Κεφάλαιο 10
Λάμβδα λογισμός

Π. Ροντογιάννης

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Ιστορική εξέλιξη λ-λογισμού

- Αναπτύχθηκε αρχικά από τον Alonzo Church στις αρχές της δεκαετίας του 1930, πολύ πριν αρχίσουν να χρησιμοποιούνται οι ηλεκτρονικοί υπολογιστές.
- Ήταν μέρος μιας γενικότερης θεωρίας με στόχο τη θεμελίωση μαθηματικών και λογικής [Chur32, Chur33]

Ιστορική εξέλιξη λ-λογισμού

- Η γενική θεωρία ήταν ασυνεπής, όπως αποδείχθηκε αργότερα [Klee35]
- Το τμήμα της, που ασχολήθηκε με συναρτήσεις είχε σημαντικές εφαρμογές στην πληροφορική, κυρίως μετά το 1960
- Το τμήμα αυτό είναι ο λάμβδα λογισμός ή λ-λογισμός

λ-Λογισμός

- Πλήρες υπολογιστικό μοντέλο
- Δύο πολύ σημαντικά αποτελέσματα
  - Όλες οι αναδρομικές συναρτήσεις παριστάνονται 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 οδήγησε στη δημιουργία των πρώτων γλωσσών προγραμματισμού

λ-Λογισμός

- Σχεδιασμός νέων αρχιτεκτονικών υπολογιστών
  - Μηχανές αναγωγής (reduction machines) και
  - Υπολογιστές ροής δεδομένων (data-flow computers)
    - Όταν πρωτο-δημιουργήθηκαν εκτελούσαν αποκλειστικά προγράμματα γραμμένα σε κάποια διάλεκτο του λ-λογισμού.
  - Δημιουργία συναρτησιακού προγραμματισμού
    - John McCarthy σχεδίασε τη γλώσσα προγραμματισμού LISP στα τέλη της δεκαετίας του 1950. Αργότερα δημιουργήθηκαν οι Scheme, ML, Miranda, Haskell, κλπ.

λ-Λογισμός

- Υπολογιστές και συναρτησιακός προγραμματισμός δεν έτυχαν ευρείας αποδοχής όπως οι υπολογιστές von Neumann και ο προστακτικός προγραμματισμός
- Οι επιδόσεις των πρώτων μηχανών αναγωγής και οι υλοποιήσεις των συναρτησιακών γλωσσών ήταν χειρότερες από τα παραδοσιακά συστήματα.

λ-Λογισμός

- Ιδιαίτερα πρόσφορος ως συμβολισμός για την περιγραφή σημασιολογικών ιδιοτήτων των γλωσσών προγραμματισμού.
- Διευκόλυνε τη μελέτη και απομόνωση προβλημάτων σχεδίασης και υλοποίησης των γλωσσών προγραμματισμού
    - Μηχανισμό κλήσης υπο-προγραμμάτων και δομή συστήματος τύπων

λ-Λογισμός

- Διατύπωση θεωρίας πεδίων
- Θεμελίωση ερευνητικού πεδίου της σημασιολογίας γλωσσών προγραμματισμού

Διαισθητική εισαγωγή

- Ο λ-λογισμός είναι θεωρία συναρτήσεων
- Δύο κύριες λειτουργίες
  - *Εφαρμογή* συνάρτησης F πάνω σε ένα όρισμα A, που συμβολίζεται με $F\ A$
  - *Αφαίρεση*. Έστω ότι x μεταβλητή και $E[x]$ έκφραση, που εξαρτάται από τη x. Η έκφραση $\lambda x.E[x]$ συμβολίζει τη συνάρτηση

$$x \longmapsto E[x]$$

Διαισθητική εισαγωγή

- Η συνάρτηση δέχεται ως όρισμα μία τιμή v και επιστρέφει ως αποτέλεσμα την τιμή $E[v]$.
- Η x δεν εμφανίζεται απαραίτητα στην έκφραση $E[x]$. Αν αυτό δεν συμβαίνει, τότε η $\lambda x.E[x]$ είναι μία σταθερή συνάρτηση.

Παράδειγμα

- Η $\lambda x.x^2 - 3x + 2$ συμβολίζει μία συνάρτηση, που σε κάθε τιμή x απεικονίζει την τιμή $x^2 - 3x + 2$.
- Αν η συνάρτηση εφαρμοσθεί στο όρισμα 8, προκύπτει

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

- Η τιμή του ορίσματος αντικαθιστά την παράμετρο x

Ελεύθερη και Δεσμευμένη Μεταβλητή

- Η αφαίρεση $\lambda x.E[x]$ δεσμεύει τη μεταβλητή x μέσα στην έκφραση $E[x]$.
- Μεταβλητή μη δεσμευμένη ονομάζεται ελεύθερη.
  - Παράδειγμα: $\lambda x.x^2 - 3y + 2$
    x δεσμευμένη και y ελεύθερη
  - Παράδειγμα: $(\lambda x.x^2 - 3y + 2)(4x + 1)$
    Η πρώτη εμφάνιση του x (στο x^2) είναι δεσμευμένη, γιατί είναι στο εσωτερικό της αφαίρεσης, ενώ η δεύτερη (στο $4x + 1$) είναι ελεύθερη)

Παράδειγμα Δέσμευσης

- Παράδειγμα:

$$\int \frac{\sin x + \cos y}{\cos x - \sin y} dx$$

- 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.
- Η μεταβλητή y είναι ελεύθερη και η τιμή του ολοκληρώματος εξαρτάται από την τιμή του y έξω από αυτό.

λ-όροι

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

Ορισμός 10.1

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

1. $x \in V \Rightarrow x \in \Lambda$
2. $M, N \in \Lambda \Rightarrow (M\ N) \in \Lambda$
3. $x \in V,\ M \in \Lambda \Rightarrow (\lambda x.M) \in \Lambda$

λ-όροι

- Τα στοιχεία του συνόλου Λ ονομάζονται επίσης λ-όροι (λ-terms). Υπάρχουν τριών ειδών:
  - Μεταβλητές (Variables), δηλ. στοιχεία του συνόλου V
  - Εφαρμογές (Applications), με μορφή $(M\ N)$, όπου M και N είναι λ-όροι
  - Αφαιρέσεις (Abstractions), με μορφή $(\lambda x.M)$, όπου x μεταβλητή και M λ-όρος
- Κατά σύμβαση χρησιμοποιούνται μικρά γράμματα του λατινικού αλφαβήτου (x, y, z κλπ) για συμβολισμό μεταβλητών και κεφαλαία (M, N, F, G, P κλπ) για λ-όρους.

λ-όροι

- Χρησιμοποιώντας αφηρημένη σύνταξη BNF και θεωρώντας ότι η συντακτική κλάση των μεταβλητών παριστάνεται με το μη-τερματικό σύμβολο $\langle var \rangle$, η γλώσσα Λ των λ-όρων περιγράφεται ισοδύναμα

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

Παράδειγμα

Οι παρακάτω είναι λ-όροι:

Παράδειγμα 10.1

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

Συμβάσεις

- Για απλοποίηση των λ-όρων και αποφυγή μεγάλου αριθμού παρενθέσεων (με αυστηρή τήρηση ορισμού 10.1) χρησιμοποιούνται οι συμβάσεις
  - Εξωτερικές παρενθέσεις δεν γράφονται
    - $\lambda x.x$ συντομογραφία του $(\lambda x.x)$
  - Η εφαρμογή είναι αριστερά προσεταιριστική
    - $F\ M_1\ M_2\ \ldots\ M_n$ συντομογραφία του $(\ldots((F\ M_1)M_2)\ldots\ M_n)$
  - Η αφαίρεση εκτείνεται όσο περισσότερο είναι δυνατό, δηλαδή ως το επόμενο κλείσιμο παρένθεσης ή το τέλος του όρου:
    - $\lambda x.\ M_1\ M_2\ \ldots\ M_n$ συντομογραφία του $\lambda x.(M_1\ M_2\ \ldots\ M_n)$

Παράδειγμα

Ακολουθώντας τις συμβάσεις, οι λ-όροι του παραδείγματος 10.1 γράφονται

Παράδειγμα 10.1

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

Σχέση $\equiv$

Ορισμός 10.2

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

1. $x \equiv y$ **αν** $x = y$
2. $(M\ N) \equiv (P\ Q)$ **αν** $M \equiv P$ **και** $N \equiv Q$
3. $(\lambda x.M) \equiv (\lambda y.N)$ **αν** $x = y$ **και** $M \equiv N$

Όπου $x = y$ 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).

Μεταβλητές

- 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 δεσμεύουσα μεταβλητή είναι πάλι η x.

Μεταβλητές

- Εμφανίσεις της μεταβλητής x που βρίσκονται στην εμβέλεια κάποιου λx ονομάζονται δεσμευμένες (bound).
- Εμφανίσεις, που δεν βρίσκονται στην εμβέλεια κανενός λx ονομάζονται ελεύθερες.

Ελεύθερη μεταβλητή

Ορισμός 10.3

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

1. $FV(x) = \{x\}$
2. $FV(M\ N) = FV(M) \cup FV(N)$
3. $FV(\lambda x.M) = FV(M) - \{x\}$

Παράδειγμα

Έστω ο παρακάτω όρος:

$$M \equiv (\lambda x.y\,x)(\lambda y.xy)$$

Ελεύθερη μεταβλητή
Δεσμευμένη μεταβλητή

Παράδειγμα

Παράδειγμα 10.2

Ακολουθώντας τον ορισμό 10.3

$$
\begin{aligned}
& FV(M) = \\
&= FV((\lambda x.yx)(\lambda y.xy)) \\
&= FV(\lambda x.yx) \cup FV(\lambda y.xy) \\
&= (FV(y\ x) - \{x\}) \cup (FV(x\ y) - \{y\}) \\
&= ((FV(y) \cup FV(x)) - \{x\}) \cup ((FV(x) \cup FV(y)) - \{y\}) \\
&= ((\{y\} \cup \{x\}) - \{x\}) \cup ((\{x\} \cup \{y\}) \cup \{y\}) \\
&= (\{x, y\} - \{x\}) \cup (\{x, y\} - \{y\}) \\
&= \{y\} \cup \{x\} \\
&= \{x, y\}
\end{aligned}
$$

Κλειστοί λ-όροι

Ορισμός 10.4

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

Κλειστοί λ-όροι

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

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

Αντικατάσταση

- Βασική πράξη συμβολικής επεξεργασίας των λ-όρων
- Επηρεάζει ελεύθερες μεταβλητές
- Ο συμβολισμός $M[x := N]$ παριστάνει το αποτέλεσμα αντικατάστασης στον όρο M όλων των ελεύθερων εμφανίσεων της μεταβλητής x με τον όρο N.

Αντικατάσταση

Ορισμός 10.5

Η αντικατάσταση 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:*

- $x[x := N] \equiv N$
- $y[x := N] \equiv y$
- $(P\,Q)[x := N] \equiv P[x := N]\,Q[x := N]$
- $(\lambda x.P)[x := N] \equiv \lambda x.P$
- $(\lambda y.P)[x := N] \equiv \lambda y.P[x := N]$,
  $y \notin FV(N)$ ή $x \notin FV(P)$
- $(\lambda y.P)[x := N] \equiv \lambda z.P[y := z][x := N]$,
  $y \in FV(N)$ *και* $x \in FV(P), z \notin FV(N) \cup FV(P)$

Παρατηρήσεις

- Αντικαθίστανται μόνο οι ελεύθερες εμφανίσεις μιας μεταβλητής
- Στο αποτέλεσμα της αντικατάστασης καμία από τις ελεύθερες μεταβλητές του όρου N δεν θα γίνει δεσμευμένη
- Η δέσμευση ελεύθερης μεταβλητής του όρου N θα μπορούσε να γίνει μόνο μέσω του τελευταίου κανόνα, στην περίπτωση, που $y \notin FV(N)$ και $x \notin FV(P)$. Ο κανόνας μεριμνά για μετονομασία της δεσμεύουσας μεταβλητής, προκειμένου να αποφευχθεί αυτό το ενδεχόμενο.

Παρατηρήσεις

- Στον τελευταίο κανόνα η επιλογή της μεταβλητής $z \in V$ είναι ελεύθερη υπό την προϋπόθεση ότι $z \notin FV(N) \cup FV(P)$.
- Για τον μονοσήμαντο ορισμό του αποτελέσματος της αντικατάστασης $M[x := N]$ πρέπει σε αυτό τον κανόνα να καθορίζεται επακριβώς ο τρόπος επιλογής της μεταβλητής αυτής.
- Τα στοιχεία του συνόλου V είναι αριθμημένα και επιλέγεται ως z το πρώτο, που ικανοποιεί την $z \notin FV(N) \cup FV(P)$.

Μετατροπές

- Τρία είδη μετατροπής (α,β και η)
- Κάθε είδος μετατροπής περιγράφεται από τον κανόνα $M \to_x N$ όπου $x \in \{\alpha, \beta, \eta\}$ και $M, N \in \Lambda$. Ο όρος M ονομάζεται x-redex ή απλά redex και ο N όρος contractum.

Ιδιότητες σχέσεων

Ορισμός 10.6

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:*

- $M \sim N \Rightarrow (M\,P) \sim (N\,P)$
- $M \sim N \Rightarrow (P\,M) \sim (P\,N)$
- $M \sim N \Rightarrow (\lambda x.M) \sim (\lambda x.N)$

Η σχέση ταυτότητας $\equiv$ μεταξύ των λ-όρων είναι συμβατή.

Ιδιότητες σχέσεων

Ορισμός 10.7

Μία σχέση αναγωγής (reduction relation) είναι μία συμβατή, ανακλαστική και μεταβατική σχέση πάνω στο Λ.

Ορισμός 10.8

Μία σχέση συμφωνίας (congruence relation) είναι μία συμβατή σχέση ισοδυναμίας πάνω στο Λ.

(Μία σχέση ονομάζεται σχέση ισοδυναμίας (equivalence relation) αν είναι ανακλαστική, μεταβατική και συμμετρική.)

α-μετατροπή

Ορισμός 10.9

Η σχέση $\to_\alpha$ 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 $y \notin FV(M)$ ισχύει

$$\lambda x.M \to_\alpha \lambda y.M[x := y]$$

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

Παράδειγμα 10.3

Οι παρακάτω α-μετατροπές είναι σωστές:

- $\lambda x.x \rightarrow_\alpha \lambda y.y$
- $\lambda x.z\,x \rightarrow_\alpha \lambda y.z\,y$
- $\lambda x.\lambda y.z\,x\,y \rightarrow_\alpha \lambda y.\lambda w.z\,y\,w$

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.

Παράδειγμα 10.4

Η παρακάτω μετατροπή είναι λανθασμένη:

$$\lambda x.\lambda z.z\,x\,y \to_{\alpha} \lambda y.\lambda z.z\,y\,y$$

γιατί δεν πληρείται ο περιορισμός

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

β-μετατροπή

Ορισμός 10.10

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 $x, y \in V$ ισχύει

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

β-μετατροπή

Παράδειγμα 10.5

Οι παρακάτω β-μετατροπές είναι σωστές

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

β-μετατροπή

Παράδειγμα 10.6

Η παρακάτω μετατροπή είναι λανθασμένη:

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

γιατί για να γίνει αντικατάσταση

$$(\lambda y.z\,x\,y)[x := wy]$$

πρέπει να μετονομασθεί 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 νέα μεταβλητή.

η-μετατροπή

Ορισμός 10.11

Η σχέση $\to_\eta$ *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* $x \notin FV(M)$ *ισχύει*

$$\lambda x. M\, x \to_\eta M$$

η-μετατροπή

Παράδειγμα 10.7

Οι παρακάτω η-μετατροπές είναι σωστές:

- $\lambda x.z\,x \to_\eta z$
- $\lambda y.z\,x\,y \to_\eta z\,x$

όχι όμως η μετατροπή $\lambda x.z\,x\,x \to_\eta z\,x$ γιατί δεν πληρείται το $x \notin FV(z\,x)$.

Σχέση $\rightarrow$

- Οι τρεις κανόνες μετατροπής α, β και η ορίζουν τρεις βασικές σχέσεις μετατροπής $\rightarrow_\alpha, \rightarrow_\beta, \rightarrow_\eta$ για το λ-λογισμό.
- Η συνολική σχέση μετατροπής $M \rightarrow N$ θα υποδηλώνει ότι ο όρος M μετατρέπεται σε ένα βήμα στον όρο N με έναν από τους τρεις κανόνες.

Σχέση $\rightarrow$

Ορισμός 10.12

Με $\rightarrow$ συμβολίζεται η ένωση των σχέσεων $\rightarrow_\alpha, \rightarrow_\beta, \rightarrow_\eta$, δηλαδή η σχέση $\rightarrow$ ορίζεται ως η μικρότερη σχέση για την οποία ισχύουν:

- $M \rightarrow_\alpha N \Rightarrow M \rightarrow N$
- $M \rightarrow_\beta N \Rightarrow M \rightarrow N$
- $M \rightarrow_\eta N \Rightarrow M \rightarrow N$

Σχέση $\twoheadrightarrow$

- Ο όρος M μετατρέπεται στον όρο N με ακολουθία βημάτων, πιθανώς κενή.

$$M \equiv N_0 \to N_1 \to N_2 \to \cdots \to N_{n-1} \to N_n \equiv N$$

  Η σχέση συμβολίζεται

$$M \twoheadrightarrow N$$

- Αν $n = 1$, τότε η μετατροπή γίνεται σε ένα βήμα $M \to N$ και άρα η σχέση $\twoheadrightarrow$ εμπεριέχει την $\to$.
- Αν $n = 0$, η ακολουθία είναι κενή και δε συμβαίνει μετατροπή, δηλαδή $M \equiv N$ και άρα η σχέση $\twoheadrightarrow$ εμπεριέχει την $\equiv$.

Σχέση $\twoheadrightarrow$

Ορισμός 10.13

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

- $M \rightarrow N \Rightarrow M \twoheadrightarrow N$
- $M \twoheadrightarrow M$
- $M \twoheadrightarrow N, N \twoheadrightarrow P \Rightarrow M \twoheadrightarrow P$

Παρατηρήσεις

- Η σχέση $\twoheadrightarrow$ είναι σχέση αναγωγής (ορισμός 10.7). Όταν ισχύει ότι $M \twoheadrightarrow N$ 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 δεξιά.

Παρατηρήσεις

- Αν επιτραπεί η ύπαρξη «αντίστροφων βημάτων, όπως

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

  τότε οδηγούμαστε σε μία γενικότερη σχέση, που συμβολίζουμε με $=$. Η σχέση αυτή, εμπεριέχει τις $\twoheadrightarrow$, $\to$ και $\equiv$.

Σχέση $=$

Ορισμός 10.14

Με $=$ συμβολίζεται η σχέση ισοδυναμίας, που προκύπτει από την $\twoheadrightarrow$. Δηλαδή η σχέση $=$ ορίζεται ως η μικρότερη σχέση για την οποία ισχύουν:

- $M \twoheadrightarrow N \Rightarrow M = N$
- $M = N \Rightarrow N = M$
- $M = N, N = P \Rightarrow M = N$

Σχέσεις $=, \equiv$

- Η σχέση $=$ είναι σχέση συμφωνίας (ορισμός 10.8). Όταν ισχύει $M = N$, ο όρος M είναι ίσος με τον όρο N.
- Με τη σχέση ισότητας $M = N$ δεν εννοείται ότι οι δύο όροι ταυτίζονται, δηλαδή δεν ισχύει $M \equiv N$ (ορισμός 10.2).
- Η διάκριση των δύο σχέσεων και η επιλογή του συμβόλου «$=$» για τη σχέση συμφωνίας του ορισμού 10.14 ενδέχεται να μπερδέψει τον αναγνώστη.

Κανονικές μορφές

- λ-λογισμός θεωρείται ως υπολογιστικό μοντέλο επεξεργασίας συναρτήσεων
- Σχέσεις μετατροπής
  - παριστάνουν την πραγματοποίηση βημάτων επεξεργασίας πάνω στους λ-όρους
  - αποσκοπούν στον υπολογισμό κάποιου αποτελέσματος

β-μετατροπή

- Κατ' εξοχήν υπολογιστική πράξη, αφού διαισθητικά παριστάνει εφαρμογή μίας συνάρτησης σε κάποιο όρισμα

α-μετατροπή

- Δεν προάγει τη διαδικασία υπολογισμού.
- Τα ονόματα των δεσμευμένων μεταβλητών δεν έχουν ουσιαστική σημασία.
- Η μετονομασία των μεταβλητών, που επιτελείται μέσω της α-μετατροπής δεν αποτελεί ουσιαστικό υπολογιστικό βήμα

η-μετατροπή

- Δεν συμβάλλει άμεσα στην υπολογιστική διαδικασία.
- Υλοποιεί έναν μηχανισμό για την υλοποίηση λ-όρων, που παριστάνουν συναρτήσεις

Παρατηρήσεις

- Η σχέση μετατροπής $M \to N$ παριστάνει ένα βήμα στη διαδικασία υπολογισμού
- Η σχέση $M \twoheadrightarrow N$ παριστάνει (πιθανώς κενή) ακολουθία από τέτοια βήματα.
- Όταν $M \twoheadrightarrow N$ μπορεί να θεωρηθεί ότι ο όρος N προέκυψε κατά την αποτίμηση του όρου M.
  - Αποτίμηση ενός όρου είναι η διαδοχική εφαρμογή κανόνων μετατροπής σε αυτόν.

Κανονικές μορφές

- Όταν σε έναν όρο M δεν εφαρμόζεται κανένα αμιγώς υπολογιστικό βήμα, δηλαδή καμία β- ή η-μετατροπή, τότε η αποτίμηση του θεωρείται ολοκληρωμένη και ο όρος είναι τελικό αποτέλεσμα.
- Τέτοιοι πλήρως αποτιμημένοι όροι ονομάζονται *κανονικές μορφές*.

Κανονικές μορφές

Ορισμός 10.15

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

Παράδειγμα

Παράδειγμα 10.8

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

Παρατήρηση

- Στις κανονικές μορφές δεν λαμβάνεται υπόψη η α-μετατροπή. Αυτό είναι σύμφωνο με τη θεώρηση του λ-λογισμού ως υπολογιστικού μοντέλου. Αν δεν συνέβαινε, τότε απλοί όροι όπως ο $\lambda x.x$ δεν θα ήταν σε κανονική μορφή λόγω δυνατής μετονομασίας κάποιας δεσμεύουσας μεταβλητής.

Παρατήρηση

- 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

Πρόταση 10.1

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

Σχόλια

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

Κανονικές μορφές

Ορισμός 10.16

Ένας όρος $M \in \Lambda$ *λέμε ότι έχει* **κανονική μορφή** *αν για κάποιον 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 10.9

Ο όρος $\lambda z.(\lambda f.\lambda x.\ f\,z\,x)\,(\lambda y.y)$ στο παράδειγμα 8 δεν είναι σε κανονική μορφή. Εφαρμόζοντας διαδοχικά βήματα

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

και άρα $\lambda z.(\lambda f.\lambda x.\ f\,z\,x)\,(\lambda y.y) \twoheadrightarrow \lambda z.z$

Καθώς $\lambda z.z$ σε κανονική μορφή και ο αρχικός όρος έχει κανονική μορφή.

Κανονικές μορφές

Παράδειγμα 10.10

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

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

Σε ένα βήμα μετατροπής, ο όρος Ω μετατρέπεται πάλι στον εαυτό του. Η διαδικασία αποτίμησης δεν τερματίζεται και ο όρος Ω δεν έχει κανονική μορφή. Εδώ η διαδικασία αποτίμησης δεν προοδεύει.

Κανονικές μορφές

Παράδειγμα 10.11

Έστω ο όρος $M \equiv (\lambda x.x\,x\,y)(\lambda x.x\,x\,y)$ με μόνο ένα β-redex. Η μόνη ακολουθία μετατροπών που εφαρμόζεται είναι η

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

Κάθε αποτέλεσμα δεν είναι σε κανονική μορφή και έχει μόνο ένα β-redex.

Κανονικές μορφές

Παράδειγμα 10.12

Έστω ο όρος $M \equiv (\lambda x.(\lambda y.\, x\, y)\, z)\, w$ με δύο β-redex για τις δύο αφαιρέσεις με δεσμεύουσες μεταβλητές x και y.

Μετατρέποντας πρώτα το πρώτο από αυτά, προκύπτει η αποτίμηση:

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

Μετατρέποντας τα με την αντίστροφη σειρά προκύπτει η αποτίμηση:

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

Κανονικές μορφές

Παράδειγμα 10.13

Έστω ο όρος $M \equiv (\lambda z.\, y)\,((\lambda x.x\,x)(\lambda x.x\,x))$ που περιέχει τον όρο Ω του παραδείγματος 10.10. Ο όρος M περιέχει δύο β-redex για τις δύο αφαιρέσεις με δεσμεύουσες μεταβλητές z και την πρώτη από τις δύο x.

Μετατρέποντας το πρώτο, οδηγούμαστε σε κανονική μορφή:

$$M \to_\beta y$$

Μετατρέποντας το δεύτερο, οδηγούμαστε πάλι στον ίδιο όρο (όπως με τη μετατροπή του Ω).

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

Κανονικές μορφές

Ορισμός 10.17

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

Κανονικές μορφές

Ορισμός 10.18

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).

Ιδιότητες του λ-λογισμού

Θεώρημα 10.1 (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* $N_2 \twoheadrightarrow N$.

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

Ιδιότητες του λ-λογισμού

Πρόταση 10.2

Κάθε όρος M έχει το πολύ μία κανονική μορφή.

Κανονικοποίηση - Normalization

Θεώρημα 10.2

Αν ο όρος M έχει κανονική μορφή, τότε η στρατηγική της αναγωγής κανονικής σειράς οδηγεί σε αυτήν.

Λογικές τιμές

- Πως μπορούμε να αναπαραστήσουμε τις λογικές τιμές στον λάμβδα-λογισμό;

Ορισμός 10.19

$$\textbf{true} \equiv \lambda x.\lambda y.x$$

$$\textbf{false} \equiv \lambda x.\lambda y.y$$

Λογικές τιμές

- Πως μπορούμε να αναπαραστήσουμε τους λογικούς τελεστές στον λάμβδα-λογισμό;

Ορισμός 10.20

$$\textbf{not} \equiv \lambda z.z \ \textbf{false} \ \textbf{true}$$

Λογικές τιμές

Θεώρημα 10.3

1. **not true** $=$ **false**
2. **not false** $=$ **true**

Απόδειξη.

1. $$\begin{array}{rcl} \textbf{not true} & \equiv & (\lambda z.z\ \textbf{false true})\ \textbf{true} \\ & \to_\beta & \textbf{true false true} \\ & \equiv & (\lambda x.\lambda y.x)\ \textbf{false true} \\ & \to_\beta & \textbf{false} \end{array}$$
2. Ομοίως.

$\square$

Λογικές τιμές

Ορισμός 10.21

$$\textbf{cond} \equiv \lambda z.\lambda x.\lambda y.z\,x\,y$$

$$\textbf{if}\ B\ \textbf{then}\ N\ \textbf{else}\ M \equiv \textbf{cond}\ B\ N\ M$$

- Αποδεικνύεται εύκολα ότι η παραπάνω δομή πληροί τις απαιτούμενες ιδιότητες.

Λογικές τιμές

Θεώρημα 10.4

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

1. $\textbf{if true then } N \textbf{ else } M = N$
2. $\textbf{if false then } N \textbf{ else } M = M$

Απόδειξη.

1. $$\begin{array}{rcl} \textbf{if true then } N \textbf{ else } M & \equiv & \textbf{cond true } N \; M \\ & \equiv & (\lambda z.\lambda x.\lambda y.z\,x\,y)\textbf{true } N\, M \\ & \to_\beta & \textbf{true } N \; M \\ & \equiv & (\lambda x.\lambda y.x) \; N \; M \\ & \to_\beta & N \end{array}$$
2. Ομοίως.

□

Διατεταγμένα ζεύγη

- Στον λ-λογισμό κωδικοποιούνται διατεταγμένα ζεύγη όρων (που με τη σειρά τους κωδικοποιούν άλλα μαθηματικά αντικείμενα). Μία τέτοια κωδικοποίηση γίνεται μέσω του **pair**.
- Ο συμβολισμός $\langle N, M \rangle$ χρησιμοποιείται για διευκόλυνση στη γραφή όρων, που κωδικοποιούν διατεταγμένα ζεύγη.

Διατεταγμένα ζεύγη

Ορισμός 10.22

$$\textbf{pair} \equiv \lambda x.\lambda y.\lambda z.\ z\,x\,y$$

$$\langle N, M\rangle \equiv \textbf{pair}\ N\ M$$

- Οι πράξεις **fst** και **snd**, που επιστρέφουν το πρώτο και το δεύτερο στοιχείο ενός διατεταγμένου ζεύγους κωδικοποιούνται ως:

Ορισμός 10.23

$$\textbf{fst} \equiv \lambda z.\ z\ \textbf{true}$$

$$\textbf{snd} \equiv \lambda z.\ z\ \textbf{false}$$

- Αποδεικνύεται ότι οι πράξεις πληρούν απαιτούμενες ιδιότητες και άρα η δομή $\langle N, M\rangle$ χρησιμεύει ως διατεταγμένο ζεύγος.

Διατεταγμένα ζεύγη

Θεώρημα 10.5

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

1. $\textbf{fst}\ \langle N, M\rangle = N$
2. $\textbf{snd}\ \langle N, M\rangle = M$

Απόδειξη.

1. $$\begin{array}{lcl} \textbf{fst}\ \langle N, M\rangle & \equiv & (\lambda z.\ z\ \textbf{true})\ \langle N, M\rangle \\ & \to_\beta & \langle N, M\rangle\ \textbf{true} \\ & \equiv & \textbf{pair}\ N\ M\ \textbf{true} \\ & \equiv & (\lambda x.\lambda y.\lambda z.\ z\,x\,y)\ N\ M\ \textbf{true} \\ & \to_\beta & \textbf{true}\ N\ M \\ & \equiv & (\lambda x.\lambda y.x)\ N\ M \\ & \to_\beta & N \end{array}$$
2. Ομοίως.

Φυσικοί αριθμοί

- Έχουν προταθεί πολλές διαφορετικές κωδικοποιήσεις των φυσικών αριθμών στον λ-λογισμό. Η πρώτη και η πιο γνωστή από αυτές είναι τα αριθμοειδή του Church.

Φυσικοί αριθμοί

Ορισμός 10.24

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

- $F^0(A) \equiv A$
- $F^{n+1}(A) \equiv F(F^n(A))$

- 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 αποδεικνύεται ότι $F^n(F\ A) \equiv F(F^n(A))$ και $F^n(F^m(A)) \equiv F^{n+m}(A)$.

Αριθμοειδή του Church - Church numerals

Ορισμός 10.25

Για κάθε φυσικό αριθμό $n \in \mathbb{N}$ *ορίζεται ένας όρος* $\mathbf{c_n} \in \Lambda$ *ως:*

$$\mathbf{c_n} \equiv \lambda f.\lambda x.\ f^n(x)$$

Αριθμοειδή του Church

- Το αριθμοειδές, που αντιστοιχεί στον αριθμό 0 είναι το $\mathbf{c}_0 \equiv \lambda f.\lambda x.\, x$, στον αριθμό 1 το $\mathbf{c}_1 \equiv \lambda f.\lambda x.\, f\, x$, στον αριθμό 2 το $\mathbf{c}_2 \equiv \lambda f.\lambda x.\, f(f\, x)$ κοκ.
- Όλα τα αριθμοειδή είναι σε β-κανονική μορφή (μάλιστα όλα εκτός του $\mathbf{c}_1$ είναι και σε η-κανονική μορφή). Ούτε όλοι οι λ-όροι είναι αριθμοειδή, ούτε όλοι ανάγονται σε αριθμοειδή.

- Λ-όροι για επόμενο, πρόσθεση, πολλαπλασιασμό και ύψωση σε δύναμη.

Ορισμός 10.26

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

- Το θεώρημα 10.7 αποδεικνύει την ορθότητα της κωδικοποίησης πράξεων του παραπάνω ορισμού. Για την απόδειξη του είναι χρήσιμο το ακόλουθο λήμμα.

Αριθμοειδή του Church

Λήμμα 10.1

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

1. $\mathbf{c_n}\ f(\mathbf{c_m}\ f\ x) = \mathbf{c_{n+m}}\ f\ x$
2. $(\mathbf{c_n}\ x)^m\ (y) = x^{nm}\ (y)$
3. *Αν* $m > 0$, *τότε* $(\mathbf{c_n})^m\ (x) = \mathbf{c_{n^m}}\ x$

Αριθμοειδή του Church

Απόδειξη.

1. $$\begin{array}{rcl} \mathbf{c_n} f(\mathbf{c_m}\, f\, x) & \equiv & (\lambda f.\lambda x.\, f^n(x))\, f((\lambda f.\lambda x.\, f^m(x)) f\, x) \\ & \twoheadrightarrow_\beta & (\lambda f.\lambda x.\, f^n(x))\, f(f^m(x)\, x) \\ & \twoheadrightarrow_\beta & f^n(f^m(x)) \\ & \equiv & f^{n+m}(x) \\ & \twoheadleftarrow_\beta & (\lambda f.\lambda x.\, f^{n+m}(x))\, f\, x \\ & \equiv & \mathbf{c_{n+m}}\, f\, x \end{array}$$

Αριθμοειδή του Church

Απόδειξη (συνέχεια).

2.
$$
\begin{aligned}
(\mathbf{c_n}\, x)^{m+1}(y) &\equiv \mathbf{c_n}\, x((\mathbf{c_n}\, x)^m(y)) \\
&= \mathbf{c_n}\, x(x^{nm}(y)) \ \textit{(επαγωγική υπόθ.)} \\
&\equiv (\lambda f.\lambda x.\, f^n(x))\, x(x^{nm}(y)) \\
&\twoheadrightarrow_\beta x^n(x^{nm}(y)) \\
&\equiv x^{n+nm}(y) \\
&\equiv x^{n(1+m)}(y)
\end{aligned}
$$

Απόδειξη (συνέχεια).

3.
$$
\begin{aligned}
(\mathbf{c_n})^{m+1}(x) &\equiv \mathbf{c_n}\left((\mathbf{c_n})^m(x)\right) \\
&= \mathbf{c_n}\left(\mathbf{c_{n^m}}\,x\right) \ (\textit{επαγωγική υπόθεση}) \\
&\equiv (\lambda f.\lambda x.\ f^n(x))\left(\mathbf{c_{n^m}}\,x\right) \\
&\twoheadrightarrow_\beta \lambda y.(\mathbf{c_{n^m}}\,x)^n(y) \\
&= \lambda y.x^{n^m n}(y) \ (\textit{λήμμα 10.1.2}) \\
&\equiv \lambda y.x^{n^{m+1}}(y) \\
&\twoheadleftarrow_\beta (\lambda f.\lambda x.f^{n^{m+1}}(x))\,x \\
&\equiv \mathbf{c_{n^{m+1}}}\,x
\end{aligned}
$$

□

Θεώρημα 10.6

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

1. $\mathbf{succ}\ \mathbf{c_n} = \mathbf{c_{n+1}}$
2. $\mathbf{A_+}\ \mathbf{c_n}\ \mathbf{c_m} = \mathbf{c_{n+m}}$
3. $\mathbf{A_*}\ \mathbf{c_n}\ \mathbf{c_m} = \mathbf{c_{nm}}$
4. *Αν $m > 0$, τότε* $\mathbf{A_{exp}}\ \mathbf{c_n}\ \mathbf{c_m} = \mathbf{c_{n^m}}$

Απόδειξη.

1. $$\begin{aligned} \mathbf{succ}\ \mathbf{c_n} &\equiv (\lambda n.\lambda f.\lambda x.\ n\, f\, (f\, x))(\lambda f.\lambda x.\ f^n(x)) \\ &\to_\beta \lambda f.\lambda x.\ (\lambda f.\lambda x.\ f^n(x))\, f\, (f\, x) \\ &\twoheadrightarrow_\beta \lambda f.\lambda x.\ f^n(f\, x) \\ &\equiv \mathbf{c_{n+1}} \end{aligned}$$
2. $$\begin{aligned} \mathbf{A_+}\ \mathbf{c_n}\ \mathbf{c_m} &\equiv (\lambda n.\lambda m.\lambda f.\lambda x.\ n\, f\, (m\, f\, x))\ \mathbf{c_n}\ \mathbf{c_m} \\ &\twoheadrightarrow_\beta \lambda f.\lambda x.\ \mathbf{c_n}\, f\, (\mathbf{c_m}\, f\, x) \\ &= \lambda f.\lambda x.\mathbf{c_{n+m}}\, f\, x\ (\text{λήμμα } 10.1.1) \\ &\to_\eta \mathbf{c_{n+m}} \end{aligned}$$

Απόδειξη (συνέχεια).

3. $$\begin{aligned} \mathbf{A}_* \; \mathbf{c_n} \; \mathbf{c_m} &\equiv (\lambda n.\lambda m.\lambda f.\, n\,(m\,f))\; \mathbf{c_n}\; \mathbf{c_m} \\ &\twoheadrightarrow_\beta \lambda f.\, \mathbf{c_n}\,(\mathbf{c_m}\, f) \\ &\equiv \lambda f.\, (\lambda f.\lambda x.\, f^n(x))\,(\mathbf{c_m}\, f) \\ &\rightarrow_\beta \lambda f.\lambda x.\, (\mathbf{c_m}\, f)^n(x) \\ &= \lambda f.\lambda x.\, f^{mn}(x) \; (\text{λήμμα } 10.1.2) \\ &\equiv \mathbf{c_{mn}} \end{aligned}$$

Αριθμοειδή του Church

Απόδειξη (συνέχεια).

4.
$$
\begin{aligned}
\mathbf{A_{exp}}\ \mathbf{c_n}\ \mathbf{c_m} &\equiv (\lambda n.\lambda m.\ m\,n)\ \mathbf{c_n}\ \mathbf{c_m} \\
&\twoheadrightarrow_\beta \mathbf{c_m}\ \mathbf{c_n} \\
&\equiv (\lambda f.\lambda x.\ f^m(x))\ \mathbf{c_n} \\
&\rightarrow_\beta (\mathbf{c_n})^m\,(x) \\
&= \mathbf{c_{n^m}}\ x\ (\textit{λήμμα 10.1.3}) \\
&\equiv \mathbf{c_{n^m}}
\end{aligned}
$$

□

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 μεθόδους περάσματος παραμέτρων των γλωσσών προγραμματισμού.

Αναλογία 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.

Οκνηρή αποτίμηση (lazy evaluation)

- Τερματισμός της αποτίμησης μιας έκφρασης πριν προκύψει πλήρως αποτιμημένο αποτέλεσμα.
- Υποστήριξη από γλώσσες συναρτησιακού προγραμματισμού
    - Haskell και Miranda

Οκνηρή αποτίμηση

- Τερματισμός όταν το αποτέλεσμα είναι τιμή (value).
- Η ακριβής μορφή των τιμών διαφέρει από γλώσσα σε γλώσσα
  - Αριθμητικές και λογικές σταθερές, καθώς και συναρτησιακές αφαιρέσεις είναι τιμές.
- Στον λ-λογισμό μόνο οι αφαιρέσεις θεωρούνται τιμές

Αναγωγή Κανονικής Μορφής

- Η στρατηγική της αναγωγής κανονικής σειράς σχετίζεται στενά με την οκνηρή αποτίμηση.
- Η μετατροπή κάθε φορά του αριστερότερου β-redex σημαίνει ότι δεν έχει προηγηθεί μετατροπή μέσα στον όρο, πάνω στον οποίο εφαρμόζεται η αφαίρεση, που αντιστοιχεί σε αυτό το β-redex.

Αναλογία με Οκνηρή Αποτίμηση

- Κατ' αναλογία στην οκνηρή αποτίμηση μία συνάρτηση καλείται χωρίς να προηγηθεί αποτίμηση των παραμέτρων της.
- Στην γλώσσα προγραμματισμού Algol 60, η συμπεριφορά αυτή επιτυγχάνεται με τη μέθοδο του περάσματος παραμέτρων κατ' όνομα (call by name).

Παράδειγμα

- Η στρατηγική της αναγωγής κανονικής σειράς οδηγεί τελικά την αναγωγή του όρου $(\lambda x.\lambda y.y)\Omega$ στην τιμή $\lambda y.y$
- Το ίδιο αποτέλεσμα προκύπτει στη γλώσσα Algol 60, αν ορισθούν οι συναρτήσεις και στη συνέχεια αποτιμηθεί η έκφραση `f(g)` στον επόμενο κώδικα:

Παράδειγμα

Πρόγραμμα 1

```
integer procedure f (x);
  integer x;
begin
  f :=42
end;
integer procedure g;
begin
  while true do
    g := 7
end;
```

Παράδειγμα

- Το αποτέλεσμα είναι 42, γιατί κατά την αποτίμηση δε χρειάζεται να γίνει η κλήση στη συνάρτηση g, που δεν τερματίζεται
- Η αποτίμηση όμως της έκφρασης `f(g)` δεν θα τερματιζόταν σε κάποιες άλλες γλώσσες προγραμματισμού
  - Pascal, C, klp.

Πρόθυμη αποτίμηση

- Στις γλώσσες αυτές το πέρασμα παραμέτρων γίνεται κατ' αξία (by value) και η τιμή της παραμέτρου g αποτιμάται πριν κληθεί η συνάρτηση `f`.
- Η κλήση της g οδηγεί σε μη τερματισμό.
- Πρόθυμη αποτίμηση (eager evaluation) είναι η αποτίμηση κατά την οποία οι παράμετροι αποτιμώνται πριν γίνει η κλήση στη συνάρτηση.

Πρόθυμη αποτίμηση στον λ-λογισμό

- Στον λ-λογισμό, το ανάλογο της πρόθυμης αποτίμησης και του περάσματος παραμέτρων κατ΄ αξία είναι μία στρατηγική αποτίμησης που κατά την αποτίμηση του όρου $(\lambda x.\lambda y.y)\,\Omega$ θα αποτιμούσε πρώτα τον όρο Ω.

Πρόθυμη αποτίμηση στον λ-λογισμό

- Τέτοιου είδους στρατηγικές προκύπτουν υποχρεωτικά αν στον κανόνα της β-μετατροπής

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

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