Θεωρία Γλωσσών Προγραμματισμού

Νίκος Παπασπύρου
nickie@softlab.ntua.gr

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχ. και Μηχ. Υπολογιστών
Εργαστήριο Τεχνολογίας Λογισμικού
Πολυτεχνειούπολη, 15780 Ζωγράφου.

Περιεχόμενα

Μαθηματική λογική (i)

- Σχέση μεταξύ λογικής και επιστήμης των υπολογιστών
  - Θεμελίωση
    - μοντέλο για υπολογισμούς
    - αντιστοιχία Curry-Howard
  - Σύστημα συλλογισμών
    - προδιαγραφές λογισμικού
    - τεχνητή νοημοσύνη
    - κ.λπ.

Μαθηματική λογική (ii)

- Πολλά είδη λογικής
  - Κλασσική (classical) λογική
  - Διαισθητική (intuitionistic) ή κατασκευαστική (constructive) λογική
  - Γραμμική (linear) λογική
  - Χρονική (temporal) και τροπική (modal) λογική

Μαθηματική λογική (iii)

- Παράδειγμα εκφραστικότητας λογικών

(1) Ο Κώστας δεν μπορεί να αρχίσει μεταπτυχιακά αν δεν πάρει πτυχίο
(2) Ο Κώστας δεν έχει ακόμα πάρει πτυχίο
$\therefore$ (3) Ο Κώστας θα πάρει πτυχίο ή δε θα κάνει ποτέ μεταπτυχιακά

(1) $\forall t.\ (t > n \wedge M(t) \rightarrow \exists t'.\ (t' < t \wedge \Pi(t')))$
(2) $\neg(\exists t.\ (t \leq n \wedge \Pi(t)))$
$\therefore$ (3) $\exists t.\ (t > n \wedge \Pi(t)) \vee \neg(\exists t.\ (t > n \wedge M(t)))$

Διάφορες υποθέσεις...

Μαθηματική λογική (iv)

- Παράδειγμα (συνέχεια)

(1) Ο Κώστας δεν μπορεί να αρχίσει μεταπτυχιακά αν δεν πάρει πτυχίο
(2) Ο Κώστας δεν έχει ακόμα πάρει πτυχίο
$\therefore$ (3) Ο Κώστας θα πάρει πτυχίο ή δε θα κάνει ποτέ μεταπτυχιακά

(1) $\Box(M \rightarrow \odot\Pi)$
(2) $\neg(\Pi \vee \odot\Pi)$
$\therefore$ (3) $\Diamond\Pi \vee \neg\Diamond M$

Χρονική λογική
$\Box$: πάντα στο μέλλον
$\Diamond$: κάποτε στο μέλλον
$\odot$: κάποτε στο παρελθόν

Προτασιακές γλώσσες

- Ζεύγη (P, O), όπου
  - P : σύνολο ατομικών προτάσεων
  - O : σύνολο τελεστών, κάθε ένας έχει συγκεκριμένο αριθμό τελουμένων
- Παράδειγμα: προτασιακός λογισμός

$$O = \{ \wedge, \vee, \neg, \rightarrow, \equiv, \mathit{true}, \mathit{false} \}$$

Κατηγορηματικές γλώσσες (i)

- Πεντάδες (P, T, V, O, Q), όπου
  - P : σύνολο κατηγορημάτων, κάθε ένα έχει συγκεκριμένο αριθμό ορισμάτων
  - $V \subseteq T$: σύνολα μεταβλητών και όρων
  - O : σύνολο τελεστών, κάθε ένας έχει συγκεκριμένο αριθμό τελουμένων
  - Q : σύνολο ποσοδεικτών
- Παράδειγμα: κατηγορηματικός λογισμός

$$O = \{ \wedge, \vee, \neg, \rightarrow, \equiv, \mathit{true}, \mathit{false} \}$$
$$Q = \{ \forall, \exists \}$$

Κατηγορηματικές γλώσσες (ii)

- Προτάσεις μιας κατηγορηματικής γλώσσας $\mathbf{L}$
  - Αν $p \in P$ με αριθμό ορισμάτων n και $t_1, \ldots, t_n \in T$, τότε $p(t_1, \ldots, t_n) \in \mathbf{L}$
  - Αν $o \in O$ με αριθμό τελουμένων n και $A_1, \ldots, A_N \in \mathbf{L}$, τότε $o(A_1, \ldots, A_n) \in \mathbf{L}$
  - Αν $q \in Q$, $x \in V$ και $A \in \mathbf{L}$, τότε $q\, x.A \in \mathbf{L}$
- Παράδειγμα από τον κατηγορηματικό λογισμό

$$\forall x.\ (\mathrm{prime}(x) \rightarrow \exists y.\ (y > x \wedge \mathrm{prime}(y)))$$

Συστήματα τιμών (i)

- Προτασιακών γλωσσών: (M, D, F)
  - M: σύνολο τιμών αληθείας, $|M| \geq 2$
  - D: σύνολο αληθών τιμών, $D \subset M$, $D \neq \varnothing$
  - F: σημασία τελεστών

$$F = \{ f_o \mid o \in O \} \text{ όπου } f_o : M^{n_o} \rightarrow M$$

- Παράδειγμα: κλασσική προτασιακή λογική

$M = \{ t, f \}$ $\quad D = \{ t \}$ $\quad F$ γνωστό

Συστήματα τιμών (ii)

- Ανάθεση $a : P \rightarrow M$
- Ερμηνεία m
  - ένα σύστημα τιμών (M, D, F)
  - μια ανάθεση a
- Για κάθε πρόταση $A \in \mathbf{L}$ ορίζουμε $[\![A]\!]_m : M$

$$\begin{aligned} [\![p]\!]_m &= a(p) \\ [\![o(A_1, \ldots, A_n)]\!]_m &= f_o([\![A_1]\!]_m, \ldots, [\![A_n]\!]_m) \end{aligned}$$

- Γράφουμε $m \Vdash A$ όταν $[\![A]\!]_m \in D$

Συστήματα τιμών (iii)

- Κατηγορηματικών γλωσσών: (M, D, F, G)
  - M, D, F: όπως πριν
  - G: σημασία ποσοδεικτών

$$G = \{ g_q \mid q \in Q \} \text{ όπου } g_q : \mathcal{P}(M) \rightarrow M$$

- Παράδειγμα: κλασσική κατηγορηματική λογική

$$G = \{ g_\forall, g_\exists \}$$

$$g_\forall(S) = \begin{cases} f, & \text{αν } f \in S \\ t, & \text{αλλιώς} \end{cases} \qquad g_\exists(S) = \begin{cases} t, & \text{αν } t \in S \\ f, & \text{αλλιώς} \end{cases}$$

Συστήματα τιμών (iv)

- Ανάθεση (a, I)
  - αν $t \in T$ τότε $a(t) \in I$
  - αν $p \in P$ και δέχεται n ορίσματα, τότε $a(p) : I^n \to M$
- Γράφουμε $a \sim_x a'$ όταν οι αναθέσεις a και a' συμφωνούν σε όλες τις τιμές, εκτός πιθανώς από αυτή που αναθέτουν στη μεταβλητή x.
  - Αν $t \in T$ και η μεταβλητή x δεν είναι ελεύθερη στο t, τότε $a(t) = a'(t)$
  - Αν $p \in P$, τότε $a(p) = a'(p)$

Συστήματα τιμών (v)

- Ερμηνεία m
  - ένα σύστημα τιμών (M, D, F, G)
  - μια ανάθεση (a, I)
- Γράφουμε $m \sim_x m'$ όταν για τις αναθέσεις a και a' των m και m' αντίστοιχα ισχύει $a \sim_x a'$.
- Για κάθε πρόταση $A \in \mathbf{L}$ ορίζουμε $[\![A]\!]_m : M$

$$\begin{aligned}
[\![p(t_1, \ldots, t_n)]\!]_m &= a(p)(a(t_1), \ldots, a(t_n)) \\
[\![o(A_1, \ldots, A_n)]\!]_m &= f_o([\![A_1]\!]_m, \ldots, [\![A_n]\!]_m) \\
[\![q\, x.A]\!]_m &= g_q(\{\, [\![A]\!]_{m'} \mid m \sim_x m' \,\})
\end{aligned}$$

Σχέσεις συνεπαγωγής (i)

- Ορισμός: $\vdash \in \mathcal{P}(\mathbf{L}) \times \mathbf{L}$
  - αν $A \in \Gamma$, τότε $\Gamma \vdash A$ — συμπερίληψη
  - αν $\Gamma \vdash A$, τότε $\Gamma, \Delta \vdash A$ — μονοτονία
  - αν $\Gamma \vdash C$ και $\Delta, C \vdash A$, τότε $\Gamma, \Delta \vdash A$ — τομή / cut
- Ενδιαφέρουσες ιδιότητες
  - αν $\Gamma \vdash A$, τότε υπάρχει πεπερασμένο $\Gamma' \subseteq \Gamma$ ώστε $\Gamma' \vdash A$ — σύμπηξη
  - αν $\Gamma \vdash A$ και σ μια αντικατάσταση, τότε $\sigma(\Gamma) \vdash \sigma(A)$ — αντικατάσταση

Σχέσεις συνεπαγωγής (ii)

- Ενδιαφέρουσες ιδιότητες (συνέχεια)
  - αν $\Gamma \vdash A \to B$, τότε και μόνο τότε $\Gamma, A \vdash B$ — συνεπαγωγή
- Σχέση ερμηνείας: για ένα σύστημα τιμών $\mathbf{M}$, ορίζουμε $\Gamma \models_{\mathbf{M}} A$ όταν για κάθε ανάθεση a, τέτοια ώστε για κάθε $B \in \Gamma$ να ισχύει $(\mathbf{M}, a) \Vdash B$, ισχύει $(\mathbf{M}, a) \Vdash A$
- Οι σχέσεις ερμηνείας είναι σχέσεις συνεπαγωγής
- Ικανοποιούν την αντικατάσταση, γενικά όμως όχι τη σύμπηξη και τη συνεπαγωγή

Θεωρία αποδείξεων

- Αποσκοπεί στο συντακτικό ορισμό σχέσεων συνεπαγωγής $\vdash$ για μια γλώσσα
- Συνηθέστεροι συμβολισμοί
  - Αποδείξεις κατά Hilbert/Frege
  - Φυσικό συμπέρασμα (natural deduction)
  - Λογισμός ακολουθητών (sequent calculus)

Κατά Hilbert/Frege (i)

- Κλασσική προτασιακή λογική

$$\begin{aligned}
A1 &: \; A \to (B \to A) \\
A2 &: \; (A \to (B \to C)) \to ((A \to B) \to (A \to C)) \\
A3 &: \; (\neg A \to \neg B) \to (B \to A)
\end{aligned}$$

$$R1 : \; \frac{A \qquad A \to B}{B} \quad \text{[modus ponens]}$$

$$\begin{aligned}
S1 &: \; A \wedge B \;\equiv\; \neg(A \to \neg B) \\
S2 &: \; A \vee B \;\equiv\; \neg A \to B \\
S3 &: \; \mathit{false} \;\equiv\; \neg(A \to A)
\end{aligned}$$

Κατά Hilbert/Frege (ii)

- Κλασσική κατηγορηματική λογική

$A4: \ (\forall x.\ A) \to A$

$A5: \ (\forall x.\ A(x)) \to A(t)$

αν το t είναι ελεύθερο για το x στο $A(x)$

$A6: \ \forall x.\ ((A \to B) \to (A \to \forall x.\ B))$

αν το x δεν είναι ελεύθερο στο $A(x)$

$$R2: \ \frac{A}{\forall x.\ A}\ \text{[generalization]}$$

$S4: \ \exists x.\ A \ \equiv \ \neg(\forall x.\ \neg A)$

Natural deduction (i)

- Κλασσική προτασιακή λογική

$$\frac{A \quad B}{A \wedge B}\ [\wedge i] \qquad \frac{A \wedge B}{A}\ [\wedge e_1] \qquad \frac{A \wedge B}{B}\ [\wedge e_2]$$

$$\frac{A}{A \vee B}\ [\vee i_1] \qquad \frac{B}{A \vee B}\ [\vee i_2]$$

$$\frac{A \vee B \quad \boxed{\begin{array}{c} A \\ \vdots \\ C \end{array}} \quad \boxed{\begin{array}{c} B \\ \vdots \\ C \end{array}}}{C}\ [\vee e] \qquad \frac{\boxed{\begin{array}{c} A \\ \vdots \\ B \end{array}}}{A \to B}\ [\to i]$$

Natural deduction (ii)

- Κλασσική προτασιακή λογική (συνέχεια)

$$\frac{A \quad A \to B}{B}\ [\to e] \qquad \frac{\mathit{false}}{A}\ [\bot e]$$

$$\frac{\boxed{\begin{array}{c} A \\ \vdots \\ \mathit{false} \end{array}}}{\neg A}\ [\neg i] \qquad \frac{A \quad \neg A}{\mathit{false}}\ [\neg e]$$

$$\frac{\neg\neg A}{A}\ [\neg\neg]$$

Natural deduction (iii)

- Κλασσική κατηγορηματική λογική

$$\frac{\boxed{\begin{array}{l} y \\ \quad\ \vdots \\ \quad A(y) \end{array}}}{\forall x.\ A(x)}\ \forall i \qquad \frac{\exists x.\ A(x) \quad \boxed{\begin{array}{c} y \quad A(y) \\ \vdots \\ B \end{array}}}{B}\ \exists e$$

αν το y δεν είναι ελεύθερο έξω από τα κουτιά

$$\frac{\forall x.\ A(x)}{A(t)}\ \forall e \qquad \frac{A(t)}{\exists x.\ A(x)}\ \exists i$$

Sequent calculus (i)

- Κλασσική προτασιακή λογική

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \wedge B}\ [\wedge i]$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ [\wedge e_1] \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ [\wedge e_2]$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ [\vee i_1] \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ [\vee i_2]$$

$$\frac{\Gamma \vdash A \vee B \quad \Delta, A \vdash C \quad E, B \vdash C}{\Gamma, \Delta, E \vdash C}\ [\vee e]$$

Sequent calculus (ii)

- Κλασσική προτασιακή λογική (συνέχεια)

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ [\to i] \qquad \frac{\Gamma \vdash A \quad \Delta \vdash A \to B}{\Gamma, \Delta \vdash B}\ [\to e]$$

$$\frac{\Gamma \vdash \mathit{false}}{\Gamma \vdash A}\ [\bot e] \qquad \frac{\Gamma, A \vdash \mathit{false}}{\Gamma \vdash \neg A}\ [\neg i]$$

$$\frac{\Gamma \vdash A \quad \Delta \vdash \neg A}{\Gamma, \Delta \vdash \mathit{false}}\ [\neg e] \qquad \frac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A}\ [\neg\neg]$$

- Παρένθεση: διαισθητική λογική
  - όπως η κλασσική, χωρίς τον κανόνα $\neg\neg$
  - δεν μπορεί να αποδειχθεί π.χ. ότι $\vdash A \vee \neg A$

Sequent calculus (iii)

- Κλασσική κατηγορηματική λογική

$$\frac{\Gamma \vdash A(y)}{\Gamma \vdash \forall x.\ A(x)}\ [\forall i] \qquad \frac{\Gamma \vdash \forall x.\ A(x)}{\Gamma \vdash A(t)}\ [\forall e]$$

αν το y δεν είναι ελεύθερο στα $\Gamma, A(x)$

$$\frac{\Gamma \vdash A(t)}{\Gamma \vdash \exists x.\ A(x)}\ [\exists i]$$

$$\frac{\Gamma \vdash \exists x.\ A(x) \qquad \Delta, A(y) \vdash B}{\Gamma, \Delta \vdash B}\ [\exists e]$$

αν το y δεν είναι ελεύθερο στα $\Gamma, A(x), B$

Συνέπεια και πληρότητα

- Με δεδομένο ένα σύστημα τιμών $\mathbf{M}$, δηλαδή μια σημασία για τη γλώσσα της λογικής
- Συνέπεια $\vdash$ έναντι $\models_{\mathbf{M}}$
  αν $\Gamma \vdash A$, τότε $\Gamma \models_{\mathbf{M}} A$
- Πληρότητα $\vdash$ έναντι $\models_{\mathbf{M}}$
  αν $\Gamma \models_{\mathbf{M}} A$, τότε $\Gamma \vdash A$
- Η συνέπεια συνήθως αποδεικνύεται εύκολα, με επαγωγή πάνω στους αποδεικτικούς κανόνες
- Η πληρότητα συνήθως αποδεικνύεται αρκετά πιο δύσκολα, με διαφορετικού τύπου τεχνικές

λ-λογισμός (i)

- Alonso Church, αρχές δεκαετίας 1930
- Θεωρία για τη θεμελίωση των μαθηματικών
- Απλό αλλά πλήρες υπολογιστικό μοντέλο
- Βάση για τη δημιουργία του συναρτησιακού μοντέλου προγραμματισμού
- Πρόσφορος συμβολισμός για την περιγραφή της σημασιολογίας γλωσσών προγραμματισμού
- Αρχική μορφή: χωρίς τύπους
- Παραλλαγές με τύπους προτάθηκαν αργότερα (Curry, Church, κ.λπ.)

λ-λογισμός (i)

- Διαισθητική περιγραφή: μια θεωρία συναρτήσεων
  - x — μεταβλητή
  - $F\ A$ — εφαρμογή
  - $\lambda x.\ E[x]$ — αφαίρεση
- Αν x είναι μια μεταβλητή και $E[x]$ μια έκφραση που περιέχει το x, τότε $\lambda x.\ E[x]$ είναι η συνάρτηση f, όπου $f(x) = E[x]$
- Παράδειγμα: $\lambda x.\ x^2 - 3x + 2$

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

λ-λογισμός (ii)

- Μεταβλητές ελεύθερες και δεσμευμένες

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

- Ανάλογο στα μαθηματικά: $\displaystyle\int_{-\pi}^{\pi} \frac{\sin x + \cos y}{\cos x - \sin y}\, dx$
- Αντικατάσταση μεταβλητής με έκφραση

$$\begin{aligned}&(\lambda x.\ x^2 - 3y + 2)[y := z + 1]\\ &\equiv\ \lambda x.\ x^2 - 3(z + 1) + 2\end{aligned}$$

λ-λογισμός (iii)

- Αντικατάσταση (συνέχεια)
  - Αποφυγή προβλημάτων $\displaystyle\int_{-\pi}^{\pi} \frac{\sin x + \cos y}{\cos x - \sin y}\, dx$
  - Όχι σε δεσμευμένες μεταβλητές!
    $$\int_{-\pi}^{\pi} \frac{\sin 42 + \cos y}{\cos 42 - \sin y}\, d42$$
  - Να μην προκαλεί δέσμευση μεταβλητών!
    $$\int_{-\pi}^{\pi} \frac{\sin x + \cos(3x + 1)}{\cos x - \sin(3x + 1)}\, dx$$

λ-λογισμός χωρίς τύπους (i)

- Σύνταξη: $x \in V$ και $M, N \in \Lambda$

  $M, N \ ::= \ x \mid (\lambda x.\, M) \mid (M\, N)$
- Συντακτικές συμβάσεις
  - Οι εξωτερικές παρενθέσεις δε γράφονται
  - Η εφαρμογή είναι αριστερά προσεταιριστική
  - Η αφαίρεση εκτείνεται όσο είναι δυνατόν
- Σχέση ταυτότητας: $M \equiv N$

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

λ-λογισμός χωρίς τύπους (ii)

- Ελεύθερες μεταβλητές

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

- Κλειστοί όροι (closed terms ή combinators)

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

λ-λογισμός χωρίς τύπους (iii)

- Αντικατάσταση

$$\begin{array}{lcll} x[x := N] & \equiv & N & \\ y[x := N] & \equiv & y & \text{αν } y \neq x \\ (P\,Q)[x := N] & \equiv & P[x := N]\, Q[x := N] & \\ (\lambda x.\, P)[x := N] & \equiv & \lambda x.\, P & \\ (\lambda y.\, P)[x := N] & \equiv & \lambda y.\, P[x := N] & \end{array}$$

αν $y \neq x$, και ($y \notin \mathrm{FV}(N)$ ή $x \notin \mathrm{FV}(P)$)

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

αν $y \neq x$, και $y \in \mathrm{FV}(N)$ και $x \in \mathrm{FV}(P)$,

όπου $z \notin \mathrm{FV}(P) \cup \mathrm{FV}(N)$

λ-λογισμός χωρίς τύπους (iv)

- Συμβατές σχέσεις

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

- Μετατροπές $\rightarrow_\alpha, \rightarrow_\beta, \rightarrow_\eta$: οι ελάχιστες συμβατές σχέσεις που πληρούν τα παρακάτω

$$\begin{array}{lcll} \lambda x.\, M & \rightarrow_\alpha & \lambda y.\, M[x := y] & y \notin \mathrm{FV}(M) \\ (\lambda x.\, M)\, N & \rightarrow_\beta & M[x := N] & \\ \lambda x.\, M\, x & \rightarrow_\eta & M & x \notin \mathrm{FV}(M) \end{array}$$

λ-λογισμός χωρίς τύπους (v)

- Αναγωγές και ισότητες (κλείσιμο της $\rightarrow$)

$$\begin{array}{ll} \twoheadrightarrow & \text{ανακλαστικό + μεταβατικό} \\ = & \text{ανακλαστικό + μεταβατικό + συμμετρικό} \end{array}$$

- Κανονική μορφή

  όρος N χωρίς β και η-redex
- Κανονικοποιήσιμος όρος

  όρος M με $M \twoheadrightarrow N$ και N κανονική μορφή
- Ισχυρά κανονικοποιήσιμος όρος

  κάθε ακολουθία μετατροπών καταλήγει σε κανονική μορφή

λ-λογισμός χωρίς τύπους (vi)

- Παραδείγματα

$$\begin{array}{lcl} M_1 & \equiv & \lambda z.\, (\lambda f.\, \lambda x.\, f\, z\, x)\, (\lambda y.\, y) \\ \Omega & \equiv & (\lambda x.\, x\, x)\, (\lambda x.\, x\, x) \\ M_2 & \equiv & (\lambda x.\, x\, x\, y)\, (\lambda x.\, x\, x\, y) \\ M_3 & \equiv & (\lambda z.\, y)\, ((\lambda x.\, x\, x)\, (\lambda x.\, x\, x)) \end{array}$$

- Ισοδυναμία κανονικών μορφών

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

λ-λογισμός χωρίς τύπους (vii)

- Θεώρημα Church-Rosser
  Έστω $M, N_1, N_2 \in \Lambda$
  τ.ω. $M \twoheadrightarrow N_1$ και $M \twoheadrightarrow N_2$.
  Τότε υπάρχει $N \in \Lambda$
  τ.ω. $N_1 \twoheadrightarrow N$ και $N_2 \twoheadrightarrow N$.

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

- Μοναδικότητα κανονικών μορφών αν υπάρχουν και modulo $=_\alpha$
- Θεώρημα κανονικοποίησης
  Αν ο όρος M έχει κανονική μορφή, τότε η μετατροπή του αριστερότερου redex οδηγεί σε αυτήν.

λ-λογισμός χωρίς τύπους (viii)

- Τελεστής σταθερού σημείου: $F\,(Y\,F) = Y\,F$

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

- Χρησιμεύει για την αναπαράσταση αναδρομικών συναρτήσεων

$$F \;\equiv\; \lambda f.\,\lambda n.\,\textbf{if}\ n = 0\ \textbf{then}\ 1\ \textbf{else}\ n \cdot f(n-1)$$
$$M \;\equiv\; Y\,F \qquad \text{παραγοντικό}$$

$$\begin{aligned} M\,2 \;\equiv\; Y\,F\,2 \;&=\; F\,(Y\,F)\,2 \\ &= \textbf{if}\ 2 = 0\ \textbf{then}\ 1\ \textbf{else}\ 2 \cdot (Y\,F)(2-1) \\ &= 2 \cdot Y\,F\,1 \;=\; 2 \cdot F\,(Y\,F)\,1 \;=\; \ldots \end{aligned}$$

Εκφραστική δύναμη (i)

- Λογικές τιμές

$$\begin{aligned} \textbf{true} &\equiv \lambda x.\,\lambda y.\,x \\ \textbf{false} &\equiv \lambda x.\,\lambda y.\,y \\ \textbf{not} &\equiv \lambda z.\,z\,\textbf{false}\,\textbf{true} \\ \textbf{cond} &\equiv \lambda z.\,\lambda x.\,\lambda y.\,z\,x\,y \\ \textbf{if}\ B\ \textbf{then}\ N\ \textbf{else}\ M &\equiv \textbf{cond}\,B\,N\,M \end{aligned}$$

Εκφραστική δύναμη (ii)

- Διατεταγμένα ζεύγη

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

Εκφραστική δύναμη (iii)

- Φυσικοί αριθμοί (αριθμοειδή του Church)

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

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

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

Εκφραστική δύναμη (iv)

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

Εκφραστική δύναμη (v)

- Πέρασμα παραμέτρων (συνέχεια)
  - το πέρασμα κατ' αξία / πρόθυμη αποτίμηση (call-by-value / eager evaluation) αντιστοιχεί στη στρατηγική αποτίμησης που ανάγει ένα β-redex μόνο αν το όρισμα είναι κανονική τιμή
  - το πέρασμα κατ' όνομα / οκνηρή αποτίμηση (call-by-name / lazy evaluation) αντιστοιχεί στην στρατηγική αποτίμησης που ανάγει το αριστερότερο β-redex

Συστήματα τύπων (i)

- Ορισμός: Συντακτικές μέθοδοι πολυωνυμικού χρόνου για την ταξινόμηση των τμημάτων ενός προγράμματος ανάλογα με τις τιμές που αυτά υπολογίζουν, με σκοπό να αποδείξουν την απουσία ορισμένων ανεπιθύμητων συμπεριφορών κατά την εκτέλεσή του
  (Benjamin Pierce, *Types and Programming Languages*, 2002)
- Θεωρία τύπων: κλάδος των μαθηματικών, της λογικής και της φιλοσοφίας
- Ισομορφισμός Curry-Howard: αντιστοιχία μεταξύ θεωρίας τύπων και θεωρίας αποδείξεων

Συστήματα τύπων (ii)

- Ιδιότητες που απορρέουν από αυτόν τον ορισμό
  - Έμφαση στις γλώσσες προγραμματισμού
  - Στατική προσέγγιση της συμπεριφοράς εκτέλεσης των προγραμμάτων
  - Συντηρητική αντιμετώπιση ανεπιθύμητων συμπεριφορών

    $$\text{if συνθήκη then } 42 \text{ else σφάλμα}$$

  - Σφάλματα τύπων κατά την εκτέλεση (run-time type errors)
  - Ασφάλεια μιας γλώσσας προγραμματισμού $\Leftrightarrow$ Συνέπεια του συστήματος τύπων

Βασικοί τύποι (i)

- Σύνταξη (εκφράσεις)

$$\begin{array}{rcl} e & ::= & n \mid -e \mid e_1 + e_2 \mid \ldots \\ & \mid & \mathit{true} \mid \mathit{false} \mid \neg e \mid e_1 \wedge e_2 \mid \ldots \\ & \mid & e_1 < e_2 \mid \text{if } e \text{ then } e_1 \text{ else } e_2 \mid \ldots \end{array}$$

- Λειτουργική σημασιολογία (operational semantics)
  - Σχέση μετάβασης $s \longrightarrow s'$ μεταξύ καταστάσεων μιας αφηρημένης μηχανής
  - Για την παραπάνω γλώσσα εκφράσεων:

$$s \ ::= \ e$$

Βασικοί τύποι (ii)

- Αποτίμηση

$$\text{if } \mathit{true} \text{ then } (15+27) \text{ else } (3+4) \longrightarrow 15+27 \longrightarrow 42$$

  - με ποια σειρά γίνονται οι πράξεις;
  - πότε σταματά η αποτίμηση;
- Τιμές

$$v \ ::= \ n \mid \mathit{true} \mid \mathit{false}$$

Βασικοί τύποι (iii)

- Σημασία τελεστών
  - Αν $\diamond$ κάποιος τελεστής με ένα τελούμενο, τότε $[\![\diamond]\!] : v \to v$ είναι η σημασία του

    π.χ. $[\![\neg]\!](\mathit{true}) = \mathit{false}$
  - Αν $\circ$ κάποιος τελεστής με δύο τελούμενα, τότε $[\![\circ]\!] : v \times v \to v$ είναι η σημασία του

    π.χ. $[\![+]\!](15, 27) = 42$

Βασικοί τύποι (iv)

- Λειτουργική σημασιολογία

$$\diamond v \longrightarrow [\![\diamond]\!](v) \qquad \text{if } \mathit{true} \text{ then } e_1 \text{ else } e_2 \longrightarrow e_1$$

$$v_1 \circ v_2 \longrightarrow [\![\circ]\!](v_1, v_2) \qquad \text{if } \mathit{false} \text{ then } e_1 \text{ else } e_2 \longrightarrow e_2$$

$$\frac{e \longrightarrow e'}{\diamond e \longrightarrow \diamond e'}$$

$$\frac{e_1 \longrightarrow e_1'}{e_1 \circ e_2 \longrightarrow e_1' \circ e_2} \qquad \frac{e_2 \longrightarrow e_2'}{v_1 \circ e_2 \longrightarrow v_1 \circ e_2'}$$

$$\frac{e \longrightarrow e'}{\text{if } e \text{ then } e_1 \text{ else } e_2 \longrightarrow \text{if } e' \text{ then } e_1 \text{ else } e_2}$$

Βασικοί τύποι (v)

- Λειτουργική σημασιολογία (συνέχεια)
  - Θεώρημα ντετερμινιστικής αποτίμησης: Αν $e \longrightarrow e'$ και $e \longrightarrow e''$ τότε $e' \equiv e''$
  - Ορισμός: e είναι κανονική μορφή όταν δεν υπάρχει e' τέτοια ώστε $e \longrightarrow e'$
  - Θεώρημα: κάθε τιμή είναι κανονική μορφή
  - Όχι αντίστροφα! $\quad \text{if } 1 \text{ then } \mathit{true} \text{ else } \mathit{false}$
  - Ορισμός: e είναι κολλημένη αν είναι κανονική μορφή χωρίς να είναι τιμή

Βασικοί τύποι (vi)

- Λειτουργική σημασιολογία (συνέχεια)
  - Ορισμός: $\longrightarrow^*$ είναι το ανακλαστικό και μεταβατικό κλείσιμο της $\longrightarrow$
  - Θεώρημα μοναδικότητας κανονικών μορφών: Αν $e \longrightarrow^* u$ και $e \longrightarrow^* u'$, όπου u, u' κανονικές μορφές, τότε $u \equiv u'$
  - Θεώρημα κανονικοποίησης (normalization) ή τερματισμού: για κάθε e υπάρχει κανονική μορφή u τέτοια ώστε $e \longrightarrow^* u$
- Σκοπός συστήματος τύπων: αποφυγή κολλημένων εκφράσεων

Βασικοί τύποι (vii)

- Σύνταξη (τύποι)

$$\tau \ ::= \ \text{Int} \mid \text{Bool}$$

- Κανόνες τύπων $\qquad e : \tau$

$$n : \text{Int} \qquad \mathit{true} : \text{Bool} \qquad \mathit{false} : \text{Bool}$$

$$\frac{e_1 : \text{Int} \quad e_2 : \text{Int}}{e_1 + e_2 : \text{Int}} \qquad \frac{e_1 : \text{Int} \quad e_2 : \text{Int}}{e_1 < e_2 : \text{Bool}}$$

$$\frac{e : \text{Int}}{-e : \text{Int}} \qquad \frac{e_1 : \text{Bool} \quad e_2 : \text{Bool}}{e_1 \wedge e_2 : \text{Bool}}$$

$$\frac{e : \text{Bool}}{\neg e : \text{Bool}} \qquad \frac{e : \text{Bool} \quad e_1 : \tau \quad e_2 : \tau}{\text{if } e \text{ then } e_1 \text{ else } e_2 : \tau}$$

Βασικοί τύποι (viii)

- Παραγωγές τύπων (typing derivations)

$$\frac{\mathit{true} : \text{Bool} \quad \dfrac{15 : \text{Int} \quad 27 : \text{Int}}{15 + 27 : \text{Int}} \quad \dfrac{3 : \text{Int} \quad 4 : \text{Int}}{3 + 4 : \text{Int}}}{\text{if } \mathit{true} \text{ then } (15 + 27) \text{ else } (3 + 4) : \text{Int}}$$

- Ιδιότητες του συστήματος τύπων
  - Θεώρημα μοναδικότητας τύπων
  - Θεώρημα μοναδικότητας παραγωγών
  - Λήμμα αντιστροφής (inversion lemma): Μέθοδος κατασκευής παραγωγών τύπου, π.χ.
    - ▷ αν $e_1 < e_2 : \tau$ τότε $\tau = \text{Bool}$, $e_1 : \text{Int}$ και $e_2 : \text{Int}$

Βασικοί τύποι (ix)

- Ιδιότητες του συστήματος τύπων (συνέχεια)
  - Θεώρημα προόδου (progress): Αν $e : \tau$ τότε είτε e είναι τιμή, είτε υπάρχει e' τέτοιο ώστε $e \longrightarrow e'$
  - Θεώρημα διατήρησης (preservation): Αν $e : \tau$ και $e \longrightarrow e'$ τότε $e' : \tau$
  - Ασφάλεια = Πρόοδος + Διατήρηση Αν η έκφραση e έχει τύπο, η αποτίμησή της δεν μπορεί να κολλήσει

Τύποι συναρτήσεων (i)

- λ-λογισμός με απλούς τύπους
- Σύνταξη (τύποι, εκφράσεις, τιμές)

$$\begin{array}{lcl} \tau & ::= & \ldots \mid \tau_1 \to \tau_2 \\ e & ::= & \ldots \mid x \mid \lambda x : \tau .\, e \mid e_1\, e_2 \\ v & ::= & \ldots \mid \lambda x : \tau .\, e \end{array}$$

- Λειτουργική σημασιολογία: call by value

$$\frac{e_1 \longrightarrow e_1'}{e_1\, e_2 \longrightarrow e_1'\, e_2} \qquad \frac{e_2 \longrightarrow e_2'}{v_1\, e_2 \longrightarrow v_1\, e_2'}$$

$$(\lambda x : \tau .\, e)\, v \longrightarrow e[x := v]$$

Τύποι συναρτήσεων (ii)

- Λειτουργική σημασιολογία: call by name

$$\frac{e_1 \longrightarrow e_1'}{e_1\, e_2 \longrightarrow e_1'\, e_2} \qquad (\lambda x : \tau .\, e)\, e' \longrightarrow e[x := e']$$

- Περιβάλλοντα τύπων Γ: σύνολα ζευγών (x, τ)
- Κανόνες τύπων $\Gamma \vdash e : \tau$

$$\frac{(x, \tau) \in \Gamma}{\Gamma \vdash x : \tau} \qquad \frac{\Gamma, x : \tau \vdash e : \tau'}{\Gamma \vdash \lambda x : \tau .\, e : \tau \to \tau'}$$

$$\frac{\Gamma \vdash e_1 : \tau \to \tau' \qquad \Gamma \vdash e_2 : \tau}{\Gamma \vdash e_1\, e_2 : \tau'}$$

Τύποι συναρτήσεων (iii)

- Ιδιότητες του συστήματος τύπων
  - Θεώρημα προόδου (progress):
    Αν $\varnothing \vdash e : \tau$ τότε είτε e είναι τιμή, είτε υπάρχει e' τέτοιο ώστε $e \longrightarrow e'$
  - Θεώρημα διατήρησης (preservation):
    Αν $\Gamma \vdash e : \tau$ και $e \longrightarrow e'$ τότε $e' : \tau$

Απλές επεκτάσεις (i)

- Τύπος μονάδας Unit (πρβλ. `void` στη C)
- Σύνταξη (τύποι, εκφράσεις, τιμές)

$$\begin{array}{lcl} \tau & ::= & \ldots \mid \mathrm{Unit} \\ e & ::= & \ldots \mid \mathit{unit} \mid e_1 ; e_2 \\ v & ::= & \ldots \mid \mathit{unit} \end{array}$$

- Λειτουργική σημασιολογία

$$\mathit{unit} ; e \longrightarrow e \qquad \frac{e_1 \longrightarrow e_1'}{e_1 ; e_2 \longrightarrow e_1' ; e_2}$$

Απλές επεκτάσεις (ii)

- Κανόνες τύπων

$$\Gamma \vdash \mathit{unit} : \mathrm{Unit} \qquad \frac{\Gamma \vdash e_1 : \mathrm{Unit} \qquad \Gamma \vdash e_2 : \tau}{\Gamma \vdash e_1 ; e_2 : \tau}$$

- Syntactic sugar: σε σημασιολογία call-by-value, η ακολουθιακή αποτίμηση μπορεί να οριστεί ως παραγόμενη μορφή (derived form)

$$e_1 ; e_2 \;\equiv\; (\lambda x : \mathrm{Unit} .\, e_2)\, e_1 \qquad \text{με } x \notin \mathrm{FV}(e_2)$$

Απλές επεκτάσεις (iii)

- Απόδοση ονομάτων — δομή let
- Σύνταξη (εκφράσεις)

$$e \;::=\; \ldots \mid \mathrm{let}\; x = e_1 \;\mathrm{in}\; e_2$$

- Λειτουργική σημασιολογία: call by value

$$\frac{e_1 \longrightarrow e_1'}{\mathrm{let}\; x = e_1 \;\mathrm{in}\; e_2 \longrightarrow \mathrm{let}\; x = e_1' \;\mathrm{in}\; e_2}$$

$$\mathrm{let}\; x = v \;\mathrm{in}\; e \longrightarrow e[x := v]$$

- Λειτουργική σημασιολογία: call by name

$$\mathrm{let}\; x = e_1 \;\mathrm{in}\; e_2 \longrightarrow e_2[x := e_1]$$

Απλές επεκτάσεις (iv)

- Κανόνας τύπων

$$\frac{\Gamma \vdash e_1 : \tau' \qquad \Gamma, x : \tau' \vdash e_2 : \tau}{\Gamma \vdash \mathrm{let}\, x = e_1 \,\mathrm{in}\, e_2 : \tau}$$

- Syntactic sugar: η δομή let μπορεί να οριστεί ως παραγόμενη μορφή, υπό την προϋπόθεση ότι η σημασιολογία της (call by value / call by name) συμφωνεί με αυτή των συναρτήσεων

$$\mathrm{let}\, x = e_1 \,\mathrm{in}\, e_2 \;\equiv\; (\lambda x : \tau'.\, e_2)\, e_1$$

Η απαλοιφή της όμως γίνεται βάσει της παραγωγής τύπων, προκειμένου να βρεθεί το τ'

Ζεύγη (i)

- Σύνταξη (τύποι, εκφράσεις)

$$\tau \;::=\; \ldots \mid \tau_1 \times \tau_2$$
$$e \;::=\; \ldots \mid \langle e_1, e_2 \rangle \mid \mathrm{fst}\, e \mid \mathrm{snd}\, e$$

- Τιμές: πρόθυμα ζεύγη

$$v \;::=\; \ldots \mid \langle v_1, v_2 \rangle$$

- Λειτουργική σημασιολογία: πρόθυμα ζεύγη

$$\mathrm{fst}\,\langle v_1, v_2 \rangle \longrightarrow v_1 \qquad \mathrm{snd}\,\langle v_1, v_2 \rangle \longrightarrow v_2$$

$$\frac{e \longrightarrow e'}{\mathrm{fst}\, e \longrightarrow \mathrm{fst}\, e'} \qquad \frac{e \longrightarrow e'}{\mathrm{snd}\, e \longrightarrow \mathrm{snd}\, e'}$$

Ζεύγη (ii)

- Λειτουργική σημασιολογία (συνέχεια)

$$\frac{e_1 \longrightarrow e_1'}{\langle e_1, e_2 \rangle \longrightarrow \langle e_1', e_2 \rangle} \qquad \frac{e_2 \longrightarrow e_2'}{\langle v_1, e_2 \rangle \longrightarrow \langle v_1, e_2' \rangle}$$

- Τιμές: οκνηρά ζεύγη

$$v \;::=\; \ldots \mid \langle e_1, e_2 \rangle$$

- Λειτουργική σημασιολογία: οκνηρά ζεύγη

$$\mathrm{fst}\,\langle e_1, e_2 \rangle \longrightarrow e_1 \qquad \mathrm{snd}\,\langle e_1, e_2 \rangle \longrightarrow e_2$$

$$\frac{e \longrightarrow e'}{\mathrm{fst}\, e \longrightarrow \mathrm{fst}\, e'} \qquad \frac{e \longrightarrow e'}{\mathrm{snd}\, e \longrightarrow \mathrm{snd}\, e'}$$

Ζεύγη (iii)

- Κανόνες τύπων

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash \langle e_1, e_2 \rangle : \tau_1 \times \tau_2}$$

$$\frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \mathrm{fst}\, e : \tau_1} \qquad \frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash \mathrm{snd}\, e : \tau_2}$$

- Επεκτάσεις των ζευγών
  - Πλειάδες (tuples) $\langle e_1, \ldots, e_n \rangle$
  - Εγγραφές (records) $\langle x_1{=}e_1, \ldots, x_n{=}e_n \rangle$

Αθροίσματα (i)

- Σύνταξη (τύποι, εκφράσεις)

$$\tau \;::=\; \ldots \mid \tau_1 + \tau_2$$
$$e \;::=\; \ldots \mid \mathrm{inl}\, e \mid \mathrm{inr}\, e \mid [e_1, e_2]$$

- Τιμές: πρόθυμα αθροίσματα

$$v \;::=\; \ldots \mid \mathrm{inl}\, v \mid \mathrm{inr}\, v \mid [v_1, v_2]$$

- Λειτουργική σημασιολογία: πρόθυμα αθροίσματα

$$[v_1, v_2]\,(\mathrm{inl}\, v) \longrightarrow v_1\, v \qquad [v_1, v_2]\,(\mathrm{inr}\, v) \longrightarrow v_2\, v$$

$$\frac{e \longrightarrow e'}{\mathrm{inl}\, e \longrightarrow \mathrm{inl}\, e'} \qquad \frac{e \longrightarrow e'}{\mathrm{inr}\, e \longrightarrow \mathrm{inr}\, e'}$$

Αθροίσματα (ii)

- Λειτουργική σημασιολογία (συνέχεια)

$$\frac{e_1 \longrightarrow e_1'}{[e_1, e_2] \longrightarrow [e_1', e_2]} \qquad \frac{e_2 \longrightarrow e_2'}{[v_1, e_2] \longrightarrow [v_1, e_2']}$$

- Τιμές: οκνηρά αθροίσματα

$$v \;::=\; \ldots \mid \mathrm{inl}\, e \mid \mathrm{inr}\, e \mid [e_1, e_2]$$

- Λειτουργική σημασιολογία: οκνηρά αθροίσματα

$$[e_1, e_2]\,(\mathrm{inl}\, e) \longrightarrow e_1\, e \qquad [e_1, e_2]\,(\mathrm{inr}\, e) \longrightarrow e_2\, e$$

Αθροίσματα (iii)

- Κανόνες τύπων

$$\frac{\Gamma \vdash e : \tau_1}{\Gamma \vdash \mathrm{inl}\, e : \tau_1 + \tau_2} \qquad \frac{\Gamma \vdash e : \tau_2}{\Gamma \vdash \mathrm{inr}\, e : \tau_1 + \tau_2}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \to \tau \qquad \Gamma \vdash e_2 : \tau_2 \to \tau}{\Gamma \vdash [e_1, e_2] : (\tau_1 + \tau_2) \to \tau}$$

- Αθροίσματα και μοναδικότητα τύπων
- Επεκτάσεις των αθροισμάτων
  - Παραλλαγές (variants)
  - Ενώσεις (unions), απαριθμήσεις (enumerations)

Αναδρομή (i)

- Σύνταξη (εκφράσεις)

$$e \ ::= \ \ldots \mid \mathrm{fix}\, e$$

- Λειτουργική σημασιολογία: call by value

$$\mathrm{fix}\,(\lambda x : \tau.\, e) \longrightarrow e[x := \mathrm{fix}\,(\lambda x : \tau.\, e)]$$

$$\frac{e \longrightarrow e'}{\mathrm{fix}\, e \longrightarrow \mathrm{fix}\, e'}$$

Αναδρομή (ii)

- Λειτουργική σημασιολογία: call by name

$$\mathrm{fix}\, e \longrightarrow e\,(\mathrm{fix}\, e)$$

- Κανόνας τύπων

$$\frac{\Gamma \vdash e : \tau \to \tau}{\Gamma \vdash \mathrm{fix}\, e : \tau}$$

Αναδρομή (iii)

- Παράδειγμα: call by value

$$\begin{aligned} p \ \equiv \ & \mathrm{fix}\,(\lambda f : \mathrm{Int} \to \mathrm{Int}.\, \lambda n : \mathrm{Int}. \\ & \quad \mathrm{if}\; n \le 1 \;\mathrm{then}\; 1 \;\mathrm{else}\; n * f\,(n-1)) \end{aligned}$$

$$\begin{aligned} p\,3 \ &\longrightarrow \ (\lambda n : \mathrm{Int}.\, \mathrm{if}\; n \le 1 \;\mathrm{then}\; 1 \;\mathrm{else}\; n * p\,(n-1))\,3 \\ &\longrightarrow \ \mathrm{if}\; 3 \le 1 \;\mathrm{then}\; 1 \;\mathrm{else}\; 3 * p\,(3-1) \\ &\longrightarrow \ \mathrm{if}\; \mathit{false} \;\mathrm{then}\; 1 \;\mathrm{else}\; 3 * p\,(3-1) \\ &\longrightarrow \ 3 * p\,(3-1) \\ &\longrightarrow \ 3 * p\,2 \\ &\longrightarrow^* \ \ldots \\ &\longrightarrow \ 3 * (2 * 1) \ \longrightarrow \ \ldots \ \longrightarrow \ 6 \end{aligned}$$

Αναδρομή (iv)

- Παράδειγμα: call by name

$$\begin{aligned} p \ \equiv \ & \mathrm{fix}\,(\lambda f : \mathrm{Int} \to \mathrm{Int}.\, \lambda n : \mathrm{Int}. \\ & \quad \mathrm{if}\; n \le 1 \;\mathrm{then}\; 1 \;\mathrm{else}\; n * f\,(n-1)) \end{aligned}$$

$$\begin{aligned} p\,3 \ &\longrightarrow \ (\lambda f : \mathrm{Int} \to \mathrm{Int}.\, \lambda n : \mathrm{Int}. \\ &\qquad \mathrm{if}\; n \le 1 \;\mathrm{then}\; 1 \;\mathrm{else}\; n * f\,(n-1))\; p\,3 \\ &\longrightarrow \ (\lambda n : \mathrm{Int}.\, \mathrm{if}\; n \le 1 \;\mathrm{then}\; 1 \;\mathrm{else}\; n * p\,(n-1))\,3 \\ &\longrightarrow \ \mathrm{if}\; 3 \le 1 \;\mathrm{then}\; 1 \;\mathrm{else}\; 3 * p\,(3-1) \\ &\longrightarrow \ \mathrm{if}\; \mathit{false} \;\mathrm{then}\; 1 \;\mathrm{else}\; 3 * p\,(3-1) \\ &\longrightarrow \ 3 * p\,(3-1) \\ &\longrightarrow \ \ldots \\ &\longrightarrow \ 3 * ((3-1) * (3-1-1)) \ \longrightarrow \ \ldots \ \longrightarrow \ 6 \end{aligned}$$

Αναδρομή (v)

- Αναδρομή και θεώρημα κανονικοποίησης

$$\omega \ \equiv \ \mathrm{fix}\,(\lambda x : \mathrm{Int}.\, x)$$

- Αποτίμηση: call by value

$$\begin{aligned} \omega \ &\longrightarrow \ \mathrm{fix}\,(\lambda x : \mathrm{Int}.\, x) \ \equiv \ \omega \\ &\longrightarrow \ \ldots \end{aligned}$$

- Αποτίμηση: call by name

$$\begin{aligned} \omega \ &\longrightarrow \ \mathrm{fix}\,(\lambda x : \mathrm{Int}.\, x) \\ &\longrightarrow \ (\lambda x : \mathrm{Int}.\, x)\,\omega \\ &\longrightarrow \ \omega \\ &\longrightarrow \ \ldots \end{aligned}$$

Αναδρομή (vi)

- Ευκολότερη σύνταξη: δομή letrec
- Σύνταξη (εκφράσεις)

$$e \ ::= \ \ldots \mid \mathrm{letrec}\, x = e_1 \,\mathrm{in}\, e_2$$

- Κανόνας τύπων

$$\frac{\Gamma, x : \tau' \vdash e_1 : \tau' \qquad \Gamma, x : \tau' \vdash e_2 : \tau}{\Gamma \vdash \mathrm{letrec}\, x = e_1 \,\mathrm{in}\, e_2 : \tau}$$

- Παραγόμενη μορφή

$$\mathrm{letrec}\, x = e_1 \,\mathrm{in}\, e_2 \ \equiv$$
$$\mathrm{let}\, x = \mathrm{fix}\,(\lambda x : \tau'.\, e_1)\,\mathrm{in}\, e_2$$

Αναφορές (i)

- Προστακτικός προγραμματισμός: μεταβλητές και ανάθεση
- Μνήμη: Αναφορές (references) ή δείκτες (pointers)
- Σύνταξη (τύποι, εκφράσεις)

$$\tau \ ::= \ \ldots \mid \mathrm{Ref}\,\tau$$
$$e \ ::= \ \ldots \mid \mathrm{ref}\, e \mid !e \mid e_1 := e_2$$

- Δέσμευση (allocation), αποδεικτοδότηση (dereferencing) και συνωνυμία (aliasing)
- Συλλογή σκουπιδιών (garbage collection)

Αναφορές (ii)

- Κανόνες τύπων

$$\frac{\Gamma \vdash e : \tau}{\Gamma \vdash \mathrm{ref}\, e : \mathrm{Ref}\,\tau} \qquad \frac{\Gamma \vdash e : \mathrm{Ref}\,\tau}{\Gamma \vdash\, !e : \tau}$$

$$\frac{\Gamma \vdash e_1 : \mathrm{Ref}\,\tau \qquad \Gamma \vdash e_2 : \tau}{\Gamma \vdash e_1 := e_2 : \mathrm{Unit}}$$

- Παράδειγμα: call by value

$$\mathrm{let}\, r = \mathrm{ref}\, 6 \,\mathrm{in}\, !r * (r := !r + 1; !r)$$

- Αποτίμηση: κατάλληλες τιμές για αναφορές, αλλαγή στον ορισμό των καταστάσεων

Αναφορές (iii)

- Θέσεις μνήμης (locations) loc_i, όπου $i \in \mathbb{N}$
- Μνήμες $m : \mathbb{N} \rightharpoonup v$, μερικές (partial) συναρτήσεις
- Σύνταξη (τιμές, εκφράσεις, καταστάσεις)

$$v \ ::= \ \ldots \mid \mathrm{loc}_i$$
$$e \ ::= \ \ldots \mid \mathrm{ref}\, e \mid !e \mid e_1 := e_2 \mid \mathrm{loc}_i$$
$$s \ ::= \ (e, m)$$

- Λειτουργική σημασιολογία
  - Κάθε υπάρχων σημασιολογικός κανόνας της μορφής $e \longrightarrow e'$ πρέπει να μετατραπεί στη νέα μορφή $(e, m) \longrightarrow (e', m')$

Αναφορές (iv)

- Λειτουργική σημασιολογία (συνέχεια)

$$\frac{(e, m) \longrightarrow (e', m')}{(!e, m) \longrightarrow (!e', m')} \qquad \frac{m(i) = v}{(!\mathrm{loc}_i, m) \longrightarrow (v, m)}$$

$$\frac{(e_1, m) \longrightarrow (e_1', m')}{(e_1 := e_2, m) \longrightarrow (e_1' := e_2, m')}$$

$$\frac{(e_2, m) \longrightarrow (e_2', m')}{(v_1 := e_2, m) \longrightarrow (v_1 := e_2', m')}$$

$$(\mathrm{loc}_i := v, m) \longrightarrow (\mathit{unit}, m\{i \mapsto v\})$$

Αναφορές (v)

- Λειτουργική σημασιολογία (συνέχεια)

$$\frac{(e, m) \longrightarrow (e', m')}{(\mathrm{ref}\, e, m) \longrightarrow (\mathrm{ref}\, e', m')}$$

$$\frac{j = \max(\mathrm{dom}(m)) + 1}{(\mathrm{ref}\, v, m) \longrightarrow (\mathrm{loc}_j, m\{j \mapsto v\})}$$

  - Η αφηρημένη μηχανή που χρησιμοποιείται στη λειτουργική σημασιολογία δεν κάνει συλλογή σκουπιδιών

Αναφορές (vi)

- Κανόνες τύπων (συνέχεια)
  - Δεν υπάρχει κανόνας τύπων για loc_i
  - Δε χρειάζεται στον ελεγκτή τύπων, γιατί δεν επιτρέπονται loc_i στα προγράμματα
  - Χρειάζεται όμως για τη μελέτη των ιδιοτήτων της γλώσσας
  - Τύποι μνήμης $M : \mathbb{N} \rightharpoonup \tau$, μερικές συναρτήσεις
  - Επέκταση σχέσης τύπων $\Gamma; M \vdash e : \tau$

$$\frac{M(i) = \tau}{\Gamma; M \vdash \mathrm{loc}_i : \mathrm{Ref}\,\tau}$$

Αναφορές (vii)

- Ιδιότητες του συστήματος τύπων
  - Ορισμός: $\Gamma \vdash m : M$ όταν $\mathrm{dom}(m) = \mathrm{dom}(M)$ και για κάθε $i \in \mathrm{dom}(m)$ ισχύει $\Gamma; M \vdash m(i) : M(i)$
  - Θεώρημα προόδου (progress):
    Αν $\varnothing; M \vdash e : \tau$ τότε είτε e είναι τιμή, είτε για κάθε m τέτοιο ώστε $\varnothing \vdash m : M$ υπάρχει (e', m') τέτοιο ώστε $(e, m) \longrightarrow (e', m')$

Αναφορές (viii)

- Ιδιότητες του συστήματος τύπων (συνέχεια)
  - Θεώρημα διατήρησης (preservation):
    Αν $\Gamma; M \vdash e : \tau$,
    $\Gamma \vdash m : M$ και
    $(e, m) \longrightarrow (e', m')$,
    τότε υπάρχει κάποιο M' τέτοιο ώστε
    $M \subseteq M'$
    $\Gamma \vdash m' : M'$ και
    $\Gamma; M' \vdash e' : \tau$

Εξαιρέσεις (i)

- Πρόβλεψη εξαιρετικών περιπτώσεων (exceptions)
- Σύνταξη (εκφράσεις)

$$e \;::=\; \ldots \mid \mathrm{abort}$$

- Λειτουργική σημασιολογία (call by value)

$$\mathrm{abort}\, e \longrightarrow \mathrm{abort} \qquad v\,\mathrm{abort} \longrightarrow \mathrm{abort}$$

  - Απαιτούνται κανόνες για την περιγραφή της αλληλεπίδρασης των εξαιρέσεων με τα άλλα χαρακτηριστικά της γλώσσας

Εξαιρέσεις (ii)

- Κανόνας τύπων

$$\Gamma \vdash \mathrm{abort} : \tau$$

- Ιδιότητες του συστήματος τύπων
  - Εξαιρέσεις και μοναδικότητα τύπων
  - Θεώρημα προόδου (progress):
    Αν $\varnothing \vdash e : \tau$ τότε είτε e είναι τιμή, είτε είναι abort, είτε υπάρχει e' τέτοιο ώστε $e \longrightarrow e'$

Εξαιρέσεις (iii)

- Παραδείγματα

$$\begin{array}{l} \mathit{checkSubtract} \;:\; \mathrm{Nat} \to \mathrm{Nat} \to \mathrm{Nat} \\ \quad \equiv\; \lambda n : \mathrm{Nat}.\, \lambda m : \mathrm{Nat}.\, \mathrm{if}\; n < m \;\mathrm{then\;abort\;else}\; n - m \end{array}$$

$$\begin{array}{l} \mathit{checkDivides} \;:\; \mathrm{Nat} \to \mathrm{Nat} \to \mathrm{Bool} \\ \quad \equiv\; \lambda n : \mathrm{Nat}.\, \mathrm{if}\; n = 0 \;\mathrm{then\;abort\;else} \\ \qquad \mathrm{fix}\,(\lambda f : \mathrm{Nat} \to \mathrm{Bool}.\, \lambda m : \mathrm{Nat}. \\ \qquad\quad \mathrm{if}\; m = 0 \;\mathrm{then}\; \mathit{true} \;\mathrm{else} \\ \qquad\qquad \mathrm{if}\; m < n \;\mathrm{then}\; \mathit{false} \;\mathrm{else}\; f\,(m - n)) \end{array}$$

Εξαιρέσεις (iv)

- Χειρισμός εξαιρέσεων (exception handling)
- Εξαιρέσεις που μεταφέρουν πληροφορία
- Σύνταξη (τύποι, εκφράσεις)

$$\begin{array}{lll} \tau & ::= & \ldots \mid \mathrm{Exn} \\ e & ::= & \ldots \mid \mathrm{throw}\, e \mid \mathrm{try}\, e_1\, \mathrm{catch}\, e_2 \end{array}$$

- Λειτουργική σημασιολογία (call by value)

$$(\mathrm{throw}\, v)\, e \longrightarrow \mathrm{throw}\, v \qquad v_1\, (\mathrm{throw}\, v_2) \longrightarrow \mathrm{throw}\, v_2$$

$$\mathrm{throw}\, (\mathrm{throw}\, v) \longrightarrow \mathrm{throw}\, v \qquad \frac{e \longrightarrow e'}{\mathrm{throw}\, e \longrightarrow \mathrm{throw}\, e'}$$

Εξαιρέσεις (v)

- Λειτουργική σημασιολογία (συνέχεια)

$$\mathrm{try}\, v\, \mathrm{catch}\, e \longrightarrow v \qquad \mathrm{try}\, \mathrm{throw}\, v\, \mathrm{catch}\, e \longrightarrow e\, v$$

$$\frac{e_1 \longrightarrow e_1'}{\mathrm{try}\, e_1\, \mathrm{catch}\, e_2 \longrightarrow \mathrm{try}\, e_1'\, \mathrm{catch}\, e_2}$$

- Κανόνες τύπων

$$\frac{\Gamma \vdash e : \mathrm{Exn}}{\Gamma \vdash \mathrm{throw}\, e : \tau} \qquad \frac{\Gamma \vdash e_1 : \tau \qquad \Gamma \vdash e_2 : \mathrm{Exn} \to \tau}{\Gamma \vdash \mathrm{try}\, e_1\, \mathrm{catch}\, e_2 : \tau}$$

Εξαιρέσεις (vi)

- Παραδείγματα

$$\begin{array}{l} \mathit{checkSubtract} \;:\; \mathrm{Nat} \to \mathrm{Nat} \to \mathrm{Nat} \\ \quad \equiv\; \lambda n : \mathrm{Nat}.\, \lambda m : \mathrm{Nat}. \\ \qquad \mathrm{if}\ n < m\ \mathrm{then}\ \mathrm{throw}\ \mathit{exn\_sub}\ \mathrm{else}\ n - m \end{array}$$

$$\begin{array}{l} \mathit{checkDivides} \;:\; \mathrm{Nat} \to \mathrm{Nat} \to \mathrm{Bool} \\ \quad \equiv\; \lambda n : \mathrm{Nat}.\, \mathrm{if}\ n = 0\ \mathrm{then}\ \mathrm{throw}\ \mathit{exn\_divzero}\ \mathrm{else} \\ \qquad \mathrm{fix}\,(\lambda f : \mathrm{Nat} \to \mathrm{Bool}.\, \lambda m : \mathrm{Nat}. \\ \qquad\quad \mathrm{if}\ m = 0\ \mathrm{then}\ \mathit{true}\ \mathrm{else} \\ \qquad\qquad \mathrm{try}\ f\,(\mathit{checkSubtract}\ m\ n) \\ \qquad\qquad\quad \mathrm{catch}\ \lambda \mathit{exn} : \mathrm{Exn}.\, \mathit{false}) \end{array}$$

Υποτύποι (i)

- Σχέση υποτύπων (subtype relation) $\quad \tau <: \tau'$
  Ο τύπος τ είναι υποτύπος του τ' όταν κάθε έκφραση τύπου τ μπορεί να χρησιμοποιηθεί σε μια θέση όπου αναμένεται έκφραση τύπου τ'
- Κανόνας τύπων (subsumption)

$$\frac{\Gamma \vdash e : \tau \qquad \tau <: \tau'}{\Gamma \vdash e : \tau'}$$

- Παραδείγματα

$$\mathrm{Nat} \;<:\; \mathrm{Int} \;<:\; \mathrm{Real}$$

$$\langle x : \mathrm{Nat}, y : \mathrm{Bool} \rangle \;<:\; \langle x : \mathrm{Nat} \rangle$$

Υποτύποι (ii)

- Σύνταξη (τύποι)

$$\tau \;::=\; \ldots \mid \mathrm{Top} \mid \mathrm{Bot}$$

- Σχέση υποτύπων

$$\tau <: \tau \qquad \frac{\tau_1 <: \tau_2 \qquad \tau_2 <: \tau_3}{\tau_1 <: \tau_3}$$

$$\tau <: \mathrm{Top} \qquad \mathrm{Bot} <: \tau$$

- Η σχέση υποτύπων αλληλεπιδρά με τους υπάρχοντες τύπους της γλώσσας κατά πολύπλοκο τρόπο

Υποτύποι (iii)

- Σχέση υποτύπων: συναρτήσεις

$$\frac{\tau_1' <: \tau_1 \qquad \tau_2 <: \tau_2'}{\tau_1 \to \tau_2 <: \tau_1' \to \tau_2'}$$

- Σχέση υποτύπων: εγγραφές

$$\langle x_i : \tau_i \mid 1 \le i \le n+k \rangle <: \langle x_i : \tau_i \mid 1 \le i \le n \rangle$$

$$\frac{\langle x_i : \tau_i \mid 1 \le i \le n \rangle \text{ μετάθεση του } \langle x_i' : \tau_i' \mid 1 \le i \le n \rangle}{\langle x_i : \tau_i \mid 1 \le i \le n \rangle <: \langle x_i' : \tau_i' \mid 1 \le i \le n \rangle}$$

$$\frac{\tau_i <: \tau_i' \text{ για κάθε } i}{\langle x_i : \tau_i \mid 1 \le i \le n \rangle <: \langle x_i : \tau_i' \mid 1 \le i \le n \rangle}$$

Υποτύποι (iv)

- Σχέση υποτύπων: αναφορές

$$\frac{\tau <: \tau' \qquad \tau' <: \tau}{\mathrm{Ref}\,\tau <: \mathrm{Ref}\,\tau'}$$

- Παράδειγμα: δεδομένου ότι $\mathrm{Int} <: \mathrm{Real}$, δεν είναι επιθυμητό $\mathrm{Ref\,Int} <: \mathrm{Ref\,Real}$, π.χ.

$$(\lambda r : \mathrm{Ref\,Real}.\, r := 3.14)\ (\mathrm{ref}\ 1)$$

- Παράδειγμα: χρήση του Bot

$$\Gamma \vdash \mathrm{abort} : \mathrm{Bot}$$

Υποτύποι (v)

- Τύποι τομής (intersection types)
- Σύνταξη (τύποι)

$$\tau \ ::= \ \ldots \mid \tau_1 \wedge \tau_2$$

- Σχέση υποτύπων: τύποι τομής

$$\tau_1 \wedge \tau_2 <: \tau_1 \qquad \tau_1 \wedge \tau_2 <: \tau_2 \qquad \frac{\tau <: \tau_1 \qquad \tau <: \tau_2}{\tau <: \tau_1 \wedge \tau_2}$$

$$(\tau \to \tau_1) \wedge (\tau \to \tau_2) <: \tau \to (\tau_1 \wedge \tau_2)$$

- Παράδειγμα: υπερφόρτωση τελεστών

$$+ \ : \ (\mathrm{Int} \to \mathrm{Int} \to \mathrm{Int}) \wedge (\mathrm{Real} \to \mathrm{Real} \to \mathrm{Real})$$

Υποτύποι (vi)

- Αναθεώρηση των αναφορών: πηγές (sources) και υποδοχείς (acceptors)
- Σύνταξη (τύποι)

$$\tau \ ::= \ \ldots \mid \mathrm{Src}\,\tau \mid \mathrm{Acc}\,\tau$$

- Κανόνες τύπων

$$\frac{\Gamma \vdash e : \mathrm{Src}\,\tau}{\Gamma \vdash\, !e : \tau} \qquad \frac{\Gamma \vdash e_1 : \mathrm{Acc}\,\tau \qquad \Gamma \vdash e_2 : \tau}{\Gamma \vdash e_1 := e_2 : \mathrm{Unit}}$$

$$\frac{\Gamma \vdash e : \tau}{\Gamma \vdash \mathrm{ref}\, e : \mathrm{Src}\,\tau \wedge \mathrm{Acc}\,\tau}$$

Υποτύποι (vii)

- Σχέση υποτύπων: πηγές και υποδοχείς

$$\frac{\tau <: \tau'}{\mathrm{Src}\,\tau <: \mathrm{Acc}\,\tau'} \qquad \frac{\tau' <: \tau}{\mathrm{Acc}\,\tau <: \mathrm{Acc}\,\tau'}$$

- Syntactic sugar: $\mathrm{Ref}\,\tau \equiv \mathrm{Src}\,\tau \wedge \mathrm{Acc}\,\tau$, οπότε

$$\mathrm{Ref}\,\tau <: \mathrm{Src}\,\tau \qquad \mathrm{Ref}\,\tau <: \mathrm{Acc}\,\tau$$

Υποτύποι (viii)

- Τύποι ένωσης (union types)
- Σύνταξη (τύποι)

$$\tau \ ::= \ \ldots \mid \tau_1 \vee \tau_2$$

- Σχέση υποτύπων: τύποι ένωσης

$$\tau_1 <: \tau_1 \vee \tau_2 \qquad \tau_2 <: \tau_1 \vee \tau_2 \qquad \frac{\tau_1 <: \tau \qquad \tau_2 <: \tau}{\tau_1 \vee \tau_2 <: \tau}$$

$$(\tau_1 \to \tau) \wedge (\tau_2 \to \tau) <: (\tau_1 \vee \tau_2) \to \tau$$

$$(\tau_1 \wedge \tau_2) \vee \tau <: (\tau_1 \vee \tau) \wedge (\tau_2 \vee \tau)$$

$$(\tau_1 \vee \tau_2) \wedge \tau <: (\tau_1 \wedge \tau) \vee (\tau_2 \wedge \tau)$$

Αναδρομικοί τύποι (i)

- Παράδειγμα: λίστες φυσικών αριθμών

$$\mathrm{list\_Nat} \ \simeq \ \mathrm{Unit} + (\mathrm{Nat} \times \mathrm{list\_Nat})$$

- Φορμαλισμός: $\mathrm{list\_Nat} \ \equiv \ \mu\alpha.\, \mathrm{Unit} + (\mathrm{Nat} \times \alpha)$
- Υλοποίηση τελεστών

$$\begin{array}{rcl}
\mathit{nil_Nat} & : & \mathrm{list_Nat} \\
 & \equiv & \mathrm{inl}\ \mathit{unit} \\
\mathit{cons_Nat} & : & \mathrm{Nat} \to \mathrm{list_Nat} \to \mathrm{list_Nat} \\
 & \equiv & \lambda h : \mathrm{Nat}.\, \lambda t : \mathrm{list_Nat}.\, \mathrm{inr}\, \langle h, t \rangle \\
\mathit{length_Nat} & : & \mathrm{list_Nat} \to \mathrm{Nat} \\
 & \equiv & \mathrm{fix}\, (\lambda f : \mathrm{list_Nat} \to \mathrm{Nat}.\, \lambda \ell : \mathrm{list_Nat}. \\
 & & \quad [\lambda u : \mathrm{Unit}.\, 0, \lambda p : \mathrm{Nat} \times \mathrm{list_Nat}.\, 1 + f\, (\mathrm{snd}\, p)]\ \ell)
\end{array}$$

Αναδρομικοί τύποι (ii)

- Πρόβλημα: ο προηγούμενος ορισμός δίνει

$$nil\_Nat \;:\; \mathrm{Unit} + (\mathrm{Nat} \times \mathrm{list\_Nat})$$

αντί του επιθυμητού

$$nil\_Nat \;:\; \mathrm{list\_Nat}$$

- Λύση 1: ισότητα τύπων (equi-recursive)

$$\mu\alpha.\,\tau \;<:\; \tau[\alpha := \mu\alpha.\,\tau]$$
$$\tau[\alpha := \mu\alpha.\,\tau] \;<:\; \mu\alpha.\,\tau$$

- Λύση 2: ισομορφισμός τύπων (iso-recursive)

$$\begin{aligned} \mathrm{unfold}_{\mu\alpha.\,\tau} &: (\mu\alpha.\,\tau) \to \tau[\alpha := \mu\alpha.\,\tau] \\ \mathrm{fold}_{\mu\alpha.\,\tau} &: \tau[\alpha := \mu\alpha.\,\tau] \to (\mu\alpha.\,\tau) \end{aligned}$$

Αναδρομικοί τύποι (ii)

- Ισο-αναδρομικοί τύποι (iso-recursive types)
- Σύνταξη (τύποι, εκφράσεις, τιμές)

$$\begin{aligned} \tau &::= \ldots \mid \alpha \mid \mu\alpha.\,\tau \\ e &::= \ldots \mid \mathrm{unfold}_\tau(e) \mid \mathrm{fold}_\tau(e) \\ v &::= \ldots \mid \mathrm{fold}_\tau(v) \end{aligned}$$

- Λειτουργική σημασιολογία

$$\mathrm{unfold}_\tau(\mathrm{fold}_{\tau'}(v)) \longrightarrow v$$

Αναδρομικοί τύποι (iii)

- Κανόνες τύπων

$$\frac{\tau' = \mu\alpha.\,\tau \qquad \Gamma \vdash e : \tau[\alpha := \tau']}{\Gamma \vdash \mathrm{fold}_{\tau'}(e) : \tau'}$$

$$\frac{\tau' = \mu\alpha.\,\tau \qquad \Gamma \vdash e : \tau'}{\Gamma \vdash \mathrm{unfold}_{\tau'}(e) : \tau[\alpha := \tau']}$$

Αναδρομικοί τύποι (iv)

- Παράδειγμα: λίστες φυσικών αριθμών

$$\begin{aligned} \mathrm{list\_Nat} &\equiv \mu\alpha.\,\mathrm{Unit} + (\mathrm{Nat} \times \alpha) \\ nil\_Nat &: \mathrm{list\_Nat} \\ &\equiv \mathrm{fold}_{\mathrm{list\_Nat}}(\mathrm{inl}\ unit) \\ cons\_Nat &: \mathrm{Nat} \to \mathrm{list\_Nat} \to \mathrm{list\_Nat} \\ &\equiv \lambda h : \mathrm{Nat}.\,\lambda t : \mathrm{list\_Nat}.\,\mathrm{fold}_{\mathrm{list\_Nat}}(\mathrm{inr}\ \langle h, t\rangle) \\ length\_Nat &: \mathrm{list\_Nat} \to \mathrm{Nat} \\ &\equiv \mathrm{fix}\,(\lambda f : \mathrm{list\_Nat} \to \mathrm{Nat}.\,\lambda \ell : \mathrm{list\_Nat}. \\ &\qquad [\lambda u : \mathrm{Unit}.\,0, \lambda p : \mathrm{Nat} \times \mathrm{list\_Nat}.\,1 + f\,(\mathrm{snd}\,p)] \\ &\qquad\quad \mathrm{unfold}_{\mathrm{list\_Nat}}(\ell)) \end{aligned}$$

Αναδρομικοί τύποι (v)

- Αναδρομικοί τύποι και κανονικοποίηση

$$\begin{aligned} fix_\tau &: (\tau \to \tau) \to \tau \\ &\equiv \lambda f : \tau \to \tau. \\ &\qquad (\lambda x : (\mu\alpha.\,\alpha \to \tau).\,f\,(x\,x)) \\ &\qquad\quad (\lambda x : (\mu\alpha.\,\alpha \to \tau).\,f\,(x\,x)) \\ diverge_\tau &: \tau \\ &\equiv fix_\tau\,(\lambda x : \tau.\,x) \end{aligned}$$

- Σχέση υποτύπων: αναδρομικοί τύποι

$$\frac{\Delta, \alpha <: \alpha' \vdash \tau <: \tau'}{\Delta \vdash \mu\alpha.\,\tau <: \mu\alpha'.\,\tau'}$$

Πολυμορφισμός 2ης τάξης (i)

- Παράδειγμα: ταυτοτική συνάρτηση

$$\begin{aligned} id &: \forall\alpha.\ \alpha \to \alpha \\ &\equiv \Lambda\alpha.\,\lambda x : \alpha.\,x \end{aligned}$$

- Παράδειγμα: σύνθεση συναρτήσεων

$$\begin{aligned} comp &: \forall\alpha.\ \forall\beta.\ \forall\gamma. \\ &\qquad (\beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma \\ &\equiv \Lambda\alpha.\,\Lambda\beta.\,\Lambda\gamma. \\ &\qquad \lambda f : \beta \to \gamma.\,\lambda g : \alpha \to \beta.\,\lambda x : \alpha.\,f\,(g\,x) \end{aligned}$$

- Χρήση αυτών

$$id\,[\mathrm{Nat}]\,42$$
$$comp\,[\mathrm{Real}]\,[\mathrm{Int}]\,[\mathrm{Nat}]\ round\ abs\ (-41.87)$$

Πολυμορφισμός 2ης τάξης (ii)

- Παράδειγμα: πολυμορφικές λίστες
  - Τύπος στοιχείου λίστας: α

$$\begin{array}{rcl} \mathit{nil} & : & \forall\alpha.\ \mu\beta.\,\mathrm{Unit} + (\alpha \times \beta) \\ & \equiv & \Lambda\alpha.\,\mathrm{fold}_{\mu\beta.\,\mathrm{Unit}+(\alpha\times\beta)}(\mathrm{inl}\ \mathit{unit}) \\ \mathit{cons} & : & \forall\alpha.\ \alpha \to (\mu\beta.\,\mathrm{Unit} + (\alpha \times \beta)) \to \\ & & \quad (\mu\beta.\,\mathrm{Unit} + (\alpha \times \beta)) \\ & \equiv & \Lambda\alpha.\,\lambda h : \alpha.\,\lambda t : (\mu\beta.\,\mathrm{Unit} + (\alpha \times \beta)). \\ & & \quad \mathrm{fold}_{\mu\beta.\,\mathrm{Unit}+(\alpha\times\beta)}(\mathrm{inr}\ \langle h, t\rangle) \end{array}$$

  - Χρήση: δημιουργία λίστας $[42, 7]$

$$\begin{array}{rcl} \ell & : & \mu\beta.\,\mathrm{Unit} + (\mathrm{Nat} \times \beta) \\ & \equiv & \mathit{cons}\,[\mathrm{Nat}]\,42\,(\mathit{cons}\,[\mathrm{Nat}]\,7\,(\mathit{nil}\,[\mathrm{Nat}])) \end{array}$$

Πολυμορφισμός 2ης τάξης (iii)

- Σύστημα F ή F_2 ή πολυμορφικός λ-λογισμός 2ης τάξης
- Σύνταξη (τύποι, εκφράσεις, τιμές)

$$\begin{array}{rcl} \tau & ::= & \ldots \mid \alpha \mid \forall\alpha.\ \tau \\ e & ::= & \ldots \mid \Lambda\alpha.\,e \mid e\,[\tau] \\ v & ::= & \ldots \mid \Lambda\alpha.\,v \end{array}$$

- Λειτουργική σημασιολογία

$$(\Lambda\alpha.\,e)\,[\tau] \longrightarrow e[\alpha := \tau] \qquad \frac{e \longrightarrow e'}{e\,[\tau] \longrightarrow e'\,[\tau]}$$

Πολυμορφισμός 2ης τάξης (iv)

- Περιβάλλοντα τύπων Γ: σύνολα με στοιχεία
  - της μορφής (x, τ), ή
  - της μορφής α
- Κανόνες τύπων

$$\frac{\Gamma, \alpha \vdash e : \tau}{\Gamma \vdash \Lambda\alpha.\,e : \forall\alpha.\ \tau} \qquad \frac{\Gamma \vdash e : \forall\alpha.\ \tau}{\Gamma \vdash e\,[\tau'] : \tau[\alpha := \tau']}$$

Πολυμορφισμός ω-τάξης (i)

- Παράδειγμα: πολυμορφικές λίστες

$$\begin{array}{rcl} \mathrm{list} & :: & * \Rightarrow * \\ & \equiv & \lambda\alpha :: *.\,\mu\beta.\,\mathrm{Unit} + (\alpha \times \beta) \\ \mathit{nil} & : & \forall\alpha :: *.\ \mathrm{list}\,\alpha \\ & \equiv & \Lambda\alpha :: *.\,\mathrm{fold}_{\mathrm{list}\,\alpha}(\mathrm{inl}\ \mathit{unit}) \\ \mathit{cons} & : & \forall\alpha :: *.\ \alpha \to \mathrm{list}\,\alpha \to \mathrm{list}\,\alpha \\ & \equiv & \Lambda\alpha :: *.\,\lambda h : \alpha.\,\lambda t : \mathrm{list}\,\alpha.\,\mathrm{fold}_{\mathrm{list}\,\alpha}(\mathrm{inr}\ \langle h, t\rangle) \\ \mathit{length} & : & \forall\alpha :: *.\ \mathrm{list}\,\alpha \to \mathrm{Nat} \\ & \equiv & \Lambda\alpha :: *.\,\mathrm{fix}\,(\lambda f : \mathrm{list}\,\alpha \to \mathrm{Int}.\,\lambda\ell : \mathrm{list}\,\alpha. \\ & & \quad [\lambda u : \mathrm{Unit}.\,0, \lambda p : \alpha \times \mathrm{list}\,\alpha.\,1 + f\,(\mathrm{snd}\,p)] \\ & & \qquad \mathrm{unfold}_{\mathrm{list}\,\alpha}(\ell)) \end{array}$$

Πολυμορφισμός ω-τάξης (ii)

- Παράδειγμα

$$\begin{array}{rcl} f & : & \forall m :: * \Rightarrow *.\ (\forall\alpha :: *.\ \alpha \to m\,\alpha) \to \\ & & \quad \forall\alpha :: *.\ \forall\beta :: *.\ (\alpha \times \beta) \to (m\,\alpha \times m\,\beta) \\ & = & \Lambda m :: * \Rightarrow *.\,\lambda g : (\forall\alpha :: *.\ \alpha \to m\,\alpha). \\ & & \quad \Lambda\alpha :: *.\,\Lambda\beta :: *.\,\lambda p : \alpha \times \beta. \\ & & \qquad \langle g\,[\alpha]\,(\mathrm{fst}\,p), g\,[\beta]\,(\mathrm{snd}\,p)\rangle \end{array}$$

- Χρήση της f

$$\begin{array}{rcl} g & \equiv & \Lambda\alpha :: *.\,\lambda x : \alpha.\,\mathit{cons}\,[\alpha]\,x\,(\mathit{nil}\,[\alpha]) \\ p & \equiv & f\,[\mathrm{list}]\,g\,[\mathrm{Nat}]\,[\mathrm{Bool}]\,\langle 42, \mathit{true}\rangle \end{array}$$

Πολυμορφισμός ω-τάξης (iii)

- Σύστημα F_ω
- Σύνταξη (είδη, τύποι, εκφράσεις, τιμές)

$$\begin{array}{rcl} \kappa & ::= & * \mid \kappa_1 \Rightarrow \kappa_2 \\ \tau & ::= & \ldots \mid \alpha \mid \forall\alpha :: \kappa.\ \tau \mid \lambda\alpha :: \kappa.\,\tau \mid \tau_1\,\tau_2 \\ e & ::= & \ldots \mid \Lambda\alpha :: \kappa.\,e \mid e\,[\tau] \\ v & ::= & \ldots \mid \Lambda\alpha :: \kappa.\,v \end{array}$$

- Λειτουργική σημασιολογία (εκφράσεις, τύποι)

$$(\Lambda\alpha :: \kappa.\,e)\,[\tau] \longrightarrow e[\alpha := \tau] \qquad \frac{e \longrightarrow e'}{e\,[\tau] \longrightarrow e'\,[\tau]}$$

$$(\lambda\alpha :: \kappa.\,\tau)\,\tau' \longrightarrow \tau[\alpha := \tau'] \qquad \text{κ.λπ.}$$

Πολυμορφισμός ω-τάξης (iv)

- Περιβάλλοντα τύπων Γ: σύνολα με στοιχεία
  - της μορφής (x, τ), ή
  - της μορφής (α, κ)
- Κανόνες τύπων (τύποι)

$$\frac{(\alpha, \kappa) \in \Gamma}{\Gamma \vdash \alpha :: \kappa} \qquad \Gamma \vdash \mathrm{Nat} :: * \qquad \Gamma \vdash \mathrm{Bool} :: *$$

$$\frac{\Gamma \vdash \tau_1 :: * \quad \Gamma \vdash \tau_2 :: *}{\Gamma \vdash \tau_1 \rightarrow \tau_2 :: *} \qquad \frac{\Gamma, \alpha :: \kappa \vdash \tau :: *}{\Gamma \vdash \forall \alpha :: \kappa.\ \tau :: *}$$

$$\frac{\Gamma, \alpha :: \kappa \vdash \tau :: \kappa'}{\Gamma \vdash \lambda \alpha :: \kappa. \tau :: \kappa \Rightarrow \kappa'} \qquad \frac{\Gamma \vdash \tau_1 :: \kappa \Rightarrow \kappa' \quad \Gamma \vdash \tau_2 :: \kappa}{\Gamma \vdash \tau_1\, \tau_2 :: \kappa'}$$

Πολυμορφισμός ω-τάξης (v)

- Ισότητα τύπων: ως $\tau = \tau'$ ορίζεται η σχέση ισοδυναμίας που προκύπτει από τη μετατροπή τύπων $\tau \longrightarrow \tau'$
- Κανόνες τύπων (εκφράσεις)

$$\frac{\Gamma \vdash \tau :: * \quad \Gamma, x : \tau \vdash e : \tau'}{\Gamma \vdash \lambda x : \tau. e : \tau \rightarrow \tau'} \qquad \frac{\Gamma \vdash e_1 : \tau \rightarrow \tau' \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash e_1\, e_2 : \tau'}$$

$$\frac{\Gamma, \alpha :: \kappa \vdash e : \tau}{\Gamma \vdash \Lambda \alpha :: \kappa. e : \forall \alpha :: \kappa.\ \tau} \qquad \frac{\Gamma \vdash e : \forall \alpha :: \kappa.\ \tau' \quad \Gamma \vdash \tau :: \kappa}{\Gamma \vdash e[\tau] : \tau'[\alpha := \tau]}$$

$$\frac{\Gamma \vdash e : \tau \quad \Gamma \vdash \tau' :: * \quad \tau = \tau'}{\Gamma \vdash e : \tau'}$$

Υπαρξιακοί τύποι (i)

- Σύνταξη (τύποι, εκφράσεις, τιμές)

$$\begin{array}{rcl} \tau & ::= & \ldots \mid \exists \alpha :: \kappa.\ \tau \\ e & ::= & \ldots \mid \mathrm{pack}(\alpha :: \kappa = \tau, e : \tau') \\ & \mid & \mathrm{open}\, e \,\mathrm{as}\, (\alpha :: \kappa, x : \tau) \,\mathrm{in}\, e' \\ v & ::= & \ldots \mid \mathrm{pack}(\alpha :: \kappa = \tau, v : \tau') \end{array}$$

- Λειτουργική σημασιολογία

$$\mathrm{open}\, \mathrm{pack}(\alpha :: \kappa = \tau, v : \tau') \,\mathrm{as}\, (\alpha' :: \kappa', x : \tau'') \,\mathrm{in}\, e \longrightarrow e[\alpha' := \tau][x := v]$$

Υπαρξιακοί τύποι (ii)

- Λειτουργική σημασιολογία (συνέχεια)

$$\frac{e \longrightarrow e'}{\mathrm{pack}(\alpha :: \kappa = \tau, e : \tau') \longrightarrow \mathrm{pack}(\alpha :: \kappa = \tau, e' : \tau')}$$

$$\frac{e_1 \longrightarrow e_1'}{\mathrm{open}\, e_1 \,\mathrm{as}\, (\alpha :: \kappa, x : \tau) \,\mathrm{in}\, e_2 \longrightarrow \mathrm{open}\, e_1' \,\mathrm{as}\, (\alpha :: \kappa, x : \tau) \,\mathrm{in}\, e_2}$$

- Κανόνες τύπων (τύποι)

$$\frac{\Gamma, \alpha :: \kappa \vdash \tau :: *}{\Gamma \vdash \exists \alpha :: \kappa.\ \tau :: *}$$

Υπαρξιακοί τύποι (iii)

- Κανόνες τύπων (εκφράσεις)

$$\frac{\Gamma \vdash \tau :: \kappa \quad \Gamma \vdash e : \tau'[\alpha := \tau]}{\Gamma \vdash \mathrm{pack}(\alpha :: \kappa = \tau, e : \tau') : \exists \alpha :: \kappa.\ \tau'}$$

$$\frac{\Gamma \vdash \tau' :: * \quad \Gamma \vdash e : \exists \alpha :: \kappa.\ \tau \quad \Gamma, \alpha :: \kappa, x : \tau \vdash e' : \tau'}{\Gamma \vdash \mathrm{open}\, e \,\mathrm{as}\, (\alpha :: \kappa, x : \tau) \,\mathrm{in}\, e' : \tau'}$$

Υπαρξιακοί τύποι (iv)

- Παράδειγμα: ροές (άπειρες λίστες)

$$\begin{array}{l} \mathrm{stream} :: * \Rightarrow * \\ \quad \equiv \lambda \alpha :: *. \exists \beta :: *.\ \beta \times ((\beta \rightarrow \alpha) \times (\beta \rightarrow \beta)) \end{array}$$

$$\begin{array}{l} \mathit{even} : \mathrm{stream\, Nat} \\ \quad \equiv \mathrm{pack}(\beta :: * = \mathrm{Nat}, \\ \qquad \langle 0, \langle \lambda n : \mathrm{Nat}. n, \lambda n : \mathrm{Nat}. n + 2 \rangle \rangle : \\ \qquad\quad \beta \times ((\beta \rightarrow \mathrm{Nat}) \times (\beta \rightarrow \beta))) \end{array}$$

Υπαρξιακοί τύποι (v)

- Παράδειγμα (συνέχεια)

$$
\begin{aligned}
\mathit{elem} \;:\; & \forall\alpha :: *.\ \mathrm{stream}\,\alpha \to \mathrm{Nat} \to \alpha \\
\equiv\; & \Lambda\alpha :: *.\ \lambda s : \mathrm{stream}\,\alpha. \\
& \quad \mathrm{open}\ s\ \mathrm{as}\ (\beta : *, i : \beta \times ((\beta \to \alpha) \times (\beta \to \beta)))\ \mathrm{in} \\
& \quad \mathrm{fix}\,(\lambda f : \beta \to \mathrm{Nat} \to \alpha.\ \lambda x : \beta.\ \lambda n : \mathrm{Nat}. \\
& \qquad \mathrm{if}\ n = 0\ \mathrm{then}\ \mathrm{fst}\,(\mathrm{snd}\,i)\,x \\
& \qquad\qquad \mathrm{else}\ \ f\,(\mathrm{snd}\,(\mathrm{snd}\,i)\,x)\,(n-1))\,(\mathrm{fst}\,i)
\end{aligned}
$$

- Χρήση

$$\mathit{elem}\ [\mathrm{Nat}]\ \mathit{even}\ 42$$

Υπαρξιακοί τύποι (vi)

- Παράδειγμα (συνέχεια)

$$
\begin{aligned}
\mathrm{strNatRep} \equiv\; & \mu\gamma.\ \mathrm{Ref}\,(\mathrm{Nat} \times (\mathrm{Unit} \to \gamma)) \\
\mathit{getData} \;:\; & \mathrm{strNatRep} \to \mathrm{Nat} \\
\equiv\; & \lambda r : \mathrm{strNatRep}.\ \mathrm{fst}\,(!\mathrm{unfold}_{\mathrm{strNatRep}}(r)) \\
\mathit{getNext} \;:\; & \mathrm{strNatRep} \to \mathrm{strNatRep} \\
\equiv\; & \lambda r : \mathrm{strNatRep}.\ \mathrm{snd}\,(!\mathrm{unfold}_{\mathrm{strNatRep}}(r))\ \mathit{unit} \\
\mathit{strEvenRep} \;:\; & \mathrm{strNatRep} \\
\equiv\; & \mathrm{fix}\,(\lambda f : \mathrm{Nat} \to \mathrm{strNatRep}.\ \lambda n : \mathrm{Nat}. \\
& \quad \mathrm{fold}_{\mathrm{strNatRep}}(\mathrm{ref}\,\langle n, \lambda u : \mathrm{Unit}.\ f\,(n+2)\rangle))\,0 \\
\mathit{even}' \;:\; & \mathrm{stream\ Nat} \\
\equiv\; & \mathrm{pack}(\beta :: * = \mathrm{strNatRep}, \\
& \quad \langle \mathit{strEvenRep}, \langle \mathit{getData}, \mathit{getNext}\rangle\rangle : \\
& \qquad \beta \times ((\beta \to \mathrm{Nat}) \times (\beta \to \beta)))
\end{aligned}
$$

Εξαρτώμενοι τύποι (i)

- Παράδειγμα: λίστες συγκεκριμένου μήκους

$$
\begin{aligned}
\mathrm{list} \;:\; & * \Rightarrow \mathrm{Nat} \Rightarrow * \\
\equiv\; & \lambda\alpha :: *.\ \lambda n : \mathrm{Nat}. \\
& \quad \mathrm{if}\ n = 0\ \mathrm{then}\ \mathrm{Unit}\ \mathrm{else}\ \alpha \times \mathrm{list}\,(n-1) \\
\mathit{nil} \;:\; & \forall\alpha :: *.\ \mathrm{list}\,a\,0 \\
\mathit{cons} \;:\; & \forall\alpha :: *.\ \Pi n : \mathrm{Nat}.\ \alpha \to \mathrm{list}\,a\,n \to \mathrm{list}\,a\,(n+1)
\end{aligned}
$$

- Τύποι που εξαρτώνται από εκφράσεις
  - Τύποι με παράμετρο έκφραση $\lambda x : \tau.\, \tau'$
  - Τύπος εξαρτώμενης συνάρτησης $\Pi x : \tau.\ \tau'$

Εξαρτώμενοι τύποι (ii)

- Σύνταξη (είδη, τύποι)

$$
\begin{aligned}
\kappa \;::=\; & \ldots \mid \tau \Rightarrow \kappa \\
\tau \;::=\; & \ldots \mid \Pi x : \tau.\ \tau' \mid \lambda x : \tau.\, \tau' \mid \tau\ e \\
\mid\; & \mathrm{if}\ e\ \mathrm{then}\ \tau_1\ \mathrm{else}\ \tau_2
\end{aligned}
$$

- Syntactic sugar: $\tau \to \tau' \;\equiv\; \Pi x : \tau.\ \tau'$
  αν $x \notin \mathrm{FV}(\tau')$ (μη εξαρτώμενη συνάρτηση)

Εξαρτώμενοι τύποι (iii)

- Κανόνες τύπων (τύποι)

$$\frac{\Gamma \vdash \tau :: * \qquad \Gamma, x : \tau \vdash \tau' :: *}{\Gamma \vdash \Pi x : \tau.\ \tau' :: *} \qquad \frac{\Gamma, x : \tau \vdash \tau' :: \kappa}{\Gamma \vdash \lambda x : \tau.\, \tau' :: \tau \Rightarrow \kappa}$$

$$\frac{\Gamma \vdash \tau :: \tau' \Rightarrow \kappa \qquad \Gamma \vdash e : \tau'}{\Gamma \vdash \tau\ e :: \kappa}$$

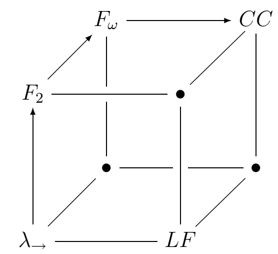
- Κανόνες τύπων (εκφράσεις)

$$\frac{\Gamma, x : \tau \vdash e : \tau' \qquad \Pi x : \tau.\ \tau' :: *}{\Gamma \vdash \lambda x : \tau.\, e : \Pi x : \tau.\ \tau'}$$

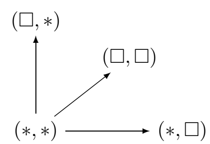
$$\frac{\Gamma \vdash e : \Pi x : \tau.\ \tau' \qquad \Gamma \vdash e' : \tau}{\Gamma \vdash e\, e' : \tau'[x := e']}$$

Εξαρτώμενοι τύποι (iv)

- Ο κύβος του Barendregt



Υπόμνημα
$*$: εκφράσεις, $\square$: τύποι



Π.χ. $(\square, *)$: εκφράσεις που εξαρτώνται από τύπους