Γραμμική λογική

Απόστολος Συρόπουλος
28ης Οκτωβρίου 366
671 00 Ξάνθη
e-mail: `apostolo@obelix.ee.duth.gr`
URL: `http://obelix.ee.duth.gr/ apostolo`

1 Απριλίου 1998

1 Εισαγωγή

Η γραμμική λογική δεν αποτελεί μια ακόμη εναλλακτική λογική, αλλά μία βελτίωση της συνηθισμένης λογικής[1]. Η γραμμική λογική ανακαλύφθηκε από το γάλλο Jean-Yves Girard, ενώ αυτός δούλευε πάνω στις ιδιότητες των χώρων συνάφειας [12, 14]. Το βασικότερο πρόβλημα, τόσο της κλασικής όσο και της διαισθητικής λογικής, που λύνει η γραμμική λογική είναι αυτό των *στατικών προτάσεων*. Σύμφωνα με την κλασική λογική κάθε πρόταση υφίσταται πάντα και είναι είτε αληθής είτε ψευδής. Δεν χρειάζονται βαρύγδουπα μαθηματικά για να δείξει κανείς ότι μια τέτοια υπόθεση δεν ευσταθεί γενικά. Για παράδειγμα η πρόταση *η θερμοκρασία είναι 35°* είναι αληθής μια ζεστή ημέρα του Αυγούστου, αλλά είναι σίγουρα ψευδής μια οποιαδήποτε ημέρα του χειμώνα. Δηλ. η αλήθεια της πρότασης δεν είναι σταθερή όπως υποθέτει η κλασική λογική. Όμως αυτό δεν είναι το μόνο πρόβλημα της κλασικής λογικής. Παρουσιάζουμε το κλασικό παράδειγμα που καταδείχνει μερικά ακόμη προβλήματα της κλασικής λογικής. Έστω οι προτάσεις:

$$\begin{aligned} D &\triangleq \text{δαπάνη ενός δολλαρίου} \\ M &\triangleq \text{αγορά ένός πακέτου Marlboro} \\ C &\triangleq \text{αγορά ένός πακέτου Camel} \end{aligned}$$

Επίσης θεωρήστε ότι οι παράστασεις $D \Rightarrow M$ και $D \Rightarrow C$ είναι αληθείς και σημαίνουν ότι με ένα δολλάριο αγοράζει κανείς ένα πακέτο Marlboro ή ένα πακέτο Camel αντίστοιχα. Τότε στην κλασική λογική μπορεί κανείς να συνάγει ότι ισχύει η πρόταση

$$D \Rightarrow (M \wedge C)$$

Με άλλα λόγια με ένα δολλάριο μπορεί κανείς να αγοράσει ένα πακέτο Marlboro και ένα πακέτο Camel! Το παράδοξο οφείλεται στο τρόπο με τον οποίο

[1] Για μια γενική εισαγωγή στη μαθηματική λογική βλέπε, π.χ., [8].

αντιλαμβάνεται κανείς το τελεστή $\wedge$: αν αυτός σημαίνει *επιλογή* μεταξύ δύο, προφανώς η παραπάνω πρόταση ευσταθεί, αν όμως σημαίνει *ταυτόχρονη κτίση* φυσικά και υπάρχει παραδοξο.

2 Σύνταξη της γραμμικής λογικής

2.1 Οι τελεστές της γραμμικής λογικής

Ένας βασικός μηχανισμός εξαγωγής συμπερασμάτων της λογικής είναι ο κανόνας *modus ponens*[2], δηλ. άμα γνωρίζουμε ότι μια πρόταση D ισχύει, ενώ ισχύει και η πρόταση $D \Rightarrow M$, τότε συνάγουμε ότι ισχύει η πρόταση M, ενώ η πρόταση D *δεν παύει* να ισχύει. Ο κανόνας modus ponens εκφρασμένος στη μορφή του λογισμού του Prawitz (natural deduction) έχει την παρακάτω μορφή:

$$\frac{D \quad D \Rightarrow M}{M}.$$

(Ότι βρίσκεται πάνω από τη γραμμή αποτελεί *υπόθεση*, ενώ αυτό που βρίσκεται κάτω από τη γραμμή *συμπέρασμα*.) Αν οι προτάσεις του κανόνα, για παράδειγμα, αντιστοιχιθούν σε αυτές του παραδείγματος της προηγούμενης ενότητας, ουσιαστικά ο κανόνας αυτό σημαίνει πως αν έχω ένα δολλάριο και αν με ένα δολλάριο αγοράζω ενα πακέτο Marlboro, τότε μπορώ να αγοράσω ένα πακέτο Marlboro αλλά και τα χρήματα μου να μην δαπανίσω! Αν και είναι προφανές, θα πρέπει να τονισθεί ότι το πρόβλημα δε βρίσκεται στον κανόνα modus ponens, αλλά στον τρόπο χρήσης του. [Νομίζω πως θα συμφωνήτε ότι κάθε υπουργός οικονομικών αν ήταν και μελετητής της λογικής (logician), θα λάτρευε την κλασική λογική...]

Τα προβλήματα της κλασικής λογικής που συναντήσαμε λύνονται από την γραμμική λογική με την εισαγωγή νέων τελεστών, οι οποίοι αντιστοιχούν στο νέο τρόπο αντιμετώπισης των πραγμάτων. Για τη γραμμική λογική υπάρχουν *καταστάσεις, δράσεις* και *αντιδράσεις*. Οι καταστάσεις αντιστοιχούν σε *αιώνιες* αλήθειες, π.χ., μαθηματικά αξιώματα, κτλ, ενώ οι δράσεις και οι αντιδράσεις σε απλές, καθημερινές προτάσεις. Η γραμμική λογική επιτρέπει την επαναλαμβανόμενη χρήση καταστάσεων, αλλά με ένα ελεγχόμενο τρόπο: η πρόταση $!D$ σημαίνει απεριόριστη δημιουργία, ενώ η πρόταση $?D$ απεριόριστη κατανάλωση. Η κατανόηση του προβλήματος της σύζευξης οδηγεί σε δύο νέους τελεστές σύζευξης: τον $\&$ (επιλογή) και τον $\otimes$ (ταυτόχρονη κτίση). Όμως υπάρχουν και δύο νέοι τελεστές διάζευξης:

- Ο $\oplus$ (συζυγής του $\&$) ο οποίος εκφράζει την επιλογή μιας δράσης μεταξύ δύο πιθανόν τύπων, για παράδειγμα, η πρόταση $D \multimap M \oplus C$ (ο τελεστής $\multimap$ είναι αυτός της γραμμικής συνεπαγωγής), σημαίνει ότι γνωρίζουμε

[2] Ο όρος σημαίνει *τρόπος τοποθέτησης*, ενώ το όνομα του συζυγούς κανόνα, δηλ. του *modus tollens*, σημαίνει *τρόπος απόσυρσης*[27]. Και οι δυό όροι είναι μεσαιωνκοί.

πως με ένα δολλάριο παίρνουμε ένα πακέτο Marlboro ή ένα πακέτο Camel αλλά δεν ξέρουμε, πλέον, τι να διαλέξουμε.

- Ο $\parr$ (συζυγής του $\otimes$) ο οποίος εκφράζει την εξάρτηση δύο τύπων δράσης· η ερμηνεία του $\parr$ δεν είναι τόσο απλή, όμως με τη χρήση της γραμμικής άρνησης μπορούμε να πούμε ότι η παράσταση $D \parr M$ *ισοδυναμεί* είτε με την παράσταση $D^{\perp} \multimap M$ είτε με την παράσταση $M^{\perp} \multimap D$.

Ο πιο σημαντικός τελεστής της γραμμικής λογικής είναι αυτός της γραμμικής άρνησης. Ο τελεστής αυτός εκφράζει την συζυγία δράσης και αντίδρασης:

$$\text{δράση τύπου } A = \text{αντίδραση τύπου } A^{\perp}$$

Η βασικότερη ιδιότητα της άρνησης είναι ότι οι παράστασεις A και $A^{\perp\perp}$ είναι ταυτόσημες (σκεφτήτε ότι στη διαισθητική λογική αυτό δεν ισχύει γενικά). Εξ' ορισμού ισχύουν οι παρακάτω σχέσεις:

$$\begin{array}{lcllcl}
(A \otimes B)^{\perp} & := & A^{\perp} \parr B^{\perp} & (A \parr B)^{\perp} & := & A^{\perp} \otimes B^{\perp} \\
(A \mathbin{\&} B)^{\perp} & := & A^{\perp} \oplus B^{\perp} & (A \oplus B)^{\perp} & := & A^{\perp} \mathbin{\&} B^{\perp} \\
(\exists x.A)^{\perp} & := & \forall x.A^{\perp} & (\forall x.A)^{\perp} & := & \exists x.A^{\perp} \\
(!A)^{\perp} & := & ?A^{\perp} & (?A)^{\perp} & := & !A^{\perp} \\
\mathbf{1}^{\perp} & := & \perp & \perp^{\perp} & := & \mathbf{1} \\
\top^{\perp} & := & \mathbf{0} & \mathbf{0}^{\perp} & := & \top \\
(p)^{\perp} & := & p^{\perp} & (p^{\perp})^{\perp} & := & p
\end{array}$$

$$A \multimap B := A^{\perp} \parr B$$

Οι προτάσεις $\perp$, $\top$, $\mathbf{1}$ και $\mathbf{0}$ αποτελούν τα *ουδέτερα στοιχεία* των τελεστών $\parr$, $\&$, $\otimes$ και $\oplus$ αντίστοιχα. Έτσι, για παράδειγμα, ισχύει $\perp \parr A = A = A \parr \perp$. Δίνουμε τώρα ένα παράδειγμα χρήσης των συνδέσμων της γραμμικής λογικής όπως αυτό παρουσιάζεται στο [24]:

> Έστω ότι έχουμε στην διάθεση μας 5 δολλάρια και έστω ότι σε ένα εστιατόριο με αυτό το ποσό μπορούμε να αγοράσουμε ένα hamburger (H), ένα αναψυκτικό (C), όσες τηγανιτές πατάτες θέλουμε (F), κρεμμυδόσουπα (O) ή σαλάτα (S), δική μας η επιλογή, και γλυκό (P) ή παγωτό (I) αναλόγως του τι διαθέτει το κατάστημα, τότε η παρακάτω παράσταση εκφράζει αυτή ακριβώς την περίπτωση:
>
> $$(D \otimes D \otimes D \otimes D \otimes D) \multimap [H \otimes C \otimes !F \otimes (O \mathbin{\&} S) \otimes (P \oplus I)]$$

2.2 Ο λογισμός Gentzen

Ο λογισμός Gentzen είναι ένα θαυμάσιο εργαλείο *επαγωγής συμπερασμάτων*. Αποτελεί, ουσιαστικά, τον πυρήνα πολλών συστημάτων αυτόματης απόδειξης θεωρημάτων, μεταξύ των οποίων και της γλώσσας προγραμματισμού

Prolog. Κάθε κανόνας του λογισμού έχει τη μορφή

$$\frac{\Gamma \vdash \Delta}{\Gamma' \vdash \Delta'},$$

όπου Γ $(= A_1, \ldots, A_n)$ και Δ $(= B_1, \ldots, B_n)$ είναι πεπερασμένες ακολουθίες προτάσεων (formulae). Η σημασία του κανόνα είναι ότι μπορούμε να αντικαταστίσουμε το $\Gamma \vdash \Delta$ με το $\Gamma' \vdash \Delta'$. Η *απορρέουσα πρόταση* (sequent) $\Gamma \vdash \Delta$ σημαίνει ότι:

$$\text{από} \quad A_1 \,\text{και} \ldots \text{και}\, A_n \quad \text{συνεπάγεται} \quad B_1 \,\text{ή} \ldots \text{ή}\, B_n$$

Φυσικά θα πρέπει να ξεκαθαραστεί η ακριβής σημασία των συνδέσμων *και*, *ή* και *συνεπάγεται*. Ο λογισμός Gentzen που αντιστοιχεί στην κλασική λογική *πάσχει* από δύο *νόσους*: του κανόνα *εξασθένησης* (weakening) και του κανόνα *συστολής* (contraction). Οι κανόνες αυτοί έχουν την εξής μορφή:

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta}\ \mathcal{L}W \qquad \frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}\ \mathcal{R}W$$
$$\frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}\ \mathcal{L}C \quad \frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\ \mathcal{R}C$$

Γιατί όμως είναι προβληματικοί αυτοί οι δύο κανόνες; Ο κανόνας εξασθένησης ουσιαστικά εισαγάγει ένα νέο όρο ο οποίος μοιάζει να έχει έρθει από το πουθενά: αυτό σημαίνει ότι έχουμε *αίτιο* χωρίς *αποτέλεσμα*. Μολονότι, ο κανόνας αυτός αποτελεί ουσιώδες εργαλείο εξαγωγής συμπερασμάτων, εντούτοις η γραμμική λογική τον απαγορεύει. Από την άλλη ο κανόνας συστολής ουσιαστικά λέει ότι αυτό που έχουμε, θα το έχουμε για πάντα ανεξάρτητα από το πώς το χρησιμοποιούμε. Όμως αυτό δεν ευσταθεί και για το λόγο αυτό η γραμμική λογική απαγορεύει αυτό το κανόνα. [Παραλείποντας κανείς και τον κανόνα *ανταλλαγής* (exchange), βλ. παρακάτω, φτάνει στη μη-μεταθετική γραμμική λογική.] Όμως λογική χωρίς αυτούς τους κανόνες δεν νοείται, έτσι η γραμμική λογική τους εισαγάγει εκ νέου αλλά μόνο στην περίπτωση που χρησιμοποιούμε καταστάσεις και πάντα με ελεγχόμενο τρόπο. Ο λογισμός που παρουσιάζεται παρακάτω έχει *μονόπλευρες* απορρέουσες προτάσεις και αυτό οφείλεται στο γεγονός ό,τι η απορρέουσα πρόταση $\Gamma \vdash \Delta$ μπορεί να παρασταθεί ως $\vdash \Gamma^{\perp}, \Delta$.

Κανόνες ταυτότητας/άρνησης

$$\frac{}{\vdash A, A^{\perp}}\ (\text{ταυτότητα}) \qquad \frac{\vdash \Gamma, A \quad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}\ (\text{κανόνας αποκοπής})$$

Δομικός κανόνας

$$\frac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, B, A, \Delta}\ (\text{κανόνας ανταλλαγής})$$

Κανόνες λογικής

$$\frac{}{\vdash \mathbf{1}}\ (\text{ένα}) \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \perp}\ (\text{ψεύδος})$$

$$\frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta} \text{ (επί)} \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A ⅋ B} \text{ (παραλληλία)}$$

$$\frac{}{\vdash \Gamma, \top} \text{ (αλήθεια)} \qquad \text{(δεν υπάρχει για το μηδέν)}$$

$$\frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \,\&\, B, \Delta} \text{ (μαζί)} \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B} \text{ (δεξιό συν)} \qquad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B} \text{ (αριστερό συν)}$$

$$\frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A} \text{ (βεβαίως)} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, ?A} \text{ (εξασθένηση)}$$

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A} \text{ (εγκατάλειψη)} \qquad \frac{\vdash \Gamma, ?A}{\vdash \Gamma, ?A, ?A} \text{ (συστολή)}$$

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, \forall x A} \text{ (για όλα τα } x \text{ που δεν είναι ελεύθερα στο } \Gamma) \qquad \frac{\vdash \Gamma, A[t/x]}{\vdash \Gamma, \exists x A} \text{ (υπάρχει)}$$

Από το κανόνα για το ⅋ μπορεί κανείς να συνάγει ότι το κόμμα συμπεριφέρεται ως κρυφό ⅋. (Στην περίπτωση που γράφουμε τον λογισμό και από τις δυο μεριές του $\vdash$, στα αριστερά το κόμμα συμπεριφέρεται ως κρυφό $\otimes$.) Το συμπέρασμα, λοιπόν, είναι ότι οι σύνδεσμοι *και*, *ή* και *συνεπάγεται* αντιστοιχούν στους τελεστές $\otimes$, ⅋ και $\multimap$. Τέλος, αξίζει να αναφερθεί πως ισχύει και στη γραμμική λογική ο κανόνας Haupsatz, δηλ. ότι αν αποδείξουμε κάποιο θεώρημα χρησιμοποιώντας τον κανόνα αποκοπής, μπορούμε να βρούμε μια απόδειξη που δεν κάνει χρήση αυτού του κανόνα.

3 Ενσωμάτωση της διαισθητικής λογικής

Η διαισθητική λογική (intuitionistic logic) [17] ως γνήσιο τέκνο των διαισθητικών μαθηματικών απορίπτει εξ' ορισμού την *αρχή του αποκλειομένου τρίτου*. Ενώ παράλληλα δέχεται ότι υπάρχει η απόδειξη μιας πρότασης αν και μόνο αν υπάρχει μια *κατασκευή* της πρότασης. Η διαισθητική λογική αποτέλεσε την βάση του συναρτησιακού προγραμματισμού [11]· έτσι από νωρίς έγινε αντιληπτό ό,τι μια διαισθητική γραμμική λογική θα επέτρεπε την εύκολη χρήση της γραμμικής λογικής στην Επιστήμη των Η/Υ. Κάθε πρόταση της διαισθητικής λογικής μπορεί να *μεταφραστεί* σε πρόταση της γραμμικής λογικής χρησιμοποιώντας τους παρακάτω κανόνες:

$$\begin{gathered}
p^* := p \quad (\text{το } p \text{ είναι } \textit{άτομο}),\\
[A \Rightarrow B]^* := !A^* \multimap B^*,\\
[A \wedge B]^* := A^* \,\&\, B^*,\\
[\forall x A]^* := \forall x A^*,\\
[A \vee B]^* := !A^* \oplus !B^*,\\
[\exists x A]^* := \exists x !A^*,\\
[\neg A]^* := !A^* \multimap \mathbf{0}
\end{gathered}$$

Επιπλέον η απορρέουσα πρόταση $\Gamma \vdash A$ αποδεικνύεται διαισθητικά αν και μόνο αν η πρόταση $!\Gamma^* \vdash A^*$, δηλ. $\vdash ?\Gamma^{*\perp}, A^*$, αποδεικνύεται γραμμικά.

4 Ενσωμάτωση της κλασικής λογικής

Αν και η μετάφραση της διαισθητικής λογικής στη γραμμική λογική είναι σχετικά απλή, η μετάφραση της κλασικής λογικής είναι πιο δύσκολη. Ο λόγος είναι ότι ο λογισμός Gentzen της κλασικής λογικής παρουσιάζει δεξιές και αριστερές συμμετρίες, δηλ. ο ίδιος κανόνας έχει δεξιά μορφή (στα δεξιά του $\vdash$) αλλά και αριστερή μορφή (στα αριστερά του $\vdash$). Για να μεταφράσουμε μια πρόταση της κλασικής λογικής στη γραμμική λογική της αναθέτουμε *πολικότητα*: *θετική* ή *αρνητική*. Η ιδέα πίσω από την ανάθεση πολικοτήτων είναι ό,τι σε μια θετική παράσταση μπορούμε να εφαρμόσουμε τους δεξιούς δομικούς κανόνες, ενώ σε μια αρνητική τους αριστερούς. Παρακάτω δίνουμε τους κανόνες μετάφρασης (οι αριθμοί αναφέρονται στην πολικότητα):

A	B	$A \vee B$	$A \wedge B$	$A \Rightarrow B$	$\neg A$	$\forall x A$	$\exists x A$
$+1$	$+1$	$A \otimes B$	$A \oplus B$	$A \multimap ?B$	$A^{\perp}$	$\forall x ?A$	$\exists x A$
-1	$+1$	$!A \otimes B$	$A \parr ?B$	$A^{\perp} \oplus B$	$A^{\perp}$	$\forall x A$	$\exists x !A$
$+1$	-1	$A \otimes !B$	$?A \parr B$	$A \multimap B$			
-1	-1	$A \,\&\, B$	$A \parr B$	$!A \multimap B$			

5 Γραμμικός συναρτησιακός προγραμματισμός

Ο συναρτησιακός προγραμματισμός ουσιαστικά έχει τις ρίζες του από τη μια στο λογισμό λ και από την άλλη στον *ισομορφισμό Curry-Howard* (βλέπε, για παράδειγμα, [5] και [16] αντίστοιχα).

Ο λογισμός λ αποτελεί το λεγόμενο *λογισμό των ανωνύμων συναρτήσεων*. Ο λογισμός επινοήθηκε από τον Alonzo Church ως τμήμα μιας γενικής θεωρίας συναρτήσεων και λογικής, προοριζομένης ως θεμέλιο των μαθηματικών. Αν και όλο το σύστημα ήταν γενικά προβληματικό, ενάς μικρός πυρήνας του, αυτό που σήμερα ονομάζουμε λογισμό λ, έδωσε νόημα στην έννοια της *υπολογίσιμης συνάρτησης* (computable function). [Κάθε συνάρτηση που είναι υπολογίσιμη από μία μηχανή Turing, μπορεί να παρασταθεί στο λογισμού λ. Βλέπε, για παράδειγμα, [9] για τον ορισμό της μηχανής Turing αλλά και τον ορισμό της υπολογίσιμης συνάρτησης.] Ο λογισμός λ παρέχει δύο βασικές πράξεις: την *εφαρμογή* (application) και την *αφαίρεση* (abstraction). Η παράσταση FA υποδηλώνει την εκτέλεση του προγράμματος F με δεδομένα εισόδου A. Αν $M \equiv M[x]$ είναι μια παράσταση περιέχουσα το x, τότε η $\lambda x.M$ υποδηλώνει την απεικόνιση $x \mapsto M[x]$. Για παράδειγμα, αν η παράσταση $\lambda x.x + 1$ εφαρμοσθεί στον αριθμό 3, παράγει την παράσταση $3 + 1$. Η βασική θεωρία του λογισμού επιτρέπει στις συναρτήσεις να έχουν ότι όρισμα θέλουμε. Για παράδειγμα η συνάρτηση $\lambda x.x$ μπορεί να δεχτεί ως όρισμα ακόμη και τον εαυτό της. Βάζοντας περιορισμό στο είδος των ορισμάτων που μπορεί μια συνάρτηση να δεχτεί, περνάμε σε λογισμούς λ με *τύπους*. Έτσι κάθε παράσταση έχει τον δικό της τύπο· ο οποίος είναι αντικείμενο συντακτικής υφής και ως εκ τούτου μπορεί να ανατεθεί σε κάθε παράσταση. Αυτό, πρακτικά, σημαίνει ότι ένας λογισμός με τύπους είναι δυνατό να είναι λιγότερο *εκφραστικός* σε σχέση με τον λογισμό χωρίς τύπους.

Ο ισομορφισμός Curry-Howard δεν είναι τίποτε άλλο από την ανακάλυψη της αντιστοιχίας μεταξύ *τύπων* και *προτάσεων* (proposition). Ο ισομορφισμός αυτός έχει τις ρίζες του στη *σημασιολογία Heyting*, η οποία σε αντιδιαστολή με την *σημασιολογία Tarski*, ρωτά:

Ποιά είναι η απόδειξη της πρότασης A;

εκεί όπου η δεύτερη ρωτά: Ποτέ η πρόταση A είναι αληθής; Έτσι οι παραστάσεις κάποιας λογικής γίνονται τύποι και οι αποδείξεις γίνονται όροι, ή πιο συγκεκριμένα: μια απόδειξη του A (ως παράσταση) γίνεται όρος τύπου A (ως τύπος). Αν για παράδειγμα θεωρήσει κανείς το τμήμα $(\wedge, \Rightarrow)$ του προτασιακού λογισμού, τότε οι απλές προτάσεις έχουν τύπο T_i, η πρόταση $p \wedge q$ έχει τύπο $T_i \times T_j$ (καρτεσιανό γινόμενο) και η πρόταση $p \Rightarrow q$ έχει τύπο $T_i \rightarrow T_j$ (απεικόνιση). Έχοντας τύπους μπορούμε να βρούμε και τους μηχανισμούς, δηλ. τις συναρτήσεις, που βρίσκουν την απόδειξη, στο πνεύμα του Heyting. Κατ' αυτό το τρόπο υπάρχει άμεση σύνδεση του λογισμού λ με τύπους με την διαισθητική λογική.

Ο Lafont [21] χρησιμοποιώντας τον γραμμικό διαισθητικό λογισμό Gentzen (ουσιαστικά πρόκειται για τον λογισμό που παρουσιάσαμε στην ενότητα 2.2), αλλά και τα αποτελέσματα του ισομορφισμού Curry-Howard, προχώρησε στον σχεδιασμό ενός γραμμικού λογισμού λ. Ο λογισμός αυτός μπορεί να χρησιμοποιηθεί για τον σχεδιασμό μιας συναρτησιακής γλώσσας προγραμματισμού. Το ενδιαφέρον σε μια τέτοια γλώσσα, όσον αφορά την υλοποίηση της, είναι ότι εκτός από τον έλεγχο τύπων που πρέπει να γίνει σε κάποιο πρόγραμμα (βλέπε [28]), θα πρέπει να γίνει και έλεγχος *γραμμικότητας* του προγράμματος ([21, 19, 18]). Από καθαρά προγραμματιστική σκοπιά οι γραμμικές συναρτησιακές γλώσσες έχουν μερικά ενδιαφέροντα χαρακτηριστικά:

- Αρμονική ανάμιξη *οκνυρών* (lazy) και *αυστηρών* (strict) μηχανισμών υπολογισμού.

- Ασφαλή διαχείρηση μνήμης μιας και υπάρχει σαφής περιορισμός στη χρήση μεταβλητών. Ουσιαστικά δεν υπάρχει λόγος για *αποκομιδή σκουπιδιών* (garbage collection).

Αν θεωρήσουμε ένα λογισμό λ με τύπους όπου σε κάθε παράσταση λ δεν υπάρχει *ελεύθερη μεταβλητή* (δηλ. αν έχουμε την παράσταση $\lambda x_1.\lambda x_2 \ldots \lambda x_n.E$, στο E δεν υπάρχουν άλλες μεταβλητές εκτός από τα $x_1, x_2, \ldots, x_n$), τότε οι παραστάσεις λ ονομάζονται *συνδιαστές* (combinators). Ο συνδιαστής $I : A \rightarrow A$, δηλ. με τύπο $A \rightarrow A$, ο οποίος αντιστοιχεί στην παράσταση $\lambda x.x$, γίνεται πλέον μια κανονική συνάρτηση. Το ενδιαφέρον είναι ότι μπορούμε να *ανακαλύψουμε* τους συνδιαστές στην θεωρία κατηγοριών [20]: αν θεωρήσουμε ότι οι προτάσεις είναι *αντικείμενα* μιας κατηγορίας και οι κανόνες *μορφισμοί*, τότε οι αποδείξεις, δηλ. τα μέσα που έχουμε στην διάθεση μας για την απόδειξη ενός θεωρήματος, αντιστοιχούν στη *σύνθεση* μορφισμών (βλ. παράρτημα Α΄). Έτσι ο *κατηγοριακός συνδιαστής* $\phi : A \to B$ είναι μια αναπαράσταση της απόδειξης $A \vdash B$. Με τον τρόπο αυτό μπορούμε εύκολα να ορίσουμε τους βασικούς κατηγοριακούς συνδιαστές μιας λογικής, οι οποίοι μπορούν να χρησιμοποιηθούν ως βάση για τη δημιουργία μιας απλής γλώσσας προγραμματισμού. Αυτή η παρατήρηση οδήγησε στον σχεδιασμό της *κατηγοριακής αφηρημένης μηχανής*, μιας μηχανής σωρών στο πνεύμα των μηχανών SECD[3] με την οποία μπορούμε να υλοποιήσουμε αυτή την γλώσσα. Αν τώρα η λογική μας είναι η διαισθητική γραμμική λογική, τότε μπορούμε να πάρουμε τους αντίστοιχους γραμμικούς συνδιαστές και από εκεί μια *γραμμική αφηρημένη μηχανή* (βλέπε [21]). Μια σημαντική ανακάλυψη, στα πλαίσια αυτής της διεργασίας, αποτελεί η παρακάτω σχέση:

$$!A = A \,\&\, \mathbf{1} \,\&\, (!A \otimes !A)$$

6 Υπάρχει μια μοναδική λογική;

6.1 Ο ενοποιημένος λογισμός LU

Στις αρχές του αιώνα η κατάσταση όσον αφόρα την λογική (αλλά και άλλες επιστήμες όπως η φυσική) ήταν λίγο ως πολύ *ξεκαθαρισμένη*· υπήρχε μία λογική: η κλασική. Η λογική αυτή χρησιμοποιούταν στα μαθηματικά χωρίς κανένα πρόβλημα. Τα ήρεμα νερά της λογικής τα ανατάραξε πρώτος ο Brouwer ο οποίος αμφισβήτησε τα γνωστά μαθηματικά και κατά συνέπεια τη λογική. Μετά από αυτό το πλήγμα στην αξιοπιστία της κλασικής λογικής ακολούθησαν πολλές ακόμη προτάσεις για νέα συστήματα λογικής, άλλα από αυτά δημιουργήθηκαν για να ικανοποιήσουν συγκεκριμένες ανάγκες και άλλα όχι. Αναγνωρίζοντας αυτή την κατάσταση ο Girard προσπάθησε να επαναφέρει κάποια τάξη ενοποιώντας τη λογική. Στο [15] προτείνει ένα *ενοποιημένο λογισμό* LU, του οποίου ο λογισμός Gentzen περιέχει, ως ειδικές περιπτώσεις, την κλασική, την διαισθητική και την γραμμική λογική. Για κάθε λογική ο αντίστοιχος λογισμός Gentzen διαφέρει από τους άλλους στο τρόπο διαχείρισης των όρων του. Στο λογισμό LU ο κάθε όρος χωρίζεται σε δύο ζώνες: μια κλασική και μία γραμμική. (Η διαισθητική λογική προκείπτει από την γραμμική λογική.) Χονδρικά κάθε απορρέουσα πρόταση έχει τη μορφή:

$$\Gamma; \Gamma' \vdash \Delta'; \Delta$$

όπου τα Γ', Δ' συμπεριφέρονται κλασικά και τα Γ, Δ γραμμικά. Ο παραγώμενος λογισμός είναι υπερβολικά μεγάλος, ενώ, όπως παραδέχεται και ο ίδιος ο Girard, δεν αποτελεί παρά έναν έξυπνο επανορισμό της γραμμικής λογικής.

[3] Stack Environment Control Dump.

6.2 Η λογικές LLL, ELL και SLLL

Δεν είναι μόνο το μέγεθος ανασταλτικός παράγοντας στην υιοθέτηση του λογισμού LU αλλά και η ανακάλυψη δύο νέων λογικών: της *ελαφράς (light) γραμμικής λογικής* (LLL) και της *στοιχειώδους (elementary) γραμμικής λογικής* (ELL) [13]. Τα λογικά αυτά συστήματα ήταν αποτέλεσμα της έρευνας πάνω σε μια λογική στην οποία κάθε απόδειξη γίνεται σε *πολυωνυμικό χρόνο*, δηλ. ο αριθμός των βημάτων που απαιτούνται φράσεται από πάνω από κάποιο πολυώνυμο ανάλογο του μήκους της προς απόδειξη πρότασης. Η LLL εισαγάγει ένα νέο σύνδεσμο τον §, για τον οποίο ισχύει $(\S A)^{\perp} = \S A^{\perp}$ και για το λόγο αυτό ονομάζεται *ουδέτερος*. Το ενδιαφέρον με την LLL αποτελεί το γεγονός ότι μπορεί να ειδωθεί ως ένα σύστημα της θεωρίας συνόλων· ενώ μπορεί να ειδωθεί και ως ένας λογισμός λ με τύπους. Για παράδειγμα οι ακέραιοι αριθμοί έχουν τύπο $\forall X.!(X \multimap X) \multimap \S(X \multimap X)$, όπου το X είναι μεταβλητή δευτέρου βαθμού. Αν και η πολυπλοκότητα της LLL είναι *ελαφριά*, η σύνταξη είναι αρκετά *βαριά*. Αυτό οδήγησε τον Asperti στο σχεδιασμό της *υπερελαφριάς (super light) γραμμικής λογικής* (SLLL) [3]. Γενικά μιλώντας όλα αυτά τα συστήματα είναι πολύ νέα και θα πρέπει να γίνει ακόμη αρκετή δουλειά ώστε να φανεί η αξία των. Τέλος, θα πρέπει να αναφερθεί ότι το σύστημα ELL διαφέρει από το LLL στο ότι περιέχει επιπλέον την αρχή $!A \otimes !B \multimap !(A \otimes B)$.

7 Γραμμική λογική και παράλληλος προγραμματισμός

Αποτελούσε πεποίθηση του Girard ό,τι η γραμμική λογική θα έπρεπε να σχετίζεται με τον *παράλληλο προγραμματισμό*, δηλ. την προγραμματιστική μεθοδολογία κατά την οποία διάφορες ενότητες κάποιου προγράμματος εκτελούνται *ταυτόχρονα*. Για τον λόγο αυτό ο Girard ανέπτυξε τα λεγόμενα *δίκτυα αποδείξεων* (proof nets), στα πλαίσια του *πολλαπλασιαστικού* τμήματος της γραμμικής λογικής, δηλ. των συνδέσμων $\otimes$, $⅋$, $\mathbf{1}$ και $\perp$. Τα δίκτυα απόδειξης δεν είναι τίποτε άλλο από μια γραφική παρουσίαση της απόδειξης ενός θεωρήματος απαλλαγμένης από τα *συμφραζόμενα* (context). Για τον ορισμό των δικτύων απόδειξης χρειαζόμαστε να ορίσουμε την έννοια της *δομής απόδειξης* (proof structure), δηλ. ενός γραφήματος το οποίο μπορεί να απαρτίζεται από τα παρακάτω δομικά αντικείμενα :

- *σύνδεσμος*:

$$A \quad A^{\perp}$$

- *αποκοπή*:

$$\underline{A \quad A^{\perp}}$$

- *λογικούς κανόνες*:

$$\frac{A \quad B}{A \otimes B} \qquad \frac{A \quad B}{A ⅋ B} \qquad \overline{\mathbf{1}} \qquad \overline{\perp}$$

Κάθε τύπος θα πρέπει να είναι συμπέρασμα ενός και μόνο ενός κανόνα, ενώ μπορεί να είναι υπόθεση το πολύ ενός κανόνα. Οι τύποι, οι οποίοι δεν είναι υποθέσεις, ονομάζονται *συμπεράσματα της δομής απόδειξης*. Τα δίκτυα απόδειξης είναι δομές απόδειξης τα οποία δομούνται σύμφωνα με τους κανόνες του γραμμικού λογισμού Gentzen:

- Οι σύνδεσμοι είναι δίκτυα απόδειξης.
- Αν το A είναι συμπέρασμα ενός δικτύου απόδειξης ν και το $A^{\perp}$ είναι συμπέρασμα του δικτύου απόδειξης ν', τότε το

$$\begin{array}{cc} \nu & \nu' \\ \vdots & \vdots \\ A & A^{\perp} \\ \hline \end{array}$$

  είναι δίκτυο απόδειξης

- Αν το A είναι συμπέρασμα ενός δικτύου απόδειξης ν και το B είναι συμπέρασμα ενός δικτύου απόδειξης ν', τότε

$$\begin{array}{cc} \nu & \nu' \\ \vdots & \vdots \\ A & B \\ \hline \multicolumn{2}{c}{A \otimes B} \end{array}$$

  είναι δίκτυο απόδειξης.

- Αν το A και B είναι συμπεράσματα του ίδιου δικτύου ν', τότε

$$\begin{array}{cc} \multicolumn{2}{c}{\nu'} \\ \vdots & \vdots \\ A & B \\ \hline \multicolumn{2}{c}{A \parr B} \end{array}$$

- Ο δομή απόδειξης $\overline{\mathbf{1}}$ είναι δίκτυο απόδειξης
- Αν ν είναι δίκτυο απόδειξης, τότε το

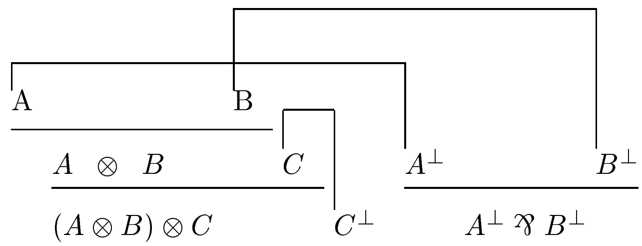
$$\frac{\nu}{\perp}$$

  είναι δίκτυο απόδειξης.

Για παράδειγμα η απόδειξη της κλασικής γραμμικής λογικής

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdash A, A^{\perp} \quad \vdash B, B^{\perp}}{\vdash A \otimes B, A^{\perp}, B^{\perp} \quad \vdash C, C^{\perp}}\otimes}{\vdash (A \otimes B) \otimes C, A^{\perp}, B^{\perp}, C^{\perp}}}{\vdash A^{\perp}, B^{\perp}, (A \otimes B) \otimes C, C^{\perp}}}{\vdash A^{\perp} \parr B^{\perp}, (A \otimes B) \otimes C, C^{\perp}}\parr$$

αντιστοιχεί στο παρακάτω δίκτυο απόδειξης:



Η σημαντικότερη ιδέα όσον αφορά την περιγραφή παραλλήλων συστημάτων είναι αυτή της *επικοινωνίας* μεταξύ μιας *ειδόδου* και μιας *εξόδου*. Αν θεωρήσουμε ως εξόδους της δράσεις και ως εισόδους τις αντιδράσεις, τότε ο κανόνας αποκοπής των δικτύων απόδειξης αντιστοιχεί σε επικοινωνία. Ενώ οι σύνδεσμοι $\parr$ και $\otimes$ αποτελούν τους μηχανισμούς περιγραφής της παραλληλίας. Σε κάθε περίπτωση η αντίδράση του τύπου $(A \ \mathrm{m}\ B)^\perp$ μπορεί να ειδωθεί ως δύο ξεχωριστές αντιδράσεις $A^\perp$ και $B^\perp$· ενώ η δράση $A \ \mathrm{m}\ B$ ως δύο δράσεις A και B. Όμως στην περίπτωση που το m είναι ο σύνδεσμος $\otimes$ δεν υπάρχει *συνεργασία*, ενώ στην αντίθετη περίπτωση υπάρχει συνεργασία (βλ. [12]). Τέλος, ο σύνδεσμος ! αντιστοιχεί για τις μεν δράσεις στην ιδέα της *εγγραφής* σε κάποιο σταθερό καταχωρητή, ενώ για τις αντιδράσεις σε ακαθόριστο αριθμό αυτών.

Χρησιμοποιώντας ως βάση τα δίκτυα αποδείξεων, ο Lafont δημίουργησε αρχικά τα *δίκτυα αλληλεπίδρασης* (interaction nets) [22] και στην συνέχεια τους *συνδιαστές αλληλεπίδρασης* (interaction combinators) [23], τα οποία αποτελούν μια νέα μορφή γλώσσας προγραμματισμού.

Ο Abramsky δημιουργώντας ένα νέο μηχανισμό, αυτό των *εκφράσεων απόδειξης*, δίνει μια διαφορετική μορφή στην ερμηνεία των παραλλήλων συστημάτων μέσω της γραμμικής λογικής [2]. Ουσιαστικά αυτό που κάνει ο Abramsky είναι η αντικατάσταση των δικτύων αποδείξεων με τις εκφράσεις απόδειξης. Όμως τί είναι μια έκφραση απόδειξης; Αποτελούν έκφράση της κλασσικής γραμμικής λογικής ως μια θεωρία τύπων με τα παρακάτω χαρακτηριστικά:

- Ένα σύνολο $\mathcal{N}$ ονομάτων, τα οποία σημειώνουμε ως x, y και z, ενώ τα $\bar{x}$, $\bar{y}$ και $\bar{z}$ αντιστοιχούν σε πεπερασμένους καταλόγους ονομάτων.

- Οι όροι είναι της μορφής:

$$
\begin{array}{cccc}
x & & & \\
* & \odot & & \\
t \otimes u & t \parr u & & \\
\mathrm{inl}(t) & \mathrm{inr}(u) & \bar{x}(P \| Q) & \\
?t & \_ & t@u & \bar{x}(P)
\end{array}
$$

- Οι *συνεξισώσεις* έχουν τη μορφή $t \perp u$, όπου τα t και u είναι όροι, ενώ τα Θ και Ξ αντιστοιχούν σε καταλόγους συνεξισώσεων.

- Οι εκφράσεις απόδειξης έχουν τη μορφή $\Theta; \bar{t}$, ενώ τα τα P και Q αντιστοιχούν σε εκφράσεις απόδειξης.

Τα διάφορα αυτά σύμβολα χρησιμοποιούνται για να γραφεί ο λογισμός Gentzen της γραμμικής λογικής σε μια νέα μορφή, έτσι ο παρακάτω κανόνας αντιστοιχεί στον κανόνα αποκοπής:

$$\frac{\vdash \Theta; \Gamma, t : A \qquad \vdash \Xi; \Delta, u : A^{\perp}}{\vdash \Theta, \Xi, t \perp u; \Gamma, \Delta}$$

ενώ ο κανόνας *ένα* εκγράζεται ως

$$\overline{\vdash; * : \mathbf{1}}$$

Το πιο ενδιαφέρον στοιχείο της κατασκευής αποτελεί το γεγονός ό,τι Abramsky δίνει μια ερμηνεία των εκφράσεων απόδειξης χρησιμοποιώντας ως βάση την *χημική αφηρημένη μηχανή* των Berry και Boudol [6].

Ο Abramsky στο [1] παρουσιάζει μια αρκετά σπουδαία ιδέα που αναφέρεται στην αντίληψη ό,τι οι αποδείξεις μπορούν να *λειτουργούν* ως *διεργασίες* (process). Με βάση αυτή την ιδέα ο Abramsky παρουσιάζει μια μετάφραση αποδείξεων της γραμμικής λογικής στον λογισμό π του Robin Milner [26] (βλ. παράρτημα Β΄ για μια σύντομη περιγραφή). Η μετάφραση αυτή αποτέλεσε τη βάση για μια άλλη μετάφραση από τους Bellin και Scott του *σύγχρονου* λογισμού π στη γραμμική λογική [4]. Αντιθέτως ο Miller παρουσιάζει μια μετάφραση του λογισμού π στη γραμμική λογική [25].

8 Αυτόματη απόδειξη θεωρημάτων

Η αυτόματη απόδειξη θεωρημάτων αποτελεί ένα σημαντικό τομέα εφαρμογής της μαθηματικής λογικής. Αποτελεί δε το μηχανισμό εκείνο με τον οποίο μπορεί κανείς να αποδείξει μια πρόταση σε μια συγκεκριμένη λογική (για περισσότερες πληροφορίες βλ. [10]). Ο Tanel Tammet πρωτοπόρησε στην έρευνα για την αυτόματη απόδειξη θεωρημάτων στη γραμμική λογική [29]. Ο τομέας αποτελεί ανοικτό αντικείμενο έρευνας.

ΠΑΡΑΡΤΗΜΑ

Α′ Κατηγοριακά μοντέλα λογικής

Μια *κατηγορία* (category) $\mathcal{C}$ αποτελείται από μια συλλογή *αντικειμένων* $\mathcal{C}_0$ και μια συλλογή *μορφισμών* $\mathcal{C}_1$ οι οποίες ικανοποιούν τις παρακάτω συνθήκες:

- Κάθε μορφισμός έχει ένα *πεδίο ορισμού* και ένα *πεδίο τιμών*· αν έχουμε το μορφισμό $f : X \to Y$, τότε το X είναι το πεδίο ορισμού ($X = \mathrm{dom}(f)$) και το Y το πεδίο τιμών ($Y = \mathrm{codom}(f)$). Ο μορφισμός f μπορεί να σημειωθεί και με τον εξής τρόπο: $X \xrightarrow{f} Y$.

- Αν έχουμε δύο μορφισμούς f και g τέτοιους ώστε $\mathrm{codom}(f) = \mathrm{dom}(g)$, η *σύνθεση* των f και g, η οποία σημειώνεται ως gf, ορίζεται και έχει πεδίο ορισμού το $\mathrm{dom}(f)$ και πεδίο τιμών το $\mathrm{codom}(g)$.

- Η σύνθεση μορφισμών είναι προσεταιριστική, δηλ. αν έχουμε τους μορφισμούς $f : X \to Y$, $g : Y \to Z$ και $h : Z \to W$, τότε $h(gf) = (hg)f$.

- Για κάθε αντικείμενο X υπάρχει ένας ταυτοτικός μορφισμός id_{X}, ο οποίος ικανοποιεί τις παρακάτω σχέσεις:

  - $\mathrm{id}_{\mathrm{X}} g = g$ για κάθε $g : Y \to X$ και
  - $f \mathrm{id}_{\mathrm{X}} = f$ για κάθε $f : X \to Y$.

Αν έχουμε δύο κατηγορίες $\mathcal{C}$ και $\mathcal{D}$, ένας διακατηγοριακός μορφισμός (functor) $F : \mathcal{C} \to \mathcal{D}$ αποτελείται από τις πράξεις $F_0 : \mathcal{C}_0 \to \mathcal{D}_0$ και $F_1 : \mathcal{C}_1 \to \mathcal{D}_1$, για τις οποίες ισχύει ότι για κάθε $f : X \to Y$, $F_1(f) : F_0(X) \to F_0(Y)$ και επιπλέον

- για κάθε σύνθεση gf έχουμε $F_1(gf) = F_1(g)F_1(f)$ και

- $F_1(\mathrm{id}_{\mathrm{X}}) = \mathrm{id}_{F_0(\mathrm{X})}$ για κάθε $\mathrm{X} \in \mathcal{C}_0$.

Αν έχουμε δύο κατηγορίες $\mathcal{C}$ και $\mathcal{D}$ ορίζουμε ως γινόμενο $\mathcal{C} \times \mathcal{D}$ μια κατηγορία η οποία έχει ως αντικείμενα τα ζεύγη $(C, D) \in \mathcal{C}_0 \times \mathcal{D}_0$ και μορφισμούς $(C, D) \to (C', D')$ τα ζεύγη (f, g) με $f : C \to C'$ στην $\mathcal{C}$ και $g : D \to D'$ στην $\mathcal{D}$.

Ένα αντικείμενο X μιας κατηγορίας $\mathcal{C}$ ονομάζεται *τερματικό* αντικείμενο, αν για κάθε άλλο αντικείμενο Y της $\mathcal{C}$ υπάρχει ακριβώς ένας μορφισμός $f \in \mathcal{C}_1$ από το Y στο X.

Έστω $\mathcal{C}$ μια κατηγορία τέτοια ώστε για κάθε ζεύγος αντικειμένων (X, Y) η συλλογή $\mathcal{C}(X, Y)$ μορφισμών από το X στο Y είναι σύνολο, τότε η κατηγορία καλείται *τοπικά μικρή* (locally small). Αν έχουμε την κατηγορία $\mathcal{C}$ τότε η κατηγορία $\mathcal{C}^{\mathrm{op}}$ έχει τα ίδια αντικείμενα και μορφισμούς με την $\mathcal{C}$, μονο που οι μορφισμοί της $\mathcal{C}^{\mathrm{op}}$ έχουν αντίθετη φορά. Π.χ, αν ο μορφισμός $f : X \to Y$ ανήκει στην $\mathcal{C}$, τότε ο μορφισμός $\bar{f} : Y \to X$ ανήκει στην $\mathcal{C}^{\mathrm{op}}$. Αν η $\mathcal{C}$

είναι τοπικά μικρή, τότε υπάρχει ένας διακατηγοριακός μορφισμός $\mathrm{Hom}(-,-) : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathrm{Set}$ ο οποίος αναθέτει το ζεύγος αντικείμένων (A, B) της $\mathcal{C}$ στο συνόλο $\mathcal{C}(A, B)$. (Set είναι η κατηγορία που έχει ως αντικείμενα όλα τα σύνολα και ως μορφισμούς τις συναρτήσεις μεταξύ των.)

Λέμε ότι το παρακάτω διάγραμμα αντιμετατίθεται αν $hk = g$.

$$\begin{tikzcd} A \arrow[rr, "k"] \arrow[dr, "g"'] & & B \arrow[dl, "h"] \\ & C & \end{tikzcd}$$

Το κατηγοριακό γινόμενο των αντικειμένων X και Y, μιας κατηγορίας $\mathcal{C}$, είναι ένα αντικείμενο $X \times Y$ μαζί με δύο μορφισμούς *προβολής* $\pi_0 : X \times Y \to X$ και $\pi_1 : X \times Y \to Y$, έτσι ωστε για κάθε αντικείμενο Z και μορφισμούς $f_0 : Z \to X$ και $f_1 : Z \to Y$, υπάρχει ακριβώς ένας μορφισμός $h : Z \to X \times Y$ ώστε το παρακάτω διάγραμμα να αντιμετατίθεται

$$\begin{tikzcd} & Z \arrow[dl, "f_0"'] \arrow[d, dotted, "h"'] \arrow[dr, "f_1"] & \\ X & X \times Y \arrow[l, "\pi_0"'] \arrow[r, "\pi_1"] & Y \end{tikzcd}$$

δηλ. $f_i = \pi_i h$. Το διακεκομένο βέλος χρησιμοποιείται για μορφισμούς που κάνουν το διάγραμμα να αντιμετατίθεται και των οποίων θεωρούμε την ύπαρξη.

Το αντικείμενο $X + Y$ μαζί με τους μορφισμούς $\iota_0 : X \to X + Y$ και $\iota_1 : Y \to X + Y$ αποτελούν το κατηγοριακό *άθροισμα* των X και Y, αν για οποιοδήποτε αντικείμενο Z και μορφισμούς $f_0 : X \to Z$ και $f_1 : Y \to Z$, υπάρχει ακριβώς ένας μορφισμός $h : X + Y \to Z$ τέτοιος ώστε το παρακάτω διάγραμμα να αντιμετατίθεται:

$$\begin{tikzcd} X \arrow[r, "\iota_0"] \arrow[dr, "f_0"'] & X + Y \arrow[d, dotted, "h"] & Y \arrow[l, "\iota_1"'] \arrow[dl, "f_1"] \\ & Z & \end{tikzcd}$$

Υποθέστε ότι για όλα τα αντικείμενα X και Y μιας κατηγορίας $\mathcal{C}$, υπάρχει το γινόμενο $X \times Y$ και ότι υπάρχει κάποιο τερματικό αντικείμενο αυτής της κατηγορίας. Η κατηγορία αυτή ονομάζεται *κλειστή καρτεσιανή κατηγορία* (κκκ) αν και μόνο αν για κάθε ζεύγος αντικειμένων X και Y υπάρχει το *εκθετικό* αντικείμενο Y^X μαζί με το μορφισμό υπολογισμού $\eta : Y^X \times X \to Y$, έτσι ώστε για κάθε αντικείμενο Z και μορφισμό $f : Z \times X \to Y$, να υπάρχει ακριβώς ένας μορφισμός $h : Z \to Y^X$ ο οποίος να κάνει το παρακάτω διάγραμμα να αντιμετατίθεται:

$$\begin{tikzcd}
Y^X \times X \arrow[r, "\eta"] & Y \\
Z \times X \arrow[u, dotted, "h \times \mathrm{id}_X"] \arrow[ur, "f"'] &
\end{tikzcd}$$

Το κατηγοριακό μοντέλο της διαισθητικής λογικής είναι η κλειστή καρτεσιανή κατηγορία.

Μια *συμμετρική μονοϊδής κατηγορία* είναι μια κατηγορία $\mathcal{C}$ εφοδιασμένη με ένα διακατηγοριακό μορφισμό $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ και ένα αντικείμενο $\mathbf{1} \in \mathcal{C}$ τέτοιο ώστε

$$X \otimes (Y \otimes Z) \cong (X \otimes Y) \otimes Z$$
$$X \cong \mathbf{1} \otimes X, \qquad X \otimes Y \cong Y \otimes X$$

Μια συμμετρική μονοϊδής κατηγορία είναι κλειστή αν

$$\mathrm{Hom}(X \otimes A, Y) \cong \mathrm{Hom}(X, A \multimap Y)$$

Ένα κατηγοριακό μοντέλο της γραμμικής διαισθητικής λογικής, χωρίς τον τελεστή $\parr$, είναι απλώς μια συμμετρική μονοϊδής κλειστή κατηγορία $(\mathcal{C}, \otimes, \mathbf{1}, \multimap)$ με πεπερασμένα γινόμενα $(\&, \top)$ και αθροίσματα $(\oplus, \mathbf{0})$. Για μια εκτεταμένη μελέτη των διαφόρων κατηγοριακών μοντέλων της γραμμικής διαισθητικής λογικής βλέπε [7].

Β΄ Λογισμός π

Στο παράρτημα αυτό παρουσιάζουμε εν συντομία την πιο απλή μορφή του λογισμού. Ο λογισμός π είναι ένα μοντέλο παράλληλης επεξεργασίας το οποίο βασίζεται στην έννοια της *ονομασίας*. Η πιο πρωτόγονη οντότητα του λογισμού είναι το *όνομα*. Τα ονόματα, τα οποία μπορούν να είναι άπειρα, σημειώνονται ως $x, y, \ldots \in \mathcal{X}$ και δεν έχουν κάποια εσωτερική δομή. Η άλλη βασική οντότητα που παρέχει ο λογισμός είναι η *διεργασία* (process). Οι διεργασίες συμβολίζονται με $P, Q, \ldots \in \mathcal{P}$, χτίζονται δε από ονόματα και η σύνταξη τους έχει ως εξής:

$$P ::= \sum_{i \in I} \pi_i . P_i \;\Big|\; P|Q \;\Big|\; !P \;\Big|\; (\nu x)P$$

όπου I είναι ένα πεπερασμένο σύνολο δεικτών· ειδικά στην περίπτωση που $I = \emptyset$ το αντίστοιχο άθροισμα γράφεται ως $\mathbf{0}$. Σε κάθε όρο $\pi.P$ το πρόθεμα π αντιπροσωπεύει μια ατομική δράση (atomic action), η οποία είναι το πρώτο πράγμα που εκτελεί η διεργασία P. Υπάρχουν δύο βασικές μορφές προθεμάτων:

$x(y)$ που σημαίνει: *εισαγωγή κάποιου ονόματος* (y) *μέσω της σύνδεσης* x.

$\bar{x}y$ που σημαίνει: *εξαγωγή του ονόματος* y *μέσω της σύνδεσης* x.

Σε κάθε περίπτωση ονομάζουμε το x *υποκείμενο* και το y *αντικείμενο*. Το άθροισμα $\sum_i \pi_i.P_i$ αντιπροσωπεύει μια διεργασία που δύναται να συμμετέχει σε μια μόνο αλλά μόνο μία επιλογή εποικοινωνιών. Η διεργασία $P|Q$ απλά σημαίνει ότι τα P και Q είναι ταυτόχρονα ενεργά· έτσι μπορούν να δρουν ανεξάρτητα αλλά και να επικοινωνούν. Η διεργασία

$$!P = \underbrace{P|P|\cdots|P}_{\nu \text{ αντίγραφα}}$$

αντιστοιχεί στην *κλωνοποίηση* της διεργασίας P. Προσέξτε ότι αυτό δεν σημαίνει ότι μια διεργασία μπορεί να έχει άπειρα αντίγραφα. Τέλος, η διεργασία $(\nu x)P$ περιορίζει τη χρήση του ονόματος x στην P μόνο.

Η σχέση $\longrightarrow$ επιτρέπει το μετασχηματισμό της διεργασίας P στη διεργασία P' σε ένα υπολογιστικό βήμα: $P \longrightarrow P'$. Κάθε υπολογιστικό βήμα αποτελείται από μια αλληλεπίδραση μεταξύ κανονικών όρων. Ο πρώτος κανόνας αναγωγής της σχέσης $\longrightarrow$ είναι αυτός της *επικοινωνίας*:

$$(\cdots + x(y).P)|(\cdots + \bar{x}z.Q) \longrightarrow P\{z/y\}|Q$$

Ο κανόνας αυτός ουσιαστικά αποτελεί το μόνο αξίωμα της σχέσης $\longrightarrow$. Οι υπόλοιποι κανόνες αναγωγής ακολουθούν:

$$\frac{P \longrightarrow P'}{P|Q \longrightarrow P'|Q}\ PAR \qquad \frac{P \longrightarrow P'}{(\nu x)P \longrightarrow (\nu x)P'}\ RES$$

$$\frac{Q \equiv P \quad P \longrightarrow P' \quad P' \equiv Q'}{Q \longrightarrow Q'}\ STRUCT$$

Ο Milner χρησιμοποιώντας τον λογισμό π ορίζει αριθμούς αλλά και απλές πράξεις στο στυλ του λογισμού λ. Έτσι ένας θετικός αριθμός ορίζεται ως εξής:

$$\underline{n}(xz) \stackrel{def}{=} (\bar{x}.)^n\bar{z}.$$

(Για περισσότερες πληροφορίες βλ. [26].)

Αναφορές

[1] Samson Abramsky. Proofs as Processes. Διαλέξη που δώθηκε στο Τμήμα Επιστήμης Η/Υ του Πανεπιστημίου του Edinburgh., 1992.

[2] Samson Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111:3–57, 1993.

[3] Andrea Asperti. Special Light Linear Logic. `http://hypatia.dcs.qmw.ac.uk/authors/A/AspertiA/SLLL.ps.gz`, 1997.

[4] G. Bellin και P.J. Scott. On the π-calculus and linear logic. *Theoretical Computer Science*, 135:11–65, 1994.

[5] Chantal Berline. From computation to foundations via functions and application: The λ-calculus and its webbed models. Τεχνική Αναφορά υπ. αρίθμ. 63, Équipe de Logique Mathématique, Université Paris VII, Ιούλιος 1997.

[6] G. Berry και G. Boudol. The chemical abstract machine. *Theoretical Computer Science*, 96:217–248, 1992.

[7] G. M. Bierman. What is a categorical model of intuitionistic linear logic? Στο *Proceedings of Conference on Typed lambda calculus and Applications*, επιμελητής: M. Dezani. Springer-Verlag LNCS 902, Απρίλιος 1995.

[8] Alonzo Church. *Introduction to Mathematical Logic.* Princeton University Press, 1996. Δέκατη εκτύπωση.

[9] Martin Davis. *Computability and Unsolvability.* Dover Publications, New York, 1982.

[10] David A. Duffy. *Principles of Automated Theorem Proving.* John Wiley & Sons, Chichester, UK, 1991.

[11] Anthony J. Field και Peter G. Harrison. *Functional Programming.* Addisson-Wesley, Wokingham, England, 1988.

[12] J. Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[13] J. Y. Girard. Light linear logic. Στο *Logic and Computational Complexity*, επιμελητής: D. Leivant, τόμος 960 στο *LNCS*, σελίδες 145–176. Springer-Verlag, 1995.

[14] J. Y. Girard. Linear logic: Its syntax and semantics. Στο *Advances in Linear Logic*, επιμελητές: J. Y. Girard, Y. Lafont και L. Regnier, σελίδες 1–42. Cambridge University Press, 1995. Πρακτικά σχολείου στη γραμμική λογική, New York, Ιούνιος 1993.

[15] Jean Yves Girard. On the unity of logic. *Annals of Pure and Applied Logic*, 59:201–217, 1993.

[16] Jean Yves Girard, Yves Lafont και P. Taylor. *Proofs and Types.* Cambridge Tracts in Theoretical Computer Science 7. Cambridge University Press, 1988.

[17] Arend Heyting. *Intuitionism: An Introduction.* North-Holland, 1976.

[18] Pétur Hilmarsson και Apostolos Syropoulos. Demokritos: A Linear Functional Language. `ftp://obelix.ee.duth.gr/pub/docs/dem.ps.gz`, 1992.

[19] S. Holmström. Linear functional programming. Στο *Proceedings of the Workshop on the Implementation of Lazy Functional Languages*, σελίδες 13–32, Aspanäes, Sweden, Σεπτέμβριος 1988.

[20] van Oosten Jaap. Basic Category Theory. Έκδοση του ιδρύματος BRICS, BRICS, Department of Computer Science, University of Aarhus, DK-8000, Aarhus C, Denmark, Ιανουάριος 1995.

[21] Yves Lafont. The linear abstract machine. *Theoretical Computer Science*, 59:157–180, 1988. Διορθώσεις στο κείμενο παρουσιάσθηκαν στον τόμο 62 (1988), σελίδες 327–328.

[22] Yves Lafont. Interaction nets. Στο *Seventeenth Annual Symposium on Principles of Programming Languages*, σελίδες 95–108, San Francisco, California, 1990. ACM Press.

[23] Yves Lafont. Interaction combinators. *Information and Computation*, 137:69–101, 1997.

[24] Patrick Lincoln. Linear logic. *ACM SIGACT Notices*, 23(2):29–37, 1992.

[25] Dale Miller. The π-calculus as a theory in linear logic: Preliminary results. Στο *Proceedings of the Workshop on Extensions of Logic Programming*, επιμελητές: E. Lamma και P. Mello, σελίδες 242–265. Springer-Verlag LNCS 660, 1992.

[26] R. Milner. The polyadic π-calculus: a tutorial. Στο *Logic and Algebra of Specifications*, επιμελητές: F. L. Bauer, W. Brauer και H. Schwichtenberg. Springer-Verlag, 1993.

[27] José Luis Rivera-Noriega. Προσωπική επικοινωνία, Φεβρουάριος 1998.

[28] Apostolos Syropoulos. A note on type checking linear functional languages. *SIGPLAN Notices*, 31:80–83, Δεκέμβριος 1996.

[29] T. Tammet. Proof strategies in linear logic. *Journal of Automated Reasoning*, 12:273–304, 1994.

Κολοφών

Το κείμενο αυτό στοιχειοθετήθηκε με την τελαυταία έκδοση του LaTeX και την ελληνική επιλογή του πακέτου babel, η οπόια αποτελεί προσωπική δουλειά

του συγγραφέα. Επιπλέον χρησιμοποιήθηκε το πακέτο proof του Makoto Tatsuta και το πακέτο pb-diagram του Paul Burchard για το σχεδιασμό των αντιμεταθετικών διαγραμμάτων.