Lamda Calculus

Optional Subtitle

Nikos Zarifis[1]

[1]ECCE
National Technical University of Athens

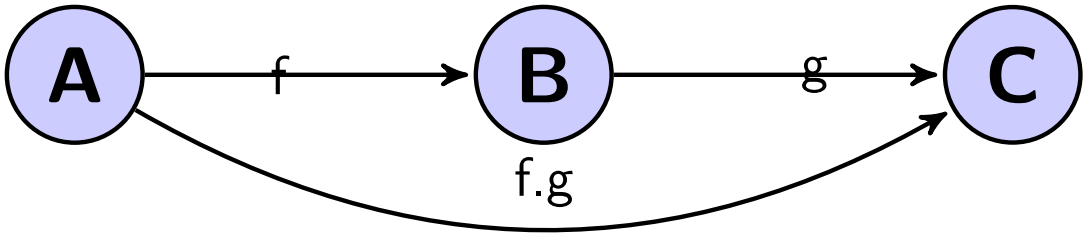
5 10, 2015

Outline

Outline

Ορισμός Κατηγορίας

- Ένα σύνολο απο αντικείμενα. Συμβολίζουμε ως : obj(C).
- Ένα σύνολο απο μορφισμούς. Πχ $f : A \to B$. Συμβολίζεται $hom_C(a, b)$.
- Την πράξη σύνθεση. Όπως γίνεται η Σύνθεση συναρτήσεων.



Συμβολίζουμε την σύνθεση με την .(dot). Κι έχουμε 2 αξιόματα:Προσετεριστική κι την ύπαρξη ουδέτερου στοιχίου.

Functors

Έστω ότι έχουμε 2 κατηγορίες, μπορούμε να όρισουμε μια συνάρτηση απο τα αντικείμενα της κατηγορίας A στην Κατηγορία B.

Example

Έστω $F : C_A -> C_B, a \in obj(A) \rightarrow F(a) \in obj(B)$

Όπως επίσης αν έχω εναν μορφισμό $f : a_A -> b_A$
τότε $F(f) : F(a_A) -> F(b_A)$.
Για την σύνθεση ισχυεί: $F(f.g) = F(f).F(g)$
και ως προς το ουδέτερο μορφισμό $F(id_A) = id_{F(A)}$

Αρχικά - Τελικά Αντικείμενα

Έχουμε επίσεις:

- **Intial Obj** : Αρχικό Αντικείμενο : Για κάθε άλλο αντικείμενο τις κατηγορίας υπάρχει ακριβώς ενας μορφισμός που να πηγαίνει σε αυτο.

Αρχικά - Τελικά Αντικείμενα

Έχουμε επίσεις:

- **Intial Obj** : Αρχικό Αντικείμενο : Για κάθε άλλο αντικείμενο τις κατηγορίας υπάρχει ακριβώς ενας μορφισμός που να πηγαίνει σε αυτο.
- **Terminal obj** : Τελικό Αντικείμενο: Είναι το δυικό του αρχικού, δηλαδή για κάθε αντικείμενο υπάρχει ακριβώς ένανς μορφισμός που να πηγαίνει στο τελικό

Δυική κατηφορία

Σε κάθε κατηγορία μπορούμε να όρισουμε μια κατηγορία όπου κάθε μορφισμός που υπάρχει στον κανόνικο , υπάρχει στο δυικό με αντιστρεμένα τα άκρα. Το αρχικό αντικείμενο γίνεται τελικό κι αντίστροφα.
Ισχυεί: $G = f.g \rightarrow G^{op} = g^{op}.f^{op}$.

Παραδείγματα

- Αν πάρουμε ως αντικείμενα το N κι όλους τους 1-1 μορφισμούς. Πχ : f(n)=n+1 . Δυικό το f(n)=n-1 .Αρχικό το 0, Τελικό δεν έχει.

Δυική κατηφορία

Σε κάθε κατηγορία μπορούμε να όρισουμε μια κατηγορία όπου κάθε μορφισμός που υπάρχει στον κανόνικο , υπάρχει στο δυικό με αντιστρεμένα τα άκρα. Το αρχικό αντικείμενο γίνεται τελικό κι αντίστροφα.
Ισχυεί: $G = f.g \rightarrow G^{op} = g^{op}.f^{op}$.
Παραδείγματα

- Αν πάρουμε ως αντικείμενα το N κι όλους τους 1-1 μορφισμούς. Πχ : f(n)=n+1 . Δυικό το f(n)=n-1 .Αρχικό το 0, Τελικό δεν έχει.
- Στους Δυανισμάτικους χώρους αν πάρουμε ως αντικείμενα όλα τα δυανίσματα, κι ως μορφισμούς όλους τους μορφισμούς τάξης n. Πχ Ο πίνακας A κι ο δυικός που είναι ο A^{-1}.

Cartesian Closed Categories

Έστω μια κατηγορια C λέμε ότι είναι Cartesian Closed (CCC) ανν:

- Έχει τερματικό αντικείμενο.

Cartesian Closed Categories

Έστω μια κατηγορια C λέμε ότι είναι Cartesian Closed (CCC) ανν:

- Έχει τερματικό αντικείμενο.
- Για κάθε 2 αντικείμενα στην C , έχουμε κι το X*Y στην C.

Cartesian Closed Categories

Έστω μια κατηγορια C λέμε ότι είναι Cartesian Closed (CCC) ανν:

- Έχει τερματικό αντικείμενο.
- Για κάθε 2 αντικείμενα στην C , έχουμε κι το X*Y στην C.
- Για κάθε 2 αντικείμενα στην C , έχουμε κι το X^Y στην C.

Products: Θα λέμε ότι ένα αντικείμενο είναι product (γινόμενο) 2 αντικείμενον ανν όριζουμε 2 μορφιφούς προβολών του X, έστω $X = X_1 * X_2$

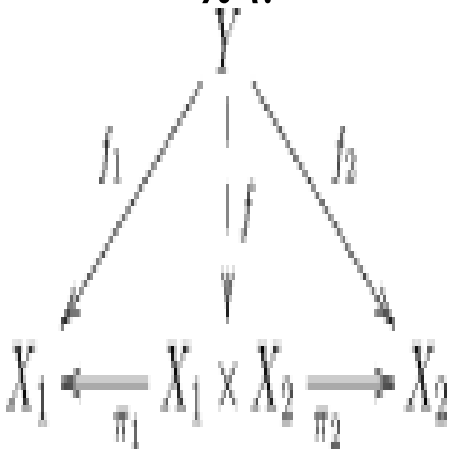
- $\pi_1 : X \to X_1$

Products: Θα λέμε ότι ένα αντικείμενο είναι product (γινόμενο) 2 αντικείμενον ανν όριζουμε 2 μορφιφούς προβολών του X, έστω $X = X_1 * X_2$

- $\pi_1 : X \to X_1$
- $\pi_2 : X \to X_1 2$

Products: Θα λέμε ότι ένα αντικείμενο είναι product (γινόμενο) 2 αντικείμενον ανν όριζουμε 2 μορφιφούς προβολών του X, έστω $X = X_1 * X_2$

- $\pi_1 : X \to X_1$
- $\pi_2 : X \to X_1 2$
  Και επίσεις έστω οι μορφισμοί $f : Y \to X$ και $f_1 : Y \to X_1$, $f_2 : Y \to X_2$ έτσι ώστε: $f(Y) = f_1(Y) * f_2(Y)$.
  Κι σε σχήμα:



Expomential Object: Αν έχουμε 2 αντικείμενα στην κατήγορία όριζουμε ως X^Y το σύνολο των μορφισμων απο το Y, X .
Κι θα ορίσουμε με την σείρα μας 2 πολύ συμαντικές συναρτήσεις:
Την eval $eval_{A,B} : B^A * A \to B$.
Όπως φαίνεται κι απο τον τύπο ισχυεί ότι $eval_{A,B}(f, A) = f(A)$.
Και την curry έτσι ωστε αν έχουμε έναν μορφισμό g τότε $curry_c(g) : C \to B^A$ και ισχυεί οι $eval_{A,B}.(curry_c(g) * id_A) = g$.

Outline

Δομή στις CCC

Αντικέιμενα-> τύπους
Όρους-> μορφισμούς
Έτσι ορίζουμε μια δομή πάνω στις CCC.(Θεωρούμε τις κατήγοριες ανάμεσα σε sets)

- Ορίζουμε το τερματικό στοιχείο (συμβ. 1) να έχει τύπο nil (void,() κτλπ) ο μοναδιαίος τύπος.

Δομή στις CCC

Αντικέιμενα-> τύπους
Όρους-> μορφισμούς
Έτσι ορίζουμε μια δομή πάνω στις CCC.(Θεωρούμε τις κατήγοριες ανάμεσα σε sets)

- Ορίζουμε το τερματικό στοιχείο (συμβ. 1) να έχει τύπο nil (void,() κτλπ) ο μοναδιαίος τύπος.
- Όριζουμε μια συνάρτηση F ως εξής:
  Για κάθε αρχικό τύπο t F(t) είναι ένα αντικείμενο.
  Για $F(t \to t_1) = F(t)^{F(t_1)}$
  Για $F(t * t_1) = F(t) * F(t_1)$

Δομή στις CCC

Αντικέιμενα-> τύπους
Όρους-> μορφισμούς
Έτσι ορίζουμε μια δομή πάνω στις CCC.(Θεωρούμε τις κατήγοριες ανάμεσα σε sets)

- Ορίζουμε το τερματικό στοιχείο (συμβ. 1) να έχει τύπο nil (void,() κτλπ) ο μοναδιαίος τύπος.
- Όριζουμε μια συνάρτηση F ως εξής:
  Για κάθε αρχικό τύπο t F(t) είναι ένα αντικείμενο.
  Για $F(t \to t_1) = F(t)^{F(t_1)}$
  Για $F(t * t_1) = F(t) * F(t_1)$
- Έστω $H = x_1 : t_1 \ldots, x_n : t_n$ τότε:
  $F() = 1$
  $F(x : t) = 1 * F(t)$
  $F(H) = F(x_1 : t_1 \ldots, x_{n-1} : t_{n-1}) * F(t_n)$

Δομή στις CCC

Αντικέιμενα-> τύπους
Όρους-> μορφισμούς
Έτσι ορίζουμε μια δομή πάνω στις CCC.(Θεωρούμε τις κατήγοριες ανάμεσα σε sets)

- Ορίζουμε το τερματικό στοιχείο (συμβ. 1) να έχει τύπο nil (void,() κτλπ) ο μοναδιαίος τύπος.
- Όριζουμε μια συνάρτηση F ως εξής:
  Για κάθε αρχικό τύπο t F(t) είναι ένα αντικείμενο.
  Για $F(t \to t_1) = F(t)^{F(t_1)}$
  Για $F(t * t_1) = F(t) * F(t_1)$
- Έστω $H = x_1 : t_1 \ldots, x_n : t_n$ τότε:
  $F() = 1$
  $F(x : t) = 1 * F(t)$
  $F(H) = F(x_1 : t_1 \ldots, x_{n-1} : t_{n-1}) * F(t_n)$
  F συνάρτηση αποτίμησεις

- Αν $H \vdash M : t$ τότε $F(H \vdash M : t) : F(H) \to F(t)$.

- Αν $H \vdash M : t$ τότε $F(H \vdash M : t) : F(H) \to F(t)$.
- Υπάρχει σημείο $F(c) : 1 \to F(\Sigma)$.

Επεκτήνοντας την ερμηνία μας έχουμε:

- $F(H \vdash c : t) = F(c).t_A$ όπου t_A είναι ο μοναδικός μορφισμός από το A στο τελικό στοιχίο 1 (Nil).
- $F(H \vdash x_i : t_i) = \pi_i$

Επεκτήνοντας την ερμηνία μας έχουμε:

- $F(H \vdash c : t) = F(c).t_A$ όπου t_A είναι ο μοναδικός μορφισμός από το A στο τελικό στοιχίο 1 (Nil).
- $F(H \vdash x_i : t_i) = \pi_i$
- Στην αφαίρεση: $F(H \vdash \lambda x : u.N\, u \to v)$ $= curry(F(H, x : u \vdash N : v))$

Επεκτήνοντας την ερμηνία μας έχουμε:

- $F(H \vdash c : t) = F(c).t_A$ όπου t_A είναι ο μοναδικός μορφισμός από το A στο τελικό στοιχίο 1 (Nil).
- $F(H \vdash x_i : t_i) = \pi_i$
- Στην αφαίρεση: $F(H \vdash \lambda x : u.N\, u \to v)$
  $= curry(F(H, x : u \vdash N : v))$
- Εφαρμογή:
  $F(H \vdash LN : s) = eval. < F(H \vdash L : t \to s), F(H \vdash N : t) >$

Επεκτήνοντας την ερμηνία μας έχουμε:

- $F(H \vdash c : t) = F(c).t_A$ όπου t_A είναι ο μοναδικός μορφισμός από το A στο τελικό στοιχίο 1 (Nil).
- $F(H \vdash x_i : t_i) = \pi_i$
- Στην αφαίρεση: $F(H \vdash \lambda x : u.N\, u \to v)$
  $= curry(F(H, x : u \vdash N : v))$
- Εφαρμογή:
  $F(H \vdash LN : s) = eval. < F(H \vdash L : t \to s), F(H \vdash N : t) >$
- Ζευγή $F(H \vdash (L, N) : s * t) =< F(H \vdash L : s), F(H \vdash N : t) >$

Επεκτήνοντας την ερμηνία μας έχουμε:

- $F(H \vdash c : t) = F(c).t_A$ όπου t_A είναι ο μοναδικός μορφισμός από το A στο τελικό στοιχίο 1 (Nil).
- $F(H \vdash x_i : t_i) = \pi_i$
- Στην αφαίρεση: $F(H \vdash \lambda x : u.N\, u \to v)$
  $= curry(F(H, x : u \vdash N : v))$
- Εφαρμογή:
  $F(H \vdash LN : s) = eval. < F(H \vdash L : t \to s), F(H \vdash N : t) >$
- Ζευγή $F(H \vdash (L, N) : s * t) =< F(H \vdash L : s), F(H \vdash N : t) >$
- first $F(H \vdash fst(N) : s) = fst.F(H \vdash N : s * t)$

Επεκτήνοντας την ερμηνία μας έχουμε:

- $F(H \vdash c : t) = F(c).t_A$ όπου t_A είναι ο μοναδικός μορφισμός από το A στο τελικό στοιχίο 1 (Nil).
- $F(H \vdash x_i : t_i) = \pi_i$
- Στην αφαίρεση: $F(H \vdash \lambda x : u.N\, u \to v)$
  $= curry(F(H, x : u \vdash N : v))$
- Εφαρμογή:
  $F(H \vdash LN : s) = eval. < F(H \vdash L : t \to s), F(H \vdash N : t) >$
- Ζευγή $F(H \vdash (L, N) : s * t) =< F(H \vdash L : s), F(H \vdash N : t) >$
- first $F(H \vdash fst(N) : s) = fst.F(H \vdash N : s * t)$
  second $F(H \vdash snd(N) : s) = snd.F(H \vdash N : s * t)$

Soundness of CCC-models

Ορισμός

Έχοντας μια δομή S σε μια CCC έστω C αν έχουμε την εξίσωση $H \vdash M_1 = M_2 : t$ τότε λέμε ότι το S ικανοποιεί την εξίσωση αν $F(H \vdash M_1 : t)$ και $F(H \vdash M_2 : t)$ είναι οι ίδιοι μορφισμοι στο C.

Και το γράφουμε : $S \models H \vdash M_1 = M_2 : t$.
Και λέμε ότι το S είναι μοντέλο της **simple typed lamda calculus** θεωρίας $T = (\Sigma, Ax)$ αν το S ικανοποιεί όλες τις εξισώσεις στο Ax δηλαδή
$S \models Ax$.

Soundness

Soundness for CCC-Models

Αν C είναι μια CCC και $T = (\Sigma, Ax)$ και S ένα μοντέλο της T στο C τ. ΑΝ $T \vdash (H \vdash M = N : t)$ τότε $S \models H \vdash M = N : t$

Proof.

- α-ισοδυναμία είναι sound στο μοντελο μας.
- β-ισοδυναμία είναι sound, Απόδειξη με χρήση λήματος αντικατάστασης.
- η-ισοδυναμία είναι sound.
  $F(H \vdash \lambda x : a.Mx : t) = F(H \vdash M : a \to t)$

Completeness

Completeness

Αν $H \vdash M : a$ και $H \vdash N : a$ τότε υπάρχει μια CCC(F) έτσι ώστε αν : $F(H \vdash M : a) = F(H \vdash N : a)$ τότε $H \vdash M = N : a$.

Curry-Howard-Lambek isomorphism

Lambek

Ισομορφισμός μεταξύ typed-lambda-calculis - intuitionistic logic - CCC.

Μορφισμοί ως όροι κι αποδείξεις, αντικείμενα ως τύποι.

Κατασκευή C(L)

Θα κατασκεύασουμε έναν functor(C) από μια typed-λ-calculus L σε μια CCC.

- Αντικείμενα στην C(L) είναι τύποι της L
- Μορφισφοί είναι σαν συναρτήσεις στην L . πχ. $x \to f(x)$
- Έχουμε id μορφισμό που είναι ο $x \to x$
- Έχουμε την σύνθεσή , αντίστοιχα με την εφάρογή στον λαμδα .
- Κι φυσικά η δομή της οριζεταί όπως είδαμε προηγουμένος .

CCC and $\lambda^{x,\to}$

$\lambda^{x,\to}$	**CCC**
unit type	τερματικό αντικίμενο unit
product type $a * b$	product $A * B$
Συναρτήσεις τύπων $a \to b$	expomental: $A \to B$

Κανόνες

Ορίζουμε τους κανόνες:

$$\frac{f : a \vdash b \qquad g : b \vdash c}{f.g : -a \vdash c}$$

Unit:

$$\frac{}{c\ \star : a \vdash \mathit{Void}}$$

Cartesian Product:

$$\frac{f : a \vdash b \qquad g : a \vdash c}{f * g : a \vdash b * c}$$

Προβολές:

$$\frac{}{\pi_1 : a * b \vdash a}$$

$$\frac{c}{\pi_2 : a * b \vdash b}$$

Curry:

$$\frac{f : a*b \vdash c}{\mathit{curry}\ f : a \vdash b \to c}$$

Continue

Εφαρμογή:

$$\frac{}{c \;\; eval : (a \to b) * a \vdash b}$$

Theorem

Έτσι λοιπόν έχουμε ότι υπάρχει μορφισμός f έτσι ώστε $f : a_1 * a_2 * \ldots a_3 \vdash b$ *ανν το* $a_1, a_2, \ldots, a_n \vdash b$ *στην ιο υντουζιανή:λογική.*

Ισομορφισμός

Theorem

Η λ -Calculus κι οι CCC είναι ισομορφικές.Συγκεκριμένα CL≅ id και LC ≅ id.

Το ευθή αποδικνίεται εύκολα με επαγωγίκο όρισμο ενός 1-1 μορφισμού.Το αντίστροφο αποδικνίεται και, με χρήση του παραπάνω functor κι με χρήση των natural transformations.

Αποτελέσματα

Αποτέλεσμα CHL:

- **Λογική**: υπολογιστίκο περιεχόμενο αποδίξεων

Αποτελέσματα

Αποτέλεσμα CHL:

- **Λογική**: υπολογιστίκο περιεχόμενο αποδίξεων
- **CS**: Θεμέλεια type-system και συναρτησιακού προγραμματισμού. Αν σκεφτούμε ότι η Haskell βασίζεται στην Category Theory

Αποτελέσματα

Αποτέλεσμα CHL:

- **Λογική**: υπολογιστίκο περιεχόμενο αποδίξεων
- **CS**: Θεμέλεια type-system και συναρτησιακού προγραμματισμού. Αν σκεφτούμε ότι η Haskell βασίζεται στην Category Theory
- **Category Theory**:Μια σύνδεση στις γλώσσες κι στην λογική. Λαμβδα λογισμός ως γλώσσα για τις CCC. Λαμδα λογισμός για υπολογισμούς σε θέματα περί CCC και αντιστρόφος.

Αποτελέσματα

Αποτέλεσμα CHL:

- **Λογική**: υπολογιστίκο περιεχόμενο αποδίξεων
- **CS**: Θεμέλεια type-system και συναρτησιακού προγραμματισμού. Αν σκεφτούμε ότι η Haskell βασίζεται στην Category Theory
- **Category Theory**:Μια σύνδεση στις γλώσσες κι στην λογική. Λαμβδα λογισμός ως γλώσσα για τις CCC. Λαμδα λογισμός για υπολογισμούς σε θέματα περί CCC και αντιστρόφος.
- **Monads**


Curry-Howard-Lambek Correspondence Subashis Chakraborty


Category Theory and the Simply Typed λ-Calculus Alfio Martini