo (f, g, h) è un *isomorfismo* sse f, $g_{w,s}$ per ogni $w \in S^*$, $s \in S$ ed h_w per ogni $w \in S^*$ sono biunivoche.

Due segnature Σ e Σ' sono *isomorfe* sse esiste un isomorfismo fra di esse. □

Esempio 1.3 Segnatura con predicati per i numeri naturali.
Sia ΣN la segnatura con predicati (SN, OPN, PRN) dove:

- $SN = \{nat\}$,
- $OPN_{\Lambda,nat} = \{Zero\}$, $OPN_{nat,nat} = \{Succ\}$, $OPN_{nat\ nat,nat} = \{Add\}$ e $OPN_{w,nat} = \emptyset$ per ogni altro w;
- $PRN_{nat} = \{Pari\}$, $PRN_{nat\ nat} = \{Magg\}$ e $PRN_w = \emptyset$ per ogni altro w.

Fine esempio

Def. 1.4 Un'*algebra con predicati su* Σ o Σ-*algebra* è una tripla

$$\mathrm{A} = (\{\mathrm{A}_s\}_{s \in S}, \{I^{\mathrm{A}}_{w,s}\}_{w \in S^*, s \in S}, \{J^{\mathrm{A}}_w\}_{w \in S^*})$$

dove:

- per ogni $s \in S$ A_s è un insieme detto *supporto* o *carrier* della sort s;
- per ogni $w \in S^*$, $s \in S$ $I^{\mathrm{A}}_{w,s}$ è una funzione totale da $OP_{w,s}$ in $(\mathrm{A}_{s_1} \times \ldots \times \mathrm{A}_{s_n} \to \mathrm{A}_s)$ detta *funzione di interpretazione delle operazioni*;
- per ogni $w \in S^*$ J^{A}_w è una funzione totale da Pr_w in $\mathcal{P}(\mathrm{A}_{s_1} \times \ldots \times \mathrm{A}_{s_n})$ detta *funzione di interpretazione dei predicati.*

$I^{\mathrm{A}}_{w,s}(op)$ e $J^{\mathrm{A}}_w(pr)$ si indicano anche con $op^{\mathrm{A}}_{w,s}$ e pr^{A}_w; inoltre se $(a_1, \ldots, a_n) \in pr^{\mathrm{A}}_w$ si dice che $pr^{\mathrm{A}}_w(a_1, \ldots, a_n)$ vale.

La classe di tutte le Σ-algebre è indicata da Alg_Σ. □

Notazioni. Se $op \in OP_{w,s}$ e non c'è ambiguità, come avviene nelle segnature disgiunte, allora $I^{\mathrm{A}}_{w,s}(op)$ viene scritta semplicemente op^{A}; analogamente per i predicati; nel seguito useremo queste notazioni lasciando al lettore la definizione (valida sempre) con $I^{\mathrm{A}}_{w,s}$ e con J^{A}_w.

Notiamo che se A è una Σ-algebra, allora per ogni altra segnatura Σ' isomorfa a Σ, A si può vedere come Σ'-algebra (basta cambiare indiciatura sui carrier, sulle operazioni e sui predicati). Ciò corrisponde ad un "cambiamento di lessico" sulle sort, sulle operazioni e sui predicati.

Quando non ci interessa distinguere fra indiciature isomorfe, e l'unica caratteristica di una segnatura che vogliamo evidenziare è la sua "struttura" (la struttura degli operatori e dei predicati), si possono considerare algebre rispetto a segnature definite a meno di isomorfismi (di segnatura).

Esempio 1.5 Algebra con predicati per i numeri naturali
NAT è la ΣN algebra con predicati definita come segue, dove ΣN è segnatura con predicati definita in Es. 1.3:

- $\mathrm{NAT}_{nat} = \mathbb{N}$,
- $Zero^{\mathrm{NAT}} = 0$, $Succ^{\mathrm{NAT}}(n) = n + 1$ e $Add^{\mathrm{NAT}}(n, m) = n + m$;
- $Pari^{\mathrm{NAT}} = \{2 \cdot n \mid n \in \mathbb{N}\}$ e $Magg^{\mathrm{NAT}} = \{(n, m) \mid n, m \in \mathbb{N}$ e $n > m\}$.

Fine esempio

Def. 1.6 Date due Σ-algebre A e B, un Σ-*omomorfismo* (più semplicemente *omomorfismo*) $f\colon \mathrm{A} \to \mathrm{B}$ è una famiglia di funzioni: $f = \{f_s\}_{s \in S}$ tali che

- per ogni $s \in S$ è $f_s\colon \mathrm{A}_s \to \mathrm{B}_s$ totale;
- f rispetta la struttura dell'algebra (operazioni e predicati), cioè, per ogni $s_1 \ldots s_n \in S^*$, $n \geq 0$, $s \in S$:
  * per ogni $op \in OP_{s_1 \ldots s_n, s}$, $a_1 \in \mathrm{A}_{s_1}, \ldots, a_n \in \mathrm{A}_{s_n}$
    $f_s(op^{\mathrm{A}}(a_1, \ldots, a_n)) = op^{\mathrm{B}}(f_{s_1}(a_1), \ldots, f_{s_n}(a_n))$;
  * per ogni $pr \in PR_{s_1 \ldots s_n}$, $a_1 \in \mathrm{A}_{s_1}, \ldots, a_n \in \mathrm{A}_{s_n}$,
    se $(a_1, \ldots, a_n) \in pr^{\mathrm{A}}$ allora $(f_{s_1}(a_1), \ldots, f_{s_n}(a_n)) \in pr^{\mathrm{B}}$. $\square$

Prop. 1.7

1. Data una Σ-algebra A, $Id^{\mathrm{A}}\colon \mathrm{A} \to \mathrm{A}$, dove Id^{A} indica la famiglia delle funzioni identiche, cioè $Id^{\mathrm{A}} = \{Id^{\mathrm{A}_s}\}_{s \in S}$, è un omomorfismo.
2. Date le Σ-algebre A, B, C e gli omomorfismi $f\colon \mathrm{A} \to \mathrm{B}$ e $g\colon \mathrm{B} \to \mathrm{C}$, la composizione $g \circ f\colon \mathrm{A} \to \mathrm{C}$ è ancora un omomorfismo, dove la composizione di due famiglie di funzioni è la famiglia delle composizioni delle componenti, cioè $g \circ f = \{g_s \circ f_s\}_{s \in S}$.
3. $\circ$ è associativa.

Dim. 1 e 3 Immediato dalla definizioni.
Proviamo 2. Per ogni $s \in S$ si ha $g_s \circ f_s\colon \mathrm{A}_s \to \mathrm{C}_s$ totale;

- per ogni $op \in OP_{s_1 \ldots s_n, s}$, $a_1 \in \mathrm{A}_{s_1}, \ldots, a_n \in \mathrm{A}_{s_n}$
  $g_s \circ f_s(op^{\mathrm{A}}(a_1, \ldots, a_n)) = g_s(f_s(op^{\mathrm{A}}(a_1, \ldots, a_n))) = g_s(op^{\mathrm{B}}(f_{s_1}(a_1), \ldots, f_{s_n}(a_n))) =$
  $op^{\mathrm{C}}(g_{s_1} \circ f_{s_1}(a_1), \ldots, g_{s_n} \circ f_{s_n}(a_n))$;
- per ogni $pr \in PR_{s_1 \ldots s_n}$, $a_1 \in \mathrm{A}_{s_1}, \ldots, a_n \in \mathrm{A}_{s_n}$
  se $(a_1, \ldots, a_n) \in pr^{\mathrm{A}}$, allora $(f_{s_1}(a_1), \ldots, f_{s_n}(a_n)) \in pr^{\mathrm{B}}$ e quindi
  $(g_{s_1} \circ f_{s_1}(a_1), \ldots, g_{s_n} \circ f_{s_n}(a_n)) \in pr^{\mathrm{C}}$. $\square$

Def. 1.8 Siano A e B due Σ-algebre.

- Un omomorfismo $f\colon \mathrm{A} \to \mathrm{B}$ è un *isomorfismo* sse esiste un omomorfismo $g\colon \mathrm{B} \to \mathrm{A}$ tale che $f \circ g = Id^{\mathrm{B}}$ e $g \circ f = Id^{\mathrm{A}}$.
- A e B si dicono *isomorfe* (scritto $\mathrm{A} \cong \mathrm{B}$) sse esiste $g\colon \mathrm{A} \to \mathrm{B}$ isomorfismo; si noti che $\cong$ è una relazione di equivalenza.
- $f\colon \mathrm{A} \to \mathrm{B}$ è detto *iniettivo* sse per ogni $s \in S$ $f_s\colon \mathrm{A}_s \to \mathrm{B}_s$ è iniettiva.
- $f\colon \mathrm{A} \to \mathrm{B}$ è detto *surgettivo* sse per ogni $s \in S$ $f_s\colon \mathrm{A}_s \to \mathrm{B}_s$ è surgettiva.
- $f\colon \mathrm{A} \to \mathrm{A}$ è detto *endomorfismo*. $\square$

Si noti che $f\colon \mathrm{A} \to \mathrm{B}$ è un isomorfismo sse per ogni $s \in S$ $f_s\colon \mathrm{A}_s \to \mathrm{B}_s$ è bigettiva.

Def. 1.9 Sia $\mathcal{C}$ una classe di Σ-algebre ed A una Σ-algebra.

- A è *iniziale* in $\mathcal{C}$ sse $\mathrm{A} \in \mathcal{C}$ e per ogni $\mathrm{B} \in \mathcal{C}$ esiste un unico omomorfismo $f\colon \mathrm{A} \to \mathrm{B}$.
- A è *finale* in $\mathcal{C}$ sse $\mathrm{A} \in \mathcal{C}$ e per ogni $\mathrm{B} \in \mathcal{C}$ esiste un unico omomorfismo $f\colon \mathrm{B} \to \mathrm{A}$. □

Prop. 1.10 (L'elemento iniziale è caratterizzato a meno di isomorfismi)
Sia $\mathcal{C}$ una classe di Σ-algebre.

1. Se A e B sono iniziali in $\mathcal{C}$ allora $\mathrm{A} \cong \mathrm{B}$;
2. se A è iniziale in $\mathcal{C}$, $\mathrm{B} \in \mathcal{C}$ e $\mathrm{B} \cong \mathrm{A}$ allora anche B è iniziale in $\mathcal{C}$.

Dim. 1 Se A e B sono entrambi iniziali, allora esistono f e g tali che il seguente diagramma commuta:

$$\begin{tikzcd} \mathrm{A} \arrow[r, shift left, "f"] & \mathrm{B} \arrow[l, shift left, "g"] \end{tikzcd}$$

Per l'esistenza e l'unicità dell'omomorfismo da B in B, si ha che $f \circ g = Id^{\mathrm{B}}$; analogamente $g \circ f = Id^{\mathrm{A}}$.

2 Se $\mathrm{B} \cong \mathrm{A}$, allora esistono $g\colon \mathrm{A} \to \mathrm{B}$ e $g^{-1}\colon \mathrm{B} \to \mathrm{A}$ inversa di g. Inoltre, A è iniziale, per ogni $\mathrm{C} \in \mathcal{C}$ esiste unica $f\colon \mathrm{A} \to \mathrm{C}$. Quindi esiste $f'\colon \mathrm{B} \to \mathrm{C}$ (basta prendere infatti $f' = f \circ g^{-1}$).

Vediamo ora che è unica: se $h\colon \mathrm{B} \to \mathrm{C}$ allora deve essere $h \circ g = f$, quindi $h = h \circ g \circ g^{-1} = f \circ g^{-1} = f'$. □

Prop. 1.11 (Equivalente della Prop. 1.10 per gli oggetti finali).

1. Se A e B sono finali in $\mathcal{C}$, allora $\mathrm{A} \cong \mathrm{B}$;
2. se A è finale in $\mathcal{C}$, $\mathrm{B} \in \mathcal{C}$ e $\mathrm{B} \cong \mathrm{A}$ allora anche B è finale in $\mathcal{C}$.

Dim. Analoga a quella della Prop. 1.10. □

1.2 Algebra dei termini

Nel seguito assumeremo sempre che le stringhe su un alfabeto siano scritte nella rappresentazione orientata da sinistra a destra, senza ambiguità.

Def. 1.12 (Algebra dei termini su X in notazione funzionale)
Sia data una famiglia di insiemi di simboli (*variabili*) $X = \{X_s\}_{s \in S}$ indiciata su S tale che X non contiene simboli di OP e di PR.

Faremo sempre l'ipotesi che i simboli di parentesi aperta "(", chiusa ")" e la virgola "," non appartengano né a OP, né a PR né a X.

L'*algebra* $\mathrm{T}_\Sigma(X)$ *dei termini su* X è definita nel modo seguente.

- Per tutti gli $s \in S$ gli insiemi $(\mathrm{T}_\Sigma(X))_s \subseteq (X \cup OP \cup \{(,)\} \cup \{,\})^*$ sono definiti induttivamente da
  - $X_s \subseteq (\mathrm{T}_\Sigma(X))_s$,
  - per ogni $op \in OP_{s_1 \ldots s_n, s}$ $n \geq 0$, $t_1 \in (\mathrm{T}_\Sigma(X))_{s_1}, \ldots, t_n \in (\mathrm{T}_\Sigma(X))_{s_n}$
    $op(t_1, \ldots, t_n) \in (\mathrm{T}_\Sigma(X))_s$.

- Le operazioni sono cosí definite: per ogni $op \in OP_{s_1 \dots s_n, s}$, $n \geq 0$, $t_1 \in (\mathrm{T}_\Sigma(X))_{s_1}$, ..., $t_n \in (\mathrm{T}_\Sigma(X))_{s_n}$ $op^{\mathrm{T}_\Sigma(X)}(t_1, \dots, t_n) = op(t_1, \dots, t_n)$.
- I predicati sono così definiti: per ogni $pr \in PR_{s_1 \dots s_n}$ $n \geq 0$, $pr^{\mathrm{T}_\Sigma(X)} = \emptyset$.

Se per ogni $s \in S$ $X_s = \emptyset$, indicheremo $\mathrm{T}_\Sigma(X)$ con T_Σ; quest'ultima è anche detta *algebra dei termini senza variabili (o costanti o ground o chiusi)* oppure *word-algebra.* □

Quando non c'è ambiguità si indica con $\mathrm{T}_\Sigma(X)$ anche l'insieme di tutti i termini $\cup_{s \in S}(\mathrm{T}_\Sigma(X))_s$.

Prop. 1.13 (Decomposizione unica dei termini in notazione funzionale)
Sia X una famiglia disgiunta di variabili indiciata su S tale che X, OP e PR non hanno elementi in comune. Per ogni $t \in \mathrm{T}_\Sigma(X)$ vale una ed una sola delle seguenti asserzioni:

- esistono unici $s \in S$ ed $x \in X_s$ t.c. $t = x$;
- esistono unici $s, s_1, \dots, s_n \in S$, $n \geq 0$, $op \in OP_{s_1 \dots s_n, s}$ e $t_1, \dots, t_n \in \mathrm{T}_\Sigma(X)$ t.c. $t = op(t_1, \dots, t_n)$. □

Def. 1.14 Date una famiglia di variabili X indiciata su S ed un'algebra $\mathrm{A} \in Alg_\Sigma$, diremo *assegnamento di valori* (o *valutazione*) $v \colon X \to \mathrm{A}$ una famiglia di funzioni totali $\{v_s\}_{s \in S}$, tale che per ogni $s \in S$ $v_s \colon X_s \to \mathrm{A}_s$. □

Teorema 1.15 Date una famiglia di variabili X indiciata su S disgiunta, un'algebra $\mathrm{A} \in Alg_\Sigma$ ed un assegnamento $v \colon X \to \mathrm{A}$, esiste un unico omomorfismo $eval^{\mathrm{A},v} \colon \mathrm{T}_\Sigma(X) \to \mathrm{A}$ che estende v, cioè t.c. per ogni $s \in S$, $x \in X_s$ $eval^{\mathrm{A},v}(x) = v(x)$.
Dim. Da Prop. 1.13 per ogni $t \in (\mathrm{T}_\Sigma(X))_s$ si verifica una ed una sola delle seguenti alternative:

- esiste unico $x \in X_s$ t.c. $t = x$;
- esistono unici $s_1 \dots s_n$, $n \geq 0$, $op \in OP_{s_1 \dots s_n, s}$, $t_1 \in (\mathrm{T}_\Sigma(X))_{s_1}$, ..., $t_n \in (\mathrm{T}_\Sigma(X))_{s_n}$ t.c. $t = op(t_1, \dots, t_n)$.

Esistenza. Basta definire $eval^{\mathrm{A},v}$ come segue:

1. $eval^{\mathrm{A},v}(x) = v(x)$ per ogni $x \in X_s$,
2. $eval^{\mathrm{A},v}(op(t_1, \dots, t_n)) = op^{\mathrm{A}}(eval^{\mathrm{A},v}(t_1), \dots, eval^{\mathrm{A},v}(t_n))$
   per ogni $op \in OP_{s_1 \dots s_n, s}$, $t_1 \in (\mathrm{T}_\Sigma(X))_{s_1}$, ..., $t_n \in (\mathrm{T}_\Sigma(X))_{s_n}$.

Notiamo che $eval^{\mathrm{A},v}$ è ben definita come funzione, per la unicità della decomposizione (in base al principio di induzione strutturale), e che è un omomorfismo. (Nota: a rigore avremo dovuto definire $eval^{\mathrm{A},v}$ come una famiglia di funzioni quantificando esplicitamente le equazioni 1 ed 2 sulle sort.)

Unicità Basta osservare che un qualsiasi omomorfismo $f \colon \mathrm{T}_\Sigma(X) \to \mathrm{A}$ deve soddisfare 2, e che se f estende v, allora deve soddisfare anche 1. Quindi, per l'esistenza e unicità della decomposizione dei termini, l'omomorfismo che estende v è unico. □

Nota 1. Per il Teor. 1.15 $\mathrm{T}_\Sigma(X)$ è detta algebra *assolutamente libera su* X perché non deve rispettare alcun vincolo o legame di uguaglianza tra i termini eccetto quelle dovute alla struttura. (Ed è per questo motivo che, per ogni assegnamento di valori in A alle variabili di X, esiste un (unico) omomorfismo che lo estende).

Nota 2. Si vede subito dalla dimostrazione che si può attenuare l'ipotesi di disgiunzione su Σ e X; per ottenere il risultato basta infatti assumere che per ogni sort s ed ogni $t \in (\mathrm{T}_\Sigma(X))_s$ esistono unici $s_1, \ldots, s_n \in S$, $op \in OP_{s_1\ldots s_n,s}$, $t_1 \in (\mathrm{T}_\Sigma(X))_{s_1}$, $\ldots$, $t_n \in (\mathrm{T}_\Sigma(X))_{s_n}$ t.c. $t = op(t_1, \ldots, t_n)$. In questo caso si dice che i termini non sono ambigui rispetto al tipo.

Notazioni. Indicheremo con $t^{\mathrm{A},v}$ il valore $eval^{\mathrm{A},v}(t)$ e, se X è la famiglia vuota, con t^{A} o $eval^{\mathrm{A}}(t)$ il valore $eval^{\mathrm{A},v}(t)$.

Cor. 1.16 Se Σ è disgiunta, allora

1. T_Σ è iniziale in Alg_Σ.
2. Un'algebra è iniziale in Alg_Σ sse è isomorfa a T_Σ.

Dim. 1 Ricordando che $\mathrm{T}_\Sigma = \mathrm{T}_\Sigma(\emptyset)$, basta osservare che per ogni Σ-algebra A esiste unico $v\colon \emptyset \to \mathrm{A}$, e che quindi esiste unico $f\colon \mathrm{T}_\Sigma \to \mathrm{A}$, infatti basta porre $f = eval^{\mathrm{A},v} = eval^{\mathrm{A}}$ ed osservare che ogni altro omomorfismo $g\colon \mathrm{T}_\Sigma \to \mathrm{A}$ deve estendere v e quindi deve coincidere con $eval^{\mathrm{A}}$. Più formalmente: $f\colon \mathrm{T}_\Sigma \to \mathrm{A} \Leftrightarrow f$ estende $v\colon \emptyset \to \mathrm{A} \Leftrightarrow f = eval^{\mathrm{A}}$.

2 È una conseguenza immediata di 1 e di Prop. 1.10. □

Def. 1.17 Sia X una famiglia indiciata su S. Una *sostituzione* sui termini costruiti su Σ ed X è un endomorfismo $h\colon \mathrm{T}_\Sigma(X) \to \mathrm{T}_\Sigma(X)$.

Una sostituzione $h\colon \mathrm{T}_\Sigma(X) \to \mathrm{T}_\Sigma(X)$ è:

- *finita* sse esiste $V \subseteq X$ finito t.c. per ogni $x \in X$ $h(x) \neq x$ sse $x \in V$;
- *unitaria* sse esiste al più un $x \in X$ t.c. $h(x) \neq x$. □

Prop. 1.18 (Ogni sostituzione è determinata in modo unico dai termini sostituiti alle variabili) Siano Σ ed X disgiunte.

1. Se $v\colon X \to \mathrm{T}_\Sigma(X)$ è un assegnamento, allora esiste un unico endomorfismo $h\colon \mathrm{T}_\Sigma(X) \to \mathrm{T}_\Sigma(X)$ che estende v.
2. Se $h\colon \mathrm{T}_\Sigma(X) \to \mathrm{T}_\Sigma(X)$ è un endomorfismo, allora esiste unico un assegnamento $v\colon X \to \mathrm{T}_\Sigma(X)$ t.c. $eval^{\mathrm{T}_\Sigma(X),v} = h$.

Dim. 1 Basta applicare Teor. 1.15.

2 Definiamo $v = \{h_{s|X_s}\}_{s\in S}$, chiaramente v è un assegnamento da X in $\mathrm{T}_\Sigma(X)$.

Sia $v'\colon X \to \mathrm{T}_\Sigma(X)$ un altro assegnamento t.c. $eval^{\mathrm{T}_\Sigma(X),v'} = h$, allora dalla definizione di $eval$ (ved. Teor. 1.15) si ha che per ogni $x \in X_s$ $v'_s(x) = (eval^{\mathrm{T}_\Sigma(X),v'})_s(x) = h_s(x) = v_s(x)$. □

Prop. 1.19 Sia X una famiglia indiciata su S ed $h\colon \mathrm{T}_\Sigma(X) \to \mathrm{T}_\Sigma(X)$ sia una sostituzione finita diversa dall'identità tale che per ogni x, x' $(x \neq x')$ t.c. $x \neq h(x)$ e $x' \neq h(x')$, si ha $x \notin Var(h(x'))$. Allora h è esprimibile come composizione di sostituzioni unitarie.

Dim. Omessa

Nota: Nel seguito faremo sempre l'ipotesi che Σ sia disgiunta. Si noti che ci si può sempre ricondurre a questo caso, passando alla segnatura $\Sigma^+ = (S, OP^+, PR^+)$ in cui $OP^+_{w,s} = OP_{w,s} \times \{(w,s)\}$, cioè in cui ad ogni operatore è associata la sua funzionalità e $PR^+_w = PR_w \times \{w\}$, cioè in cui ad ogni predicato è associata la sua arità. Si vede subito che Σ^+ è isomorfa a Σ; perciò ogni Σ-algebra è una Σ^+-algebra e viceversa. Inoltre essendo Σ^+ disgiunta, T_{Σ^+} è iniziale nella classe delle Σ-algebre. Perciò le proposizioni successive valgono ancora per Σ qualunque purchè si sostituisca T_{Σ^+} a T_Σ.

1.3 Sottoalgebre e algebre term-generated (o induttive o minimali)

Def. 1.20 Data $\mathrm{A} \in Alg_\Sigma$, sia B una famiglia di insiemi $\{B_s\}_{s\in S}$ tale che:

1. per ogni $s \in S$ $B_s \subseteq \mathrm{A}_s$.
2. per ogni $op \in OP_{s_1\ldots s_n,s}$, $n \geq 0$, $b_1 \in B_{s_1}$, ..., $b_n \in B_{s_n}$, $op^\mathrm{A}(b_1,\ldots,b_n) \in B_s$ (cioè i B_s sono chiusi rispetto alle operazioni).

Allora la Σ-algebra

$$\mathrm{B} = (\{B_s\}_{s\in S}, \{op^\mathrm{B}\}_{op\in OP}, \{pr^\mathrm{B}\}_{pr\in PR})$$

dove $op^\mathrm{B} = op^\mathrm{A}_{|B_{s_1}\times\ldots\times B_{s_n}}$ per ogni $op \in OP_{s_1\ldots s_n,s}$ e $pr^\mathrm{B} = pr^\mathrm{A} \cap (B_{s_1} \times \ldots \times B_{s_n})$, si dice *sottoalgebra* di A.

Indicheremo il fatto che B è una sottoalgebra di A con $\mathrm{B} \subseteq \mathrm{A}$. □

Def. 1.21 Data una famiglia $B = \{B_s\}_{s\in S}$, tale che per ogni $s \in S$ $B_s \subseteq \mathrm{A}_s$, diremo *chiusura induttiva di B in* A la famiglia $B_\mathrm{A} = \{(B_\mathrm{A})_s\}_{s\in S}$ definita come la famiglia dei minimi insiemi che soddisfano:

1. per ogni $s \in S$ $B_s \subseteq (B_\mathrm{A})_s$,
2. per ogni $op \in OP_{s_1\ldots s_n,s}, n \geq 0$, $b_1 \in (B_\mathrm{A})_{s_1}, \ldots, b_n \in (B_\mathrm{A})_{s_n}$, $op^\mathrm{A}(b_1,\ldots,b_n) \in (B_\mathrm{A})_s$. □

Prop. 1.22 Con le notazioni della Def. 1.21 B_A è la famiglia di supporti di una Σ-sottoalgebra di A, che indicheremo con $\mathrm{B_A}$, ed è la minima che include B.
Dim. Per induzione in base a Def. 1.21 1, 1.21 2 e da $B_s \subseteq \mathrm{A}_s$ si prova che $(B_\mathrm{A})_s \subseteq \mathrm{A}_s$ e quindi vale Def. 1.20 1 e per 1.21 2 è verificata anche Def. 1.20 2 e quindi $\mathrm{B_A}$ è una sottoalgebra di A ed è per definizione la minima che include B, cioè B_A è la minima famiglia di supporti includente B. □

Def. 1.23 Se $\mathrm{B_A} \cong \mathrm{A}$, allora B è detto *sistema di generatori* per A ed A si dice *generata (induttivamente) da B*. □

Def. 1.24 $\mathrm{A} \in Alg_\Sigma$ è detta *generata dai termini (term-generated) o minimale o induttiva* sse $\mathrm{A} = \emptyset_\mathrm{A}$ (dove con $\emptyset$ si intende la famiglia di insiemi vuoti). Indicheremo con Gen_Σ la classe $\{\mathrm{A} \in Alg_\Sigma \mid \mathrm{A}$ è term-generated $\}$. □

Prop. 1.25 Sia $h\colon \mathrm{A} \to \mathrm{B}$ un omomorfismo. Si ha che sulla famiglia $h(\mathrm{A}) = \{h_s(\mathrm{A}_s)\}_{s\in S}$ è definibile una Σ-sottoalgebra di B che indicheremo ancora con $h(\mathrm{A})$.
Dim. Basta provare che $h(\mathrm{A})$ è chiusa rispetto alle operazioni.

Per ogni $op \in OP_{s_1\ldots s_n,s}$, $n \geq 0$, $b_1 \in h_{s_1}(\mathrm{A}_{s_1}), \ldots, b_n \in h_{s_n}(\mathrm{A}_{s_n})$, per definizione di $h_{s_i}(\mathrm{A}_{s_i})$ si ha che esistono $a_1 \in \mathrm{A}_{s_1}$, ..., $a_n \in \mathrm{A}_{s_n}$ tali che $b_i = h_{s_i}(a_i)$; quindi
$op^\mathrm{B}(b_1,\ldots,b_n) = op^\mathrm{B}(h_{s_1}(a_1),\ldots,h_{s_n}(a_n)) = h_s(op^\mathrm{A}(a_1,\ldots,a_n)) \in h_s(\mathrm{A}_s)$. □

Prop. 1.26

1. A $\in Alg_\Sigma$ è term-generated sse non ammette sottoalgebre proprie (cioè diverse da se stessa).

2. Data A $\in Alg_\Sigma$, si ha che $\emptyset_{\mathrm{A}} = \{\{t^{\mathrm{A}} \mid t \in (\mathrm{T}_\Sigma)_s\}\}_{s \in S} = eval^{\mathrm{A}}(\mathrm{T}_\Sigma)$; cioè A è term-generated sse $eval^{\mathrm{A}}$ è surgettivo.

3. Se A, B $\in Alg_\Sigma$, A è term-generated e $h{:}\, \mathrm{A} \to \mathrm{B}$, allora l'algebra immagine $h(\mathrm{A})$ è term-generated.

Dim. 1, 2 immediate dalle definizioni; 3 $eval^{h(\mathrm{A})}$ deve coincidere con $h \circ eval^{\mathrm{A}}$ e quindi è surgettivo; da cui per 2 si ha l'asserto. □

Prop. 1.27 Siano A, B $\in Alg_\Sigma$.

1. Se $h{:}\, \mathrm{A} \to \mathrm{B}$, allora per ogni $t_1, \ldots, t_n \in \mathrm{T}_\Sigma$ $h(t_1^{\mathrm{A}}) = t_1^{\mathrm{B}}$ e per ogni $pr \in PR$ se $pr^{\mathrm{A}}(t_1^{\mathrm{A}}, \ldots, t_n^{\mathrm{A}})$ vale allora $pr^{\mathrm{B}}(h(t_1^{\mathrm{A}}), \ldots, h(t_n^{\mathrm{A}})) = pr^{\mathrm{B}}(t_1^{\mathrm{B}}, \ldots, t_n^{\mathrm{B}})$ vale;

2. se A $\in Gen_\Sigma$, esiste al più un omomorfismo da A in B;

3. se B $\in Gen_\Sigma$ ed $h{:}\, \mathrm{A} \to \mathrm{B}$, allora h è surgettivo;

4. se $\mathrm{A}, \mathrm{B} \in Gen_\Sigma$ e h è una famiglia di funzioni di A in B, allora h è un omomorfismo sse per ogni $t_1, \ldots, t_n \in \mathrm{T}_\Sigma$ $h(t_1^{\mathrm{A}}) = t_1^{\mathrm{B}}$ e per ogni $pr \in PR$ se $pr^{\mathrm{A}}(t_1^{\mathrm{A}}, \ldots, t_n^{\mathrm{A}})$ vale allora $pr^{\mathrm{B}}(t_1^{\mathrm{B}}, \ldots, t_n^{\mathrm{B}}) vale$;

5. se $\mathrm{A}, \mathrm{B} \in Gen_\Sigma$, $h_1{:}\, \mathrm{A} \to \mathrm{B}$ e $h_2{:}\, \mathrm{B} \to \mathrm{A}$, allora A e B sono isomorfe.

Dim. 1 Per induzione strutturale.
2, 3 Conseguenza immediata di 1.
4 $\Rightarrow$ vale per 1; $\Leftarrow$ facile verifica che h è un omomorfismo, essendo A, B $\in Gen_\Sigma$.
5 $h_2 \circ h_1$ è un omomorfismo da A in A e quindi coincide con l'identità (o direttamente da 1 si ha $h_2 \circ h_1(t^{\mathrm{A}}) = t^{\mathrm{A}}$); quindi h_1, surgettiva per 3, è anche iniettiva e cioè un isomorfismo. □

Concludiamo questa sezione con una caratterizzazione dell'algebra iniziale nella classe delle Σ-algebre che è indipendente dalla rappresentazione dei termini e vale inoltre anche nel caso di segnature con overloading.

Prop. 1.28 Sia Σ una segnatura anche non disgiunta. Allora A è iniziale nella classe delle Σ-algebre sse A è term-generated e

(+) per ogni $s \in S$, ogni elemento di A_s ammette unica decomposizione nelle sottocomponenti con relative funzionalità, cioè per ogni $a \in \mathrm{A}_s$ esistono unici op, $s_1, \ldots, s_n \in S$, $n \geq 0$, $a_1 \in \mathrm{A}_{s_1}$, ..., $a_n \in \mathrm{A}_{s_n}$ t.c. $a = op^{\mathrm{A}}(a_1, \ldots, a_n)$;

(∗) $pr^{\mathrm{A}} = \emptyset$ per ogni $pr \in PR$.

Dim. Si ricordi che Σ^+ definita precedentemente è disgiunta, per cui T_{Σ^+} è iniziale nella classe delle Σ^+ algebre (che coincide con la classe delle Σ-algebre).

$\Rightarrow$ Se A è iniziale nelle Σ-algebre, allora è isomorfa a T_{Σ^+} e quindi, come immagine omomorfa di un'algebra term-generated, è term-generated essa stessa, per la Prop. 1.26 3. Inoltre le proprietà (+) e (∗) valgono per T_{Σ^+} e dunque, sempre per l'isomorfismo fra A e T_{Σ^+}, valgono anche per A.

$\Leftarrow$ Sia B una Σ-algebra; occorre provare che esiste unico un omomorfismo $h\colon \mathrm{A} \to \mathrm{B}$. Essendo A term-generated, per la Prop. 1.27 2, esiste al più un solo tale h.

D'altra parte, per (+) e (*) la seguente è la definizione di una famiglia di funzioni $\{h_s\}$, che per costruzione soddisfa le condizioni di omomorfismo:

$h_s(op^{\mathrm{A}}(a_1, \ldots, a_n)) = op^{\mathrm{B}}(h_{s_1}(a_1), \ldots, h_{s_n}(a_n))$ per ogni $op \in OP_{s_1 \ldots s_n, s}$. □

1.4 Congruenze, algebre quozienti, congruenze associate ad omomorfismi

Def. 1.29 Sia data $\mathrm{A} \in Alg_\Sigma$ ed una coppia

$$R = (\{R_s\}_{s \in S}, \{R_{pr}\}_{pr \in PR})$$

tale che

- per ogni $s \in S$ R_s è una relazione di equivalenza su A_s,
- per ogni $pr \in PR_{s_1 \ldots s_n}$ $R_{pr} \subseteq A_{s_1} \times \ldots A_{s_n}$.

Diremo che R è una *congruenza* su A sse rispetta le operazioni e la verità dei predicati, cioè:

- per ogni $op \in OP_{s_1 \ldots s_n, s}$, $a_1, a'_1 \in \mathrm{A}_{s_1}, \ldots, a_n, a'_n \in \mathrm{A}_{s_n}$, $a_1\ R_{s_1}\ a'_1, \ldots, a_n\ R_{s_n}\ a'_n$ implica $op^{\mathrm{A}}(a_1, \ldots, a_n)\ R_s\ op^{\mathrm{A}}(a'_1, \ldots, a'_n)$ (rispetta le operazioni);
- per ogni $pr \in PR$ $pr^{\mathrm{A}} \subseteq R_{pr}$ e

  per ogni $pr \in PR_{s_1 \ldots s_n}$, $a_1, a'_1 \in \mathrm{A}_{s_1}, \ldots, a_n, a'_n \in \mathrm{A}_{s_n}$,

  se $a_1\ R_{s_1}\ a'_1, \ldots, a_n\ R_{s_n}\ a'_n$ e $(a_1, \ldots, a_n) \in R_{pr}$, allora $(a'_1, \ldots, a'_n) \in R_{pr}$

  (rispetta la verità dei predicati). □

Def. 1.30 Data $\mathrm{A} \in Alg_\Sigma$ ed una congruenza R su A, diremo *algebra quoziente su* A *modulo* R, e la indicheremo con A/R, l'algebra i cui carrier sono: $\{\mathrm{A}_s/R_s\}_{s \in S}$ e le cui operazioni e predicati sono così definiti:

- per ogni $op \in OP_{s_1 \ldots s_n, s}$, $n \geq 0$, $a_1 \in \mathrm{A}_{s_1}, \ldots, a_n \in \mathrm{A}_{s_n}$
  $op^{\mathrm{A}/R}([a_1]_R, \ldots, [a_n]_R) = [op^{\mathrm{A}}(a_1, \ldots, a_n)]_R$;
- per ogni $pr \in PR_{s_1 \ldots s_n}$, $n \geq 0$, $pr^{\mathrm{A}/R} = \{([a_1]_R, \ldots, [a_n]_R) \mid (a_1, \ldots, a_n) \in R_{pr}\}$.

Notiamo che questa è una buona definizione, poiché non dipende dagli a_i scelti come rappresentanti delle classi di equivalenza, essendo R una congruenza. □

Teorema 1.31 Siano A, B elementi di Alg_Σ e sia $f\colon \mathrm{A} \to \mathrm{B}$ un omomorfismo. Definiamo la relazione

$$\equiv_f = (\{\{(a, a') \mid f_s(a) = f_s(a')\}\}_{s \in S}, \{\{(a_1, \ldots, a_n) \mid (f_{s_1}(a_1), \ldots, f_{s_n}(a_n)) \in pr^{\mathrm{B}}\}\}_{pr \in PR})$$

(cioè due elementi sono in relazione per $\equiv_f$ sse hanno la stessa immagine per f ed un predicato vale su degli elementi sse vale in B sulle loro immagini rispetto ad f).

Si ha che $\equiv_f$ è una congruenza su A (*congruenza associata all'omomorfismo* f, detta anche *nucleo (kernel)* di f).

Dim. Si prova facilmente che $\equiv_f$ è una relazione di equivalenza. Proviamo la proprietà di congruenza.

- Sia $op \in OP_{s_1 \dots s_n, s}$; se $a_1 (\equiv_f)_{s_1} a'_1, \dots, a_n (\equiv_f)_{s_n} a'_n$, cioè se $f_{s_1}(a_1) = f_{s_1}(a'_1), \dots, f_{s_n}(a_n) = f_{s_n}(a'_n)$ allora per la definizione di omomorfismo

  $f_s(op^{\mathrm{A}}(a_1, \dots, a_n)) = op^{\mathrm{B}}(f_{s_1}(a_1), \dots, f_{s_n}(a_n)) = op^{\mathrm{B}}(f_{s_1}(a'_1), \dots, f_{s_n}(a'_n)) =$

  $f_s(op^{\mathrm{A}}(a'_1, \dots, a'_n))$ e quindi $op^{\mathrm{A}}(a_1, \dots, a_n) \ (\equiv_f)_s \ op^{\mathrm{A}}(a'_1, \dots, a'_n)$.

- Sia $pr \in PR$;

  se $(a_1, \dots, a_n) \in pr^{\mathrm{A}}$, allora $(f_{s_1}(a_1), \dots, f_{s_n}(a_n)) \in pr^{\mathrm{B}}$ e quindi $(a_1, \dots, a_n) \in (\equiv_f)_{pr}$; inoltre se $(a_1, \dots, a_n) \in (\equiv_f)_{pr}$ e $a_1 (\equiv_f)_{s_1} a'_1, \dots, a_n (\equiv_f)_{s_n} a'_n$, allora per la definizione di omomorfismo $(f_{s_1}(a_1), \dots, f_{s_n}(a_n)) \in pr^{\mathrm{B}}$, da cui essendo anche $f_{s_i}(a_i) = f_s(a'_i)$ per $i = 1, \dots, n$, si ha $(f_{s_1}(a'_1), \dots, f_{s_n}(a'_n)) \in pr^{\mathrm{B}}$, cioè $(a'_1, \dots, a'_n) \in (\equiv_f)_{pr}$. $\square$

Teorema 1.32 Siano A, B $\in Alg_\Sigma$ ed $h \colon \mathrm{A} \to \mathrm{B}$ un omomorfismo.

1. *Fattorizzazione epi-mono o primo teorema di omomorfismo.* Definiamo $\pi_h \colon \mathrm{A} \to \mathrm{A}/{\equiv_h}$ tale che $(\pi_h)_s(a) = [a]$ (cioè π_h è l'omomorfismo surgettivo che trasforma un elemento nella sua classe di equivalenza).

   Allora esiste un unico $g \colon A/{\equiv_h} \to \mathrm{B}$ iniettivo che rende commutativo il seguente diagramma:

$$\begin{tikzcd} \mathrm{A} \arrow[r, "h"] \arrow[d, "\pi_h"'] & \mathrm{B} \\ \mathrm{A}/{\equiv_h} \arrow[ur, "g"'] & \end{tikzcd}$$

2. Se h è surgettivo ed $f \colon \mathrm{A} \to \mathrm{C}$ è un omomorfismo, allora esiste $k \colon \mathrm{B} \to \mathrm{C}$ che rende commutativo il seguente diagramma sse $\equiv_h \subseteq \equiv_f$; inoltre k, se esiste, è unico.

$$\begin{tikzcd} \mathrm{A} \arrow[r, "h"] \arrow[d, "f"'] & \mathrm{B} \arrow[dl, "k"] \\ \mathrm{C} & \end{tikzcd}$$

Dim.

1 Per ogni $s \in S$, $a \in \mathrm{A}_s$ definiamo $g_s([a]) = h_s(a)$.

- g è un omomorfismo, infatti:
  - per ogni $op \in OP_{s_1 \dots s_n, s}$, $n \geq 0$, $a_1 \in \mathrm{A}_{s_1}, \dots, a_n \in \mathrm{A}_{s_n}$
    $g_s(op^{\mathrm{A}/\equiv_h}([a_1], \dots, [a_n])) = g_s([op^{\mathrm{A}}(a_1, \dots, a_n)]) = h_s(op^{\mathrm{A}}(a_1, \dots, a_n)) =$
    $op^{\mathrm{B}}(h_{s_1}(a_1), \dots, h_{s_n}(a_n)) = op^{\mathrm{B}}(g_{s_1}([a_1]), \dots, g_{s_n}([a_n]))$;
  - per ogni $pr \in PR_{s_1 \dots s_n}$, $n \geq 0$, $a_1 \in \mathrm{A}_{s_1}, \dots, a_n \in \mathrm{A}_{s_n}$
    $([a_1], \dots, [a_n]) \in pr^{\mathrm{A}/\equiv_h} \Rightarrow (a_1, \dots, a_n) \in (\equiv_h)_{pr} \Rightarrow (h_{s_1}(a_1), \dots, h_{s_n}(a_n)) \in pr^{\mathrm{B}}$;
    allora $(g_{s_1}([a_1]), \dots, g_{s_n}([a_n])) = (h_{s_1}(a_1), \dots, h_{s_n}(a_n)) \in pr^{\mathrm{B}}$.
- g è unico, infatti ogni altro omomorfismo f che rende commutativo il diagramma deve soddisfare $f_s([a]) = h_s(a) = g_s([a])$.

– g è iniettivo, poiché $g_s([a]) = g_s([a']) \Rightarrow h_s(a) = h_s(a') \Rightarrow a \equiv_h a' \Rightarrow [a] = [a']$.

2

"$\Leftarrow$" Sia $\equiv_h \subseteq \equiv_f$; per ogni $s \in S$ definiamo

(#) $\quad k_s(b) = f_s(a)$ se $h_s(a) = b$ per ogni $b \in \mathrm{B}_s$,

k è totale, infatti tale a esiste sempre perchè h è surgettiva.

k è ben definita, infatti, $b = h_s(a) = h_s(a') \Rightarrow a \equiv_h a' \Rightarrow a \equiv_f a' \Rightarrow f_s(a) = f_s(a')$.

k è un omomorfismo, infatti:

- per ogni $op \in OP_{s_1 \ldots s_n, s}$, $n \geq 0$, $b_1 \in \mathrm{B}_{s_1}, \ldots, b_n \in \mathrm{B}_{s_n}$, essendo h surgettiva esistono $a_1, \ldots, a_n$ tali che $h_{s_1}(a_1) = b_1, \ldots, h_{s_n}(a_n) = b_n$;
  $k(op^{\mathrm{B}}(b_1, \ldots, b_n)) = k(op^{\mathrm{B}}(h_{s_1}(a_1), \ldots, h_{s_n}(a_n)))$ = essendo h un omomorfismo $= k(h_s(op^{\mathrm{A}}(a_1, \ldots, a_n)))$ = per definizione di $k = f_s(op^{\mathrm{A}}(a_1, \ldots, a_n)) =$ essendo f un omomorfismo $= op^{\mathrm{C}}(f_{s_1}(a_1), \ldots, f_{s_n}(a_n))$ = per definizione di $k =$ $op^{\mathrm{C}}(k_{s_1}(b_1), \ldots, k_{s_n}(b_n))$.
- Per ogni $pr \in PR_{s_1 \ldots s_n}$, $n \geq 0$, $b_1 \in \mathrm{B}_{s_1}, \ldots, b_n \in \mathrm{B}_{s_n}$,
  sia $(b_1, \ldots, b_n) \in pr^{\mathrm{B}}$, essendo h surgettiva esistono $a_1, \ldots, a_n$ tali che $h_{s_1}(a_1) = b_1, \ldots, h_{s_n}(a_n) = b_n$;
  $(k_{s_1}(b_1), \ldots, k_{s_n}(b_n)) = (k_{s_1}(h_{s_1}(a_1)), \ldots, k_{s_n}(h_{s_1}(a_n)))$ = per definizione di k $= (f_{s_1}(a_1), \ldots, f_{s_n}(a_n))$; essendo f un omomorfismo e dato che $(a_1, \ldots, a_n) \in (\equiv_h)_{pr} \subseteq (\equiv_f)_{pr}$ si ha che $(f_{s_1}(a_1), \ldots, f_{s_n}(a_n)) \in pr^{\mathrm{C}}$.

È facile vedere che k rende commutativo il diagramma.

"$\Rightarrow$" Sia $a \equiv_h a'$, allora $h(a) = h(a')$, $f(a) = k(h(a)) = k(h(a')) = f(a')$ e quindi $a \equiv_f a'$.
Sia $(a_1, \ldots, a_n) \in (\equiv_h)_{pr}$, allora $(h_{s_1}(a_1), \ldots, h_{s_n}(a_n)) \in pr^{\mathrm{B}}$,
$(k_{s_1}(h_{s_1}(a_1)), \ldots, k_{s_n}(h_{s_n}(a_n))) \in pr^{\mathrm{C}}$, cioè $(f_{s_1}(a_1), \ldots, f_{s_n}(a_n)) \in pr^{\mathrm{C}}$, e quindi $(a_1, \ldots, a_n) \in (\equiv_f)_{pr}$.

Infine osserviamo che k è unico, infatti ogni altro omomorfismo che rende commutativo il diagramma deve soddisfare (#). $\square$

<u>Notazioni.</u> nel seguito $\equiv_{eval^{\mathrm{A}}}$ verrà abbreviato $\equiv_{\mathrm{A}}$.

Prop. 1.33

1. Se R è una congruenza su T_Σ, allora $\mathrm{T}_\Sigma / R \in Gen_\Sigma$.
2. $\mathrm{A} \in Gen_\Sigma$ sse $\mathrm{A} \cong \mathrm{T}_\Sigma / \equiv_{\mathrm{A}}$.

Dim. 1 Immediata; vediamo 2.

$\Rightarrow$ Sappiamo che esiste unica $eval^{\mathrm{A}} \colon \mathrm{T}_\Sigma \to \mathrm{A}$ e che esiste unica $g \colon \mathrm{T}_\Sigma / \equiv_{\mathrm{A}} \to \mathrm{A}$ iniettiva che rende commutativo il diagramma:

$$\begin{tikzcd} \mathrm{T}_\Sigma \arrow[r, "eval^{\mathrm{A}}"] \arrow[d, "\pi"'] & \mathrm{A} \\ \mathrm{T}_\Sigma / \equiv_{\mathrm{A}} \arrow[ur, "g"'] & \end{tikzcd}$$

Inoltre, se $\mathrm{A} \in Gen_\Sigma$, allora g è anche surgettiva e quindi g è bigettiva.

$\Leftarrow$ Se $\mathrm{A} \cong \mathrm{T}_\Sigma/{\equiv_\mathrm{A}}$, allora la (unica) g che fa commutare il diagramma deve essere surgettiva, quindi per ogni $a \in \mathrm{A}_s$ esiste $t \in \mathrm{T}_\Sigma$ t.c. $a = g_s([t]_{\equiv_\mathrm{A}}) = eval^\mathrm{A}(t)$, quindi $eval^\mathrm{A}$ è surgettiva. $\square$

Prop. 1.34

1. Se $\mathrm{A}, \mathrm{B} \in Alg_\Sigma$ ed esiste $h: \mathrm{A} \to \mathrm{B}$, allora $\equiv_\mathrm{A} \subseteq \equiv_\mathrm{B}$;
2. se $\mathrm{A}, \mathrm{B} \in Gen_\Sigma$, allora $h: \mathrm{A} \to \mathrm{B}$ sse $\equiv_\mathrm{A} \subseteq \equiv_\mathrm{B}$;
3. se R è una congruenza su T_Σ e $R \subseteq \equiv_\mathrm{A}$, allora esiste unico $\phi: \mathrm{T}_\Sigma/R \to \mathrm{A}$.

Dim.

1 Per ogni $s \in S$ $t\ (\equiv_\mathrm{A})_s\ t' \Rightarrow t^\mathrm{A} = t'^\mathrm{A} \Rightarrow h(t^\mathrm{A}) = h(t'^\mathrm{A}) \Rightarrow t^\mathrm{B} = t'^\mathrm{B} \Rightarrow t\ (\equiv_\mathrm{B})_s\ = t'$.
Per ogni $pr \in PR_{s_1 \ldots s_n}$, $n \geq 0$ $(t_1, \ldots, t_n) \in (\equiv_\mathrm{A})_{pr} \Rightarrow (t_1^\mathrm{A}, \ldots, t_n^\mathrm{A}) \in pr^\mathrm{A} \Rightarrow$
$(h_{s_1}(t_1^\mathrm{A}), \ldots, h_{s_n}(t_n^\mathrm{A})) \in pr^\mathrm{B} \Rightarrow (t_1^\mathrm{B}, \ldots, t_n^\mathrm{B}) \in pr^\mathrm{B} \Rightarrow (t_1, \ldots, t_n) \in (\equiv_\mathrm{B})_{pr}$.

2 Poichè $t^\mathrm{A} = t'^\mathrm{A} \Rightarrow t^\mathrm{B} = t'^\mathrm{B}$, allora si può definire un'applicazione h mediante $h(t^\mathrm{A}) = t^\mathrm{B}$, che risulta un omomorfismo per la Prop. 1.27 4.

3 Poiché $R \subseteq \equiv_\mathrm{A}$, si può definire un omomorfismo $\psi: \mathrm{T}_\Sigma/R \to \mathrm{T}_\Sigma/{\equiv_\mathrm{A}}$ dato da: $\psi_s([t]_R) = [t]_{\equiv_\mathrm{A}}$ (questa è una buona definizione perché $t\ R\ t' \Rightarrow t \equiv_\mathrm{A} t'$, e inoltre si può dimostrare facilmente che è un omomorfismo). Notiamo che ψ è unico, perché T_Σ/R è term-generated (Prop. 1.27). Quindi il seguente diagramma commuta

$$\begin{tikzcd}
 & \mathrm{T}_\Sigma \arrow[r, "eval^\mathrm{A}"] \arrow[dl, "\pi_R"'] \arrow[d, "\pi_{\equiv_\mathrm{A}}"] & \mathrm{A} \\
\mathrm{T}_\Sigma/R \arrow[r, "\psi"] & \mathrm{T}_\Sigma/{\equiv_\mathrm{A}} \arrow[ur, "g"'] &
\end{tikzcd}$$

cioè esiste un omomorfismo $\phi: \mathrm{T}_\Sigma/R \to \mathrm{A}$: possiamo scegliere infatti $\phi = g \circ \psi$; inoltre tale omomorfismo è unico perché T_Σ/R è term-generated (Prop. 1.27). $\square$

Cor. 1.35 Sia X una famiglia disgiunta di variabili indiciate su S, A una Σ-algebra e v una valutazione di X in A. Se R è una congruenza su $\mathrm{T}_\Sigma(X)$ t.c. $R \subseteq \equiv_\mathrm{A}$, allora esiste un unico omomorfismo $h: \mathrm{T}_\Sigma(X)/R \to \mathrm{A}$ t.c. il seguente diagramma commuti.

$$\begin{tikzcd}
\mathrm{T}_\Sigma(X) \arrow[r, "eval^{\mathrm{A},v}"] \arrow[d, "\pi"'] & \mathrm{A} \\
\mathrm{T}_\Sigma(X)/R \arrow[ur, "h"'] &
\end{tikzcd}$$

Dim. Definiamo una nuova segnatura $\Sigma^X = (S, OP^X, PR^X)$ nel seguente modo: per ogni $s \in S$ $OP^X_{\Lambda,s} = OP_{\Lambda,s} \cup X_s$, per ogni $w \in S^+$, $s \in S$ $OP^X_{w,s} = OP_{w,s}$ e per ogni $w \in S^*$ $PR^X_w = PR_w$. Data la Σ-algebra A si può definire una Σ^X-algebra A^X come segue:
– per ogni $s \in S$, $(\mathrm{A}^X)_s = \mathrm{A}_s$;

– per ogni $op \in OP$, $op^{\mathrm{A}^X} = op^{\mathrm{A}}$;
– per ogni $x \in X$, $x^{\mathrm{A}^X} = v(x)$;
– per ogni $pr \in PR$, $pr^{\mathrm{A}^X} = pr^{\mathrm{A}}$.

Si ha che $T_{\Sigma^X} \cong \mathrm{T}_\Sigma(X)$ ed $\mathrm{A} \cong A^X$ come Σ-algebre. È facile vedere che: se Q è una congruenza su $\mathrm{T}_\Sigma(X)$, allora Q è anche una congruenza su T_{Σ^X} e $\equiv_{\mathrm{A}} = \equiv_{\mathrm{A}^X}$.

Quindi possiamo applicare la Prop. 1.34, e si ha che esiste unico $h: T_{\Sigma^X}/R \to \mathrm{A}^X$ (Σ^X-omomorfismo); è facile vedere che $h: T_{\Sigma^X}/R \to \mathrm{A}^X$ (Σ-omomorfismo) ed è anche l'unico. □

1.5 Teoremi di inizialità e finalità

Prop. 1.36 Se I è iniziale in $\mathcal{C} \subseteq Alg_\Sigma$, allora I non ha sottoalgebre proprie in $\mathcal{C}$.
Dim. Sia I iniziale in $\mathcal{C}$, e supponiamo che esista I' sottoalgebra propria di I. Sia $f: \mathrm{I} \to \mathrm{I}'$ l'unico omomorfismo tra I e I' e sia $g: \mathrm{I}' \to \mathrm{I}$ l'immersione di I' in I, cioè l'omomorfismo $g(a) = a$ per ogni $a \in \mathrm{I}'$, che non è surgettivo.

Allora per inizialità, $Id^{\mathrm{I}} = g \circ f$, e dunque g è surgettiva e quindi si ottiene una contraddizione. □

Cor. 1.37 Se $\mathcal{C} \subseteq Alg_\Sigma$, $\mathcal{C}$ è chiusa rispetto alle sottoalgebre ed I è iniziale in $\mathcal{C}$, allora I è term-generated. □

Prop. 1.38 Se I è iniziale in $\mathcal{C}$, $\mathrm{I} \in Gen_\Sigma$ sse $eval^{\mathrm{I}}(\mathrm{T}_\Sigma) \in \mathcal{C}$.
Dim.

$\Rightarrow$ Se I è iniziale in $\mathcal{C}$ ed $\mathrm{I} \in Gen_\Sigma$, allora $\mathrm{I} = eval^{\mathrm{I}}(\mathrm{T}_\Sigma) \in \mathcal{C}$.

$\Leftarrow$ Se I è iniziale in $\mathcal{C}$ ed $eval^{\mathrm{I}}(\mathrm{T}_\Sigma) \in \mathcal{C}$, allora, poiché $eval^{\mathrm{I}}(\mathrm{T}_\Sigma) \subseteq \mathrm{I}$, ed I non ha sottoalgebre proprie in $\mathcal{C}$ deve essere $eval^{\mathrm{I}}(\mathrm{T}_\Sigma) = \mathrm{I}$ e quindi $\mathrm{I} \in Gen_\Sigma$. □

Prop. 1.39 Se I è iniziale in $\mathcal{C}$ ed $\mathrm{I} \in Gen_\Sigma$, allora $\mathrm{I} \cong \mathrm{T}_\Sigma/\equiv_{\mathcal{C}}$, dove $\equiv_{\mathcal{C}} = \cap\{\equiv_{\mathrm{A}}\}_{A \in \mathcal{C}}$; ossia $t \equiv_{\mathcal{C}} t' \Leftrightarrow$ per ogni $\mathrm{A} \in \mathcal{C}$ $t \equiv_{\mathrm{A}} t'$.

Viceversa se $\mathrm{I} \cong \mathrm{T}_\Sigma/\equiv_{\mathcal{C}}$ ed $\mathrm{I} \in \mathcal{C}$, allora I è iniziale in $\mathcal{C}$.
Dim. Se $\mathrm{I} \in Gen_\Sigma$, allora si ha $\mathrm{I} = eval^{\mathrm{I}}(\mathrm{T}_\Sigma) \cong \mathrm{T}_\Sigma/\equiv_{\mathrm{I}}$.

Per inizialità, si ha che per ogni $\mathrm{A} \in \mathcal{C}$ esiste unica $f: \mathrm{I} \to \mathrm{A}$ e quindi per la Prop. 1.34, si ha $\equiv_{\mathrm{I}} \subseteq \equiv_{\mathrm{A}}$, e quindi $\equiv_{\mathrm{I}} \subseteq \equiv_{\mathcal{C}}$.

D'altra parte $\equiv_{\mathcal{C}} \subseteq \equiv_{\mathrm{I}}$ poiché $\mathrm{I} \in \mathcal{C}$, quindi $\equiv_{\mathcal{C}} = \equiv_{\mathrm{I}}$ e di conseguenza
$\mathrm{I} \cong \mathrm{T}_\Sigma/\equiv_{\mathcal{C}} = \mathrm{T}_\Sigma/\equiv_{\mathrm{I}}$.

Viceversa, essendo $\equiv_{\mathcal{C}} \subseteq \equiv_{\mathrm{A}}$ per ogni $\mathrm{A} \in \mathcal{C}$, per la Prop. 1.34, esiste unica $f: \mathrm{T}_\Sigma/\equiv_{\mathcal{C}} \to \mathrm{A}$ e dunque, essendo $\mathrm{I} \cong \mathrm{T}_\Sigma/\equiv_{\mathcal{C}}$, esiste unica $f: \mathrm{I} \to \mathrm{A}$; ma allora, poichè $\mathrm{I} \in \mathcal{C}$, I è iniziale. □

Prop. 1.40 Se $\mathcal{C} \subseteq Gen_\Sigma$, allora $\mathrm{Z} \in \mathcal{C}$ è finale in $\mathcal{C}$ sse

1. per ogni $t, t' \in \mathrm{T}_\Sigma$, $t \equiv_{\mathrm{Z}} t'$ sse esiste $\mathrm{A} \in \mathcal{C}$ t.c. $t \equiv_{\mathrm{A}} t'$;
2. $(t_1, \ldots, t_n) \in (\equiv_{\mathrm{Z}})_{pr} \Leftrightarrow$ esiste $\mathrm{A} \in \mathcal{C}$ $(t_1, \ldots, t_n) \in (\equiv_{\mathrm{A}})_{pr}$.

Dim.

$\Rightarrow$ Sia Z finale in $\mathcal{C}$; allora le parti $\Rightarrow$ di 1 e di 2 sono ovvie poichè $\mathrm{Z} \in \mathcal{C}$. Viceversa se $t \equiv_{\mathrm{A}} t'$, allora, detto h l'omomorfismo di A in Z, è $h(t^{\mathrm{A}}) = t^{\mathrm{Z}} = h(t'^{\mathrm{A}}) = t'^{\mathrm{Z}}$ per la Prop. 1.27 1; se $(t_1, \ldots, t_n) \in (\equiv_{\mathrm{A}})_{pr}$ allora $(t_1^{\mathrm{A}}, \ldots, t_n^{\mathrm{A}}) \in pr^{\mathrm{A}} \Rightarrow (h_{s_1}(t_1^{\mathrm{A}}), \ldots, h_{s_n}(t_n^{\mathrm{A}})) \in pr^{\mathrm{Z}} \Rightarrow (t_1, \ldots, t_n) \in (\equiv_{\mathrm{Z}})_{pr}$

$\Leftarrow$ Viceversa se 1 e 2 sono soddisfatte, allora per ogni A si può definire un omomorfismo di A in Z ponendo $h(t^{\mathrm{A}}) = t^{\mathrm{Z}}$, per la Prop. 1.27 4; d'altra parte per la Prop. 1.27 2 questo omomorfismo è unico; per l'ipotesi $\mathrm{Z} \in \mathcal{C}$ si ha che è finale in $\mathcal{C}$. □

2 Specifiche del primo ordine

2.1 Linguaggio logico del primo ordine

Nel seguito Σ denota una segnatura (S, OP, PR) ed X una famiglia di insiemi di variabili indiciate su S; inoltre per semplicità di notazione assumiamo che Σ ed X siano disgiunte; ci si puó sempre ricondurre a questo caso considerando Σ^+ (vedi Sez. 1.2) e X^+ definito da $X_s^+ = X_s \times \{s\}$, per ogni $s \in S$.

Def. 2.1 L'insieme $A_\Sigma(X)$ degli *atomi* o *formule atomiche* e $FOF_\Sigma(X)$ delle *formule (logiche) del primo ordine (con eguaglianza)* su Σ ed X sono definiti induttivamente come segue:

- $pr(t_1, \ldots, t_n) \in A_\Sigma(X)$ per ogni $pr \in PR_{s_1 \ldots s_n}$, $t_1 \in \mathrm{T}_\Sigma(X)_{s_1}$, $\ldots$, $t_n \in \mathrm{T}_\Sigma(X)_{s_n}$, $s_1, \ldots, s_n \in S$;
- $t = t' \in A_\Sigma(X)$ per ogni $t, t' \in \mathrm{T}_\Sigma(X)_s$, $s \in S$; (le formule di questo tipo sono dette *equazioni* o anche *formule atomiche equazionali*);
- $A_\Sigma(X) \subseteq FOF_\Sigma(X)$;
- $\mathrm{F} \wedge \mathrm{F}', \mathrm{F} \vee \mathrm{F}', \mathrm{F} \supset \mathrm{F}', \neg \mathrm{F} \in FOF_\Sigma(X)$ per ogni $\mathrm{F}, \mathrm{F}' \in FOF_\Sigma(X)$;
- $\forall\, x\,.\,\mathrm{F}, \exists\, x\,.\,\mathrm{F} \in FOF_\Sigma(X)$ per ogni $\mathrm{F} \in FOF_\Sigma(X)$ ed $x \in X$.

Una formula si dice *ground* se non contiene variabili. Una formula $\mathrm{F} \in FOF_\Sigma(X)$ è detta *chiusa* se l'insieme delle variabili libere in F, *Free*(F), è vuoto, dove *Free*(F) è definito per induzione strutturale su F nel seguente modo.

- $Free(pr(t_1, \ldots, t_n)) = Var(t_1) \cup \ldots \cup Var(t_n)$
- $Free(t = t') = Var(t) \cup Var(t')$
- $Free(\mathrm{F} \wedge \mathrm{F}') = Free(\mathrm{F}) \cup Free(\mathrm{F}')$
- $Free(\mathrm{F} \vee \mathrm{F}') = Free(\mathrm{F}) \cup Free(\mathrm{F}')$
- $Free(\mathrm{F} \supset \mathrm{F}') = Free(\mathrm{F}) \cup Free(\mathrm{F}')$
- $Free(\neg \mathrm{F}) = Free(\mathrm{F})$
- $Free(\forall\, x\,.\,\mathrm{F}) = Free(\mathrm{F}) - \{x\}$
- $Free(\exists\, x\,.\,\mathrm{F}) = Free(\mathrm{F}) - \{x\}$

dove

- $Var(x) = \{x\}$
- $Var(op(t_1, \ldots, t_n)) = Var(t_1) \cup \ldots \cup Var(t_n)$. $\square$

Notazioni. nel caso in cui si voglia mettere in evidenza il tipo (o sort) di una variabile o quando si usi un insieme di variabili, si deve usare $\forall\, x{:}\, s\,.\,\mathrm{F}$ ed $\exists\, x{:}\, s\,.\,\mathrm{F}$; di conseguenza una formula $\forall\, x_1 \ldots \forall\, x_n.\mathrm{F}$, dove per $i = 1, \ldots, n$ x_i è una variabile di tipo s_i si scrive $\forall\, x_1{:}\, s_1 \ldots \forall\, x_n{:}\, s_n.\mathrm{F}$; inoltre nelle quantificazioni multiple si scrive $\forall\, \{x_1, \ldots, x_n\}\mathrm{F}$ per $\forall\, x_1\,.\,\forall\, x_2 \ldots \forall\, x_n\,.\,\mathrm{F}$. Si noti peró che nel caso di X disgiunto (come noi assumiamo) non si puó avere contemporaneamente $x{:}\, s$ ed $x{:}\, s'$ con $s \neq s'$.

Se F è una formula aperta, si indica con F^C la *chiusura universale* di F, cioè la formula $\forall\ \mathit{Free}(\mathrm{F})\ \mathrm{F}$.

Diamo ora la nozione di soddisfacimento (o verità) di una formula su una Σ-algebra rispetto ad una valutazione e quella, derivatta, di validità.

Def. 2.2 Sia A una Σ-algebra, $\mathrm{F} \in \mathit{FOF}_\Sigma(X)$ e v una valutazione di X in A (vedi Def. 1.14).

Si dice che A *soddisfa* F *rispetto a* v o che F *è vera in* A *rispetto a* v, e si scrive $\mathrm{A}, v \models \mathrm{F}$ secondo la seguente definizione induttiva.

$\mathrm{A}, v \models pr(t_1, \ldots, t_n)$	se	$(t_1^{\mathrm{A},v}, \ldots, t_n^{\mathrm{A},v}) \in pr^{\mathrm{A}}$
$\mathrm{A}, v \models t = t'$	se	$t^{\mathrm{A},v} = t'^{\mathrm{A},v}$
$\mathrm{A}, v \models \mathrm{F} \wedge \mathrm{F}'$	se	$\mathrm{A}, v \models \mathrm{F}$ e $\mathrm{A}, v \models \mathrm{F}'$
$\mathrm{A}, v \models \mathrm{F} \vee \mathrm{F}'$	se	$\mathrm{A}, v \models \mathrm{F}$ oppure $\mathrm{A}, v \models \mathrm{F}'$
$\mathrm{A}, v \models \neg \mathrm{F}$	se	$\mathrm{A}, v \not\models \mathrm{F}$
$\mathrm{A}, v \models \mathrm{F} \supset \mathrm{F}'$	se	$\mathrm{A}, v \models \mathrm{F}'$ oppure $\mathrm{A}, v \not\models \mathrm{F}$
$\mathrm{A}, v \models \forall\, x\,.\, \mathrm{F}$	se	per ogni $a \in \mathrm{A}_s$ $\mathrm{A}, v[a/x] \models \mathrm{F}$, con s sort di x
$\mathrm{A}, v \models \exists\, x\,.\, \mathrm{F}$	se	esiste $a \in \mathrm{A}_s$ t.c. $\mathrm{A}, v[a/x] \models \mathrm{F}$, con s sort di x

(Si ricorda che $v[a/x]$ indica la valutazione tale che $v[a/x](x) = a$ e $v[a/x](y) = v(y)$ per ogni $y \neq x$; inoltre, come da Teor. 1.15, $t^{\mathrm{A},v} =_{def} eval^{\mathrm{A},v}(t)$, dove $eval^{\mathrm{A},v}$ è l'unico omomorfismo da $\mathrm{T}_\Sigma(X)$ in A che estende la valutazione v).

Si dice che A *soddisfa* F o che F *è valida in* A e si scrive $\mathrm{A} \models \mathrm{F}$ sse per ogni valutazione v di $\mathit{Free}(\mathrm{F})$ in A si ha che $\mathrm{A}, v \models \mathrm{F}$; talvolta si usa la dizione "F è vera in A", però è sempre meglio riservare l'aggettivo "vero" al soddisfacimento locale in un certo punto, e quello "valido" alla descrizione di un soddisfacimento globale.

Sia $\mathcal{C}$ una classe di Σ-algebre, si dice che $\mathcal{C}$ *soddisfa* F o che F *è valida in* $\mathcal{C}$ e si scrive $\mathcal{C} \models \mathrm{F}$ sse per ogni $\mathrm{A} \in \mathcal{C}$ si ha che $\mathrm{A} \models \mathrm{F}$. $\square$

Si noti bene che nella definizione di validità la valutazione è relativa alle sole variabili libere della formula. La nozione di validità si potrebbe cambiare, limitatamente al caso eterogeneo, se si considerasse una famiglia di variabili contenente propriamente le variabili libere.

Infatti, si osservi che, se $x \in X_s$ e $\mathrm{A}_s = \emptyset$, allora non esiste alcuna valutazione $v{:}\, X \to \mathrm{A}$ e dunque banalmente $\mathrm{A}, v \models \mathrm{F}$ per ogni $\mathrm{F} \in \mathit{FOF}_\Sigma(X)$, e ogni $v{:}\, X \to \mathrm{A}$ (si ricordi che "per ogni v vale $\mathrm{A}, v \models \mathrm{F}$" equivale a "non esiste v t.c. non vale $\mathrm{A}, v \models \mathrm{F}$").

Dunque, nel caso in cui ci sono supporti vuoti e non vuoti, la validità puó dipendere dal fatto che X includa o no variabili di sort il cui carrier è vuoto in A. La proposizione seguente dà una condizione per cui non fa differenza la famiglia di variabili considerate; nel caso omogeneo (una sola sort) la condizione è sempre soddisfatta.

Prop. 2.3 Se i supporti di A sono tutti vuoti o tutti non vuoti (per esempio, se $(\mathrm{T}_\Sigma)_s \neq \emptyset$ per ogni $s \in S$), allora se $Y \supseteq \mathit{Free}(\mathrm{F})$,

$$\mathrm{A} \models \mathrm{F} \qquad \text{sse} \qquad \mathrm{A}, v \models \mathrm{F} \quad \text{per ogni } v{:}\ Y \to \mathrm{A}.$$

Dim. Si noti che per ogni $\mathrm{F} \in \mathit{FOF}_\Sigma(X)$, ogni $v{:}\, X \to \mathrm{A}$ ed ogni $Y \supseteq \mathit{Free}(\mathrm{F})$

$$\mathrm{A}, v \models \mathrm{F} \qquad \text{sse} \qquad \mathrm{A}, v_{|Y} \models \mathrm{F}$$

dove $v_{|Y}$ indica la restrizione di v agli elementi di Y, e si proceda per induzione sulla struttura delle formule (per esercizio). $\square$

Diamo ora due proprietà importanti di $\models$.

Prop. 2.4 Sia $\mathrm{F} \in \mathit{FOF}_\Sigma(X)$ e siano A, B due Σ-algebre.

1. Se A e B sono isomorfe, allora $\mathrm{A} \models \mathrm{F}$ sse $\mathrm{B} \models \mathrm{F}$;
2. $\mathrm{A} \models \mathrm{F}$ sse $\mathrm{A} \models \mathrm{F}^C$.

Dim. Si verificano facilmente dalle definizioni (per esercizio). □

2.2 Specifiche semplici

Def. 2.5 Una *specifica* o *presentazione semplice del primo ordine*, o semplicemente *specifica* se non c'è ambiguità, è una coppia $\mathrm{SP} = (\Sigma, AX)$, dove Σ è una segnatura ed AX un insieme di formule logiche del primo ordine su Σ.

$\mathit{Mod}(\mathrm{SP})$ indica la classe dei *modelli* di SP, cioè

$$\mathit{Mod}(\mathrm{SP}) = \{A \mid \mathrm{A} \in \mathit{Alg}_\Sigma \text{ e per ogni } \mathrm{F} \in AX\ \mathrm{A} \models \mathrm{F}\}.$$

$\mathit{GMod}(\mathrm{SP})$ indica la classe dei modelli *term-generated* (o *induttivi* o *reachable*) di SP, cioè

$$\mathit{GMod}(\mathrm{SP}) = \{\mathrm{A} \mid \mathrm{A} \in \mathit{Gen}_\Sigma \text{ e per ogni } \mathrm{F} \in AX\ \mathrm{A} \models \mathrm{F}\}. \Box$$

C'è una importante proposizione che lega i modelli delle specifiche alle proprietà da essi soddisfatte, cioè alle formule valide in essi.

Def. 2.6 Siano $\mathcal{K} \subseteq \mathit{Alg}_\Sigma$ ed $AX \subseteq \mathit{FOF}_\Sigma(X)$.

- $Th(\mathcal{K}, X) =_{def} \{\mathrm{F} \mid \mathrm{F} \in \mathit{FOF}_\Sigma(X), \mathcal{K} \models \mathrm{F}\}$, si dice *teoria* in X di $\mathcal{K}$;
- $Th(\mathcal{K}) =_{def} \bigcup\{Th(\mathcal{K}, X) \mid X$ famiglia S-indiciata di variabili $\}$, si dice *teoria* di $\mathcal{K}$;
- un'*assiomatizzazione* di $\mathcal{K}$ è un insieme di assiomi AX tali che $\mathcal{K} = \mathit{Mod}(\Sigma, AX)$. □

Si noti che le assiomatizzazioni non sono uniche, come si vede subito notando che sia AX che $Th(\mathit{Mod}(\Sigma, AX))$ sono due assiomatizzazioni di $\mathit{Mod}(\Sigma, AX)$, che in generale non coincidono.

Nel seguito quando la segnatura Σ si evince dal contesto, scriviamo semplicemente $\mathit{Mod}(AX)$ e $\mathit{GMod}(AX)$ per $\mathit{Mod}(\Sigma, AX)$ e $\mathit{GMod}(\Sigma, AX)$.

Prop. 2.7 Siano $\mathcal{F} \subseteq \mathit{FOF}_\Sigma(X)$ e $\mathcal{C} \subseteq \mathit{Alg}_\Sigma$.

1. Le funzioni $Th_\mathcal{F}\colon \mathcal{P}(\mathcal{C}) \to \mathcal{P}(\mathcal{F})$, $\mathit{Mod}_\mathcal{C}\colon \mathcal{P}(\mathcal{F}) \to \mathcal{P}(\mathcal{C})$ definite da $Th_\mathcal{F}(\mathcal{K}) = Th(\mathcal{K}) \cap \mathcal{F}$ e $\mathit{Mod}_\mathcal{C}(AX) = \mathit{Mod}(AX) \cap \mathcal{C}$ costituiscono una *connessione di Galois* fra $\mathcal{F}$ e $\mathcal{C}$, cioè per ogni AX, $AX' \subseteq \mathcal{F}$ e $\mathcal{K}$, $\mathcal{K}' \subseteq \mathcal{C}$

   G1 $AX \subseteq Th_\mathcal{F}(\mathit{Mod}_\mathcal{C}(AX))$;

   G2 $\mathcal{K} \subseteq \mathit{Mod}_\mathcal{C}(Th_\mathcal{F}(\mathcal{K}))$;

   G3 $AX \subseteq AX' \Rightarrow \mathit{Mod}_\mathcal{C}(AX) \supseteq \mathit{Mod}_\mathcal{C}(AX')$;

   G4 $\mathcal{K} \subseteq \mathcal{K}' \Rightarrow Th_\mathcal{F}(\mathcal{K}) \supseteq Th_\mathcal{F}(\mathcal{K}')$.

2. G5 $\mathit{Mod}_\mathcal{C}(AX) = \mathit{Mod}_\mathcal{C}(Th_\mathcal{F}(\mathit{Mod}_\mathcal{C}(AX)))$

   G6 $Th_\mathcal{F}(\mathcal{K}) = Th_\mathcal{F}(\mathit{Mod}_\mathcal{C}(Th_\mathcal{F}(\mathcal{K})))$.

Dim.

1 Semplice verifica dalle definizioni:

G1: per definizione di $Mod_{\mathcal{C}}(AX)$, $AX \subseteq Th(Mod_{\mathcal{C}}(AX))$, ed essendo $AX \subseteq \mathcal{F}$, è anche $AX \subseteq Th(Mod_{\mathcal{C}}(AX)) \cap \mathcal{F} = Th_{\mathcal{F}}(Mod_{\mathcal{C}}(AX))$;

G2: analogamente a G1 scambiando teoremi con modelli (si dice per dualità);

G3: se $AX \subseteq AX'$, ogni modello in $Mod_{\mathcal{C}}(AX')$ soddisfa per definizione ogni assioma di AX' e quindi ogni assioma di AX, cioè sta in $Mod_{\mathcal{C}}(AX)$;

G4: analogamente a G3 per dualità.

2 Conseguenze valide per ogni connessione di Galois, cioè da G1, G2, G3, G4. Proviamo G5: da G1 e G3 si ha $Mod_{\mathcal{C}}(AX) \supseteq Mod_{\mathcal{C}}(Th_{\mathcal{F}}(Mod_{\mathcal{C}}(AX))$; da G2 per $\mathcal{K} = Mod_{\mathcal{C}}(AX)$ si ha la relazione opposta e quindi G5; G6 si ottiene analogamente per dualità. □

Le relazioni G5 e G6 si possono scrivere in un modo piú sintetico e suggestivo, indicando sia $Mod_{\mathcal{C}}$ che $Th_{\mathcal{F}}$ con *, il che non crea ambiguità, dal momento che l'argomento definisce la funzionalità di *. Allora abbiamo

G5 $AX^* = AX^{***}$,

G6 $\mathcal{K}^* = \mathcal{K}^{***}$.

Si noti che una connessione di Galois stabilisce sempre una corrispondenza biunivoca; in questo caso la corrispondenza è fra teorie ($\mathcal{K}^*$) e classi di modelli (AX^*). In altre parole $Th_{\mathcal{F}}$ e $Mod_{\mathcal{C}}$, ristretti rispettivamente a teorie (in $\mathcal{F}$) ed a classi di modelli (in $\mathcal{C}$) sono l'una l'inversa dell'altra.

Specializziamo ora, per maggior chiarezza, la connesione di Galois al caso di tutti i modelli (cioè $\mathcal{C} = Alg_{\Sigma}$) e dei modelli term-generated (cioè $\mathcal{C} = Gen_{\Sigma}$).

Prop. 2.8 Siano SP e SP′ due specifiche su Σ. Allora

1. $Mod(\mathrm{SP}) \subseteq Mod(\mathrm{SP}')$ sse $Th(Mod(\mathrm{SP}')) \subseteq Th(Mod(\mathrm{SP}))$;
2. $GMod(\mathrm{SP}) \subseteq GMod(\mathrm{SP}')$ sse $Th(GMod(\mathrm{SP}')) \subseteq Th(GMod(\mathrm{SP}))$ □.

La relazione $Mod(\mathrm{SP}) \subseteq Mod(\mathrm{SP}')$ (o anche se si considerano iadt invece di adt $GMod(\mathrm{SP}) \subseteq GMod(\mathrm{SP}')$) è assai importante e viene descritta dicendo che SP è un *raffinamento* di SP′ (talvolta si dice anche *implementazione* di SP′) e scrivendo SP′ $\sim\sim\!\!>$ SP. Vedremo in seguito che molte nozioni di implementazione anche su segnature diverse si riconducono a questa relazione.

2.3 Semantica di specifiche semplici

2.3.1 Generalità

Negli ultimi anni si è raggiunto un grande consenso sulla seguente definizione

Def. 2.9 Un *tipo di dato concreto* è un'algebra su una segnatura. □

Quando a partire da un tipo di dato concreto, si astrae dalla particolare rappresentazione degli oggetti, si ottiene un tipo di dato astratto; l'operazione di astrazione è modellata formalmente dalla chiusura per isomorfismo.

Def. 2.10 Un *tipo di dato astratto* (in breve *adt*) è una classe di isomorfismo di algebre su una segnatura (cioè la classe di isomorfismo di un tipo di dato concreto).

Un *tipo di dato astratto term-generated* (o *induttivo* o *reachable*, in breve *iadt*) è una classe di isomorfismo di algebre term-generated (o induttive o reachable) cioè in Gen_Σ. □

Si noti che per alcuni autori un adt è per definizione un iadt. Naturalmente ogni algebra in Alg_Σ puó essere vista come algebra in $Gen_{\Sigma'}$, per una segnatura Σ' ottenuta da Σ aggiungendo opportune (in generale infinite) costanti e quindi ogni adt su Σ diventa un iadt su Σ'.

È importante convenire su dei modi per definire un adt o classi di adt. Un approccio "model-oriented" è quello di produrre un tipo di dato concreto (o classi di) e poi astrarre per isomorfismo (approccio aristotelico; cfr. la nozione di "delegation" nello stile object-oriented). Un approccio piú "property-oriented" consiste nel definire le proprietà che le algebre dell'adt (o della classe) devono soddisfare (approccio platonico; cfr. la nozione di "classe" e "inheritance" nello stile object-oriented); questo è l'approccio tipico delle specifiche logiche.

La semantica di una specifica è sempre una classe di algebre chiusa per isomorfismo, cioè una classe di adt (o di iadt).

Si possono dare diverse semantiche; le più importanti sono le seguenti:

- *approccio loose*: $Mod(\mathrm{SP})$ ($GMod(\mathrm{SP})$);
- *approccio iniziale*: classe delle algebre iniziali in $Mod(\mathrm{SP})$ (in $GMod(\mathrm{SP})$);
- *approccio finale*: classe delle algebre finali in $Mod(\mathrm{SP})$ (in $GMod(\mathrm{SP})$);
- *approccio osservazionale*: si identificano classi di algebre (chiuse per isomorfismo) in base a dei requisiti di comportamento rispetto a certe osservazioni (da cui anche il nome di approccio "behavioural"); si noti che non sempre le algebre in questa classe stanno in $Mod(\mathrm{SP})$.

Una specifica si dice *monomorfa* se il modello iniziale e finale coincidono.

Esempio 2.11 I numeri naturali.
Sia ΣN la segnatura con unica sort $s = nat$ e operazioni $0\colon \to nat$ e $succ\colon nat \to nat$.

Tipo concreto: le varie algebre su ΣN dove i numeri naturali sono rappresentati come stringhe di simboli usando la notazione decimale, binaria, ottale, esadecimale eccetera, forniscono diverse versioni del tipo di dato concreto numeri naturali.

Specifica: $\mathrm{NAT} = (\Sigma N, \emptyset)$.

La classe di adt associata alla specifica NAT varia a secondo degli approcci:

- *approccio loose:* $Mod(\mathrm{NAT})$, cioè la classe di tutti i modelli della specifica. In questo caso $Mod(\mathrm{NAT}) = Alg_{\Sigma N}$, che però non corrisponde alla nostra idea di "numeri naturali" (anche l'algebra banale su ΣN appartiene ad $Alg_{\Sigma N}$).
- *approccio iniziale:* la classe delle algebre iniziali in $Mod(\mathrm{NAT})$, che in questo caso corrisponde alla classe delle algebre isomorfe a $T_{\Sigma N}$ (l'algebra dei termini in notazione funzionale). Quindi l'adt definito seguendo l'approccio iniziale è quello dove si è astratto dalla particolare rappresentazione scelta per i numeri naturali (si veda che le algebre corrispondenti ai numeri naturali rappresentati in binario, decimale, ottale, esadecimale sono tutte iniziali in $Mod(\mathrm{NAT})$).
- *approccio finale:* l'algebra finale in $Mod(NAT)$, che è anche finale in $GMod(NAT)$ e consiste nell'algebra banale su ΣN (in cui il supporto è costituito da un unico elemento). Ovviamente quest'algebra non corrisponde all'idea di "naturali".

Fine esempio

Esempio 2.12 I numeri interi
Specifica: INT $= (\Sigma I, IAX)$, dove:
$\Sigma I = (\{int\}, OPI, PRI)$ con $OPI_{\Lambda,int} = \{0\}$, $OPI_{int,int} = \{Succ, Pred\}$,
$OPI_{w,int} = \emptyset$ per ogni altro $w \in \{int\}^*$ e $PRI_w = \emptyset$ per ogni $w \in \{int\}^*$; e

$$IAX = \{Pred(Succ(n)) = n, Succ(Pred(n)) = n\}.$$

- *approccio loose:* anche in questo caso prendere come adt associati ad INT la classe delle classi di isomorfismo degli elementi di Mod(INT) non corrisponde alla nostra idea di numeri interi (anche l'algebra banale su ΣI è un modello di INT).

- *approccio iniziale:* in questo caso è difficile caratterizzare direttamente quali sono le algebre iniziali in Mod(INT); nelle sezioni seguenti ne sarà data una caratterizzazione esplicita, che risulterà anche qui corrispondente alla nostra idea usuale degli interi.

– *approccio finale:* algebra banale su ΣI.

Fine esempio

Esempio 2.13 Monoidi commutativi
Specifica: MC $= (\Sigma M, MAx)$, dove
ΣM è la segnatura $(\{elem\}, OPM, PRM)$ tale che $OPM_{\Lambda,elem} = \{0\}$,
$OPM_{elem\ elem,elem} = \{Add\}$, $OPM_{w,elem} = \emptyset$ per ogni altro $w \in \{elem\}^*$ e
$PRM_w = \emptyset$ per ogni $w \in \{elem\}^*$; e

$$MAx = \{Add(x,y) = Add(y,x),\ Add(Add(x,y),z) = Add(x, Add(y,z),\ Add(x,0) = x\}.$$

- *approccio loose:* in questo caso Mod(MC) rappresenta in maniera appropriata la classe degli adt "monoidi commutativi"; infatti gli elementi di Mod(MC) corrispondono a tutti i possibili monoidi commutativi.

- *approccio iniziale e finale:* le algebre iniziali in Mod(MC) sono isomorfe all'algebra banale su ΣM (corrispondono ai monoidi con un unico elemento: il valore di 0), che è finale; e quindi in questo caso l'approccio iniziale e finale coincidono, cioè la specifica è monomorfa.

Fine esempio

2.3.2 Semantica iniziale e finale

L'approccio iniziale e finale corrispondono all'introduzione di classi di adt soddisfacenti due insiemi di requisiti in un certo senso duali.

Si noti che dal punto di vista dell'informatica prima vengono i requisiti e poi la caratterizzazione delle classi da un punto di vista puramente formale, come classi iniziali e finali.

Def. 2.14 Data una specifica SP su Σ, I(SP) indica la classe di tutte le Σ-algebre che soddisfano le seguenti proprietà:

Mod) $I(\mathrm{SP}) \subseteq Mod(\mathrm{SP})$;

Gen) $I(\mathrm{SP}) \subseteq Gen_\Sigma$;

MinTruth) per ogni A $\in I(\mathrm{SP})$ ed $\alpha \in A_\Sigma$ $\quad$ A $\models \alpha \Leftrightarrow Mod(\mathrm{SP}) \models \alpha$. □

Si noti intanto che la classe $I(\mathrm{SP})$ è chiusa per isomorfismo ed è perció una classe di adt. Inoltre si osservi che *Mod* e *Gen* sono requisiti abbastanza naturali (soddisfacimento degli assiomi e rappresentabilità di ogni elemento di un'algebra), mentre *MinTruth* è proprio quella caratterizzante l'approccio iniziale: un atomo ground (cioè senza variabili) è vero in un'algebra della classe $I(\mathrm{SP})$ sse è vero in tutti i modelli sse la sua verità è conseguenza logica degli assiomi di SP.

L'approccio finale è proprio caratterizzato dalla proprietà duale di *MinTruth*: una atomo ground è vero in un'algebra della classe finale sse esiste almeno un modello in cui è vero sse la sua falsità non è conseguenza logica degli assiomi di SP.

Def. 2.15 Data una specifica SP su Σ, $F(\mathrm{SP})$ indica la classe di tutte le Σ-algebre che soddisfano le seguenti proprietà:

Mod) $F(\mathrm{SP}) \subseteq Mod(\mathrm{SP})$

Gen) $F(\mathrm{SP}) \subseteq Gen_\Sigma$

MaxTruth) per ogni A $\in F(\mathrm{SP})$ e $\alpha \in A_\Sigma$ $\quad$ A $\models \alpha \Leftrightarrow$ esiste B $\in GMod(\mathrm{SP})$ t.c. B $\models \alpha$. □

Anche qui si noti che $F(\mathrm{SP})$ è chiusa per isomorfismo.

Vediamo ora delle proprietà generali delle classi $I(\mathrm{SP})$ e $F(\mathrm{SP})$ che giustificano tra l'altro il nome di approccio iniziale e finale rispettivamente.

Teorema 2.16 Sia SP una specifica su Σ.

1. I $\in I(\mathrm{SP}) \Rightarrow$ I è iniziale in $Mod(\mathrm{SP})$.
2. Se I è iniziale in $Mod(\mathrm{SP})$ ed I $\in Gen_\Sigma \Rightarrow$ I $\in I(\mathrm{SP})$.
3. I $\in I(\mathrm{SP}) \Leftrightarrow (I \cong \mathrm{T}_\Sigma/\!\equiv_{Mod(\mathrm{SP})}$ ed I $\in Mod(\mathrm{SP}))$.

Dim.

1 Sia I $\in I(\mathrm{SP})$; da *Gen* si ha che I $\cong \mathrm{T}_\Sigma/\!\equiv$ per una qualche congruenza $\equiv$; da *MinTruth* si ha che $\equiv\; = \;\equiv_{Mod(\mathrm{SP})}$. Dalla Prop. 1.39 e dal requisito *Mod* segue che I è iniziale in $Mod(\mathrm{SP})$.

2 Dalla Prop. 1.39 si ha che I $\cong\ \mathrm{T}_\Sigma/\!\equiv_{Mod(\mathrm{SP})}$, quindi i requisiti *Gen*, *Mod* e *MinTruth* sono verificati e cioè I $\in I(\mathrm{SP})$.

3 Segue dalla Prop. 1.39.□

Il fatto che $\mathrm{T}_\Sigma/\!\equiv_{Mod(\mathrm{SP})} \in Mod(\mathrm{SP})$ dipende dalla forma degli assiomi di SP e non sempre avviene. Si noti invece che, poiché gli oggetti iniziali sono sempre isomorfi si ha che $I(\mathrm{SP})$ individua sempre al piú una classe di isomorfismo, cioè un adt.

Esempio 2.17 Sia Σ_1 la segnatura avente un'unica sort s ed un'unica operazione zeraria $a\colon \to s$ e SP_1 la specifica $(\Sigma_1, \{\exists\, x \,.\, x \neq a\})$. Si ha subito che

$$Mod(\mathrm{SP}_1) = \{\mathrm{A} \mid \mathrm{A} \text{ è una } \Sigma_1\text{-algebra ed } \mathrm{A} \notin Gen_{\Sigma_1}\},$$

cioè SP_1 ha solo modelli non term-generated; quindi $I(\mathrm{SP}_1) = \emptyset$, poiché il requisito *Gen* è incompatibile con *Mod*.

Sia Σ_2 la segnatura avente un'unica sort s e tre operazioni $a, b, c\colon \to s$ e SP_2 la specifica $(\Sigma_2, \{a = b \;\vee\; a = c\})$.

È facile vedere che esistono $\mathrm{M}_1, \mathrm{M}_2 \in Mod(\mathrm{SP}_2)$ tali che $\mathrm{M}_1 \models a \neq b$ e $\mathrm{M}_2 \models a \neq c$; pertanto $T_{\Sigma_2}/\!\equiv_{Mod(\mathrm{SP}_2)} \not\models a = b \;\vee\; a = c$; quindi $I(\mathrm{SP}_2) = \emptyset$, poiché il requisito *MinTruth* è incompatibile con *Mod*. **Fine esempio**

Questi ultimi esempi mostrano che non sempre l'approccio iniziale individua un adt: in alcuni casi si ha che $I(\mathrm{SP}) = \emptyset$; nelle sezioni seguenti illustreremo alcuni casi importanti in cui l'approccio iniziale definisce esattamente un adt.

Si ha un teorema analogo per la classe $F(\mathrm{SP})$.

Teorema 2.18

1. $\mathrm{Z} \in F(\mathrm{SP}) \Leftrightarrow$ è finale in $GMod(\mathrm{SP})$;

2. $\mathrm{Z} \in F(\mathrm{SP}) \Leftrightarrow (\mathrm{Z} \cong \mathrm{T}_\Sigma / \sim_{GMod(\mathrm{SP})}$ e $\mathrm{Z} \in GMod(\mathrm{SP}))$
   dove $t_1 \sim_{GMod(\mathrm{SP})} t_2$ sse esiste $\mathrm{A} \in GMod(\mathrm{SP})$ t.c. $t_1^{\mathrm{A}} = t_2^{\mathrm{A}}$ e
   $(t_1, \ldots, t_n) \in (\sim_{GMod(\mathrm{SP})})_{pr}$ sse esiste $\mathrm{A} \in GMod(\mathrm{SP})$ t.c. $\mathrm{A} \models pr(t_1, \ldots, t_n)$.

Dim. Facilmente dalla Prop. 1.40. □

2.4 Deduzione logica

I sistemi deduttivi o logico formali sono particolari sistemi induttivi definenti insiemi di formule valide; il loro scopo è quello di fornire delle tecniche di prova della validità delle formule. Nella sottosezione 2.4.1 richiamiamo la nozione di sistema deduttivo.

Ci sono vari stili nella presentazione dei sistemi logico formali. Presentiamo due esempi notevoli, uno secondo lo stile detto hilbertiano (da Hilbert) e l'altro secondo lo stile detto "a sequenti" (da Gentzen). Si noti che tali sistemi sono per le formule del primo ordine omogene. Il caso eterogeneo viene discusso nella sezione 2.4.2.

2.4.1 Sistemi deduttivi (o logico-formali)

I sistemi deduttivi o logico-formali sono semplicemente particolari sistemi induttivi in cui gli oggetti degli insiemi definiti sono formule logiche. Richiamiamo qui le definizioni essenziali sui sistemi induttivi adattate al caso di un sistema (logico) deduttivo. Per un'esposizione più completa si vedano le lezioni di E.Astesiano per il corso di TAMC (a.a. 1990-91).

Def. 2.19 Sia *WFF* un insieme, detto delle *formule ben formate*. Un *sistema (logico) deduttivo* o *sistema logico formale* o *teoria logica* è costituito da un insieme *Reg* di *regole di inferenza*, dove una regola di inferenza è un sottinsieme di $\mathcal{P}(WFF) \times WFF$.

Un sistema deduttivo si dice *finitario* se per ogni $R \in Reg$ e per ogni $(Pr, c) \in R$, Pr è finito.

Se R è una regola di inferenza di un sistema deduttivo ed $r \in R$, r si dice *istanza* (o *regola*) di R; se $r = (Pr, c)$, allora Pr e c sono dette rispettivamente *premesse* e *conseguenza* dell'istanza r; se $Pr = \emptyset$ allora l'istanza r si dice *assioma* ed usualmente si scrive c per $(\emptyset, c)$. □

Si noti che *WFF* puó essere un insieme di formule qualunque; ben formate vuol dire che è un particolare sottoinsieme di stringhe su un alfabeto, appunto le stringhe ben formate secondo certe regole di struttura.

Def. 2.20 Sia $D = (WFF, Reg)$ un sistema deduttivo.

- L'insieme dei *teoremi* di D, o delle *formule deducibili* in D, è l'intersezione di tutti i sottoinsiemi di *WFF* chiusi rispetto a *Reg*, dove $\Gamma \subseteq WFF$ è *chiuso* rispetto a *Reg* se

  $$\text{per ogni } R \in Reg, \text{ per ogni } (Pr, c) \in R \ (Pr \subseteq \Gamma \Rightarrow c \in \Gamma).$$

  Usualmente si scrive $\vdash^D \mathrm{F}$ per indicare che F è un teorema del sistema deduttivo D; $\vdash^D$ si dice *relazione di deduzione associata a D*.

- Se $\Gamma \subseteq \mathit{WFF}$, si scrive $\Gamma \vdash^D \mathrm{F}$ per $\vdash^{D'} \mathrm{F}$ dove D' è il sistema deduttivo $(\mathit{WFF}, \mathit{Reg} \cup \{(\emptyset, \mathrm{F}) \mid \mathrm{F} \in \Gamma\})$.

- Un *albero di prova* (o *prova ad albero*) per F in D è un albero (non ordinato) etichettato su *WFF* ed appartenente all'insieme definito induttivamente da:

  – se $(\emptyset, \mathrm{F})$ è un'istanza di una regola di inferenza di D, allora l'albero costituito dalla sola radice etichettata F è un albero di prova per F in D;

  – se t_i, per $i \in I$, è un albero di prova per F_i in D e $(\{\mathrm{F}_i \mid i \in I\}, \mathrm{F})$ è un'istanza di una regola di inferenza di D, allora l'albero ottenuto dai t_i aggiungendo una radice etichettata F come padre dei nodi etichettati F_i, è un albero di prova per F in D.

- Se D è finitario, una *prova lineare* di F_n in D è una sequenza finita di elementi di *WFF* $\mathrm{F}_1, \ldots, \mathrm{F}_n$ tale che per $i = 1, \ldots, n$ F_i è un assioma di D oppure esistono $R \in \mathit{Reg}, (Pr, c) \in R$ tali che $Pr \subseteq \{\mathrm{F}_1, \ldots, \mathrm{F}_{i-1}\}$ e $c = \mathrm{F}_i$.$\square$

Nel seguito spesso useremo $\vdash^D$ o $\Gamma \vdash^D$ ad indicare il sistema deduttivo a cui queste relazioni di deduzione si riferiscono.

Le seguenti proprietà sono casi particolari di proprietà valide per ogni sistema induttivo.

Prop. 2.21 Sia $D = (\mathit{WFF}, \mathit{Reg})$ un sistema deduttivo.

– $\vdash^D \mathrm{F}$ sse esiste una prova ad albero per F in D;

– $\vdash^D \mathrm{F}$ sse F è deducibile nel sistema ottenuto da D restringendo le regole (o istanze) a quelle in cui le premesse sono costituite da teoremi;

– se D è finitario, allora $\vdash^D \mathrm{F}$ sse esiste una prova lineare $\mathrm{F}_1, \ldots, \mathrm{F}_n$ tale che $\mathrm{F}_n = \mathrm{F}$.$\square$

In genere le regole di inferenza sono insiemi infiniti e quindi occorre esprimerle in un modo finitario; ció avviene dando un insieme finito di "metaregole", cioè di schemi di regole. Una metaregola in genere è scritta nella forma: $\dfrac{Pre}{ce}\ cond$, dove:

cond è una metaespressione tale che per ogni istanziazione delle metavariabili ritorna, se definita, un valore booleano;

Pre è una metaespressione tale che per ogni istanziazione delle metavariabili soddisfacenti *cond* ritorna un insieme di formule ben formate;

ce è una metaespressione tale che per ogni istanziazione delle metavariabili soddisfacenti *cond* ritorna una formula ben formata.

In genere *Pre* consiste in un insieme di schemi di formule, cioè in un insieme di metaespressioni che per ogni istanziazione delle metavariabili danno una formula ben formata; analogamente *ce* è usualmente uno schema di formula.

Le istanze di una metaregola $\dfrac{Pre}{ce}\ cond$ sono tutte e sole quelle ottenute istanziando *Pre* e *ce* per una (stessa) istanziazione delle variabili soddisfacente *cond*.

Per tradizione si identifica una metaregola con la regola di inferenza associata e quindi una metaregola è anche detta regola di inferenza.

Esempio 2.22 Metaregole ed istanze

$\dfrac{\mathrm{F} \quad \mathrm{F} \supset \mathrm{G}}{\mathrm{G}}$ è la metaregola o regola di inferenza detta del "modus ponens" nella logica del primo ordine; in questo caso "F" e "G" sono le metavariabili (di tipo formule ben formate), "F F $\supset$ G" e "G" le metaespressioni, mentre "*cond*" è omessa essendo sempre vera, cioè non essendoci condizioni sulle possibili istanziazioni delle metavariabili.

Le due regole (istanziate)

$$\frac{x = 0 \quad x = 0 \supset x * y = 0}{x * y = 0} \quad \text{e} \quad \frac{x = y \wedge y = z \qquad x = y \wedge y = z \supset x = z}{x = z}$$

sono esempi di istanze del modus ponens in una teoria del primo ordine con eguaglianza. L'insieme delle regole o istanze è $\{(\{\mathrm{F}, \mathrm{F} \supset \mathrm{G}\}, \mathrm{G}) \mid \mathrm{F}, \mathrm{G} \in \mathit{FOF}_\Sigma(X)\}$. **Fine esempio**

Osservazione importante Molto spesso con sistema logico deduttivo o logico formale si intende un sistema deduttivo in cui la procedura di generazione dei teoremi è effettiva, cioè in cui l'insieme dei teoremi è ricorsivamente numerabile; diciamo *effettivo* tale sistema logico. Per questo una condizione (sufficiente) spesso imposta è che Reg sia finito e che ogni regola di inferenza sia un insieme ricorsivo, cioè che per ogni regola R sia decidibile se $(Pr, c) \in R$.

Diamo una condizione ancora più generale, anche se raramente occorre considerare sistemi logici siffatti.

Prop. 2.23 Sia $D = (\mathit{WFF}, \mathit{Reg})$ un sistema deduttivo in cui Reg è un insieme ricorsivamente numerabile e per ogni $R \in \mathit{Reg}$ è decidibile se $(Pr, c) \in R$. Allora D è effettivo, cioè l'insieme dei teoremi di D è ricorsivamente numerabile.

Dim. Per esercizio. $\square$

Molto spesso in un sistema logico l'insieme $\overline{\mathit{Reg}} \subseteq \mathit{Reg}$ delle regole di inferenza proprie (cioè che non sono assiomi) è finito, ciascuna è decidibile e l'insieme degli assiomi è ricorsivo; in questo caso il sistema logico (o teoria logica) si dice *assiomatico*. Si noti che, anche se la teoria è assiomatica, l'insieme dei teoremi non è in generale ricorsivo (provarlo per esercizio, utilizzando le macchine di Turing). Quando l'insieme dei teoremi è ricorsivo la *teoria* si dice *decidibile*.

Sono di fondamentale importanza le seguenti nozioni che mettono in relazione la deducibilità e la validità semantica.

Def. 2.24 Sia $D = (\mathit{WFF}, \mathit{Reg})$ un sistema deduttivo, $\mathcal{C}$ una classe di modelli ed $\mathcal{F} \subseteq \mathit{WFF}$. Assumiamo che sia definita per gli elementi di $\mathcal{C}$ una nozione di validità delle formule in WFF, indicata con $\models \subseteq \mathcal{C} \times \mathit{WFF}$.

- D si dice *sound* (o *corretto*) rispetto a $\mathcal{C}$ ed a $\mathcal{F}$ se

$$\text{per ogni } \mathrm{F} \in \mathcal{F} \quad \vdash^D \mathrm{F} \Rightarrow (\forall\, \mathrm{C} \in \mathcal{C} \quad \mathrm{C} \models \mathrm{F});$$

- D si dice *completo* rispetto a $\mathcal{C}$ ed a $\mathcal{F}$ se

$$\text{per ogni } \mathrm{F} \in \mathcal{F} \quad \forall\, \mathrm{C} \in \mathcal{C} \quad \mathrm{C} \models \mathrm{F}) \Rightarrow \vdash^D \mathrm{F}. \square$$

Poichè solo i sistemi sound sono sensati, spesso con completo si intende sound e completo.

Dal momento che un sistema deduttivo è un particolare sistema induttivo, applicando il principio d'induzione si ottiene facilmente il seguente risultato.

Def. 2.25 Con le notazioni della Def. 2.24, diciamo che (Pr, F) *è sound* rispetto a $\mathcal{C}$ se

$$(S) \qquad \text{per ogni } \mathrm{A} \in \mathcal{C} \quad [(\mathrm{A} \models \mathrm{G} \ \text{ per ogni } \mathrm{G} \in Pr) \Rightarrow \mathrm{A} \models \mathrm{F}].$$

Diciamo che R è *sound* rispetto a $\mathcal{C}$ se (Pr, F) è sound rispetto a $\mathcal{C}$ per ogni $(Pr, \mathrm{F}) \in R$. □

Prop. 2.26 Sia $D = (WFF, Reg)$ un sistema deduttivo e $\mathcal{C}$ una classe di modelli. Se R è sound rispetto a $\mathcal{C}$ per ogni $R \in Reg$, allora D è sound rispetto a $\mathcal{C}$ ed a WFF.

Dim. Basta applicare il principio d'induzione.□

Osservazione 1. Occorre fare attenzione alla differenza fra la nozione di soundness di una regola come definita da (S) e quella definita dalla seguente proprietà

$$(GS) \ \ (\mathcal{C} \models \mathrm{G} \ \text{ per ogni } \mathrm{G} \in Pr) \quad \Rightarrow \quad \mathcal{C} \models \mathrm{F}.$$

Si vede subito che $(S) \Rightarrow (GS)$. Che non valga il viceversa è banalmente provato prendendo $AX = \emptyset$. Allora $Mod(AX) = Alg_\Sigma$ per un qualche Σ e la regola $\dfrac{x = y}{\neg\, x = y}$ è (GS)-sound in Alg_Σ in quanto $Alg_\Sigma \not\models x = y$ e dunque (GS) è banalmente soddisfatta; mentre nell'algebra banale con i supporti ridotti ad un solo elemento la premessa è valida mentre non lo è la conseguenza. Naturalmente anche per la (GS)-soundness vale una proposizione analoga alla Prop. 2.26. Si noti ora che se D è un sistema sound rispetto a $\mathcal{C}$, allora ci si puó restringere a considerare regole che sono sound in entrambi i sensi; infatti D puó essere ristretto, senza variare l'insieme dei suoi teoremi, a quelle regole le cui premesse sono teoremi, cioè formule valide in $\mathcal{C}$.

Osservazione 2. Analoga attenzione occorre per distinguere un'implicazione "$\mathrm{F} \supset \mathrm{G}$" da una regola $(\{\mathrm{F}\}, \mathrm{G})$. $\mathcal{C} \models \mathrm{F} \supset \mathrm{G}$ significa che per ogni algebra A in $\mathcal{C}$ $\mathrm{A} \models \mathrm{F} \supset \mathrm{G}$ e cioè per ogni valutazione $v \colon X \to \mathrm{A}$, vale $\mathrm{A}, v \models \mathrm{F} \supset \mathrm{G}$; mentre la (S)-soundness in $\mathcal{C}$ di $(\{\mathrm{F}\}, \mathrm{G})$ significa che per ogni $\mathrm{A} \in \mathcal{C}$, $\mathrm{A} \models \mathrm{F} \Rightarrow \mathrm{A} \models \mathrm{G}$. Quindi $\mathcal{C} \models \mathrm{F} \supset \mathrm{G}$ implica la soundness di $(\{\mathrm{F}\}, \mathrm{G})$ in $\mathcal{C}$, ma non vale il viceversa. Per esempio, in generale un'istanza di$(\{\mathrm{F}\}, \forall\, x \,.\, \mathrm{F})$ non implica la validità di $\mathrm{F} \supset \forall\, x \,.\, \mathrm{F}$, come è facile verificare su semplici esempi.

Osservazione 3. Si noti anche che in logica si usa dire che, dato un insieme di formule Γ, una formula F è *conseguenza logica* di Γ (o che F è *implicata logicamente* da Γ), e si scrive $\Gamma \models \mathrm{F}$, esattamente quando $Alg_\Sigma \models \ \wedge\, \Gamma \supset \mathrm{F}$; occorre fare attenzione a non confondere $\Gamma \models \mathrm{F}$ con "dalla validità di Γ segue la validità di F" che corrisponde invece a dire che la regola (Γ, F) è sound, o secondo (S) o secondo (GS), rispetto a tutte le Σ-algebre.

Analoghe definizioni e distinzioni valgono se si considerano particolari classi di algebre al posto di Alg_Σ.

Osservazione 4. Infine si osservi che se $D = (WFF, Reg)$ e $\Gamma \vdash^D \mathrm{F}$ dove $\Gamma \subseteq WFF$, $F \in WFF$, allora si può aggiungere al sistema D la regola (Γ, F) e ciò lascia invariati i teoremi di D. □

2.4.2 Un sistema di deduzione hilbertiano

Def. 2.27 (Sistema completo di deduzione per la logica omogenea del primo ordine, stile hilbertiano)

È il sistema induttivo definito dalle seguenti regole di inferenza, suddivise come d'uso in schemi di assiomi e regole di inferenza propriamente dette.

(Schemi di) assiomi logici

1) $\mathrm{F} \supset (\mathrm{G} \supset \mathrm{F})$
2) $(\mathrm{F} \supset (\mathrm{G} \supset \mathrm{L})) \supset ((\mathrm{F} \supset \mathrm{G}) \supset (\mathrm{F} \supset \mathrm{L}))$
3) $(\neg \mathrm{G} \supset \neg \mathrm{F}) \supset ((\neg \mathrm{G} \supset \mathrm{F}) \supset \mathrm{G})$
4) $\forall\, x\,.\,\mathrm{F} \supset \mathrm{F}[t/x]$
   per ogni termine t libero per x in F, cioè tale che ogni occorrenza di una variabile in t resta libera in $\mathrm{F}[t/x]$. (*Istanziazione/Concretizzazione*)
5) $(\forall\, x\,.\,(\mathrm{F} \supset \mathrm{G})) \supset (\mathrm{F} \supset \forall\, x\,.\,\mathrm{G})$ se $x \notin \mathit{Free}(\mathrm{F})$

Regole di inferenza

i) $\dfrac{\mathrm{F} \quad \mathrm{F} \supset \mathrm{G}}{\mathrm{G}}$ (*Modus ponens*)

ii) $\dfrac{\mathrm{F}}{\forall\, x\,.\,\mathrm{F}}$ (*Generalizzazione*)

Indichiamo con FO il sistema logico del primo ordine così definito e con $\vdash^{\mathrm{FO}}$ l'associata relazione di deduzione. Una *teoria (o sistema) del primo ordine* con *assiomi propri* AX ($AX \subseteq FOF_\Sigma(X)$) è il sistema deduttivo ottenuto da FO aggiungendo gli assiomi AX; scriviamo $AX \vdash^{\mathrm{FO}} \mathrm{F}$ sse F appartiene all'insieme induttivo generato dal sistema con assiomi propri AX. □

Come già detto useremo spesso $\vdash^{\mathrm{FO}}$ o $AX \vdash^{\mathrm{FO}}$ per indicare anche i sistemi deduttivi a cui queste relazioni di deduzione si riferiscono.

In ambito algebrico si è particolarmente interessati alla deduzione logica di formule atomiche, cioè alla deduzione logica di atomi costruiti da predicati e da eguaglianze fra termini. Ció significa che le teorie logiche (teoria logica = assiomi propri + regole di inferenza [con eventuali assiomi logici]) che considereremo saranno teorie con uguaglianza (cioè con predicato di uguaglianza soddisfacente particolari assiomi).

Una *teoria omogenea del primo ordine con eguaglianza* è una teoria con un predicato binario "=" in cui sono teoremi le formule 6) e 7) seguenti. Quindi 6) e 7) possono essere considerate come assiomi logici di ogni teoria con uguaglianza.

Assiomi logici di ogni teoria con eguaglianza

6) $\forall\, x\,.\,(x = x)$ (*Riflessività*)
7) $x = y \supset (\mathrm{F}[x, x] \supset \mathrm{F}[x, y/x])$ (*Sostitutività*)
   ($\mathrm{F}[x, y/x]$ indica sostituzione di y al posto di alcune occorrenze di x, naturalmente senza collisioni, cioè a condizione che dopo la sostituzione ogni occorrenza di y derivante dalla sostituzione resti libera in $\mathrm{F}[x, y/x]$).

Indichiamo con FOE il sistema deduttivo del primo ordine con eguaglianza; nel seguito ometteremo FO e FOE sopra il simbolo $\vdash$ quando non c'è ambiguità.

Prop. 2.28 In ogni teoria del primo ordine con eguaglianza sono teoremi le formule;
$x = x, \qquad x = y \supset y = x, \qquad x = y \supset (y = z \supset x = z).\square$

Questa proposizione dice che la relazione definita dal predicato $=$ è in ogni modello (di una teoria del primo ordine con uguaglianza) un'equivalenza. Anzi, tenuto conto di 7), questa relazione risulta una congruenza. Peró non necessariamente questa equivalenza è un'uguaglianza; quando questo avviene il modello si dice *normale*. Si noti che, per come abbiamo dato la definizione di validità dell'uguaglianza in un'algebra, ogni algebra in $Mod(AX)$ deve essere un modello normale.

Richiamiamo qui alcuni risultati delle teorie omogenee (una sola sort) del primo ordine, che saranno utilizzati nel seguito.

- Ogni teoria del primo ordine consistente, cioè in cui non tutte le formule sono teoremi, ha un modello di cardinalità α, per ogni $\alpha \geq \aleph_0$ (cardinalità del numerabile); ogni teoria del primo ordine che ha un modello non vuoto è consistente e quindi ha un modello di cardinalità α per ogni $\alpha \geq \aleph_0$ (Teorema di Skolem-Löwenheim).

- Ad ogni modello M di una teoria con eguaglianza indicata con $=$ è ovviamente associato un modello normale M$'$, quoziente di M rispetto alla congruenza $=^{\mathrm{M}}$; ogni teoria con eguaglianza che sia consistente ha un modello finito o numerabile *normale* (cioè in cui l'eguaglianza è interpretata come identità); inoltre ogni teoria con eguaglianza che ha un modello infinito normale, ne ha uno numerabile normale.

- I sistemi $AX \vdash^{\mathrm{FO}}$ e $AX \vdash^{\mathrm{FOE}}$ sono sound e completi per ogni AX (teorema di completezza di Gödel).

2.4.3 Un sistema di deduzione naturale a sequenti

Vediamo ora un altro sistema di deduzione per la logica del primo ordine. È un sistema di "deduzione naturale" nel senso che l'insieme di assiomi e regole d'inferenza è sufficientemente ricco da dedurre formule valide in un modo molto "naturale" (contrariamente a quanto succede ad esempio per il sistema hilbertiano della Def. 2.27. Le formule ben formate di questo sistema sono della forma

$$\Gamma \Rightarrow \mathrm{G}$$

dove Γ è un insieme di formule del primo ordine e G una formula del primo ordine.

$\Gamma \Rightarrow \mathrm{G}$ si dice *sequente* (sequent) o *formula di assunzione* (assumption formula).
$\Gamma \Rightarrow \mathrm{G}$ è un'abbreviazione per $(\wedge \Gamma) \supset \mathrm{G}$, cioè $A_1, \ldots, A_n \Rightarrow \mathrm{G}$ sta per $A_1 \wedge \ldots \wedge A_n \supset \mathrm{G}$. La semantica di $\Gamma \Rightarrow \mathrm{G}$ è perciò data da $\mathrm{A} \models \Gamma \Rightarrow \mathrm{G}$ sse $\mathrm{A} \models (\wedge \Gamma) \supset \mathrm{G}$, dove A è una Σ-algebra; notare che $\Rightarrow \mathrm{G}$ (quando cioè $\Gamma = \emptyset$) sta per G.

Estendiamo le formule del primo ordine FO e FOE con due costanti **true** e **false** (vero e falso); ovviamente, per ogni A ed ogni v:
$\mathrm{A}, v \models \mathbf{true} \qquad \mathrm{A}, v \not\models \mathbf{false}$.

Def. 2.29 Sistema di deduzione naturale con sequenti (di Gentzen).

Assiomi e regole di base

a) **Assiomi**

$$\frac{}{\Gamma, \mathrm{F} \Rightarrow \mathrm{F}} \quad \textit{(Assioma dell'assunzione)}$$

$$\frac{\Gamma \Rightarrow \mathrm{F}}{\Gamma, G \Rightarrow \mathrm{F}} \quad \textit{(Introduzione di un'assunzione)}$$

$$\frac{\Gamma, \mathrm{F} \Rightarrow \mathrm{G} \quad \Gamma, \neg \mathrm{F} \Rightarrow \mathrm{G}}{\Gamma \Rightarrow \mathrm{G}} \quad \textit{(Eliminazione di un'assunzione)}$$

b) **Simboli di verità**

$$\frac{}{\Gamma \Rightarrow \mathbf{true}} \quad \textit{(Assioma del vero)}$$

$$\frac{}{\Gamma \Rightarrow \neg \ \mathbf{false}} \quad \textit{(Assioma del falso)}$$

Regole per i connettivi

a) **Regole per** $\vee$

$$\frac{\Gamma \Rightarrow \mathrm{F}}{\Gamma \Rightarrow \mathrm{F} \vee \mathrm{G}} \qquad \frac{\Gamma \Rightarrow \mathrm{G}}{\Gamma \Rightarrow \mathrm{F} \vee \mathrm{G}} \quad \textit{(Introduzione dell'} \vee \textit{)}$$

$$\frac{\Gamma, \mathrm{F} \Rightarrow H \quad \Gamma, \mathrm{G} \Rightarrow H \quad \Gamma \Rightarrow \mathrm{F} \vee \mathrm{G}}{\Gamma \Rightarrow H} \quad \textit{(Eliminazione dell'} \vee \textit{)}$$

b) **Regole per** $\wedge$

$$\frac{\Gamma \Rightarrow \mathrm{F} \quad \Gamma \Rightarrow \mathrm{G}}{\Gamma \Rightarrow \mathrm{F} \wedge \mathrm{G}} \quad \textit{(Introduzione dell'} \wedge \textit{)}$$

$$\frac{\Gamma \Rightarrow \mathrm{F} \wedge \mathrm{G}}{\Gamma \Rightarrow \mathrm{F}} \qquad \frac{\Gamma \Rightarrow \mathrm{F} \wedge \mathrm{G}}{\Gamma \Rightarrow \mathrm{G}} \quad \textit{(Eliminazione dell'} \wedge \textit{)}$$

c) **Regole per** $\supset$

$$\frac{\Gamma, \mathrm{F} \Rightarrow \mathrm{G}}{\Gamma \Rightarrow \mathrm{F} \supset \mathrm{G}} \quad \textit{(Introduzione del } \supset \textit{ (regola di scarico))}$$

$$\frac{\Gamma \Rightarrow \mathrm{F} \quad \Gamma \Rightarrow \mathrm{F} \supset \mathrm{G}}{\Gamma \Rightarrow \mathrm{G}} \quad \textit{(Eliminazione del } \supset \textit{ (modus ponens))}$$

d) **Regole per** $\neg$

$$\frac{\Gamma, \mathrm{F} \Rightarrow \mathrm{G} \quad \Gamma, \mathrm{F} \Rightarrow \neg \mathrm{G}}{\Gamma \Rightarrow \neg \mathrm{F}} \quad \textit{(Introduzione del } \neg \textit{ (riduzione ad assurdo))}$$

$$\frac{\Gamma \Rightarrow \mathrm{F} \quad \Gamma \Rightarrow \neg \mathrm{F}}{\Gamma \Rightarrow \mathrm{G}} \quad \textit{(Eliminazione del } \neg \textit{)}$$

e) **Regole per** $\neg\ \neg$

$$\frac{\Gamma \Rightarrow \mathrm{F}}{\Gamma \Rightarrow \neg \neg \mathrm{F}} \quad \textit{(Introduzione del } \neg\ \neg \textit{)}$$

$$\frac{\Gamma \Rightarrow \neg \neg \mathrm{F}}{\Gamma \Rightarrow \mathrm{F}} \quad \textit{(Eliminazione del } \neg\ \neg \textit{)}$$

f) Regole per $\equiv$

$$\frac{\Gamma \Rightarrow F \supset G \quad \Gamma \Rightarrow G \supset F}{\Gamma \Rightarrow F \equiv G} \qquad (\textit{Introduzione dell'}\equiv)$$

$$\frac{\Gamma \Rightarrow F \equiv G}{\Gamma \Rightarrow F \supset G} \quad \frac{\Gamma \Rightarrow F \equiv G}{\Gamma \Rightarrow G \supset F} \qquad (\textit{Eliminazione dell'}\equiv)$$

Regole per i quantificatori

a) Regole per $\forall$

(*Introduzione del* $\forall$ *(generalizzazione)*)

$$\frac{\Gamma \Rightarrow F}{\Gamma \Rightarrow \forall\, x \,.\, F} \quad x \text{ non occorre libera in alcun elemento di } \Gamma)$$

(*Eliminazione di* $\forall$ *(concretizzazione/istanziazione)*)

$$\frac{\Gamma \Rightarrow \forall\, x \,.\, F}{\Gamma \Rightarrow F[t/x]} \quad t \text{ libero per } x \text{ in } F$$

b) Regole per $\exists$

$$\frac{\Gamma \Rightarrow F[t/x]}{\Gamma \Rightarrow \exists\, x \,.\, F} \quad t \text{ libero per } x \text{ in } F \qquad (\textit{Introduzione dell'}\exists\,)$$

$$\frac{\Gamma \Rightarrow \exists\, x \,.\, F \quad \Gamma, F \Rightarrow G}{\Gamma \Rightarrow G} \quad x \text{ non libera in } \Gamma \text{ nè in } G \qquad (\textit{Eliminazione dell'}\exists\,)$$

Regole per l'uguaglianza

$$\frac{}{\Gamma \Rightarrow t = t} \qquad (\textit{Assioma dell'}=)$$

$$\frac{\Gamma \Rightarrow t_1 = t_2}{\Gamma \Rightarrow F[t_1/x] \equiv F[t_2/x]} \quad t_1,\, t_2 \text{ liberi per } x \text{ in } F \qquad (\textit{Regola dell'}=)$$

2.4.4 Il caso eterogeneo

È facile vedere che il sistema logico FO puó essere interpretato come sistema many-sorted, semplicemente ammettendo che le formule su Σ ed X siano many-sorted; analogamente per FOE assumendo che per ogni $s \in S$ esista un predicato di eguaglianza $=_{ss}$ di tipo ss. Tuttavia la soundness puv́enire a mncare e occorre dunque fare attenzione.

È facile vedere che in generale un sistema deduttivo per il caso omogeneo (unica sort) può non essere sound se applicato direttamente al caso eterogeneo. Il seguente esempio (dovuto a Huet, Goguen e Meseguer), chiarisce la situazione per il sistema FOE.

Esempio 2.30 Sia ΣH la segnatura $(\{bool, alpha\}, OPH, PRH)$, dove
$OPH_{\Lambda,bool} = \{True, False\}$, $OPH_{bool,bool} = \{Not\}$, $OPH_{bool\ bool,bool} = \{\ And\ ,\ Or\ \}$,
$OPH_{alpha,bool} = \{F\}$ e $OPH_{w,s} = \emptyset$ per ogni altro w, s;
$PRH_w = \emptyset$ per ogni w;
CH sia il seguente insieme di equazioni:
$Not\ Not\ z = z$, $z\ And\ z = z$, $z\ Or\ z = z$, $z\ Or\ Not\ z = True$,
$z\ And\ Not\ z = False$, $F(y) = Not\ F(y)$

ed indichiamo con HGM la specifica $(\Sigma H, CH)$.

Si vede facilmente che

$CH \vdash^{\mathrm{FOE}} True = False$.

Tuttavia $True = False$ è falso nel modello A in cui $\mathrm{A}_{bool} = \{vero, falso\}$, $\mathrm{A}_{alpha} = \emptyset$, $F^{\mathrm{A}} = \emptyset$ e le altre operazioni sono definite come al solito. **Fine esempio**

Un'analisi della deduzione con $CH \vdash^{\mathrm{FOE}}$ mostra che non è sound la regola di modus ponens che permette di ottenere $True = False$ per transitività da $True = F(y)$ ed $F(y) = False$ (entrambe valide in ogni modello); come pure è non sound l'assioma di istanziazione che permette di ottenere da $\forall\ y\ .\ True = False$, valida in ogni modello, $True = False$. Tuttavia si puó capire che il fallimento del modus ponens avviene solo se si considerano formule non quantificate, mentre l'assioma di istanziazione fallisce anche su formule quantificate.

Di qui si intuisce che un sistema logico sound per teorie many-sorted deve essenzialmente modificare l'assioma di istanziazione; poichè il fallimento di questo è dovuto all'istanziazione su termini a cui non sempre corrisponde un elemento di un'algebra, occorre correggerlo vincolando l'istanziazione a termini che denotano un valore in ogni algebra oppure considerare solamente segnature i cui modelli non hanno carrier vuoti (segnature "sensible") e dove quindi ogni termine denota un elemento.

In generale, invece di discutere un sistema sound e completo per logiche many-sorted, facciamo vedere come, traducendo le formule many-sorted in formule one-sorted, si puó usare il sistema $\vdash^{\mathrm{FOE}}$ per una deduzione sound e completa, come è stato osservato da Wirsing e Padawitz. Nel caso poi che si considerino segnature i cui modelli non hanno supporti vuoti, si puó tranquillamente usare il sistema deduttivo analogo a quello del caso omogeneo, pur di rispettare i tipi.

Def. 2.31 Per ogni segnatura many-sorted Σ, definiamo una segnatura one-sorted Σ', ottenuta da Σ unificando le sort e aggiungendo la famiglia dei predicati unari $\{p_s\}_{s \in S}$; analogamente se X è una S-famiglia disgiunta di variabili, X' è l'insieme $\bigcup_{s \in S} X_s$.

Per ogni formula $\mathrm{F} \in FOF_{\Sigma}(X)$ definiamo induttivamente la sua traduzione $\mathrm{F}' \in FOF_{\Sigma'}(X')$, come segue.

- Se F è atomica e $Free(\mathrm{F}) = \{x_1{:}\, s_1, \ldots, x_n{:}\, s_n\}$ $\mathrm{F}' =_{def} p_{s_1}(x_1)\ \wedge\ \ldots\ \wedge\ p_{s_n}(x_n)\ \supset\ \mathrm{F}$ (se $Free(\mathrm{F}) = \emptyset$, allora $\mathrm{F}' = \mathrm{F}$);
- $(\mathrm{F}\ \wedge\ \mathrm{G})' =_{def} \mathrm{F}'\ \wedge\ \mathrm{G}'$ $\qquad (\mathrm{F}\ \vee\ \mathrm{G})' =_{def} \mathrm{F}'\ \vee\ \mathrm{G}'$ $\qquad (\neg\ \mathrm{F})' =_{def}\ \neg\ \mathrm{F}'$;
- $(\forall\ x{:}\, s\ \mathrm{F})' =_{def} \forall\ x\ .\ (p_s(x)\ \supset\ \mathrm{F}')$;
- $(\exists\ x{:}\, s\ \mathrm{F})' =_{def} \exists\ x\ .\ (p_s(x)\ \wedge\ \mathrm{F}')$. $\square$

Indichiamo con *WFAX* (Well Formedness Axioms) la famiglia di assiomi

$$p_{s_1}(x_1)\ \wedge\ \ldots\ \wedge\ p_{s_n}(x_n)\ \supset\ p_s(op(x_1, \ldots, x_n)) \quad op{:}\, s_1 \times \ldots \times s_n \to s$$

(questi assiomi definiscono le correttezza della struttura dei termini many-sorted).

Si puó allora provare il seguente teorema di soundness e completezza.

Teorema 2.32 Per ogni insieme AX di assiomi many-sorted su Σ

$$Mod(AX) \models \mathrm{F} \qquad \text{sse} \qquad AX' \cup WFAX \vdash^{\mathrm{FOE}} \mathrm{F}' \quad \text{per ogni } \mathrm{F} \in FOF_{\Sigma}(X). \square$$

Se si considera l'esempio 2.30, si capisce facilmente che, da $CH' \cup WFAX$ si riesce a dedurre $p_{\alpha}(y)\ \supset\ True = False$, ma non si riesce ad eliminare $p_{\alpha}(y)$ dalla premessa. In generale nella deduzione a partire da $AX' \cup WFAX$, la quantificazione esplicita delle variabili $(\forall\ Y)$ viene sostituita dalle corrispondenti asserzioni di appartenenza alla sort $p_s(y)$ per ogni $y \in Y_s$.

2.4.5 Deduzione logica in Gen_Σ

Occorre fare molta attenzione quando si considerano soltanto modelli term-generated di una specifica.

Premettiamo alcuni risultati elementari, usando, per $\mathcal{K} \subseteq Alg_\Sigma$, le notazioni:
$Atom_X(\mathcal{K}) = \{\alpha \mid \alpha \in A_\Sigma(X)\} \cap Th(\mathcal{K})$
$Atom(\mathcal{K}) = \cup\{Atom_X(\mathcal{K}) \mid X$ famiglia S-indiciata di variabili $\}$
$GAtom(\mathcal{K}) = Atom_\emptyset(\mathcal{K})$.

Prop. 2.33 Siano AX, $UAX \subseteq FOF_\Sigma(X)$ e le formule di UAX siano senza quantificatori.

1. $Th_\mathcal{F}(Mod(AX)) \subseteq Th_\mathcal{F}(GMod(AX))$ per ogni $\mathcal{F} \subseteq FOF_\Sigma(X)$;
2. $\mathrm{A} \in Mod(UAX) \Rightarrow \emptyset_\mathrm{A} \in GMod(UAX)$

   (dove $\emptyset_\mathrm{A}$ indica la sottoalgebra term-generated di A, cioè generata dall'insieme vuoto);
3. $Mod(UAX) \models \alpha$ sse $GMod(UAX) \models \alpha$ per ogni $\alpha \in A_\Sigma$;
4. $GMod(UAX) = GMod(UAX_g)$

   dove $UAX_g =_{def}$

   $\{\mathrm{G}[t_1/x_1, \ldots t_n/x_n]) \mid \mathrm{G} \in UAX, Free(\mathrm{G}) = \{x_1{:}\, s_1, \ldots, x_n{:}\, s_n\}, t_i \in (\mathrm{T}_\Sigma)_{s_i}, i = 1, \ldots, n\}$;
5. per ogni $\mathrm{A}, \mathrm{B} \in Gen_\Sigma$, $\equiv_\mathrm{A} \subseteq \equiv_\mathrm{B} \Rightarrow Atom_X(\mathrm{A}) \subseteq Atom_X(\mathrm{B})$; quindi per ogni $\mathrm{A} \in Gen_\Sigma$, $Atom_X(\mathrm{A}) = Atom_X(GMod(\equiv_\mathrm{A}))$; in particolare $GAtom(GMod(\equiv_\mathrm{A})) = GAtom(\mathrm{A}) = \equiv_\mathrm{A}$.

Dim. 1, 2, 3, 4 per esercizio.

Se $\mathrm{B} \in Gen_\Sigma$, $t \in \mathrm{T}_\Sigma(X)$ e $v{:}\, X \to \mathrm{B}$ è una valutazione, indichiamo con $t_{x,v,\mathrm{B}}$ per $x \in X$, un termine di T_Σ (che esiste perché $\mathrm{B} \in Gen_\Sigma$) tale che $t^\mathrm{B}_{x,v,\mathrm{B}} = v(x)$, e con $t_{x,\mathrm{B}}$ il termine ottenuto da t sostituendo $t_{x,v,\mathrm{B}}$ ad ogni x in t. Allora è facile provare per induzione che $eval^{\mathrm{B},v}(t) = eval^\mathrm{B}(t_{v,\mathrm{B}})$.

Sia ora $\mathrm{A} \in Gen_\Sigma$, $\equiv_\mathrm{A} \subseteq \equiv_\mathrm{B}$. Allora (v., Prop. 1.34), esiste unico $h{:}\, \mathrm{A} \to \mathrm{B}$ e quindi $eval^\mathrm{B} = h \circ eval^\mathrm{A}$. Perció se $\alpha \in Atom_X(\mathrm{A})$, proviamo che $\alpha \in Atom_X(\mathrm{B})$. Si sono previsti due casi.

- $\alpha = (t = t')$, per ogni valutazione $v{:}\, X \to \mathrm{B}$

  $eval^{\mathrm{B},v}(t) = eval^\mathrm{B}(t_{v,\mathrm{B}}) = h \circ eval^\mathrm{A}(t_{v,\mathrm{B}}) = h(eval^\mathrm{A}(t_{v,\mathrm{B}})) = h(eval^\mathrm{A}(t'_{v,\mathrm{B}})) = eval^{\mathrm{B},v}(t')$; dunque $\mathrm{B}, v \models t = t'$ per ogni v e cioè $t = t' \in Atom_X(\mathrm{B})$.

- $\alpha = pr(t_1, \ldots, t_n)$, per ogni valutazione $v{:}\, X \to \mathrm{B}$

  $(eval^{\mathrm{B},v}(t_1), \ldots, eval^{\mathrm{B},v}(t_n)) =$

  $(eval^\mathrm{B}((t_1)_{v,\mathrm{B}}), \ldots, eval^\mathrm{B}((t_n)_{v,\mathrm{B}})) = (h(eval^\mathrm{A}((t_1)_{v,\mathrm{B}})), \ldots, h(eval^\mathrm{A}((t_n)_{v,\mathrm{B}}))$ e questo appartiene a pr^B perchè

  $(eval^{\mathrm{A},v}(t_1), \ldots, eval^{\mathrm{A},v}(t_n)) = (eval^\mathrm{A}((t_1)_{v,\mathrm{B}}), \ldots, eval^\mathrm{A}((t_n)_{v,\mathrm{B}})) \in pr^\mathrm{A}$ ed h è un omomorfismo; dunque $\mathrm{B}, v \models pr(t_1, \ldots, t_n)$ per ogni v e cioè $pr(t_1, \ldots, t_n) \in Atom_X(\mathrm{B})$

Se ora $\mathrm{B} \in GMod(\equiv_\mathrm{A})$, essendo $\equiv_\mathrm{B}$ l'insieme degli atomi ground che valgono in B e dovendo B soddisfare $\equiv_\mathrm{A}$, deve essere $\equiv_\mathrm{A} \subseteq \equiv_\mathrm{B}$ e dunque per quanto visto prima $Atom_X(\mathrm{A}) \subseteq Atom_X(\mathrm{B})$, da cui $Atom_X(\mathrm{A}) \subseteq Atom_X(GMod(\equiv_\mathrm{A}))$.

Dal momento che vale anche la relazione inversa (v. Prop. 2.7 G4), si ha l'eguaglianza per ogni X e quindi anche per $X = \emptyset$ e perció anche per l'ultima parte. $\square$

Studiamo ora la deducibilità equazionale in Gen_Σ.

Prop. 2.34 1. Se per $AX \subseteq FOF_\Sigma(X)$, $AX \vdash$ è un sistema sound per $FOF_\Sigma(X)$ rispetto a $Mod(AX)$, allora $AX \vdash F \Rightarrow GMod(AX) \models F$ per ogni $F \in FOF_\Sigma(X)$; non vale in generale il viceversa, neppure se $AX \vdash$ è completo rispetto a $Mod(AX)$;

2. se UAX è un insieme di formule senza quantificatori e $UAX \vdash$ è completo per gli atomi rispetto a $Mod(UAX)$

$$UAX \vdash \alpha \qquad \text{sse} \qquad GMod(UAX) \models \alpha \ \text{ per ogni } \alpha \in A_\Sigma;$$

3. esistono AX (anche solo costituiti da insiemi di atomi ground o da insiemi finiti di atomi) per cui non esiste un sistema assiomatico deduttivo sound e completo per gli atomi in Σ, X rispetto a $GMod(AX)$

Dim.

1 La prima parte segue dalla Prop. 2.33 1; che non valga il viceversa si puó vedere ponendo $AX = \emptyset$ e Σ la segnatura con un'unica sort ed un'unica operazione zeraria $a\colon s$; si vede subito che $GMod(\emptyset) = Gen_\Sigma \models x = a$; e tuttavia non si puó ottenere $\emptyset \vdash x = a$ perché $x = a$ non è valido in Alg_Σ.

2 Si ottiene immediatamente da Prop. 2.33 3, ricordando che $UAX \vdash$ è sound e completo.

3 Si puó dimostrare che esistono algebre $\mathrm{A} \in Gen_\Sigma$ per cui $Atom(\mathrm{A})$ non è ricorsivamente numerabile; allora, essendo per la Prop. 2.33 5, $Atom(\mathrm{A}) = Atom(GMod(\equiv_{\mathrm{A}}))$, posto $AX = \equiv_{\mathrm{A}}$ si ha la prima parte dell'asserzione, poiché ogni sistema assiomatico è effettivo, cioè l'insieme dei teoremi è ricorsivamente numerabile.

Si noti che $\equiv_{\mathrm{A}}$ è un insieme in generale infinito (e non ricorsivo, come vedremo in seguito). Tuttavia, si puó anche provare che esistono specifiche SP consistenti in un insieme finito di equazioni tali che $Atom(\mathrm{I}_{\mathrm{SP}})$ non è ricorsivamente numerabile, dove I_{SP} indica un'algebra iniziale in $Mod(\mathrm{SP})$. Ora $\mathrm{A} \in GMod(\mathrm{SP}) \Rightarrow \exists\, h\colon \mathrm{I}_{\mathrm{SP}} \to \mathrm{A} \Rightarrow \equiv_{\mathrm{I}_{\mathrm{SP}}} \subseteq \equiv_{\mathrm{A}}$; dunque, per la Prop. 2.33 5, $Atom(\mathrm{I}_{\mathrm{SP}}) \subseteq Atom(GMod(\mathrm{SP}))$ e quindi ancora $Atom(\mathrm{I}_{\mathrm{SP}}) = Atom(GMod(\mathrm{SP}))$. Perció gli atomi validi in $GMod(\mathrm{SP})$ non sono riscorsivamente numerabili e non possono essere dedotti da alcun sistema assiomatico. $\square$

Osservazione. L'asserzione 3 della Prop. 2.34 è assai forte in quanto asserisce l'incompletezza dei sistemi effettivi rispetto a una classe ristretta di formule, gli atomi, ed a un'ampia classe di modelli, tutti quelli term-generated.

Se ci si limita a considerare formule qualunque ed un solo modello, allora abbiamo vari celebri risultati di incompletezza essenzialmente dovuti a Gödel (1931), che possono essere sintetizzati nel modo seguente.

– L'insieme delle formule valide nel modello standard dell'aritmetica ($\mathbb{N}$, con le operazioni $0^{\mathbb{N}}$, $Succ^{\mathbb{N}}$, $+^{\mathbb{N}}$, $*^{\mathbb{N}}$ usuali) non è ricorsivamente numerabile; quindi non esiste un sistema effettivo sound e completo per queste formule rispetto a $\mathbb{N}$.

– Sia D un sistema deduttivo sound per le formule e $\mathbb{N}$, soddisfacente le seguenti proprietà

  a) l'insieme degli assiomi è ricorsivo;

  b) le prove possono essere effettivamente numerate;

  c) è decidibile se una prova è prova per una formula.

Allora si puó costruire una formula chiusa F che non è deducibile in D, ma è vera in $\mathbb{N}$ (e di conseguenza neppure $\neg$ F è deducibile in D). Una formula di questo tipo si dice *indecidibile*. (È interessante notare che la formula F costruita da Gödel ha come interpretazione in $\mathbb{N}$ "non esiste una prova in D per F", che è quindi vera, dal momento che F non è deducibile in D; questa formula è un'astuta versione del paradosso del mentitore "Io sto mentendo"). Si puó dimostrare che il risultato precedente si estende a ogni teoria che includa l'aritmetica sempre relativamente ad un sistema D che soddisfi le proprietà a), b), c). □

È facile capire perché un sistema $AX \vdash$ sound e completo rispetto ad Alg_Σ non lo è rispetto a Gen_Σ. Consideriamo Σ omogenea con le sole operazioni zerarie $\{a, b\}$ ed $AX = \{a = b\}$; allora si ha che $GMod(AX) \models x = y$ e tuttavia $x = y$ non è deducibile in $AX \vdash$. Ora $x = y$ è valida in $GMod(AX)$ perché vale $a = b$, cioè $(x = y)[t_1/x_1, t_2/x_2]$ per ogni coppia di termini t_1, t_2 in T_Σ (in questo caso solo a e b). Per poter dedurre $x = y$ possiamo perció pensare di aggiungere una regola infinitaria (ω-regola) di inferenza, detta *induzione infinitaria* o *principio di generalizzazione*

$$\mathrm{II} \quad \frac{\mathrm{F}[t/x] \quad \text{per ogni } t \in \mathrm{T}_\Sigma}{\mathrm{F}} \quad \mathrm{F} \in FOF_\Sigma(X)$$

dove al solito $\mathrm{F}[t/x]$ indica la formula ottenuta da F sostituendo t ad ogni occorrenza libera di x. Allora nell'esempio, poiché $AX \vdash a = b$, applicando due volte la regola II deduciamo $x = y$.

Vediamo l'applicazione ad un caso piú interessante; consideriamo la specifica degli interi con le operazioni $0\colon \to int$; $Pred, Succ\colon int \to int$ e gli assiomi AX

$Succ(Pred(0)) = 0$
$Pred(Succ(0)) = 0$
$Succ(Pred(x)) = x \supset Succ(Pred(Pred(x))) = Pred(x)$
$Pred(Succ(x)) = x \supset Pred(Succ(Succ(x))) = Succ(x)$

Si prova facilmente, per induzione sulla struttura dei termini, che $AX \vdash^{\mathrm{FOE}} Pred(Succ(t)) = t$ per ogni $t \in \mathrm{T}_\Sigma$; tuttavia non si puó ottenere $AX \vdash^{\mathrm{FOE}} Pred(Succ(x)) = x$.

Indicando invece con $AX \vdash_{\mathrm{II}}$ il sistema ottenuto da $AX \vdash$ con l'aggiunta della ω-regola II, si ha applicando II

$$\frac{Pred(Succ(t)) = t \quad \text{per ogni } t \in \mathrm{T}_\Sigma}{Pred(Succ(x)) = x}$$

e quindi, potendosi dedurre le premesse, cioè essendo $AX \vdash_{\mathrm{II}} Pred(Succ(t)) = t$ per ogni $t \in \mathrm{T}_\Sigma$, si ha la conseguenza $AX \vdash_{\mathrm{II}} Pred(Succ(x)) = x$.

Riassumiamo la discussione precedente in una proposizione, suddivisa in due parti per comodità tecnica.

Prop. 2.35

1. Se UAX è un insieme di assiomi senza quantificatori ed $UAX \vdash$ è un sistema sound e completo per gli atomi rispetto a $Mod(UAX)$, allora per ogni $\alpha \in A_\Sigma(X)$

$$UAX \vdash_{\mathrm{II}} \alpha \qquad \text{sse} \qquad GMod(UAX) \models \alpha;$$

2. se $AX \vdash$ è un sistema sound e completo per $FOF_\Sigma(X)$ rispetto a $Mod(AX)$, allora per ogni $\mathrm{F} \in FOF_\Sigma(X)$

$$AX \vdash_{\mathrm{II}} \mathrm{F} \qquad \text{sse} \qquad GMod(AX) \models \mathrm{F}.$$

Dim.

1 La soundness è ovvia; proviamo la completezza; se $GMod(UAX) \models \alpha$, allora per la definizione di validità $GMod(UAX) \models \alpha'$ per ogni α' ottenuto da α istanziando le variabili con termini ground. Allora essendo $UAX \vdash$ completo per gli atomi rispetto a $Mod(UAX)$, per la Prop. 2.33 3 si ha che $UAX \vdash \alpha'$, e quindi, applicando una volta II, anche $UAX \vdash_{\mathrm{II}} \alpha$.

2 Analogamente. $\square$

2.5 Specifiche condizionali (con clausole di Horn)

2.5.1 Deduzione per specifiche condizionali

Fra le specifiche hanno particolare importanza, per le loro proprietà e flessibilità di impiego, quelle dette condizionali. Una formula appartenente a $FOF_\Sigma(X)$ avente forma

$$\alpha_1 \ \wedge \ \ldots \ \wedge \ \alpha_n \ \supset \ \alpha_{n+1}$$

dove $n \geq 0$ ed $\alpha_i \in A_\Sigma(X)$ $i = 1, \ldots, n$ è detta *assioma condizionale*.

Notare che gli atomi sono particolari tipi di assiomi condizionali ottenuti nel caso $n = 0$.

Un assioma condizionale si dice anche *clausola di Horn stretta*; puó essere scritta nella forma logicamente equivalente $(\bigvee_{i=1}^{n} \neg \alpha_i) \ \vee \ \alpha_{n+1}$.

Le formule del tipo $\alpha_1 \ \wedge \ \ldots \ \wedge \ \alpha_n \ \supset \ e$ dove e è α_{n+1} oppure $\neg \alpha_{n+1}$, equivalentemente di forma $(\bigvee_{i=1}^{n+1} \neg \alpha_i)$ oppure $(\bigvee_{i=1}^{n} \neg \alpha_i) \ \vee \ \alpha_{n+1}$, si dicono invece *clausole di Horn*.

Se gli atomi α_i sono tutte equazioni, allora una clausola di Horn si dice equazionale.

Def. 2.36 Una specifica (Σ, AX) è detta *condizionale* se AX è un insieme di assiomi condizionali su Σ. Una specifica è detta *equazionale* se gli assiomi sono della forma $t = t'$, con $t, t' \in \mathrm{T}_\Sigma(X)$. $\square$

Def. 2.37 Il *sistema deduttivo condizionale di Birkhoff* o *sistema condizionale ordinario* associato a Σ, indicato da $\vdash^{\mathsf{B}}$, è costituito dalle seguenti regole di inferenza, dove gli atomi si intendono ben formati su una famiglia di variabili X indiciata su S.

equivalenza

B1 $\dfrac{}{t = t}$ B2 $\dfrac{t = t'}{t' = t}$ B3 $\dfrac{t = t' \quad t' = t''}{t = t''}$

congruenza

B4 $\dfrac{t_i = t_i' \ \ i = 1, \ldots, n}{op(t_1, \ldots, t_n) = op(t_1', \ldots, t_n')}$ B4′ $\dfrac{t_i = t_i' \ \ i = 1, \ldots, n \quad pr(t_1, \ldots, t_n)}{pr(t_1', \ldots, t_n')}$

sostituzione

B5 $\dfrac{\mathrm{F}}{\overline{v}(\mathrm{F})}$ $v: X \to \mathrm{T}_\Sigma(X)$ $\overline{v}$ estensione omomorfa di v, F formula condizionale

modus ponens

B6 $\dfrac{\alpha_1 \ \wedge \ \ldots \ \wedge \ \alpha_n \ \supset \ \alpha \qquad \alpha_i \quad i = 1, \ldots, n}{\alpha}$ $\square$

Esempio 2.38 Si consideri la specifica INT definita nell'esempio 2.12. Allora si puó ottenere

$$IAX \vdash^{\mathrm{B}} Succ(Pred(Succ(0)) = Succ(0)$$

con la seguente prova lineare

$Succ(Pred(x)) = x$	assioma proprio;
$Succ(Pred(Succ(0))) = Succ(0)$	da B5 con v tale che $v(x) = Succ(0)$ e $v(y) = y$ per ogni $y \neq x$.

Fine esempio

Il sistema $\vdash^{\mathrm{B}}$ è l'usuale sistema di deduzione equazionale usato in aritmetica ("Calcolo delle espressioni" nella scuola media), in algebra e in generale nella matematica classica.

È un sistema particolarmente semplice per specifiche condizionali, derivato dal sistema FOE, logica omogenea del primo ordine con eguaglianza. Tutte le volte che non c'è differenza tra deduzione logica eterogenea e omogenea conviene usare questo sistema semplificato. In seguito vedremo delle condizioni perchè questo avvenga. Tuttavia in generale, come già avveniva per il sistema FOE, il sistema $\vdash^{\mathrm{B}}$ non è sound, come si vede subito sullo stesso esempio HGM già utilizzato per FOE.

Teorema 2.39 Sia SP $= (\Sigma, C)$ una specifica condizionale tale che
per ogni $s, s' \in S$ $(\mathrm{T}_\Sigma)_s \neq \emptyset)$ o $S = \{s\}$.
Allora SP $\vdash^{\mathrm{B}}$ è sound rispetto agli atomi su Σ ed a *Mod*(SP), cioè per ogni $\alpha \in A_\Sigma(X)$

$$\mathrm{SP} \vdash^{\mathrm{B}} \alpha \Rightarrow \text{ per ogni } \mathrm{M} \in Mod(\mathrm{SP})\ \mathrm{M} \models \alpha.$$

Dim. Per induzione matematica sul sistema di deduzione (Prop. 2.26).□

Def. 2.40 Sia X una famiglia di variabili indiciata su S e C un insieme di assiomi condizionali su Σ ed X. Allora $\equiv_{C(X)}$ è la congruenza su $\mathrm{T}_\Sigma(X)$ definita da:

- $t\ (\equiv_{C(X)})_s\ t' \Leftrightarrow C \vdash^{\mathrm{B}} t = t'$ e
- $(t_1, \ldots, t_n) \in (\equiv_{C(X)})_{pr} \Leftrightarrow C \vdash^{\mathrm{B}} pr(t_1, \ldots, t_n)$.

Se $X = \emptyset$ allora $\equiv_{C(X)}$ viene indicata semplicemente da $\equiv_C$.□

Si noti che, se SP $\vdash^{\mathrm{B}}$ è sound e completo rispetto a *Mod*(SP), allora $t \equiv_{C(X)} t'$ e $(t_1, \ldots, t_n) \in (\equiv_{C(X)})_{pr}$ coincidono rispettivamente con la validità di $t = t'$ e di $pr(t_1, \ldots, t_n)$ rispetto ai modelli di SP, $Mod(\mathrm{SP}) \models t = t'$ e $Mod(\mathrm{SP}) \models pr(t_1, \ldots, t_n)$.

Lemma 2.41 Sia SP $= (\Sigma, C)$ una specifica condizionale. Allora $\mathrm{T}_\Sigma(X)/\equiv_{C(X)} \in Mod(\Sigma, C)$.
Dim. Assumiamo $X = \{x_1, \ldots, x_n\}$ ed $\mathrm{F} \in C$; proviamo che $\mathrm{T}_\Sigma(X)/\equiv_{C(X)} \models \mathrm{F}$. Sono possibili tre casi.

$\mathrm{F} = (t = t')$

$\mathrm{T}_\Sigma(X)/\equiv_{C(X)} \models t = t' \Leftrightarrow$ per ogni $v \colon X \to \mathrm{T}_\Sigma(X)/\equiv_{C(X)}$ $t^{\mathrm{T}_\Sigma(X)/\equiv_{C(X)}, v} = t'^{\mathrm{T}_\Sigma(X)/\equiv_{C(X)}, v}$ $\Leftrightarrow$ per ogni v, $t_1 \in v(x_1), \ldots, t_n \in v(x_n)$ $t[t_1/x_1, \ldots, t_n/x_n] \equiv_{C(X)} t'[t_1/x_1, \ldots, t_n/x_n] \Leftrightarrow$ per ogni v, $t_1 \in v(x_1), \ldots, t_n \in v(x_n)$ $C \vdash^{\mathrm{B}} t[t_1/x_1, \ldots, t_n/x_n] = t'[t_1/x_1, \ldots, t_n/x_n]$; e quest'ultima condizione è vera (la prova si ottiene usando B5).

F $= pr(t_1, \ldots, t_n)$

Analogamente al caso precedente

F $= \bigwedge_{j=1}^{m} \alpha_j \supset \alpha$

Sia $v: X \to \mathrm{T}_\Sigma(X)/\equiv_{C(X)}$ e siano $t_1 \in v(x_1), \ldots, t_n \in v(x_n)$.

Dalla regola B5 $C \vdash^{\mathrm{B}} \bigwedge_{j=1}^{m} \alpha_j[t_1/x_1, \ldots, t_n/x_n] \supset \alpha[t_1/x_1, \ldots, t_n/x_n]$.

Sono possibili due casi:

- $C \vdash^{\mathrm{B}} \alpha_j[t_1/x_1, \ldots, t_n/x_n]$ per $j = 1, \ldots, m$;
  allora usando la regola B6, $C \vdash^{\mathrm{B}} \alpha[t_1/x_1, \ldots, t_n/x_n]$;
  allora per definizione $\mathrm{T}_\Sigma(X)/\equiv_{C(X)} \models \alpha[t_1/x_1, \ldots, t_n/x_n]$.
- $C \not\vdash^{\mathrm{B}} \alpha_j[t_1/x_1, \ldots, t_n/x_n]$ per qualche j, $j \in \{1, \ldots, m\}$;
  allora per definizione $\mathrm{T}_\Sigma(X)/\equiv_{C(X)} \not\models \alpha_j[t_1/x_1, \ldots, t_n/x_n]$.

Pertanto $\mathrm{T}_\Sigma(X)/\equiv_{C(X)} \models \bigwedge_{j=1}^{m} \alpha_j[t_1/x_1, \ldots, t_n/x_n] \supset \alpha[t_1/x_1, \ldots, t_n/x_n]$ e quindi $\mathrm{T}_\Sigma(X)/\equiv_{C(X)}, v \models \bigwedge_{j=1}^{m} \alpha_j \supset \alpha$. □

Teorema 2.42 Se SP $= (\Sigma, C)$ è una specifica condizionale, allora SP $\vdash^{\mathrm{B}}$ è completo rispetto agli atomi ed a *Mod*(SP), cioè per ogni $\alpha \in A_\Sigma(X)$

$$(\text{per ogni } \mathrm{M} \in Mod(\mathrm{SP})\ \mathrm{M} \models \alpha) \qquad \Rightarrow \qquad \mathrm{SP} \vdash^{\mathrm{B}} \alpha.$$

Dim. Supponiamo che per ogni M $\in$ *Mod*(SP) M $\models \alpha$.

Dal lemma 2.41 segue che $\mathrm{T}_\Sigma(X)/\equiv_{C(X)} \models \alpha$; ma dalla definizione di $\mathrm{T}_\Sigma(X)/\equiv_{C(X)}$ si ha allora che $\alpha \in \equiv_{C(X)}$ e dunque, per definizione di $\equiv_{C(X)}$, si ottiene SP $\vdash^{\mathrm{B}} \alpha$.□

Si noti che $\vdash^{\mathrm{B}}$ non è completo rispetto agli assiomi condizionali, come si vede dal seguente esempio.

Esempio 2.43 Sia $\Sigma 0$ la segnatura con un'unica sort s e le cui uniche operazioni sono $A, B, C, D\colon \to s$ e sia SP0 la specifica condizionale $(\Sigma 0, \{A = B\})$.

Si ha che per ogni M $\in$ *Mod*(SP0) M $\models C = D \supset A = B$, ma SP0 $\not\vdash^{\mathrm{B}} C = D \supset A = B$
Fine esempio

Dato che il sistema deduttivo $\vdash^{\mathrm{B}}$ è completo solamente rispetto agli atomi si puó anche considerare un altro sistema deduttivo completo le cui formule sono solo atomi, come quello introdotto dalla seguente definizione.

Def. 2.44 Sia SP $= (\Sigma, C)$ una specifica condizionale. Il *sistema condizionale alla Birkhoff* associato a Σ con formule ben formate di tipo atomico, indicato SP $\vdash^{\mathrm{B}}_{\mathrm{A}}$, è definito come SP $\vdash^{\mathrm{B}}$ eccetto che B5 e B6 sono sostituite da:

B5′ $\dfrac{\overline{v}(\alpha_i) i = 1, \ldots, n}{\overline{v}(\alpha)}$ $\alpha_1 \wedge \ldots \wedge \alpha_n \supset \alpha \in C, v\colon X \to \mathrm{T}_\Sigma(X), \overline{v}$ unica estensione omomorfa di v
□

Prop. 2.45 Sia SP $= (\Sigma, C)$ una specifica condizionale tale che $(\mathrm{T}_\Sigma)_s \neq \emptyset$, per ogni $s \in S$. Allora il sistema di Def. 2.44 è sound e completo rispetto agli atomi ed a *Mod*(SP).
Dim. Analoga a quella per $\vdash^{\mathrm{B}}$. □

2.5.2 Modelli iniziali per le specifiche condizionali

Nel caso di specifiche condizionali esiste un modello iniziale, unico a meno di isomorfismi.

Ricordiamo che $\equiv_C$ è $\equiv_{C(\emptyset)}$ e quindi $t_1 \equiv_C t_2$ e $(t_1, \ldots, t_n) \in (\equiv_C)_{pr}$ per $t_1, \ldots, t_n \in \mathrm{T}_\Sigma$, pr predicato di Σ sse $t_1 = t_2$ e $pr(t_1, \ldots, t_n)$ sono deducibili nel sistema alla Birkhoff associato alla specifica (Σ, C); indichiamo questo generico sistema con $C \vdash^{\mathrm{B}}$.

Teorema 2.46 Sia $\mathrm{SP} = (\Sigma, C)$ una specifica condizionale con $(\mathrm{T}_\Sigma)_s \neq \emptyset$, per ogni $s \in S$. Allora

1. $\mathrm{T}_\Sigma/\equiv_{Mod(\mathrm{SP})} = \mathrm{T}_\Sigma/\equiv_C \in Mod(\mathrm{SP})$;

2. $I(\mathrm{SP}) = \{\mathrm{A} \mid \mathrm{A} \cong \mathrm{T}_\Sigma/\equiv_{Mod(\mathrm{SP})}\} = \{\mathrm{A} \mid \mathrm{A} \cong \mathrm{T}_\Sigma/\equiv_C\}$.

Dim.

1 Dal lemma 2.41 ricordando che $\equiv_C$ è $\equiv_{C(\emptyset)}$, si ha che $\mathrm{T}_\Sigma/\equiv_C \in Mod(\mathrm{SP})$ e poiché $C \vdash^{\mathrm{B}}$ è sound e completo rispetto a $Mod(\mathrm{SP})$ ed agli atomi, $\equiv_C$ coincide con $\equiv_{Mod(\mathrm{SP})}$ e dunque l'asserzione. (Si puó anche provare direttamente che $\mathrm{T}_\Sigma/\equiv_{Mod(\mathrm{SP})} \in Mod(\mathrm{SP})$ (per esercizio).)

2 Dal Teor. 2.16 3. $\square$

È interessante notare che le specifiche condizionali sono piú espressive delle specifiche equazionali, cioè esistono algebre che possono essere specificate nell'approccio iniziale per mezzo di una specifica condizionale ma non con una equazionale. Più precisamente, esiste una Σ-algebra A per cui non esiste un insieme finito di equazioni Eq su Σ tale che $I(\Sigma, Eq)$ coincide con la classe delle algebre isomorfe ad A; mentre esiste un insieme finito di equazioni condizionali C su Σ tale che $I(\Sigma, C)$ coincide con la classe delle algebre isomorfe ad A.

Si puó inoltre provare che aggiungendo a Σ alcune operazioni, si puó dare una specifica equazionale sulla nuova segnatura Σ' che ammette un modello iniziale I, tale che la sua restrizione I_Σ a Σ-algebra coincide con A.

3 Un (meta)linguaggio per specifiche strutturate

In questa sezione introduciamo uno schema di (meta)linguaggio per esprimere specifiche algebriche in maniera strutturata e modulare, cioè un linguaggio tale che ogni sua espressione corrisponda ad una specifica algebrica, ovvero la semantica di ogni sua espressione sia una coppia

> $(\Sigma, \mathcal{MOD})$, dove Σ è una segnatura e $\mathcal{MOD}$ è una classe di Σ-algebre chiusa per isomorfismo.

Sia $\mathcal{SPEC}$ la classe di tutte le coppie di tale forma e *SPEC-EXPR* l'insieme di tutte espressioni del linguaggio (espressioni specifica) definite nel seguito; allora la semantica del linguaggio è data da una funzione

$$\mathbf{S}\colon \mathit{SPEC\text{-}EXPR} \to \mathcal{SPEC}.$$

Nel seguito usiamo la seguente notazione: se $\xi = (\Sigma, \mathcal{MOD}) \in \mathcal{SPEC}$, scriviamo $Sig(\xi)$ e $Mod(\xi)$ per Σ e $\mathcal{MOD}$ rispettivamente. Inoltre, unione e contenuto di segnature sono definite componente per componente (e rispetto alle tre componenti: S, OP e PR).

Specifiche semplici Le specifiche semplici (o base) sono i blocchi base per costruire le specifiche, quindi:

($\bullet$) $\mathbf{init}(\Sigma, AX), \mathbf{loose}(\Sigma, AX) \in \mathit{SPEC\text{-}EXPR}$

per tutte le segnature Σ e gli insiemi AX di formule del prim'ordine su Σ;

$\mathbf{S}[\mathbf{loose}(\Sigma, AX)] = (\Sigma, Mod(\Sigma, AX))$;

$\mathbf{S}[\mathbf{init}(\Sigma, AX)] =$

se AX contiene solo assiomi condizionali allora

$(\Sigma, \{\mathrm{A} \mid \mathrm{A}$ è iniziale in $Mod(\Sigma, AX)\})$

altrimenti indefinito.

Usualmente per denotare una specifica semplice SP = **init** (Σ, AX), dove $\Sigma = (S, OP, PR)$, si usa una sintassi infissa scrivendo:

spec SP =
 init
 sorts S
 opns
  OP
 preds
  PR
 axioms
  AX

analogamente per specifiche loose.

Somma di specifiche La somma è il costrutto base per combinare assieme delle specifiche per costruire specifiche piú complesse.

($\bullet$) $\mathrm{SP}_1 + \mathrm{SP}_2 \in \mathit{SPEC}\text{-}\mathit{EXPR}$ per ogni $\mathrm{SP}_1, \mathrm{SP}_2 \in \mathit{SPEC}\text{-}\mathit{EXPR}$;
$\mathbf{S}[\mathrm{SP}_1 + \mathrm{SP}_2] = (\Sigma, \{\mathrm{A} \mid \mathrm{A}$ è una Σ-algebra e $\mathrm{A}_{|\Sigma_i} \in \mathit{Mod}(\mathbf{S}[\mathrm{SP}_i])$ per $i = 1, 2\})$,
dove $\Sigma = \Sigma_1 \cup \Sigma_2$ e $\Sigma_i = \mathit{Sig}(\mathbf{S}[\mathrm{SP}_i])$ per $i = 1, 2$.

Si ricorda che data una Σ-algebra A ed $\Sigma' \subseteq \Sigma$, $\mathrm{A}_{|\Sigma}$ è la Σ' algebra B definita come segue:

- $\mathrm{B}_s = \mathrm{A}$ per ogni sort s di Σ',
- $op^{\mathrm{B}} = op^{\mathrm{A}}$ per ogni operazione op di Σ',
- $pr^{\mathrm{B}} = pr^{\mathrm{A}}$ per ogni predicato pr di Σ'.

In pratica molto spesso è utile un costrutto derivato, **enrich**, definito da:
enrich SP' **by sort** S **opns** OP **preds** PR **axioms** AX $=_{\mathrm{def}}$
$\mathrm{SP}' +$ **sort** $S \cup S'$ **opns** $OP \cup OP'$ **preds** $PR \cup PR'$ **axioms** AX,
dove $\mathit{Sig}(\mathbf{S}[\mathrm{SP}']) = (S', OP', PR')$.

Ridenominazione Questo costrutto è usato per evitare collisioni di nomi quando si combinano assieme delle specifiche e per riutilizzare specifiche definite precedentemente; noi consideriamo solamente ridenominazioni bigettive. L'idea intuitiva è molto semplice: una ridenominazione bigettiva è semplicemente una corrispondenza bigettiva di sort, simboli di operazione e di predicato che "preserva" le arità.

Tutto ció è formalizzato usando il concetto di isomomorfismo di segnatura (vedere Def. 1.2).

($\bullet$) $\rho \circ \mathrm{SP} \in \mathit{SPEC}\text{-}\mathit{EXPR}$ per ogni $\mathrm{SP} \in \mathit{SPEC}\text{-}\mathit{EXPR}$ ed ogni isomorfismo di segnatura ρ;
$\mathbf{S}[\rho \circ \mathrm{SP}] =$
se ρ è un isomorfismo da $\mathit{Sig}(\mathbf{S}[\mathrm{SP}])$ in Σ' allora
$(\Sigma', \{\mathrm{B} \mid \mathrm{B}$ è una Σ'-algebra e $\rho^{-1}(\mathrm{B}) \in \mathit{Mod}(\mathbf{S}[\mathrm{SP}])\})$
altrimenti indefinito,
dove se $\rho\colon \Sigma \to \Sigma'$ è un isomorfismo di segnature e B è una Σ'-algebra, allora $\mathrm{A} = \rho^{-1}(\mathrm{B})$ è la Σ-algebra definita da:

- $\mathrm{A}_s = \mathrm{B}_{\rho(s)}$ per ogni sort s di Σ,
- $op^{\mathrm{A}} = \rho(op)^{\mathrm{B}}$ per ogni operazione op di Σ,
- $pr^{\mathrm{A}} = \rho(pr)^{\mathrm{B}}$ per ogni predicato pr di Σ.

Restrizione a modelli (parzialmente) generati da termini Questo costrutto è usato per restringere i modelli di una specifica a quelli in cui gli elementi dei carrier sono rappresentabili per mezzo di certi termini anche non ground (cioè in cui appaiono variabili).

($\bullet$) SP **reachable on** $\Sigma \in \mathit{SPEC}\text{-}\mathit{EXPR}$ per ogni $\mathrm{SP} \in \mathit{SPEC}\text{-}\mathit{EXPR}$ ed ogni segnatura Σ;
$\mathbf{S}[\mathrm{SP}$ **reachable on** $\Sigma] =$
se $\Sigma \subseteq \mathit{Sig}(\mathbf{S}[\mathrm{SP}]) = \Sigma'$ allora
$(\Sigma', \{\mathrm{A} \mid \mathrm{A} \in \mathit{Mod}(\mathbf{S}[\mathrm{SP}])$ e A è Σ-generata $\})$

altrimenti indefinito.

dove se $\Sigma \subseteq \Sigma'$ e $Rsorts(\Sigma)$ indica il seguente insieme di sort

$$\{\ s \mid s \text{ è una sort di } \Sigma \text{ t.c. esiste } op \text{ operazione di } \Sigma \text{ di sort } s\ \},$$

una Σ'-algebra A è Σ-*generata* se: per ogni $s \in Rsorts(\Sigma)$, per ogni $a \in \mathrm{A}_s$, esiste una famiglia di variabili X t.c. $X_{s'} = \emptyset$ per ogni $s' \in Rsorts(\Sigma)$, un assegnamento di valori alle variabili $v: X \to \mathrm{A}$ e $t \in T_\Sigma(X)_s$ t.c. $t^{v,\mathrm{A}} = a$.

Qualora $\Sigma = Sig(\mathbf{S}[\mathrm{SP}])$, SP **reachable on** Σ è abbreviata in **reachable** SP e corrisponde a considerare tuti e soli i modelli term-generated di SP.

Restrizione della visibilità Questo costrutto è usato per definire quali parti di una specifica (sort e/o operazioni e/o predicati) devono essere "visibili dall'esterno"; alternativamente, definisce quali parti devono essere nascoste (cioè quelle non rese visibili).

($\bullet$) $\mathrm{SP}_{|\Sigma} \in$ *SPEC-EXPR* per ogni SP $\in$ *SPEC-EXPR* ed ogni segnatura Σ

$\mathbf{S}[\mathrm{SP}_{|\Sigma}] =$

se $\Sigma \subseteq Sig(\mathbf{S}[\mathrm{SP}])$, allora $(\Sigma, \{\mathrm{A}_{|\Sigma} \mid \mathrm{A} \in Mod(\mathbf{S}[\mathrm{SP}])\})$

altrimenti indefinito.

3.1 Relazioni tra una specifica strutturata ed una sua parte

Def. 3.1 Una *specifica gerarchica* è una coppia $(\mathrm{SP}, \mathrm{SP}')$, dove $\mathrm{SP} = (\Sigma, AX)$ e $\mathrm{SP}' = (\Sigma', AX')$ sono due specifiche t.c. $\Sigma' \subseteq \Sigma'$ e $AX' \subseteq AX'$; SP$'$ è detta *parte primitiva*.

Sia $t \in \mathrm{T}_\Sigma$; se $t \in \mathrm{T}_{\Sigma'}$, allora si dice *primitivo*; mentre se $t \in (\mathrm{T}_\Sigma)_s$ con s sort di Σ', allora si dice *di sort primitiva*. $\Box$

Esempio 3.2 Specifiche gerarchiche
Si consideri la seguente semplice specifica dei naturali.

spec NAT =
  init
  sorts *nat*
  opns
    $0: \to nat$
    $Succ: nat \to nat$

Ora diamo una prima versione di due specifiche gerarchiche aventi NAT come parte primitiva, precisamente gli insiemi e gli stack.

Gli stack di naturali

spec STACK1 =
  enrich NAT **by**
  sorts *stack*
  opns
    $\Lambda: \to stack$
    $Head: stack \to nat$
    $Tail: stack \to stack$
    $Put: nat \times stack \to stack$
  axioms
    $Head(Put(n, s)) = n$
    $Tail(Put(n, s)) = s$

Gli insiemi di naturali

> **spec** SET1 =
>   **enrich** NAT **by**
>   **sorts** *set*
>   **opns**
> $\emptyset\colon \to set$
> $Sing\colon nat \to set$
> $Union\colon set \times set \to set$
> $Card\colon set \to nat$
>   **axioms**
> $Union(\emptyset, s) = s$
> $Union(s_1, s_2) = Union(s_2, s_1)$
> $Union(s, s) = s$
> $Union(Union(s_1, s_2), s_3) = Union(s_1, Union(s_2, s_3))$
> $Card(\emptyset) = 0$
> $Card(Union(Sing(n), s)) = Succ(Card(s))$

Fine esempio

A questo punto è interessante considerare quali sono le relazioni tra una specifica gerarchica e la sua parte primitiva; cioè il tipo di dato determinato dalla parte primitiva è rimasto inalterato all'interno della specifica gerachica? Per esempio, vediamo cosa succede nelle due specifiche dell'esempio 3.2.

Nella specifica STACK1 non è richiesto che il termine di sort *nat* $Head(\Lambda)$ sia identificato con un qualche termine di sort *nat* della specifica NAT (formalmente per ogni $t \in (\mathrm{T}_{Sig(\mathrm{NAT})})_{nat}$, STACK1 $\not\vdash^{\mathrm{B}}$ $Head(\Lambda) = t$); pertanto restringendo alla segnatura di NAT il modello iniziale di STACK1 si ottiene un'algebra differente dal modello iniziale di NAT. In questo caso il tipo di dato dei naturali è stato modificato all'interno della specifica gerarchica STACK1 aggiungendo nuovi elementi.

Anche la specifica SET1 modifica i naturali; infatti a causa degli assiomi sull'operazione $Card$ si ha che SET1 $\vdash^{\mathrm{B}}$ $Succ(0) = Succ^2(0)$, poiché SET1 $\vdash^{\mathrm{B}}$ $Sing(0) = Union(Sing(0), Sing(0))$ e quindi anche SET1 $\vdash^{\mathrm{B}}$ $Succ^m(0) = Succ^n(0)$ per ogni $n, m > 1$, cioè tutti i naturali positivi sono stati identificati. Anche in questo caso restringendo alla segnatura di NAT il modello iniziale di SET1 si ottiene un'algebra differente dal modello iniziale di NAT.

Def. 3.3 Una specifica gerarchica $\mathrm{HSP} = (\mathrm{SP}, \mathrm{SP}')$ è detta:

– *sufficientemente completa* sse

  per ogni termine t di sort primitiva, esiste t' primitivo t.c. $\mathrm{SP} \vdash t = t'$ per un qualche sistema deduttivo sound e completo;

– *gerarchicamente consistente* sse

  * per ogni t, t' primitivi, $\mathrm{SP} \vdash t = t'$ sse $\mathrm{SP}' \vdash t = t$;
  * per ogni pr predicato di Σ', $t_1, \ldots, t_n$ primitivi,
    $\mathrm{SP} \vdash pr(t_1, \ldots, t_n)$ sse $\mathrm{SP}' \vdash pr(t_1, \ldots, t_n)$. $\square$

STACK1 non è sufficientemente completa rispetto a NAT, mentre SET1 non è gerarchicamente consistente rispetto a NAT.

Prop. 3.4 Se SP, SP′ sono specifiche condizionali e HSP = (SP, SP′) è gerarchicamente consistente e sufficientemente completa, allora
$(\mathrm{I}_{\mathrm{SP}})_{|\Sigma'} = \mathrm{I}_{\mathrm{SP}}$.

Esempio 3.5 Specifiche gerarchiche preservanti le parti primitive
Diamo due nuove versioni delle specifiche degli stack e degli insiemi di naturali che però preservano la parte primitiva dei naturali.

spec STACK =
  enrich NAT **by**
  sorts *stack*
  opns
    $\Lambda\colon \to stack$
    $Head\colon stack \to nat$
    $Tail\colon stack \to stack$
    $Put\colon nat \times stack \to stack$
  axioms
    $Head(Put(n, s)) = n$
    $Tail(Put(n, s)) = s$
    $Head(\Lambda) = 0$
    $Tail(\Lambda) = \Lambda$

spec SET =
  enrich NAT **by**
  sorts *set*
  opns
    $\emptyset\colon \to set$
    $Sing\colon nat \to set$
    $Union\colon set \times set \to set$
    $Card\colon set \to nat$
    $-\colon set \times set \to set$
  preds
    $Diff\colon nat \times nat$
  axioms
    $Diff(0, Succ(n))$
    $Diff(Succ(n), 0)$
    $Diff(n, m) \;\supset\; Diff(Succ(n), Succ(m))$
    $-(s, \emptyset) = s$
    $-(s, Union(s_1, s_2)) = -(-(s, s_1), s_2)$
    $-(Union(s_1, s_2), s) = Union(-(s_1, s), -(s_2, s))$
    $-(Sing(n), Sing(n)) = \emptyset$
    $Diff(n, m) \;\supset\; -(Sing(n), Sing(m)) = Sing(n)$
    $Union(\emptyset, s) = s$
    $Union(s_1, s_2) = Union(s_2, s_1)$
    $Union(s, s) = s$
    $Union(Union(s_1, s_2), s_3) = Union(s_1, Union(s_2, s_3))$
    $Card(\emptyset) = 0$
    $Card(Union(Sing(n), s)) = Succ(Card(-(s, Sing(n))))$

STACK e SET sono sufficientemente complete e gerarchicamente consitenti rispetto a NAT.
Fine esempio

4 Implementazione di specifiche

Siano SP e SP′ due specifiche; quando SP è implementata da SP′? Ci sono almeno due punti di vista da considerare:

- implementare significa raffinare, quindi SP′ deve essere un raffinamento di SP, cioè "cose" non precisate in SP sono rese precise in SP′ aggiungendo ulteriori requisiti;
- implementare significa realizzare i dati, le operazioni ed i predicati specificati astrattamente in SP, usando i dati, le operazioni ed i predicati di SP′.

Formalmente, si ha che SP è *implementata da* SP′ *rispetto ad* α, una funzione da specifiche in specifiche, (scritto SP $\rightsquigarrow^{\alpha}$SP′) sse $Mod(\alpha(\mathrm{SP}')) \subseteq Mod(\mathrm{SP})$.

La funzione α descrive come le parti di SP sono realizzate in SP′ (implementazione come realizzazione); mentre implementazione come raffinamento è data dalla richiesta di inclusione tra le classi dei modelli.

Chiaramente non tutte le funzioni tra specifiche sono accettabili; per esempio se α aggiunge tutti i predicati e le operazioni di SP a SP′ abbiamo una implementazione banale. Tuttavia, tale definition include come casi particolari le varie definizioni proposte fino ad ora. Usualmente α è una combinazione delle varie operazioni per strutturare specifiche viste precedentemente. Quando α è una composizione di una ridenominazione, un arrichimento con operazioni e predicati derivati, una restrizione della visibilità ed un arrichimento con assiomi, si ha la cosidetta implementazione per ridenominazione-arrichimento-restrizione-identificazione, che è il corrispondente per le specifiche algebriche della implementazione di tipi di dato concreti di Hoare.

Avendo ora una definizione di implementazione, è interessante studiare le sue relazioni con la struttura delle specifiche. Ci limitiamo a considerare implementazioni dove α è l'identità, usando per questo caso la notazione SP $\rightsquigarrow$ SP′ (notare che quindi SP e SP′ devono avere la stessa segnatura). Questa non è una restrizione: dalle proprietà di $\rightsquigarrow$ si possono ottenere proprietà di $\rightsquigarrow^{\alpha}$, poiché α è definita come una combinazione di operazioni per strutturare specifiche; infatti, SP $\rightsquigarrow^{\alpha}$SP′ sse SP $\rightsquigarrow \alpha(\mathrm{SP}')$.

Fatto 4.1

1. $\rightsquigarrow$ è un ordine parziale (cioè è riflesivo, antisimmetrico e transitivo).
2. Tutte le operazioni tra specifiche di Sez. 3 sono monotone rispetto a $\rightsquigarrow$: se SP, SP′, SP″ sono specifiche, Σ è contenuta in $Sig(\mathrm{SP})$, ρ è un isomorfismo di segnatura e SP $\rightsquigarrow$ SP′, allora

   $\mathrm{SP} + \mathrm{SP}'' \rightsquigarrow \mathrm{SP}' + \mathrm{SP}''$;

   $\mathrm{SP}_{|\Sigma} \rightsquigarrow \mathrm{SP}'_{|\Sigma}$;

   SP **reachable on** $\Sigma \rightsquigarrow$ SP′ **reachable on** Σ;

   $\rho \cdot \mathrm{SP} \rightsquigarrow \rho \cdot \mathrm{SP}'$. □

4.1 Sviluppo di specifiche di tipi di dato

Come abbiamo visto nella Sez. **??** i due tipi di approccio alle specifiche, quello iniziale e quello loose, differiscono in quanto il primo, l'iniziale, individua un solo tipo di dato astratto, in cui valgono solo le proprietà date negli assiomi e quelle che ne seguono logicamente, mentre il secondo, il loose, individua diversi tipi di dato astratto, in cui valgono le proprietà date negli

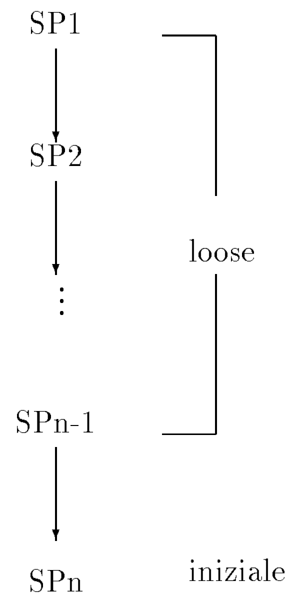

Figure 1: Sviluppo top-down di una specifica

assiomi e quelle che ne seguono logicamente e l'interpretazione di ogni altra formula logica è vera o falsa a seconda del tipo di dato.

Questo suggerisce che i due approcci possano essere utilizzati per specificare un tipo di dato a diversi livelli di astrazione:

- grazie all'approccio loose, inizialmente si specificano solo proprietà generali, senza indagare sulla struttura del tipo di dato, poi si richiede che ulteriori proprietà siano soddisfatte dal tipo di dato, aggiungendo via via ulteriori requisiti che fissano l'interpretazione di un numero sempre maggiore di formule e/o specificano la struttura del tipo di dato in modo sempre piú dettagliato;

- grazie all'approccio iniziale, quando si arriva a specificare tutte le proprietà che si vuole siano soddisfatte dal tipo di dato astratto da rappresentare, si ha la sicurezza che tutto ciò che non segue logicamente dagli assiomi sia falso nei modelli della specifica, ossia si individua non piú una classe di tipi di dato astratti ma un solo tipo di dato astratto, quello voluto.

Questo sviluppo della specifica di un tipo di dato astratto a diversi livelli di astrazione, ottenuta con vari passi di raffinamento fornisce una sequenza di specifiche loose, $\mathrm{SP}_1, \ldots, \mathrm{SP}_{n-1}$ ed una specifica iniziale SP_n che rappresenta il tipo di dato (vedi Figura 1). Chiaramente la specifica data al passo i-esimo, SP_i, deve raffinare, cioè estendere aggiungendo nuovi requisiti, quella data al livello precedente, ovviamente oltre che realizzare tutti i requisiti di SP_i; cioè SP_i deve implementare SP_{i-1}. Formalmente, SP_i *implementa* SP_{i-1} se tutti i modelli di $\alpha(\mathrm{SP}_i)$ sono anche modelli di SP_{i-1}, dove α è una funzione tra specifiche che esprime come realizzare i costituenti di SP_{i-1} con i costituenti di SP_i e come togliere tutto ciò che si è aggiunto in SP_i; in particolare α stabilisce la corrispondenza tra i nomi delle componenti delle due specifiche (come sono cambiati i nomi di SP_{i-1} nella sua implementazione SP_i).

Esempio 4.2 Sviluppo top-down della specifica degli interi
Specifichiamo gli interi in modo top-down, aggiungendo via, via nuovi requisiti, fino ad ottenere la specifica con approccio iniziale che rappresenta gli interi con l'operazione somma.

Si vuole rappresentare il tipo di dato astratto semigruppo caratterizzato dal fatto che sugli elementi del tipo di dato è definita un'operazione binaria e associativa, ossia un'operazione che presi due elementi restituisce un elemento del tipo di dato e tale che il risultato di varie applicazioni dell'operazione agli argomenti sia indipendente dall'ordine in cui vengono effettuate. Visto che la proprietà richiesta al tipo di dato astratto semigruppo è molto generale e non interessa la struttura degli elementi del tipo di dato l'approccio alle specifiche algebriche adeguato è quello loose.

Consideriamo la seguente specifica:

spec INT_1 =
  loose
  sorts s
  opns
    $@: s \times s \to s$
  axioms
    $@(@(x,y),z) = @(x,@(y,z))$

Con l'approccio loose la classe dei tipi di dato astratti rappresentati dalla specifica è costituita dall'insieme dei tipi di dato astratto, anche tra loro non isomorfi, che soddisfino almeno la proprietà data dall'assioma.

La specifica precedente richiede che vi sia un carrier nell'algebra e che sugli elementi del carrier sia definita un'operazione che presi due elementi restituisca ancora un elemento del carrier e che sia associativa. In conclusione tutte le algebre con un carrier ed un'operazione binaria sugli elementi del carrier che sia associativa, cioè soddisfi l'assioma, è, modulo isomorfismo, un tipo di dato astratto appartenente alla classe, qualunque siano le altre proprietà soddisfatte dall'algebra; quindi questa specifica rappresenta la classe di tutti i semigruppi.

Vediamo due algebre che sono, modulo isomorfismo, due tipi di dato astratti appartenenti alla classe; la prima algebra A è definita da:
Carrier $s^{\mathrm{A}} = \mathbb{Z}$
Interpretazione dei nomi di operazione
$@^{\mathrm{A}}: \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ $\quad @^{\mathrm{A}}(x,y) = x + y$;
la seconda algebra B è definita da:
Carrier $s^{\mathrm{B}} = \{0,1\}$
Interpretazione dei nomi di operazione
$@^{\mathrm{B}}: \{0,1\} \times \{0,1\} \to \{0,1\}$
$@^{\mathrm{B}}(0,0) = 0, \quad @^{\mathrm{B}}(0,1) = 1, \quad @^{\mathrm{B}}(1,0) = 1, \quad @^{\mathrm{B}}(1,1) = 0.$

Le algebre A, B sono, modulo isomorfismo, due tipi di dato astratto della classe, ma non sono lo stesso tipo di dato astratto perchè non sono isomorfe tra loro.

Notiamo che se ricorriamo all'approccio di tipo iniziale per individuare il tipo di dato astratto rappresentato dalla specifica precedente, ci accorgiamo che è l'algebra vuota. Infatti il carrier associato alla sort s è l'insieme vuoto, in quanto ogni oggetto del carrier deve essere rappresentabile da un termine, ma l'insieme dei termini chiusi, cioè senza variabili, sulla segnatura è vuoto, perché non vi sono termini costanti nè tantomeno termini costruiti su di essi. Se invece ricorriamo all'approccio di tipo loose non abbiamo la condizione di rappresentabilità degli oggetti dei carrier delle algebre considerate, quindi le considerazioni precedenti che ci hanno portato alla conclusione che il tipo di dato astratto individuato dalla specifica è l'algebra vuota non sono piú pertinenti.

Definiamo una specifica INT_2, sempre con approccio loose, che implementa INT_1, ma restringe l'insieme dei modelli perché vogliamo individuare non piú le algebre che sono solo dei semigruppi, ma vogliamo anche che tale funzione abbia l'elemento neutro, ossia un elemento tale che la funzione applicata ad esso ed a qualsiasi altro elemento dia come risultato l'altro elemento.

Quindi INT_2 è definita arricchendo INT_1 con una costante "U" che rappresenta l'elemento neutro rispetto all'operazione binaria.

spec $\mathrm{INT}_2 =$
  loose
  sorts s
  opns
    $@: s \times s \to s$
    $U: \to s$
  axioms
    $@(@(x, y), z) = @(x, @(y, z))$
    $@(x, U) = x$
    $@(U, x) = x$

Le algebre A e B date precedentemente si possono estendere a due modelli di INT_2, rispettivamente A' e B'.

Consideriamo infatti l'algebra A' con lo stesso carrier e la stessa interpretazione del nome di operazione di A e con in piú l'interpretazione del nome di costante $U^{\mathrm{A}'} = 0$.

Analogamente l'algebra B' ha lo stesso carrier e la stessa interpretazione del nome di operazione di B ed in piú l'interpretazione del nome di costante: $U^{\mathrm{B}'} = 0$.

Le algebre A', B' sono, modulo isomorfismo, due tipi di dato astratto della classe monoide (semigruppo con elemento neutro), ma non sono lo stesso tipo di dato astratto perché non sono isomorfe tra loro.

Notiamo che in questo caso la funzione tra specifiche α_1, che serve per dimostrare che INT_2 implementa INT_1, è tale che nasconde la costante "U" aggiunta ad INT_1.

Raffiniamo ancora INT_2 perché vogliamo individuare non piú le algebre che sono solo dei monoidi, ossia in cui l'interpretazione del nome di operazione binaria sia una funzione associativa, con l'elemento neutro, ma vogliamo anche che tale funzione abbia l'inverso, ossia per ogni elemento del carrier vogliamo che esista un elemento tale che la funzione applicata ad essi dia come risultato l'elemento neutro. In conclusione vogliamo rappresentare la classe dei tipi di dato gruppo (monoide con inverso).

Definiamo quindi la specifica INT_3, sempre con approccio loose, che implementa INT_2, ottenuta aggiungendo un nome di operazione unaria, Inv, la cui interpretazione è una funzione che preso un elemento restituisce il suo inverso. Otteniamo quindi la specifica seguente:

spec $\mathrm{INT}_3 =$
  loose
  sorts s
  opns
    $@: s \times s \to s$
    $U: \to s$
    $Inv: s \to s$
  axioms
    $@(@(x, y), z) = @(x, @(y, z))$
    $@(x, U) = x$
    $@(U, x) = x$
    $@(x, Inv(x)) = U$
    $@(Inv(x), x) = U$

Consideriamo l'algebra A'' con lo stesso carrier e le stesse interpretazioni di nomi di costante e di operazione di A' ed anche l'interpretazione del nome di operazione Inv seguente:

$Inv^{A''}: \mathbb{Z} \to \mathbb{Z}$
$Inv^{A''}(x) = -x;$
e l'algebra B″ analoga a B′ con in piú l'interpretazione del nome di operazione Inv seguente:
$Inv^{B''}: \{0,1\} \to \{0,1\}$
$Inv^{B''}(0) = 0, \qquad Inv^{B''}(1) = 1$

Le algebre A″, B″ sono, modulo isomorfismo, due tipi di dato astratto della classe gruppo (monoide con inverso), ma non sono lo stesso tipo di dato astratto perché non sono isomorfe tra loro.

Notiamo che in questo caso la funzione tra specifiche α_2, che serve per dimostrare che INT_3 implementa INT_2, è tale che nasconde l'operazione unaria "Inv" aggiunta ad INT_2.

Fino ad ora si è fatto ricorso all'approccio loose alle specifiche, perché interessati a rappresentare tutti i tipi di dato astratto che siano gruppi rispetto ad un'operazione binaria; questi requisiti generali non specificano ad esempio come devono essere costruiti gli elementi dell'insieme.

Se si vuole arrivare ad individuare un unico tipo di dato astratto definito dalla specifica, ad esempio il gruppo additivo $\mathbb{Z}$, bisogna specificare anche quali sono i costruttori degli elementi, quali condizioni sono soddisfatte da questi, come sono definite le operazioni ed inoltre utilizzare una specifica con semantica iniziale.

Definiamo quindi la specifica INT_4, con semantica iniziale, che implementa INT_3, ottenuta aggiungendo i costruttori degli elementi *Pred* e *Succ*, entrambi nomi di operazione unari e le formule che ne esprimono le proprietà e relazioni con gli altri nomi di operazione. Otteniamo quindi la specifica seguente:

spec INT_4 =
  init
  sorts *int*
  opns
    $+: int \times int \to int$
    $0: \to int$
    $Pred, Succ, -: int \to int$
  axioms
    $Succ(Pred(x)) = x$
    $Pred(Succ(x)) = x$
    $+(x, 0) = x$
    $+(x, Succ(y)) = Succ(+(x, y))$
    $+(x, Pred(y)) = Pred(+(x, y))$ axiom-(0) = 0
    $-(Succ(x)) = Pred(-(x))$
    $-(Pred(x)) = Succ(-(x))$

Il tipo di dato astratto rappresentato dalla specifica è la classe di isomorfismo rappresentata da $(\mathbb{Z}, +, 0, -)$ come voluto.

Notiamo che in questo caso la funzione tra specifiche α_3, che serve per dimostrare che INT_4 implementa INT_3, cambia la sort *int* in s e le operazioni $+$, 0 e $-$ rispettivamante in @, U ed In.

Si noti che in questa specifica non sono dati esplicitamente gli assiomi di INT_3, poiché essi seguono logicamente da quelli di INT_4. **Fine esempio**

Contents