Κεφάλαιο 2

Λάμβδα λογισμός

Ο λ-λογισμός αναπτύχθηκε αρχικά από τον Alonso Church στις αρχές της δεκαετίας του 1930, πολύ πριν αρχίσουν να χρησιμοποιούνται οι ηλεκτρονικοί υπολογιστές. Ήταν μέρος μιας γενικότερης θεωρίας, αντίστοιχης της θεωρίας συνόλων, που αποσκοπούσε στη θεμελίωση των μαθηματικών και της λογικής [Chur32, Chur33]. Παρά το γεγονός ότι η γενική θεωρία ήταν ασυνεπής, όπως αποδείχθηκε αργότερα [Klee35], το τμήμα της που ασχολείται με τις συναρτήσεις βρήκε σημαντικότατες εφαρμογές στην πληροφορική, κυρίως μετά το 1960. Το τμήμα αυτό έγινε γνωστό με το όνομα *λάμβδα-λογισμός* (lambda calculus), ή σε συντομογραφία *λ-λογισμός* (λ-calculus).

Παρά την απλότητά του, είναι αξιοσημείωτο ότι ο λ-λογισμός είναι ένα πλήρες υπολογιστικό μοντέλο. Από τη δεκαετία του 1930 ήταν ήδη γνωστά δυο πολύ σημαντικά αποτελέσματα:

- Όλες οι αναδρομικές συναρτήσεις μπορούν να παρασταθούν στο λ-λογισμό [Klee35].
- Ως υπολογιστικό μοντέλο, ο λ-λογισμός είναι ισοδύναμος με τη μηχανή Turing [Turi37].

Η μηχανή Turing αποτέλεσε τη βάση των *υπολογιστών von Neumann*, στους οποίους ανήκουν οι σημερινοί υπολογιστές, και συγχρόνως οδήγησε στη δημιουργία των πρώτων γλωσσών προστακτικού προγραμματισμού, όπως η FORTRAN και η ALGOL. Από την άλλη πλευρά:

- Ο λ-λογισμός και παραλλαγές αυτού οδήγησαν στο σχεδιασμό νέων αρχιτεκτονικών υπολογιστών. Παραδείγματα τέτοιων υπολογιστών αποτελούν οι *μηχανές αναγωγής* (reduction machines) και οι *υπολογιστές ροής δεδομένων* (data-flow computers), που όταν πρωτοδημιουργήθηκαν ήταν σε θέση να εκτελούν σχεδόν αποκλειστικά προγράμματα γραμμένα σε κάποια διάλεκτο του λ-λογισμού.
- Ο λ-λογισμός αποτέλεσε τη βάση για τη δημιουργία του *συναρτησιακού προγραμματισμού* (functional programming). Εμπνευσμένος από το λ-λογισμό, ο John McCarthy σχεδίασε τη γλώσσα προγραμματισμού LISP στα τέλη της δεκαετίας του 1950. Αυτή με τη σειρά της έδωσε το έναυσμα για τη δημιουργία πολλών άλλων γλωσσών συναρτησιακού προγραμματισμού, όπως η Scheme, η ML, η Miranda και η Haskell.

Οι υπολογιστές που βασίστηκαν στο λ-λογισμό και ο συναρτησιακός προγραμματισμός δεν έτυχαν της ίδιας ευρείας αποδοχής που γνώρισαν οι υπολογιστές von Neumann και ο προστακτικός προγραμματισμός. Αυτό οφείλεται κυρίως στο γεγονός ότι τόσο οι πρώτες μηχανές αναγωγής όσο και οι πρώτες υλοποιήσεις συναρτησιακών γλωσσών είχαν επιδόσεις σημαντικά χειρότερες από αυτές των ανταγωνιστών τους. Με το πέρασμα του χρόνου, οι μηχανές von Neumann δείχνουν σήμερα να έχουν κυριαρχήσει πλήρως. Από το 1970 όμως, έχουν προταθεί νέες τεχνικές υλοποίησης των συναρτησιακών γλωσσών, όπως η αναγωγή γράφων (graph reduction). Η χρήση τέτοιων τεχνικών στους μεταγλωττιστές των μοντέρνων συναρτησιακών γλωσσών έδωσε μια νέα ώθηση στο συναρτησιακό προγραμματισμό που, από το 1990 και μετά, έχει αποκτήσει αξιόλογη θέση στα ερευνητικά δρώμενα.

Η κύρια όμως συμβολή του λ-λογισμού στην επιστήμη της πληροφορικής δεν έγκειται στη χρήση του ως προγραμματιστικού μοντέλου. Κατά τη δεκαετία του 1960, οι Christopher Strachey, Peter J. Landin και άλλοι παρατήρησαν ότι ο λ-λογισμός είναι ιδιαίτερα πρόσφορος ως συμβολισμός για την περιγραφή σημασιολογικών ιδιοτήτων των γλωσσών προγραμματισμού. Επίσης, παρατήρησαν ότι πολλά προβλήματα σχεδίασης και υλοποίησης των γλωσσών προγραμματισμού, και κυρίως αυτά που σχετίζονται με το μηχανισμό κλήσης υποπρογραμμάτων και τη δομή του συστήματος τύπων, μπορούν να απομονωθούν και να μελετηθούν ευκολότερα στο λ-λογισμό. Οι διαπιστώσεις αυτές και η προσπάθεια για την εύρεση σημασιολογικών μοντέλων του λ-λογισμού, οδήγησαν αργότερα στη διατύπωση της *θεωρίας πεδίων* (domain theory) και στη θεμελίωση του ερευνητικού πεδίου της *σημασιολογίας γλωσσών προγραμματισμού* (programming language semantics).

Στην αρχική του μορφή, όπως διατυπώθηκε από τον Church, ο λ-λογισμός είναι μια θεωρία *χωρίς τύπους*. Κάθε έκφραση θεωρείται ως μια συνάρτηση και μπορεί να εφαρμοστεί σε οποιαδήποτε έκφραση, συμπεριλαμβανομένου και του εαυτού της. Αργότερα, όμως, προτάθηκαν παραλλαγές του λ-λογισμού με τύπους [Curr34, Chur40]. Στη συνέχεια, ο όρος "λ-λογισμός" χωρίς περαιτέρω προσδιορισμούς θα αναφέρεται στη μορφή χωρίς τύπους.

2.1 Μια διαισθητική εισαγωγή

Ο λ-λογισμός είναι μια θεωρία συναρτήσεων. Σε αυτήν, δύο είναι οι κύριες λειτουργίες:

- Η *εφαρμογή* μιας συνάρτησης F πάνω σε ένα όρισμα A, που συμβολίζεται με $F\,A$.
- Η *αφαίρεση*. Έστω ότι x είναι μια μεταβλητή και $E[x]$ είναι μια έκφραση που εξαρτάται από τη μεταβλητή x (δηλαδή περιέχει το x). Τότε η έκφραση $\lambda x.\,E[x]$ συμβολίζει τη συνάρτηση

  $$x \;\mapsto\; E[x]$$

  που όταν δέχεται ως όρισμα μια τιμή v επιστρέφει ως αποτέλεσμα την τιμή $E[v]$. Η μεταβλητή x δεν είναι απαραίτητο να εμφανίζεται στην έκφραση $E[x]$. Αν αυτό δε συμβαίνει, τότε η $\lambda x.\,E[x]$ είναι μια σταθερή συνάρτηση.

Για να γίνει αντιληπτός ο τρόπος συνεργασίας της αφαίρεσης και της εφαρμογής, θα χρησιμοποιήσουμε ως παράδειγμα συναρτήσεις από τα μαθηματικά. Η αφαίρεση

$$\lambda x.\,x^2 - 3x + 2$$

συμβολίζει μια (ανώνυμη) συνάρτηση που σε κάθε τιμή x απεικονίζει την τιμή $x^2 - 3x + 2$. Αν αυτή η συνάρτηση εφαρμοστεί στο όρισμα 8, προκύπτει το ακόλουθο αποτέλεσμα:[1]

$$(\lambda x.\, x^2 - 3x + 2)\, 8 \;=\; 8^2 - 3 \cdot 8 + 2 \;=\; 42$$

Κατά την εφαρμογή της συνάρτησης, δηλαδή, η τιμή του ορίσματος 8 αντικαθιστά την παράμετρο x στον τύπο ορισμού της συνάρτησης.

Η αφαίρεση $\lambda x.\, E[x]$ *δεσμεύει* τη μεταβλητή x μέσα στην έκφραση $E[x]$. Μια μεταβλητή που δεν είναι δεσμευμένη ονομάζεται *ελεύθερη*. Για παράδειγμα, στην έκφραση

$$\lambda x.\, x^2 - 3y + 2$$

η μεταβλητή x είναι δεσμευμένη ενώ η y είναι ελεύθερη. Είναι όμως δυνατό σε μια έκφραση κάποια εμφάνιση μιας μεταβλητής να είναι δεσμευμένη και κάποια άλλη εμφάνιση της ίδιας μεταβλητής να είναι ελεύθερη. Για παράδειγμα, στην έκφραση

$$(\lambda x.\, x^2 - 3y + 2)\, (4x + 1)$$

η πρώτη εμφάνιση του x (στο x^2) είναι δεσμευμένη, γιατί βρίσκεται στο εσωτερικό της αφαίρεσης, ενώ η δεύτερη εμφάνιση του x (στο $4x$) είναι ελεύθερη.

Παρόμοια έννοια δέσμευσης συναντάται και στα μαθηματικά. Για παράδειγμα, στο ολοκλήρωμα

$$\int_{-\pi}^{\pi} \frac{\sin x + \cos y}{\cos x - \sin y}\, dx$$

η μεταβλητή x στο εσωτερικό του ολοκληρώματος είναι δεσμευμένη και, κατά συνέπεια, η τιμή του ολοκληρώματος δεν εξαρτάται από την τιμή του x έξω από αυτό. Αντίθετα, η μεταβλητή y είναι ελεύθερη και άρα η τιμή του ολοκληρώματος εξαρτάται από την τιμή που έχει το y έξω από αυτό.

Στο λ-λογισμό, όπως άλλωστε και στα μαθηματικά, ιδιαίτερη θέση έχει η έννοια της αντικατάστασης μιας μεταβλητής με κάποια έκφραση. Όπως ήδη αναφέρθηκε, η αντικατάσταση χρησιμοποιείται στην εφαρμογή συναρτήσεων: στην εφαρμογή $(\lambda x.\, E[x])\, A$ το όρισμα A αντικαθιστά τις εμφανίσεις της μεταβλητής x στην έκφραση $E[x]$. Η δέσμευση μεταβλητών όμως είναι δυνατό να δημιουργήσει προβλήματα κατά την αντικατάσταση, όπως φαίνεται στα επόμενα παραδείγματα από τα μαθηματικά:

- Σε μια έκφραση δεν πρέπει να επιτρέπεται η αντικατάσταση μεταβλητών που είναι δεσμευμένες. Για παράδειγμα, στο παραπάνω ολοκλήρωμα, η (εσφαλμένη) αντικατάσταση της μεταβλητής x με την τιμή 42 θα οδηγούσε στο παράλογο αποτέλεσμα

$$\int_{-\pi}^{\pi} \frac{\sin 42 + \cos y}{\cos 42 - \sin y}\, d42$$

[1] Ας σημειωθεί ότι οι παρενθέσεις χρησιμοποιούνται μόνο για την αποφυγή παρερμηνειών στη δομή των εκφράσεων. Αντίθετα με τα μαθηματικά, όπου η εφαρμογή μιας συνάρτησης f σε ένα όρισμα a συμβολίζεται συνήθως με $f(a)$, η εφαρμογή στο λ-λογισμό συμβολίζεται χωρίς τη χρήση παρενθέσεων.

- Η αντικατάσταση της μεταβλητής x με μια έκφραση A δεν πρέπει να προκαλεί τη δέσμευση μεταβλητών της A που ήταν προηγουμένως ελεύθερες. Για παράδειγμα, στο προηγούμενο ολοκλήρωμα, έστω ότι θέλουμε να αντικαταστήσουμε την ελεύθερη μεταβλητή y με την έκφραση $3x+1$. Στην έκφραση αυτή το x είναι ελεύθερο, θα μπορούσε π.χ. να έχει την τιμή 4, και σε αυτή την περίπτωση η έκφραση $3x+1$ θα είχε την τιμή 13. Η (εσφαλμένη) όμως αντικατάσταση του y με αυτή την έκφραση θα είχε ως αποτέλεσμα το ολοκλήρωμα

  $$\int_{-\pi}^{\pi} \frac{\sin x + \cos(3x+1)}{\cos x - \sin(3x+1)}\, dx$$

  στο οποίο όλες οι εμφανίσεις της μεταβλητής x είναι δεσμευμένες. Στο εσωτερικό του ολοκληρώματος, η έκφραση $3x+1$ παύει να έχει την τιμή 13: η τιμή της τώρα μεταβάλλεται καθώς το x παίρνει τιμές στο διάστημα $-\pi$ έως π. Κατά συνέπεια, το τελικό αποτέλεσμα δε θα είναι το αναμενόμενο.

2.2 Λάμβδα όροι

Ο λ-λογισμός είναι μια τυπική γλώσσα Λ, η σύνταξη της οποίας δίνεται από τον ακόλουθο επαγωγικό ορισμό:

Ορισμός 2.1 *Έστω V ένα αριθμήσιμο σύνολο μεταβλητών. Το σύνολο Λ των όρων του λ-*λογισμού *είναι το μικρότερο σύνολο που ικανοποιεί τις παρακάτω ιδιότητες:*

$$\begin{array}{lcl} x \in V & \Rightarrow & x \in \Lambda \\ M, N \in \Lambda & \Rightarrow & (M\,N) \in \Lambda \\ x \in V, M \in \Lambda & \Rightarrow & (\lambda x.\, M) \in \Lambda \end{array}$$

Τα στοιχεία του συνόλου Λ ονομάζονται επίσης λ-όροι *(λ-terms) ή* λ-εκφράσεις *(λ-expressions).*

Από τον ορισμό 2.1 προκύπτει ότι οι υπάρχουν τριων ειδών λ-όροι:

1. *Μεταβλητές* (variables), δηλαδή στοιχεία του συνόλου V.

2. *Εφαρμογές* (applications), που έχουν τη μορφή $(M\,N)$, όπου M και N είναι λ-όροι.

3. *Αφαιρέσεις* (abstractions), που έχουν τη μορφή $(\lambda x.\, M)$, όπου x μια μεταβλητή και M ένας λ-όρος.

Κατά σύμβαση θα χρησιμοποιούμε μικρά γράμματα του λατινικού αλφαβήτου, (x, y, z, κ.λπ.) για να συμβολίσουμε μεταβλητές και κεφαλαία γράμματα του λατινικού αλφαβήτου (M, N, F, G, P, Q, κ.λπ.) για να συμβολίσουμε λ-όρους.

Χρησιμοποιώντας αφηρημένη σύνταξη σε μορφή BNF και θεωρώντας ότι η συντακτική κλάση των μεταβλητών παριστάνεται με το μη-τερματικό σύμβολο $\langle\text{var}\rangle$, η γλώσσα Λ των λ-όρων περιγράφεται ισοδύναμα ως εξής:

$$\begin{array}{lcl} \langle\text{term}\rangle & ::= & \langle\text{var}\rangle \\ & | & (\ \langle\text{term}\rangle\ \langle\text{term}\rangle\ ) \\ & | & (\ \lambda\ \langle\text{var}\rangle\ .\ \langle\text{term}\rangle\ ) \end{array}$$

Παράδειγμα 2.1 Οι παρακάτω είναι λ-όροι:

$(x\,y)$
$(\lambda x.\,x)$
$(\lambda x.\,(\lambda y.\,(x\,y)))$
$(((\lambda x.\,x)\,y)\,(\lambda x.\,z))$
$((\lambda x.\,(\lambda y.\,z))\,(\lambda x.\,x))$
$(\lambda x.\,((\lambda y.\,y)\,(\lambda z.\,x)))$

□

Παρότι η διαισθητική ερμηνεία των λ-όρων εξηγήθηκε στην ενότητα 2.1, η σύνταξη του λ-λογισμού όπως δίνεται στον ορισμό 2.1 δεν αποδίδει κανένα ιδιαίτερο νόημα στους λ-όρους, που μπορούν να θεωρούνται ως αυθαίρετες συμβολικές εκφράσεις.

Για την απλοποίηση των λ-όρων και την αποφυγή του μεγάλου αριθμού παρενθέσεων που συνεπάγεται η αυστηρή τήρηση του ορισμού 2.1, θα χρησιμοποιούμε στη συνέχεια τις ακόλουθες συμβάσεις:

1. Οι εξωτερικές παρενθέσεις δε γράφονται:

$$\lambda x.\,x \quad \text{είναι συντομογραφία του} \quad (\lambda x.\,x)$$

2. Η εφαρμογή είναι αριστερά προσεταιριστική:

$$F\,M_1\,M_2\,\ldots\,M_n \quad \text{είναι συντομογραφία του} \quad (\ldots((F\,M_1)\,M_2)\,\ldots\,M_n)$$

3. Η αφαίρεση εκτείνεται όσο περισσότερο είναι δυνατό, δηλαδή ως το επόμενο κλείσιμο παρένθεσης ή το τέλος του όρου:

$$\lambda x.\,M_1\,M_2\,\ldots\,M_n \quad \text{είναι συντομογραφία του} \quad \lambda x.\,(M_1\,M_2\,\ldots\,M_n)$$

Παράδειγμα 2.2 Ακολουθώντας τις παραπάνω συμβάσεις, οι λ-όροι του παραδείγματος 2.1 γράφονται σε απλοποιημένη μορφή ως εξής:

$x\,y$
$\lambda x.\,x$
$\lambda x.\,\lambda y.\,x\,y$
$((\lambda x.\,x)\,y)\,(\lambda x.\,z)$
$(\lambda x.\,\lambda y.\,z)\,(\lambda x.\,x)$
$\lambda x.\,(\lambda y.\,y)\,(\lambda z.\,x)$

□

Μια από τις σχέσεις που χρησιμοποιείται για τη σύγκριση μεταξύ λ-όρων είναι η σχέση $\equiv$ που υποδηλώνει ότι δυο όροι είναι πανομοιότυποι. Πρόκειται ουσιαστικά για τη βασική σχέση ισότητας μεταξύ των στοιχείων του συνόλου Λ.

Ορισμός 2.2 *Η σχέση ταυτότητας $\equiv$ στο σύνολο των λ-όρων ορίζεται επαγωγικά ως εξής:*

$$
\begin{array}{lcll}
x & \equiv & y & \textit{αν } x = y \\
(M\,N) & \equiv & (P\,Q) & \textit{αν } M \equiv P \textit{ και } N \equiv Q \\
(\lambda x.\, M) & \equiv & (\lambda y.\, N) & \textit{αν } x = y \textit{ και } M \equiv N
\end{array}
$$

όπου με $x = y$ συμβολίζεται η σχέση ισότητας στο σύνολο V (δηλαδή x και y είναι η ίδια μεταβλητή). Αν $M \equiv N$, οι όροι $M, N \in \Lambda$ ονομάζονται ταυτόσημοι *(identical).*

Η μεταβλητή x στην αφαίρεση $\lambda x.\, M$ ονομάζεται *δεσμεύουσα μεταβλητή* (binding variable). Η *εμβέλεια* (scope) της αφαίρεσης λx είναι ο λ-όρος M, εκτός από τυχόν αφαιρέσεις που αυτός περιέχει και στις οποίες η δεσμεύουσα μεταβλητή είναι πάλι η x. Εμφανίσεις της μεταβλητής x που βρίσκονται στην εμβέλεια κάποιου λx ονομάζονται *δεσμευμένες* (bound). Αντίθετα, εμφανίσεις που δε βρίσκονται στην εμβέλεια κανενός λx ονομάζονται *ελεύθερες* (free). Τα παραπάνω συνοψίζονται στον ακόλουθο ορισμό.

Ορισμός 2.3 *Το σύνολο των* ελεύθερων μεταβλητών *(free variables) ενός λ-όρου $M \in \Lambda$ συμβολίζεται με $\boldsymbol{FV}(M)$ και ορίζεται επαγωγικά ως εξής:*

$$
\begin{array}{lcl}
\boldsymbol{FV}(x) & = & \{\, x \,\} \\
\boldsymbol{FV}(M\,N) & = & \boldsymbol{FV}(M) \cup \boldsymbol{FV}(N) \\
\boldsymbol{FV}(\lambda x.\, M) & = & \boldsymbol{FV}(M) - \{\, x \,\}
\end{array}
$$

Παράδειγμα 2.3 Έστω ο παρακάτω όρος:

$$
M \;\equiv\; (\lambda x.\, \underset{\text{ελεύθερη}}{y}\, \underset{\text{δεσμευμένη}}{x})\, (\lambda y.\, \underset{\text{ελεύθερη}}{x}\, \underset{\text{δεσμευμένη}}{y})
$$

Ακολουθώντας τον ορισμό 2.3 έχουμε:

$$
\begin{array}{lcl}
\boldsymbol{FV}(M) & = & \boldsymbol{FV}((\lambda x.\, y\, x)(\lambda y.\, x\, y)) \\
 & = & \boldsymbol{FV}(\lambda x.\, y\, x) \cup \boldsymbol{FV}(\lambda y.\, x\, y) \\
 & = & (\boldsymbol{FV}(y\, x) - \{\, x \,\}) \cup (\boldsymbol{FV}(x\, y) - \{\, y \,\}) \\
 & = & ((\boldsymbol{FV}(y) \cup \boldsymbol{FV}(x)) - \{\, x \,\}) \cup ((\boldsymbol{FV}(x) \cup \boldsymbol{FV}(y)) - \{\, y \,\}) \\
 & = & ((\{\, y \,\} \cup \{\, x \,\}) - \{\, x \,\}) \cup ((\{\, x \,\} \cup \{\, y \,\}) - \{\, y \,\}) \\
 & = & (\{\, x, y \,\} - \{\, x \,\}) \cup (\{\, x, y \,\} - \{\, y \,\}) \\
 & = & \{\, y \,\} \cup \{\, x \,\} \\
 & = & \{\, x, y \,\}
\end{array}
$$

□

Ορισμός 2.4 *Ένας λ-όρος $M \in \Lambda$ ονομάζεται* κλειστός λ-όρος *(closed λ-term ή combinator), αν $\boldsymbol{FV}(M) = \varnothing$. Το σύνολο των κλειστών λ-όρων συμβολίζεται με Λ^0.*

Παράδειγμα 2.4 Οι ακόλουθοι λ-όροι είναι κλειστοί και στη βιβλιογραφία ονομάζονται συχνά *πρότυποι κλειστοί όροι* (standard combinators).

$$
\begin{array}{lcl}
\mathsf{I} & \equiv & \lambda x.\, x \\
\mathsf{K} & \equiv & \lambda x.\, \lambda y.\, x \\
\mathsf{K}_* & \equiv & \lambda x.\, \lambda y.\, y \\
\mathsf{S} & \equiv & \lambda x.\, \lambda y.\, \lambda z.\, (x\, z)\, (y\, z)
\end{array}
$$

$\square$

2.3 Αντικατάσταση

Η αντικατάσταση είναι η βασική πράξη συμβολικής επεξεργασίας των λ-όρων και επηρεάζει τις ελεύθερες μεταβλητές. Ο συμβολισμός $M[x := N]$ παριστάνει το αποτέλεσμα της αντικατάστασης στον όρο M όλων των ελεύθερων εμφανίσεων της μεταβλητής x με τον όρο N.

Ορισμός 2.5 *Η αντικατάσταση (substitution) της μεταβλητής $x \in V$ στον όρο $M \in \Lambda$ με τον όρο $N \in \Lambda$ συμβολίζεται με $M[x := N]$ και ορίζεται επαγωγικά ως εξής:*

$$
\begin{array}{lcll}
x[x := N] & \equiv & N & \\
y[x := N] & \equiv & y & \textit{αν } y \neq x \\
(P\,Q)[x := N] & \equiv & P[x := N]\,Q[x := N] & \\
(\lambda x.\,P)[x := N] & \equiv & \lambda x.\,P & \\
(\lambda y.\,P)[x := N] & \equiv & \lambda y.\,P[x := N] & \textit{αν } y \neq x, \textit{ και } (y \notin \boldsymbol{FV}(N) \textit{ ή } x \notin \boldsymbol{FV}(P)) \\
(\lambda y.\,P)[x := N] & \equiv & \lambda z.\,P[y := z][x := N] & \textit{αν } y \neq x, \textit{ και } y \in \boldsymbol{FV}(N) \textit{ και } x \in \boldsymbol{FV}(P), \\
 & & & \qquad \textit{όπου } z \notin \boldsymbol{FV}(P) \cup \boldsymbol{FV}(N)
\end{array}
$$

Από τον παραπάνω ορισμό της αντικατάστασης εξασφαλίζεται αφενός ότι αντικαθίστανται μόνο οι ελεύθερες εμφανίσεις μιας μεταβλητής, αφετέρου ότι στο αποτέλεσμα της αντικατάστασης καμία από τις ελεύθερες μεταβλητές του όρου N δε θα έχει γίνει δεσμευμένη. Ειδικότερα για το δεύτερο, από τον ορισμό 2.5 διαπιστώνει κανείς ότι δέσμευση μιας ελεύθερης μεταβλητής του όρου N θα μπορούσε να γίνει μόνο μέσω του τελευταίου κανόνα, στην περίπτωση που $y \in \boldsymbol{FV}(N)$ και $x \in \boldsymbol{FV}(P)$. Ο κανόνας όμως αυτός μεριμνά για τη μετονομασία της δεσμεύουσας μεταβλητής, προκειμένου να αποφευχθεί αυτό το ενδεχόμενο.[2]

Στον τελευταίο κανόνα του ορισμού 2.5 η επιλογή της μεταβλητής $z \in V$ είναι ελεύθερη, υπό την προϋπόθεση να ισχύει $z \notin \boldsymbol{FV}(P) \cup \boldsymbol{FV}(N)$. Για να οριστεί όμως με μονοσήμαντο τρόπο το αποτέλεσμα της αντικατάστασης $M[x := N]$, πρέπει σε αυτό τον κανόνα να καθορίζεται ακριβώς πώς γίνεται η επιλογή αυτής της μεταβλητής. Για το λόγο αυτό, θεωρούμε ότι τα στοιχεία του συνόλου V είναι αριθμημένα και επιλέγουμε ως z το πρώτο στοιχείο που ικανοποιεί τη σχέση $z \notin \boldsymbol{FV}(P) \cup \boldsymbol{FV}(N)$.

2.4 Μετατροπές

Στο λ-λογισμό υπάρχουν τρία είδη *μετατροπής* (conversion), που παραδοσιακά συμβολίζονται με τα τρία γράμματα α, β και η. Κάθε είδος μετατροπής περιγράφεται από έναν κανόνα της μορφής

$$M \;\to_\chi\; N \qquad (\chi)$$

όπου $\chi \in \{\,\alpha, \beta, \eta\,\}$ και $M, N \in \Lambda$. Ο όρος M ονομάζεται *χ-redex*, ή απλά *redex*, και ο όρος N ονομάζεται *contractum*. Για παράδειγμα, στον κανόνα της β-μετατροπής, ο όρος M που μετατρέπεται ονομάζεται β-redex. Οι τρεις κανόνες μετατροπής ορίζουν τους τρεις νόμιμους

[2] Ο ορισμός 2.5 σε αυτή τη μορφή προέρχεται από την εργασία [Curr58] και επαναλαμβάνεται στην [Hind86].

τρόπους "ισοδύναμου" μετασχηματισμού των λ-όρων και, κατ' αυτό τον τρόπο, περιγράφουν τη σημασία αυτών των όρων, την οποία εξηγήσαμε διαισθητικά στην ενότητα 2.1.

Πριν προχωρήσουμε στον ορισμό των τριων κανόνων μετατροπής είναι σκόπιμο να ορίσουμε μερικές χρήσιμες ιδιότητες σχέσεων μεταξύ λ-όρων. Η πρώτη από αυτές είναι η ιδιότητα της συμβατότητας, που δηλώνει μια μορφή κλειστότητας ως προς την κατασκευή των λ-όρων.

Ορισμός 2.6 *Μια σχέση $\sim$ πάνω στο Λ ονομάζεται* συμβατή *(compatible) αν για κάθε $x \in V$ και για κάθε $M, N, P \in \Lambda$ ισχύουν οι παρακάτω ιδιότητες:*

$$\begin{aligned} M \sim N &\Rightarrow (M\,P) \sim (N\,P) \\ M \sim N &\Rightarrow (P\,M) \sim (P\,N) \\ M \sim N &\Rightarrow (\lambda x.\,M) \sim (\lambda x.\,N) \end{aligned}$$

Εύκολα διαπιστώνει κανείς ότι η σχέση ταυτότητας $\equiv$ μεταξύ των λ-όρων είναι συμβατή.

Βάσει της συμβατότητας, μπορούν να οριστούν δυο ακόμα ιδιότητες σχέσεων μεταξύ λ-όρων που θα είναι χρήσιμες στη συνέχεια.

Ορισμός 2.7 *Μια* σχέση αναγωγής *(reduction relation) είναι μια συμβατή, ανακλαστική και μεταβατική σχέση πάνω στο Λ.*

Ορισμός 2.8 *Μια* σχέση συμφωνίας *(congruence relation) είναι μια συμβατή σχέση ισοδυναμίας πάνω στο Λ.*[3]

2.4.1 α-μετατροπή

Σύμφωνα με το συνήθη συμβολισμό του συναρτησιακού λογισμού στα μαθηματικά, τα ονόματα των παραμέτρων μιας συνάρτησης δεν έχουν ιδιαίτερη σημασία: η συνάρτηση $f(x) = 3x + 1$ είναι ταυτόσημη με τη συνάρτηση $f(y) = 3y + 1$. Αν θεωρήσουμε ότι οι λ-όροι συμβολίζουν συναρτήσεις, αντιλαμβάνεται κανείς ότι η επιλογή μιας συγκεκριμένης δεσμεύουσας μεταβλητής x στην αφαίρεση $\lambda x.\,M$ δεν είναι ιδιαίτερα σημαντική. Ο κανόνας της α-μετατροπής επιτρέπει τη μετονομασία της δεσμεύουσας μεταβλητής σε μια αφαίρεση.

Ορισμός 2.9 *Η σχέση $\rightarrow_\alpha$ ορίζεται ως η μικρότερη συμβατή σχέση πάνω στο Λ για την οποία για κάθε $M \in \Lambda$ και για κάθε $x, y \in V$ τέτοια ώστε $y \notin \boldsymbol{FV}(M)$, ισχύει*

$$\lambda x.\,M \quad \rightarrow_\alpha \quad \lambda y.\,M[x := y] \tag{α}$$

Ο περιορισμός $y \notin \boldsymbol{FV}(M)$ εξασφαλίζει ότι η μετονομασία της δεσμεύουσας μεταβλητής δεν προκαλεί τη δέσμευση μεταβλητών του όρου M που ήταν αρχικά ελεύθερες.

Παράδειγμα 2.5 Οι παρακάτω α-μετατροπές είναι σωστές:

$$\begin{aligned} &\lambda x.\,x &&\rightarrow_\alpha \lambda y.\,y \\ &\lambda x.\,z\,x &&\rightarrow_\alpha \lambda y.\,z\,y \\ &\lambda x.\,\lambda y.\,z\,x\,y &&\rightarrow_\alpha \lambda y.\,\lambda w.\,z\,y\,w \end{aligned}$$

[3] Υπενθυμίζεται ότι μια σχέση ονομάζεται *σχέση ισοδυναμίας* (equivalence relation) αν είναι ανακλαστική, μεταβατική και συμμετρική.

Παρατηρούμε ότι στην τρίτη μετατροπή, η αντικατάσταση $(\lambda y.\, z\, x\, y)[x := y]$ προκάλεσε τη μετονομασία της δεσμεύουσας μεταβλητής, προκειμένου να μη δεσμευθεί στο αποτέλεσμα της αντικατάστασης η ελεύθερη εμφάνιση της y.

Αντίθετα η παρακάτω μετατροπή είναι λανθασμένη:

$$\lambda x.\, \lambda z.\, z\, x\, y \;\not\to_{\alpha}\; \lambda y.\, \lambda z.\, z\, y\, y$$

γιατί δεν πληροίται ο περιορισμός $y \notin \boldsymbol{FV}(\lambda z.\, z\, x\, y)$. □

2.4.2 β-μετατροπή

Όπως ήδη αναφέρθηκε στην ενότητα 2.1, ένας όρος της μορφής $(\lambda x.\, M)\, N$ συμβολίζει την εφαρμογή της συνάρτησης $\lambda x.\, M$ στο όρισμα N. Ο υπολογισμός του αποτελέσματος αυτής της εφαρμογής γίνεται αντικαθιστώντας την παράμετρο x με το όρισμα N στον όρο M. Με άλλα λόγια, σε αυτό τον υπολογισμό, ο όρος $(\lambda x.\, M)\, N$ μετατρέπεται στον όρο $M[x := N]$. Ο κανόνας της β-μετατροπής περιγράφει ακριβώς αυτή τη διαδικασία.

Ορισμός 2.10 *Η σχέση $\to_\beta$ ορίζεται ως η μικρότερη συμβατή σχέση πάνω στο Λ για την οποία για κάθε $M, N \in \Lambda$ και για κάθε $x \in V$ ισχύει*

$$(\lambda x.\, M)\, N \quad \to_\beta \quad M[x := N] \tag{β}$$

Παράδειγμα 2.6 Οι παρακάτω β-μετατροπές είναι σωστές:

$$\begin{array}{ll} (\lambda x.\, z\, x)\, w & \to_\beta \; z\, w \\ (\lambda x.\, \lambda y.\, z\, x\, y)\, w & \to_\beta \; \lambda y.\, z\, w\, y \\ (\lambda y.\, z\, y\, (\lambda x.\, x\, y))\, w & \to_\beta \; z\, w\, (\lambda x.\, x\, w) \end{array}$$

ενώ αντίθετα η επόμενη μετατροπή είναι λανθασμένη:

$$(\lambda x.\, \lambda y.\, z\, x\, y)\, (w\, y) \;\not\to_\beta\; \lambda y.\, z\, (w\, y)\, y$$

γιατί για να γίνει η αντικατάσταση $(\lambda y.\, z\, x\, y)[x := w\, y]$ πρέπει να μετονομαστεί η δεσμεύουσα μεταβλητή y, διαφορετικά η εμφάνιση της y στον όρο $w\, y$ θα δεσμευόταν. Το σωστό αποτέλεσμα της παραπάνω β-μετατροπής είναι

$$(\lambda x.\, \lambda y.\, z\, x\, y)\, (w\, y) \;\to_\beta\; \lambda t.\, z\, (w\, y)\, t$$

όπου t μια νέα μεταβλητή. □

2.4.3 η-μετατροπή

Σύμφωνα με το συνήθη ορισμό της ισότητας συναρτήσεων στα μαθηματικά, δυο συναρτήσεις είναι ίσες αν και μόνο αν επιστρέφουν το ίδιο αποτέλεσμα, όταν εφαρμοστούν στο ίδιο όρισμα. Η αρχή αυτή εκφράζεται από τον κανόνα της η-μετατροπής στο λ-λογισμό.

Έστω η συνάρτηση που παριστάνεται από τον λ-όρο $(\lambda x.\, M\, x)$, όπου η μεταβλητή x δεν εμφανίζεται ελεύθερη στον όρο M. Σύμφωνα με τον ορισμό 2.10, το αποτέλεσμα της εφαρμογής αυτής της συνάρτησης σε ένα τυχαίο όρισμα N β-μετατρέπεται σε $M\, N$. Από την άλλη πλευρά, αν ο όρος M θεωρηθεί ως μια συνάρτηση, τότε το αποτέλεσμα της εφαρμογής του στο όρισμα N είναι ίσο με $M\,N$. Επομένως, οι συναρτήσεις $(\lambda x.\, M\,x)$ και M είναι ίσες και είναι εύλογο να θεωρήσει κανείς ότι η δεύτερη προκύπτει από την απλοποίηση της πρώτης.

Ορισμός 2.11 *Η σχέση $\rightarrow_\eta$ ορίζεται ως η μικρότερη συμβατή σχέση πάνω στο Λ για την οποία για κάθε $M \in \Lambda$ και για κάθε $x \in V$ τέτοια ώστε $x \notin \boldsymbol{FV}(M)$, ισχύει*

$$\lambda x.\, M\,x \quad \rightarrow_\eta \quad M \qquad (\eta)$$

Παράδειγμα 2.7 Οι παρακάτω η-μετατροπές είναι σωστές:

$$\begin{array}{lll} \lambda x.\, z\,x & \rightarrow_\eta & z \\ \lambda y.\, z\,x\,y & \rightarrow_\eta & z\,x \end{array}$$

Δεν ισχύει όμως το ίδιο για την επόμενη μετατροπή:

$$\lambda x.\, z\,x\,x \;\not\rightarrow_\eta\; z\,x$$

γιατί δεν πληροίται ο περιορισμός $x \notin \boldsymbol{FV}(z\,x)$. □

2.4.4 Μετατροπή, αναγωγή και ισότητα

Οι τρεις κανόνες μετατροπής (α), (β) και (η) που περιγράφηκαν στις προηγούμενες ενότητες ορίζουν τις τρεις βασικές σχέσεις μετατροπής $\rightarrow_\alpha$, $\rightarrow_\beta$ και $\rightarrow_\eta$ για το λ-λογισμό. Η συνολική σχέση μετατροπής $M \rightarrow N$, που ορίζεται παρακάτω, υποδηλώνει ότι ο όρος M μετατρέπεται *σε ένα βήμα* στον όρο N με κάποιον από αυτούς τους τρεις κανόνες.

Ορισμός 2.12 *Με $\rightarrow$ συμβολίζουμε την ένωση των σχέσεων $\rightarrow_\alpha$, $\rightarrow_\beta$ και $\rightarrow_\eta$. Δηλαδή, η σχέση $\rightarrow$ ορίζεται ως η μικρότερη σχέση για την οποία ισχύουν:*

$$\begin{array}{lll} M \rightarrow_\alpha N & \Rightarrow & M \rightarrow N \\ M \rightarrow_\beta N & \Rightarrow & M \rightarrow N \\ M \rightarrow_\eta N & \Rightarrow & M \rightarrow N \end{array}$$

Συχνά είναι χρήσιμο να αναφέρεται κανείς σε μεγαλύτερες ακολουθίες βημάτων μετατροπής με τη γενική μορφή

$$M \;\equiv\; N_0 \;\rightarrow\; N_1 \;\rightarrow\; N_2 \;\rightarrow\; \ldots \;\rightarrow\; N_{n-1} \;\rightarrow\; N_n \;\equiv\; N$$

Στην περίπτωση αυτή, ο όρος M μετατρέπεται στον όρο N *με μια ακολουθία βημάτων*, πιθανώς κενή. Η σχέση αυτή συμβολίζεται με $M \twoheadrightarrow N$. Αν $n = 1$ τότε η μετατροπή γίνεται σε ένα βήμα $M \rightarrow N$ και άρα η σχέση $\twoheadrightarrow$ εμπεριέχει την $\rightarrow$. Επίσης, αν $n = 0$ τότε η ακολουθία είναι κενή και δε συμβαίνει μετατροπή, δηλαδή $M \equiv N$ και άρα η σχέση $\twoheadrightarrow$ εμπεριέχει την $\equiv$.

Ορισμός 2.13 *Με $\twoheadrightarrow$ συμβολίζουμε το ανακλαστικό και μεταβατικό κλείσιμο της σχέσης $\rightarrow$. Δηλαδή, η σχέση $\twoheadrightarrow$ ορίζεται ως η μικρότερη σχέση για την οποία ισχύουν:*

$$\begin{array}{lll} M \rightarrow N & \Rightarrow & M \twoheadrightarrow N \\ & & M \twoheadrightarrow M \\ M \twoheadrightarrow N, N \twoheadrightarrow P & \Rightarrow & M \twoheadrightarrow P \end{array}$$

Εύκολα διαπιστώνει κανείς ότι η σχέση $\twoheadrightarrow$ είναι σχέση αναγωγής (βλ. ορισμό 2.7). Όταν ισχύει $M \twoheadrightarrow N$, λέμε ότι ο όρος M *ανάγεται* στον όρο N.

Όταν $M \twoheadrightarrow N$, όλα τα βήματα στην ακολουθία μετατροπής έχουν την ίδια κατεύθυνση, από αριστερά προς τα δεξιά. Αν σε αυτή την ακολουθία επιτρέψουμε και την ύπαρξη "αντίστροφων" βημάτων, όπως π.χ. στην περίπτωση

$$M \equiv N_0 \rightarrow N_1 \leftarrow N_2 \rightarrow N_3 \leftarrow N_4 \leftarrow N_5 \rightarrow N_6 \equiv N$$

τότε οδηγούμαστε σε μια γενικότερη σχέση την οποία συμβολίζουμε με $=$. Η σχέση αυτή, όπως και η $\twoheadrightarrow$, εμπεριέχει τις $\rightarrow$ και $\equiv$, αλλά εμπεριέχει και την ίδια την $\twoheadrightarrow$.

Ορισμός 2.14 *Με $=$ συμβολίζουμε τη σχέση ισοδυναμίας που προκύπτει από την $\twoheadrightarrow$. Δηλαδή, η σχέση $=$ ορίζεται ως η μικρότερη σχέση για την οποία ισχύουν:*

$$\begin{array}{lcl} M \twoheadrightarrow N & \Rightarrow & M = N \\ M = N & \Rightarrow & N = M \\ M = N, N = P & \Rightarrow & M = P \end{array}$$

Εύκολα διαπιστώνει κανείς ότι η σχέση $=$ είναι σχέση συμφωνίας (βλ. ορισμό 2.8). Όταν ισχύει $M = N$, λέμε ότι ο όρος M *είναι ίσος* με τον όρο N.

Στο σημείο αυτό πρέπει να τονισθεί ότι με τη σχέση ισότητας $M = N$ δεν εννοείται ότι οι δύο όροι ταυτίζονται, κάτι που σύμφωνα με τον ορισμό 2.2 συμβολίζεται ως $M \equiv N$. Η διάκριση αυτών των δυο σχέσεων και η επιλογή του συμβόλου $=$ για τη σχέση συμφωνίας του ορισμού 2.14 μπορεί να μπερδέψει προς στιγμήν τον αναγνώστη. Η διαισθητική αιτιολόγηση αυτής της επιλογής θα πρέπει να περιμένει μέχρι την πρόταση 2.2 στην ενότητα 2.6.

Τόσο η σχέση αναγωγής $\twoheadrightarrow$ όσο και η σχέση ισότητας $=$ προέκυψαν ως επεκτάσεις της συνολικής σχέσης μετατροπής $\rightarrow$. Παρόμοιες σχέσεις μπορούν να ορισθούν για τις επιμέρους σχέσεις μετατροπής. Για παράδειγμα, η σχέση $\twoheadrightarrow_\beta$ ορίζεται ως το ανακλαστικό και μεταβατικό κλείσιμο της σχέσης $\rightarrow_\beta$ και η σχέση $=_\alpha$ ορίζεται ως η σχέση ισοδυναμίας που προκύπτει από την $\twoheadrightarrow_\alpha$.

2.5 Κανονικές μορφές

Θεωρώντας τον λ-λογισμό ως ένα υπολογιστικό μοντέλο επεξεργασίας συναρτήσεων, οι διάφορες σχέσεις μετατροπής παριστάνουν την πραγματοποίηση βημάτων επεξεργασίας πάνω στους λ-όρους, που γενικά αποσκοπούν στον υπολογισμό κάποιου αποτελέσματος. Πιο συγκεκριμένα:

- Η β-μετατροπή είναι η κατ' εξοχήν υπολογιστική πράξη στο λ-λογισμό, αφού διαισθητικά παριστάνει την εφαρμογή μιας συνάρτησης σε κάποιο όρισμα.

- Η α-μετατροπή δεν προάγει τη διαδικασία του υπολογισμού. Εφόσον τα ονόματα των δεσμευμένων μεταβλητών δεν έχουν ουσιαστική σημασία, η μετονομασία των μεταβλητών που επιτελείται μέσω της α-μετατροπής δεν αποτελεί ουσιαστικό υπολογιστικό βήμα.[4]

[4] Για το λόγο αυτό, στη βιβλιογραφία συχνά η α-μετατροπή παραλείπεται εντελώς και δυο όροι $M, N \in \Lambda$ για τους οποίους ισχύει $M =_\alpha N$ θεωρούνται ταυτόσημοι.

- Η η-μετατροπή, παρότι δε συμβάλλει άμεσα στην υπολογιστική διαδικασία, υλοποιεί ένα μηχανισμό για την απλοποίηση λ-όρων που παριστάνουν συναρτήσεις.[5]

Με όλα αυτά κατά νου, εύκολα διαπιστώνει κανείς ότι η σχέση μετατροπής $M \rightarrow N$ παριστάνει ένα βήμα στη διαδικασία υπολογισμού, ενώ η σχέση $M \twoheadrightarrow N$ παριστάνει μια (πιθανώς κενή) ακολουθία από τέτοια βήματα. Όταν $M \twoheadrightarrow N$, μπορεί να θεωρηθεί ότι ο όρος N έχει προκύψει κατά την *αποτίμηση* (evaluation) του όρου M. Με αυτή την έννοια, αποτίμηση ενός όρου ονομάζεται η διαδοχική εφαρμογή κανόνων μετατροπής σε αυτόν.

Όταν σε έναν όρο M δεν μπορούμε να εφαρμόσουμε κανένα αμιγώς υπολογιστικό βήμα, δηλαδή καμία β- ή η-μετατροπή, τότε μπορεί να θεωρηθεί ότι η αποτίμησή του έχει ολοκληρωθεί και ότι ο όρος αυτός αποτελεί ένα τελικό αποτέλεσμα. Τέτοιοι πλήρως αποτιμημένοι όροι ονομάζονται *κανονικές μορφές*.

Ορισμός 2.15 *Ένας όρος $M \in \Lambda$* είναι σε κανονική μορφή *(normal form) αν δεν περιέχει κανένα β-redex ή η-redex.*

Η κανονική μορφή του ορισμού 2.15 ονομάζεται συχνά στη βιβλιογραφία και ως $\beta\eta$-κανονική μορφή ($\beta\eta$-normal form). Περιορίζοντας περισσότερο το είδος της μετατροπής, ένας όρος M είναι σε *β-κανονική μορφή* (β-normal form) αν δεν περιέχει κανένα β-redex και ομοίως σε *η-κανονική μορφή* (η-normal form) αν δεν περιέχει κανένα η-redex.

Παράδειγμα 2.8 Οι όροι $\lambda x.\, x$ και $\lambda f.\, f\,(\lambda x.\, x\, f)$ είναι σε κανονική μορφή. Αντίθετα, ο όρος $\lambda z.\,(\lambda f.\, \lambda x.\, f\, z\, x)\,(\lambda y.\, y)$ δεν είναι σε κανονική μορφή γιατί περιέχει το β-redex:

$$(\lambda f.\, \lambda x.\, f\, z\, x)\,(\lambda y.\, y) \;\rightarrow_\beta\; \lambda x.\,(\lambda y.\, y)\, z\, x$$

□

Στον ορισμό 2.15 παρατηρεί κανείς ότι στις κανονικές μορφές δε λαμβάνεται υπόψη η α-μετατροπή. Αυτό είναι σύμφωνο με τη θεώρηση του λ-λογισμού ως υπολογιστικού μοντέλου και, αν δε συνέβαινε, απλοί όροι όπως ο $\lambda x.\, x$ δε θα ήταν σε κανονική μορφή λόγω της δυνατής μετονομασίας κάποιας δεσμεύουσας μεταβλητής. Κατά συνέπεια η σχέση συμφωνίας $=_\alpha$, βάσει της οποίας όλοι οι όροι που προκύπτουν με α-μετατροπές είναι ισοδύναμοι, αποτελεί θεμελιώδη σχέση ισοδυναμίας μεταξύ κανονικών μορφών. Το γεγονός αυτό αποδίδεται και από την επόμενη πρόταση.

Πρόταση 2.1 *Αν ο όρος $M \in \Lambda$ είναι σε κανονική μορφή και ισχύει $M \twoheadrightarrow N$ για κάποιον όρο $N \in \Lambda$, τότε $M =_\alpha N$.*

Υπάρχουν λ-όροι η αποτίμηση των οποίων οδηγεί, μετά από διαδοχικές μετατροπές, σε κάποια κανονική μορφή. Υπάρχουν όμως και όροι για τους οποίους αυτό δεν ισχύει, δηλαδή η αποτίμησή τους μπορεί να συνεχίζεται επ' άπειρον χωρίς να καταλήγει σε κανονική μορφή.

[5] Στη βιβλιογραφία συχνά η η-μετατροπή αντιμετωπίζεται ως δευτερεύουσας σημασίας και πολλές φορές αγνοείται εντελώς.

Ορισμός 2.16 *Ένας όρος $M \in \Lambda$ έχει κανονική μορφή αν για κάποιο όρο $N \in \Lambda$ ισχύει $M \twoheadrightarrow N$ και ο N είναι σε κανονική μορφή. Στην περίπτωση αυτή, ο όρος M ονομάζεται κανονικοποιήσιμος (normalizing).*

Παράδειγμα 2.9 Ο όρος $\lambda z.\,(\lambda f.\,\lambda x.\,f\,z\,x)\,(\lambda y.\,y)$ στο παράδειγμα 2.8 είδαμε ότι δεν είναι σε κανονική μορφή. Όμως, εφαρμόζοντας διαδοχικά βήματα μετατροπής προκύπτει:

$$\begin{aligned}
\lambda z.\,(\lambda f.\,\lambda x.\,f\,z\,x)\,(\lambda y.\,y) &\to_\beta \lambda z.\,\lambda x.\,(\lambda y.\,y)\,z\,x \\
&\to_\beta \lambda z.\,\lambda x.\,z\,x \\
&\to_\eta \lambda z.\,z
\end{aligned}$$

και άρα $\lambda z.\,(\lambda f.\,\lambda x.\,f\,z\,x)\,(\lambda y.\,y) \twoheadrightarrow \lambda z.\,z$. Καθώς ο όρος $\lambda z.\,z$ είναι σε κανονική μορφή, ο αρχικός όρος έχει αυτή την κανονική μορφή. □

Παράδειγμα 2.10 Έστω ο όρος $\Omega \equiv (\lambda x.\,x\,x)\,(\lambda x.\,x\,x)$, που δεν είναι σε κανονική μορφή. Ο όρος αυτός περιέχει μόνο ένα β-redex και κατά συνέπεια η μόνη μετατροπή που μπορεί να εφαρμοσθεί είναι η:

$$\Omega \equiv (\lambda x.\,x\,x)\,(\lambda x.\,x\,x) \to_\beta (\lambda x.\,x\,x)\,(\lambda x.\,x\,x) \equiv \Omega$$

Σε ένα βήμα μετατροπής, ο όρος Ω μετατρέπεται πάλι στον εαυτό του. Επομένως η διαδικασία αποτίμησης δεν μπορεί να τερματιστεί και ο όρος Ω δεν έχει κανονική μορφή. □

Στην περίπτωση του Ω, η διαδικασία της αποτίμησης δεν παρουσιάζει πρόοδο. Υπάρχουν όμως όροι των οποίων η αποτίμηση όχι μόνο δεν παρουσιάζει πρόοδο, αλλά οδηγεί στην ανεξέλεγκτη αύξησή τους.

Παράδειγμα 2.11 Έστω ο όρος $M \equiv (\lambda x.\,x\,x\,y)\,(\lambda x.\,x\,x\,y)$, που περιέχει μόνο ένα β-redex. Η μόνη ακολουθία μετατροπών που μπορεί να εφαρμοσθεί είναι η ακόλουθη:

$$\begin{aligned}
M \equiv\ & (\lambda x.\,x\,x\,y)\,(\lambda x.\,x\,x\,y) \\
\to_\beta\ & (\lambda x.\,x\,x\,y)\,(\lambda x.\,x\,x\,y)\,y && \equiv M\,y \\
\to_\beta\ & (\lambda x.\,x\,x\,y)\,(\lambda x.\,x\,x\,y)\,y\,y && \equiv M\,y\,y \\
\to_\beta\ & (\lambda x.\,x\,x\,y)\,(\lambda x.\,x\,x\,y)\,y\,y\,y && \equiv M\,y\,y\,y \\
\to_\beta\ & \ldots
\end{aligned}$$

Κάθε αποτέλεσμα σε αυτή την ακολουθία μετατροπών δεν είναι σε κανονική μορφή και έχει ένα μόνο β-redex. □

Αν ένας όρος M περιέχει περισσότερα του ενός redex, η αποτίμησή του μπορεί να ακολουθήσει περισσότερους διαφορετικούς δρόμους, όπως φαίνεται στο ακόλουθο παράδειγμα.

Παράδειγμα 2.12 Έστω ο όρος $M = (\lambda x.\,(\lambda y.\,x\,y)\,z)\,w$ που περιέχει δύο β-redex για τις δύο αφαιρέσεις με δεσμεύουσες μεταβλητές τις x και y. Μετατρέποντας πρώτα το πρώτο από αυτά προκύπτει η εξής αποτίμηση:

$$M \to_\beta (\lambda y.\,w\,y)\,z \to_\beta w\,z$$

ενώ μετατρέποντάς τα με την αντίστροφη σειρά προκύπτει η εξής αποτίμηση:

$$M \ \rightarrow_\beta \ (\lambda x.\, x\, z)\, w \ \rightarrow_\beta \ w\, z$$

□

Το γεγονός ότι οι δύο δρόμοι αποτίμησης στο προηγούμενο παράδειγμα καταλήγουν στην ίδια κανονική μορφή δεν είναι τυχαίο, όπως θα δούμε στην επόμενη ενότητα. Υπάρχουν όμως όροι για τους οποίους η εύρεση ή όχι κανονικής μορφής εξαρτάται από τη σειρά με την οποία θα γίνουν οι μετατροπές.

Παράδειγμα 2.13 Έστω ο όρος $M = (\lambda z.\, y)\, ((\lambda x.\, x\, x)\, (\lambda x.\, x\, x))$, που περιέχει τον όρο Ω του παραδείγματος 2.10. Ο όρος M περιέχει δύο β-redex για τις δύο αφαιρέσεις με δεσμεύουσες μεταβλητές την z και την πρώτη από τις δύο x. Μετατρέποντας το πρώτο, οδηγούμαστε αμέσως σε κανονική μορφή:

$$M \ \rightarrow_\beta \ y$$

Αντίθετα, μετατρέποντας το δεύτερο οδηγούμαστε και πάλι στον ίδιο όρο, ακριβώς όπως συμβαίνει με τη μετατροπή του όρου Ω:

$$M \ \rightarrow_\beta \ (\lambda z.\, y)\, ((\lambda x.\, x\, x)\, (\lambda x.\, x\, x)) \ \equiv \ M$$

Διαπιστώνει λοιπόν κανείς ότι, αν και ο όρος M διαθέτει κανονική μορφή, αν επιλέγουμε κάθε φορά να μετατρέπουμε το δεύτερο β-redex τότε η διαδικασία της αποτίμησης δε θα τερματιστεί ποτέ και άρα η κανονική μορφή δε θα βρεθεί. □

Ορισμός 2.17 *Ένας όρος M ονομάζεται* ισχυρά κανονικοποιήσιμος *(strongly normalizing) αν όλες οι ακολουθίες μετατροπής που ξεκινούν με τον M καταλήγουν σε κανονική μορφή.*

Όπως είδαμε στο παράδειγμα 2.13, υπάρχουν κανονικοποιήσιμοι όροι που δεν είναι ισχυρά κανονικοποιήσιμοι.

Η ύπαρξη περισσότερων της μιας δυνατών ακολουθιών μετατροπής που να ξεκινούν από ένα δεδομένο λ-όρο καθιστά τη διαδικασία της αποτίμησης στη γενική περίπτωση μη ντετερμινιστική. Μια ντετερμινιστική μέθοδος η οποία αποφασίζει σε κάθε βήμα της αποτίμησης ποιο redex θα μετατραπεί ονομάζεται *στρατηγική αναγωγής* (reduction strategy). Μια από τις απλούστερες και σημαντικότερες, όπως θα δούμε στη συνέχεια, στρατηγικές αναγωγής είναι η ακόλουθη.

Ορισμός 2.18 *Η στρατηγική κατά την οποία επιλέγεται για μετατροπή κάθε φορά το αριστερότερο redex ενός όρου, δηλαδή αυτό του οποίου το σύμβολο λ βρίσκεται όσο το δυνατόν πιο αριστερά, ονομάζεται* στρατηγική της αναγωγής κανονικής σειράς *(normal order reduction strategy).*

2.6 Ιδιότητες του λογισμού

Όπως αναφέραμε ήδη πολλές φορές, ο λ-λογισμός σχεδιάστηκε με σκοπό να αποτελέσει μια γενική θεωρία συναρτήσεων. Σε μια τέτοια θεωρία, θα περίμενε κανείς ότι η αποτίμηση ενός όρου M

μπορεί να δώσει το πολύ ένα τελικό αποτέλεσμα, δηλαδή το πολύ μια κανονική μορφή. Επομένως, λαμβάνοντας υπόψη τη σχέση ισοδυναμίας μεταξύ κανονικών μορφών, θα περίμενε κανείς ότι αν η αποτίμηση του όρου M μπορεί να οδηγήσει σε δύο κανονικές μορφές N_1 και N_2, τότε αυτές θα πρέπει να είναι ισοδύναμες, δηλαδή να ισχύει $N_1 =_\alpha N_2$. Αυτή η ιδέα της μοναδικότητας των κανονικών μορφών εκφράζεται ακριβώς από την πρόταση 2.3. Η απόδειξή της στηρίζεται στο ακόλουθο γενικότερο θεώρημα, το οποίο βεβαιώνει ότι αν η διαδικασία της αποτίμησης ενός όρου μπορεί να ακολουθήσει δυο διαφορετικούς δρόμους, τότε αυτοί υποχρεωτικά κάπου συγκλίνουν.

Θεώρημα 2.1 (Church-Rosser) *Έστω όροι $M, N_1, N_2 \in \Lambda$ τέτοιοι ώστε $M \twoheadrightarrow N_1$ και $M \twoheadrightarrow N_2$. Τότε υπάρχει όρος $N \in \Lambda$ τέτοιος ώστε $N_1 \twoheadrightarrow N$ και $N_2 \twoheadrightarrow N$. Σε μορφή διαγράμματος:*

$$
\begin{tikzcd}
 & M \arrow[dl] \arrow[dr] & \\
N_1 \arrow[dr, dotted] & & N_2 \arrow[dl, dotted] \\
 & N &
\end{tikzcd}
$$

Η ιδιότητα που περιγράφεται στο θεώρημα 2.1 ονομάζεται *ιδιότητα Church-Rosser* (Church-Rosser property), από τα ονόματα των ερευνητών που την απέδειξαν, ή *ιδιότητα της συμβολής* (confluence property).[6] Η απόδειξη του θεωρήματος είναι αρκετά δύσκολη και ξεφεύγει από το σκοπό αυτού του βιβλίου.

Από το θεώρημα 2.1 προκύπτουν εύκολα οι ακόλουθες προτάσεις ως συμπεράσματα. Σύμφωνα με την πρώτη, η απόδειξη της οποίας παραλείπεται, δύο ίσοι όροι μπορούν να αναχθούν στον ίδιο όρο, ο οποίος όμως δεν είναι απαραίτητα κανονική μορφή. Σύμφωνα με τη δεύτερη, η κανονική μορφή ενός όρου (αν υπάρχει) είναι μοναδική.

Πρόταση 2.2 *Αν $M_1 = M_2$, τότε υπάρχει ένας όρος N τέτοιος ώστε $M_1 \twoheadrightarrow N$ και $M_2 \twoheadrightarrow N$.*

Πρόταση 2.3 *Κάθε όρος M έχει το πολύ μία κανονική μορφή, δεδομένης της ισοδυναμίας που ορίζεται από τη σχέση συμφωνίας $=_\alpha$.*

Απόδειξη Έστω ότι ο όρος M έχει δύο κανονικές μορφές N_1 και N_2. Τότε ισχύει $M \twoheadrightarrow N_1$ και $M \twoheadrightarrow N_2$ και άρα $N_1 = N_2$. Από την πρόταση 2.2 προκύπτει ότι υπάρχει όρος P τέτοιος ώστε $N_1 \twoheadrightarrow P$ και $N_2 \twoheadrightarrow P$. Όμως, από την πρόταση 2.1 συνεπάγεται ότι $N_1 =_\alpha P$ και $N_2 =_\alpha P$ και άρα $N_1 =_\alpha N_2$. $\square$

[6] Τα δύο αυτά ονόματα μπορούν να χρησιμοποιηθούν για οποιαδήποτε σχέση μεταξύ λ-όρων η οποία αποδίδεται από ένα παρόμοιο διάγραμμα σε σχήμα ρόμβου. Το θεώρημα 2.1 αναφέρεται στη σχέση $\twoheadrightarrow$, μπορεί όμως να αποδειχθεί ότι την ιδιότητα της συμβολής έχει επίσης η σχέση $=$.

Όπως είδαμε στο παράδειγμα 2.13 της ενότητας 2.5, η αποτίμηση ενός κανονικοποιήσιμου λ-όρου δεν οδηγεί απαραίτητα στην κανονική του μορφή. Από την πρόταση 2.3 εξασφαλίζεται ότι αν για έναν κανονικοποιήσιμο όρο δύο στρατηγικές αποτίμησης οδηγούν σε δύο κανονικές μορφές, τότε οι τελευταίες θα είναι ισοδύναμες. Δεν εξασφαλίζεται όμως ότι κάθε στρατηγική αποτίμησης θα τερματίζει. Το επόμενο θεώρημα αναδεικνύει μια στρατηγική αποτίμησης που εγγυάται την εύρεση της κανονικής μορφής κάθε κανονικοποιήσιμου όρου.

Θεώρημα 2.2 (Κανονικοποίηση – Normalization) *Αν ο όρος M έχει κανονική μορφή, τότε η στρατηγική της αναγωγής κανονικής σειράς οδηγεί σε αυτήν.*

Το παρακάτω θεώρημα καθιστά δυνατή την αναπαράσταση στο λ-λογισμό αναδρομικών συναρτήσεων. Με αυτό το θέμα δε θα ασχοληθούμε εκτενέστερα στις σημειώσεις αυτές.

Θεώρημα 2.3 (Σταθερό σημείο – Fixed point)

(i) *Για κάθε όρο $F \in \Lambda$ υπάρχει όρος $X \in \Lambda$ τέτοιος ώστε να ισχύει $F\,X = X$.*[7]

(ii) *Υπάρχει ένας* τελεστής σταθερού σημείου, *δηλαδή ένας όρος $\mathsf{Y} \in \Lambda$ τέτοιος ώστε για κάθε $F \in \Lambda$ να ισχύει $F\,(\mathsf{Y}\,F) = \mathsf{Y}\,F$.*

Απόδειξη

(i) Έστω $W \;\equiv\; \lambda x.\,F\,(x\,x)$ και $X \;\equiv\; W\,W$. Τότε ισχύει:

$$X \;\equiv\; W\,W \;\equiv\; (\lambda x.\,F\,(x\,x))\,W \;\to_\beta\; F\,(W\,W) \;\equiv\; F\,X$$

(ii) Έστω ο κλειστός όρος $\mathsf{Y} \in \Lambda$ που ορίζεται ως

$$\mathsf{Y} \;\equiv\; \lambda f.\,(\lambda x.\,f\,(x\,x))\,(\lambda x.\,f\,(x\,x))$$

και έστω όρος $F \in \Lambda$. Χρησιμοποιώντας την απόδειξη του (i) έχουμε

$$\mathsf{Y}\,F \;\equiv\; (\lambda f.\,(\lambda x.\,f\,(x\,x))\,(\lambda x.\,f\,(x\,x)))\,F \;\to_\beta\; (\lambda x.\,F\,(x\,x))\,(\lambda x.\,F\,(x\,x)) \;\equiv\; X$$

Καθώς $X = F\,X$, από τη συμβατότητα της σχέσης $=$ εύκολα προκύπτει $\mathsf{Y}\,F = F\,(\mathsf{Y}\,F)$.

□

2.7 Η εκφραστική δύναμη του λάμβδα λογισμού

Ο λ-λογισμός εκ πρώτης όψεως μοιάζει με μια πολύ πρωτόγονη γλώσσα, η οποία δεν μπορεί να εκφράσει πολλές έννοιες. Η πραγματικότητα όμως είναι εντελώς διαφορετική. Ο λ-λογισμός μπορεί να περιγράψει τις περισσότερες έννοιες που χρησιμοποιούνται στις σύγχρονες γλώσσες προγραμματισμού. Στη συνέχεια θα εξετάσουμε πώς γνωστές έννοιες από τα μαθηματικά, τη λογική και τον προγραμματισμό μπορούν να αναπαρασταθούν με τη μορφή λ-όρων.

[7] Στη μαθηματική ορολογία, ο όρος X ονομάζεται *σταθερό σημείο* της συνάρτησης F.

2.7.1 Λογικές τιμές

Μπορεί κανείς σχετικά εύκολα και με πολλούς διαφορετικούς τρόπους να κωδικοποιήσει στο λ-λογισμό τις λογικές τιμές ***true*** και ***false***. Η συνηθέστερη κωδικοποίηση είναι η ακόλουθη:

Ορισμός 2.19
$$\begin{array}{lcl} \textbf{\textit{true}} & \equiv & \lambda x.\, \lambda y.\, x \\ \textbf{\textit{false}} & \equiv & \lambda x.\, \lambda y.\, y \end{array}$$

Εύκολα μπορεί κανείς να κωδικοποιήσει τις λογικές πράξεις πάνω σε λ-όρους. Για παράδειγμα, η λογική άρνηση ***not*** μπορεί να κωδικοποιηθεί με τον ακόλουθο τρόπο.

Ορισμός 2.20
$$\textbf{\textit{not}} \;\equiv\; \lambda z.\, z\, \textbf{\textit{false}}\; \textbf{\textit{true}}$$

(όπου τα ***true*** και ***false*** είναι οι όροι που ορίστηκαν παραπάνω). Εύκολα διαπιστώνει κανείς ότι το ***not*** διαθέτει τις απαιτούμενες ιδιότητες.

Θεώρημα 2.4

(i) $\textbf{\textit{not}}\ \textbf{\textit{true}} = \textbf{\textit{false}}$

(ii) $\textbf{\textit{not}}\ \textbf{\textit{false}} = \textbf{\textit{true}}$

Απόδειξη Θα δείξουμε μόνο το πρώτο (η απόδειξη του δεύτερου γίνεται παρόμοια).

$$\begin{array}{rcl} \textbf{\textit{not}}\ \textbf{\textit{true}} & \equiv & (\lambda z.\, z\, \textbf{\textit{false}}\ \textbf{\textit{true}})\ \textbf{\textit{true}} \\ & \to_\beta & \textbf{\textit{true}}\ \textbf{\textit{false}}\ \textbf{\textit{true}} \\ & \equiv & (\lambda x.\, \lambda y.\, x)\ \textbf{\textit{false}}\ \textbf{\textit{true}} \\ & \twoheadrightarrow_\beta & \textbf{\textit{false}} \end{array}$$

□

Μπορούμε επίσης να ορίσουμε το λ-όρο ***cond*** που κωδικοποιεί τη δομή ***if-then-else***. Θα χρησιμοποιούμε κατά παράβαση αυτή τη δομή για να διευκολύνουμε τη γραφή των όρων που κωδικοποιούν συνθήκες.

Ορισμός 2.21
$$\begin{array}{lcl} \textbf{\textit{cond}} & \equiv & \lambda z.\, \lambda x.\, \lambda y.\, z\, x\, y \\ \textbf{\textit{if}}\ B\ \textbf{\textit{then}}\ N\ \textbf{\textit{else}}\ M & \equiv & \textbf{\textit{cond}}\, B\, N\, M \end{array}$$

Αποδεικνύεται εύκολα ότι η παραπάνω δομή πληροί τις απαιτούμενες ιδιότητες.

Θεώρημα 2.5 *Για κάθε $N, M \in \Lambda$ ισχύουν τα παρακάτω:*

(i) $\textbf{\textit{if}}\ \textbf{\textit{true}}\ \textbf{\textit{then}}\ N\ \textbf{\textit{else}}\ M = N$

(ii) $\textbf{\textit{if}}\ \textbf{\textit{false}}\ \textbf{\textit{then}}\ N\ \textbf{\textit{else}}\ M = M$

Απόδειξη Θα δείξουμε και πάλι μόνο το πρώτο (η απόδειξη του δεύτερου γίνεται παρόμοια).

$$
\begin{aligned}
\textbf{\textit{if true then}}\ N\ \textbf{\textit{else}}\ M &\equiv \textbf{\textit{cond true}}\, N\, M \\
&\equiv (\lambda z.\, \lambda x.\, \lambda y.\, z\, x\, y)\, \textbf{\textit{true}}\, N\, M \\
&\twoheadrightarrow_\beta \textbf{\textit{true}}\, N\, M \\
&\equiv (\lambda x.\, \lambda y.\, x)\, N\, M \\
&\twoheadrightarrow_\beta N
\end{aligned}
$$

□

2.7.2 Διατεταγμένα ζεύγη

Στο λ-λογισμό μπορεί επίσης κανείς να κωδικοποιήσει διατεταγμένα ζεύγη όρων (που με τη σειρά τους μπορούν να κωδικοποιούν άλλα μαθηματικά αντικείμενα). Μία τέτοια δυνατή κωδικοποίηση είναι μέσω του όρου ***pair*** που ορίζεται παρακάτω. Κατά παράβαση, θα χρησιμοποιούμε το συμβολισμό $\langle N, M\rangle$ για να διευκολύνουμε τη γραφή όρων που κωδικοποιούν διατεταγμένα ζεύγη.

Ορισμός 2.22

$$
\begin{aligned}
\textbf{\textit{pair}} &\equiv \lambda x.\, \lambda y.\, \lambda z.\, z\, x\, y \\
\langle N, M\rangle &\equiv \textbf{\textit{pair}}\, N\, M
\end{aligned}
$$

Οι πράξεις ***fst*** και ***snd*** που επιστρέφουν το πρώτο και το δεύτερο στοιχείο ενός διατεταγμένου ζεύγους μπορούν να κωδικοποιηθούν ως εξής:

Ορισμός 2.23

$$
\begin{aligned}
\textbf{\textit{fst}} &\equiv \lambda z.\, z\, \textbf{\textit{true}} \\
\textbf{\textit{snd}} &\equiv \lambda z.\, z\, \textbf{\textit{false}}
\end{aligned}
$$

Εύκολα μπορεί κανείς να δείξει ότι οι πράξεις αυτές πληρούν τις απαιτούμενες ιδιότητες και άρα η δομή $\langle N, M\rangle$ μπορεί να χρησιμεύσει ως διατεταγμένο ζεύγος.

Θεώρημα 2.6 *Για κάθε $N, M \in \Lambda$ ισχύουν τα εξής:*

(i) $\textbf{\textit{fst}}\, \langle N, M\rangle = N$

(ii) $\textbf{\textit{snd}}\, \langle N, M\rangle = M$

Απόδειξη Θα δείξουμε μόνο το πρώτο (η απόδειξη του δεύτερου γίνεται παρόμοια).

$$
\begin{aligned}
\textbf{\textit{fst}}\, \langle N, M\rangle &\equiv (\lambda z.\, z\, \textbf{\textit{true}})\, \langle N, M\rangle \\
&\rightarrow_\beta \langle N, M\rangle\, \textbf{\textit{true}} \\
&\equiv \textbf{\textit{pair}}\, N\, M\, \textbf{\textit{true}} \\
&\equiv (\lambda x.\, \lambda y.\, \lambda z.\, z\, x\, y)\, N\, M\, \textbf{\textit{true}} \\
&\twoheadrightarrow_\beta \textbf{\textit{true}}\, N\, M \\
&\equiv (\lambda x.\, \lambda y.\, x)\, N\, M \\
&\twoheadrightarrow_\beta N
\end{aligned}
$$

□

2.7.3 Φυσικοί αριθμοί

Έχουν προταθεί πολλές διαφορετικές κωδικοποιήσεις των φυσικών αριθμών στο λ-λογισμό. Η πρώτη και πιο γνωστή από αυτές είναι τα αριθμοειδή του Church, που περιγράφονται παρακάτω.

Ορισμός 2.24 *Έστω φυσικός αριθμός $n \in \mathbb{N}$ και όροι $F, A \in \Lambda$. Ο όρος $F^n(A) \in \Lambda$ ορίζεται επαγωγικά ως*

$$\begin{aligned} F^0(A) &\equiv A \\ F^{n+1}(A) &\equiv F\,(F^n(A)) \end{aligned}$$

Ο ορισμός του $F^n(A)$ θα μπορούσε ισοδύναμα να δοθεί με τη μορφή $F^{n+1}(A) \equiv F^n(F\,A)$. Με επαγωγή μπορεί κανείς να αποδείξει εύκολα ότι $F^n(F\,A) \equiv F\,(F^n(A))$ και ότι $F^n(F^m(A)) \equiv F^{n+m}(A)$.

Ορισμός 2.25 (Αριθμοειδή του Church – Church numerals) *Για κάθε φυσικό αριθμό $n \in \mathbb{N}$ ορίζεται ένας όρος $\boldsymbol{c}_n \in \Lambda$ ως*

$$\boldsymbol{c}_n \equiv \lambda f.\, \lambda x.\, f^n(x)$$

Το αριθμοειδές που αντιστοιχεί στον αριθμό 0 είναι το $\boldsymbol{c}_0 \equiv \lambda f.\, \lambda x.\, x$, στον αριθμό 1 αντιστοιχεί το $\boldsymbol{c}_1 \equiv \lambda f.\, \lambda x.\, f\,x$, στον αριθμό 2 το $\boldsymbol{c}_2 \equiv \lambda f.\, \lambda x.\, f\,(f\,x)$, κ.ο.κ. Όπως εύκολα διαπιστώνεται από τον ορισμό τους, όλα τα αριθμοειδή είναι σε β-κανονική μορφή (μάλιστα όλα εκτός του $\boldsymbol{c}_1$ είναι και σε η-κανονική μορφή). Προφανώς, ούτε όλοι οι λ-όροι είναι αριθμοειδή, ούτε και μπορούν όλοι να αναχθούν σε αριθμοειδή.

Μπορεί κανείς τώρα να ορίσει την κωδικοποίηση πάνω στα αριθμοειδή μερικών βασικών πράξεων των φυσικών αριθμών: της συνάρτησης που επιστρέφει τον επόμενο φυσικό αριθμό και των πράξεων της πρόσθεσης, του πολλαπλασιασμού και της ύψωσης σε δύναμη.

Ορισμός 2.26

$$\begin{aligned} \boldsymbol{succ} &\equiv \lambda n.\, \lambda f.\, \lambda x.\, n\,f\,(f\,x) \\ \boldsymbol{A}_+ &\equiv \lambda n.\, \lambda m.\, \lambda f.\, \lambda x.\, n\,f\,(m\,f\,x) \\ \boldsymbol{A}_* &\equiv \lambda n.\, \lambda m.\, \lambda f.\, n\,(m\,f) \\ \boldsymbol{A}_{\exp} &\equiv \lambda n.\, \lambda m.\, m\,n \end{aligned}$$

Το θεώρημα 2.7 αποδεικνύει την ορθότητα της κωδικοποίησης των πράξεων του παραπάνω ορισμού. Για την απόδειξή του είναι χρήσιμο το παρακάτω λήμμα.

Λήμμα 2.1 *Για κάθε $n, m \in \mathbb{N}$ και για κάθε $x, y \in \Lambda$ ισχύουν τα εξής:*

(i) $\boldsymbol{c}_n\,f\,(\boldsymbol{c}_m\,f\,x) = \boldsymbol{c}_{n+m}\,f\,x$

(ii) $(\boldsymbol{c}_n\,x)^m(y) = x^{nm}(y)$

(iii) *Αν $m > 0$, τότε $(\boldsymbol{c}_n)^m(x) = \boldsymbol{c}_{n^m}\,x$*

Απόδειξη

(i)
$$\begin{array}{rcl} \boldsymbol{c}_n\, f\, (\boldsymbol{c}_m\, f\, x) & \equiv & (\lambda f.\, \lambda x.\, f^n(x))\, f\, ((\lambda f.\, \lambda x.\, f^m(x)))\, f\, x) \\ & \twoheadrightarrow_\beta & (\lambda f.\, \lambda x.\, f^n(x))\, f\, (f^m(x)) \\ & \twoheadrightarrow_\beta & f^n(f^m(x)) \\ & \equiv & f^{n+m}(x) \\ & \twoheadleftarrow_\beta & (\lambda f.\, \lambda x.\, f^{n+m}(x))\, f\, x \\ & \equiv & \boldsymbol{c}_{n+m}\, f\, x \end{array}$$

(ii) Με επαγωγή στο m. Για $m = 0$ προφανώς ισχύει $y = y$. Έστω ότι ισχύει για τυχαίο m. Τότε:

$$\begin{array}{rcll} (\boldsymbol{c}_n\, x)^{m+1}(y) & \equiv & \boldsymbol{c}_n\, x\, ((\boldsymbol{c}_n\, x)^m(y)) & \\ & = & \boldsymbol{c}_n\, x\, (x^{nm}(y)) & \langle \text{ επαγωγική υπόθεση } \rangle \\ & \equiv & (\lambda f.\, \lambda x.\, f^n(x))\, x\, (x^{nm}(y)) & \\ & \twoheadrightarrow_\beta & x^n(x^{nm}(y)) & \\ & \equiv & x^{n+nm}(y) & \\ & \equiv & x^{n(m+1)}(y) & \end{array}$$

(iii) Με επαγωγή στο m. Για $m = 1$ προφανώς ισχύει $\boldsymbol{c}_n\, x = \boldsymbol{c}_n\, x$. Έστω ότι ισχύει για τυχαίο $m > 0$. Τότε:

$$\begin{array}{rcll} (\boldsymbol{c}_n)^{m+1}(x) & \equiv & \boldsymbol{c}_n\, ((\boldsymbol{c}_n)^m(x)) & \\ & = & \boldsymbol{c}_n\, (\boldsymbol{c}_{n^m}\, x) & \langle \text{ επαγωγική υπόθεση } \rangle \\ & \equiv & (\lambda f.\, \lambda x.\, f^n(x))\, (\boldsymbol{c}_{n^m}\, x) & \\ & \rightarrow_\beta & \lambda y.\, (\boldsymbol{c}_{n^m}\, x)^n(y) & \\ & = & \lambda y.\, x^{n^m n}(y) & \langle \text{ λήμμα 2.1 (ii) } \rangle \\ & \equiv & \lambda y.\, x^{n^{m+1}}(y) & \\ & \leftarrow_\beta & (\lambda f.\, \lambda x.\, f^{n^{m+1}}(x))\, x & \\ & \equiv & \boldsymbol{c}_{n^{m+1}}\, x & \end{array}$$

□

Θεώρημα 2.7 *Για κάθε $n, m \in \mathbb{N}$ ισχύουν τα εξής:*

(i) $\boldsymbol{succ}\ \boldsymbol{c}_n = \boldsymbol{c}_{n+1}$

(ii) $\boldsymbol{A}_+\, \boldsymbol{c}_n\, \boldsymbol{c}_m = \boldsymbol{c}_{n+m}$

(iii) $\boldsymbol{A}_*\, \boldsymbol{c}_n\, \boldsymbol{c}_m = \boldsymbol{c}_{nm}$

(iv) *Αν $m > 0$ τότε* $\boldsymbol{A}_{\exp}\, \boldsymbol{c}_n\, \boldsymbol{c}_m = \boldsymbol{c}_{n^m}$

Απόδειξη

(i)
$$\begin{array}{rcl} \boldsymbol{succ}\ \boldsymbol{c}_n & \equiv & (\lambda n.\, \lambda f.\, \lambda x.\, n\, f\, (f\, x))\, (\lambda f.\, \lambda x.\, f^n(x)) \\ & \rightarrow_\beta & \lambda f.\, \lambda x.\, (\lambda f.\, \lambda x.\, f^n(x))\, f\, (f\, x) \\ & \twoheadrightarrow_\beta & \lambda f.\, \lambda x.\, f^n(f\, x) \\ & \equiv & \boldsymbol{c}_{n+1} \end{array}$$

$$
\begin{array}{llll}
\text{(ii)} & \boldsymbol{A}_{+}\, \boldsymbol{c}_n\, \boldsymbol{c}_m & \equiv & (\lambda n.\, \lambda m.\, \lambda f.\, \lambda x.\, n\, f\, (m\, f\, x))\, \boldsymbol{c}_n\, \boldsymbol{c}_m \\
 & & \twoheadrightarrow_\beta & \lambda f.\, \lambda x.\, \boldsymbol{c}_n\, f\, (\boldsymbol{c}_m\, f\, x) \\
 & & = & \lambda f.\, \lambda x.\, \boldsymbol{c}_{n+m}\, f\, x \qquad \langle \textit{λήμμα 2.1 (i)} \rangle \\
 & & \twoheadrightarrow_\eta & \boldsymbol{c}_{n+m}
\end{array}
$$

$$
\begin{array}{llll}
\text{(iii)} & \boldsymbol{A}_{*}\, \boldsymbol{c}_n\, \boldsymbol{c}_m & \equiv & (\lambda n.\, \lambda m.\, \lambda f.\, n\, (m\, f))\, \boldsymbol{c}_n\, \boldsymbol{c}_m \\
 & & \twoheadrightarrow_\beta & \lambda f.\, \boldsymbol{c}_n\, (\boldsymbol{c}_m\, f) \\
 & & \equiv & \lambda f.\, (\lambda f.\, \lambda x.\, f^n(x))\, (\boldsymbol{c}_m\, f) \\
 & & \rightarrow_\beta & \lambda f.\, \lambda x.\, (\boldsymbol{c}_m\, f)^n(x) \\
 & & = & \lambda f.\, \lambda x.\, f^{mn}(x) \qquad \langle \textit{λήμμα 2.1 (ii)} \rangle \\
 & & \equiv & \boldsymbol{c}_{nm}
\end{array}
$$

$$
\begin{array}{llll}
\text{(iv)} & \boldsymbol{A}_{\mathrm{exp}}\, \boldsymbol{c}_n\, \boldsymbol{c}_m & \equiv & (\lambda n.\, \lambda m.\, m\, n)\, \boldsymbol{c}_n\, \boldsymbol{c}_m \\
 & & \twoheadrightarrow_\beta & \boldsymbol{c}_m\, \boldsymbol{c}_n \\
 & & \equiv & (\lambda f.\, \lambda x.\, f^m(x))\, \boldsymbol{c}_n \\
 & & \rightarrow_\beta & \lambda x.\, (\boldsymbol{c}_n)^m(x) \\
 & & = & \lambda x.\, \boldsymbol{c}_{n^m}\, x \qquad \langle \textit{λήμμα 2.1 (iii)} \rangle \\
 & & \rightarrow_\eta & \boldsymbol{c}_{n^m}
\end{array}
$$

□

2.7.4 Λάμβδα λογισμός και πέρασμα παραμέτρων

Η σχέση ομοιότητας του λ-λογισμού με τις γλώσσες προγραμματισμού δεν περιορίζεται στην εκφραστική του ικανότητα ως προγραμματιστικού μοντέλου. Οι στρατηγικές αναγωγής λ-όρων είναι πολύ στενά συνδεδεμένες με τις μεθόδους περάσματος παραμέτρων των γλωσσών προγραμματισμού.

Στην αναλογία μεταξύ λ-λογισμού και γλωσσών προγραμματισμού:

- οι λ-όροι αντιστοιχούν σε εκφράσεις ή εντολές,
- η αφαίρεση και η εφαρμογή αντιστοιχούν στον ορισμό και την κλήση συναρτήσεων ή διαδικασιών, και
- η διαδικασία της αναγωγής αντιστοιχεί στην αποτίμηση εκφράσεων ή την εκτέλεση εντολών.

Σε ορισμένες γλώσσες προγραμματισμού, η αποτίμηση μιας έκφρασης πολλές φορές σταματά πριν προκύψει ένα πλήρως αποτιμημένο αποτέλεσμα. Το είδος αυτό της αποτίμησης ονομάζεται *οκνηρή αποτίμηση* (lazy evaluation) και υποστηρίζεται κυρίως από γλώσσες συναρτησιακού προγραμματισμού, όπως η Haskell και η Miranda. Η οκνηρή αποτίμηση μιας έκφρασης τερματίζεται όταν το αποτέλεσμα που προκύπτει είναι μια *τιμή* (value).[8] Η ακριβής μορφή των τιμών διαφέρει

[8] Στην αγγλική ορολογία, οι τιμές ονομάζονται επίσης *canonical forms*, όμως ο ελληνικός όρος *κανονική μορφή* έχει ήδη χρησιμοποιηθεί για να αποδώσει τον αγγλικό όρο *normal form*. Για το λόγο αυτό, η χρήση του όρου canonical form αποφεύγεται.

από γλώσσα σε γλώσσα, γενικά όμως οι αριθμητικές και λογικές σταθερές καθώς και οι συναρτησιακές αφαιρέσεις είναι τιμές. Κατ' αναλογία, στο λ-λογισμό μόνο οι αφαιρέσεις μπορούν να θεωρηθούν τιμές.[9]

Η στρατηγική της αναγωγής κανονικής σειράς σχετίζεται στενά με την οκνηρή αποτίμηση. Η μετατροπή κάθε φορά του αριστερότερου β-redex σημαίνει ότι δεν μπορεί να έχει προηγηθεί μετατροπή μέσα στον όρο, πάνω στον οποίο εφαρμόζεται η αφαίρεση που αντιστοιχεί σε αυτό το β-redex. Κατ' αναλογία, στην οκνηρή αποτίμηση μια συνάρτηση καλείται χωρίς να έχει προηγηθεί η αποτίμηση των παραμέτρων της. Στη γλώσσα προγραμματισμού Algol 60, η συμπεριφορά αυτή επιτυγχάνεται με τη μέθοδο του περάσματος παραμέτρων *κατ' όνομα* (call by name). Η στρατηγική της αναγωγής κανονικής σειράς οδηγεί τελικά την αναγωγή του όρου $(\lambda x.\, \lambda y.\, y)\, \Omega$ στην τιμή $\lambda y.\, y$. Στο ίδιο αποτέλεσμα καταλήγουμε στη γλώσσα Algol 60, αν ορίσουμε τις συναρτήσεις

```
integer procedure f (x);
   integer x;
begin
   f := 42
end;

integer procedure g;
begin
   while true do
      g := 7
end;
```

και στη συνέχεια αποτιμήσουμε την έκφραση `f(g)`. Το αποτέλεσμα είναι η τιμή `42` γιατί κατά την αποτίμηση δε χρειάζεται να γίνει η κλήση στη συνάρτηση `g`, που δεν τερματίζεται.

Αντίθετα, η αποτίμηση της έκφρασης `f(g)` στο παραπάνω παράδειγμα δε θα τερματιζόταν σε μια γλώσσα προγραμματισμού όπως η Algol 68, η Pascal και η C. Στις γλώσσες αυτές το πέρασμα των παραμέτρων γίνεται *κατ' αξία* (by value) και, κατά συνέπεια, η τιμή της παραμέτρου `g` πρέπει να αποτιμηθεί πριν κληθεί η συνάρτηση `f`. Η κλήση της `g` οδηγεί προφανώς σε μη τερματισμό. Αυτού του είδους η αποτίμηση κατά την οποία οι παράμετροι αποτιμώνται πριν γίνει η κλήση στη συνάρτηση ονομάζεται *πρόθυμη αποτίμηση* (eager evaluation).

Στο λ-λογισμό, το ανάλογο της πρόθυμης αποτίμησης και του περάσματος παραμέτρων κατ' αξία είναι μια στρατηγική αποτίμησης που κατά την αποτίμηση του όρου $(\lambda x.\, \lambda y.\, y)\, \Omega$ θα επιχειρούσε πρώτα να αποτιμήσει τον όρο Ω. Τέτοιου είδους στρατηγικές προκύπτουν υποχρεωτικά αν στον κανόνα της β-μετατροπής

$$(\lambda x.\, M)\, N \quad \to_\beta \quad M[x := N]$$

προσθέσουμε τον περιορισμό ότι ο όρος N πρέπει να είναι τιμή.

[9] Αν κανείς περιοριστεί στο σύνολο Λ^0 των κλειστών λ-όρων, οι (κλειστές) κανονικές μορφές είναι υποχρεωτικά αφαιρέσεις και άρα κάθε κανονική μορφή είναι τιμή. Το αντίστροφο όμως δεν είναι αληθές.

Ασκήσεις

2.1 Να αποδειχθεί ότι οι σχέσεις $\twoheadrightarrow_\alpha$ και $=_\alpha$ είναι ίσες, δηλαδή για κάθε $M, N \in \Lambda$ ισχύει $M \twoheadrightarrow_\alpha N \;\Leftrightarrow\; M =_\alpha N$.

2.2 Ο κλειστός όρος $\Theta \equiv (\lambda x.\, \lambda y.\, y\,(x\,x\,y))\,(\lambda x.\, \lambda y.\, y\,(x\,x\,y))$ είναι γνωστός ως *τελεστής σταθερού σημείου του Turing*. Να αποδειχθεί ότι ο όρος αυτός διαθέτει την ιδιότητα $\Theta\, F = F\,(\Theta\, F)$ και να σχολιασθεί κατά πόσο αυτός υπερτερεί έναντι του τελεστή Y.

2.3 Να οριστούν οι λογικές πράξεις ***and***, ***or*** και ***xor*** πάνω στην κωδικοποίηση των λογικών τιμών που περιγράφεται στην ενότητα 2.7.1 και να αποδειχθεί ότι πληρούν τις απαιτούμενες ιδιότητες.

2.4 Να αποδειχθεί ότι και ο όρος $\lambda x.\, \lambda y.\, \lambda z.\, y\,(x\,y\,z) \in \Lambda$ κωδικοποιεί τη συνάρτηση που επιστρέφει τον επόμενο φυσικό αριθμό, πάνω στα αριθμοειδή του Church.

2.5 Να αποδειχθεί ότι ο όρος

$$\boldsymbol{iszero} \;\equiv\; \lambda x.\, x\,(\lambda y.\, \lambda x.\, \lambda y.\, y)\,(\lambda x.\, \lambda y.\, x)$$

κωδικοποιεί τη συνάρτηση που, αν εφαρμοστεί στο αριθμοειδές του Church $\boldsymbol{c}_n$, επιστρέφει ***true*** αν $n = 0$ και ***false*** αν $n > 0$.

2.6 Να αποδειχθεί ότι ο όρος (που οφείλεται στον J. Velmans)

$$\boldsymbol{pred} \;\equiv\; \lambda x.\, \lambda y.\, \lambda z.\, x\,(\lambda p.\, \lambda q.\, q\,(p\,y))\,(\lambda y.\, z)\,(\lambda x.\, x)$$

κωδικοποιεί τη συνάρτηση που επιστρέφει τον προηγούμενο φυσικό αριθμό, πάνω στα αριθμοειδή του Church. Να μελετηθεί επίσης τί προκύπτει αν αυτός ο όρος εφαρμοστεί στο $\boldsymbol{c}_0$.

2.7 Να οριστεί ένας τρόπος κωδικοποίησης των διατεταγμένων n-άδων στο λ-λογισμό. Για κάθε $n \in \mathbb{N}$ και για κάθε $M_0, M_1, \ldots, M_{n-1} \in \Lambda$, ο όρος

$$\boldsymbol{tuple}\; \boldsymbol{c}_n\, M_0\, M_1\, \ldots\, M_{n-1}$$

πρέπει να αποτελεί κωδικοποίηση της διατεταγμένης n-άδας $\langle M_0, M_1, \ldots, M_{n-1}\rangle$. Επιπλέον, για κάθε $i \in \mathbb{N}$ τέτοιο ώστε $i < n$, ο όρος $\boldsymbol{prj}\; \boldsymbol{c}_i$ πρέπει να κωδικοποιεί την πράξη της προβολής του i-οστού στοιχείου της n-άδας. Αφού οριστούν κατάλληλα οι όροι $\boldsymbol{tuple}, \boldsymbol{prj} \in \Lambda$ να αποδειχθεί ότι, με τους παραπάνω περιορισμούς, ικανοποιείται η σχέση

$$\boldsymbol{prj}\; \boldsymbol{c}_i\, \langle M_0,\, M_1\, \ldots\, M_{n-1}\rangle \;=\; M_i$$

2.8 Στη γλώσσα προγραμματισμού της αρεσκείας σας, υλοποιήστε ένα διερμηνέα για το λ-λογισμό.