Γλώσσες Προγραμματισμού II

http://courses.softlab.ntua.gr/pl2/

Κωστής Σαγώνας
kostis@cs.ntua.gr

Νίκος Παπασπύρου
nickie@softlab.ntua.gr

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

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

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

  - 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 $[\![\diamond]\!] : v \rightarrow v$ είναι η σημασία του

    π.χ. $[\![\neg]\!](\mathit{true}) \;=\; \mathit{false}$
  - Αν $\circ$ κάποιος τελεστής με δύο τελούμενα, τότε $[\![\circ]\!] : v \times v \rightarrow v$ 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! if 1 then *true* else *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 $e : \tau$

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

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

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

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

Βασικοί τύποι (viii)

- Παραγωγές τύπων (typing derivations)

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

- 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 $\tau = \mathrm{Bool}$, $e_1 : \mathrm{Int}$ και $e_2 : \mathrm{Int}$

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)

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

- 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 $\Gamma \vdash e' : \tau$

Απλές επεκτάσεις (i)

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

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

- 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

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

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

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

Απλές επεκτάσεις (iii)

- Απόδοση ονομάτων — δομή let
- Σύνταξη (εκφράσεις)

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

- 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)

$$\begin{array}{lcl} \tau & ::= & \ldots \mid \tau_1 \times \tau_2 \\ e & ::= & \ldots \mid \langle e_1, e_2 \rangle \mid \mathrm{fst}\, e \mid \mathrm{snd}\, e \end{array}$$

- Τιμές: πρόθυμα ζεύγη

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

- 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)

$$\begin{array}{lcl} \tau & ::= & \ldots \mid \tau_1 + \tau_2 \\ e & ::= & \ldots \mid \mathrm{inl}\, e \mid \mathrm{inr}\, e \mid [e_1, e_2] \end{array}$$

- 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

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

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

Αναδρομή (iv)

- Παράδειγμα: call by name

$$
\begin{array}{rcl}
p & \equiv & \text{fix}\,(\lambda f : \text{Int} \to \text{Int}.\, \lambda n : \text{Int}. \\
& & \qquad \text{if}\; n \leq 1 \;\text{then}\; 1 \;\text{else}\; n * f\,(n-1)) \\
p\,3 & \longrightarrow & (\lambda f : \text{Int} \to \text{Int}.\, \lambda n : \text{Int}. \\
& & \qquad \text{if}\; n \leq 1 \;\text{then}\; 1 \;\text{else}\; n * f\,(n-1))\; p\; 3 \\
& \longrightarrow & (\lambda n : \text{Int}.\, \text{if}\; n \leq 1 \;\text{then}\; 1 \;\text{else}\; n * p\,(n-1))\; 3 \\
& \longrightarrow & \text{if}\; 3 \leq 1 \;\text{then}\; 1 \;\text{else}\; 3 * p\,(3-1) \\
& \longrightarrow & \text{if}\; \mathit{false} \;\text{then}\; 1 \;\text{else}\; 3 * p\,(3-1) \\
& \longrightarrow & 3 * p\,(3-1) \\
& \longrightarrow & \ldots \\
& \longrightarrow & 3 * ((3-1) * (3-1-1)) \;\longrightarrow\; \ldots \;\longrightarrow\; 6
\end{array}
$$

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)

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

- 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

$$\begin{array}{l} \mathrm{letrec}\, x = e_1 \,\mathrm{in}\, e_2 \quad \equiv \\ \qquad \mathrm{let}\, x = \mathrm{fix}\,(\lambda x : \tau' .\, e_1)\,\mathrm{in}\, e_2 \end{array}$$

Αναφορές (i)

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

$$\begin{array}{lcl} \tau & ::= & \ldots \mid \mathrm{Ref}\,\tau \\ e & ::= & \ldots \mid \mathrm{ref}\,e \mid !e \mid e_1 := e_2 \end{array}$$

- 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)

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

- 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)

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

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

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

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

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)

$$e \quad ::= \quad \dots \mid \mathrm{abort}$$

- 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)

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

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

- 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 (iv)

- Λειτουργική σημασιολογία (συνέχεια)

$$\text{try}\ v\ \text{catch}\ e \longrightarrow v \qquad\qquad \text{try throw}\ v\ \text{catch}\ e \longrightarrow e\ v$$

$$\frac{e_1 \longrightarrow e_1'}{\text{try}\ e_1\ \text{catch}\ e_2 \longrightarrow \text{try}\ e_1'\ \text{catch}\ e_2}$$

- Κανόνες τύπων

$$\frac{\Gamma \vdash e : \text{Exn}}{\Gamma \vdash \text{throw}\ e : \tau} \qquad\qquad \frac{\Gamma \vdash e_1 : \tau \qquad \Gamma \vdash e_2 : \text{Exn} \rightarrow \tau}{\Gamma \vdash \text{try}\ e_1\ \text{catch}\ e_2 : \tau}$$

Εξαιρέσεις (v)

- Μηχανισμός δομημένων αλμάτων
- Παράδειγμα: memoization (σε OCaml)

```
let rec fib n =
  if n <= 1 then n else fib (n-1) + fib (n-2)

let tbl = Table.create 101
let rec memo_fib n =
  try
    Table.find tbl n
  with Not_found ->
    let v =
      if n <= 1 then n
      else memo_fib (n-1) + memo_fib (n-2) in
    Table.add tbl n v; v
```

Υποτύποι (i)

- Σχέση υποτύπων (subtype relation) $\tau <: \tau'$
  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

$$\text{Nat} \ <: \ \text{Int} \ <: \ \text{Real}$$

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

Υποτύποι (ii)

- Σύνταξη (τύποι)

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

- Σχέση υποτύπων

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

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

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

Υποτύποι (iii)

- Σχέση υποτύπων: συναρτήσεις

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

- Σχέση υποτύπων: εγγραφές

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

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

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

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)

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

- 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

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

Υποτύποι (vi)

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

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

- 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

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

- 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)

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

- 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

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

- Φορμαλισμός: $\quad \text{list\_Nat} \;\equiv\; \mu\alpha.\, \text{Unit} + (\text{Nat} \times \alpha)$

- Υλοποίηση τελεστών

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

Αναδρομικοί τύποι (ii)

- Πρόβλημα: ο προηγούμενος ορισμός δίνει

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

αντί του επιθυμητού

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

- Λύση 1: ισότητα τύπων (equi-recursive)

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

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

- Λύση 2: ισομορφισμός τύπων (iso-recursive)

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

Αναδρομικοί τύποι (iii)

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

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

- Λειτουργική σημασιολογία

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

Αναδρομικοί τύποι (iv)

- 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 (v)

- Παράδειγμα: λίστες φυσικών αριθμών

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

Αναδρομικοί τύποι (vi)

- Αναδρομικοί τύποι και κανονικοποίηση

$$\begin{array}{rcl} \mathit{fix}_\tau & : & (\tau \to \tau) \to \tau \\ & \equiv & \lambda f : \tau \to \tau. \\ & & \quad (\lambda x : (\mu\alpha.\, \alpha \to \tau).\, f\,(x\,x)) \\ & & \qquad (\lambda x : (\mu\alpha.\, \alpha \to \tau).\, f\,(x\,x)) \end{array}$$

$$\begin{array}{rcl} \mathit{diverge}_\tau & : & \tau \\ & \equiv & \mathit{fix}_\tau\,(\lambda x : \tau.\, x) \end{array}$$

- 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

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

- Παράδειγμα: σύνθεση συναρτήσεων

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

- 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

$$\begin{array}{rcl} \mathit{nil} & : & \forall\alpha.\ \mu\beta.\,\mathrm{Unit} + (\alpha \times \beta) \\ & \equiv & \Lambda\alpha.\,\mathrm{fold}_{\mu\beta.\,\mathrm{Unit}+(\alpha\times\beta)}(\mathrm{inl}\ \mathit{unit}) \end{array}$$

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

  - 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)

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

- 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 2ης τάξης (v)

- Σύστημα F: impredicative πολυμορφισμός
- impredicativity: ο τύπος τ στο $e\,[\tau]$ μπορεί να είναι πολυμορφικός
- predicativity: ο τύπος τ στο $e\,[\tau]$ πρέπει να είναι μονομορφικός
- impredicativity $\Rightarrow$ undecidable type inference
- let-polymorphism / monomorphic restriction στην ML και εν γένει στις γλώσσες με type inference και σύστημα τύπων Hindley-Milner

Πολυμορφισμός ω-τάξης (i)

- Παράδειγμα: πολυμορφικές λίστες

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

Πολυμορφισμός ω-τάξης (ii)

- Παράδειγμα

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

- Χρήση της f

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

Πολυμορφισμός ω-τάξης (iii)

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

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

- 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)

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

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

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

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)

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

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

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

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

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

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)

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

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

Υπαρξιακοί τύποι (iv)

- Παράδειγμα: ροές (άπειρες λίστες)

$$
\begin{aligned}
\text{stream} &:: * \Rightarrow * \\
&\equiv \lambda\alpha :: *.\, \exists\beta :: *.\ \beta \times ((\beta \to \alpha) \times (\beta \to \beta))
\end{aligned}
$$

$$
\begin{aligned}
\mathit{even} &: \text{stream}\,\text{Nat} \\
&\equiv \text{pack}(\beta :: * = \text{Nat}, \\
&\qquad\qquad \langle 0, \langle \lambda n : \text{Nat}.\, n, \lambda n : \text{Nat}.\, n + 2\rangle\rangle : \\
&\qquad\qquad\quad \beta \times ((\beta \to \text{Nat}) \times (\beta \to \beta)))
\end{aligned}
$$

Υπαρξιακοί τύποι (v)

- Παράδειγμα (συνέχεια)

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

  - Χρήση

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

Υπαρξιακοί τύποι (vi)

- Παράδειγμα (συνέχεια)

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

Εξαρτώμενοι τύποι (i)

- Παράδειγμα: λίστες συγκεκριμένου μήκους

$$\begin{aligned}\mathrm{list} \quad &:: \quad * \Rightarrow \mathrm{Nat} \Rightarrow * \\ &\equiv \quad \lambda\alpha :: *.\, \lambda n : \mathrm{Nat}. \\ &\qquad\quad \mathrm{if}\; n = 0 \;\mathrm{then}\; \mathrm{Unit} \;\mathrm{else}\; \alpha \times \mathrm{list}\,(n-1)\end{aligned}$$

$$\begin{aligned}\mathit{nil} \quad &: \quad \forall\alpha :: *.\ \mathrm{list}\, a\, 0 \\ \mathit{cons} \quad &: \quad \forall\alpha :: *.\ \Pi n : \mathrm{Nat}.\ \alpha \to \mathrm{list}\, a\, n \to \mathrm{list}\, a\, (n+1)\end{aligned}$$

- 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)

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

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

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

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

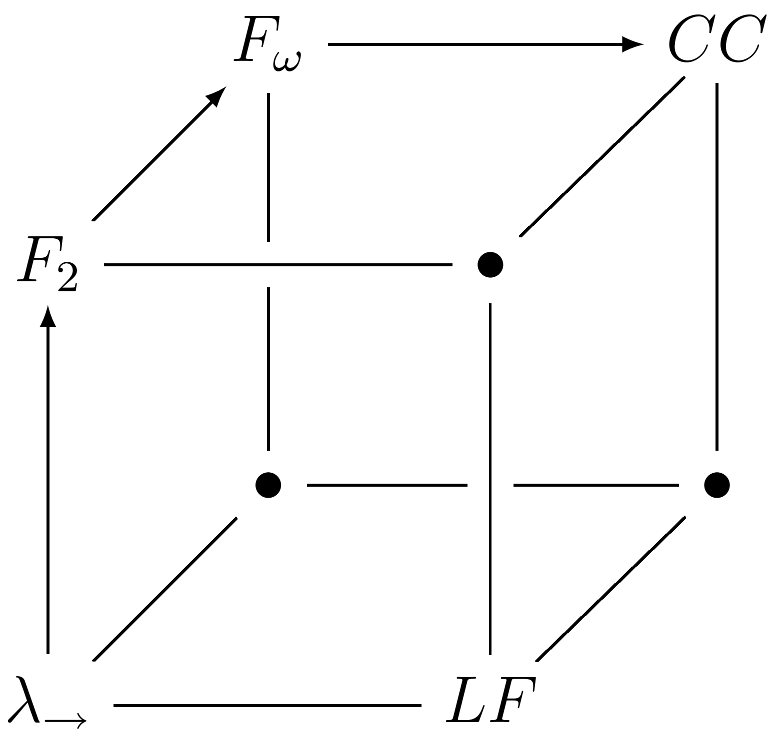
Εξαρτώμενοι τύποι (iv)

- Ο κύβος του Barendregt

<u>Υπόμνημα</u>

$*$: εκφράσεις, $\Box$: τύποι

$$\begin{tikzcd}
(\Box, *) & & \\
& (\Box, \Box) & \\
(*, *) \arrow[uu] \arrow[ur] \arrow[rr] & & (*, \Box)
\end{tikzcd}$$



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