ΕΦΑΡΜΟΓΕΣ ΤΗΣ ΛΟΓΙΚΗΣ

ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ

ΣΗΜΕΙΩΣΕΙΣ ΠΑΡΑΔΟΣΕΩΝ

ΠΕΤΡΟΣ ΣΤΕΦΑΝΕΑΣ

ΕΙΣΑΓΩΓΗ

Η λογική με τη συμβολική ή μη μορφή της έχει διεισδύσει σε όλες τις πτυχές της πληροφορικής. Η ανάπτυξη της συμβολικής λογικής και ειδικά της θεωρίας αλγορίθμων μαζί με την εξέλιξη των ηλεκτρονικών και της κυβερνητικής αποτέλεσαν άλλωστε το κατάλληλο επιστημονικό και τεχνολογικό περιβάλλον για τη γέννηση της πληροφορικής. Δεν είναι τυχαίο εξάλλου ότι ορισμένοι από τους πρωτεργάτες της πληροφορικής ήταν ήδη διακεκριμένοι ερευνητές της λογικής (Turing, Von Neumann, Ulam).

Εξαιτίας της παραπάνω ιστορική σχέσης ανάμεσα τα δύο αντικείμενα, οι έννοιες, η μεθοδολογία και οι ερευνητικές κατευθύνσεις της (θεωρητικής) πληροφορικής είναι φανερά επηρεασμένες απ' αυτές της λογικής. Αλλά και αντίστροφα, οι ανάγκες της πληροφορικής τροφοδοτούν συνεχώς με προβλήματα την εφαρμοσμένη λογική που πολλές φορές τα αντιμετωπίζει με τη δημιουργία νέων θεωριών. Ολόκληροι κλάδοι της πληροφορικής όπως η σημασιολογία των προγραμμάτων (semantics of programs), οι τυπικές μέθοδοι (formal methods), η αποδεικτική θεωρημάτων (theorem proving), ο εξισωτικός (equational) και ο λογικός προγραμματισμός (logic programming) αναπτύχθηκαν λόγω αυτής της αμφίδρομης σχέσης ανάμεσα στα δύο γνωστικά αντικείμενα. Άλλοι κλάδοι της πληροφορικής όπως οι βάσεις δεδομένων (data bases), οι γλώσσες προδιαγραφών (specification languages), η τεχνολογία λογισμικού (software engineering) και οι ταυτοχρονικές διαδικασίες (concurrent processes) χρησιμοποιούν συστηματικά τη συμβολική λογική. Επίσης πολλά στοιχεία (μη συμβολικής) λογικής μεθοδολογίας διακρίνονται σε διάφορες κλασικές φάσεις ανάπτυξης ενός πληροφοριακού συστήματος όπως εμφανίζονται για παράδειγμα στο μοντέλο του καταρράκτη (waterfall model). Τέτοιες φάσεις είναι η ανάλυση απαιτήσεων (requirements analysis), η ανάπτυξη προδιαγραφών (specifications) και ο σχεδιασμός (design). Πολλές προσπάθειες έχουν γίνει για το συγκερασμό της συμβολικής με τη μη – συμβολική λογική μεθοδολογία σε σχέση με ενδεχόμενες εφαρμογές τους στα πληροφοριακά συστήματα αλλά και τα συστήματα γενικά. Σαν τέτοια μπορεί να αναφέρει κανείς τα ιβριδικά συστήματα (hybrid systems).

Ιδιαίτερο λόγο στη σχέση πληροφορικής λογικής παίζει η άλγεβρα. Η θεωρία πεδίων (domain theory), η θεωρία κατηγοριών (category theory), οι αλγεβρικές προδιαγραφές και οι αλγεβρικές θεωρίες του Lawvere (Lawvere theories), ο σχεσιακός λογισμός (relational calculus) και οι άλγεβρες Boole (Boolean algebras) αποτελούν χαρακτηριστικά παραδείγματα εφαρμοσμένων αλγεβρικών θεωριών που σχετίζονται άμεσα με τη λογική και έχουν βρει σημαντικές εφαρμογές στην πληροφορική. Ας σημειωθεί ότι είναι ιδιαίτερα δύσκολο να χαραχθεί διαχωριστική γραμμή ανάμεσα σ’ αυτού του είδους την άλγεβρα και την τυπική λογική.

Η λογική βρίσκει εφαρμόζεται στην πληροφορική μέσω συγκεκριμένων *λογικών συστημάτων*. Τόσο η χρήση λογικών συστημάτων όσο και ο αριθμός εκείνων που αξιοποιούνται από την πληροφορική αναπτύσσεται ταχύτατα. Τέτοια λογικά συστήματα είναι για παράδειγμα:

- η εξισωτική λογική (equational logic),
- η λογική των προτάσεων του Horn (Horn – clause logic),
- η πρωτοβάθμια λογική (first order logic),
- η δευτεροβάθμια λογική (second order logic),
- η χρονική λογική (temporal logic),
- η τροπιή (modal) και
- η δυναμική (dynamic) λογική,
- η ιντουσιονιστική θεωρία τύπων υψηλής τάξεως (intuitionistic higher order type theory),
- η γραμμική λογική (linear logci),
- η εξισωτική λογική με διατεταγμένους τύπους (order sorted equational logic),
- η εξισωτική λογική με κρυμμένους τύπους (hidden sorted equational logic),
- η κατηγοριακή εξισωτική λογική (categorical equational logic),
- η επιστημονική λογική (epistemic logic),
- η δεοντική λογική (deontic logic), κ.τ.λ.

Νέα λογικά συστήματα ορίζονται συνεχώς για τις ανάγκες της πληροφορικής όπως η διαγραμματική λογική (diagram logic) και η λογική της αναγραφής (rewriting logic). Η διαγραμματική λογική είναι η βάση της σημασιολογίας του εκπαιδευτικού περιβάλλοντος λογισμικού Hyperproof το οποίο έχει αναπτυχθεί για τη διδασκαλία της θεωρίας αποδείξεων. Η λογική της αναγραφής έχει γνωρίσει ιδιαίτερη ανάπτυξη το τελευταίο διάστημα σαν ένα από τα σημαντικότερα σημασιολογικά μοντέλα των παράλληλων διαδικασιών.

ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΚΑΤΗΓΟΡΙΚΗ ΣΗΜΑΣΙΟΛΟΓΙΑ

1. ΟΡΙΣΜΟΙ – ΒΑΣΙΚΕΣ ΚΑΤΑΣΚΕΥΕΣ

Μία **κατηγορία** C αποτελείται από:

1. μία συλλογή από **αντικείμενα**
2. μία συλλογή από **μορφισμούς**
3. τελεστές που επισυνάπτουν σε κάθε μορφισμό f ένα αντικείμενο dom f (domain = πεδίο) και ένα αντικείμενο cod f (codomain = συμπεδίο).

   Γράφουμε $f{:}A \to B$ ή $A \xrightarrow{f} B$ τους μορφισμούς f με dom f = A και cod f = B.

   Τη συλλογή των μορφισμών από το A στο B τη συμβολίζουμε με C(A,B).
4. ένα τελεστή σύνθεσης που επισυνάπτει σε κάθε ζεύγος μορφισμών f και g με cod f = dom g, τη **σύνθεση** των μορφισμών $g \circ f : \mathrm{dom} f \to \mathrm{cod} g$, έτσι ώστε να ικανοποιεί την ακόλουθη **προσεταιριστική ιδιότητα:** για κάθε δύο μορφισμούς $f{:}A \to B$, $g{:}B \to C$ και $h{:}C \to D$ να ισχύει ότι:

$$h \circ (g \circ f) = (h \circ g) f$$

5. για κάθε αντικείμενο A, να υπάρχει ένας **ταυτοτικός μορφισμός** $id_A : A \to A$ που να ικανοποιεί τον ακόλουθο **ταυτοτικό νόμο**: για κάθε μορφισμό $f{:}A \to B$,

$$id_B \circ f = f \quad \text{και} \quad f \circ id_A = f \,.$$

Παράδειγμα 1. Η κατηγορία **Set** (των συνόλων) έχει ως **αντικείμενα σύνολα** και ως **μορφισμούς συναρτήσεις μεταξύ συνόλων**.

Η σύνθεση των μορφισμών είναι η συνολοθεωρητική σύνθεση συναρτήσεων.

Οι ταυτοτικοί μορφισμοί είναι οι ταυτοτικές συναρτήσεις.

Παράδειγμα 2. Η κατηγορία των μερικά διατεταγμένων συνόλων **PoSet.**

Μια **μερική διάταξη** $\leq_p$ σε ένα σύνολο P είναι μια σχέση με τις εξής ιδιότητες:

(1) $p \leq p$

(2) $p \leq p' \leq p'' \Rightarrow p \leq p''$

(3) $p \leq p'$ και $p' \leq p \Rightarrow p = p'$

για κάθε $p, p', p'' \in P$.

Μια συνάρτηση $f : P \to Q$ από το $(P, \leq_P)$ στο $(Q, \leq_Q)$ ονομάζεται **συνάρτηση που διατηρεί τη μερική διάταξη** εάν ισχύει:

Εάν $p \leq_p p'$ τότε $f(p) \leq_Q f(p')$.

Η **PoSet** ως **κατηγορία** έχει:

αντικείμενα: μερικά διατεταγμένα σύνολα

μορφισμούς: συναρτήσεις που διατηρούν τη μερική διάταξη (αύξουσες συναρτήσεις).

Ένας **μορφισμός** $f:(P, \leq_P) \to (Q, \leq_Q)$ στην $P \circ \mathrm{Set}$ είναι μια συνάρτηση από το P στο Q που διατηρεί τη διάταξη στο P, έτσι ώστε εάν $p \leq_P p'$ τότε $f(p) \leq_Q f(p')$

$$\mathrm{dom}\, f = (P, \leq_P)$$

$$\mathrm{cod}\, f = (Q, \leq_Q)$$

και

$$f \in \mathrm{PoSet}((P, \leq_P),(Q \leq_Q))$$

Επίσης, για κάθε μερική διάταξη $(P, \leq_P)$ η ταυτοτική συνάρτηση id_p διατηρεί τη διάταξη στο P και ικανοποιεί τον ταυτοτικό νόμο.

Η σύνθεση δύο συναρτήσεων που διατηρούν τη μερική διάταξη είναι μια συνάρτηση που διατηρεί τη μερική διάταξη.

Έτσι αν έχουμε τις εξής συναρτήσεις που διατηρούν τη μερική διάταξη:

$$g : P \to Q$$

$$g : Q \to R$$

ισχύει ότι εάν $p \leq_P p'$, τότε αφού η f διατηρεί τη μερική διάταξη ισχύει ότι:

$$f(p) \leq_Q f(p')$$

και επειδή η g διατηρεί τη μερική διάταξη ισχύει ότι:

$$g(f(p)) \leq_R g(f(p'))$$

Άρα η σύνθεση $f \circ g$ διατηρεί τη μερική διάταξη.

Παράδειγμα 3. Η κατηγορία των μονοειδών (monoids) **Mon** με **αντικείμενα** τα **μονοειδή** και **μορφισμούς** τους **ομομορφισμούς μονοειδών**.

Ορισμός. Ένα **μονοειδές** (monoid) αποτελείται από ένα σύνολο M, μια διμελή πράξη $\cdot$ και ένα ουδέτερο στοιχείο e. – Το συμβολίζουμε ως (M, $\cdot$, e) – ώστε να ισχύει

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z \quad \text{για κάθε } x,y,z \in M$$

και

$$e \cdot x = x = x \cdot e \quad \text{για κάθε } x \in M.$$

Ένας **ομομορφισμός μονοειδών** από το (M, $\cdot$, e) στο (M′, $\cdot'$, e′) είναι μια συνάρτηση $f : M \to M'$ έτσι ώστε $f(e) = e'$ και $f(x \cdot y) = f(x) \cdot f(y)$.

Παράδειγμα 4 (Κατηγορική Λογική).

Ας θεωρήσουμε ότι σε μια κατηγορία ***τα αντικείμενα αντιστοιχούν σε προτάσεις*** και οι ***μορφισμοί σε αποδείξεις***.

Ένας μορφισμός $f : A \to B$ είναι στην ουσία η «απόδειξη» του B από το A.

Επίσης εάν $f : A \to B$ και $g : B \to C$

Τότε ισχύσει ότι $g \circ f : A \to C$.

Παράδειγμα 5. Ας θεωρήσουμε μια απλή συναρτησιακή γλώσσα προγραμματισμού με τους εξής **τύπους**:

Int	integers (ακέραιοι)
Real	reals (πραγματικοί)
Bool	truth values (αληθοτιμές)
Unit	a – one – ellement type (μονοσύνολο)

και τους εξής τελεστές:

is zero:	Int $\to$ Bool	έλεγχος εάν κάτι είναι μηδέν
not:	Bool $\to$ Bool	άρνηση
$succ_{Int}$:	Int $\to$ Int	επόμενος ακέραιος

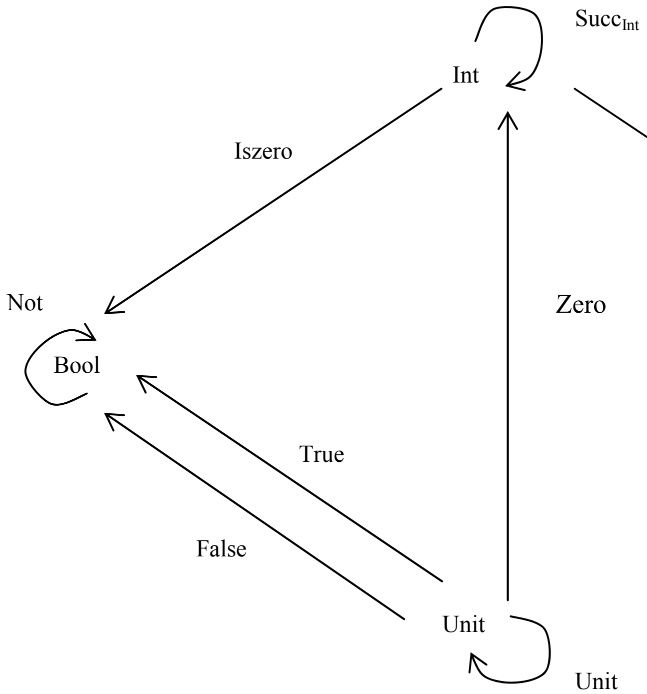
$succ_{Real}$:	Real → Real	"επόμενος" πραγματικός
to Real	Int → Real	μετατροπέας των ακεραίων σε πραγματικούς

και σταθερές:

zero:	Int
true:	Bool
false:	Bool
unit:	Unit

Τα παραπάνω σχηματίζουν μια κατηγορία FPL ως εξής:

1. Int, Real, Bool και Unit είναι **αντικείμενα**
2. Iszero, not, $succ_{Int}$, $succ_{Real}$ και toReal είναι **μορφισμοί**
3. Οι σταθερές zero, true, false και unit είναι μορφισμοί από το αντικείμενο Unit στα Int, Bool, Bool, Unit αντίστοιχα, οι οποίοι αντιστοιχούν το μοναδικό στοιχείο του Unit στα αντίστοιχα στοιχεία αυτών των τύπων.



Κατηγορία	Αντικείμενα	Μορφισμοί
Set	Sets – σύνολα	Συναρτήσεις
Pfn	Σύνολα	Μερικές συναρτήσεις
FinSet	Πεπερασμένα σύνολα	Πεπερασμένες συναρτήσεις
Mon	Μονοειδή	Ομομορφισμοί μονοειδών
P ∘ Set	Μερικά διατεταγμένα σύνολα	Μονότονες συναρτήσεις
Grp	Ομάδες	Ομομορφισμοί ομάδων
Ω – Alg	Ω – άλγεβρες	Ω – ομομορφισμοί
CPO	Πλήρεις μερικές διατάξεις	Συνεχείς συναρτήσεις
Vect	Διανυσματικοί χώροι	Γραμμικοί μετασχηματισμοί
Top	Τοπολογικοί χώροι	Συνεχείς συναρτήσεις

Άσκηση 1. Μια ομάδα $(G, \cdot, ^{-1}, e)$ είναι ένα σύνολο G μαζί με ένα δυαδικό τελεστή $\cdot$, ένα μοναδιαίο τελεστή – 1, και ένα διακεκριμένο στοιχείο e έτσι ώστε:

(α) $(x \cdot y) \cdot z = x \cdot (y \cdot z) \ \forall x,y,z \in G$

(β) $e \cdot x = x = x \cdot e \ \forall x \in G$

(γ) $x \cdot x^{-1} = e = x^{-1} \cdot x \ \forall x \in G$.

Δείξτε ότι μια ομάδα μπορεί να θεωρηθεί ως κατηγορία.

2. ΔΙΑΓΡΑΜΜΑΤΑ

Γραφική Αναπαράσταση των Ιδιοτήτων των Κατηγοριών

Ορισμός. Ένα διάγραμμα σε μια κατηγορία C είναι μια συλλογή από «ακμές» και «κόμβους» που αντιστοιχούν σε μορφισμούς και αντικείμενα της C με συνεπή τρόπο. Αυτό σημαίνει ότι εάν μια «ακμή» σε ένα διάγραμμα αντιστοιχεί σε ένα μορφισμό f και ότι αυτός έχει τη μορφή $f : A \to B$ τότε οι κόμβοι αυτής της ακμής είναι το A και το B.

- Τα διαγράμματα χρησιμοποιούνται για να τεθούν και να αποδειχθούν ιδιότητες κατηγορικών κατασκευών.

Ορισμός. Ένα διάγραμμα σε μια κατηγορία C ονομάζεται μεταθετικό εάν για κάθε ζεύγος από κόμβους X και Y, όλα τα «μονοπάτια» στο διάγραμμα από το X στο Y είναι ίσα, με την έννοια ότι κάθε μονοπάτι στο διάγραμμα καθορίζει ένα μορφισμό και αυτοί οι μορφισμοί είναι ίσοι στη C.

Για παράδειγμα, λέγοντας ότι το διάγραμμα είναι μεταθετικό

$$\begin{tikzcd} X \arrow[r, "f'"] \arrow[d, "g'"'] & Z \arrow[d, "g"] \\ W \arrow[r, "f"'] & Y \end{tikzcd}$$

f

Z

σημαίνει ότι $f \circ g' = g \circ f'$.

Η μεταθετικότητα του διαγράμματος

$$X \underset{g}{\rightrightarrows} Y \xrightarrow[h]{\quad}$$

σημαίνει ότι: $h \circ f = h \circ g$ (αυτό δεν σημαίνει πως $f = g$).

Παράδειγμα. Στην κατηγορία $\mathrm{Set}^{\to}$ τα αντικείμενα είναι οι Set – μορφισμοί $f{:}A \to B$. Ένας $\mathrm{Set}^{\to}$ – μορφισμός από τον $f{:}A \to B$ στον $f'{:}A' \to B'$ είναι ένα ζεύγος (a, b) από Set – μορφισμούς ώστε το εξής διάγραμμα να είναι μεταθετικό στην Set (κατηγορία των συνόλων).

$$\begin{tikzcd}
A \arrow[r, "a"] \arrow[d, "f"'] & A' \arrow[d, "f'"] \\
B \arrow[r, "b"'] & B'
\end{tikzcd}$$

Η σύνθεση των $\mathrm{Set}^{\to}$ – μορφισμών

$$(a,b) : (f : A \to B) \to (f' : A' \to B')$$

$$(a',b') : (f' : A' \to B') \to (f'' : A'' \to B'')$$

και

$$(a',b') \circ (a,b) = (a' \circ a,\ b' \circ b)$$

αντιστοιχεί στο εξής διάγραμμα

$$\begin{array}{ccccc}
 & \alpha & & \alpha' & \\
A & & A' & & A'' \\
f & & f' & & f'' \\
 & b & & b' & \\
B & & B' & & B''
\end{array}$$

το οποίο θα πρέπει να είναι μεταθετικό.

Πρόταση. Εάν και τα δύο εσωτερικά τετράγωνα είναι μεταθετικά, το αυτό συμβαίνει και με το «εξωτερικό παραλληλόγραμμο»

$$
\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "a"'] & B \arrow[r, "f'"] \arrow[d, "b"] & C \arrow[d, "c"] \\
A' \arrow[r, "g"'] & B' \arrow[r, "g'"'] & C'
\end{tikzcd}
$$

Απόδειξη:

$$
\begin{aligned}
(g' \circ g) \circ a &= g' \circ (g \circ a) \text{ προσεταιριστική ιδιότητα} \\
&= g' \circ (b \circ f) \text{ μεταθετικότητα } 1^{ου} \text{ τετραγώνου} \\
&= (g' \circ b) \circ f \text{ προσεταιρισκή ιδιότητα} \\
&= (c \circ f') \circ f \text{ μεταθετικότητα } 2^{ου} \text{ τετραγώνου} \\
&= c \circ (f' \circ f) \text{ προσεταιριστική ιδιότητα.}
\end{aligned}
$$

Παράδειγμα. Στην κατηγορία FPL το παρακάτω διάγραμμα είναι μεταθετικό:

$$
\begin{tikzcd}
\mathrm{Int} \arrow[r, "\mathrm{Succ}_{\mathrm{Int}}"] \arrow[d, "\mathrm{toReal}"'] & \mathrm{Int} \arrow[d, "\mathrm{toReal}"] \\
\mathrm{Real} \arrow[r, "\mathrm{Succ}_{\mathrm{Real}}"'] & \mathrm{Real}
\end{tikzcd}
$$

Δηλαδή στην κατηγορία FPL ισχύει ότι: το αποτέλεσμα του μετασχηματισμού ενός α-κεραίου σε πραγματικό αριθμό και ο υπολογισμός του επόμενου πραγματικού αριθμού είναι ταυτόσημο με τον υπολογισμό του επόμενου ακεραίου και το μετασχηματισμό του σε πραγματικό αριθμό.

Η διάταξη στη σύνθεση δεν επηρεάζει το αποτέλεσμα.

3. ΜΟΝΟΜΟΡΦΙΣΜΟΙ – ΕΠΙΜΟΡΦΙΣΜΟΙ – ΙΣΟΜΟΡΦΙΣΜΟΙ

Ορισμός. Ένας μορφισμός $f : B \to C$ σε μια κατηγορία C είναι ένας **μονομορφισμός** εάν για κάθε ζεύγος C – μορφισμών $g : A \to B$ και $h : A \to B$ η ισότητα $f\circ g = f\circ h$ συνεπάγεται ότι $g = h$.

Πρόταση. Στην κατηγορία των συνόλων Set οι **μονομορφισμοί** είναι οι (1 – 1) συναρτήσεις (οι συναρτήσεις $f : f(x) = f(y) \Rightarrow x = y$).

Ορισμός. Ένας μορφισμός $f : A \to B$ είναι **επιμορφισμός** εάν για κάθε ζεύγος μορφισμών $g : B \to C$ και $h : B \to C$, η ισότητα $g\circ f = h\circ f$ συνεπάγεται ότι $g = h$.

Πρόταση. Στην κατηγορία Set οι επιμορφισμοί είναι οι συναρτήσεις επί. (Μία συνάρτηση $f : A \to B$ είναι επί εάν για κάθε $b\in B$ υπάρχει ένα $a\in A$ για το οποίο ισχύει $f(a) = b$).

Ορισμός. Ένας μορφισμός $f : A \to B$ είναι ισομορφισμός εάν υπάρχει ένας μορφισμός $f^{-1} : A \to B$ (αντίστροφος του f) έτσι ώστε $f^{-1}\circ f = id_A$ και $f\circ f^{-1} = id_B$.

Δύο αντικείμενα A και B είναι **ισομορφικά** εάν υπάρχει ένας ισομορφισμός μεταξύ τους.

4. ΑΡΧΙΚΑ ΚΑΙ ΤΕΛΙΚΑ ΑΝΤΙΚΕΙΜΕΝΑ

Ορισμός. Ένα αντικείμενο O καλείται αρχικό αντικείμενο, εάν για κάθε αντικείμενο A υπάρχει ένας ακριβώς μορφισμός από το O στο A.

Ορισμός (Δυϊκός του παραπάνω). Ένα αντικείμενο 1 καλείται τελικό εάν για κάθε αντικείμενο A, υπάρχει ακριβώς ένας μορφισμός από το A στο 1.

$$A \xrightarrow{\ !\ } 1 \qquad \text{(Συμβολισμός)}$$

Παραδείγματα. Στην Set το κενό σύνολο { } είναι το μόνο αρχικό αντικείμενο για κάθε σύνολο S, η κενή συνάρτηση είναι η μόνη συνάρτηση από το { } στο S.

Κάθε σύνολο με ένα μόνο στοιχείο {x} είναι ένα τελικό αντικείμενο, αφού για κάθε σύνολο S υπάρχει μια συνάρτηση από το S στο σύνολο με ένα στοιχείο {x} που απεικονίζει κάθε στοιχείο του S στο x και πολύ περισσότερο αυτή είναι η μόνη ολική συνάρτηση από το S στο$\{x\}$.

Στην κατηγορία Ω – Aly των αλγεβρών με χαρακτηριστική Ω, το αρχικό αντικείμενο είναι η αρχική άλγεβρα (ή άλγεβρα όρων) της οποίας οι φορείς αποτελούνται από όλα τα πεπερασμένα δένδρα όπου κάθε κόμβος αντιστοιχεί σε ένα τελεστή από το Ω και όπου κάθε κόμβος έχει το πολύ ar(ω) υποδένδρα.

Ο μοναδικός ομομορφισμός από την άλγεβρα όρων σε μια άλλη Ω – άλγεβρα συνιστά μια συνάρτηση σημασιολογικής ερμηνείας.

5. ΓΙΝΟΜΕΝΑ

Το καρτεσιανό γινόμενο δύο συνόλων A και B ορίζεται σαν:

$$A \times B = \{(a, b) \mid a \in A \text{ και } b \in B\}$$

Τα στοιχεία ενός συνόλου μπορούν να αναπαρασταθούν από μορφισμούς από τα σύνολα με ένα μόνο στοιχείο στο σύνολο

$$\{x\} \to S \qquad \{x\text{: στοιχεία του } S\}$$

Είναι σε (1 – 1) αντιστοιχία με τα στοιχεία του συνόλου S.

Δηλαδή εάν x είναι ένα στοιχείο του S $x : 1 \to S$, 1: σύνολο με ένα στοιχείο και $f : S \to T$, τότε το στοιχείο f(x) είναι το μοναδικό στοιχείο του T που είναι στην εικόνα της σύνθετης συνάρτησης $f \circ x$.

Χρειαζόμαστε ένα χαρακτηρισμό των γινομένων μέσα από μορφισμούς μεταξύ αντικειμένων.

Όταν ορίζουμε το γινόμενο μεταξύ δύο συνόλων A και B, ορίζουμε επίσης τις συναρτήσεις προβολής $\pi_1 : A \times B \to A$ και $\pi_2 : A \times B \to B$. Στην ουσία ορίζουμε επίσης τριάες της μορφής $(A \times B, \pi_1, \pi_2)$.

Εάν ορίσουμε όλες τις τριάδες (X, f_1, f_2) που αποτελούνται από ένα σύνολο X και δύο συναρτήσεις $f_1 : X \to A$ και $f_2 : X \to B$, τότε η τριάδα $(A \times B, \pi_1, \pi_2)$ είναι «βέλτιστος αντιπρόσωπος» αυτών των συνόλων με την εξής έννοια:

Ας υποθέσουμε ότι για κάποιο σύνολο C, υπάρχουν δύο συναρτήσεις $f : C \to A$ και $g : C \to B$. Τότε, σχηματίζουμε μια συνάρτηση γινόμενο $<f, g> : C \to A \times B$, που ορίζεται ως εξής:

$$<f, g>(x) = (f(x),g(x)), \text{ όπου}$$
$$f = \pi_1 \circ <f, g> \text{ και}$$
$$g = \pi_2 \circ <f, g>$$

Η συνάρτηση $<f, g>$ είναι η **μόνη** συνάρτηση από το C στο $A \times B$ με αυτή την ιδιότητα.

Ορισμός. Ένα **γινόμενο (product)** δύο αντικειμένων A και B είναι ένα αντικείμενο $A \times B$, μαζί με δύο προβολικούς μορφισμούς $\pi_1 : A \times B \to A$ και $\pi_2 : A \times B \to B$, έτσι ώστε για κάθε αντικείμενο C και ζεύγος μορφισμών $f : C \to A$ και $g : C \to B$ υπάρχει ακριβώς ένας ενδιάμεσος μορφισμός $<f, g> : C \to A \times B$ ο οποίος κάνει το παρακάτω διάγραμμα μεταθετικό:

$$\begin{tikzcd}
 & C \arrow[dl, "f"'] \arrow[d, dashed, "<f,g>"] \arrow[dr, "g"] & \\
A & A\times B \arrow[l, "\pi 1"] \arrow[r, "\pi 2"'] & B
\end{tikzcd}$$

έτσι ώστε

$$\pi_1 \circ <f, g> = f$$

και

$$\pi_2 \circ <f, g> = g.$$

Εάν μια κατηγορία C έχει ένα γινόμενο $A \times B$ για κάθε ζεύγος αντικειμένων λέμε ότι η **κατηγορία έχει γινόμενα**.

Ορισμός. Εάν $A \times C$ και $B \times D$ είναι δύο γινόμενα, τότε για κάθε ζεύγος από μορφισμούς $f : A \to B$ και $g : C \to D$, η **απεικόνιση γινόμενο**

$$f \times g : A \times C \to B \times D$$

είναι η απεικόνιση – μορφισμός

$$<f\circ\pi_1, g\circ\pi_2>.$$

Η δυαδική έννοια είναι το συν – γινόμενο.

Ορισμός. Ένα **συν – γινόμενο (co – product)** δύο αντικειμένων A και B είναι ένα αντικείμενο A + B, μαζί με δύο μορφισμούς

$$i_1 : A \rightarrow A + B \quad \text{(π.χ. «περιέχεσθαι»)}$$

και

$$i_2 : A \rightarrow A + B$$

έτσι ώστε για κάθε αντικείμενο C και ζεύγος μορφισμών $f : A \rightarrow C$ και $g : B \rightarrow C$ υπάρχει ακριβώς ένας μορφισμός

$$[f, g] : A + B \rightarrow C$$

που κάνει το παρακάτω διάγραμμα μεταθετικό:

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[dr, "f"'] & A + B \arrow[d, dashed, "{[f, g]}"] & B \arrow[l, "i_2"'] \arrow[dl, "g"] \\
& C &
\end{tikzcd}$$

Ορισμός. Ένα γινόμενο μιας οικογένειας $(A_i)_{i\in I}$ από αντικείμενα

$$\left(\pi_i : \left(\prod_{i\in I} A_i\right) \rightarrow A_i\right)_{i\in I}$$

αποτελείται από ένα αντικείμενο $\prod_{i\in I} A_i$ και μια οικογένεια από προβολικούς μορφισμούς $\left(\pi_i : \left(\prod_{i\in I} A_i\right) \rightarrow A_i\right)_{i\in I}$ έτσι ώστε για κάθε αντικείμενο C και κάθε οικογένεια μορφισμών $(f_i : C \rightarrow A_i)_{i\in I}$ υπάρχει ένας μοναδικός μορφισμός

$$\langle f_i \rangle_{i\in I} : C \to (\Pi_{i\in I} A_i)$$

έτσι ώστε το ακόλουθο διάγραμμα να είναι μεταθετικό για κάθε $i \in I$:

$$\begin{tikzcd}
C \arrow[d, dashed, "\langle f_i \rangle_{i\in I}"'] \arrow[dr, "f_i"] & \\
\prod_{i\in I} A_i \arrow[r, "\pi_i"'] & A_i
\end{tikzcd}$$

Ορισμός. Έστω $f,g : A \to B$. Ένας μορφισμός $e : X \to A$ καλείται **εξισωτής** των f, g αν

1. $f \circ e = g \circ e$
2. $\forall e' : X' \to A$ τέτοιο ώστε $f \circ e' = g \circ e'$ $\exists! k : X' \to X$ τέτοιο ώστε $e \circ k = e'$

$$\begin{tikzcd}
X \arrow[r, "e"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \\
X' \arrow[u, dashed, "k"] \arrow[ur, "e'"'] & &
\end{tikzcd}$$

Παραδείγματα.

- **Set:**

$$f : \mathbb{R} \times \mathbb{R} \to \mathbb{R},\ f(x,y) = x^2 - y^2$$

$$g : \mathbb{R} \times \mathbb{R} \to \mathbb{R},\ g(x,y) = 1$$

$$X = \{(x,y) \in \mathbb{R} \times \mathbb{R} \,|\, x^2 + y^2 = 1\}$$ (μοναδιαίος κύκλος)

Ο $i : X \to \mathbb{R} \times \mathbb{R}$, $i(x,y = (x,y)$ είναι εξισωτής των f, g.

- Μερική διάταξη (ως κατηγορία): για $e' = id_A$,

$$\begin{tikzcd} X \arrow[r, "e"] & A \\ A \arrow[u, dashed, "k"] \arrow[ur, "id_A"'] & \end{tikzcd}$$

$X \leq A$, $A \leq X \Rightarrow A = X$ άρα $k = id_A$ [το πολύ ένα βέλος από το Α στο Α], $e \circ id_A = id_A$ $\Rightarrow e = id_A$.

Πρόταση. Κάθε εξισωτής είναι μονομορφισμός.

Απόδειξη:

$$\begin{tikzcd} C \arrow[r, shift left, "h"] \arrow[r, shift right, "K"'] & X \arrow[r, "e"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \end{tikzcd}$$

Έστω $e \circ h = e \circ k = \ell$.

Τότε

$$f \circ \ell = f \circ (e \circ h) = (f \circ e) \circ h = (g \circ e) \circ h = g \circ (e \circ h) = g \circ \ell$$

δηλαδή $\exists! m : C \to X$ τέτοιος ώστε $e \circ m = \ell$ [e εξισωτής]

δηλαδή $m = h = k$.

Πρόταση. Έστω $e : X \to A$, $e' : X' \to A$ εξισωτές των $f,g : A \to B$. Τότε υπάρχει μοναδικός ισομορφισμός $j : X \to X'$ τέτοιος ώστε: $e' \circ j = e$.

Απόδειξη:

$$\begin{tikzcd}
X \arrow[r, "e"] \arrow[d, shift left, "j"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \\
X' \arrow[u, shift left, "j'"] \arrow[ur, "e'"'] & &
\end{tikzcd}$$

$\exists! j' : X' \to X$ τέτοιος ώστε: $e' = e \circ j'$ [e εξισωτής],

$\exists! j : X \to X'$ τέτοιος ώστε $e = e' \circ j$ [e΄ εξισωτής]

$e \circ j' \circ j = e' \circ j = e = e \circ id_X$

δηλαδή: $j' \circ j = id_X$

Ομοίως $j \circ j' = id_X$, δηλαδή ο j είναι ισομορφισμός.

Πρόταση. Ένας εξισωτής που είναι επιμορφισμός είναι ισομορφισμός..

Απόδειξη:

$$\begin{tikzcd}
X \arrow[r, "e"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \\
A \arrow[u, dashed, "k"] \arrow[ur, "id_A"'] & &
\end{tikzcd}$$

$f \circ e = g \circ e$ [e εξισωτής]

$\Rightarrow f = g$ [e επιμορφισμός]

$\Rightarrow f \circ id_A = g \circ id_A$

$\Rightarrow \exists! k : A \to X$ τέτοιο ώστε $e \circ k = id_A$

$\Rightarrow e \circ k \circ e = id_A \circ e = e = e \circ id_X$

$\Rightarrow k \circ e = id_X$ [e μονομορφισμός]

Ορισμός. Έστω f,g : A → B. Ένα βέλος e : B → C καλείται **συνεξισωτής** των f, g αν και μόνο εάν:

1. $e \circ f = e \circ g$
2. $\forall e' : B \to D$ τ.ω. $e' \circ f = e' \circ g$ $!k : C \to D$ τ.ω. $k \circ e = e'$.

$$\begin{tikzcd}
A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \arrow[r, "e"] \arrow[dr, "e'"'] & C \arrow[d, dashed, "k"] \\
 & & D
\end{tikzcd}$$

Παράδειγμα: Στην κατηγορία **Set**:

$$\begin{tikzcd}
A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B
\end{tikzcd}
\qquad
\begin{tikzcd}
 & f(a) \\
a \arrow[ur, "f"] \arrow[dr, "g"'] & \\
 & g(a)
\end{tikzcd}$$

$$R \overset{ορ}{=} \{(f(a), g(a)) \mid a \in B\} \subseteq B \times B$$

Έστω $\tilde{R}$ η ελάχιστη σχέση ισοδυναμίας που περιέχει την R:

$$\dot{\bar{R}}_1 \overset{ορ}{=} \{(b_2, b_1 \mid (b_1, b_2) \in R\} \cup \{(b,b) \mid b \in B\} \;\; \mathrm{U}\, R$$

$$R_{i+1} \overset{ορ}{=} \{(b_1, b_2 \mid \exists b \in B (b_1, b) \in R_1 \text{ και } (b, b_2) \in R_2\} \;\; i \geq 1$$

$$\tilde{R} \overset{ορ}{=} \bigcup_{i \geq 1} R_i \qquad \text{(μεταβατική κλειστότητα της } R_1\text{)}$$

$$\text{Έστω } [b] \overset{ορ}{=} \{c \in B \mid (b,c) \in \tilde{R} \qquad \text{(κλάση ισοδυναμίας)}$$

$$B/\tilde{R} \overset{ορ}{=} \{[b] \mid b \in B\} \qquad \text{(σύνολο πηλίκο)}$$

$j : B \to B/\tilde{R}$ j συνεξίσωσης των f, g

$b \mapsto [b]$.

Ορισμός. Εφέλκυση (pullback) των βελών $f : A \to C$, $g : B \to C$ καλείται ένα αντικείμενο P με βέλη $f' : P \to B$, $g' : P \to A$ τ.ω. $f \circ g' = g \circ f'$ και αν $i : X \to A$, $j : X \to B$ είναι τ.ω. $f \circ i = g \circ j$ τότε υπάρχει μοναδικό $k : X \to P$ τ.ω. $I = g' \circ k$, $j = f' \circ k$.

$$\begin{tikzcd}
X \arrow[dr, dashed, "k"] \arrow[drr, bend left, "j"] \arrow[ddr, bend right, "i"'] & & \\
& P \arrow[r, "f'"] \arrow[d, "g'"'] & B \arrow[d, "g"] \\
& A \arrow[r, "f"'] & C
\end{tikzcd}$$

Παραδείγματα.

- Αντίστροφη εικόνα συνόλου:

$$\begin{tikzcd}
P = f^{-1}[B] \arrow[r, "f/_P"] \arrow[d, "\subseteq"'] & B \arrow[d, "\subseteq"] \\
A \arrow[r, "f"'] & C
\end{tikzcd}$$

- Τομή: $A, B \subseteq C$

$$\begin{tikzcd} A\cap B \arrow[r, "\subseteq"] \arrow[d, "\subseteq"'] & B \arrow[d, "\subseteq"] \\ A \arrow[r, "\subseteq"'] & C \end{tikzcd}$$

- Περιορισμένο γινόμενο:

$$\begin{tikzcd} P \arrow[r, "\Pi_2\,|_P"] \arrow[d, "\Pi_1\,|_P"'] & B \arrow[d, "g"] \\ A \arrow[r, "f"'] & C \end{tikzcd}$$

$P \overset{\text{ορ}}{=} \{(a,b) \in A \times B | f(a) = g(b)\}$.

- Έστω C κατηγορία με τελικό αντικείμενο 1. Αν το παρακάτω διάγραμμα είναι ε-φέλκυση

$$\begin{tikzcd} P \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "!"] \\ A \arrow[r, "!"'] & 1 \end{tikzcd}$$

τότε $P = A \times B$, $g = \pi_1$, $f = \pi_2$.

- Έστω τυχούσα κατηγορία C. Αν το παρακάτω διάγραμμα είναι εφέλκυση

$$\begin{tikzcd} X \arrow[r, "e"] \arrow[d, "e"'] & A \arrow[d, "g"] \\ A \arrow[r, "f"'] & B \end{tikzcd}$$

τότε e εξισωτής των f, g.

Λήμμα (εφέλκυσης). Έστω το ακόλουθο μεταθετικό διάγραμμα:

$$\begin{tikzcd} \cdot \arrow[r] \arrow[d] \arrow[dr, phantom, "\mathrm{I}"] & \cdot \arrow[r] \arrow[d] \arrow[dr, phantom, "\mathrm{II}"] & \cdot \arrow[d] \\ \cdot \arrow[r] & \cdot \arrow[r] & \cdot \end{tikzcd}$$

(i) Αν τα εσωτερικά τετράγωνα I και II είναι εφελκύσεις τότε και το εξωτερικό τετράγωνο είναι εφέλκυση.

(ii) Αν το εξωτερικό τετράγωνο και το δεξί τετράγωνο είναι εφελκύσεις τότε και το αριστερό τετράγωνο είναι εφέλκυση.

Απόδειξη: Άσκηση.

Πρόταση. Έστω:

$$\begin{tikzcd} P \arrow[r, "f'"] \arrow[d, "g'"'] & B \arrow[d, "g"] \\ A \arrow[r, "f"'] & C \end{tikzcd}$$

εφέλκυση και g μονομορφισμός. Τότε g’ μονομορφισμός.

Παράδειγμα. Επαλήθευση προγραμμάτων (Floyd – Hoave λογική).

1. {x < 3} **προ**συνθήκη

2. x : = x + 1

3. {x < 24} **μετα**συνθήκη,

ασθενέστερη προσυνθήκη: {x < 23}.

Για κάθε μετασυνθήκη υπάρχει ασθενέστερη προσυνθήκη.

$$\begin{tikzcd} \text{\textbf{ασθενεστ. προ}συνθήκη } f^{-1}[s] \text{ (\textbf{εφέλκυση})} \arrow[r] \arrow[d] & D \text{ inputs} \arrow[d, "f \quad \text{τμήμα προγράμματος}"] \\ S \text{ \textbf{μετα}συνθήκη} \arrow[r] & E \text{ outputs} \end{tikzcd}$$

6. ΚΑΘΟΛΙΚΕΣ ΚΑΤΑΣΚΕΥΕΣ

Μια **καθολική κατασκευή** περιλαμβάνει μια κλάση αντικειμένων και μορφισμών που τα συνοδεύουν, τα οποία χαρακτηρίζονται από μια κοινή ιδιότητα και επιλέγει τα αντικείμενα τα οποία είναι «τελικά» όταν αυτή η κλάση θεωρηθεί σαν κατηγορία.

Πιο συγκεκριμένα, ο ορισμός των γινομένων των A και B (X, x_1, x_2) στη C

$$\begin{tikzcd} & X \arrow[dl, "x_1"'] \arrow[dr, "x_2"] & \\ A & & B \end{tikzcd}$$

$x_1: X \to A$, $x_2: X \to B$.

Τα αντικείμενα είναι οι τριάδες (X, x_1, x_2) και οι μορφισμοί $m : (W, w_1, w_2) \to (X, x_1, x_2)$ είναι C – μορφισμοί $m : W \to X$ έτσι ώστε $w_1 = x_1 \circ m$ και $w_2 = x_2 \circ m$:

$$
\begin{tikzcd}
& W \arrow[ddl, "w_1"'] \arrow[d, "m"] \arrow[ddr, "w_2"] & \\
& X \arrow[dl, "x_1"] \arrow[dr, "x_2"'] & \\
A & & B
\end{tikzcd}
$$

Οι **μορφισμοί** που ορίζονται με μια καθολική κατασκευή ονομάζονται **καθολικοί** ή ότι ικανοποιούν την καθολική ιδιότητα.

Μία συν – καθολική κατασκευή είναι μια δυϊκή της παραπάνω αντί να έχουμε αρχικά αντικείμενα αντί για τελικά.

Ένα τελικό αντικείμενο σ' αυτή την κατηγορία είναι η τριάδα (P, p_1, p_2):

$$
\begin{tikzcd}
& W \arrow[ddl, "w_1"'] \arrow[d, dashed, "!"] \arrow[ddr, "w_2"] & \\
& P \arrow[dl, "p_1 = \pi_1"] \arrow[dr, "p_2 = \pi_2"'] & \\
A & & B
\end{tikzcd}
$$

Το αντικείμενο $P = A \times B$. Ο μοναδικός μορφισμός από το (W, w_1, w_2) στο $(A \times B, \pi_1, \pi_2)$ ονομάζεται **τελικός** και γράφεται $<w_1, w_2>$.

7. ΟΡΙΑ

Οι μέχρι τώρα έννοιες που εξετάσαμε (αρχικά και τελικά αντικείμενα, εξισωτές, συν – εξισωτές, εφελκύσεις και εξωθήσεις) αποτελούν παραδείγματα καθολικών κατασκευών.

Αποτελούν δε ειδικές περιπτώσεις των πιο γενικών εννοιών του ορίου και του συν – ορίου ενός διαγράμματος.

Ορισμός. Έστω C μια κατηγορία και D ένα διάγραμμα στη C. Ένας κώνος του D είναι ένα C – αντικείμενο X και μορφισμοί $f_i : X \to D_i$ (ένα για κάθε αντικείμενο $D_ι$ στο D) έτσι ώστε για κάθε μορφισμό g στο D, το διάγραμμα

$$\begin{tikzcd} & X \arrow[dl, "f_i"'] \arrow[dr, "f_j"] & \\ D_i \arrow[rr, "g"'] & & D_j \end{tikzcd}$$

να είναι μεταθετικό.

Χρησιμοποιούμε το συμβολισμό $\{ f_i : X \to D_i\}$ για τους κώνους.

Ορισμός. Ένα όριο για το διάγραμμα D είναι ένας κώνος $\{f_i : X \to D_i\}$ με την ιδιότητα ότι εάν $\{f_i' : X' \to D_i\}$ είναι ένας άλλος κώνος για το D τότε υπάρχει ένας μοναδικός μορφισμός $k : X' \to X$ έτσι ώστε το διάγραμμα

$$\begin{tikzcd} X' \arrow[rr, dashed, "k"] \arrow[dr, "f_i'"'] & & X \arrow[dl, "f_i"] \\ & D_i & \end{tikzcd}$$

να είναι μεταθετικό για κάθε D_i στο D.

«Οι κώνοι για ένα διάγραμμα D αποτελούν μια κατηγορία».

Ένα όριο είναι ένα τελικό αντικείμενο σ' αυτή την κατηγορία.

Παραδείγματα ορίων

1) Έστω δύο C – αντικείμενα A και B, και έστω D το διάγραμμα

$$A \qquad\qquad\qquad B$$

με δύο κορυφές (A, B) και καθόλου ακμές.

Ένας κώνος γι' αυτό το διάγραμμα είναι ένα αντικείμενο X με δύο μορφισμούς f και g της μορφής:

$$\begin{tikzcd} A & X \arrow[l, "f"'] \arrow[r, "g"] & B \end{tikzcd}$$

Ένας κώνος όριο των παραπάνω είναι ένα γινόμενο των A και B.

2) Έστω D το παραπάνω διάγραμμα:

$$\begin{tikzcd} & B \arrow[d, "g"] \\ A \arrow[r, "f"] & C \end{tikzcd}$$

με τρεις άκρες και δύο ακμές.
Ένας κώνος για το D είναι ένα αντικείμενο P και τρεις μορφισμοί f΄, g΄ και h έτσι ώστε το ακόλουθο διάγραμμα να είναι μεταθετικό:

$$\begin{tikzcd} P \arrow[r, "f'"] \arrow[d, "g'"'] \arrow[dr, "h"] & B \arrow[d, "g"] \\ A \arrow[r, "f"'] & C \end{tikzcd}$$

Επειδή $f \circ g' = h = g \circ f$ το παραπάνω διάγραμμα είναι ισοδύναμο με το

$$\begin{tikzcd} D \arrow[r, "f'"] \arrow[d, "g'"'] & B \arrow[d, "g"] \\ A \arrow[r, "f"'] & C \end{tikzcd}$$

Δυϊκά:

Ορισμός. Ένας (συν) κώνος για ένα διάγραμμα D σε μια κατηγορία C είναι ένα C – αντικείμενο X και μια συλλογή από μορφισμούς $f_i : D_i \to X$ έτσι ώστε $f_i \circ g = f_i$ για κάθε g στο D. Ένα (συν) όριο ή αντίστροφο όριο για το D είναι τότε ένας συν – κώνος $\{f_i : D_i \to X\}$ με την συν – καθολική ιδιότητα ότι για κάθε άλλο συν – κώνο $\{f_i{'}: D_i\, X' \to X'\}$ υπάρχει ένας μοναδικός μορφισμός

$$k : X \to X'$$

έτσι ώστε το διάγραμμα

$$\begin{tikzcd}
X \arrow[rr, dashed, "k"] \arrow[dr, "f_i"'] & & X' \arrow[dl, "f_i{'}"] \\
& D_i &
\end{tikzcd}$$

να είναι μεταθετικό για κάθε αντικείμενο D_i στο D.

Διακριτές κατηγορίες είναι αυτές οι κατηγορίες που οι μόνοι μορφισμοί είναι οι ταυτοτικοί.

Θεώρημα (Θεώρημα Ορίων). Έστω D ένα διάγραμμα σε μια κατηγορία C, με V το σύνολο των ακμών και E το σύνολο των κορυφών. Εάν κάθε E – οικογένεια από αντικείμενα στη C έχει ένα γινόμενο και κάθε ζεύγος μορφισμών στη C έχει ένα εξισωτή, **τότε το D έχει ένα όριο**.

8. ΕΚΘΕΤΙΚΟΤΗΤΑ

Δίνει τη θεωρητική ερμηνεία του currying (μια συνάρτηση με πεδίο ορισμού ένα καρτεσιανό γινόμενο $A \times B$ ανάγεται σε μια συνάρτηση που έχει πεδίο ορισμού το A ή το B).

Εάν τα A και B είναι σύνολα, η συλλογή

$$B^A = \{f : A \to B\}$$

όλων των συναρτήσεων από το A στο B είναι ένα σύνολο.

Εάν θέλουμε να χαρακτηρίσουμε το B^A με τη βοήθεια μορφισμών τότε έχουμε ότι σε κάθε B^A αντιστοιχεί ένας μορφισμός

$$\text{eval} : (B^A \times A) \to B$$

$$\text{eval}(f,a) = f(a).$$

Δηλαδή, για $f : A \to B$ και $\alpha \in A$ συνεπάγεται ότι $f(a) \in B$. Η eval έχει καθολική ιδιότητα μεταξύ όλων των συναρτήσεων $g : (C \times A) \to B$. Για κάθε g, υπάρχει ακριβώς μια συνάρτηση $curry(g) : C \to B^A$ έτσι ώστε το ακόλουθο διάγραμμα να είναι μεταθετικό:

$$\begin{tikzcd}
B^A \times A \arrow[r, "eval"] & B \\
C \times A \arrow[u, dashed, "curry(g) \times id_A"] \arrow[ur, "g"'] &
\end{tikzcd}$$

$$\boxed{g_c(a) = g(c,a)}$$

$$curry(g)(c) = g_c.$$

Δηλαδή για κάθε $(c,a) \in C \times A$ ισχύει:

$$\begin{aligned}
(eval \circ (curry(g) \times id_A))(c,a) &= eval(curry(g)(c),a) \\
&= eval(g_c,a) \\
&= g_c(a) \\
&= g(c,a).
\end{aligned}$$

Έτσι το διάγραμμα είναι μεταθετικό.

Ορισμός. Έστω C μια κατηγορία με γινόμενα A και B αντικείμενα στη C.

Ένα αντικείμενο B^A είναι ένα **εκθετικό αντικείμενο** εάν υπάρχει ένας μορφισμός $eval_{AB} : (B^A \times A) \to B$ έτσι ώστε για κάθε αντικείμενο C και μορφισμό $g : (C \times A) \to B$ υπάρχει ένας μοναδικός μορφισμός $curry(g) : C \to B^A$ που κάνει το εξής διάγραμμα μεταθετικό:

$$\begin{tikzcd}
B^A \times A \arrow[r, "eval_{AB}"] & B \\
C \times A \arrow[u, dashed, "curry(g) \times id_A"] \arrow[ur, "g"'] &
\end{tikzcd}$$

έτσι ώστε υπάρχει μοναδικό curry(g) :

$$\mathrm{eval}_{AB} \circ (\mathrm{curry}(g) \times \mathrm{id}_A) = g.$$

Ορισμός. Μια καρτεσιανή κλειστή κατηγορία είναι μια κατηγορία με:

→ Τελικό αντικείμενο

→ (δυαδικό) γινόμενο

→ εκθετικότητα.

Παραδείγματα

1) Η κατηγορία Set είναι καρτεσιανά κλειστή, με $B^A = \mathrm{Set}(A,B)$.
2) Η κατηγορία CPO διατάξεων και συνεχών συναρτήσεων είναι καρτεσιανά κλειστή με B^A το μερικά διατεταγμένο σύνολο των συνεχών συναρτήσεων από το A στο B.

ΣΥΝΑΡΤΗΤΕΣ

Ορισμός. Έστω B_1 και B_2 δύο κατηγορίες. Τότε, ένας συναρτητής $F : B_1 \to B_2$ αποτελείται από:

- μια απεικόνιση $F_O : |O_1| \to |O_2|$, που στέλνει κάθε o_1 από το B_1 στο $F_O(o_1)$ του B_2,
- και μια απεικόνιση $F_A : |I_1| \to |I_2|$, που στέλνει κάθε μορφισμό $f : o_1 \to o_2$ του B_1 σε ένα μορφισμό $F_A(f) : F_A(o_1) \to F_A(o_2)$ του B_2, έτσι ώστε:
  - $F_A(1_o) = 1_{F_o(o)}$,
  - και $F_A(F ; g) = F_A(f) ; F_A(g)$, όταν ορίζεται η σύνδεση μορφισμών $f ; g$.

Παράδειγμα: Δοθέντος ενός συνόλου S, μπορούμε να σχηματίσουμε το σύνολο List(S) από πεπερασμένες λίστες με στοιχεία από το S. Η απεικόνιση List μπορεί να θεωρηθεί σαν τη συνάρτηση που στέλνει αντικείμενα από το **Set** στο **Set** (στέλνει σύνολα σε σύνολα από πεπερασμένες λίστες συμβόλων). Επίσης, κάθε συνάρτηση $f : S \to S'$ αντιστοιχεί σε μια συνάρτηση $\mathrm{List}(f) : \mathrm{List}(S) \to \mathrm{List}(S')$ η οποία δοθείσεις μιας λίστας $L = [s_1,s_2,\ldots,s_n]$, την απεικονίζει στην $\mathrm{List}(f)(L) = [f(s_1), f(s_2),\ldots, f(s_n)]$.

ΑΠΌ ΤΙΣ ΚΑΤΗΓΟΡΙΕΣ ΣΤΗ ΛΟΓΙΚΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Η θεωρία των κατηγοριών προτείνει μια ιδιαίτερη οπτική αναφορικά με την περιγραφή των μαθηματικών αντικειμένων. Σύμφωνα μ' αυτή οι ιδιότητες των μαθηματικών δομών προκύπτουν μέσα από τη μελέτη των μορφισμών που τις διατηρούν. Η προαναφερθείσα μέθοδος επεκτείνεται και στη συμβολική λογική εξαιτίας της μαθηματικής φύσεως της τελευταίας. Έτσι, η θεωρία των κατηγοριών έχει εφαρμοστεί σε περιοχές της λογικής όπως η θεωρία αλγορίθμων, η θεωρία τύπων, η θεωρία αποδείξεων, η αφηρημένη θεωρία μοντέλων και η εξισωτική λογική. Οι εφαρμογές αυτές τοποθετούνται σε μια ολόκληρη σχολή έρευνας που σκοπό της έχει την αλγεβροποίηση της συμβολικής λογικής. Τα τελευταία χρόνια η κατηγοριακή προσέγγιση της συμβολικής λογικής έχει γνωρίσει ιδιαίτερη άνθιση κυρίως λόγω των ευρύτατων εφαρμογών της στη θεωρητική πληροφορική. Η θεωρία κατηγοριών μπορεί να μας βοηθήσει να απαντήσουμε το ερώτημα:

σε τι συνίσταται ένα λογικό σύστημα;

Πιο συγκεκριμένα παρουσιάζουμε, συνοπτικά βέβαια, τις δύο πιο σημαντικές σχετικές θεωρίες;

- την **κατηγοριακή λογική** κατά J. Lambek,
- την αφηρημένη θεωρία μοντέλων κατά J. Goguen και R. Burstall ή **θεωρία θεσμών** (**theory of institutions**).

Οι έννοιες από τη θεωρία κατηγοριών που χρησιμοποιούμε αποκτούν περιεχόμενο συμβατό με τη συβμολική λογική μόνο όταν ερμηνευτούν και ονομαδοτηθούν κατάλληλα. Έτσι, για παράδειγμα, μια κατηγορία μπορεί άλλοτε να ερμηνευτεί ως κατηγορία προτάσεων και άλλοτε ως κατηγορία μοντέλων ενός λογικού συστήματος, ανάλογα με το πώς θα ονομαστούν τα αντικείμενα και οι μορφισμοί της.

Η ΚΑΤΗΓΟΡΙΑΚΗ ΛΟΓΙΚΗ ΚΑΤΑ J. LAMBEK

Η κατηγοριακή λογική κατά J. Lambek αποτελεί μια από τις πιο γενικές θεωρίες απόδείξεων. Ασχολείται με λογικές παραγωγές (deductions) της μορφής : $A_1A_2\ldots A_n \rightarrow A_{n+1}$, όπου οι τύποι (formulas) $A_1 \ldots A_n$ (οι υποθέσεις) παράγουν λογικά το A_{n+1}. Στα παρακάτω θα παρουσιάσουμε ένα σύστημα κανόνων λογικής παραγωγής τύπου Gentzen με τη βοήθεια της κατηγοριακής λογικής του J. Lambek.

Ορισμός 1. Ένα πολυγράφημα (**multigraph**) αποτελείται από:

1. μια κλίση A από βέλη (arrows), που θα τα καλούμε επίσης *λογικές παραγωγές*,

2. μια κλάση O από αντικείμενα (objects), που θα τα καλούμε επίσης τύπους και δύο απεικονίσεις:

   - $\theta_0 : A \to O^*$
   - $\theta_1 : A \to O$.

Η θ_0 ονομάζεται **αρχή** (**domain**) και η θ_1 **τέλος** (**codomain**). Το O^* αποτελείται από λίστες της μορφής $A_{i_1}A_{i_2}...A_{i_k}$, όπου $A_{i_1}A_{i_2}...A_{i_k} \in O$. Έτσι, για κάθε βέλος του πολυγραφήματος

$$f : A_1A_2\ldots A_n \to A_{n+1}$$

έχουμε ότι

$$\theta_0(f) = A_1A_2\ldots A_n \quad \text{και} \quad \theta_1(f) = A_{n+1}$$

Ορισμός 2. Ένα σύστημα λογικής παραγωγής (deductive system) κατά J. Lambek είναι ένα πολυγράφημα, που για κάθε αντικείμενο A υπάρχει ένας ταυτοτικός μορφισμός 1_A:A → A και για κάθε δύο μορφισμούς $f : \Theta \ \frac{f : \Gamma AA \to B}{f^i : \Gamma A \to B} \to A$ και $g : \Gamma A\Delta \to B$, υπάρχει ένας μορφισμός $g(f) : \Gamma\Theta\Delta \to B$ έτσι ώστε να ισχύει:

$$\frac{f : \Theta \to A \quad g : \Gamma A\Delta \to B}{g(f) : \Gamma\Theta\Delta \to B}$$

Η παραπάνω ιδιότητα καλείται **τομή κατά Gentzen**.

Εάν ένα λογικό σύστημα κατά Lambek εμπλουτιστεί και με τους εξής τρεις κανόνες:

$$\frac{f : \Gamma AB\Delta \to C}{f^i : \Gamma BA\Delta \to C}$$

(εναλλαγή – interchange)

$$\frac{f : \Gamma AA \to B}{f^i : \Gamma A \to B}$$

(συστολή – contraction)

$$\frac{f : \Gamma \to B}{f^i : \Gamma A \to B}$$

(εκπλεπτινση – thinning)

τότε καλείται λογικό **σύστημα κατά Gentzen**.

Ο ΑΛΓΕΒΡΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Είναι ιδιαίτερα επιθυμητό, δεδομένης και της πληθώρας των λογικών συστημάτων που εμφανίζονται στην πληροφορική, να υπάρξει μια κοινή μεθοδολογία που να αντιμετωπίζει

> *πολλές από τις σημαντικές εφαρμογές με τρόπο ανεξάρτητο από ένα συγκεκριμένο λογικό σύστημα.*

Η παραπάνω ιδέα οδήγησε τους J. Goguen και R. Burstall να εισάγουν την αφηρημένη θεωρία μοντέλων στην πληροφορική, υπό τη μορφή της **θεωρίας θεσμών** (**theory of institutions**). Η θεωρία θεσμών προτείνει μια **γενική έννοια λογικού συστήματος** η οποία μας επιτρέπει την ανάπτυξη μιας σειράς εφαρμογών της πληροφορικής ανεξάρτητα από ένα συγκεκριμένο λογικό σύστημα. Η θεωρία θεσμών προτείνει καταρχήν ένα αλγεβρικό ορισμό της αφηρημένης έννοιας του λογικού συστήματος. Οι θεσμοί είναι δηλαδή μια αλγεβρική συνεισφορά στο ανοικτό φιλοσοφικό ζήτημα του ορισμού μιας έννοιας αφηρημένου λογικού συστήματος. Στη φιλοσοφία βέβαια, η πιο διαδεδομένη απάντηση στο παραπλήσιο ερώτημα *τι είναι η λογική*; είναι εκείνη που υποστηρίζει πως λογική είναι η επιστήμη που έχει ως αντικείμενο μελέτης το συλλογισμό και τους νόμους της λογικής απαγωγής (deduction). Άλλες απαντήσεις είναι πιο ειδικές δίνοντας μεγαλύτερη έμφαση σε συγκεκριμένες συλλογιστικές πρακτικές ή ορισμούς λογικών συστημάτων, ή υποστηρίζουν πως η λογική συνδέεται με πτυχές του φαινομένου της αντίληψης και της συνείδησης (ψυχολογισμός).

Πολλές φορές στην πληροφορική υπάρχει η ανάγκη χρησιμοποίησης περισσότερων από ένα λογικών συστημάτων, ταυτόχρονα, κατά τρόπο παράλληλο και οργανωμένο σ' αυτό που διαισθητικά αντιλαμβανόμαστε σαν δομή ή σύστημα. Ειδικά στην περίπτωση των προδιαγραφών είναι επιθυμητή η χρησιμοποίηση περισσότερων από ένα λογικών συστημάτων μια και έτσι μπορούν **να αξιοποιηθούν τα ξεχωριστά πλεονεκτήματα του κάθε ενός από τα λογικά συστήματα** με σκοπό τη βελτιστοποίηση της προδιαγραφής του συστήματος που μας ενδιαφέρει. Ακόμα, τα οργανωμένα με αυτό τον τρόπο λογικά συστήματα μας επιτρέπουν ανάμεσα σε άλλα και τη δυνατότητα ορθής μετάφρασης προδιαγραφών από το ένα λογικό σύστημα σε ένα άλλο. Η θεωρία θεσμών προσφέρει κατάλληλο μαθηματικό πλαίσιο για την αξιοποίηση από την τεχνολογία λογισμικού περισσοτέρων του ενός λογικών συστημάτων.

Συνοψίζοντας, μπορεί να διακρίνει κανείς τρεις τουλάχιστον φάσεις στη σχέση λογικής και πληροφορικής.

1. Πρώτη φάση: Οι έννοιες του αλγορίθμου, οι άλγεβρες του Boole, η λ – ορισιμότητα, οι μηχανές Turing και τα αυτόματα του John von Neumann παίζουν ξεχωριστό ρόλο.

2. Δεύτερη φάση: Η πληροφορική αξιοποιεί συγκεκριμένα λογικά συστήματα όπως της λογικής του Horn, της εξωτερικής λογικής, του λ – λογισμού, κ.τ.λ.

3. Τρίτη φάση: Εδώ τον κύριο ρόλο παίζει μια αφηρημένη έννοια λογικού συστήματος (π.χ. ένας θεσμός). Τα συγκεκριμένα λογικά συστήματα αντιμετωπίζονται σαν ειδικές περιπτώσεις αυτής της αφηρημένης έννοιας, ενώ οι εφαρμογές διατυπώνονται στη μέγιστη δυνατή γενικότητά τους.

Η ΘΕΩΡΙΑ ΘΕΣΜΩΝ

Η θεωρία θεσμών αποτελεί γενίκευση της *αφηρημένης θεωρίας μοντέλων* του Barwise με καταρχήν στόχο να καλύψει όλο το εύρος των λογικών συστημάτων που χρησιμοποιούνται στην πληροφορική. Όμως, παρά τη γενετική σχέση τους, η αφηρημένη θεωρία μοντέλων και η θεωρία θεσμών κάθε άλλο παρά έχουν την ίδια φιλοσοφία. Πιο συγκεκριμένα, τα βασικά συστατικά στοιχεία ενός λογικού συστήματος (συντακτικό, σημασιολογία, σχέση αλήθειας) σύμφωνα με την αφηρημένη θεωρία μοντέλων κινούνται μέσα στο πνεύμα της πρωτοβάθμιας ή της δευτεροβάθμιας λογικής. Αντίθετα, τα ίδια στοιχεία αντιμετωπίζονται από τη θεωρία θεσμών στο επίπεδο γενικότητας της θεωρίας κατηγοριών.

- η παρατήρηση ότι

  πολλές από τις πιο σημαντικές εφαρμογές της πληροφορικής είναι ανεξάρτητες από εάν συγκεκριμένο λογικό σύστημα

- η ανάγκη ορθών μεταφράσεων των προτάσεων ενός λογικού συστήματος σε προτάσεις ενός άλλου συστήματος

- ο μεγάλος αριθμός λογικών συστημάτων που χρησιμοποιείται από την πληροφορική και η συνακόλουθη ανάγκη οργάνωσης και ταξινόμησής τους μέσα σε ένα ενιαίο λογικό πλαίσιο

- η ανάγκη για σύνθεση νέων λογικών συστημάτων από τη συνένωση δύο ή και περισσότερων άλλων παλαιοτέρων, όπως έχουν αποδείξει ότι απαιτούν οι ανάγκες της πληροφορικής.

Οι θεσμοί προσπαθούν να συμπεριλάβουν ως ειδικές περιπτώσεις όλα τα λογικά συστήματα που χρησιμοποιούνται στην πληροφορική καθώς και όλα εκείνα που είναι πιθανόν να χρησιμοποιηθούν στο μέλλον.

Περιγραφικά ένας **θεσμός** συνίσταται από

- μια κατηγορία *χαρακτηριστικών* (*signatures*). Η κατηγορία αυτή περιλαμβάνει όλο το απαραίτητο (μη λογικό) υλικό για το σχηματισμό των προτάσεων του λογικού συστήματος. Οι χαρακτηριστικές μπορούν να είναι είτε μεμονωμένα σύμβολα είτε σύνολα συμβόλων είτε ακόμα και κατηγορίες συμβόλων.

- Ενώ σε κάθε *χαρακτηριστική* Σ αντιστοιχεί:

  – μία κατηγορία από Σ *προτάσεις*

  – μία κατηγορία από Σ *σημασιολογίες* ή *μοντέλα* και

  – μία Σ *σχέση ικανοποίηση* (satisfaction relation) Σ *προτάσεων* από Σ *σημασιολογίες* έτσι ώστε

    κάθε αλλαγή σε επίπεδο συμβόλων (δηλαδή κάθε μορφισμός της κατηγορίας των συμβόλων) να οδηγεί αυτόματα σε μια αντίστοιχη ομαλή μεταβολή της σχέσης ικανοποίησης.

    Η ιδιότητα αυτή στη γλώσσα της θεωρίας θεσμών καλείται **συνθήκη ικανοποιησης** (satisfaction condition).

Οι γενικευμένοι θεσμοί

Η σχέση ικανοποίησης στους θεσμούς έχει οριστεί με βάση τη δίτιμη λογική. Έτσι, αποκλείονται αυτόματα από τη θεωρία θεσμών όλα εκείνα τα λογικά συστήματα που αποδέχονται την ύπαρξη περισσοτέρων από δύο αληθοτιμών όπως αυτά της τρίτιμης λογικής, της ασαφούς λογικής, κ.τ.λ. Για να καλυφθεί αυτό το κενό της θεωρίας οι J. Goguen και R. Burstall εισήγαγαν την έννοια του **γενικευμένου θεσμού** . Οι γενικευμένοι θεσμοί επιτρέπουν στην αλήθεια να λαμβάνει τιμές πάνω σε μια τυχαία κατηγορία V.

Ορισμός των θεσμών

Ορισμός 3. Ένας **θεσμός I** αποτελείται από:

1. μια κατηγορία **Sign**, τα αντικείμενα της οποίας καλούνται **χαρακτηριστικές**,

2. ένα συναρτητή **Sen**: **Sign** $\rightarrow$ **Set**, έτσι ώστε σε κάθε χαρακτηριστική να αντιστοιχεί ένα σύνολο του οποίου τα στοιχεία καλούνται **προτάσεις** πάνω σ’ αυτή τη χαρακτηριστική,

3. ένα συναρτητή **Mod**: **Sign** $\rightarrow$ **Cat**^{op} που να στέλνει σε κάθε χαρακτηριστική Σ μια κατηγορία από αντικείμενα που ονομάζονται Σ – σημασιολογίας ή μοντέλα και μορφισμούς που ονομάζονται Σ – μορφισμοί μοντέλων,

4. μία σχέση |= $\Sigma \subseteq$| **Mod**(Σ) | $\times$ **Sen**(Σ) για κάθε $\Sigma \in$| **Sign** | ονομάζεται **Σ – ικανοποίηση** έτσι ώστε για κάθε μορφισμό ο : Σ $\rightarrow$ Σ΄ της **Sign** η ακόλουθη συνθήκη ικανοποίησης ισχύει:

   m |= **Sen**(o)(e) εάν και μόνο εάν **Mod**(o)(m) |= e, για κάθε m|**Mod**(Σ)| και $e \in$ **Sen**(Σ).

Η ουσία της συνθήκης ικανοποίησης συνοψίζεται στο ότι

η αλήθεια είναι αναλλοίωτη σε σχέση με τις αλλαγές συμβόλων.

Η παρακάτω εικόνα οπτικοποιεί τη συνθήκη ικανοποίησης:

Οι θεσμοί σχηματίζουν μια κατηγορία **INS** (που καλείται κατηγορία θεσμών) η οποία έχει ως αντικείμενα θεσμούς και ως μορφισμούς θεσμών. Οι μορφισμοί θεσμών μας προσφέρουν το κατάλληλο πλαίσιο για την ορθή μετάφραση προτάσεων από ένα λογικό σύστημα σε ένα άλλο. Μέσα στην κατηγορία θεσμών INS έχει νόημα να εισάγει και να μελετήσει κανείς διάφορες κατασκευές που να βασίζονται στις ιδέες της κατηγοριακής θεωρίας γενικών συστημάτων . Έτσι, για παράδειγμα, μπορεί να προτείνει κάποιος τη σύνθεση ενός νέου θεσμού από τη συνένωση μέσω συν – γινομένων (co – products) δύο ή και περισσότερων άλλων θεσμών.

Παραδείγματα θεσμών

Έχει αποδειχθεί ότι τα αξιώματα της θεωρίας θεσμών ικανοποιούνται από λογικά συστήματα όπως εξισωτική λογική (με ή και χωρίς τύπους), εξισωτική λογική με διατεταγμένους τύπους (order sorted equational logic), εξισωτική λογική με κρυμμένους τύπους (hidden sorted equational logic), κατηγοριακή εξισωτική λογική (με ή και χωρίς περιορισμούς (constraints)). (πρωτοβάθμια λογική, δευτεροβάθμια λογική, λογική των τύπων του Horn (με ισότητα ή όχι), τροπική λογική (modal logic) κ.ά.

Οι εφαρμογές στην πληροφορική

Η έννοια της θεωρίας πάνω σε ένα θεσμό παίζει το σημαντικότερο ρόλο όσον αφορά τις εφαρμογές της θεωρίας θεσμών στην τεχνολογία λογισμικού (software engineering). Σαν μια θεωρία ορίζεται μια συλλογή Σ προτάσεων που είναι κλειστή κάτω από τη σημασιολογική παραγωγή (semantic entailment). Οι θεωρίες πάνω σε θεσμούς καθώς και οι μορφισμοί τους χρησιμοποιούνται για να ορίσουν *βασικές έννοιες της τεχνολογίας λογισμικού ανεξάρτητα από ένα συγκεκριμένο λογικό σύστημα*. Τέτοιες έννοιες περιλαμβάνουν για παράδειγμα αυτές της προδιαγραφής (specification), του τεμαχίου λογισμικού (module), του αντικειμένου (object), του παραμετρικού τεμαχίου (parametrized module), της όψης (view) κ.τ.λ.

Η έννοια του θεσμού πρωτοεμφανίστηκε στη γλώσσα προδιαγραφών Clear υπό τον όρο *γλώσσα* (*language*). Χρησιμοποιώντας τη μεθοδολογία της Clear μπορούμε να σχηματίσουμε μεγαλύτερα τεμάχια λογισμικού και προδιαγραφών από μικρότερα. Τα τεμάχια στα οποία αναφερόμαστε είναι θεωρίες πάνω σε ένα θεσμό. Η Clear είναι μία γλώσσα η οποία υποστηρίζει έντονα τόσο την τμηματοποίηση (modularization), όσο και το κρύψιμο πληροφορίας (information hidding). Η γλώσσα εξισωτικού προγραμματισμού (equational programming language) OBJ μπορεί επίσης να θεωρηθεί ειδική περίπτωση της γλώσσας Clear για το θεσμό της εξισωτικής λογικής με διατεταγμένους τύπους. Εκτός από την Clear υπάρχουν άλλες δύο γλώσσες προδιαγραφών που βασίζονται στη θεωρία θεσμών. Πρόκειται για την ASL και την Extended ML που οφείλονται στους D. Sannela και A. Tarlecki. Στοιχεία της θεωρίας θεσμών καθώς και των χαρτών (charters) και περγαμηνών (parchments) έχουν προταθεί σαν σημασιολογία του πολύ ισχυρού αποδείκτη θεωρημάτων (theorem prover) 2OBJ. O 2OBJ επιτρέπει αποδεικτική θεωρημάτων πάνω σε οποιοδήποτε λογικό σύστημα μπορεί να κωδικοποιηθεί στην εξισωτική λογική με διατεταγμένους τύπους.

Η σχέση της θεωρίας θεσμών με την πληροφορική περιλαμβάνει ανάμεσα στα άλλα ένα λογισμό για την τεμαχιοποίηση και τις επεκτάσεις τεμαχίων λογισμικού, μια θεωρία για παραμετρικού προγραμματισμού (parametrized programming), εργασίες πάνω στους τύπους και τη χρήση τους στον προγραμματισμό, τη βελτιστοποίηση προδιαγραφών (refinment). Ακόμα, υπάρχουν εφαρμογές των θεσμών στη σημασιολογία του αντικειμενοστραφούς προγραμματισμού μέσω των λογικών συστημάτων με κρυμμένους τύπους. Τα λογικά συστήματα με κρυμμένους τύπους χρησιμοποιούνται στη σημασιολογία της αντικειμενοστραφούς γλώσσας προγραμματισμού FOOPS. Έχει αποδειχθεί ότι κάθε θεσμός έχει μια επέκταση με κρυμμένους τύπους που είναι επίσης θεσμός. Μια άλλη εφαρμογή της θεωρίας θεσμών συνίσταται στη χρησιμοποίησή της στη σημασιολογία της εξισωτικής και λογικής γλώσσας προγραμματισμού με περιορισμούς (constraints) EpLog. Αλλά αποτελέσματα σχετιζόμενα με τη θεωρία θεσμών περιλαμβάνουν την απόδειξη της ισοδυναμίας μεταξύ του λήμματος παρεμβολής του Craig (Craig interpolation lemma) και του θεωρήματος μη αντιφατικότητας του Robinson σε επίπεδο θεσμών ή ακόμα τη μελέτη ελεύθερων αλγεβρικών κατασκευών πάνω σε θεσμούς.

Οι θεσμοί έχουν συνδεθεί και με θεωρίες λογικής παραγωγής (deduction). Έτσι, για παράδειγμα, οι λογικές παραγωγές θεωρούνται μορφισμοί στην κατηγορία των προτάσεων ενός θεσμού όπως ακριβώς και στην κατηγορική λογική υψηλής τάξεως (higher – order categorical logic). Μια άλλη σχετική προσέγγιση είναι αυτή των **γενικευμένων λογικών συστημάτων** (general logics) του J. Meseguer. Στην εργασία αυτή προτείνεται ένας γενικός ορισμός λογικού συστήματος που βασίζεται σε ένα συνδυασμό μιας επέκτασης του αφηρημένου ορισμού της λογικής παραγωγής κατά Dana Scott και της θεωρίας θεσμών. Στη συγκεκριμένη εργασία εισάγεται επίσης μια γενική μορφή αποδεικτικού λογισμού (proof calculi). Σύμφωνα με τον Meseguer με κάθε γλώσσα λογικού προγραμματισμού είναι συνδεδεμένος και ένας θεσμός, έστω I, έτσι ώστε οι εντολές της να είναι προτάσεις του I, η λειτουργική σημασιολογία της (operational semantics) είναι ένα σύστημα λογικής παραγωγής (deductive system) συνδεδεμένο με τον I ενώ η σημασιολογία της δίδεται από το συναρτητή μοντέλων του I. Οι γενικές λογικές επεκείνουν του π – θεσμούς που εισήχθησαν από του Fiadeiro και Sernada. Ανάλογη έννοια λογικής παραγωγής μ’ αυτή των π – θεσμών έχει προταθεί από τους Harper, Sannella και Tarleski με τη βοήθεια λογικών πλαισίων (Logical Frameworks ή LF) και θεσμών.

Αφηρημένες Αλγεβρικές Προδιαγραφές

Μια προδιαγραφή δεν είναι τίποτε περισσότερο παρά μία θεωρία που καθορίζει τη συμπεριφορά ενός συστήματος ή ενός προγράμματος με την ευρεία έννοια. Από τη σκοπιά της λογικής μια *θεωρία* αποτελείται από όλες τις προτάσεις ενός λογικού συστήματος που είναι αληθινές σε σχέση με αυτή τη συμπεριφορά. Περιγραφικά μια θεωρία πάνω σε ένα τυχαίο αλλά συγκεκριμένο θεσμό αποτελείται από μια χαρακτηριστική Σ και μια κλειστή (ως προς τη σημασιολογική απαγωγή) συλλογή από Σ – προτάσεις.

Ορισμός. Έστω I ένας συγκεκριμένος πλην όμως τυχαίος θεσμός. Τότε

1. Μια Σ – **παρουσίαση** είναι ένα ζεύγος ⟨Σ, Ε⟩, όπου Σ μια χαρακτηριστική και Ε μια συλλογή από Σ – προτάσεις.
2. Ένα Σ – μοντέλο Α **ικανοποιεί** μια παρουσίαση ⟨Σ, Ε⟩ εάν ικανοποιεί κάθε πρόταση του Ε. Αυτό θα το συμβολίζουμε με: Α |= Ε.

3. Έστω μια συλλογή Ε από Σ – προτάσεις, θα συμβολίζουμε με Ε* τη συλλογή όλων των Σ – μοντέλων που ικανοποιούν κάθε πρόταση της Ε.
4. Έστω μια συλλογή Μ από Σ – μοντέλα, έστω Μ* η συλλογή όλων των Σ – προτάσεων που ικανοποιούνται από κάθε μοντέλο που ανήκει στη Μ. Θα συμβολίζουμε με Μ* το ζεύγος (Σ, Μ*) και θα το ονομάζουμε **θεωρία** του Μ.
5. Η **κλειστότητα** μιας συλλογής Ε από Σ – προτάσεις είναι η Ε**, και θα τη συμβολίζουμε με Ε*.
6. Μια συλλογή Ε από Σ – προτάσεις θα ονομάζεται **κλειστή** εάν και μόνο εάν Ε=Ε*.
7. Μια Σ – **θεωρία** είναι μια παρουσίαση $\langle$Σ, Ε$\rangle$ έτσι ώστε η Ε να είναι κλειστή.
8. Μι α Σ – θεωρία που **παρουσιάζεται** μέσω του ζεύγους $\langle$Σ, Ε$\rangle$ είναι το ζεύγος $\langle$Σ,Ε*$\rangle$.
9. Μια Σ – πρόταση e **προκύπτει σημασιολογικά** από μια συλλογή Ε από Σ – προτάσεις, (θα γράφουμε Ε |= e), εάν και μόνο εάν $e \in E^*$.

Ορισμός. Εάν Τ και Τ΄ είναι θεωρίες, έστω οι $\langle$Σ, Ε$\rangle$ και $\langle$Σ΄, Ε΄$\rangle$, τότε ένας μορφισμός θεωριών από την Τ στην Τ΄ είναι ένας μορφισμός χαρακτηριστικών $\varphi : \Sigma \rightarrow \Sigma'$ έτσι ώστε $\varphi(e) \in E'$ για κάθε $e \in E$. Ένα τέτοιο μορφισμό θα το συμβολίζουμε με $\varphi : T \rightarrow T'$. Η **κατηγορία των θεωριών** έχει σαν αντικείμενα θεωρίες, και σαν μορφισμούς θεωριών, με τη σύνθεση και τον ταυτοτικό μορφισμό να είναι ορισμένος όπως στην περίπτωση των μορφισμών χαρακτηριστικών. Την κατηγορία αυτή τη συμβολίζουμε με **Th**.

Ας σημειωθεί ότι υπάρχει ένας συναρτητής που ξεχνάει (forgetful functor) *Sign* : **Th** → *Sign* που στέλνει κάθε ζεύγος $\langle$Σ, Ε$\rangle$ στο Σ, και στέλνει κάθε μορφισμό θεωριών φ στο φ σαν μορφισμό χαρακτηριστικών.

ΟΙ ΑΛΓΕΒΡΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ

Η καλύτερη μέθοδος για να εισάγει κανείς τις αλγεβρικές προδιαγραφές είναι μέσω των αφηρημένων τύπων δεδομένων (abstract data type). Ένας αφηρημένος τύπος δεδομένων αποτελείται άτυπα από μια συλλογή από τιμές (που λέμε ότι ανήκουν στο *σύνολο φορέας* (*carrier*)) και από ένα σύνολο τελεστών που δρουν πάνω σ' αυτές, είναι δηλαδή, αυτό που στη μαθηματική ορολογία ονομάζεται άλγεβρα (*algebra*).

Ο Parnas, ο Zilles και ο Guttag ήταν αυτοί που πρωτοσυνέθεσαν τους τύπους δεδομένων (data types) με τις άλγεβρες. Ο Zilles μάλιστα πρότεινε ένα νέο είδος άλγεβρας που την ονόμασε *άλγεβρα δεδομένων* (*data algebra*). Ο Guttag εισήγαγε ιδιαίτερα σημαντικές ιδέες όπως της *μη – αντιφατικότητας* και της *ικανής πληρότητας* που αργότερα οδήγησαν στη μελέτη των *τμημάτων με καταστάσεις* (*module with states*). Το ουσιαστικό ξεκίνημα των αλγεβρικών προδιαγραφών έγινε όμως από την ομάδα ADJ (μέλη της ή-ταν οι διακεκριμένοι επιστήμονες: Goguen, Thatcher, Wagner και Wright), με τον Goguen να παίζει το σημαντικότερο ρόλο.

Η μαθηματική θεωρία που υποστηρίζει την ανάπτυξη των αλγεβρικών προδιαγραφών είναι αυτή της καθολικής άλγεβρας (universal algebra) με πολλούς τύπους. Το λογικό σύστημα που συνδέεται μ' αυτήν είναι η εξισωτική λογική με πολλούς τύπους (many sorted equational logic). Η εξισωτική λογική διαθέτει ένα μόνο λογικό σύμβολο, αυτό της ισότητας $=$, οι δε «προτάσεις» της έχουν τη γενική μορφή $(\forall X)(t_1 = t_2)$, όπου t_1, t_2 είναι όροι με πολλούς τύπους. Από την εποχή της ADJ μέχρι σήμερα έχουν χρησιμοποιηθεί επίσης και διάφορες επεκτάσεις της εξισωτικής λογικής όπως η εξισωτική λογική με διατεταγμένους τύπους (order sorted equational logic) ή με περιορισμούς (constaints). Σήμερα τα πάντα γίνονται στο επίπεδο αφαίρεσης που μας προσφέρει η θεωρία και η κατηγοριακή λογική του R. Diaconescu .

Η βασική συνεισφορά της ομάδας ADJ έγκειται στη χρησιμοποίηση των αρχικών αλεβρών (initial algebras), των θεωριών του Lawverse και των κατηγοριοθεωρητικών εννοιών στη σημασιολογία του προγραμματισμού και στη θεωρία των προδιαγραφών. Είναι η ομάδα που γέννησε στην ουσία αυτές που λέγονται σήμερα **αλγεβρικές προδιαγραφές** (**algebraic specifications**) και τις μεταμόρφωσε σε ξεχωριστό κλάδο της πληροφορικής.

Οι πριν την ADJ προσεγγίσεις βασίζονταν στην ιδέα ότι ένας συγκεκριμένος τύπος δεδομένων είναι μια άλγεβρα με πολλούς τύπους (many sorted algebra). Αυτές οι προσεγγίσεις στους αφηρημένους τύπους δεδομένων προχωρούν και στο να δώσουν και το ποιές εξισώσεις πρέπει να ικανοποιούν οι τελεστές της άλγεβρας αλλά το ποιά μοντέλα είναι τα πιο κατάλληλα γι' αυτό. Η προσέγγιση της ADJ που βασίζεται στην έννοια της αρχικής άλγεβρας μας δίνει ένα συγκεκριμένο μοντέλο που ορίζεται με μοναδικό τρόπο. Πιο συγκεκριμένα, το σύνολο των εξισώσεων πάνω σε τελεστές στην πραγματικότητα ορίζει πλήρως τα δεδομένα. Δεν μας ενδιαφέρει δηλαδή το πώς αναπαρίστανται τα δεδομένα μια και μέσω της αρχικής άλγεβρας ορίζονται μοναδικά σε σχέση με κάθε δυνατή μετονομασία τους.

Με δεδομένη μια χαρακτηριστική (δηλαδή ένα σύνολο από μη λογικά σύμβολα) και ένα σύνολο από εξισώσεις που χρησιμοποιούν σύμβολα απ' αυτή, θα ονομάζουμε μια αναπαράσταση δεδομένων σαν την πιο **αντιπροσωπευτική** (**standard**) εάν και μόνο εάν ικανοποιούνται:

1. **Η ιδιότητα της καθολικής χρησιμότητας** (**No junk**): Η αναπαράσταση των δεδομένων δεν πρέπει να έχει στοιχεία που να μη χρησιμοποιούνται. Αυτό σημαίνει ότι κάθε στοιχείο των δεδομένων μπορεί να αναπαρασταθεί μόνο με τη βοήθεια των συμβόλων που ανήκουν στη χαρακτηριστική. Οι αναπαραστάσεις εκείνες που περιλαμβάνουν στοιχεία που δεν μπορούν να αντιστοιχηθούν με στοιχεία του τύπου δεδομένων λέγεται ότι περιλαμβάνουν περιττά στοιχεία (junk).

2. **Η ιδιότητα της σαφήνειας** (**No confusion**): Δύο στοιχεία της αναπαράστασης ονομάζονται ίσα εάν και μόνο εάν μπορούν να αποδειχθούν ίσα με τη βοήθεια των δοθέντων αξιωμάτων (με τη χρησιμοποίηση των κανόνων λογικής απαγωγής της εξισωτικής λογικής). Δύο στοιχεία τα οποία είναι ίσα αλλά δεν μπορούν να αποδειχθούν σαν ίσα από τις εξισώσεις ονομάζονται συγκεκριμένα (confused).

Οι δύο αυτές ιδιότητες από κοινού είναι ισοδύναμες με τη συνθήκη του **μοναδικού ομομορφισμού** (**unique homomorphism**) χαρακτηρίζει την **αρχικότητα**. Ένα από τα πλεονεκτήματα της χρησιμοποίησης της αρχικότητας για την αναπαράσταση δεδομένων είναι ότι δεν οδηγεί σε ανεπιθύμητες ιδιότητες που χαρακτηρίζουν πολλά από τα άλλα μοντέλα αναπαράστασης των δεδομένων. Αυτή η προσέγγιση έχει αποδειχθεί ένα κατάλληλο πλαίσιο για τον ορισμό κριτηρίων ορθότητας, συνέπειας και πληρότητας των προδιαγραφών. Η μοναδική (ως προς ισομορφισμούς) αρχική άλγεβρα χρησιμοποιείται σαν την πιο **χαρακτηριστική αναπαράσταση των τύπων δεδομένων** με το σκεπτικό ότι περιλαμβάνει μόνο ότι απαιτεί η προδιαγραφή και τίποτε περισσότερο, δηλαδή συνιστά ότι πιο πιστό υπάρχει.

Η γλώσσα OBJ

Η OBJ αποτελεί την πιο χαρακτηριστική από τις γλώσσες που χρησιμοποιούνται για αλγεβρικές προδιαγραφές. Είναι μια γλώσσα με συναρτησιακό χαρακτήρα που όμως βασίζεται στην εξισωτική λογική με διατεταγμένους τύπους (ordered sorted equational logic). Μας επιτρέπει τόσο το να προδιαγράψουμε κάτι με τη βοήθεια των αλγεβρικών προδιαγραφών όσο και να αποδείξουμε διάφορες ιδιότητες γι' αυτές. Η γλώσσα αυτή προτάθηκε από τον Goguen αναπτύχθηκε δε στα πανεπιστήμια UCLA και Stanford. Η σημασιολογική βάση της γλώσσας είναι ιδιαίτερα πλούσια μια και επιτρέπει ένα δηλωτικό ύφος προγραμματισμού (program verfication), και υποστηρίζει τη χρήση αποδεικτών θεωρημάτων (theorem prover). Αυτή η βάση επιτρέπει σε προδιαγραφές να γραφούν με τη μορφή προγραμμάτων δηλωτικού τύπου που να είναι στην ουσία αλγεβρικά. Ο εκτελέσιμος κώδικας της OBJ αποτελείται από εξισώσεις που ερμηνεύονται σαν κανόνες αναγραφής (rewriting rules). Το είδος αναγραφής που υποστηρίζει η OBJ ακολουθεί τη μεταβατική, την αντιμεταθετική και την ταυτοτική ιδιότητα.

Η OBJ βασίζεται στις αρχές του παραμετρικοποιημένου προγραμματισμού (parametrized programming). Έτσι μπορεί να αξιοποιηθεί στην παραγωγή, το σχεδιασμό, την επαναχρησιμοποίηση και τη συντήρηση του λογισμικού. Χρησιμοποιεί δύο είδη τμημάτων (modules):

- **αντικείμενα** (**objects**) – κυρίως για τον ορισμό των αφηρημένων τύπων δεδομένων,
- **θεωρίες** (**theories**) – που έχουν τόσο σημασιολογικές όσο και συντακτικές ιδιότητε.

Η OBJ υποστηρίζει συνθέσεις τμημάτων και δημιουργία νέων μεγαλύτερων τμημάτων από τη σύνθεση μικρότερων. Διάφορες εκφράσεις με τη χρήση τμημάτων (module expresions) μπορούν να χρησιμοποιηθούν για περίπλοκους συνδυασμούς τμημάτων που έχουν ήδη οριστεί, για το μετασχηματισμό τμημάτων, για την αποτίμηση ενός τμήματος (module evaluation), και γενικά για την κατασκευή ενός συστήματος λογισμικού από διάφορα μικρότερα μέρη. Διάφορα τμήματα ακόμα να εισάγουν άλλα τμήματα.

Επίσης τα τμήματα που ορίζονται στην OBJ μπορούν να παραμετρικοποιηθούν έχοντας άλλα τμήματα σαν παραμέτρους. Οι διάφοροι παράμετροι λαμβάνουν συγκεκριμένες τιμές μέσα από κατασκευές που ονομάζονται **οπτικές γωνίες** (**views**). Αυτές μαζί με τα αντικείμενα και τις θεωρίες αποτελούν τον κορμό της γλώσσας OBJ.

Τα αξιώματα ενός τμήματος της OBJ είναι *εξισώσεις* που ικανοποιούν ή όχι συγκεκριμένα αξιώματα υπό συνθήκες ή όχι (conditional or unconditional axioms).

Η γλώσσα προδιαγραφών Clear

Η προδιαγραφή (specification) ενός προβλήματος αποτελεί βασικό στοιχείο για το σωστό σχεδιασμό και υλοποίηση του αλγορίθμου επίλυσής του. Επίσης, παίζει ιδιαίτερα σημαντικό ρόλο στην απόδειξη της ορθότητας του αντίστοιχου προγράμματος. Υπάρχουν ειδικά σχεδιασμένες γλώσσες προδιαγραφών που έχουν γενική χρήση όπως για παράδειγμα η Clear, η Z και η VDM. Άλλες έχουν εξειδικευμένη χρήση όπως για Παράδειγμα η ESTELLE, η LOTOS και η SDL που χρησιμοποιούνται ειδικά στο χώρο των δικτύων και των τηλεπικοινωνιών.

Ανάμεσα στις γλώσσες προδιαγραφών η Clear είναι μια από τις πιο αναπτυγμένες και καταξιωμένες. Η γλώσσα αυτή υποστηρίζει τον ορισμό προδιαγραφών με τη βοήθεια δομημένων θεωριών σε τμηματοποιημένη μορφή. Οι θεωρίες αυτές ορίζονται με τη βοήθεια της θεωρίας θεσμών για συγκεκριμένα άλλα τυχαία λογικά (ή πιο γενικά συμβολικά) συστήματα. Η βασική φιλοσοφία της Clear συνοψίζεται στο ότι οι προδιαγραφές μεγάλων συστημάτων μπορούν να συντεθούν από πολλές μικρότερες προδιαγραφές που συνήθως είναι ήδη έτοιμες, βρίσκονται σε βιβλιοθήκες και μπορούν να επαναχρησιμοποιηθούν. Οι θεωρίες αυτές είναι οργανωμένες σε μια κατηγορία μέσα στην οποία μπορούν να εφαρμοστούν οι βασικές αρχές της γενικευμένης θεωρίας συστημάτων. Οι βασικοί τελεστές της Clear είναι:

- *η σύνθεση* (*sum*) *προδιαγραφών*. Βασίζεται στη σύνθεση θεωριών μέσα από κατάλληλα συν – όρια (colimits) της κατηγορίας των θεωριών

- *ο εμπλουτισμός* (*enrich*) *προδιαγραφών*. Στην ουσία γίνεται εμπλουτισμός των προδιαγραφών με νέα δεδομένα. Οι θεωρίες δηλαδή αποκτούν νέα αξιώματα,

- *ο μετασχηματισμός* (*derive*) *μιας προδιαγραφής σε μια άλλη*. Οι θεωρίες δηλαδή α-ποκτούν τροποποιημένα αξιώματα με βάση συγκεκριμένους κανόνες μετασχηματι-σμού.

Οι τελεστές αυτού δρούν πάνω σε ήδη υπάρχουσες προδιαγραφές και μας δίνουν νέες. Η Clear υποστηρίζει επίσης τις παραμετρικοποιημένες προδιαγραφές αλλά και δίνει κατάλληλους μηχανισμούς για τη συγκεκριμενοποίησή τους (instantiation).

Η εξισωτική λογική

Η εξισωτική λογική μπορεί να χρησιμοποιηθεί για την προδιαγραφή και επαλήθευση δηλωτικών προγραμμάτων. Επίσης, η εξισωτική απαγωγή είναι σημαντικά απλούστερη συγκριτικά με την πρωτοβάθμια απαγωγή και έχει πολλά πλεονεκτήματα σχετικά με τις ενδεχόμενες εφαρμογές της στην πληροφορική. Η εξισωτική λογική χρησιμοποιείται ήδη για πολλά προβλήματα στις προδιαγραφές υλικού και λογισμικού. Η εξισωτική λο-γική μπορεί επίσης να χρησιμοποιηθεί σαν μια μεταγλώσσα για την περιγραφή άλλων λογικών συστημάτων. Κάτι τέτοιο έχει γίνει ήδη για το λογικό πλαίσιο 2OBJ system, ένα μετα – λογικό σύστημα που υποστηρίζει απαγωγές σε όλα τα λογικά συστήματα που μπορούν να κωδικοποιηθούν στην εξισωτική λογική.

Η κλασική εξισωτική λογική με τύπους

Έστω I ένα σύνολο (δεικτών), τότε ένα I – **σύνολο με δείκτες** A επισυνάπτει από ένα σύνολο A_i σε κάθε **δείκτη** i του I. Έστω ότι A και B είναι δύο I – σύνολα με δείκτες, τότε καλούμε **απεικόνιση** ή **μορφισμό** μεταξύ των δύο I – συνόλων με δείκτες, (και συμβολίζουμε με $f : A \to B$), την I – οικογένεια συναρτήσεων $f_i : A_i \to B_i$ για κάθε $i \in I$. Η σύνθεση μεταξύ δύο I – οικογενειών συναρτήσεων ορίζεται με τον προφανή τρόπο $(f\,;\,g)_i = f_i\,;\,g_i$ για κάθε δείκτη $i \in I$. Έτσι, ορίζουμε την κατηγορία $\mathbf{Set}_I$ των I – συνόλων με δείκτες. Θα συμβολίσουμε με $A = \langle A_i \mid i \in I)$ ένα I – σύνολο με δείκτες και στοιχεία A_i και με $f = \langle f_i \mid i \in I)$ I – οικογένεια συναρτήσεων $A \to B$ μεταξύ συνόλων I – συνό-λων με δείκτες όπου $f_i : A_i \to B_i$. Όλες οι βασικές έννοιες της θεωρίας συνόλων επε-κτείνονται άμεσα στα I – σύνολα με δείκτες. Για παράδειγμα, $A \subseteq B = \langle A_i \subseteq B_i \mid i \in I \rangle$, $A \cap B = \langle A_i \cap B_i \mid i \in I \rangle$, κ.τ.λ.

Τα μη λογικά σύμβολα της εξισωτικής λογικής με τύπους

Τα **μη λογικά σύμβολα** της εξισωτικής λογικής με τύπους προέρχονται από ένα ζεύγος $\langle S, \Sigma \rangle$, όπου S είναι ένα σύνολο (από **τύπου**) και Σ είναι μια $S^* \times S$ οικογένεια συνόλων (**τελεστών**). Τα ζεύγη αυτά για λόγους συντομίας θα τα καλούμε από εδώ και στο εξής (**εξισωτική**) **χαρακτηριστική**. Για λόγους απλότητας πολλές φορές γράφουμε Σ αντί για $\langle S, \Sigma\rangle$. Τα σύμβολα σ του $\Sigma_{u,s}$ λέμε ότί έχουν **αρχή** u, **τέλος** s, και **ίχνος** u,s. Θα τα συμβολίζουμε δε με $\sigma : u \to s$.

Ένας μορφισμός φ μεταξύ των χαρακτηριστικών $\langle S, \Sigma\rangle$ και $\langle S', \Sigma'\rangle$ είναι εάν ζεύγος $\langle f, g\rangle$ αποτελούμενο από μία απεικόνιση $f : S \to S'$ μεταξύ των συνόλων των τύπων και μια $S^* \times S$ – οικογένεια μορφισμών $g_{u,s} : \Sigma_{u,s} \to \Sigma'_{f^*(u),f(s)}$ μεταξύ συμβόλων τελεστών, όπου $f^* : S^* \to (S')$ είναι η επέκταση του f σε αλυσίδες συμβόλων[1]. Συχνά χρησιμοποιούμε τους συμβολισμούς $\varphi(s)$ αντί για $f(s)$, $\varphi(u)$ αντί για $f^*(u)$, και $\varphi(\sigma)$ αντί για $g_{u,s}(\sigma)$ όταν $\sigma \in \Sigma_{u,s}$.

Η **κατηγορία των χαρακτηριστικών**, έστω **Sig**, έχει τις χαρακτηριστικές σαν αντικείμενα, και τους μορφισμούς μεταξύ χαρακτηριστικών σαν μορφισμούς.

Σ – άλγεβρες

Έστω μια χαρακτηριστική Σ, τότε μια Σ – άλγεβρα ερμηνεύει κάθε τύπο σαν ένα σύνολο και κάθε τελεστή σαν μια συνάρτηση. Οι Σ – άλγεβρες αντιστοιχούν στην προγραμματιστική έννοια των διακριτών τύπων δεδομένων (discrete data types).

Έστω μια χαρακτηριστική $\langle S, \Sigma\rangle$. Τότε μια Σ – **άλγεβρα** A αποτελείται από μια S – οικογένεια συνόλων $| A | = \langle A_s \mid s \in S\rangle$ (από **φορείς** του A), μαζί με μια $S^* \times S$ – οικογένεια α συναρτήσεων $\alpha_{u,s}\ \Sigma_{u,s} \to [A_u \to A_s]$ για $u \in S^*$ και $s \in S$, όπου $A_{s1\ldots sn} = A_{s1} \times\ldots\times \alpha_{sn}$ και $[A \to b]$ δηλώνει το σύνολο όλων των συναρτήσεων από το A στο B.

Για $u = s1\ldots sn$, για $\sigma \in \Sigma_{u,s}$ και για $(\alpha1,\ldots,\alpha n) \in A_u$ θα γράφουμε $(\alpha1,\ldots,\alpha n)$ αντί για $\alpha_{u,s}(\sigma)(\alpha1,\ldots,\alpha n)$.

Έστω μια χαρακτηριστική Σ, τότε ένας Σ – ομομορφισμός από μια Σ – άλγεβρα $\langle A, \alpha\rangle$ σε μια άλλη Σ – άλγεβρα $\langle A', \alpha'\rangle$ είναι μια S – οικογένεια $f : A \to A'$ έτσι ώστε για κάθε $\sigma \in \Sigma_{u,s}$ και κάθε $\alpha = (\alpha1,\ldots,\alpha n) \in A_u$ να ισχύει η εξής **συνθήκη ομομορφισμού**

$$f_s(\alpha(\sigma)(\alpha1,\ldots,\alpha n)) = \alpha'(\sigma)(f_{s1}(\alpha1),\ldots,f_{sn}(\alpha n))$$

δηλαδή

[1] Το $f^* : S^* \to (S')$ ορίζεται ως ακολούθως $f^*([\ ,\ ]) = [\ ,\ ]$, όπου $[\ ,\ ]$ συμβολίζει την κενή σελίδα συμβόλων και $f^*(\omega s) = f^*(\omega)f(s)$, για κάθε και $s \in S$.

Η **κατηγορία $\mathbf{Alg}_\Sigma$** των Σ – **αλγεβρών** έχει τις Σ – άλγεβρες σαν αντικείμενα και τους Σ – ομομορφισμούς σαν μορφισμούς.

1. $\Sigma_{X,s} \subseteq T\Sigma_{,s}$ και

2. $\Sigma_{s1\ldots sn,s}$ και $ti \in T_{\Sigma,si}$ έπεται ότι η αλυσίδα συμβόλων $\sigma(t1,\ldots,tn)$ ανήκει στην $T_{\Sigma,s}$.

Ο συναρτητής **Alg**: Sig $\rightarrow$ **Cat** ^{op} στέλνει κάθε χαρακτηριστική Σ την κατηγορία **Alg**(Σ) όλων των Σ – αλγεβρών, και κάθε μορφισμό $\varphi = \langle f : S \rightarrow S', g : \Sigma \rightarrow \Sigma' \rangle$ στο συναρτητη **Alg**(φ) : $\mathbf{Alg}_{\Sigma'} \rightarrow \mathbf{Alg}_\Sigma$ στέλνοντας

1. κάθε Σ – άλγεβρα $\langle A', \alpha' \rangle$ στη Σ – άλγεβρα $\langle A, \alpha \rangle$ με $A_s = A'_{f(s)}$, $\alpha = g\,;\,\alpha'$ και

2. κάθε Σ' – ομομορφισμό $h' : A' \rightarrow B'$ στον Σ – ομομορφισμό

$$\mathbf{Alg}(\varphi)(h') = h : \mathbf{Alg}(\varphi)(A') \rightarrow \mathbf{Alg}(\varphi)(B') \quad \text{οριζόμενο μέσω } h_s = h'_{f(s)}.$$

Πολλές φορές συμβολίζουμε με $\varphi(A')$ την **Alg**(φ)(A') και με $\varphi(h')$ τον **Alg**(φ)(h').

Έστω S το σύνολο των τύπων μιας χαρακτηριστικής Σ, τότε υπάρχει ένας **συναρτητής** που **ξεχνάει** $U : \mathbf{Alg}_\Sigma \rightarrow \mathbf{Set}_S$ ο οποίος στέλνει κάθε άλγεβρα στην S – οικογένεια φορέων της, και κάθε Σ – ομομορφισμός στη σχετιζόμενη S – οικογένεια απεικονίσεων.

Για κάθε S – σύνολο με δείκτες X, μπορούμε να ορίσουμε μια **ελεύθερη άλγεβρα** ή μια **άλγεβρα όρων**, που θα συμβολίζουμε με $T_\Sigma(X)$, με $|T_\Sigma(X)|_s$ να αποτελείται από όλους τους Σ – όρους τύπου s χρησιμοποιώντας μεταβλητές από το X, δηλαδή κάθε φορέας $(T_\Sigma(X))_s$, αποτελείται από όλους τους όρους τύπου s με μεταβλητές από το X και κατασκευάζονται με τη βοήθεια τελεστών από το Σ. Το S – σύνολο με δείκτες $T_\Sigma(X)$, σχηματίζει μια Σ – άλγεβρα. Ορίζουμε τους φορείς $(T_\Sigma)_s$ ως το ελάχιστο σύνολο από αλυσίδες συμβόλων έτσι ώστε

3. $\Sigma_{X,s} \subseteq T\Sigma_{,s}$ και

4. $\Sigma_{s1\ldots sn,s}$ και $ti \in T_{\Sigma,si}$ έπεται ότι η αλυσίδα συμβόλων $\sigma(t1,\ldots,tn)$ ανήκει στην $T_{\Sigma,s}$.

Ορίζουμε τη συνάρτηση α που επιτρέπει στην $T_\Sigma(X)$ να σχηματίζει μια Σ – άλγεβρα ως ακολούθως:

1. για κάθε $\alpha \in \Sigma_{X,s}$ έστω $\alpha(\sigma)$ η αλυσίδα που περιλαμβάνει μόνο το σύμβολο σ μήκους 1 και ανήκει στο $T_{\Sigma,s}$ και

2. για κάθε $\sigma \in \Sigma_{s1\ldots sn,s}$ και $ti \in T_{\Sigma,si}$ έστω ότι $\alpha(\sigma)(t1,\ldots,tn)$ είναι η αλυσίδα συμβόλων $\sigma(t1,\ldots,tn)$ στο $T_{\Sigma,s}$.

Ορίζουμε $\Sigma(X)$ να είναι η χαρακτηριστική που ορίζεται ως $(\Sigma(X))_{[\,,\,],s} = \Sigma_{[\,,\,],s} \cup X_s$ και $(\Sigma(X))_{u,s} = \Sigma_{u,s}$ εάν $u \neq [\,,\,]$. Τότε, σαν $T_\Sigma(X)$ ορίζουμε τη Σ – άλγεβρα $T_{\Sigma(X)}$. Η Σ – άλγεβρα $T_{\Sigma(X)}$ είναι ελεύθερη όπως προκύπτει από το παρακάτω θεώρημα.

Θεώρημα. Έστω ότι ο μορφισμός $i_X : X \to U(T_\Sigma(X))$ δηλώνει το περιέχεσθαι του X στο $U(T_\Sigma(X))$[2]. Τότε, ισχύει η εξής καθολική (universal) ιδιότητα για κάθε Σ – άλγεβρα B, κάθε S – οικογένεια συναρτήσεων $f : X \to B$, (που καλείται **συνάρτηση επισύναψης** (**assignment**), επεκτείνεται μοναδικά σε ένα Σ – ομομορφισμό $f^\# : T_\Sigma(X) \to B$ έτσι ώστε $i_X : U(f^\#) = f$.

Μια πρώτη ιδέα της απόδειξης μπορεί να προέλθει από το επόμενο σχήμα:

Όταν $X = \emptyset$, τότε υπάρχει ένας μοναδικός Σ – ομοιομορφισμός από την T_Σ σε κάθε άλλη Σ – άλγεβρα. Συνεπώς, η T_Σ είναι **αρχική** (**initial**) Σ – άλγεβρα.

Εξισώσεις και σχέση ικανοποίησης

Ορίζουμε τώρα την έννοια της εξίσωσης και της ικανοποίησης μιας εξίσωσης από μία Σ – άλγεβρα. Καταρχήν, έστω ένα άπειρο σύνολο $\mathcal{X}$ από μεταβλητές. Τότε μια **συνάρτηση επισύναψης τύπων** (**sort assignment**) είναι μια μερική συνάρτηση $X: \mathcal{X} \to S$ όπου S είναι το ένα σύνολο τύπων. Έστω, επίσης ότι το $\mathcal{X}$ δηλώνει το S – σύνολο με δείκτες με $X_s = \{x \in \mathcal{X} \mid X(x) = s\}$.

Ορισμός. Μια Σ – **εξίσωση** e είναι μια τριάδα $\langle X, t1, t2\rangle$, όπου X είναι μια συνάρτηση επισύναψης τύπων $\mathcal{X} \to S$ με S το σύνολο των τύπων του Σ, και $t1, t2 \in |T_\Sigma(X)|$, είναι ο έχοντας τον ίδιο τελικό τύπο $s \in S$. Μια τέτοια εξίσωση τη γράφουμε πολλές φορές $(\forall X)t1 = t2$.

[2] Από εδώ και στο εξής το U θα παραλείπεται.

Ορισμός. Μία **Σ – άλγεβρα ικανοποιεί μια Σ – εξίσωση** $(\forall X)t1 = t2$ εάν και μόνο εάν $\alpha^{\#}(t1) = \alpha^{\#}(t2)$ για κάθε συνάρτηση επισύναψης $\alpha : X \rightarrow |A|$. Σ' αυτή την περίπτωση γράφουμε $A \models e$.

Ορισμός. Έστω $\varphi : \Sigma \rightarrow \Sigma'$ ένας μορφισμός χαρακτηριστικών $\langle f : S \rightarrow S' \rangle$ επίσης έστω $X : \mathrm{X} \rightarrow S$ μια συνάρτηση επισύναψης τύπων, και X' η συνάρτηση επισύναψης τύπων $X ; f$. Στη συνέχεια ορίζουμε την S – οικογένεια απεικονίσεων $\varphi^{\sim} : |T_{\Sigma}(X)|$ $|\varphi(T_{\Sigma'}(X'))|$ ως εξής: Καταρχήν $X \subseteq |(T_{\Sigma'}(X'))|$ αφού εάν $x \in X_s$ τότε $x \in X'_{f(s)}$ $X'_{f(s)} \subseteq |T_{\Sigma'}(X')|_{f(s)} = |\varphi(T_{\Sigma'}(X'))|_s$. Έστω $j : X \rightarrow |\varphi(T_{\Sigma'}(X'))|$ δηλώνει το περιέχεσθαι του X στο $|\varphi(T_{\Sigma'}(X'))|$. Τότε ο j έχει μια μοναδική επέκταση σαν ένα Σ – ομομορφισμό $j^{\#} : T_{\Sigma}(X) \rightarrow |\varphi(T_{\Sigma'}(X'))$, και έτσι ορίζουμε $\varphi^{\sim} = |j^{\#}|$.

Ορισμός. Ο **συναρτητής** Eqn : **Sig** → **Set** στέλνει κάθε χαρακτηριστική Σ σύνολο Eqn(Σ) όλων των Σ – εξισώσεων και κάθε ζεύγος $\varphi = \langle f, g \rangle : \Sigma \rightarrow \Sigma'$ στη συνάρτηση $\mathrm{Eqn}(\varphi) : \mathrm{Eqn}(\Sigma) \rightarrow \mathrm{Eqn}(\Sigma')$ που ορίζεται ως εξής:

$$\mathrm{Eqn}(\varphi)(\langle X, t1, t2 \rangle) = \langle X ; f, \varphi^{\sim}(t1), \varphi^{\sim}(t2) \rangle .$$

Αντί για Eqn(φ)(e) θα γράφουμε πολλές φορές απλά φ(e).

Πρόταση. Έστω $\varphi : \Sigma \rightarrow \Sigma'$ ένας μορφισμός χαρακτηριστικών, e μια Σ – εξίσωση, A′ μια Σ′– άλγεβρα, τότε ισχύει η εξής **συνθήκη ικανοποίησης**:

$$A' \models \varphi(e) \text{ εάν και μόνο εάν } \varphi(A') \models e.$$

Η λογική προτάσεων του Horn (Horn Clause Logic)

Η **λογική προτάσεων του Horn (Horn clause logic συνοπτικά (HOL))** αποτελεί τη βάση για τη γλώσσα λογικού προγραμματισμού Prolog.

Ορισμός. Μια **πρωτοβάθμια χαρακτηριστική** Ω αποτελείται από την εξής τριάδα $\langle S, \Sigma, \Pi \rangle$, όπου

1. S είναι ένα σύνολο (από **τύπους**),
2. Σ είναι μια $S^* \times S$ – οικογένεια συνόλων (από **συναρτησιακά σύμβολα**), και
3. Π είναι μια S^* – οικογένεια συνόλων (από **κατηγορηματικά** ή **σχεσικά σύμβολα**).

Ένας μορφισμός **πρωτοβάθμιων χαρακτηριστικών**, από το Ω στο Ω′, είναι μια τριάδα $\langle \varphi_1, \varphi_2, \varphi_3 \rangle$, όπου

1. $\varphi_1 : S \to S'$ είναι μια συνάρτηση,
2. $\varphi_2 : \Sigma \to \Sigma'$ είναι μια $S^* \times S$ – οικογένεια συναρτήσεων
$$(\varphi_2)_{u,s} : \Sigma_{u,s} \to \Sigma'_{\varphi_1^*(u),\varphi_1(s)}$$
3. $\varphi_3 : \Pi \to \Pi'$ είναι μια S^* – οικογένεια συναρτήσεων
$$(\varphi_3)_u : \Pi_u \to \Pi'_{\varphi_1^*(u)} .$$

Έστω **FoSig** ή **κατηγορία των πρωτοβαθμίων χαρακτηριστικών** με τις πρωτοβάθμιες χαρακτηριστικές σαν αντικείμενα, και μορφισμούς πρωτοβάθμιων χαρακτηριστικών σαν μορφισμούς.

Ορισμός. Έστω Ω μια πρωτοβάθμια χαρακτηριστική (first order signature), τότε Ω – μοντέλο (ή μια Ω – **δομή**) A αποτελείται από:

1. μια S – οικογένεια $|A|$ αποτελούμενη από μη κενά σύνολα $\langle A_s \mid s \in S\rangle$, όπου A_s καλείται **φορέας** τύπου s, και
2. μια $S^* \times S$ – οικογένεια α από συναρτήσεις $\alpha_{u,s} : \Sigma_{u,s} \to [A_u \to A_s]$ που επισυνάπτει συναρτήσεις σε κάθε συναρτησιακό σύμβολο, μια S^*– οικογένεια β από συναρτήσεις $\beta_u : \Pi_u \to Pow(A_u)$ που επισυνάπτουν μια σχέση σε κάθε κατηγορηματικό σύμβολο όπου Pow(A) δηλώνει το σύνολο όλων των υποσυνόλων του A.

Ένα *μορφισμός πρωτοβάθμιων μοντέλων* $h : M \to M'$ μεταξύ (Σ, Π) μοντέλων M και M′ είναι ένας Σ – ομομορφισμός έτσι ώστε για κάθε κατηγορηματικό σύμβολο $pi \in Pi_{s_1 \ldots s_n}$, εάν $m \in \pi_M$ τότε $h(m) \in \pi_{M'}$.

Έστω **FoMod** η κατηγορία πρωτοβάθμιων μοντέλων με πρωτοβάθμια μοντέλα σαν αντικείμενα και μορφισμούς πρωτοβάθμιων μοντέλων σαν μορφισμούς. Η **FoMod** επεκτείνεται σε ένα συναρτητή $\mathbf{FoSig} \to \mathrm{Cat}^{op}$ ως ακολούθως: Έστω ένας μορφισμός μεταξύ πρωτοβάθμιων χαρακτηριστικών $\varphi : \Omega \to \Omega'$, τότε ορίζουμε το συναρτητή ως εξής
$$\mathbf{FoMod}(\varphi) : \mathbf{FoMod}(\Omega') \to \mathbf{FoMod}(\Omega)$$
να στέλνει καταρχήν τα $A' \in |\mathbf{FoMod}(\Omega')|$ στο $A = \varphi A'$ όπου

1. $A_s = A'_{s'}$ για sS με $s' = \varphi_1(s)'$,
2. $\alpha_{u,s}(\sigma) = \alpha'_{u',s'}((\varphi_2)_{u,s}(\sigma))$ για $u \in S^*$, $s \in S$ και $\sigma \in \Sigma_{u,s}$ όπου $u' = \varphi_1^*(u)$ και $s' = \varphi_1(s)$,
3. $\beta_u(\pi) = \beta'_{u'}((\varphi_3)_{u,s}(\sigma))$ για $u \in S^*$ και $\pi \in \Pi_u$ με u′ όπως παραπάνω,

και στη συνέχεια, να στέλνει κάθε $f' : A' \to B'$ από το **FoMod**(Ω') στο $f = \varphi f' : A \to B$ του **FoMod**(Ω), όπου $A = \varphi A'$ και $B = \varphi B'$, οριζόμενο ως $f_s = f'_{s'}$ όπου $s' = \varphi_1(s)$.

Οι προτάσεις κατά Horn ορίζονται πάνω σε μια πρωτοβάθμια χαρακτηριστική Ω. Έστω $\mathcal{X}$ ένα άπειρο σύνολο μεταβλητών, και έστω $X : \mathcal{X} \to S$ μια μερική συνάρτηση που επισυνάπτει τύπους στις μεταβλητές, το X επίσης δηλώνει το S – σύνολο με $X_s = \{x \in \mathcal{X} \mid X(x) = s\}$. Επίσης ορίζουμε την S – οικογένεια $\mathrm{term}_X(\Omega)$ αποτελούμενη από (Ω, **X**) – **όρους** πού ανήκουν στους φορείς του $T_\Sigma(X)$ (την ελεύθερη Σ – άλγεβρα με γεννήτορες X). Ορίζουμε τώρα το σύνολο term(Ω) σαν τη ξεχωριστή (disjoint) ένωση όλων των $\mathrm{term}_X(\Omega)$.

Ορισμός. Μία καλώς ορισμένη (Ω,X) – **πρόταση του Horn** (**Horn – clause**) έχει τη μορφή:

$$\pi(t1,\ldots,tn) \mid \pi \in \Pi_u \text{ όπου } u = s1\ldots sn \text{ και } ti\,\mathrm{term}_X(\Omega)_{si}.$$

Έστω HCL(Ω) το σύνολο όλων των Ω – προτάσεων του Horn, δηλαδή η διακριτή ένωση των συνόλων $\mathrm{HCL}_X(\Omega)$ από κλειστές (Ω, X) – προτάσεις.

Τώρα ορίζουμε τη σχέση ικανοποίησης. Έστω A ένα πρωτοβάθμιο μοντέλο, και έστω $\mathrm{Asgn}_X(A)$ το σύνολο όλων των **συναρτήσεων επισύναψης** τιμών του A σε μεταβλητές X, δηλαδή είναι το σύνολο $[X \to A]$, όλων των S – οικογενειών συναρτήσεων $f : X \to A$.

Ορισμός. Έστω P μια πρόταση του Horn, ορίζουμε σαν $\mathrm{Asgn}_X(A, P)$, το σύνολο των συναρτήσεων επισύναψης στο A για το οποίο η P είναι αληθινή, επαγωγικά ως εξής:

- εάν $P = \pi(t1,\ldots,tn)$ τότε $f \in \mathrm{Asgn}_X(A, P)$ iff $(f^{\#}(t1),\ldots,f^{\#}(tn)) \in \beta(\pi)$, όπου $f^{\#}(t)$ δηλώνει την τίμηση (evaluation) του Σ – όρου t στην Σ – άλγεβρα – μέρος του A όπως ακριβώς στην περίπτωση της κλασικής εξισωτικής λογικής με πολλούς τύπους.

Ένα μοντέλο A ικανοποιεί μια πρόταση P του Horn με μεταβλητές από το X εάν μόνο εάν $\mathrm{Asgn}_X(A, P) = \mathrm{Asgn}_X(A, P)$ (θα το γράφουμε $A \models P$).

ΣΤΟΙΧΕΙΑ ΤΗΣ ΘΕΩΡΙΑΣ ΤΩΝ ΠΟΛΥΕΙΔΩΝ ΑΛΓΕΒΡΩΝ

Μια άλγεβρα Birkhoff αποτελείται από ένα σύνολο (μη κενό), που καλείται φορέας της άλγεβρας και από μία οικογένεια από συναρτήσεις ορισμένες πάνω σε καρτεσιανά γινόμενα του φορέα και τιμές στο φορέα. Πιο συγκεκριμένα:

Ορισμός. Μία άλγεβρα κατά Birkhoff είναι κάθε ζεύγος (A, B), όπου A είναι ένα μη κενό σύνολο και B είναι μία οικογένεια συναρτήσεων της μορφής

$$f : A \times A \times \ldots \times A \to A.$$

Πολλές από τις πιο γνωστές αλγεβρικές δομές ικανοποιούν τον παραπάνω ορισμό (π.χ. οι ομάδες, τα σώματα, οι δακτύλιοι, οι ημιομάδες). Από την άλλη μεριά όμως είναι φανερό ότι άλλες αλγεβρικές δομές όπως για παράδειγμα τα αυτόματα (ακολουθιακές μηχανές) ή τα modules ("διανυσματικοί χώροι πάνω σε δακτύλιου") δεν είναι άλγεβρες.

Μερικοί προκαταρκτικοί ορισμοί

Έστω S ένα μη κενό σύνολο (το *σύνολο των τύπων*). Ονομάζουμε τότε S – *σύνολο* κάθε οικογένεια συνόλων της μορφής $\{A_s \; s \in S\}$, όπου σε κάθε στοιχείο s του S αντιστοιχεί και ένα σύνολο A.

Παράδειγμα. Έστω S = {Nat, Real}, το Nat είναι συντομογραφία για το Natural Numbers (Φυσικοί Αριθμοί) και το Real για το Real Numbers (Πραγματικοί Αριθμοί) (η πιο συνήθης περίπτωση στην καθολική άλγεβρα είναι το S να αποτελείται από ονόματα ή συντμήσεις ονομάτων γι' αυτό και πολλές φορές ονομάζεται και *σύνολο ονομάτων*) και

A_{Nat} = **N** (το σύνολο των φυσικών αριθμών),

B_{Real} = **R** (το σύνολο των πραγματικών αριθμών).

Ορισμός. Μια S – συνάρτηση $F : A \to B$ μεταξύ δύο S – συνόλων A και B, είναι μια οικογένεια $\{f_s \mid s \in S\}$ από συναρτήσεις της μορφής $f_s : A_s \to B_s$.

Δοθέντων δύο S – συναρτήσεων $F : A \to B$ και $g : B \to C$ ορίζουμε σαν *σύνθεσή* τους την οικογένεια $\{f_s \,;\, g_s \mid s \in S\}$ όπου $f_s \,;\, g_s$ είναι η συνήθης σύνθεση των συναρτήσεων f $_s$ και g $_s$. Η *ταυτοτική* S – συνάρτηση $1_A : A \to A$ ορίζεται σαν ένα σύνολο $\{1_{A_s} \mid s \in S\}$ από ταυτοτικές συναρτήσεις. Μία S – συνάρτηση $f : A \to B$ λέγεται (1 – 1) (αντ. επί) εάν κάθε $f_s : A_s \to B_s$ είναι (1 – 1) (αντ. επί).

Ορισμός. Καλούμε S – χαρακτηριστική (S – sorted signature) Σ κάθε S × S* – οικογένεια $\{\Sigma_{<w,s>} \mid w \in S^*, s \in S\}$ όπου S* είναι το σύνολο που αποτελείται από όλες τις (πεπερασμένες) λίστες στοιχείων του S (δηλαδή $w = s_1 \ldots s_n$, όπου $s_1, \ldots, s_n \in S$). Τα στοιχεία $\sigma_{w,s}$ των $\Sigma_{<w,s>}$ ονομάζονται *συναρτησιακά σύμβολα τύπου* $<W,S>$. Τα στοιχεία των $\Sigma_{<[\ ,\ ]S>}$ (όπου [,] η κενή λίστα) ονομάζονται *σταθερές τύπου s*.

Ορισμός. Έστω Σ μια S – χαρακτηριστική. Τότε μια Σ – *άλγεβρα* (many – sorted algebra) Α αποτελείται από ένα S – σύνολο $\{As \mid s \in S\}$ (τα στοιχεία του καλούνται φορείς της άλγεβρας) και μια $S^* \times S$ – οικογένεια $\alpha = \{\alpha_{w,s} \mid w \in S^*, s \in S\}$ όπου $\alpha_{w,s}$ είναι απεικονίσεις της μορφής

$$A_{w,s} : \Sigma_{<w,s>} \to [A^w \to A_s]$$

(οι απεικονίσεις αυτές λέγονται και ερμηνείες των συναρτησιακών συμβόλων) και $A^w = A_{s_1} \times \ldots \times A_{s_n}$ όπου $w = s_1 \ldots s_n$, ενώ $[A^w \to A_s]$ είναι το σύνολο όλων των συναρτήσεων από το A^w στο Α. Θα συμβολίσουμε με σ_A το $\alpha_{<w,s>}(\sigma)$.

Παραδείγματα πολυειδών αλγεβρών

1. *Αυτόματα*

Ένα αυτόματο αποτελείται από τρία σύνολα X, S, Y (που καλούμε σύνολο *εισόδου*, *σύνολο καταστάσεων* και *σύνολο εξόδου* αντίστοιχα), ένα στοιχείο $s_0 \in S$ που καλείται *αρχική κατάσταση* και δύο συναρτήσεις $f : X \times S \to S$ και $g : S \to Y$ (καλούνται *συνάρτηση μεταφοράς* και *συνάρτηση εξόδου* αντίστοιχα).

Έστω S^{AYT} = {Input (Είσοδος), State (Κατάσταση), Output (Έξοδος)} το σύνολο των ειδών. Τότε η S^{AYT} – χαρακτηριστική Σ^{AYT} ορίζεται ως εξής:

$\Sigma^{AYT}_{\langle [,],\ \mathrm{State} \rangle} = \{S_0\}$ (Η S_0 καλείται αρχική κατάσταση),

$\Sigma^{AYT}_{\langle \mathrm{Input,\ State,\ State} \rangle} = \{f\}$,

$\Sigma^{AYT}_{\langle \mathrm{State,\ Output} \rangle} = \{g\}$ και $\Sigma_{w,s} = \varnothing$ για όλα τα άλλα $w \in (S^{AYT})$ και $s \in S^{AYT}$.

Συνεπώς τα αυτόματα είναι Σ^{AYT}– άλγεβρες.

2. *Modules*

Έστω Σ^{MOD} είναι μία S^{MOD} – χαρακτηριστική με $S^{MOD} = \{\Delta_1, \Delta_2\}$ όπου Δ_1 = Δακτύλιος και Δ_2 = Διανύσματα και

$\Sigma^{MOD}_{\langle \Delta_1, \Delta_2 \rangle} = \{+, -, 0, \times 1\}$,

$\Sigma^{MOD}_{\langle \Delta_1, \Delta_2 \rangle} = \{+, -, 0\}$,

$\Sigma^{MOD}_{\langle \Delta_1, \Delta_2, \Delta_1 \rangle} = \{\nu\}$, (βαθμωτό γινόμενο).

Τότε ένα module πάνω στο δακτύλιο R είναι μια Σ^{MOD} – άλγεβρα M όπου $M_{\Delta_1} = R$ και $M_{\Delta_2} = V$ (ένα σύνολο διανυσμάτων).

Άλλα παραδείγματα πολυειδών αλγεβρών είναι τα γραφήματα (graphs), οι φυσικοί α-ριθμοί κατά Peano, τα μονοειδή δρώντα επί συνόλων και άλλα.

Συμβολισμός. Εάν A και B είναι Σ – άλγεβρες, τότε $A \subseteq B$ σημαίνει $A_s \subseteq B_s$ για όλα τα $s \in S$.

Ορισμός. Μια Σ – άλγεβρα B θα καλείται υποάλγεβρα της Σ – άλγεβρας A εάν $A \subseteq B$ για όλα τα $s \in S$ και για κάθε σε $\sigma \in \Sigma$ ισχύει $\sigma_A(\alpha_1,\ldots,\alpha_1) = \sigma_A(\alpha_1,\ldots,\alpha_n)$ όπου $\alpha_i \in A_{s_i}$ i = 1,…,n.

Ορισμός. Έστω A και B δύο Σ – άλγεβρες, τότε ένας Σ – *ομομορφισμός* $h : A \to B$ εί-ναι μια οικογένεια συναρτήσεων $\langle h_s : A_s \to B_s \rangle$ $s \in S$ έτσι ώστε

(i) Εάν $\sigma \in \Sigma_{\langle [s], S \rangle}$ τότε $h_s(\sigma_A) = \sigma_B$, και

(ii) Εάν $\sigma \in \Sigma_{\langle s_1 \ldots s_n, s \rangle}$ και $\langle \alpha_1, \ldots, \alpha_n \rangle \in A_{s_1} \times \ldots \times A_{s_n}$ τότε

$$h_s[\sigma_A(\alpha_1, \ldots, \alpha_n)] = \sigma_B\left[h_{s_1}(\alpha_1), \ldots, h_{s_\nu}(\alpha_n)\right].$$

Ένας Σ – ομορφισμός $h : A \to B$ θα καλείται (1 – 1) (αντ. επί) όταν κάθε $h_s : A_s \to B_s$ είναι (1 – 1) (αντ. επί) όταν κάθε $s \in S$.

Η σύνθεση δύο ομομορφισμών είναι επίσης ομομορφισμός. Μάλιστα η σύνθεση ομο-μορφισμών είναι προσεταιριστική. Ο ταυτοτικός ομομορφισμός από μια Σ – άλγεβρα στον εαυτό της είναι η ταυτοτική συνάρτηση από το φορέα της A στον εαυτό του.

Είναι προφανές πως μια κλάση |C| αποτελούμενη από Σ – άλγεβρες μαζί με την κλάση όλων Σ – ομομορφισμών μεταξύ αυτών έχουν τη δομή της κατηγορίας. Ας καλέσουμε C αυτή την κατηγορία.

Αρχικές άλγεβρες

Ορισμός. Μια Σ – άλγεβρα Α καλείται *αρχική* (initial) σε σχέση με μια κατηγορία Σ – πολυειδών αλγεβρών C εάν και μόνο εάν για κάθε Σ – πολυειδή άλγεβρα Β της C υπάρχει ένας και μόνο ένας ομομορφισμός $h : A \to B$.

Η κατασκευή της άλγεβρας των λέξεων

Έστω μια S – χαρακτηριστική Σ. Τότε ορίζουμε τα σύνολα $T_{\Sigma,S}$ (όλων των Σ – όρων τύπου s) ως εξής:

(1) $\Sigma_{\langle [s],S\rangle} \subseteq T_{\Sigma,S}$ για κάθε $s \in S$ και

(2) $\sigma(t_1,\ldots,t_n) \in T_{\Sigma,S}$ για κάθε $\sigma \in \Sigma_{S_1\ldots S_n,S}$ και $t_i \in T_{\Sigma,S}$ για $i = 1,2,\ldots,n$.

Η S –οικογένεια $T = < T_{\Sigma,S} \mid s \in S$ συνιστά μια Σ – άλγεβρα, που καλείται *άλγεβρα λέξεων* ή *σύμπαν κατά Herbrand* ως ακολούθως:

Οι συναρτήσεις ερμηνείας $\alpha_{w,s}$ για κάθε $\sigma \in \Sigma$ και w = [ ] στέλνουν τις n – άδες (t1,…,tn) από το $(T_\Sigma)^W$ στον όρο σ (T1,…,tn) $\in T$, στην περίπτωση δε που w = [ ] στέλνουν τα $\sigma \in \Sigma$ στο σύμβολο $\sigma \in T$.

Το ακόλουθο θεώρημα έχει ιδιαίτερη σημασία.

Θεώρημα. Η άλγεβρα T_Σ είναι αρχική στην κατηγορία Alg_Σ όλων των Σ –αλγεβρών.

Απόδειξη. Η S – οικογένεια Τ γράφεται και ως εξής: $T_\Sigma = \bigcup_n T_\Sigma^{[n]}$.

Όπου τα $T_\Sigma^{[n]}$ ορίζονται επαγωγικά ως προς $n \in N$:

$$T_\Sigma^{[o]} = \left\langle \Sigma_{\langle [,],S\rangle} \mid s \in S \right\rangle$$

$$T_\Sigma^{[n+1]} = T_\Sigma^{[o]} \cup \left\langle \left\{ \sigma(t_1,\ldots,t_n)\sigma \in \Sigma_{S_1\ldots S_n,S} \right.\right.$$

και $$\left.\left. t_i \in \left(T_\Sigma^{[n]}\right)_s, \text{ για } i = 1,\ldots,n \text{ και } s_1,\ldots,s_n \in S \right\} \mid s \in S \right\rangle.$$

Έστω μία τυχαία Σ – άλγεβρα Α και δύο ομομορφισμοί h, h΄: $T_\Sigma \to A$. Θα δείξουμε ότι h = h΄ δια της επαγωγής επί του n. Όταν n = 0 οι ομομορφισμοί h και h΄ ταυτίζονται επειδή h(σ) = σ = h΄(σ).

Ας υποθέσουμε ότι οι ομομορφισμοί h και h΄ ταυτίζονται επί του . $T_\Sigma^{[n]}$ Τότε και επειδή

$$h(\sigma(t_1,\ldots,t_n)) = \sigma(h(t_1),\ldots,h(t_n)) = \sigma(h'(t_1),\ldots,h(t_n)) = h'(\sigma(t_1,\ldots,t_n)),$$

οι ομομορφισμοί h και h΄ ταυτίζονται και επί του $T_\Sigma^{[n+1]}$.

Συνεπώς οι h και h΄ ταυτίζονται επί του T_Σ.

Η ύπαρξη του ομομορφισμού h εξασφαλίζεται πάλι δια της επαγωγής επί του n. Είναι προφανές ότι μπορούμε να ορίσουμε τον h τόσο επί του $T_\Sigma^{[o]}$ σαν h(σ) = σ όσο και επί του $T_\Sigma^{[n+1]}$ αφού $h(\sigma(t_1,\ldots,t_n)) = \sigma(h(t_1),\ldots,h(t_n))$ και έχουμε υποθέσει λόγω επαγωγής ότι ο h έχει ήδη ορισθεί επί του $T_\Sigma^{[n]}$.

Η παρακάτω πρόταση έχει οδηγήσει στον ορισμό της ιδιαίτερα σημαντικής για την πληροφορική έννοια του αφηρημένου σχήματος δεδομένων (abstract data type). Ο ορισμός οφείλεται στην ομάδα ADJ.

Πρόταση. Έστω δύο Σ – άλγεβρες Α και Α΄. Εάν και οι δύο είναι αρχικές σε σχέση με την κατηγορία Alg_Σ όλων των Σ – αλγεβρών, τότε οι Α και Α΄ είναι ισόμορφες. Εάν μια Άλγεβρα Α΄΄ στην Alg_Σ είναι ισόμορφη με μια άλγεβρα Α που είναι αρχική Alg_Σ, τότε και η Α΄΄ θα είναι επίσης αρχική.

Απόδειξη. Αφού οι άλγεβρες Α και Α΄ είναι αρχικές στην κατηγορία Alg_Σ όλων των Σ – αλγεβρών, έπεται ότι υπάρχουν δύο (μοναδικοί) ομομορφισμοί h : A → A΄ και h΄ : A΄ → A . Κατά συνέπεια και εξαιτίας της μοναδικότητας τους h : h΄ = 1_A : A → A. Ομοίως h΄ = 1_A : A΄ → A΄. Άρα ο h είναι ένας ισομορφισμός.

Έστω h : A → A΄ ένας ισομορφισμός με αντίστροφο g΄ : A΄΄ → A και A αρχική στην Alg_Σ.

Έστω Α΄ μια τυχαία Σ – πολυειδής άλγεβρα. Επειδή η Α είναι αρχική έπεται ότι υπάρχει h : A → A΄ και κατά συνέπεια έχουμε ένα ομομορφισμό h΄΄ = h ; g : A΄΄ → A΄.

Εάν υπάρχει ομομορφισμός h΄ : A΄΄ → A τότε h΄ ; f : A → A΄΄ και h΄ ; f = h. Επίσης h΄΄ ; f = h. Κατά συνέπεια h΄ ; f = h΄΄ έτσι ώστε h΄΄ ; f ; g = h΄΄ άρα h = h΄. Δηλαδή η Α΄΄ είναι αρχική.

Είναι προφανές ότι οι ισομορφισμοί ορίζουν μια σχέση ισοδυναμίας επί της κλάσης των $\Sigma -$ αλγεβρών. Καλούμε κλάση ισομορφισμών $\Sigma -$ αλγεβρών κάθε κλάση ισοδυναμίας $\Sigma -$ αλγεβρών που προκύπτει από τους ισομορφισμούς.

Ορισμός. Καλούμε *αφηρημένο σχήμα δεδομένων* (abstract data type) κάθε κλάση ισομορφισμού που περιλαμβάνει μια αρχική άλγεβρα από την κατηγορία Alg_Σ των $\Sigma -$ αλγεβρών.

ΠΑΡΑΔΕΙΓΜΑΤΑ ΑΠΌ ΤΟΝ ΑΛΓΕΒΡΙΚΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

1. Χαρακτηριστικές

Οι **τύποι (sorts)** αντιστοιχούν σε διάφορους τύπους δεδομένων (data types) όπως ακέραιοι, μεταβλητές, πίνακες, μπουλιανές εκφράσεις, κ.τ.λ.

Οι **τελεστές (operators)** ορίζονται σαν «συναρτήσεις» με τη βοήθεια των τύπων.

Παράδειγμα

Sorts, Var, Exp, Pgm

όπου Var οι μεταβλητές ως τύπος

Exp οι εκφράσεις ως τύπος

Pgm τα προγράμματα ως τύπος.

Ο τελεστής

$$\text{ορ}\ \_ : = \_ : \text{Var Exp} \to \text{Pgm}$$

έχει ως είσοδο» μια μεταβλητή και μια έκφραση και ως «έξοδο» ένα πρόγραμμα.

Δηλαδή για παράδειγμα το

$$\mathbf{X : = X + 1}$$

έχει τύπο Pgm με την υπόθεση ότι το X έχει τύπο Var και ότι οι εκφράσεις με τύπο Exp περιλαμβάνουν εκφράσεις όπως X + 1.

Ορισμός. Μια χαρακτηριστική Σ αποτελείται από S από τύπους και μια οικογένεια $\{\Sigma_{\omega,s} \mid \omega \in \Sigma^*, s \in S\}$ από σύνολα (τελεστών).

Το αντικείμενο των φυσικών αριθμών

obj NAT is

sort Nat.

op O : $\to$ Nat.

op S_ : Nat → Nat.

endo.

Μια θεωρία για τα αυτόματα

th AUTOM is

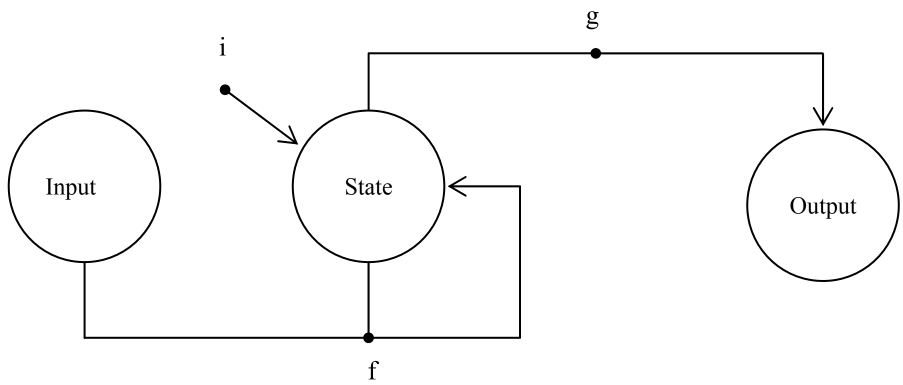
sorts Input Output

op i : → State.

op f : Input State → State.

op g : State → Output.

endth



ADJ διάγραμμα για τα αυτόματα

Ένα αντικείμενο για τις εκφράσεις των φυσικών αριθμών είναι:

obj NATEXP is

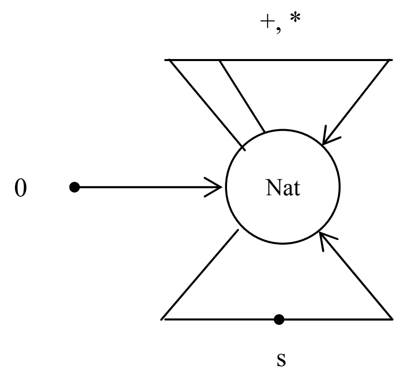
op O : → Exp.

op S_ : Exp → Exp.

Op _ * _ : Exp Exp → Exp.

end o

ADJ Διάγραμμα



Η θεωρία των γραφημάτων

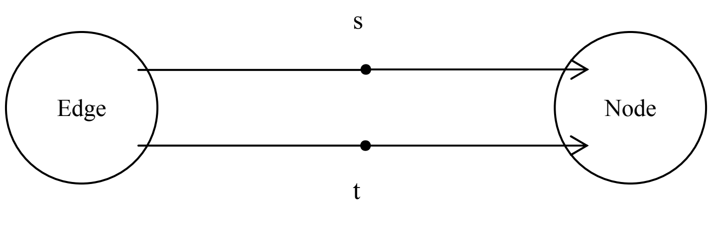
th GRAPH is

sorts Edge Node.

ops (s _)(t _) : Edge Node.

end th

Το ADJ διάγραμμα είναι:



subsort $t_1 < t_2$

ότι τα σύνολα που αντιστοιχούν στους τύπους t_1 και t_2 σχετίζονται με τη σχέση του περιέχεσθαι. Π.χ.

subsort Nat < Int.

subsort Nat < Int < Rat < Real.

Σ-ΑΛΓΕΒΡΕΣ

$$A_{Var} \times A_{Exp} \rightarrow A_{Pgm}$$

είναι η συνάρτηση που ερμηνεύει τον τελεστή που χρησιμοποιήσαμε ως παράδειγμα:

A_{Var} : το σύνολο των αντικειμένων με τύπο Var

A_{Exp} : το σύνολο των αντικειμένων με τύπο Exp (το σύνολο των εκφράσεων)

A_{Pgm} : το σύνολο των αντικειμένων με τύπο Pgm (το σύνολο των προγραμμάτων).

Ορισμός. Έστω μια χαρακτηριστική Σ με σύνολο τύπων Σ, μια Σ – άλγεβρα A αποτελείται από σύνολα φορέων (carrier set) A_S για κάθε τύπο $s \in S$, μαζί με μια συνάρτηση

$$A_\sigma : A_{s_1} \times \ldots \times A_{s_n} \rightarrow A_s$$

για κάθε τελεστή $\sigma : s_1 \ldots s_n \ \ s_n \in \Sigma_{\omega,s}$ όπου $\omega = s_1 \ldots s_n \neq$ □ και μια σταθερά $A_\sigma \in A_s$ για κάθε $\sigma \in \Sigma_{[\,],s}$.

Η Άλγεβρα είναι η **ερμηνεία** ή το **μοντέλο** για τη χαρακτηριστική Σ.

Παράδειγμα. Έστω Σ^{NAT} η χαρακτηριστική του αντικειμένου NAT. Τότε οι φυσικοί αριθμοί είναι μια Σ^{NAT} – άλγεβρα με τον εξής τρόπο:

$$A_{Nat} = \omega = \{0,1,2,\ldots\}$$

$$A_0 = 0$$

$$A_s(n) = n + 1$$

Η παραπάνω ερμηνεία δεν είναι όμως η μοναδική.

Ιδού τα εξής παραδείγματα:

Ας ορίσουμε την άλγεβρα B ως εξής

$$B_{Nat} = \{0, 1\}$$

$$B_0 = 0$$

και

$$B_s(n) = 1 - n.$$

Ομοίως, την άλγεβρα C ως εξής:

$$C_{Nat} = \{0\}$$

$$C_0 = 0$$

$$C_s(0) = 0.$$

Όροι

Το σύνολο των Σ – όρων τύπου S, συμβ. $T_{\Sigma,s}$ για $s \in S$ ορίζεται από τις εξής δύο συνθήκες:

(1) εάν $\sigma \in \Sigma_{[\,],s}$, τότε $\sigma \in T_{\Sigma,s}$

(2) εάν $\sigma \in T_{\omega,s}$ όπου $\omega = s_1 \ldots s_n \neq$ □ και εάν $t_i \in T_{\Sigma,s_i}$ για $i = 1,\ldots,n$ τότε $\sigma(t_{1,\ldots,}t_n) \in T_{\Sigma,s}$

$$\boxed{T_\Sigma = \left(T_{\Sigma,s}\right)_{s \in S}}$$

Η αρχική άλγεβρα στην κατηγορία των Σ – αλγεβρών (ονομάζεται άλγεβρα όρων).

Το T_Σ είναι άλγεβρα ως εξής:

$$s \to (T_\Sigma)_s$$

$$\sigma \in \Sigma_{□,s} \to \sigma \in T_{\Sigma,s}$$

$$\sigma \in T_{\omega,s} : (t_1,\ldots,t_n) \to \sigma(t_1,\ldots,t_n)$$

$t_1,\ldots,t_n \to \sigma(t_1,\ldots,t_n)$, $t_i \in T_{\Sigma,s_i}$ $i = 1,\ldots,n$.

Χρησιμοποιώντας τη χαρακτηριστική Σ^{NATEXP} σχηματίζουμε τους παρακάτω όρους τύπου Exp:

(ss0) * (ss0)

(ss0) + ((s0) * (sss)))

(ss((ss0) * (ssss0))) + ((sss0 * (sss0))

Για τα αυτόματα η αρχική άλγεβρα είναι

$$T_{\Sigma^{AUTOM},Input} = \varnothing$$

$$T_{\Sigma^{AUTOM},State} = \{i\}$$

$$T_{\Sigma^{AUTOM},Output} = \{g(i)\}.$$

Μεταβλητές και Συναρτήσεις Επισύναψης

Δηλώνουμε τις μεταβλητές ως εξής: var X,Y,Z : Exp.

Ορισμός. Έστω μια χαρακτηριστική Σ, μια χαρακτηριστική που αποτελείται μόνο από σταθερές (ground siguature) = ξεχωριστή από τη Σ και μια Σ – άλγεβρα Α, τότε μια επισύναψη τιμών από το Α σε «μεταβλητές» από τη Ξ είναι μια οικογένεια συναρτήσεων

$$\Theta_s : \Xi\ \Sigma_{\square,,s} \to A_s,$$

μια για κάθε $s \in S$.

Η Θ είναι μια ερμηνεία (interpretation)

$$\Theta : \Xi \to A$$

την οποία ορίζουμε ως εξής:

$$\overline{\Theta}_s : \left(T_{\Sigma\cup\Xi}\right)_s \to A_s$$

δηλαδή

(1) $\overline{\Theta}_s(X) = \Theta(X)$ για X στο $\Xi_{\square,,s}$ και $\overline{\Theta}_s(c) = A_c$ για c στο $\Sigma_{\square,,s}$.

(2) Έστω $\sigma \in \Sigma_{s_1,\ldots,s_{n},s}$ και $t_i \in T_{\Sigma,s_i}$ με $\overline{\Theta}(t_i) = \alpha_i$ για i = 1,…,n, τότε $\overline{\Theta}(\sigma(t_1,\ldots,t_n)) = A_\sigma(\alpha_1,\ldots,\alpha_\nu$

$$\overline{\Theta} : T_{\Sigma\cup\Xi} \to A$$

$\overline{\Theta}$: Σ – ομομορφισμός, $T_{\Sigma\cup\Xi}$: άλγεβρα όρων

Εξισώσεις

$$\text{eq } X + Y = Y + X \quad (\text{στην OBJ})$$

Ορισμός. Μία Σ – εξίσωση αποτελείται από μια χαρακτηριστική που αποτελείται μόνο από σταθερές Ξ (ξεχωριστά από τη Σ) και δύο (Σ $\cup$Ξ) – όρους του ίδιου τύπου που τους ονομάζουμε αριστερό και δεξιό σκέλος της εξίσωσης.

Συμβολισμός:: $(\forall \Xi)\ t_1 = t_2$ με

$$t_1, t_2 \in \left(T_{\Sigma\cup\Xi}\right)_s$$

για κάποιο τύπο $s \in S$.

Χρειαζόμαστε ένα ακριβή ορισμό για το πότε μια άλγεβρα ικανοποιεί μια εξίσωση.

Ορισμός. Έστω μια Σ – εξίσωση e της μορφής $(\forall \Xi)\ t = t'$ και μια Σ – άλγεβρα A. Λέμε ότι η A ικανοποιεί την e εάν και μόνο ένα $\overline{\Theta}(t) = \overline{\Theta}(t')$ για κάθε ερμηνεία $\Theta : \Xi \to A$.

Είναι ιδιαίτερα χρήσιμο πολλές φορές να έχουμε εξισώσεις υπό συνθήκη (conditional equations).

Παράδειγμα:

$$\text{cq } X = Y \text{ if}$$
$$X * Z == Y * Z$$
$$\text{and } Z =/= 0.$$

Μια Σ – εξίσωση υπό συνθήκη αποτελείται από τρεις Σ – όρους, έστω t_1, t_2, t_3 με μεταβλητές από κάποια χαρακτηριστική Ξ που αποτελείται μόνο από σταθερές έτσι ώστε οι να είναι του ιδίου τύπου t_1, t_2 και t_3 να είναι τύπου Bool

$$“(\forall \Xi)\ t_1 = t_2 \text{ if } t_3”$$

Μια Σ – εξίσωση και συνθήκη ικανοποιείται από μια Σ – άλγεβρα A εάν και μόνο εάν για κάθε αντικατάσταση $\Theta : \Xi \to A$ έχουμε ότι $\overline{\Theta}(t_1) = \overline{\Theta}(t_2)$ όταν $\overline{\Theta}(t_3) = \text{true}$.

Παραδείγματα

(1) Η αλγεβρική δομή της ομάδας

th GROUP

sort Elt

op e : $\to$ Elt

op $-^{-1}$: Elt $\to$ Elt

op $-\ -$: EltElt $\to$ Elt

vars X, Y, Z : Elt.

eq X * e = X.

eq X *(X – 1) = e.

eq (X * Y) * Z = X * (Y * Z).

endth

(2) Η δομή της λίστας

obj NATLIST is

sorts Nat NeList List.

subsorts Nat < NeList < List.

op O : $\to$ Nat.

op s_{-} : Nat $\to$ Nat.

op nil : $\to$ List.

op – * – : List List $\to$ List.

op – * – : NeList List $\to$ NeList.

op – * – : List NeList $\to$ NeList.

op head_{-} : NeList $\to$ Nat.

op tail_{-} : NeList $\to$ List.

vars X, Y, Z : List.

Var N : Nat.

eq X * nil = X.

eq nil * X = X.

eq(X * Y) * Z = X * (Y * Z).

```
    eq head (N * X) = N.
    eq tail (N * X) = X.
    eq head N = N.
    eq tail N = nil.
end 0.
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Arnon Avron, Furio Honsell, and Ian Mason. Using typed lambda calculus to implement formal systems on a computer. Technical Report ECS – LFCS – 87 – 31, Laboratory for Computer Science, University of Edinburgh, 1987.

[2] K. J. Barwise. Axioms for Absract Model theory. *Ann. Math. Logic,* 7:221 – 265, 1974.

[3] K. J. Barwise and S. Feferman. *Model theoretic logics*. North Holland, 1985.

[4] R. Burstall and R. Diaconescu. Hiding in institutions, draft. April 1992. Programming Research Group, Oxford University.

[5] Rod Burstall and Joseph Goguen. Putting theories together to make specifications. In Raj Reddy. Editor. *Proceedings, Fifth International Joint Conference on Artificial Intelligence*, pages 1045 – 1058. Department of Computer Science, Carnegie – Mellon University, 1977.

[6] Rod Burstall and Joseph Goguen. The semantics of Clear, a specification language, volume 86, pages 292 – 332. Springer – Verlag LNCS, 1980.

[7] Razvan Diaconescu, Joseph Goguen and Petros Stefaneas. Logical support for moularisation, *Proceedings* of Workshop on Logical Frameworks, Cambridge University Press, 1993.

[8] Razvan Diaconescu. Translating between equational logics, *preliminary draft*, 1992. Oxford PRG.

[9] Donald Sannella and Andrzej Tarleki. Extended ML: an Institution indepndent framework for formal program development, volume 240, pages 364 – 389. Lecture Notes in Computer Science, 1886. Springer – Verlag.

[10] José Fiadeiro and Amilcar Sernadas. Structuring theories on consequence. In Donald Sannella and Andrzej Tarlecki, editors. *Recent Trends in Data Type Specification*, pages 44 – 72, Springer, 1988. Lecture Notes in Computer Science, Volume 332.

[11] Kokichi Fuatsugi, Joseph Goguen, Jean – Pierre Jouannaud and Jose Meseguer. Principles of OBJ2, pages 52 – 66. *12th ACM Symposium on Principles of Programming Languages*, Association for Computing Machinery, 1985.

[12] Joseph Goguen. Categorical foundations for general systems theory. In F. Pichler and R. Trappl, editors, *Advances in Cybernetics and Systems Research*, pages 121 – 130. Transcripta Book, 1973.

[13] Joseph Goguen. Types as theories. In George Michael Reed, Andrew William Roscoe, and Ralph F. Wachter, editors. *Topology and Category Theory in Computer Science*, pages 357 – 390. Oxford, 1991. Proceedings of a Conference held at Oxford, June 1989.

[14] Joseph Goguen and Rod Burstall. A study in the foundations of programming methology: Specifications, institutions, charters and parchments. In David Pitt. Samson Abramski, Axel Poigné and David Rydeheard, editors. *Proceedings, Conference on Category Theory and Computer Programming*, pages 313 – 333, Springer, 1986. Lecture Notes in Computer Science, Volume 240; also, Report CSLI – 86 – 54. Center for the Study of Language and Information, Stanford University, June 1986.

[15] Joseph Goguen and Rod Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95 – 146, January 1992. Draft appears as Report ECS – LFCS – 90 – 106, Computer Science Department, University of Edinburgh, January 1990; an early ancestor is "Introducing Institutions" *in Proceedings, Logics of Programming Workshop*, Edward Clarke and Dexter Kozen, Eds., Springer Lecture Notes in Computer Science, Volume 164, pages 221 – 256, 1984.

[16] Joseph Goguen and José Meseguer. Eqlog: Equality, types and generic modules for logic programming. In Douglas DeGroot and Gary Lindstrom, editors, *Logic Programming*: *Functions, Relations and Equations*, pages 295 – 363, Prentice – Hall, 1986. An earlier version appears in *Journal of Logic Programming*, Volume 1, Number 2, pages 179 – 210, September 1984.

[17] Joseph Goguen and Rod Burstall. Institutions: Abstract Model Theory for specification and programming, 1989. Submitted for publication a preliminary version appears as Report CSLI – 85 – 30, Stanford University.

[18] Joseph Goguen and Andrew Stevesn, Keith Hobley and Hendrik Hilberdink. 2OBJ, a metalogical framework based on equational logic. *Phlosophical Transactions of the Royal Society*. Series A, 339:69 – 86, 1992. Also in *Mechanized Reasoning and Hardware Design*, edited by C.A.R. Hoare and M.J.C. Gordon, Prentice – Hall, 1992, pages 69 – 86.

[19] Joseph Goguen and Timothy Winkler, Introducing OBJ3. Technical report, 1988. Technical Report SRI – CSL – 88 – 9. SRI International.

[20] P. Halmos. *Algebraic Logic*. Academic Press.

[21] Robert Harper, Furio Honsell and Gordon Plotkin. A framework for defining logics. In *Proceedings, Second Symposium on Logic in Computer Science*, pages 194 – 204. IEEE Computer Society, 1987.

[22] Robert Harper, Donald Sannella and Andrzej Tarlecki. Logic representation in LF. In David Pitt, David Rydeheard, Peter Bybjer, Andrew Pitts and Axel Poigné, editors. *Proceedings, Conference on Category Theory and Computer Science*, pages 250 – 272. Springer, 1989. Lecture Notes in Computer Science, Volume 389.

[23] J. Lambek and P.J. Scott. *Introduction to higher order categorical logic*. Cambridge Studies in Advanced Mathematics, 1986.

[24] Joachim Lambek and Phil Scott. *Introduction to Higher Order Categorical Logic*. Cambridge 1986. Cambridge Studies in Advanced Mathematics, 1986.

[25] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, 1971.

[26] P. Lindestrom. First order logic and generalized quantifiers. *Theoria*, 32:186 – 195, 1966.

[27] José Mesguer. General logics. In H. – D. Ebbinhaus et al., editors, *Proceedings, Logic Colloquium*, 1987, pages 275 – 329. North – Holland, 1989.

[28] B. J. Pierce. *Basic Category Theory for Computer Scientists*. The MIT Press, 1991.

[29] Lucia Rapanotti and Adolfo Socorro. Introducing FOOPS. Technical report, Programming Research Group. Oxford University, 1992.

[30] D. Sannella and A. Tarlecki. Specifications in an arbitrary institution. *Information and Computation*, 76:165 – 210, 1988.

[31] Dana Scott. Completeness and axiomatizability in many – valued logic. In Leon Henkin et al., editor, *Proceedings. Tarski Symposium*, pages 411 – 435. American Mathematical Society, 1974.

[32] Andrzej Tarlecki. Bits and pieces of the theory of institutions. In David Pitt, Samson Abramsky, Axel Poigné and David Rydeheard, editors, *Proceedings, Summer Workshop on Category Theory and Computer Programming*, pages 334 – 360, Springer, 1986. Lecture Notes in Computer Science, Volume 240.

[33] Andrzej Tarlecki. On the existence of free models in abstract algebraic institutions. *Theoretical Computer Science*, pages 333 – 360, 1986.

[34] Ρ. Βορεάδου. *Μικρή Εισαγωγή στη Θεωρία των Κατηγοριών*, Έκδοση, Έδρα Γενικών Μαθηματικών, Πανεπιστήμιο Αθηνών, 1977.