Λάμβδα Λογισμός

Γιάννης Κασσιός

Δομή της Σημερινής Διάλεξης

- Ιστορικά στοιχεία για τον λ-λογισμό
- Μεταθεωρίες
- Διαισθητική εισαγωγή
- λ-όροι
- Μετατροπές
- Κανονικές μορφές
- Ιδιότητες λογισμού
- Εκφραστική δύναμη
- Σχέση με έννοιες γλωσσών προγραμματισμού
- Επισκόπηση

Δομή της Σημερινής Διάλεξης

- Ιστορικά στοιχεία για τον λ-λογισμό
- Μεταθεωρίες
- Διαισθητική εισαγωγή
- λ-όροι
- Μετατροπές
- Κανονικές μορφές
- Ιδιότητες λογισμού
- Εκφραστική δύναμη
- Σχέση με έννοιες γλωσσών προγραμματισμού
- Επισκόπηση

Δομή της Σημερινής Διάλεξης

- Ιστορικά στοιχεία για τον λ-λογισμό
- Μεταθεωρίες
- Διαισθητική εισαγωγή
- λ-όροι
- Μετατροπές
- Κανονικές μορφές
- Ιδιότητες λογισμού
- Εκφραστική δύναμη
- Σχέση με έννοιες γλωσσών προγραμματισμού
- Επισκόπηση

Δομή της Σημερινής Διάλεξης

- Ιστορικά στοιχεία για τον λ-λογισμό
- Μεταθεωρίες
- Διαισθητική εισαγωγή
- λ-όροι
- Μετατροπές
- Κανονικές μορφές
- Ιδιότητες λογισμού
- Εκφραστική δύναμη
- Σχέση με έννοιες γλωσσών προγραμματισμού
- Επισκόπηση

Δομή της Σημερινής Διάλεξης

- Ιστορικά στοιχεία για τον λ-λογισμό
- Μεταθεωρίες
- Διαισθητική εισαγωγή
- λ-όροι
- Μετατροπές
- Κανονικές μορφές
- Ιδιότητες λογισμού
- Εκφραστική δύναμη
- Σχέση με έννοιες γλωσσών προγραμματισμού
- Επισκόπηση

Δομή της Σημερινής Διάλεξης

- Ιστορικά στοιχεία για τον λ-λογισμό
- Μεταθεωρίες
- Διαισθητική εισαγωγή
- λ-όροι
- Μετατροπές
- Κανονικές μορφές
- Ιδιότητες λογισμού
- Εκφραστική δύναμη
- Σχέση με έννοιες γλωσσών προγραμματισμού
- Επισκόπηση

Δομή της Σημερινής Διάλεξης

- Ιστορικά στοιχεία για τον λ-λογισμό
- Μεταθεωρίες
- Διαισθητική εισαγωγή
- λ-όροι
- Μετατροπές
- Κανονικές μορφές
- Ιδιότητες λογισμού
- Εκφραστική δύναμη
- Σχέση με έννοιες γλωσσών προγραμματισμού
- Επισκόπηση

Δομή της Σημερινής Διάλεξης

- Ιστορικά στοιχεία για τον λ-λογισμό
- Μεταθεωρίες
- Διαισθητική εισαγωγή
- λ-όροι
- Μετατροπές
- Κανονικές μορφές
- Ιδιότητες λογισμού
- Εκφραστική δύναμη
- Σχέση με έννοιες γλωσσών προγραμματισμού
- Επισκόπηση

Δομή της Σημερινής Διάλεξης

- Ιστορικά στοιχεία για τον λ-λογισμό
- Μεταθεωρίες
- Διαισθητική εισαγωγή
- λ-όροι
- Μετατροπές
- Κανονικές μορφές
- Ιδιότητες λογισμού
- Εκφραστική δύναμη
- Σχέση με έννοιες γλωσσών προγραμματισμού
- Επισκόπηση

Δομή της Σημερινής Διάλεξης

- Ιστορικά στοιχεία για τον λ-λογισμό
- Μεταθεωρίες
- Διαισθητική εισαγωγή
- λ-όροι
- Μετατροπές
- Κανονικές μορφές
- Ιδιότητες λογισμού
- Εκφραστική δύναμη
- Σχέση με έννοιες γλωσσών προγραμματισμού
- Επισκόπηση

Ιστορικά Στοιχεία για τον λ-λογισμό

- Αναπτύχθηκε το 1933 από τον Alonso Church
- Μέρος μίας ευρύτερης θεωρίας θεμελείωσης των μαθηματικών
  - η ευρεία θεωρία αποδείχτηκε ασυνεπής
- Πλήρες, αλλά εξαιρετικά απλό, υπολογιστικό μοντέλο
  - στηρίζεται μόνο σε συναρτήσεις
- Οδήγησε στη δημιουργία του Συναρτησιακού Προγραμματισμού
  - Lisp ($\longrightarrow$ Scheme), ML ($\longrightarrow$ Haskell) κ.α.
- Θεμελειώδης στην έρευνα για Γλώσσες Προγραμματισμού

Ιστορικά Στοιχεία για τον λ-λογισμό

- Αναπτύχθηκε το 1933 από τον Alonso Church
- Μέρος μίας ευρύτερης θεωρίας θεμελείωσης των μαθηματικών
  - η ευρεία θεωρία αποδείχτηκε ασυνεπής
- Πλήρες, αλλά εξαιρετικά απλό, υπολογιστικό μοντέλο
  - στηρίζεται μόνο σε συναρτήσεις
- Οδήγησε στη δημιουργία του Συναρτησιακού Προγραμματισμού
  - Lisp ($\longrightarrow$ Scheme), ML ($\longrightarrow$ Haskell) κ.α.
- Θεμελειώδης στην έρευνα για Γλώσσες Προγραμματισμού

Ιστορικά Στοιχεία για τον λ-λογισμό

- Αναπτύχθηκε το 1933 από τον Alonso Church
- Μέρος μίας ευρύτερης θεωρίας θεμελείωσης των μαθηματικών
  - η ευρεία θεωρία αποδείχτηκε ασυνεπής
- Πλήρες, αλλά εξαιρετικά απλό, υπολογιστικό μοντέλο
  - στηρίζεται μόνο σε συναρτήσεις
- Οδήγησε στη δημιουργία του Συναρτησιακού Προγραμματισμού
  - Lisp ($\longrightarrow$ Scheme), ML ($\longrightarrow$ Haskell) κ.α.
- Θεμελειώδης στην έρευνα για Γλώσσες Προγραμματισμού

Ιστορικά Στοιχεία για τον λ-λογισμό

- Αναπτύχθηκε το 1933 από τον Alonso Church
- Μέρος μίας ευρύτερης θεωρίας θεμελείωσης των μαθηματικών
  - η ευρεία θεωρία αποδείχτηκε ασυνεπής
- Πλήρες, αλλά εξαιρετικά απλό, υπολογιστικό μοντέλο
  - στηρίζεται μόνο σε συναρτήσεις
- Οδήγησε στη δημιουργία του Συναρτησιακού Προγραμματισμού
  - Lisp ($\longrightarrow$ Scheme), ML ($\longrightarrow$ Haskell) κ.α.
- Θεμελειώδης στην έρευνα για Γλώσσες Προγραμματισμού

Ιστορικά Στοιχεία για τον λ-λογισμό

- Αναπτύχθηκε το 1933 από τον Alonso Church
- Μέρος μίας ευρύτερης θεωρίας θεμελείωσης των μαθηματικών
  - η ευρεία θεωρία αποδείχτηκε ασυνεπής
- Πλήρες, αλλά εξαιρετικά απλό, υπολογιστικό μοντέλο
  - στηρίζεται μόνο σε συναρτήσεις
- Οδήγησε στη δημιουργία του Συναρτησιακού Προγραμματισμού
  - Lisp ($\longrightarrow$ Scheme), ML ($\longrightarrow$ Haskell) κ.α.
- Θεμελειώδης στην έρευνα για Γλώσσες Προγραμματισμού

Μεταθεωρίες

- Μεταθεωρία (metatheory) = μία θεωρία που περιγράφει μία άλλη θεωρία
- Μεταμεταβλητή (metavariable) = μεταβλητή της μεταθεωρίας
  - παράδειγμα: "για κάθε μεταβλητή x της θεωρίας, ισχύει..."
  - x μεταμεταβλητή που παίρνει ως τιμές μεταβλητές της θεωρίας
- Θα περιγράψουμε το λ-λογισμό με μια μεταθεωρία
  - $x, y, z, \ldots, X, Y, Z, \ldots$ μετα-μεταβλητές
  - $\mathtt{x}, \mathtt{y}, \mathtt{z}, \ldots, \mathtt{X}, \mathtt{Y}, \mathtt{Z}, \ldots$ μεταβλητές του λ-λογισμού

Μεταθεωρίες

- Μεταθεωρία (metatheory) = μία θεωρία που περιγράφει μία άλλη θεωρία
- Μεταμεταβλητή (metavariable) = μεταβλητή της μεταθεωρίας
  - παράδειγμα: "για κάθε μεταβλητή x της θεωρίας, ισχύει..."
  - x μεταμεταβλητή που παίρνει ως τιμές μεταβλητές της θεωρίας
- Θα περιγράψουμε το λ-λογισμό με μια μεταθεωρία
  - $x, y, z, \ldots, X, Y, Z, \ldots$ μετα-μεταβλητές
  - $\mathtt{x}, \mathtt{y}, \mathtt{z} \ldots, \mathtt{X}, \mathtt{Y}, \mathtt{Z}, \ldots$ μεταβλητές του λ-λογισμού

Μεταθεωρίες

- Μεταθεωρία (metatheory) = μία θεωρία που περιγράφει μία άλλη θεωρία
- Μεταμεταβλητή (metavariable) = μεταβλητή της μεταθεωρίας
  - παράδειγμα: "για κάθε μεταβλητή x της θεωρίας, ισχύει..."
  - x μεταμεταβλητή που παίρνει ως τιμές μεταβλητές της θεωρίας
- Θα περιγράψουμε το λ-λογισμό με μια μεταθεωρία
  - $x, y, z, \ldots, X, Y, Z, \ldots$ μετα-μεταβλητές
  - `x,y,z...,X,Y,Z,...` μεταβλητές του λ-λογισμού

- Οι εκφράσεις του λ-λογισμού είναι όλες συναρτήσεις
- Δύο κύριες λειτουργίες συναρτήσεων υποστηρίζονται:
  - εφαρμογή (application) συνάρτησης
  - αφαίρεση (abstraction) δηλ. κατασκευή συνάρτησης
- Συμβολισμός: $E[\cdot]$ μία έκφραση λ-λογισμού με "τρύπες"
- Εφαρμογή: $F\ X$
- Αφαίρεση: $\lambda x \cdot E[x]$
  - κατασκευάζει ανώνυμη συνάρτηση που όταν δέχεται τιμή v επιστρέφει $E[v]$
- Παράδειγμα (λ-λογισμός + αριθμητική)
  `(λx. x*x - 3*x +2)8 = 8*8-3*8+2 = 42`

Διαισθητική Εισαγωγή

- Οι εκφράσεις του λ-λογισμού είναι όλες <u>συναρτήσεις</u>
- Δύο κύριες λειτουργίες συναρτήσεων υποστηρίζονται:
  - <u>εφαρμογή (application)</u> συνάρτησης
  - <u>αφαίρεση (abstraction)</u> δηλ. κατασκευή συνάρτησης
- Συμβολισμός: $E[\cdot]$ μία έκφραση λ-λογισμού με "τρύπες"
- Εφαρμογή: $F\ X$
- Αφαίρεση: $\lambda x \cdot E[x]$
  - κατασκευάζει <u>ανώνυμη συνάρτηση</u> που όταν δέχεται τιμή v επιστρέφει $E[v]$
- Παράδειγμα (λ-λογισμός + αριθμητική)
  `(λx. x*x - 3*x +2)8 = 8*8-3*8+2 = 42`

Διαισθητική Εισαγωγή

- Οι εκφράσεις του λ-λογισμού είναι όλες συναρτήσεις
- Δύο κύριες λειτουργίες συναρτήσεων υποστηρίζονται:
  - εφαρμογή (application) συνάρτησης
  - αφαίρεση (abstraction) δηλ. κατασκευή συνάρτησης
- Συμβολισμός: $E[\cdot]$ μία έκφραση λ-λογισμού με "τρύπες"
- Εφαρμογή: $F\ X$
- Αφαίρεση: $\lambda x \cdot E[x]$
  - κατασκευάζει ανώνυμη συνάρτηση που όταν δέχεται τιμή v επιστρέφει $E[v]$
- Παράδειγμα (λ-λογισμός + αριθμητική)
  `(λx. x*x - 3*x +2)8 = 8*8-3*8+2 = 42`

Διαισθητική Εισαγωγή

- Οι εκφράσεις του λ-λογισμού είναι όλες συναρτήσεις
- Δύο κύριες λειτουργίες συναρτήσεων υποστηρίζονται:
  - εφαρμογή (application) συνάρτησης
  - αφαίρεση (abstraction) δηλ. κατασκευή συνάρτησης
- Συμβολισμός: $E[\cdot]$ μία έκφραση λ-λογισμού με "τρύπες"
- Εφαρμογή: $F\ X$
- Αφαίρεση: $\lambda x \cdot E[x]$
  - κατασκευάζει ανώνυμη συνάρτηση που όταν δέχεται τιμή v επιστρέφει $E[v]$
- Παράδειγμα (λ-λογισμός + αριθμητική)

```
(λx. x*x - 3*x +2)8 = 8*8-3*8+2 = 42
```

Διαισθητική Εισαγωγή

- Οι εκφράσεις του λ-λογισμού είναι όλες <u>συναρτήσεις</u>
- Δύο κύριες λειτουργίες συναρτήσεων υποστηρίζονται:
  - <u>εφαρμογή (application)</u> συνάρτησης
  - <u>αφαίρεση (abstraction)</u> δηλ. κατασκευή συνάρτησης
- Συμβολισμός: $E[\cdot]$ μία έκφραση λ-λογισμού με "τρύπες"
- Εφαρμογή: $F\ X$
- Αφαίρεση: $\lambda x \cdot E[x]$
  - κατασκευάζει <u>ανώνυμη συνάρτηση</u> που όταν δέχεται τιμή v επιστρέφει $E[v]$
- Παράδειγμα (λ-λογισμός + αριθμητική)
  $(\lambda x.\ x*x\ -\ 3*x\ +2)8\ =\ 8*8-3*8+2\ =\ 42$

Διαισθητική Εισαγωγή

- Οι εκφράσεις του λ-λογισμού είναι όλες συναρτήσεις
- Δύο κύριες λειτουργίες συναρτήσεων υποστηρίζονται:
  - εφαρμογή (application) συνάρτησης
  - αφαίρεση (abstraction) δηλ. κατασκευή συνάρτησης
- Συμβολισμός: $E[\cdot]$ μία έκφραση λ-λογισμού με "τρύπες"
- Εφαρμογή: $F\ X$
- Αφαίρεση: $\lambda x \cdot E[x]$
  - κατασκευάζει ανώνυμη συνάρτηση που όταν δέχεται τιμή v επιστρέφει $E[v]$
- Παράδειγμα (λ-λογισμός + αριθμητική)

  `(λx. x*x - 3*x +2)8 = 8*8-3*8+2 = 42`

- Η αφαίρεση $\lambda x.\ E[x]$, δεσμεύει (binds) τη μεταβλητή x στην $E[x]$
  - εμφανίσεις x στην $E[x]$: δεσμευμένες (bound)
  - μη δεσμευμένη εμφάνιση μεταβλητής: ελεύθερη (free)
- Παράδειγμα:
  `λx. x*x + y`
  `(λx. x*x + y) (4*x + 2)`
  - δεσμευμένες εμφανίσεις
  - ελεύθερες εμφανίσεις
- Η δεσμευμένη x και η ελεύθερη x συμβολίζουν διαφορετικό αντικείμενο
- Δύο δεσμευμένες εμφανίσεις της ίδιας μεταβλητής συμβολίζουν διαφορετικό αντικείμενο αν δεσμεύονται από διαφορετικό λ:
  `(λx. x+1)7 + (λx. x*2)17`

- Η αφαίρεση $\lambda x.\ E[x]$, δεσμεύει (binds) τη μεταβλητή x στην $E[x]$
  - εμφανίσεις x στην $E[x]$: δεσμευμένες (bound)
  - μη δεσμευμένη εμφάνιση μεταβλητής: ελεύθερη (free)
- Παράδειγμα:
  `λx. x*x + y`
  `(λx. x*x + y) (4*x + 2)`
  - δεσμευμένες εμφανίσεις
  - ελεύθερες εμφανίσεις
- Η δεσμευμένη x και η ελεύθερη x συμβολίζουν διαφορετικό αντικείμενο
- Δύο δεσμευμένες εμφανίσεις της ίδιας μεταβλητής συμβολίζουν διαφορετικό αντικείμενο αν δεσμεύονται από διαφορετικό λ:
  `(λx. x+1)7 + (λx. x*2)17`

Διαισθητική Εισαγωγή
Δέσμευση Μεταβλητών - Ι

- Η αφαίρεση $\lambda x.\ E[x]$, <u>δεσμεύει (binds)</u> τη μεταβλητή x στην $E[x]$
  - εμφανίσεις x στην $E[x]$: <u>δεσμευμένες (bound)</u>
  - μη δεσμευμένη εμφάνιση μεταβλητής: <u>ελεύθερη (free)</u>
- Παράδειγμα:
  `λx. x*x + y`
  `(λx. x*x + y) (4*x + 2)`
  - δεσμευμένες εμφανίσεις
  - ελεύθερες εμφανίσεις
- Η δεσμευμένη x και η ελεύθερη x συμβολίζουν διαφορετικό αντικείμενο
- Δύο δεσμευμένες εμφανίσεις της ίδιας μεταβλητής συμβολίζουν διαφορετικό αντικείμενο αν δεσμεύονται από διαφορετικό λ:
  `(λx. x+1)7 + (λx. x*2)17`

- Η αφαίρεση $\lambda x.\ E[x]$, δεσμεύει (binds) τη μεταβλητή x στην $E[x]$
    - εμφανίσεις x στην $E[x]$: δεσμευμένες (bound)
    - μη δεσμευμένη εμφάνιση μεταβλητής: ελεύθερη (free)
- Παράδειγμα:

```
λx. x*x + y
(λx. x*x + y) (4*x + 2)
```

    - δεσμευμένες εμφανίσεις
    - ελεύθερες εμφανίσεις
- Η δεσμευμένη x και η ελεύθερη x συμβολίζουν διαφορετικό αντικείμενο
- Δύο δεσμευμένες εμφανίσεις της ίδιας μεταβλητής συμβολίζουν διαφορετικό αντικείμενο αν δεσμεύονται από διαφορετικό λ:

```
(λx. x+1)7 + (λx. x*2)17
```

- Η αφαίρεση $\lambda x.\ E[x]$, <u>δεσμεύει (binds)</u> τη μεταβλητή x στην $E[x]$
  - εμφανίσεις x στην $E[x]$: <u>δεσμευμένες (bound)</u>
  - μη δεσμευμένη εμφάνιση μεταβλητής: <u>ελεύθερη (free)</u>
- Παράδειγμα:

  `λx. x*x + y`

  `(λx. x*x + y) (4*x + 2)`
  - δεσμευμένες εμφανίσεις
  - ελεύθερες εμφανίσεις
- Η δεσμευμένη x και η ελεύθερη x συμβολίζουν διαφορετικό αντικείμενο
- Δύο δεσμευμένες εμφανίσεις της ίδιας μεταβλητής συμβολίζουν διαφορετικό αντικείμενο αν δεσμεύονται από διαφορετικό λ:

  `(λx. x+1)7 + (λx. x*2)17`

- Η αφαίρεση $\lambda x.\ E[x]$, δεσμεύει (binds) τη μεταβλητή x στην $E[x]$
  - εμφανίσεις x στην $E[x]$: δεσμευμένες (bound)
  - μη δεσμευμένη εμφάνιση μεταβλητής: ελεύθερη (free)
- Παράδειγμα:

  λ`x. x*x + y`

  `(`λ`x. x*x + y) (4*x + 2)`
  - δεσμευμένες εμφανίσεις
  - ελεύθερες εμφανίσεις
- Η δεσμευμένη x και η ελεύθερη x συμβολίζουν διαφορετικό αντικείμενο
- Δύο δεσμευμένες εμφανίσεις της ίδιας μεταβλητής συμβολίζουν διαφορετικό αντικείμενο αν δεσμεύονται από διαφορετικό λ:

  `(`λ`x.  x+1)7 + (`λ`x.  x*2)17`

- Η αφαίρεση $\lambda x.\ E[x]$, δεσμεύει (binds) τη μεταβλητή x στην $E[x]$
  - εμφανίσεις x στην $E[x]$: δεσμευμένες (bound)
  - μη δεσμευμένη εμφάνιση μεταβλητής: ελεύθερη (free)
- Παράδειγμα:

  `λx. x*x + y`

  `(λx. x*x + y) (4*x + 2)`

  - δεσμευμένες εμφανίσεις
  - ελεύθερες εμφανίσεις
- Η δεσμευμένη x και η ελεύθερη x συμβολίζουν διαφορετικό αντικείμενο
- Δύο δεσμευμένες εμφανίσεις της ίδιας μεταβλητής συμβολίζουν διαφορετικό αντικείμενο αν δεσμεύονται από διαφορετικό λ:

  `(λx.  x+1)7 + (λx.  x*2)17`

- Μία έκφραση εξαρτάται μόνο από τις ελεύθερες μεταβλητές της:
  - `λx. x*x + y` εξαρτάται μόνο από το `y`
    - για `y=0` συμβολίζει την $f(x) = x^2$
    - για `y=2` συμβολίζει την $f(x) = x^2 + 2$
  - `(λx. x*x + y)(4*x + 2) = 16*x*x + 16*x + 4 + y`
- Αντικατάσταση μεταβλητών:
  - δεν αντικαθιστούμε δεσμευμένες μεταβλητές, π.χ.(`x=1`) `λx.x+1` $\neq$ `λ1.1+1`
  - η αντικατάσταση δε δεσμεύει μεταβλητές, π.χ. (`y=x+1`) `λx.x+y` $\neq$ `λx.x+x+1`
    - π.χ. `x=0`, `y=1` κάνουν:
      `λx.x+1` και

- Μία έκφραση εξαρτάται μόνο από τις ελεύθερες μεταβλητές της:
  - `λx. x*x + y` εξαρτάται μόνο από το `y`
    - για `y=0` συμβολίζει την $f(x) = x^2$
    - για `y=2` συμβολίζει την $f(x) = x^2 + 2$
  - `(λx. x*x + y)(4*x + 2) = 16*x*x + 16*x + 4 + y`
- Αντικατάσταση μεταβλητών:
  - δεν αντικαθιστούμε δεσμευμένες μεταβλητές, π.χ.(`x=1`) `λx.x+1` $\neq$ `λ1.1+1`
  - η αντικατάσταση δε δεσμεύει μεταβλητές, π.χ. (`y=x+1`) `λx.x+y` $\neq$ `λx.x+x+1`
    - π.χ. `x=0`, `y=1` κάνουν:
      `λx.x+1` και

- Μία έκφραση εξαρτάται μόνο από τις ελεύθερες μεταβλητές της:
  - `λx.  x*x + y` εξαρτάται μόνο από το `y`
    - για `y=0` συμβολίζει την $f(x) = x^2$
    - για `y=2` συμβολίζει την $f(x) = x^2 + 2$
  - `(λx.  x*x + y)(4*x + 2) =`
    `16*x*x + 16*x + 4 + y`
- Αντικατάσταση μεταβλητών:
  - δεν αντικαθιστούμε δεσμευμένες μεταβλητές, π.χ.(`x=1`)
    `λx.x+1` $\neq$ `λ1.1+1`
  - η αντικατάσταση δε δεσμεύει μεταβλητές, π.χ. (`y=x+1`)
    `λx.x+y` $\neq$ `λx.x+x+1`
    - π.χ. `x=0`, `y=1` κάνουν:
      `λx.x+1` και

- Μία έκφραση εξαρτάται μόνο από τις ελεύθερες μεταβλητές της:
  - `λx.  x*x + y` εξαρτάται μόνο από το `y`
    - για `y=0` συμβολίζει την $f(x) = x^2$
    - για `y=2` συμβολίζει την $f(x) = x^2 + 2$
  - `(λx.  x*x + y)(4*x + 2) =`
    `16*x*x + 16*x + 4 + y`
- Αντικατάσταση μεταβλητών:
  - δεν αντικαθιστούμε δεσμευμένες μεταβλητές, π.χ.(`x=1`)
    `λx.x+1` $\neq$ `λ1.1+1`
  - η αντικατάσταση δε δεσμεύει μεταβλητές, π.χ. (`y=x+1`)
    `λx.x+y` $\neq$ `λx.x+x+1`
    - π.χ. `x=0, y=1` κάνουν:
      `λx.x+1` και `λx.2*x+1`

- Μία έκφραση εξαρτάται μόνο από τις ελεύθερες μεταβλητές της:
  - λx. x*x + y εξαρτάται μόνο από το y
    - για y=0 συμβολίζει την $f(x) = x^2$
    - για y=2 συμβολίζει την $f(x) = x^2 + 2$
  - (λx. x*x + y)(4*x + 2) = 16*x*x + 16*x + 4 + y
- Αντικατάσταση μεταβλητών:
  - δεν αντικαθιστούμε δεσμευμένες μεταβλητές, π.χ.(x=1)
    λx.x+1 ≠ λ1.1+1
  - η αντικατάσταση δε δεσμεύει μεταβλητές, π.χ. (y=x+1)
    λx.x+y ≠ λx.x+x+1
    - π.χ. x=0, y=1 κάνουν:
      λx.x+1 και λx.2*x+1

- Μία έκφραση εξαρτάται μόνο από τις ελεύθερες μεταβλητές της:
  - λx. x*x + y εξαρτάται μόνο από το y
    - για y=0 συμβολίζει την $f(x) = x^2$
    - για y=2 συμβολίζει την $f(x) = x^2 + 2$
  - (λx. x*x + y)(4*x + 2) =
    16*x*x + 16*x + 4 + y
- Αντικατάσταση μεταβλητών:
  - δεν αντικαθιστούμε δεσμευμένες μεταβλητές, π.χ.(x=1)
    λx.x+1 $\neq$ λ1.1+1
  - η αντικατάσταση δε δεσμεύει μεταβλητές, π.χ. (y=x+1)
    λx.x+y $\neq$ λx.x+x+1
    - π.χ. x=0, y=1 κάνουν:
      λx.x+1 και λx.2*x+1

- Μία έκφραση εξαρτάται μόνο από τις ελεύθερες μεταβλητές της:
  - `λx. x*x + y` εξαρτάται μόνο από το `y`
    - για `y=0` συμβολίζει την $f(x) = x^2$
    - για `y=2` συμβολίζει την $f(x) = x^2 + 2$
  - `(λx. x*x + y)(4*x + 2) =`
    `16*x*x + 16*x + 4 + y`
- Αντικατάσταση μεταβλητών:
  - δεν αντικαθιστούμε δεσμευμένες μεταβλητές, π.χ.(`x=1`)
    `λx.x+1` $\neq$ `λ1.1+1`
  - η αντικατάσταση δε δεσμεύει μεταβλητές, π.χ. (`y=x+1`)
    `λx.x+y` $\neq$ `λx.x+x+1`
    - π.χ. `x=0,y=1` κάνουν:
      `λx.x+1` και `λx.2*x+1`

- Έστω V αριθμήσιμο σύνολο μεταβλητών
  - θα χρησιμοποιούμε `a, b, c, ...` για τα στοιχεία του V
- Το σύνολο Λ των λ-όρων (λ-terms) είναι το μικρότερο που ικανοποιεί:
  - $V \subset \Lambda$
  - $M \in \Lambda \ \wedge \ N \in \Lambda \ \Rightarrow \ (M\,N) \in \Lambda$
  - $x \in V \ \wedge \ E \in \Lambda \ \Rightarrow \ (\lambda x\,.\,E) \in \Lambda$
- Σύμβαση:
  - πεζές μεταμεταβλητές $a, b, c, \ldots$ για τα στοιχεία του V
  - κεφαλαίες μεταμεταβλητές $A, B, C, \ldots$ για τα στοιχεία του Λ

- Έστω V αριθμήσιμο σύνολο μεταβλητών
  - θα χρησιμοποιούμε a, b, c, . . . για τα στοιχεία του V
- Το σύνολο Λ των λ-όρων (λ-terms) είναι το μικρότερο που ικανοποιεί:
  - $V \subset \Lambda$
  - $M \in \Lambda \ \wedge \ N \in \Lambda \ \Rightarrow \ (M\,N) \in \Lambda$
  - $x \in V \ \wedge \ E \in \Lambda \ \Rightarrow \ (\lambda x.E) \in \Lambda$
- Σύμβαση:
  - πεζές μεταμεταβλητές $a, b, c, \ldots$ για τα στοιχεία του V
  - κεφαλαίες μεταμεταβλητές $A, B, C, \ldots$ για τα στοιχεία του Λ

- Έστω V αριθμήσιμο σύνολο μεταβλητών
  - θα χρησιμοποιούμε $\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots$ για τα στοιχεία του V
- Το σύνολο Λ των <u>λ-όρων (λ-terms</u>) είναι το μικρότερο που ικανοποιεί:
  - $V \subset \Lambda$
  - $M \in \Lambda \ \wedge \ N \in \Lambda \ \Rightarrow \ (M\,N) \in \Lambda$
  - $x \in V \ \wedge \ E \in \Lambda \ \Rightarrow \ (\lambda x\,.\,E) \in \Lambda$
- Σύμβαση:
  - πεζές μεταμεταβλητές $a, b, c, \ldots$ για τα στοιχεία του V
  - κεφαλαίες μεταμεταβλητές $A, B, C, \ldots$ για τα στοιχεία του Λ

- Έστω V αριθμήσιμο σύνολο μεταβλητών
  - θα χρησιμοποιούμε $\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots$ για τα στοιχεία του V
- Το σύνολο Λ των λ-όρων (λ-terms) είναι το μικρότερο που ικανοποιεί:
  - $V \subset \Lambda$
  - $M \in \Lambda \;\wedge\; N \in \Lambda \;\Rightarrow\; (M\,N) \in \Lambda$
  - $x \in V \;\wedge\; E \in \Lambda \;\Rightarrow\; (\lambda x\,.\,E) \in \Lambda$
- Σύμβαση:
  - πεζές μεταμεταβλητές $a, b, c, \ldots$ για τα στοιχεία του V
  - κεφαλαίες μεταμεταβλητές $A, B, C, \ldots$ για τα στοιχεία του Λ

- Έστω V αριθμήσιμο σύνολο μεταβλητών
  - θα χρησιμοποιούμε $\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots$ για τα στοιχεία του V
- Το σύνολο Λ των λ-όρων (λ-terms) είναι το μικρότερο που ικανοποιεί:
  - $V \subset \Lambda$
  - $M \in \Lambda \;\wedge\; N \in \Lambda \;\Rightarrow\; (M\,N) \in \Lambda$
  - $x \in V \;\wedge\; E \in \Lambda \;\Rightarrow\; (\lambda x\,.\,E) \in \Lambda$
- Σύμβαση:
  - πεζές μεταμεταβλητές $a, b, c, \ldots$ για τα στοιχεία του V
  - κεφαλαίες μεταμεταβλητές $A, B, C, \ldots$ για τα στοιχεία του Λ

- Έστω V αριθμήσιμο σύνολο μεταβλητών
  - θα χρησιμοποιούμε $\mathtt{a, b, c, \ldots}$ για τα στοιχεία του V
- Το σύνολο Λ των<u>λ-όρων (λ-terms</u>) είναι το μικρότερο που ικανοποιεί:
  - $V \subset \Lambda$
  - $M \in \Lambda \;\wedge\; N \in \Lambda \;\Rightarrow\; \mathtt{(}M\,N\mathtt{)} \in \Lambda$
  - $x \in V \;\wedge\; E \in \Lambda \;\Rightarrow\; \mathtt{(}\lambda x\,\mathtt{.}\,E\mathtt{)} \in \Lambda$
- Σύμβαση:
  - πεζές μεταμεταβλητές $a, b, c, \ldots$ για τα στοιχεία του V
  - κεφαλαίες μεταμεταβλητές $A, B, C, \ldots$ για τα στοιχεία του Λ

Παραδείγματα λ-όρων:

```
(x y)                      (λx.x)
(λx.(λy. (x y)))           (((λx.x)y)(λx.z))
((λx.(λy.z)) (λx.x))       (λx.((λy.y) (λz.x)))
```

Συντομογραφίες:

- οι εξωτερικές παρενθέσεις δε γράφονται
- η εφαρμογή είναι αριστερά προσεταιριστική: $FXY = (FX)Y$
- η αφαίρεση εκτείνεται όσο πιο αριστερά γίνεται

Παραδείγματα λ-όρων:

```
(x y)                       (λx.x)
(λx.(λy.  (x y)))           (((λx.x)y)(λx.z))
((λx.(λy.z)) (λx.x))        (λx.((λy.y) (λz.x)))
```

Συντομογραφίες:

- οι εξωτερικές παρενθέσεις δε γράφονται
- η εφαρμογή είναι αριστερά προσεταιριστική: $FXY = (FX)Y$
- η αφαίρεση εκτείνεται όσο πιο αριστερά γίνεται

Παραδείγματα λ-όρων:

```
(x y)                          (λx.x)
(λx.(λy.  (x y)))              (((λx.x)y)(λx.z))
((λx.(λy.z)) (λx.x))           (λx.((λy.y) (λz.x)))
```

Συντομογραφίες:

- οι εξωτερικές παρενθέσεις δε γράφονται
- η εφαρμογή είναι αριστερά προσεταιριστική: $FXY = (FX)Y$
- η αφαίρεση εκτείνεται όσο πιο αριστερά γίνεται

Παραδείγματα λ-όρων:

```
(x y)                        (λx.x)
(λx.(λy.  (x y)))            (((λx.x)y)(λx.z))
((λx.(λy.z)) (λx.x))         (λx.((λy.y) (λz.x)))
```

Συντομογραφίες:

- οι εξωτερικές παρενθέσεις δε γράφονται
- η εφαρμογή είναι αριστερά προσεταιριστική: $FXY = (FX)Y$
- η αφαίρεση εκτείνεται όσο πιο αριστερά γίνεται

Παραδείγματα λ-όρων:

`x y`	`(`λ`x.x)`
`(`λ`x.(`λ`y. (x y)))`	`(((`λ`x.x)y)(`λ`x.z))`
`((`λ`x.(`λ`y.z)) (`λ`x.x))`	`(`λ`x.((`λ`y.y) (`λ`z.x)))`

Συντομογραφίες:

- οι εξωτερικές παρενθέσεις δε γράφονται
- η εφαρμογή είναι αριστερά προσεταιριστική: $FXY = (FX)Y$
- η αφαίρεση εκτείνεται όσο πιο αριστερά γίνεται

Παραδείγματα λ-όρων:

```
x y                    λx.x
(λx.(λy.  (x y)))      (((λx.x)y)(λx.z))
((λx.(λy.z)) (λx.x))   (λx.((λy.y) (λz.x)))
```

Συντομογραφίες:

- οι εξωτερικές παρενθέσεις δε γράφονται
- η εφαρμογή είναι αριστερά προσεταιριστική: $FXY = (FX)Y$
- η αφαίρεση εκτείνεται όσο πιο αριστερά γίνεται

Παραδείγματα λ-όρων:

`x y`	λ`x.x`
λ`x.`λ`y.x y`	`(((`λ`x.x)y)(`λ`x.z))`
`((`λ`x.(`λ`y.z)) (`λ`x.x))`	`(`λ`x.((`λ`y.y) (`λ`z.x)))`

Συντομογραφίες:

- οι εξωτερικές παρενθέσεις δε γράφονται
- η εφαρμογή είναι αριστερά προσεταιριστική: $FXY = (FX)Y$
- η αφαίρεση εκτείνεται όσο πιο αριστερά γίνεται

Παραδείγματα λ-όρων:

`x y`	`λx.x`
`λx.λy.x y`	`(λx.x)y(λx.z)`
`((λx.(λy.z)) (λx.x))`	`(λx.((λy.y) (λz.x)))`

Συντομογραφίες:

- οι εξωτερικές παρενθέσεις δε γράφονται
- η εφαρμογή είναι αριστερά προσεταιριστική: $FXY = (FX)Y$
- η αφαίρεση εκτείνεται όσο πιο αριστερά γίνεται

Παραδείγματα λ-όρων:

`x y`	λ`x.x`
λ`x.`λ`y.x y`	`(`λ`x.x)y(`λ`x.z)`
`(`λ`x.`λ`y.z)(`λ`x.x)`	`(`λ`x.((`λ`y.y) (`λ`z.x)))`

Συντομογραφίες:

- οι εξωτερικές παρενθέσεις δε γράφονται
- η εφαρμογή είναι αριστερά προσεταιριστική: $FXY = (FX)Y$
- η αφαίρεση εκτείνεται όσο πιο αριστερά γίνεται

Παραδείγματα λ-όρων:

`x y`	`λx.x`
`λx.λy.x y`	`(λx.x)y(λx.z)`
`(λx.λy.z)(λx.x)`	`λx.(λy.y)(λz.x)`

Συντομογραφίες:

- οι εξωτερικές παρενθέσεις δε γράφονται
- η εφαρμογή είναι αριστερά προσεταιριστική: $FXY = (FX)Y$
- η αφαίρεση εκτείνεται όσο πιο αριστερά γίνεται

- $M \equiv N$ ταυτόσημοι (identical) όροι
- $\lambda x.M$
  - x: δεσμεύουσα (binding) μεταβλητή
  - M: εμβέλεια (scope) του x (εκτός από αφαιρέσεις που δεσμεύουν το x)
- Ελευθερες μεταβλητές:

$$
\begin{aligned}
FV(x) &= \{x\} \\
FV((MN)) &= FV(M) \cup FV(N) \\
FV((\lambda x.E)) &= FV(E) - \{x\}
\end{aligned}
$$

- Κλειστός όρος (closed term) M ανν $FV(M) = \emptyset$
  - σύνολο κλειστών όρων: Λ^0

- $M \equiv N$ ταυτόσημοι (identical) όροι
- $\lambda x.M$
  - x: δεσμεύουσα (binding) μεταβλητή
  - M: εμβέλεια (scope) του x (εκτός από αφαιρέσεις που δεσμεύουν το x)
- Ελευθερες μεταβλητές:

$$
\begin{aligned}
FV(x) &= \{x\} \\
FV((MN)) &= FV(M) \cup FV(N) \\
FV((\lambda x.E)) &= FV(E) - \{x\}
\end{aligned}
$$

- Κλειστός όρος (closed term) M ανν $FV(M) = \emptyset$
  - σύνολο κλειστών όρων: Λ^0

- $M \equiv N$ ταυτόσημοι (identical) όροι
- $\lambda x.M$
  - x: δεσμεύουσα (binding) μεταβλητή
  - M: εμβέλεια (scope) του x (εκτός από αφαιρέσεις που δεσμεύουν το x)
- Ελευθερες μεταβλητές:

$$FV(x) = \{x\}$$
$$FV((MN)) = FV(M) \cup FV(N)$$
$$FV((\lambda x.E)) = FV(E) - \{x\}$$

- Κλειστός όρος (closed term) M ανν $FV(M) = \emptyset$
  - σύνολο κλειστών όρων: Λ^0

- $M \equiv N$ ταυτόσημοι (identical) όροι
- $\lambda x.M$
  - x: δεσμεύουσα (binding) μεταβλητή
  - M: εμβέλεια (scope) του x (εκτός από αφαιρέσεις που δεσμεύουν το x)
- Ελευθερες μεταβλητές:

$$
\begin{aligned}
FV(x) &= \{x\} \\
FV((MN)) &= FV(M) \cup FV(N) \\
FV((\lambda x.E)) &= FV(E) - \{x\}
\end{aligned}
$$

- Κλειστός όρος (closed term) M ανν $FV(M) = \emptyset$
  - σύνολο κλειστών όρων: Λ^0

- $M \equiv N$ ταυτόσημοι (identical) όροι
- $\lambda x.M$
  - x: δεσμεύουσα (binding) μεταβλητή
  - M: εμβέλεια (scope) του x (εκτός από αφαιρέσεις που δεσμεύουν το x)
- Ελευθερες μεταβλητές:

$$
\begin{aligned}
FV(x) &= \{x\} \\
FV((MN)) &= FV(M) \cup FV(N) \\
FV((\lambda x.E)) &= FV(E) - \{x\}
\end{aligned}
$$

- Κλειστός όρος (closed term) M ανν $FV(M) = \emptyset$
  - σύνολο κλειστών όρων: Λ^0

- Μετασυμβολισμός $M[x := N]$
- $x[x := N] \;\equiv\; N$
- $y[x := N] \;\equiv\; y$, αν $y \neq x$
- $(PQ)[x := N] \;\equiv\; (P[x := N]\; Q[x := N])$
- $(\lambda x.P)[x := N] \;\equiv\; (\lambda x.P)$
- $(\lambda y.P)[x := N] \;\equiv\; (\lambda y.P[x := N])$, αν

$$x \neq y \;\wedge\; (y \notin FV(N) \vee x \notin FV(P))$$

- $(\lambda y.P)[x := N] \;\equiv\; \lambda z.P[y := z][x := N]$, αν

$$x \neq y \;\wedge\; y \in FV(N) \;\wedge\; x \in FV(P) \;\wedge\; z \notin FV(N) \cup FV(P)$$

- Μετασυμβολισμός $M[x := N]$
- $x[x := N] \;\equiv\; N$
- $y[x := N] \;\equiv\; y$, αν $y \neq x$
- $(PQ)[x := N] \;\equiv\; (P[x := N]\; Q[x := N])$
- $(\lambda x.P)[x := N] \;\equiv\; (\lambda x.P)$
- $(\lambda y.P)[x := N] \;\equiv\; (\lambda y.P[x := N])$, αν

$$x \neq y \;\wedge\; (y \notin FV(N) \vee x \notin FV(P))$$

- $(\lambda y.P)[x := N] \;\equiv\; \lambda z.P[y := z][x := N]$, αν

$$x \neq y \;\wedge\; y \in FV(N) \;\wedge\; x \in FV(P) \;\wedge\; z \notin FV(N) \cup FV(P)$$

- Μετασυμβολισμός $M[x := N]$
- $x[x := N] \;\equiv\; N$
- $y[x := N] \;\equiv\; y$, αν $y \neq x$
- $(PQ)[x := N] \;\equiv\; (P[x := N]\, Q[x := N])$
- $(\lambda x.P)[x := N] \;\equiv\; (\lambda x.P)$
- $(\lambda y.P)[x := N] \;\equiv\; (\lambda y.P[x := N])$, αν

$$x \neq y \;\wedge\; (y \notin FV(N) \vee x \notin FV(P))$$

- $(\lambda y.P)[x := N] \;\equiv\; \lambda z.P[y := z][x := N]$, αν

$$x \neq y \;\wedge\; y \in FV(N) \;\wedge\; x \in FV(P) \;\wedge\; z \notin FV(N) \cup FV(P)$$

- Μετασυμβολισμός $M[x := N]$
- $x[x := N] \;\equiv\; N$
- $y[x := N] \;\equiv\; y$, αν $y \neq x$
- $(PQ)[x := N] \;\equiv\; (P[x := N]\; Q[x := N])$
- $(\lambda x.P)[x := N] \;\equiv\; (\lambda x.P)$
- $(\lambda y.P)[x := N] \;\equiv\; (\lambda y.P[x := N])$, αν

$$x \neq y \;\wedge\; (y \notin FV(N) \vee x \notin FV(P))$$

- $(\lambda y.P)[x := N] \;\equiv\; \lambda z.P[y := z][x := N]$, αν

$$x \neq y \;\wedge\; y \in FV(N) \;\wedge\; x \in FV(P) \;\wedge\; z \notin FV(N) \cup FV(P)$$

- Μετασυμβολισμός $M[x := N]$
- $x[x := N] \;\equiv\; N$
- $y[x := N] \;\equiv\; y$, αν $y \neq x$
- $(PQ)[x := N] \;\equiv\; (P[x := N]\; Q[x := N])$
- $(\lambda x.P)[x := N] \;\equiv\; (\lambda x.P)$
- $(\lambda y.P)[x := N] \;\equiv\; (\lambda y.P[x := N])$, αν

$$x \neq y \;\wedge\; (y \notin FV(N) \vee x \notin FV(P))$$

- $(\lambda y.P)[x := N] \;\equiv\; \lambda z.P[y := z][x := N]$, αν

$$x \neq y \;\wedge\; y \in FV(N) \;\wedge\; x \in FV(P) \;\wedge\; z \notin FV(N) \cup FV(P)$$

- Μετασυμβολισμός $M[x := N]$
- $x[x := N] \equiv N$
- $y[x := N] \equiv y$, αν $y \neq x$
- $(PQ)[x := N] \equiv (P[x := N]\ Q[x := N])$
- $(\lambda x.P)[x := N] \equiv (\lambda x.P)$
- $(\lambda y.P)[x := N] \equiv (\lambda y.P[x := N])$, αν

$$x \neq y \ \wedge \ (y \notin FV(N) \vee x \notin FV(P))$$

- $(\lambda y.P)[x := N] \equiv \lambda z.P[y := z][x := N]$, αν

$$x \neq y \ \wedge \ y \in FV(N) \ \wedge \ x \in FV(P) \ \wedge \ z \notin FV(N) \cup FV(P)$$

- Μετασυμβολισμός $M[x := N]$
- $x[x := N] \;\equiv\; N$
- $y[x := N] \;\equiv\; y$, αν $y \neq x$
- $(PQ)[x := N] \;\equiv\; (P[x := N]\; Q[x := N])$
- $(\lambda x.P)[x := N] \;\equiv\; (\lambda x.P)$
- $(\lambda y.P)[x := N] \;\equiv\; (\lambda y.P[x := N])$, αν

$$x \neq y \;\wedge\; (y \notin FV(N) \vee x \notin FV(P))$$

- $(\lambda y.P)[x := N] \;\equiv\; \lambda z.P[y := z][x := N]$, αν

$$x \neq y \;\wedge\; y \in FV(N) \;\wedge\; x \in FV(P) \;\wedge\; z \notin FV(N) \cup FV(P)$$

$$\lambda x.E \ \rightarrow_\alpha \ \lambda y.E[x := y]$$

(για $y \notin FV(E)$)

- Το όνομα της δεσμεύουσας μεταβλητής δεν είναι σημαντικό
- α-μετατροπή: αλλαγή του ονόματος δεσμεύουσας μεταβλητής
- $y \notin FV(E)$: δε δεσμεύονται ελεύθερες περιπτώσεις της νέας μεταβλητής
- Παραδείγματα:
  - $\lambda\mathtt{x.x} \quad \rightarrow_\alpha \quad \lambda\mathtt{y.y}$
  - $\lambda\mathtt{x.z\ x} \quad \rightarrow_\alpha \quad \lambda\mathtt{y.z\ y}$
  - $\lambda\mathtt{x.}\lambda\mathtt{y.z\ x\ y} \quad \rightarrow_\alpha \quad \lambda\mathtt{y.}\lambda\mathtt{w.z\ y\ w}$
  - $\lambda\mathtt{x.}\lambda\mathtt{z.z\ x\ y} \quad \not\rightarrow_\alpha \quad \lambda\mathtt{y.}\lambda\mathtt{z.z\ y\ y}$

$$\lambda x.E \;\rightarrow_\alpha\; \lambda y.E[x := y]$$

(για $y \notin FV(E)$)

- Το όνομα της δεσμεύουσας μεταβλητής δεν είναι σημαντικό
- α-μετατροπή: αλλαγή του ονόματος δεσμεύουσας μεταβλητής
- $y \notin FV(E)$: δε δεσμεύονται ελεύθερες περιπτώσεις της νέας μεταβλητής
- Παραδείγματα:
  - $\lambda\texttt{x.x} \;\rightarrow_\alpha\; \lambda\texttt{y.y}$
  - $\lambda\texttt{x.z x} \;\rightarrow_\alpha\; \lambda\texttt{y.z y}$
  - $\lambda\texttt{x.}\lambda\texttt{y.z x y} \;\rightarrow_\alpha\; \lambda\texttt{y.}\lambda\texttt{w.z y w}$
  - $\lambda\texttt{x.}\lambda\texttt{z.z x y} \;\not\rightarrow_\alpha\; \lambda\texttt{y.}\lambda\texttt{z.z y y}$

$$\lambda x.E \;\rightarrow_\alpha\; \lambda y.E[x := y]$$

(για $y \notin FV(E)$)

- Το όνομα της δεσμεύουσας μεταβλητής δεν είναι σημαντικό
- α-μετατροπή: αλλαγή του ονόματος δεσμεύουσας μεταβλητής
- $y \notin FV(E)$: δε δεσμεύονται ελεύθερες περιπτώσεις της νέας μεταβλητής
- Παραδείγματα:
  - $\lambda x.x \quad \rightarrow_\alpha \quad \lambda y.y$
  - $\lambda x.z\ x \quad \rightarrow_\alpha \quad \lambda y.z\ y$
  - $\lambda x.\lambda y.z\ x\ y \quad \rightarrow_\alpha \quad \lambda y.\lambda w.z\ y\ w$
  - $\lambda x.\lambda z.z\ x\ y \quad \not\rightarrow_\alpha \quad \lambda y.\lambda z.z\ y\ y$

$$\lambda x.E \;\rightarrow_\alpha\; \lambda y.E[x := y]$$

(για $y \notin FV(E)$)

- Το όνομα της δεσμεύουσας μεταβλητής δεν είναι σημαντικό
- α-μετατροπή: αλλαγή του ονόματος δεσμεύουσας μεταβλητής
- $y \notin FV(E)$: δε δεσμεύονται ελεύθερες περιπτώσεις της νέας μεταβλητής
- Παραδείγματα:
  - $\lambda x.x \;\rightarrow_\alpha\; \lambda y.y$
  - $\lambda x.z\ x \;\rightarrow_\alpha\; \lambda y.z\ y$
  - $\lambda x.\lambda y.z\ x\ y \;\rightarrow_\alpha\; \lambda y.\lambda w.z\ y\ w$
  - $\lambda x.\lambda z.z\ x\ y \;\not\rightarrow_\alpha\; \lambda y.\lambda z.z\ y\ y$

$$\lambda x.E \ \rightarrow_\alpha \ \lambda y.E[x := y]$$

(για $y \notin FV(E)$)

- Το όνομα της δεσμεύουσας μεταβλητής δεν είναι σημαντικό
- α-μετατροπή: αλλαγή του ονόματος δεσμεύουσας μεταβλητής
- $y \notin FV(E)$: δε δεσμεύονται ελεύθερες περιπτώσεις της νέας μεταβλητής
- Παραδείγματα:
  - $\lambda\texttt{x.x} \ \rightarrow_\alpha \ \lambda\texttt{y.y}$
  - $\lambda\texttt{x.z x} \ \rightarrow_\alpha \ \lambda\texttt{y.z y}$
  - $\lambda\texttt{x.}\lambda\texttt{y.z x y} \ \rightarrow_\alpha \ \lambda\texttt{y.}\lambda\texttt{w.z y w}$
  - $\lambda\texttt{x.}\lambda\texttt{z.z x y} \ \not\rightarrow_\alpha \ \lambda\texttt{y.}\lambda\texttt{z.z y y}$

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

- Υπολογισμός = εφαρμογή συναρτήσεων
- β-μετατροπή: ένα υπολογιστικό βήμα
- Παραδείγματα:
  - $(\lambda\texttt{x.z x)w} \;\rightarrow_\beta\; \texttt{z w}$
  - $(\lambda\texttt{x.}\lambda\texttt{y.z x y)w} \;\rightarrow_\beta\; \lambda\texttt{y.z w y}$
  - $(\lambda\texttt{y.z y (}\lambda\texttt{x.x y))w} \;\rightarrow_\beta\; \texttt{z w(}\lambda\texttt{x.x w)}$
  - $(\lambda\texttt{x.}\lambda\texttt{y.z x y)(w y)} \;\not\rightarrow_\beta\; \lambda\texttt{y.z (w y) y}$

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

- Υπολογισμός = εφαρμογή συναρτήσεων
- β-μετατροπή: ένα υπολογιστικό βήμα
- Παραδείγματα:
  - `(λx.z x)w` $\to_\beta$ `z w`
  - `(λx.λy.z x y)w` $\to_\beta$ `λy.z w y`
  - `(λy.z y (λx.x y))w` $\to_\beta$ `z w(λx.x w)`
  - `(λx.λy.z x y)(w y)` $\not\to_\beta$ `λy.z (w y) y`

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

- Υπολογισμός = εφαρμογή συναρτήσεων
- β-μετατροπή: ένα υπολογιστικό βήμα
- Παραδείγματα:
  - `(`λ`x.z x)w` $\rightarrow_\beta$ `z w`
  - `(`λ`x.`λ`y.z x y)w` $\rightarrow_\beta$ λ`y.z w y`
  - `(`λ`y.z y (`λ`x.x y))w` $\rightarrow_\beta$ `z w(`λ`x.x w)`
  - `(`λ`x.`λ`y.z x y)(w y)` $\not\rightarrow_\beta$ λ`y.z (w y) y`

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

- Υπολογισμός = εφαρμογή συναρτήσεων
- β-μετατροπή: ένα υπολογιστικό βήμα
- Παραδείγματα:
  - $(\lambda \mathtt{x.z\ x})\mathtt{w} \ \rightarrow_\beta \ \mathtt{z\ w}$
  - $(\lambda \mathtt{x.}\lambda \mathtt{y.z\ x\ y})\mathtt{w} \ \rightarrow_\beta \ \lambda \mathtt{y.z\ w\ y}$
  - $(\lambda \mathtt{y.z\ y\ (}\lambda \mathtt{x.x\ y))w} \ \rightarrow_\beta \ \mathtt{z\ w(}\lambda \mathtt{x.x\ w)}$
  - $(\lambda \mathtt{x.}\lambda \mathtt{y.z\ x\ y)(w\ y)} \ \not\rightarrow_\beta \ \lambda \mathtt{y.z\ (w\ y)\ y}$

$$(\lambda x.M\ x) \ \rightarrow_\eta \ M$$

$$(\text{για } x \notin FV(M))$$

- Ισότητα συναρτήσεων = ίδιο αποτέλεσμα για την ίδια παράμετρο
- η-μετατροπή: εξαλοίφει μία αφαίρεση χρησιμοποιώντας ισότητα συναρτήσεων
- Παραδείγματα:
  - λx.z x $\rightarrow_\eta$ z
  - λy.z x y $\rightarrow_\eta$ z x
  - λx.z x x $\not\rightarrow_\eta$ z x

$$(\lambda x.M\,x) \;\rightarrow_\eta\; M$$

$$(\text{για } x \notin FV(M))$$

- Ισότητα συναρτήσεων = ίδιο αποτέλεσμα για την ίδια παράμετρο
- η-μετατροπή: εξαλοίφει μία αφαίρεση χρησιμοποιώντας ισότητα συναρτήσεων
- Παραδείγματα:
  - λx.z x $\rightarrow_\eta$ z
  - λy.z x y $\rightarrow_\eta$ z x
  - λx.z x x $\not\rightarrow_\eta$ z x

$$(\lambda x.M\, x) \;\rightarrow_\eta\; M$$

$$(\text{για } x \notin FV(M))$$

- Ισότητα συναρτήσεων = ίδιο αποτέλεσμα για την ίδια παράμετρο
- η-μετατροπή: εξαλοίφει μία αφαίρεση χρησιμοποιώντας ισότητα συναρτήσεων
- Παραδείγματα:
  - $\lambda x.z\ x \quad \rightarrow_\eta \quad z$
  - $\lambda y.z\ x\ y \quad \rightarrow_\eta \quad z\ x$
  - $\lambda x.z\ x\ x \quad \not\rightarrow_\eta \quad z\ x$

$$(\lambda x.M\,x) \;\to_\eta\; M$$

(για $x \notin FV(M)$)

- Ισότητα συναρτήσεων = ίδιο αποτέλεσμα για την ίδια παράμετρο
- η-μετατροπή: εξαλοίφει μία αφαίρεση χρησιμοποιώντας ισότητα συναρτήσεων
- Παραδείγματα:
  - λx.z x $\to_\eta$ z
  - λy.z x y $\to_\eta$ z x
  - λx.z x x $\not\to_\eta$ z x

- Μετατροπή $M \to_\chi N$
  - M: χ-redex
  - N: χ-contractum
- Μετατροπή $\to$

$$\to \;=\; \to_\alpha \cup \to_\beta \cup \to_\eta$$

- Αναγωγή $\twoheadrightarrow$

$$\twoheadrightarrow = \bigcup_{i=0}^{\infty} \to^i$$

- Ισότητα λ-όρων =

$$M \twoheadrightarrow N \;\Rightarrow\; M = N$$

$$M = N \;\Rightarrow\; N = M$$

$$M = N \wedge N = P \;\Rightarrow\; M = P$$

- Ομοίως ορίζονται: $\twoheadrightarrow_\chi$ και $=_\chi$

- Μετατροπή $M \to_\chi N$
  - M: χ-redex
  - N: χ-contractum
- Μετατροπή $\to$

$$\to \;=\; \to_\alpha \cup \to_\beta \cup \to_\eta$$

- Αναγωγή $\twoheadrightarrow$

$$\twoheadrightarrow = \bigcup_{i=0}^{\infty} \to^i$$

- Ισότητα λ-όρων =

$$M \twoheadrightarrow N \;\Rightarrow\; M = N$$

$$M = N \;\Rightarrow\; N = M$$

$$M = N \wedge N = P \;\Rightarrow\; M = P$$

- Ομοίως ορίζονται: $\twoheadrightarrow_\chi$ και $=_\chi$

- Μετατροπή $M \rightarrow_\chi N$
  - M: χ-redex
  - N: χ-contractum
- Μετατροπή $\rightarrow$

$$\rightarrow \;=\; \rightarrow_\alpha \cup \rightarrow_\beta \cup \rightarrow_\eta$$

- Αναγωγή $\twoheadrightarrow$

$$\twoheadrightarrow = \bigcup_{i=0}^{\infty} \rightarrow^i$$

- Ισότητα λ-όρων =

$$M \twoheadrightarrow N \;\Rightarrow\; M = N$$

$$M = N \;\Rightarrow\; N = M$$

$$M = N \wedge N = P \;\Rightarrow\; M = P$$

- Ομοίως ορίζονται: $\twoheadrightarrow_\chi$ και $=_\chi$

- Μετατροπή $M \to_\chi N$
  - M: χ-redex
  - N: χ-contractum
- Μετατροπή $\to$

$$\to \; = \; \to_\alpha \cup \to_\beta \cup \to_\eta$$

- Αναγωγή $\twoheadrightarrow$

$$\twoheadrightarrow = \bigcup_{i=0}^{\infty} \to^i$$

- Ισότητα λ-όρων =

$$M \twoheadrightarrow N \;\Rightarrow\; M = N$$

$$M = N \;\Rightarrow\; N = M$$

$$M = N \wedge N = P \;\Rightarrow\; M = P$$

- Ομοίως ορίζονται: $\twoheadrightarrow_\chi$ και $=_\chi$

- Μετατροπή $M \rightarrow_\chi N$
  - M: χ-redex
  - N: χ-contractum
- Μετατροπή $\rightarrow$

$$\rightarrow \; = \; \rightarrow_\alpha \cup \rightarrow_\beta \cup \rightarrow_\eta$$

- Αναγωγή $\twoheadrightarrow$

$$\twoheadrightarrow = \bigcup_{i=0}^{\infty} \rightarrow^i$$

- Ισότητα λ-όρων =

$$M \twoheadrightarrow N \;\Rightarrow\; M = N$$

$$M = N \;\Rightarrow\; N = M$$

$$M = N \wedge N = P \;\Rightarrow\; M = P$$

- Ομοίως ορίζονται: $\twoheadrightarrow_\chi$ και $=_\chi$

- Υπολογιστικό βήμα: $\to_\beta$ και $\to_\eta$
- Όταν δε μπορεί να γίνει άλλο υπολογιστικό βήμα, έχουμε τέλος υπολογισμού
- Κανονική μορφή (canonical form) M: όρος που δεν περιέχει β- και η-redexes
- Παραδείγματα κανονικών μορφών:
  `λx.x`   `λf.f(λx.x f)`
- Όχι κανονική μορφή:
  `λz.(λf.λx.f z x)(λy.y)` $\to_\beta$ `λz.λx.(λy.y) z x`
- M σε κανονική μορφή $\wedge$ $M \twoheadrightarrow N \Rightarrow M =_\alpha N$

- Υπολογιστικό βήμα: $\longrightarrow_\beta$ και $\longrightarrow_\eta$
- Όταν δε μπορεί να γίνει άλλο υπολογιστικό βήμα, έχουμε τέλος υπολογισμού
- Κανονική μορφή (canonical form) M: όρος που δεν περιέχει β- και η-redexes
- Παραδείγματα κανονικών μορφών:
  λx.x $\quad$ λf.f(λx.x f)
- Όχι κανονική μορφή:
  λz.(λf.λx.f z x)(λy.y) $\longrightarrow_\beta$ λz.λx.(λy.y) z x
- M σε κανονική μορφή $\wedge$ $M \twoheadrightarrow N \Rightarrow M =_\alpha N$

Κανονικές Μορφές
Ορισμοί

- Υπολογιστικό βήμα: $\rightarrow_\beta$ και $\rightarrow_\eta$
- Όταν δε μπορεί να γίνει άλλο υπολογιστικό βήμα, έχουμε τέλος υπολογισμού
- Κανονική μορφή (canonical form) M: όρος που δεν περιέχει β- και η-redexes
- Παραδείγματα κανονικών μορφών:
  `λx.x` `λf.f(λx.x f)`
- Όχι κανονική μορφή:
  `λz.(λf.λx.f z x)(λy.y)` $\rightarrow_\beta$ `λz.λx.(λy.y) z x`
- M σε κανονική μορφή $\wedge\ M \twoheadrightarrow N \ \Rightarrow\ M =_\alpha N$

- Υπολογιστικό βήμα: $\longrightarrow_\beta$ και $\longrightarrow_\eta$
- Όταν δε μπορεί να γίνει άλλο υπολογιστικό βήμα, έχουμε τέλος υπολογισμού
- Κανονική μορφή (canonical form) M: όρος που δεν περιέχει β- και η-redexes
- Παραδείγματα κανονικών μορφών:
  $\lambda\texttt{x.x}$ $\quad\quad$ $\lambda\texttt{f.f}(\lambda\texttt{x.x f})$
- Όχι κανονική μορφή:
  $\lambda\texttt{z.}(\lambda\texttt{f.}\lambda\texttt{x.f z x})(\lambda\texttt{y.y}) \longrightarrow_\beta \lambda\texttt{z.}\lambda\texttt{x.}(\lambda\texttt{y.y})\texttt{ z x}$
- M σε κανονική μορφή $\wedge$ $M \twoheadrightarrow N \;\Rightarrow\; M =_\alpha N$

Κανονικές Μορφές
Ορισμοί

- Υπολογιστικό βήμα: $\longrightarrow_\beta$ και $\longrightarrow_\eta$
- Όταν δε μπορεί να γίνει άλλο υπολογιστικό βήμα, έχουμε τέλος υπολογισμού
- Κανονική μορφή (canonical form) M: όρος που δεν περιέχει β- και η-redexes
- Παραδείγματα κανονικών μορφών:
  λx.x $\qquad$ λf.f(λx.x f)
- Όχι κανονική μορφή:
  λz.(λf.λx.f z x)(λy.y) $\longrightarrow_\beta$ λz.λx.(λy.y) z x
- M σε κανονική μορφή $\wedge$ $M \twoheadrightarrow N \quad \Rightarrow \quad M =_\alpha N$

- Υπολογιστικό βήμα: $\rightarrow_\beta$ και $\rightarrow_\eta$
- Όταν δε μπορεί να γίνει άλλο υπολογιστικό βήμα, έχουμε τέλος υπολογισμού
- Κανονική μορφή (canonical form) M: όρος που δεν περιέχει β- και η-redexes
- Παραδείγματα κανονικών μορφών:
  `λx.x` `λf.f(λx.x f)`
- Όχι κανονική μορφή:
  `λz.(λf.λx.f z x)(λy.y)` $\rightarrow_\beta$ `λz.λx.(λy.y) z x`
- M σε κανονική μορφή $\wedge$ $M \twoheadrightarrow N$ $\Rightarrow$ $M =_\alpha N$

- Υπολογιστικό βήμα: $\longrightarrow_\beta$ και $\longrightarrow_\eta$
- Όταν δε μπορεί να γίνει άλλο υπολογιστικό βήμα, έχουμε τέλος υπολογισμού
- Κανονική μορφή (canonical form) M: όρος που δεν περιέχει β- και η-redexes
- Παραδείγματα κανονικών μορφών:
  $\lambda\mathtt{x.x}$ $\qquad\qquad$ $\lambda\mathtt{f.f}(\lambda\mathtt{x.x\ f})$
- Όχι κανονική μορφή:
  $\lambda\mathtt{z.}(\lambda\mathtt{f.}\lambda\mathtt{x.f\ z\ x})(\lambda\mathtt{y.y}) \longrightarrow_\beta \lambda\mathtt{z.}\lambda\mathtt{x.}(\lambda\mathtt{y.y})\ \mathtt{z\ x}$
- M σε κανονική μορφή $\wedge\ M \twoheadrightarrow N \;\Rightarrow\; M =_\alpha N$

- Κανονικοποιήσιμος (normalizing) όρος: μπορεί να αναχθεί σε κανονική μορφή
- Κανονικοποιήσιμος όρος:

$$\begin{array}{ll} & \lambda z.(\lambda f.\lambda x.f\ z\ x)(\lambda y.y) \\ \rightarrow & \lambda z.\lambda x.(\lambda y.y)\ z\ x \\ \rightarrow & \lambda z.\lambda x.z\ x \end{array}$$

- Μη κανονικοποιήσιμος όρος:

$$\Omega \equiv (\lambda x.\ \ x\ x)\ (\lambda x.\ \ x\ x)$$

$$\Omega \rightarrow_\beta \Omega$$

- Μη κανονικοποιήσιμος όρος:

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

$$M \rightarrow_\beta M\,y \rightarrow_\beta M\,y\ \ y \rightarrow_\beta M\,y\ \ y\ \ y \rightarrow_\beta \dots$$

- Κανονικοποιήσιμος (normalizing) όρος: μπορεί να αναχθεί σε κανονική μορφή
- Κανονικοποιήσιμος όρος:

$$\begin{aligned} & \lambda z.(\lambda f.\lambda x.f\ z\ x)(\lambda y.y) \\ \to\ & \lambda z.\lambda x.(\lambda y.y)\ z\ x \\ \to\ & \lambda z.\lambda x.z\ x \end{aligned}$$

- Μη κανονικοποιήσιμος όρος:

$$\Omega \equiv (\lambda x.\ \ x\ x)\ (\lambda x.\ \ x\ x)$$

$$\Omega \to_\beta \Omega$$

- Μη κανονικοποιήσιμος όρος:

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

$$M \to_\beta M\,y \to_\beta M\,y\ \ y \to_\beta M\,y\ \ y\ \ y \to_\beta \ldots$$

- Κανονικοποιήσιμος (normalizing) όρος: μπορεί να αναχθεί σε κανονική μορφή
- Κανονικοποιήσιμος όρος:

$$\begin{array}{ll} & \lambda z.(\lambda f.\lambda x.f\ z\ x)(\lambda y.y) \\ \rightarrow & \lambda z.\lambda x.(\lambda y.y)\ z\ x \\ \rightarrow & \lambda z.\lambda x.z\ x \end{array}$$

- Μη κανονικοποιήσιμος όρος:

$$\Omega \equiv (\lambda x.\ \ x\ x)\ (\lambda x.\ \ x\ x)$$

$$\Omega \rightarrow_\beta \Omega$$

- Μη κανονικοποιήσιμος όρος:

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

$$M \rightarrow_\beta M\,y \rightarrow_\beta M\,y\ \ y \rightarrow_\beta M\,y\ \ y\ \ y \rightarrow_\beta \ldots$$

- Κανονικοποιήσιμος (normalizing) όρος: μπορεί να αναχθεί σε κανονική μορφή
- Κανονικοποιήσιμος όρος:

$$\begin{array}{ll} & \lambda \mathtt{z}.(\lambda \mathtt{f}.\lambda \mathtt{x}.\mathtt{f}\ \mathtt{z}\ \mathtt{x})(\lambda \mathtt{y}.\mathtt{y}) \\ \longrightarrow & \lambda \mathtt{z}.\lambda \mathtt{x}.(\lambda \mathtt{y}.\mathtt{y})\ \mathtt{z}\ \mathtt{x} \\ \longrightarrow & \lambda \mathtt{z}.\lambda \mathtt{x}.\mathtt{z}\ \mathtt{x} \end{array}$$

- Μη κανονικοποιήσιμος όρος:

$$\Omega \equiv (\lambda \mathtt{x}.\ \mathtt{x}\ \mathtt{x})\ (\lambda \mathtt{x}.\ \mathtt{x}\ \mathtt{x})$$

$$\Omega \longrightarrow_\beta \Omega$$

- Μη κανονικοποιήσιμος όρος:

$$M \equiv (\lambda \mathtt{x}.\ \mathtt{x}\ \mathtt{x}\ \mathtt{y})\ (\lambda \mathtt{x}.\ \mathtt{x}\ \mathtt{x}\ \mathtt{y})$$

$$M \longrightarrow_\beta M\,\mathtt{y} \longrightarrow_\beta M\,\mathtt{y}\ \mathtt{y} \longrightarrow_\beta M\,\mathtt{y}\ \mathtt{y}\ \mathtt{y} \longrightarrow_\beta \ldots$$

- Κανονικοποιήσιμος (normalizing) όρος: μπορεί να αναχθεί σε κανονική μορφή
- Κανονικοποιήσιμος όρος:

$$\begin{array}{ll} & \lambda\texttt{z.}(\lambda\texttt{f.}\lambda\texttt{x.f z x})(\lambda\texttt{y.y}) \\ \to & \lambda\texttt{z.}\lambda\texttt{x.}(\lambda\texttt{y.y})\texttt{ z x} \\ \to & \lambda\texttt{z.}\lambda\texttt{x.z x} \end{array}$$

- Μη κανονικοποιήσιμος όρος:

$$\Omega \equiv (\lambda\texttt{x. x x})\ (\lambda\texttt{x. x x})$$

$$\Omega \to_\beta \Omega$$

- Μη κανονικοποιήσιμος όρος:

$$M \equiv (\lambda\texttt{x. x x y})\ (\lambda\texttt{x. x x y})$$

$$M \to_\beta M\,\texttt{y} \to_\beta M\,\texttt{y y} \to_\beta M\,\texttt{y y y} \to_\beta \dots$$

- Κανονικοποιήσιμος (normalizing) όρος: μπορεί να αναχθεί σε κανονική μορφή
- Κανονικοποιήσιμος όρος:

$$
\begin{array}{ll}
 & \lambda \mathtt{z}.(\lambda \mathtt{f}.\lambda \mathtt{x}.\mathtt{f\ z\ x})(\lambda \mathtt{y}.\mathtt{y}) \\
\longrightarrow & \lambda \mathtt{z}.\lambda \mathtt{x}.(\lambda \mathtt{y}.\mathtt{y})\ \mathtt{z\ x} \\
\longrightarrow & \lambda \mathtt{z}.\lambda \mathtt{x}.\mathtt{z\ x}
\end{array}
$$

- Μη κανονικοποιήσιμος όρος:

$$\Omega \equiv (\lambda \mathtt{x}.\ \mathtt{x\ x})\ (\lambda \mathtt{x}.\ \mathtt{x\ x})$$

$$\Omega \longrightarrow_\beta \Omega$$

- Μη κανονικοποιήσιμος όρος:

$$M \equiv (\lambda \mathtt{x}.\ \mathtt{x\ x\ y})\ (\lambda \mathtt{x}.\ \mathtt{x\ x\ y})$$

$$M \longrightarrow_\beta M\,\mathtt{y} \longrightarrow_\beta M\,\mathtt{y\ y} \longrightarrow_\beta M\,\mathtt{y\ y\ y} \longrightarrow_\beta \ldots$$

- Διαφορετικοί δρόμοι αναγωγής:

$$M \equiv (\lambda \mathtt{x}.(\lambda \mathtt{y}.\mathtt{x}\ \mathtt{y})\ \mathtt{z})\mathtt{w}$$

$$M \to (\lambda \mathtt{y}.\mathtt{w}\ \mathtt{y})\ \mathtt{z} \to \mathtt{w}\ \mathtt{z}$$

$$M \to (\lambda \mathtt{x}.\mathtt{x}\ \mathtt{z})\mathtt{w} \to \mathtt{w}\ \mathtt{z}$$

- Δεν οδηγούν όλοι οι δρόμοι σε κανονικοποίηση:

$$M \equiv (\lambda \mathtt{z}.\mathtt{y})\Omega$$

$$M \to_\beta \mathtt{y} \quad \text{αλλά } (\Omega \to_\beta \Omega)\text{: } M \to_\beta M$$

- Ισχυρά κανονικοποιήσιμος (strongly normalizing) όρος: όλες οι στρατηγικές οδηγούν σε κανονική μορφή
- Αναγωγή κανονικής σειράς (normal order reduction): β-μετατροπή του πρώτου λ-όρου από αριστερά

- Διαφορετικοί δρόμοι αναγωγής:

$$M \equiv (\lambda\mathtt{x}.(\lambda\mathtt{y}.\mathtt{x}\ \mathtt{y})\ \mathtt{z})\mathtt{w}$$

$$M \to (\lambda\mathtt{y}.\mathtt{w}\ \mathtt{y})\ \mathtt{z} \to \mathtt{w}\ \mathtt{z}$$

$$M \to (\lambda\mathtt{x}.\mathtt{x}\ \mathtt{z})\mathtt{w} \to \mathtt{w}\ \mathtt{z}$$

- Δεν οδηγούν όλοι οι δρόμοι σε κανονικοποίηση:

$$M \equiv (\lambda\mathtt{z}.\mathtt{y})\Omega$$

$$M \to_\beta \mathtt{y} \quad \text{αλλά } (\Omega \to_\beta \Omega)\text{: } M \to_\beta M$$

- Ισχυρά κανονικοποιήσιμος (strongly normalizing) όρος: όλες οι στρατηγικές οδηγούν σε κανονική μορφή
- Αναγωγή κανονικής σειράς (normal order reduction): β-μετατροπή του πρώτου λ-όρου από αριστερά

- Διαφορετικοί δρόμοι αναγωγής:

$$M \equiv (\lambda \mathtt{x}.(\lambda \mathtt{y}.\mathtt{x}\ \mathtt{y})\ \mathtt{z})\mathtt{w}$$

$$M \to (\lambda \mathtt{y}.\mathtt{w}\ \mathtt{y})\ \mathtt{z} \to \mathtt{w}\ \mathtt{z}$$

$$M \to (\lambda \mathtt{x}.\mathtt{x}\ \mathtt{z})\mathtt{w} \to \mathtt{w}\ \mathtt{z}$$

- Δεν οδηγούν όλοι οι δρόμοι σε κανονικοποίηση:

$$M \equiv (\lambda \mathtt{z}.\mathtt{y})\Omega$$

$$M \to_\beta \mathtt{y} \quad \text{αλλά} \quad (\Omega \to_\beta \Omega)\text{:}\ M \to_\beta M$$

- Ισχυρά κανονικοποιήσιμος (strongly normalizing) όρος: όλες οι στρατηγικές οδηγούν σε κανονική μορφή
- Αναγωγή κανονικής σειράς (normal order reduction): β-μετατροπή του πρώτου λ-όρου από αριστερά

- Διαφορετικοί δρόμοι αναγωγής:

$$M \equiv (\lambda\mathtt{x}.(\lambda\mathtt{y}.\mathtt{x\ y})\ \mathtt{z})\mathtt{w}$$

$$M \rightarrow (\lambda\mathtt{y}.\mathtt{w\ y})\ \mathtt{z} \rightarrow \mathtt{w\ z}$$

$$M \rightarrow (\lambda\mathtt{x}.\mathtt{x\ z})\mathtt{w} \rightarrow \mathtt{w\ z}$$

- Δεν οδηγούν όλοι οι δρόμοι σε κανονικοποίηση:

$$M \equiv (\lambda\mathtt{z}.\mathtt{y})\Omega$$

$$M \rightarrow_\beta \mathtt{y} \quad \text{αλλά} \quad (\Omega \rightarrow_\beta \Omega)\text{:}\ M \rightarrow_\beta M$$

- Ισχυρά κανονικοποιήσιμος (strongly normalizing) όρος: όλες οι στρατηγικές οδηγούν σε κανονική μορφή
- Αναγωγή κανονικής σειράς (normal order reduction): β-μετατροπή του πρώτου λ-όρου από αριστερά

- Διαφορετικοί δρόμοι αναγωγής:

$$M \equiv (\lambda \mathtt{x}.(\lambda \mathtt{y}.\mathtt{x}\ \mathtt{y})\ \mathtt{z})\mathtt{w}$$

$$M \rightarrow (\lambda \mathtt{y}.\mathtt{w}\ \mathtt{y})\ \mathtt{z} \rightarrow \mathtt{w}\ \mathtt{z}$$

$$M \rightarrow (\lambda \mathtt{x}.\mathtt{x}\ \mathtt{z})\mathtt{w} \rightarrow \mathtt{w}\ \mathtt{z}$$

- Δεν οδηγούν όλοι οι δρόμοι σε κανονικοποίηση:

$$M \equiv (\lambda \mathtt{z}.\mathtt{y})\Omega$$

$$M \rightarrow_\beta \mathtt{y} \quad \text{αλλά} \quad (\Omega \rightarrow_\beta \Omega)\text{:}\ M \rightarrow_\beta M$$

- Ισχυρά κανονικοποιήσιμος (strongly normalizing) όρος: όλες οι στρατηγικές οδηγούν σε κανονική μορφή
- Αναγωγή κανονικής σειράς (normal order reduction): β-μετατροπή του πρώτου λ-όρου από αριστερά

- Διαφορετικοί δρόμοι αναγωγής:

$$M \equiv (\lambda \mathtt{x}.(\lambda \mathtt{y}.\mathtt{x}\ \mathtt{y})\ \mathtt{z})\mathtt{w}$$

$$M \to (\lambda \mathtt{y}.\mathtt{w}\ \mathtt{y})\ \mathtt{z} \to \mathtt{w}\ \mathtt{z}$$

$$M \to (\lambda \mathtt{x}.\mathtt{x}\ \mathtt{z})\mathtt{w} \to \mathtt{w}\ \mathtt{z}$$

- Δεν οδηγούν όλοι οι δρόμοι σε κανονικοποίηση:

$$M \equiv (\lambda \mathtt{z}.\mathtt{y})\Omega$$

$$M \to_\beta \mathtt{y} \ \text{ αλλά } (\Omega \to_\beta \Omega)\text{: } M \to_\beta M$$

- Ισχυρά κανονικοποιήσιμος (strongly normalizing) όρος: όλες οι στρατηγικές οδηγούν σε κανονική μορφή
- Αναγωγή κανονικής σειράς (normal order reduction): β-μετατροπή του πρώτου λ-όρου από αριστερά

- Διαφορετικοί δρόμοι αναγωγής:

$$M \equiv (\lambda \texttt{x.}(\lambda \texttt{y.x y}) \texttt{ z)w}$$

$$M \rightarrow (\lambda \texttt{y.w y}) \texttt{ z} \rightarrow \texttt{w z}$$

$$M \rightarrow (\lambda \texttt{x.x z)w} \rightarrow \texttt{w z}$$

- Δεν οδηγούν όλοι οι δρόμοι σε κανονικοποίηση:

$$M \equiv (\lambda \texttt{z.y})\Omega$$

$$M \rightarrow_\beta \texttt{y} \quad \text{αλλά} \quad (\Omega \rightarrow_\beta \Omega)\text{: } M \rightarrow_\beta M$$

- Ισχυρά κανονικοποιήσιμος (strongly normalizing) όρος: όλες οι στρατηγικές οδηγούν σε κανονική μορφή
- Αναγωγή κανονικής σειράς (normal order reduction): β-μετατροπή του πρώτου λ-όρου από αριστερά

- Διαφορετικοί δρόμοι αναγωγής:

$$M \equiv (\lambda \mathtt{x.}(\lambda \mathtt{y.x\ y})\ \mathtt{z})\mathtt{w}$$

$$M \rightarrow (\lambda \mathtt{y.w\ y})\ \mathtt{z} \rightarrow \mathtt{w\ z}$$

$$M \rightarrow (\lambda \mathtt{x.x\ z})\mathtt{w} \rightarrow \mathtt{w\ z}$$

- Δεν οδηγούν όλοι οι δρόμοι σε κανονικοποίηση:

$$M \equiv (\lambda \mathtt{z.y})\Omega$$

$$M \rightarrow_\beta \mathtt{y} \quad \text{αλλά} \quad (\Omega \rightarrow_\beta \Omega)\text{:}\ M \rightarrow_\beta M$$

- Ισχυρά κανονικοποιήσιμος (strongly normalizing) όρος: όλες οι στρατηγικές οδηγούν σε κανονική μορφή
- Αναγωγή κανονικής σειράς (normal order reduction): β-μετατροπή του πρώτου λ-όρου από αριστερά

- Επιθυμούμε ένας όρος να έχει το πολύ μία κανονική μορφή (εκτός από α-μετατροπές)
- Θεώρημα Church-Rosser: Αν $M \twoheadrightarrow M_1$ και $M \twoheadrightarrow M_2$, τότε υπάρχει N ώστε $M_1 \twoheadrightarrow N$ και $M_2 \twoheadrightarrow N$
- Κάθε όρος έχει το πολύ μία κανονική μορφή:

$$M \twoheadrightarrow N_1 \;\wedge\; M \twoheadrightarrow N_2 \;\wedge\; N_1, N2 \text{ κανονικές μορφές} \;\Rightarrow\; N_1 =_\alpha N_2$$

- Η αναγωγή κανονικής σειράς εξασφαλίζει την εύρεση κανονικής μορφής εάν αυτή υπάρχει

- Επιθυμούμε ένας όρος να έχει το πολύ μία κανονική μορφή (εκτός από α-μετατροπές)
- Θεώρημα Church-Rosser: Αν $M \twoheadrightarrow M_1$ και $M \twoheadrightarrow M_2$, τότε υπάρχει N ώστε $M_1 \twoheadrightarrow N$ και $M_2 \twoheadrightarrow N$
- Κάθε όρος έχει το πολύ μία κανονική μορφή:

$$M \twoheadrightarrow N_1 \;\wedge\; M \twoheadrightarrow N_2 \;\wedge\; N_1, N2 \text{ κανονικές μορφές} \quad \Rightarrow \quad N_1 =_\alpha N_2$$

- Η αναγωγή κανονικής σειράς εξασφαλίζει την εύρεση κανονικής μορφής εάν αυτή υπάρχει

- Επιθυμούμε ένας όρος να έχει το πολύ μία κανονική μορφή (εκτός από α-μετατροπές)
- Θεώρημα Church-Rosser: Αν $M \twoheadrightarrow M_1$ και $M \twoheadrightarrow M_2$, τότε υπάρχει N ώστε $M_1 \twoheadrightarrow N$ και $M_2 \twoheadrightarrow N$
- Κάθε όρος έχει το πολύ μία κανονική μορφή:

$$M \twoheadrightarrow N_1 \ \wedge \ M \twoheadrightarrow N_2 \ \wedge \ N_1, N2 \text{ κανονικές μορφές} \ \Rightarrow \ N_1 =_\alpha N_2$$

- Η αναγωγή κανονικής σειράς εξασφαλίζει την εύρεση κανονικής μορφής εάν αυτή υπάρχει

- Επιθυμούμε ένας όρος να έχει το πολύ μία κανονική μορφή (εκτός από α-μετατροπές)
- Θεώρημα Church-Rosser: Αν $M \twoheadrightarrow M_1$ και $M \twoheadrightarrow M_2$, τότε υπάρχει N ώστε $M_1 \twoheadrightarrow N$ και $M_2 \twoheadrightarrow N$
- Κάθε όρος έχει το πολύ μία κανονική μορφή:

$$M \twoheadrightarrow N_1 \;\wedge\; M \twoheadrightarrow N_2 \;\wedge\; N_1, N2 \text{ κανονικές μορφές} \;\Rightarrow\; N_1 =_\alpha N_2$$

- Η αναγωγή κανονικής σειράς εξασφαλίζει την εύρεση κανονικής μορφής εάν αυτή υπάρχει

- Σταθερό σημείο (fixed point) ενός όρου F είναι ένας όρος X ώστε $F\ X \;=\; X$
- Κάθε λ-όρος έχει σταθερό σημείο
- Τελεστής σταθερού σημείου Y:

$$Y \equiv \lambda \mathtt{f}.(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))$$

$$F(Y\ F) \;=\; Y\ F$$

- Απόδειξη:

$$\begin{aligned}
Y\ F &= (\lambda \mathtt{f}.(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F \\
&= (\lambda \mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.F(\mathtt{x}\ \mathtt{x})) \\
&= F((\lambda \mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.F(\mathtt{x}\ \mathtt{x}))) \\
&= F((\lambda \mathtt{f}.(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F) \\
&= F(Y\ F)
\end{aligned}$$

Ιδιότητες λ-λογισμού

Σταθερά Σημεία

- Σταθερό σημείο (fixed point) ενός όρου F είναι ένας όρος X ώστε $F\ X \;=\; X$
- Κάθε λ-όρος έχει σταθερό σημείο
- Τελεστής σταθερού σημείου Y:

$$Y \equiv \lambda\mathtt{f}.(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))$$

$$F(Y\ F) = Y\ F$$

- Απόδειξη:

$$\begin{aligned} Y\ F &= (\lambda\mathtt{f}.(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F \\ &= (\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x})) \\ &= F((\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x}))) \\ &= F((\lambda\mathtt{f}.(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F) \\ &= F(Y\ F) \end{aligned}$$

Ιδιότητες λ-λογισμού
Σταθερά Σημεία

- Σταθερό σημείο (fixed point) ενός όρου F είναι ένας όρος X ώστε $F\,X \;=\; X$
- Κάθε λ-όρος έχει σταθερό σημείο
- Τελεστής σταθερού σημείου Y:

$$Y \equiv \lambda\mathtt{f}.(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))$$

$$F(Y\,F) \;=\; Y\,F$$

- Απόδειξη:

$$\begin{aligned} Y\,F &= (\lambda\mathtt{f}.(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F \\ &= (\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x})) \\ &= F((\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x}))) \\ &= F((\lambda\mathtt{f}.(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F) \\ &= F(Y\,F) \end{aligned}$$

- Σταθερό σημείο (fixed point) ενός όρου F είναι ένας όρος X ώστε $F\,X \;=\; X$
- Κάθε λ-όρος έχει σταθερό σημείο
- Τελεστής σταθερού σημείου Y:

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

$$F(Y\,F) \;=\; Y\,F$$

- Απόδειξη:

$$\begin{aligned}
Y\,F \;&= (\lambda\mathtt{f}.(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F \\
&= (\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x})) \\
&= F((\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x}))) \\
&= F((\lambda\mathtt{f}.(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F) \\
&= F(Y\,F)
\end{aligned}$$

- Σταθερό σημείο (fixed point) ενός όρου F είναι ένας όρος X ώστε $F\ X \ = \ X$
- Κάθε λ-όρος έχει σταθερό σημείο
- Τελεστής σταθερού σημείου Y:

$$Y \ \equiv \ \lambda \mathtt{f}.(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \ \mathtt{x}))(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \ \mathtt{x}))$$

$$F(Y\ F) \ = \ Y\ F$$

- Απόδειξη:

$$\begin{aligned}
Y\ F \ &= (\lambda \mathtt{f}.(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \ \mathtt{x}))(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \ \mathtt{x})))F \\
&= (\lambda \mathtt{x}.F(\mathtt{x}\ \ \mathtt{x}))(\lambda \mathtt{x}.F(\mathtt{x}\ \ \mathtt{x})) \\
&= F((\lambda \mathtt{x}.F(\mathtt{x}\ \ \mathtt{x}))(\lambda \mathtt{x}.F(\mathtt{x}\ \ \mathtt{x}))) \\
&= F((\lambda \mathtt{f}.(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \ \mathtt{x}))(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \ \mathtt{x})))F) \\
&= F(Y\ F)
\end{aligned}$$

- Σταθερό σημείο (fixed point) ενός όρου F είναι ένας όρος X ώστε $F\ X\ =\ X$
- Κάθε λ-όρος έχει σταθερό σημείο
- Τελεστής σταθερού σημείου Y:

$$Y \equiv \lambda \mathtt{f}.(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))$$

$$F(Y\ F) = Y\ F$$

- Απόδειξη:

$$\begin{aligned} Y\ F &= (\lambda \mathtt{f}.(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F \\ &= (\lambda \mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.F(\mathtt{x}\ \mathtt{x})) \\ &= F((\lambda \mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.F(\mathtt{x}\ \mathtt{x}))) \\ &= F((\lambda \mathtt{f}.(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F) \\ &= F(Y\ F) \end{aligned}$$

- Σταθερό σημείο (fixed point) ενός όρου F είναι ένας όρος X ώστε $F\ X \;=\; X$
- Κάθε λ-όρος έχει σταθερό σημείο
- Τελεστής σταθερού σημείου Y:

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

$$F(Y\ F) \;=\; Y\ F$$

- Απόδειξη:

$$\begin{aligned} Y\ F &= (\lambda\mathtt{f}.(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F \\ &= (\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x})) \\ &= F((\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.F(\mathtt{x}\ \mathtt{x}))) \\ &= F((\lambda\mathtt{f}.(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda\mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F) \\ &= F(Y\ F) \end{aligned}$$

- Σταθερό σημείο (fixed point) ενός όρου F είναι ένας όρος X ώστε $F\ X \ = \ X$
- Κάθε λ-όρος έχει σταθερό σημείο
- Τελεστής σταθερού σημείου Y:

$$Y \equiv \lambda \mathtt{f}.(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))$$

$$F(Y\ F) = Y\ F$$

- Απόδειξη:

$$\begin{aligned} Y\ F &= (\lambda \mathtt{f}.(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F \\ &= (\lambda \mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.F(\mathtt{x}\ \mathtt{x})) \\ &= F((\lambda \mathtt{x}.F(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.F(\mathtt{x}\ \mathtt{x}))) \\ &= F((\lambda \mathtt{f}.(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x}))(\lambda \mathtt{x}.\mathtt{f}(\mathtt{x}\ \mathtt{x})))F) \\ &= F(Y\ F) \end{aligned}$$

- Σταθερό σημείο (fixed point) ενός όρου F είναι ένας όρος X ώστε $F\ X \ = \ X$
- Κάθε λ-όρος έχει σταθερό σημείο
- Τελεστής σταθερού σημείου Y:

$$Y \equiv \lambda \texttt{f.}(\lambda \texttt{x.f(x x)})(\lambda \texttt{x.f(x x)})$$

$$F(Y\ F) \ = \ Y\ F$$

- Απόδειξη:

$$\begin{aligned} Y\ F &= (\lambda \texttt{f.}(\lambda \texttt{x.f(x x)})(\lambda \texttt{x.f(x x)}))F \\ &= (\lambda \texttt{x.}F\texttt{(x x)})(\lambda \texttt{x.}F\texttt{(x x)}) \\ &= F((\lambda \texttt{x.}F\texttt{(x x)})(\lambda \texttt{x.}F\texttt{(x x)})) \\ &= F((\lambda \texttt{f.}(\lambda \texttt{x.f(x x)})(\lambda \texttt{x.f(x x)}))F) \\ &= F(Y\ F) \end{aligned}$$

- Κωδικοποίηση αληθοτιμών: ζητάμε: *true* $\neq$ *false*
  - αρκούν δύο άνισες κανονικές τιμές
- Κωδικοποίηση:

$$\mathit{true} \equiv \lambda \mathtt{x}.\lambda \mathtt{y}.\mathtt{x}$$

$$\mathit{false} \equiv \lambda \mathtt{x}.\lambda \mathtt{y}.\mathtt{y}$$

- Κωδικοποίηση *not*: ζητάμε *not true* $=$ *false* και αντίστροφα
- Κωδικοποίηση: *not* $\equiv \lambda \mathtt{z}.\mathtt{z}\ \mathtt{false}\ \mathtt{true}$
- *not true = true false true = false*

- Κωδικοποίηση αληθοτιμών: ζητάμε: *true* $\neq$ *false*
  - αρκούν δύο άνισες κανονικές τιμές
- Κωδικοποίηση:

$$\begin{aligned} \textit{true} &\equiv \lambda \mathtt{x}.\lambda \mathtt{y}.\mathtt{x} \\ \textit{false} &\equiv \lambda \mathtt{x}.\lambda \mathtt{y}.\mathtt{y} \end{aligned}$$

- Κωδικοποίηση *not*: ζητάμε *not true* $=$ *false* και αντίστροφα
- Κωδικοποίηση: *not* $\equiv \lambda \mathtt{z}.\mathtt{z}$ `false true`
- *not true = true false true = false*

- Κωδικοποίηση αληθοτιμών: ζητάμε: *true* $\neq$ *false*
  - αρκούν δύο άνισες κανονικές τιμές
- Κωδικοποίηση:

$$\begin{aligned} \mathit{true} &\equiv \lambda \mathtt{x} . \lambda \mathtt{y} . \mathtt{x} \\ \mathit{false} &\equiv \lambda \mathtt{x} . \lambda \mathtt{y} . \mathtt{y} \end{aligned}$$

- Κωδικοποίηση *not*: ζητάμε *not true* $=$ *false* και αντίστροφα
- Κωδικοποίηση: *not* $\equiv \lambda \mathtt{z} . \mathtt{z}\ \mathtt{false}\ \mathtt{true}$
- *not true = true false true = false*

- Κωδικοποίηση αληθοτιμών: ζητάμε: *true* $\neq$ *false*
    - αρκούν δύο άνισες κανονικές τιμές
- Κωδικοποίηση:

$$\begin{aligned} \textit{true} &\equiv \lambda\mathtt{x}.\lambda\mathtt{y}.\mathtt{x} \\ \textit{false} &\equiv \lambda\mathtt{x}.\lambda\mathtt{y}.\mathtt{y} \end{aligned}$$

- Κωδικοποίηση *not*: ζητάμε *not true* $=$ *false* και αντίστροφα
- Κωδικοποίηση: *not* $\equiv \lambda\mathtt{z}.\mathtt{z}\ \mathtt{false}\ \mathtt{true}$
- *not true = true false true = false*

- Κωδικοποίηση αληθοτιμών: ζητάμε: *true* $\neq$ *false*
  - αρκούν δύο άνισες κανονικές τιμές
- Κωδικοποίηση:

$$\begin{aligned} \textit{true} &\equiv \lambda \mathtt{x}.\lambda \mathtt{y}.\mathtt{x} \\ \textit{false} &\equiv \lambda \mathtt{x}.\lambda \mathtt{y}.\mathtt{y} \end{aligned}$$

- Κωδικοποίηση *not*: ζητάμε *not true* $=$ *false* και αντίστροφα
- Κωδικοποίηση: *not* $\equiv \lambda \mathtt{z}.\mathtt{z}\ \mathtt{false}\ \mathtt{true}$
- *not true = true false true = false*

- Κωδικοποίηση *if*: ζητάμε:

$$\textit{if true then T else F} = T$$

$$\textit{if false then T else F} = F$$

- Κωδικοποίηση:

$$\textit{cond} \equiv \lambda z.\lambda x.\lambda y.\lambda z.z\ x\ y$$

$$\textit{if B then T else F} \equiv \textit{cond B T F}$$

- *if true then T else F = true T F = T*
  if true then T else F = false T F = F
- Όλοι οι λογικοί τελεστές κωδικοποιούνται με *if*

- Κωδικοποίηση *if*: ζητάμε:

$$\textit{if true then T else F} = T$$

$$\textit{if false then T else F} = F$$

- Κωδικοποίηση:

$$\textit{cond} \equiv \lambda z.\lambda x.\lambda y.\lambda z.z\ x\ y$$

$$\textit{if B then T else F} \equiv \textit{cond B T F}$$

- *if true then T else F = true T F = T*
  if true then T else F = false T F = F
- Όλοι οι λογικοί τελεστές κωδικοποιούνται με *if*

- Κωδικοποίηση *if*: ζητάμε:

$$\textit{if true then T else F} = T$$

$$\textit{if false then T else F} = F$$

- Κωδικοποίηση:

$$\textit{cond} \equiv \lambda \mathtt{z}.\lambda \mathtt{x}.\lambda \mathtt{y}.\lambda \mathtt{z}.\mathtt{z}\ \mathtt{x}\ \mathtt{y}$$

$$\textit{if B then T else F} \equiv \textit{cond B T F}$$

- *if true then T else F = true T F = T*
  if true then T else F = false T F = F
- Όλοι οι λογικοί τελεστές κωδικοποιούνται με *if*

- Κωδικοποίηση *if*: ζητάμε:

$$\textit{if true then T else F} = T$$

$$\textit{if false then T else F} = F$$

- Κωδικοποίηση:

$$\textit{cond} \equiv \lambda \mathtt{z}.\lambda \mathtt{x}.\lambda \mathtt{y}.\lambda \mathtt{z}.\mathtt{z}\ \mathtt{x}\ \mathtt{y}$$

$$\textit{if B then T else F} \equiv \textit{cond B T F}$$

- *if true then T else F = true T F = T*
  if true then T else F = false T F = F
- Όλοι οι λογικοί τελεστές κωδικοποιούνται με *if*

- Κωδικοποίηση ζευγών: ζητάμε

$$fst\langle M, N\rangle = M$$
$$snd\langle M, N\rangle = N$$

- Κωδικοποίηση $\langle\rangle$:

$$pair \equiv \lambda z.\lambda x.\lambda y.\lambda z.z\ x\ y$$
$$\langle M, N\rangle \equiv pair\ M\ N$$

- Κωδικοποίηση *fst* και *snd*:

$$fst \equiv \lambda z.z\ true$$
$$snd \equiv \lambda z.z\ false$$

- $fst\langle M, N\rangle = \langle M, N\rangle true = true\ M\ N = M$
  $snd\langle M, N\rangle = \langle M, N\rangle false = false\ M\ N = N$

- Κωδικοποίηση ζευγών: ζητάμε

$$\begin{aligned} fst\langle M, N\rangle &= M \\ snd\langle M, N\rangle &= N \end{aligned}$$

- Κωδικοποίηση $\langle\rangle$:

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

- Κωδικοποίηση *fst* και *snd*:

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

- $fst\langle M, N\rangle = \langle M, N\rangle true = true\ M\ N = M$
  $snd\langle M, N\rangle = \langle M, N\rangle false = false\ M\ N = N$

- Κωδικοποίηση ζευγών: ζητάμε

$$\begin{aligned} fst\langle M, N\rangle &= M \\ snd\langle M, N\rangle &= N \end{aligned}$$

- Κωδικοποίηση $\langle\rangle$:

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

- Κωδικοποίηση *fst* και *snd*:

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

- $fst\langle M, N\rangle = \langle M, N\rangle true = true\ M\ N = M$
  $snd\langle M, N\rangle = \langle M, N\rangle false = false\ M\ N = N$

- Κωδικοποίηση ζευγών: ζητάμε

$$
\begin{aligned}
\mathit{fst}\langle M, N\rangle &= M \\
\mathit{snd}\langle M, N\rangle &= N
\end{aligned}
$$

- Κωδικοποίηση $\langle\rangle$:

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

- Κωδικοποίηση *fst* και *snd*:

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

- $\mathit{fst}\langle M, N\rangle = \langle M, N\rangle \mathit{true} = \mathit{true}\ M\ N = M$
  $\mathit{snd}\langle M, N\rangle = \langle M, N\rangle \mathit{false} = \mathit{false}\ M\ N = N$

- Κωδικοποίηση ακεραίων: ζητάμε (αξιώματα του Peano)

$$0 \neq succ\ N$$

$$succ\ N = succ\ M \Rightarrow N = M$$

- Πολλαπλή εφαρμογή:

$$F^0 \equiv id \equiv \lambda \mathtt{x}.\mathtt{x}$$

$$F^{n+1} \equiv \lambda \mathtt{x}.F(F^n\ \mathtt{x})$$

- Αριθμοειδή του Church:

$$n = \lambda \mathtt{f}.\lambda \mathtt{x}.\mathtt{f}^n \mathtt{x}$$

- *succ*:

$$succ \equiv \lambda n.\lambda f.\lambda x.nf(fx)$$

- Αξιώματα του Peano ισχύουν (κανονικές μορφές)

- Κωδικοποίηση ακεραίων: ζητάμε (αξιώματα του Peano)

$$0 \neq succ\ N$$

$$succ\ N = succ\ M \ \Rightarrow\ N = M$$

- Πολλαπλή εφαρμογή:

$$F^0 \equiv id \equiv \lambda \mathtt{x}\,.\,\mathtt{x}$$

$$F^{n+1} \equiv \lambda \mathtt{x}\,.\,F\,(\,F^n\ \ \mathtt{x}\,)$$

- Αριθμοειδή του Church:

$$n = \lambda \mathtt{f}\,.\,\lambda \mathtt{x}\,.\,\mathtt{f}^n \mathtt{x}$$

- *succ*:

$$succ \equiv \lambda n.\lambda f.\lambda x.nf(fx)$$

- Αξιώματα του Peano ισχύουν (κανονικές μορφές)

- Κωδικοποίηση ακεραίων: ζητάμε (αξιώματα του Peano)

$$0 \neq succ\ N$$

$$succ\ N = succ\ M \ \Rightarrow\ N = M$$

- Πολλαπλή εφαρμογή:

$$F^0 \equiv id \equiv \lambda \mathtt{x}.\mathtt{x}$$

$$F^{n+1} \equiv \lambda \mathtt{x}.F(F^n\ \mathtt{x})$$

- Αριθμοειδή του Church:

$$n = \lambda \mathtt{f}.\lambda \mathtt{x}.\mathtt{f}^n \mathtt{x}$$

- *succ*:

$$succ \equiv \lambda n.\lambda f.\lambda x.nf(fx)$$

- Αξιώματα του Peano ισχύουν (κανονικές μορφές)

- Κωδικοποίηση ακεραίων: ζητάμε (αξιώματα του Peano)

$$0 \neq succ\ N$$

$$succ\ N = succ\ M \;\Rightarrow\; N = M$$

- Πολλαπλή εφαρμογή:

$$F^0 \equiv id \equiv \lambda\mathtt{x}.\mathtt{x}$$

$$F^{n+1} \equiv \lambda\mathtt{x}.F(F^n\ \mathtt{x})$$

- Αριθμοειδή του Church:

$$n = \lambda\mathtt{f}.\lambda\mathtt{x}.\mathtt{f}^n\mathtt{x}$$

- *succ*:

$$succ \equiv \lambda n.\lambda f.\lambda x.nf(fx)$$

- Αξιώματα του Peano ισχύουν (κανονικές μορφές)

- Κωδικοποίηση ακεραίων: ζητάμε (αξιώματα του Peano)

$$0 \neq succ\ N$$

$$succ\ N = succ\ M \;\Rightarrow\; N = M$$

- Πολλαπλή εφαρμογή:

$$F^0 \equiv id \equiv \lambda \mathtt{x}.\mathtt{x}$$

$$F^{n+1} \equiv \lambda \mathtt{x}.F(F^n\ \mathtt{x})$$

- Αριθμοειδή του Church:

$$n = \lambda \mathtt{f}.\lambda \mathtt{x}.\mathtt{f}^n \mathtt{x}$$

- *succ*:

$$succ \equiv \lambda n.\lambda f.\lambda x.nf(fx)$$

- Αξιώματα του Peano ισχύουν (κανονικές μορφές)

- λ-όροι $\leftrightarrow$ εντολές / εκφράσεις
- λ-αφαίρεση $\leftrightarrow$ ορισμός υποπρογραμμάτων / συναρτήσεων
- Εφαρμογή (β-μετατροπή) $\leftrightarrow$ κλήση προγράμματος / εφαρμογή συνάρτησης
- Αναγωγή $\leftrightarrow$ εκτέλεση εντολών / αποτίμηση εκφράσεων
- Κανονική αποτίμηση $\leftrightarrow$ οκνηρή αποτίμηση
- Για να πετύχουμε πρόθυμη αποτίμηση επιτρέπουμε εφαρμογή μόνο σε τιμές
  - τιμή (value) = πλήρως αποτιμημένη έκφραση
  - στο λ-λογισμό: τιμή = λ-αφαίρεση (ή κανονική τιμή)

- λ-όροι ↔ εντολές / εκφράσεις
- λ-αφαίρεση ↔ ορισμός υποπρογραμμάτων / συναρτήσεων
- Εφαρμογή (β-μετατροπή) ↔ κλήση προγράμματος / εφαρμογή συνάρτησης
- Αναγωγή ↔ εκτέλεση εντολών / αποτίμηση εκφράσεων
- Κανονική αποτίμηση ↔ οκνηρή αποτίμηση
- Για να πετύχουμε πρόθυμη αποτίμηση επιτρέπουμε εφαρμογή μόνο σε τιμές
  - τιμή (value) = πλήρως αποτιμημένη έκφραση
  - στο λ-λογισμό: τιμή = λ-αφαίρεση (ή κανονική τιμή)

- λ-όροι $\leftrightarrow$ εντολές / εκφράσεις
- λ-αφαίρεση $\leftrightarrow$ ορισμός υποπρογραμμάτων / συναρτήσεων
- Εφαρμογή (β-μετατροπή) $\leftrightarrow$ κλήση προγράμματος / εφαρμογή συνάρτησης
- Αναγωγή $\leftrightarrow$ εκτέλεση εντολών / αποτίμηση εκφράσεων
- Κανονική αποτίμηση $\leftrightarrow$ οκνηρή αποτίμηση
- Για να πετύχουμε πρόθυμη αποτίμηση επιτρέπουμε εφαρμογή μόνο σε τιμές
  - τιμή (value) = πλήρως αποτιμημένη έκφραση
  - στο λ-λογισμό: τιμή = λ-αφαίρεση (ή κανονική τιμή)

- λ-όροι $\leftrightarrow$ εντολές / εκφράσεις
- λ-αφαίρεση $\leftrightarrow$ ορισμός υποπρογραμμάτων / συναρτήσεων
- Εφαρμογή (β-μετατροπή) $\leftrightarrow$ κλήση προγράμματος / εφαρμογή συνάρτησης
- Αναγωγή $\leftrightarrow$ εκτέλεση εντολών / αποτίμηση εκφράσεων
- Κανονική αποτίμηση $\leftrightarrow$ οκνηρή αποτίμηση
- Για να πετύχουμε πρόθυμη αποτίμηση επιτρέπουμε εφαρμογή μόνο σε τιμές
  - τιμή (value) = πλήρως αποτιμημένη έκφραση
  - στο λ-λογισμό: τιμή = λ-αφαίρεση (ή κανονική τιμή)

- λ-όροι $\leftrightarrow$ εντολές / εκφράσεις
- λ-αφαίρεση $\leftrightarrow$ ορισμός υποπρογραμμάτων / συναρτήσεων
- Εφαρμογή (β-μετατροπή) $\leftrightarrow$ κλήση προγράμματος / εφαρμογή συνάρτησης
- Αναγωγή $\leftrightarrow$ εκτέλεση εντολών / αποτίμηση εκφράσεων
- Κανονική αποτίμηση $\leftrightarrow$ οκνηρή αποτίμηση
- Για να πετύχουμε πρόθυμη αποτίμηση επιτρέπουμε εφαρμογή μόνο σε τιμές
  - τιμή (value) = πλήρως αποτιμημένη έκφραση
  - στο λ-λογισμό: τιμή = λ-αφαίρεση (ή κανονική τιμή)

- λ-όροι $\leftrightarrow$ εντολές / εκφράσεις
- λ-αφαίρεση $\leftrightarrow$ ορισμός υποπρογραμμάτων / συναρτήσεων
- Εφαρμογή (β-μετατροπή) $\leftrightarrow$ κλήση προγράμματος / εφαρμογή συνάρτησης
- Αναγωγή $\leftrightarrow$ εκτέλεση εντολών / αποτίμηση εκφράσεων
- Κανονική αποτίμηση $\leftrightarrow$ οκνηρή αποτίμηση
- Για να πετύχουμε πρόθυμη αποτίμηση επιτρέπουμε εφαρμογή μόνο σε <u>τιμές</u>
    - <u>τιμή (value)</u> = πλήρως αποτιμημένη έκφραση
    - στο λ-λογισμό: τιμή = λ-αφαίρεση (ή κανονική τιμή)

Επισκόπηση - Ι

- Ο λ-λογισμός είναι θεμελειώδης θεωρία συναρτήσεων και πλήρες υπολογιστικό μοντέλο
    - αναπτύχθηκε το 1933 από τον Church
    - αποτέλεσε το μαθηματικό υπόβαθρο για τις συναρτησιακές γλώσσες αλλά και για τη μελέτη των γλωσσών προγραμματισμού γενικότερα
- Χρησιμοποιούμε μία μεταθεωρία με μεταμεταβλητές για να περιγράψουμε τη θεωρία του λ-λογισμού
- Οι εκφράσεις του λ-λογισμού είναι όλες συναρτήσεις
    - μεταβλητές
    - λ-αφαιρέσεις (κατασκευή συνάρτησης)
    - εφαρμογές συνάρτησης

- Ο λ-λογισμός είναι θεμελειώδης θεωρία συναρτήσεων και πλήρες υπολογιστικό μοντέλο
  - αναπτύχθηκε το 1933 από τον Church
  - αποτέλεσε το μαθηματικό υπόβαθρο για τις συναρτησιακές γλώσσες αλλά και για τη μελέτη των γλωσσών προγραμματισμού γενικότερα
- Χρησιμοποιούμε μία μεταθεωρία με μεταμεταβλητές για να περιγράψουμε τη θεωρία του λ-λογισμού
- Οι εκφράσεις του λ-λογισμού είναι όλες συναρτήσεις
  - μεταβλητές
  - λ-αφαιρέσεις (κατασκευή συνάρτησης)
  - εφαρμογές συνάρτησης

Επισκόπηση - Ι

- Ο λ-λογισμός είναι θεμελειώδης θεωρία συναρτήσεων και πλήρες υπολογιστικό μοντέλο
    - αναπτύχθηκε το 1933 από τον Church
    - αποτέλεσε το μαθηματικό υπόβαθρο για τις συναρτησιακές γλώσσες αλλά και για τη μελέτη των γλωσσών προγραμματισμού γενικότερα
- Χρησιμοποιούμε μία μεταθεωρία με μεταμεταβλητές για να περιγράψουμε τη θεωρία του λ-λογισμού
- Οι εκφράσεις του λ-λογισμού είναι όλες συναρτήσεις
    - μεταβλητές
    - λ-αφαιρέσεις (κατασκευή συνάρτησης)
    - εφαρμογές συνάρτησης

Επισκόπηση - II

- Οι εμφανίσεις μεταβλητών χωρίζονται σε <u>δεσμευμένες</u> (από μία λ-αφαίρεση) ή <u>ελεύθερες</u>
  - μία έκφραση εξαρτάται από τις ελεύθερες μεταβλητές της
  - τα ονόματα δεσμευμένων μεταβλητών μπορούν να αλλάξουν κατά βούληση
  - μια μετατροπή δε μπορεί να δημιουργήσει καινούριες δεσμεύσεις ή να αντικαταστήσει δεσμευμένες μεταβλητές
- Μετατροπές:
  - α-μετατροπή: αλλαγή ονόματος δεσμεύουσας μεταβλητής
  - β-μετατροπή: εφαρμογή συνάρτησης
  - η-μετατροπή: ισοδυναμία συναρτήσεων
- Αναγωγή: μηδέν ή περισσότερες μετατροπές
- Ισότητα: σχέση ισοδυναμίας που προκύπτει από την αναγωγή

Επισκόπηση - II

- Οι εμφανίσεις μεταβλητών χωρίζονται σε <u>δεσμευμένες</u> (από μία λ-αφαίρεση) ή <u>ελεύθερες</u>
  - μία έκφραση εξαρτάται από τις ελεύθερες μεταβλητές της
  - τα ονόματα δεσμευμένων μεταβλητών μπορούν να αλλάξουν κατά βούληση
  - μια μετατροπή δε μπορεί να δημιουργήσει καινούριες δεσμεύσεις ή να αντικαταστήσει δεσμευμένες μεταβλητές
- Μετατροπές:
  - α-μετατροπή: αλλαγή ονόματος δεσμεύουσας μεταβλητής
  - β-μετατροπή: εφαρμογή συνάρτησης
  - η-μετατροπή: ισοδυναμία συναρτήσεων
- Αναγωγή: μηδέν ή περισσότερες μετατροπές
- Ισότητα: σχέση ισοδυναμίας που προκύπτει από την αναγωγή

Επισκόπηση - II

- Οι εμφανίσεις μεταβλητών χωρίζονται σε <u>δεσμευμένες</u> (από μία λ-αφαίρεση) ή <u>ελεύθερες</u>
  - μία έκφραση εξαρτάται από τις ελεύθερες μεταβλητές της
  - τα ονόματα δεσμευμένων μεταβλητών μπορούν να αλλάξουν κατά βούληση
  - μια μετατροπή δε μπορεί να δημιουργήσει καινούριες δεσμεύσεις ή να αντικαταστήσει δεσμευμένες μεταβλητές
- Μετατροπές:
  - α-μετατροπή: αλλαγή ονόματος δεσμεύουσας μεταβλητής
  - β-μετατροπή: εφαρμογή συνάρτησης
  - η-μετατροπή: ισοδυναμία συναρτήσεων
- Αναγωγή: μηδέν ή περισσότερες μετατροπές
- Ισότητα: σχέση ισοδυναμίας που προκύπτει από την αναγωγή

Επισκόπηση - II

- Οι εμφανίσεις μεταβλητών χωρίζονται σε <u>δεσμευμένες</u> (από μία λ-αφαίρεση) ή <u>ελεύθερες</u>
  - μία έκφραση εξαρτάται από τις ελεύθερες μεταβλητές της
  - τα ονόματα δεσμευμένων μεταβλητών μπορούν να αλλάξουν κατά βούληση
  - μια μετατροπή δε μπορεί να δημιουργήσει καινούριες δεσμεύσεις ή να αντικαταστήσει δεσμευμένες μεταβλητές
- Μετατροπές:
  - α-μετατροπή: αλλαγή ονόματος δεσμεύουσας μεταβλητής
  - β-μετατροπή: εφαρμογή συνάρτησης
  - η-μετατροπή: ισοδυναμία συναρτήσεων
- Αναγωγή: μηδέν ή περισσότερες μετατροπές
- Ισότητα: σχέση ισοδυναμίας που προκύπτει από την αναγωγή

Επισκόπηση - III

- Κανονική μορφή: δεν περιέχει άλλα $\beta\eta$-redexes
  - (αναγωγή = υπολογισμός = αποτίμηση = κανονικοποίηση)
- Κάθε λ-όρος έχει το πολύ μία κανονική μορφή
  - δεν βρίσκουν όλες οι στρατηγικές αναγωγής την κανονική μορφή ενός όρου
- Κανονική αναγωγή: επιλέγουμε πάντα το αριστερότερο λ
  - βρίσκει την κανονική μορφή, εάν υπάρχει
- Κάθε όρος έχει σταθερό σημείο ($X = F\ X$). Ο τελεστής Y επιστρέφει ένα σταθερό σημείο μίας συνάρτησης
- λ-λογισμός μπορεί να εκφράσει booleans, αριθμούς και όλους τους όρους υπολογισμού

Επισκόπηση - ΙΙΙ

- Κανονική μορφή: δεν περιέχει άλλα $\beta\eta$-redexes
  - (αναγωγή = υπολογισμός = αποτίμηση = κανονικοποίηση)
- Κάθε λ-όρος έχει το πολύ μία κανονική μορφή
  - δεν βρίσκουν όλες οι στρατηγικές αναγωγής την κανονική μορφή ενός όρου
- Κανονική αναγωγή: επιλέγουμε πάντα το αριστερότερο λ
  - βρίσκει την κανονική μορφή, εάν υπάρχει
- Κάθε όρος έχει σταθερό σημείο ($X = F\ X$). Ο τελεστής Y επιστρέφει ένα σταθερό σημείο μίας συνάρτησης
- λ-λογισμός μπορεί να εκφράσει booleans, αριθμούς και όλους τους όρους υπολογισμού

Επισκόπηση - III

- Κανονική μορφή: δεν περιέχει άλλα $\beta\eta$-redexes
  - (αναγωγή = υπολογισμός = αποτίμηση = κανονικοποίηση)
- Κάθε λ-όρος έχει το πολύ μία κανονική μορφή
  - δεν βρίσκουν όλες οι στρατηγικές αναγωγής την κανονική μορφή ενός όρου
- Κανονική αναγωγή: επιλέγουμε πάντα το αριστερότερο λ
  - βρίσκει την κανονική μορφή, εάν υπάρχει
- Κάθε όρος έχει σταθερό σημείο ($X = F\ X$). Ο τελεστής Y επιστρέφει ένα σταθερό σημείο μίας συνάρτησης
- λ-λογισμός μπορεί να εκφράσει booleans, αριθμούς και όλους τους όρους υπολογισμού

Επισκόπηση - III

- Κανονική μορφή: δεν περιέχει άλλα $\beta\eta$-redexes
  - (αναγωγή = υπολογισμός = αποτίμηση = κανονικοποίηση)
- Κάθε λ-όρος έχει το πολύ μία κανονική μορφή
  - δεν βρίσκουν όλες οι στρατηγικές αναγωγής την κανονική μορφή ενός όρου
- Κανονική αναγωγή: επιλέγουμε πάντα το αριστερότερο λ
  - βρίσκει την κανονική μορφή, εάν υπάρχει
- Κάθε όρος έχει σταθερό σημείο ($X = F\,X$). Ο τελεστής Y επιστρέφει ένα σταθερό σημείο μίας συνάρτησης
- λ-λογισμός μπορεί να εκφράσει booleans, αριθμούς και όλους τους όρους υπολογισμού

Επισκόπηση - III

- Κανονική μορφή: δεν περιέχει άλλα $\beta\eta$-redexes
  - (αναγωγή = υπολογισμός = αποτίμηση = κανονικοποίηση)
- Κάθε λ-όρος έχει το πολύ μία κανονική μορφή
  - δεν βρίσκουν όλες οι στρατηγικές αναγωγής την κανονική μορφή ενός όρου
- Κανονική αναγωγή: επιλέγουμε πάντα το αριστερότερο λ
  - βρίσκει την κανονική μορφή, εάν υπάρχει
- Κάθε όρος έχει σταθερό σημείο ($X = F\ X$). Ο τελεστής Y επιστρέφει ένα σταθερό σημείο μίας συνάρτησης
- λ-λογισμός μπορεί να εκφράσει booleans, αριθμούς και όλους τους όρους υπολογισμού