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.

Σημασιολογία (i)

- Σύνταξη (syntax) και σημασιολογία (semantics)
- Παράδειγμα: σημασιολογία της εντολής

`while` ⟨λογική συνθήκη⟩ `do` ⟨εντολή⟩

"Αρχικά γίνεται ο έλεγχος της λογικής συνθήκης.
Αν το αποτέλεσμα είναι αληθές, τότε γίνεται είσοδος στο βρόχο και εκτελείται η εντολή μία φορά.
Στη συνέχεια η συνθήκη ελέγχεται και πάλι, κ.ο.κ.
Όταν η συνθήκη γίνει ψευδής, ο βρόχος παρακάμπτεται και ο έλεγχος μεταφέρεται στην πρώτη εντολή που ακολουθεί τη δομή του βρόχου."

Σημασιολογία (ii)

- Τυπική σημασιολογία: τρεις κύριες μέθοδοι
  - Λειτουργική (operational) σημασιολογία: η σημασία των προγραμμάτων περιγράφεται με μια σχέση μετάβασης μεταξύ των καταστάσεων μιας αφηρημένης μηχανής

    π.χ. $$\frac{(e, s) \longrightarrow v}{(x := e, s) \longrightarrow s[x := v]}$$

    $x \in Var, e \in Expr$ συντακτικοί όροι
    $s \in S = Var \to V$ η μνήμη της αφηρημένης μηχανής

Σημασιολογία (iii)

- Τυπική σημασιολογία (συνέχεια)
  - Δηλωτική (denotational) σημασιολογία: η σημασία των προγραμμάτων περιγράφεται μέσω μαθηματικών αντικειμένων, π.χ. συναρτήσεων που παίρνουν ως παραμέτρους τα δεδομένα του προγράμματος και υπολογίζουν τα αποτελέσματα

    π.χ. $$\begin{aligned} [\![e]\!] &: S \to V \\ [\![c]\!] &: S \to S \end{aligned}$$

    $$[\![x := e]\!](s) = s[x := [\![e]\!](s)]$$

Σημασιολογία (iv)

- Τυπική σημασιολογία (συνέχεια)
  - Αξιωματική (axiomatic) σημασιολογία: η σημασία των προγραμμάτων περιγράφεται έμμεσα ως το σύνολο των λογικών προτάσεων που μπορούν να αποδειχθούν για την εκτέλεση του προγράμματος

    π.χ. $\{P[x := e]\}\ x := e\ \{P\}$

    Αν πριν την εκτέλεση της ανάθεσης ισχύει $P[x := e]$, τότε μετά την εκτέλεση αυτής θα ισχύει P, όπου P κάποια λογική πρόταση

Δηλωτική σημασιολογία (i)

- Παράδειγμα: δυαδικές ακολουθίες
- Σύνταξη:

$$\mathbf{S} ::= 0 \mid 1 \mid \mathbf{S}0 \mid \mathbf{S}1$$

- Σημασιολογική συνάρτηση: $[\![\mathbf{S}]\!] : \mathbb{N}$

$$\begin{aligned} [\![0]\!] &= 0 & [\![\mathbf{S}0]\!] &= 2 \times [\![\mathbf{S}]\!] \\ [\![1]\!] &= 1 & [\![\mathbf{S}1]\!] &= 2 \times [\![\mathbf{S}]\!] + 1 \end{aligned}$$

Συνθεσιμότητα (compositionality)

Δηλωτική σημασιολογία (ii)

- Παράδειγμα: δυαδικές ακολουθίες (συνέχεια)

$$
\begin{aligned}
[\![1100]\!] &= 2 \times [\![110]\!] \\
&= 2 \times (2 \times [\![11]\!] \\
&= 2 \times (2 \times (2 \times [\![1]\!] + 1)) \\
&= 2 \times (2 \times (2 \times 1 + 1)) \\
&= 12
\end{aligned}
$$

Προστακτικές γλώσσες (i)

- Ζητούμενο: δηλωτική σημασιολογία για μια προστακτική γλώσσα εκφράσεων και εντολών
- Σύνταξη:

$$
\begin{aligned}
\mathbf{C} &::= \texttt{skip} \mid \mathbf{C}_0 ; \mathbf{C}_1 \mid \texttt{for } \mathbf{N} \texttt{ do } \mathbf{C} \\
&\mid \texttt{if } \mathbf{B} \texttt{ then } \mathbf{C}_0 \texttt{ else } \mathbf{C}_1 \\
\mathbf{B} &::= \texttt{true} \mid \texttt{not } \mathbf{B} \mid \mathbf{B}_0 \texttt{ and } \mathbf{B}_1 \\
&\mid \mathbf{N}_0 < \mathbf{N}_1 \mid \mathbf{N}_0 = \mathbf{N}_1 \mid \mathbf{B}_0 = \mathbf{B}_1 \\
&\mid \texttt{if } \mathbf{B} \texttt{ then } \mathbf{B}_0 \texttt{ else } \mathbf{B}_1 \\
\mathbf{N} &::= \texttt{0} \mid \texttt{succ } \mathbf{N} \\
&\mid \texttt{if } \mathbf{B} \texttt{ then } \mathbf{N}_0 \texttt{ else } \mathbf{N}_1
\end{aligned}
$$

Προστακτικές γλώσσες (ii)

- Κατάσταση (state): $s \in S$
- Σημασιολογικές συναρτήσεις

$$
\begin{aligned}
\mathcal{C}[\![\mathbf{C}]\!] &: S \to S \\
\mathcal{B}[\![\mathbf{B}]\!] &: S \to \{\, true, false \,\} \\
\mathcal{N}[\![\mathbf{N}]\!] &: S \to \mathbb{N}
\end{aligned}
$$

- Σημασιολογικές εξισώσεις

$$
\begin{aligned}
\mathcal{N}[\![\texttt{0}]\!]s &= 0 \\
\mathcal{N}[\![\texttt{succ } \mathbf{N}]\!]s &= \mathcal{N}[\![\mathbf{N}]\!]s + 1
\end{aligned}
$$

Προστακτικές γλώσσες (iii)

- Σημασιολογικές εξισώσεις (συνέχεια)

$$
\mathcal{B}[\![\texttt{true}]\!]s = true
$$

$$
\mathcal{B}[\![\texttt{not } \mathbf{B}]\!]s = \begin{cases} true, & \text{αν } \mathcal{B}[\![\mathbf{B}]\!]s = false \\ false, & \text{αν } \mathcal{B}[\![\mathbf{B}]\!]s = true \end{cases}
$$

$$
\mathcal{B}[\![\mathbf{B}_0 \texttt{ and } \mathbf{B}_1]\!]s = \begin{cases} true, & \text{αν } \mathcal{B}[\![\mathbf{B}_0]\!]s = \mathcal{B}[\![\mathbf{B}_1]\!]s = true \\ false, & \text{διαφορετικά} \end{cases}
$$

Προστακτικές γλώσσες (iv)

- Σημασιολογικές εξισώσεις (συνέχεια)

$$
\mathcal{B}[\![\mathbf{N}_0 < \mathbf{N}_1]\!]s = \begin{cases} true, & \text{αν } \mathcal{N}[\![\mathbf{N}_0]\!]s < \mathcal{N}[\![\mathbf{N}_1]\!]s \\ false, & \text{διαφορετικά} \end{cases}
$$

$$
\mathcal{B}[\![\mathbf{N}_0 = \mathbf{N}_1]\!]s = \begin{cases} true, & \text{αν } \mathcal{N}[\![\mathbf{N}_0]\!]s = \mathcal{N}[\![\mathbf{N}_1]\!]s \\ false, & \text{διαφορετικά} \end{cases}
$$

$$
\mathcal{B}[\![\mathbf{B}_0 = \mathbf{B}_1]\!]s = \begin{cases} true, & \text{αν } \mathcal{B}[\![\mathbf{B}_0]\!]s = \mathcal{B}[\![\mathbf{B}_1]\!]s \\ false, & \text{διαφορετικά} \end{cases}
$$

Προστακτικές γλώσσες (v)

- Σημασιολογικές εξισώσεις (συνέχεια)

$$
\mathcal{N}[\![\texttt{if } \mathbf{B} \texttt{ then } \mathbf{N}_0 \texttt{ else } \mathbf{N}_1]\!]s = \begin{cases} \mathcal{N}[\![\mathbf{N}_0]\!]s, & \text{αν } \mathcal{B}[\![\mathbf{B}]\!]s = true \\ \mathcal{N}[\![\mathbf{N}_1]\!]s, & \text{διαφορετικά} \end{cases}
$$

$$
\mathcal{B}[\![\texttt{if } \mathbf{B} \texttt{ then } \mathbf{B}_0 \texttt{ else } \mathbf{B}_1]\!]s = \begin{cases} \mathcal{B}[\![\mathbf{B}_0]\!]s, & \text{αν } \mathcal{B}[\![\mathbf{B}]\!]s = true \\ \mathcal{B}[\![\mathbf{B}_1]\!]s, & \text{διαφορετικά} \end{cases}
$$

$$
\mathcal{C}[\![\texttt{if } \mathbf{B} \texttt{ then } \mathbf{C}_0 \texttt{ else } \mathbf{C}_1]\!]s = \begin{cases} \mathcal{C}[\![\mathbf{C}_0]\!]s, & \text{αν } \mathcal{B}[\![\mathbf{B}]\!]s = true \\ \mathcal{C}[\![\mathbf{C}_1]\!]s, & \text{διαφορετικά} \end{cases}
$$

Προστακτικές γλώσσες (vi)

- Σημασιολογικές εξισώσεις (συνέχεια)

$$\begin{aligned} \mathcal{C}[\![\texttt{skip}]\!]s &= s \\ \mathcal{C}[\![\mathbf{C}_0;\mathbf{C}_1]\!]s &= \mathcal{C}[\![\mathbf{C}_1]\!](\mathcal{C}[\![\mathbf{C}_0]\!]s) \\ \mathcal{C}[\![\texttt{for } \mathbf{N} \texttt{ do } \mathbf{C}]\!]s &= (\mathcal{C}[\![\mathbf{C}]\!])^n(s),\ \text{όπου } n = \mathcal{N}[\![N]\!]s \end{aligned}$$

- Αυτή η γλώσσα είναι τετριμμένη: εύκολα μπορεί να δειχθεί (με επαγωγή) ότι για κάθε εντολή $\mathbf{C}$ και κατάσταση s ισχύει $\mathcal{C}[\![\mathbf{C}]\!]s = s$

Προστακτικές γλώσσες (vii)

- Μεταβλητές και αναθέσεις
- Σύνταξη $\quad \mathbf{i} \in \mathit{Var}, \pi(\mathbf{i}) \in \{\, \mathit{natural}, \mathit{boolean} \,\}$

$$\begin{aligned} \mathbf{N} \ &::= \ \ldots \\ &\ \ |\ \ \mathbf{i} && \text{όπου } \pi(\mathbf{i}) = \mathit{natural} \\ \mathbf{B} \ &::= \ \ldots \\ &\ \ |\ \ \mathbf{i} && \text{όπου } \pi(\mathbf{i}) = \mathit{boolean} \\ \mathbf{C} \ &::= \ \ldots \\ &\ \ |\ \ \mathbf{i} := \mathbf{N} && \text{όπου } \pi(\mathbf{i}) = \mathit{natural} \\ &\ \ |\ \ \mathbf{i} := \mathbf{B} && \text{όπου } \pi(\mathbf{i}) = \mathit{boolean} \end{aligned}$$

Προστακτικές γλώσσες (viii)

- Καταστάσεις

$$\begin{aligned} S = \{\ & s : \mathit{Var} \to \mathbb{N} \cup \{\, \mathit{true}, \mathit{false} \,\} \\ |\ & s(\mathbf{i}) \in \mathbb{N} \quad \text{αν} \quad \pi(\mathbf{i}) = \mathit{natural} \quad \text{και} \\ & s(\mathbf{i}) \in \{\, \mathit{true}, \mathit{false} \,\} \quad \text{αν} \quad \pi(\mathbf{i}) = \mathit{boolean} \quad \} \end{aligned}$$

- Σημασιολογικές εξισώσεις

$$\begin{aligned} \mathcal{N}[\![\mathbf{i}]\!]s &= s(\mathbf{i}) \\ \mathcal{C}[\![\mathbf{i} := \mathbf{N}]\!]s &= s[\mathbf{i} := \mathcal{N}[\![\mathbf{N}]\!]s] \\ \mathcal{B}[\![\mathbf{i}]\!]s &= s(\mathbf{i}) \\ \mathcal{C}[\![\mathbf{i} := \mathbf{B}]\!]s &= s[\mathbf{i} := \mathcal{B}[\![\mathbf{B}]\!]s] \end{aligned}$$

Προστακτικές γλώσσες (ix)

- Αόριστες επαναλήψεις, εντολή `while`
- Σύνταξη

$$\mathbf{C} \ ::= \ \ldots \ |\ \texttt{while } \mathbf{B} \texttt{ do } \mathbf{C}$$

- Ενδεχόμενο μη τερματισμού
- Μερικές σημασιολογικές συναρτήσεις

$$\mathcal{C}[\![\mathbf{C}]\!] \ : \ S \rightharpoonup S$$

$\mathcal{C}[\![\mathbf{C}]\!]s$ μη ορισμένο αν η εκτέλεση της εντολής $\mathbf{C}$ με αρχική κατάσταση s δεν τερματίζεται

Προστακτικές γλώσσες (x)

- Σημασιολογία της εντολής `while`
- Δύο εσφαλμένοι ορισμοί

$$\mathcal{C}[\![\texttt{while } \mathbf{B} \texttt{ do } \mathbf{C}]\!] = \mathcal{C}[\![\texttt{if } \mathbf{B} \texttt{ then } (\mathbf{C}; \texttt{while } \mathbf{B} \texttt{ do } \mathbf{C})]\!]$$

$$\mathcal{C}[\![\texttt{while } \mathbf{B} \texttt{ do } \mathbf{C}]\!]s = \begin{cases} \mathcal{C}[\![\texttt{while } \mathbf{B} \texttt{ do } \mathbf{C}]\!](\mathcal{C}[\![\mathbf{C}]\!]s), & \text{αν } \mathcal{B}[\![\mathbf{B}]\!]s = \mathit{true} \\ s, & \text{αν } \mathcal{B}[\![\mathbf{B}]\!]s = \mathit{false} \end{cases}$$

Προστακτικές γλώσσες (xi)

- Σημασιολογία της εντολής `while` (συνέχεια)
- Ζητείται μια λύση c της εξίσωσης

$$c(s) = \begin{cases} c(\mathcal{C}[\![\mathbf{C}]\!]s), & \text{αν } \mathcal{B}[\![\mathbf{B}]\!]s = \mathit{true} \\ s, & \text{αν } \mathcal{B}[\![\mathbf{B}]\!]s = \mathit{false} \end{cases}$$

- Ένας σωστός ορισμός: $\mathcal{C}[\![\texttt{while } \mathbf{B} \texttt{ do } \mathbf{C}]\!] = c_\infty$

$$\begin{aligned} c_0(s) &= \text{μη ορισμένο} \\ c_{i+1}(s) &= \begin{cases} c_i(\mathcal{C}[\![\mathbf{C}]\!]s), & \text{αν } \mathcal{B}[\![\mathbf{B}]\!]s = \mathit{true} \\ s, & \text{αν } \mathcal{B}[\![\mathbf{B}]\!]s = \mathit{false} \end{cases} \end{aligned}$$

Προστακτικές γλώσσες (xii)

- Παράδειγμα `while (n > 0) do (n := prev n)`

$$
\begin{aligned}
c_0(s) &= \text{μη ορισμένο} \\
c_1(s) &= \begin{cases} \text{μη ορισμένο}, & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s > 0 \\ s, & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s = 0 \end{cases}
\end{aligned}
$$

Προστακτικές γλώσσες (xiii)

- Παράδειγμα `while (n > 0) do (n := prev n)`

$$
\begin{aligned}
c_2(s) &= \begin{cases} c_1(\mathcal{C}\llbracket \mathtt{n := prev\ n} \rrbracket s), & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s > 0 \\ s, & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s = 0 \end{cases} \\
&= \begin{cases} \text{μη ορισμένο}, & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s > 0 \\ & \text{και } \mathcal{N}\llbracket \mathtt{n} \rrbracket (\mathcal{C}\llbracket \mathtt{n := prev\ n} \rrbracket s) > 0 \\ \mathcal{C}\llbracket \mathtt{n := prev\ n} \rrbracket s, & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s > 0 \\ & \text{και } \mathcal{N}\llbracket \mathtt{n} \rrbracket (\mathcal{C}\llbracket \mathtt{n := prev\ n} \rrbracket s) = 0 \\ s, & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s = 0 \end{cases} \\
&= \begin{cases} \text{μη ορισμένο}, & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s > 0 \text{ και } \mathcal{N}\llbracket \mathtt{n} \rrbracket s \geq 2 \\ s[\mathtt{n} := \mathcal{N}\llbracket \mathtt{n} \rrbracket s - 1], & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s > 0 \text{ και } \mathcal{N}\llbracket \mathtt{n} \rrbracket s = 1 \\ s, & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s = 0 \end{cases}
\end{aligned}
$$

Προστακτικές γλώσσες (xiv)

- Παράδειγμα `while (n > 0) do (n := prev n)`

$$
\begin{aligned}
c_2(s) &= \begin{cases} \text{μη ορισμένο}, & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s > 0 \text{ και } \mathcal{N}\llbracket \mathtt{n} \rrbracket s \geq 2 \\ s[\mathtt{n} := \mathcal{N}\llbracket \mathtt{n} \rrbracket s - 1], & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s > 0 \text{ και } \mathcal{N}\llbracket \mathtt{n} \rrbracket s = 1 \\ s, & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s = 0 \end{cases} \\
&= \begin{cases} \text{μη ορισμένο}, & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s \geq 2 \\ s[\mathtt{n} := 0], & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s < 2 \end{cases}
\end{aligned}
$$

Επαγωγικά αποδεικνύεται ότι:

$$
\begin{aligned}
c_i(s) &= \begin{cases} \text{μη ορισμένο}, & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s \geq i \\ s[\mathtt{n} := 0], & \text{αν } \mathcal{N}\llbracket \mathtt{n} \rrbracket s < i \end{cases} \\
c_\infty(s) &= s[\mathtt{n} := 0] = \mathcal{C}\llbracket \mathtt{n := 0} \rrbracket s
\end{aligned}
$$

Θεωρία πεδίων (i)

- Scott και Strachey, τέλη δεκαετίας 1960
- Μερικώς διατεταγμένο σύνολο $(D, \sqsubseteq)$

$x \sqsubseteq x$	ανακλαστική
αν $x \sqsubseteq y$ και $y \sqsubseteq z$ τότε $x \sqsubseteq z$	μεταβατική
αν $x \sqsubseteq y$ και $y \sqsubseteq x$ τότε $x = y$	αντισυμμετρική

- ω-αλυσίδα (ω-chain) είναι μια ακολουθία $\{ d_i \in D \mid i \in \mathbb{N} \}$ τέτοια ώστε

$$d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq \ldots \sqsubseteq d_i \sqsubseteq d_{i+1} \sqsubseteq \ldots$$

Θεωρία πεδίων (ii)

- Το d είναι άνω όριο (upper bound) του $P \subseteq D$ αν $p \sqsubseteq d$ για κάθε $p \in P$
- Το d είναι ελάχιστο άνω όριο (least upper bound — lub) του $P \subseteq D$ αν είναι άνω όριο του P και $d \sqsubseteq d'$ για κάθε άνω όριο d' του P
- Το ελάχιστο άνω όριο του P (αν υπάρχει) είναι μοναδικό και συμβολίζεται με $\bigsqcup P$
- Το $(D, \sqsubseteq)$ είναι ω-πλήρες αν για κάθε ω-αλυσίδα υπάρχει ελάχιστο άνω όριο

$$\bigsqcup_{i=0}^{\infty} d_i \in D$$

Θεωρία πεδίων (iii)

- Ένα μερικώς διατεταγμένο σύνολο $(D, \sqsubseteq)$ που είναι ω-πλήρες ονομάζεται πεδίο (domain)

  Ο ορισμός της έννοιας του πεδίου ποικίλλει σημαντικά στη βιβλιογραφία! Ο παραπάνω είναι ένας από τους απλούστερους δυνατούς ορισμούς που επαρκεί για τις ανάγκες του μαθήματος.

- $\bot$ και $\top$: ελάχιστο και μέγιστο στοιχείο ενός πεδίου (αν υπάρχουν) $\bot \sqsubseteq x \sqsubseteq \top$

Θεωρία πεδίων (iv)

- Κατασκευές πεδίων: έστω το σύνολο S και τα πεδία $(D, \sqsubseteq_D)$ και $(E, \sqsubseteq_E)$
  - Το ζεύγος $(S, =)$ ορίζει ένα πεδίο διακριτής διάταξης (discretely ordered)
  - Ανυψωμένο (lifted) πεδίο

$$
\begin{array}{lll}
D_\perp & = & D \cup \{ \perp \} \qquad \text{όπου } \perp \notin D \\
\perp \sqsubseteq d & & \text{για κάθε } d \in D \\
d_1 \sqsubseteq d_2 & & \text{αν } d_1 \sqsubseteq_D d_2
\end{array}
$$

  - Αν το D είναι πεδίο διακριτής διάταξης, το $D_\perp$ λέγεται επίπεδο (flat) πεδίο

Θεωρία πεδίων (v)

- Κατασκευές πεδίων: (συνέχεια)
  - Γινόμενο (product)

$$
\begin{array}{l}
D \times E \;=\; \{ \langle d, e \rangle \mid d \in D, e \in E \} \\
\langle d_1, e_1 \rangle \sqsubseteq \langle d_2, e_2 \rangle \;\; \text{αν } d_1 \sqsubseteq_D d_2 \text{ και } e_1 \sqsubseteq_E e_2
\end{array}
$$

  - Διαχωρισμένο άθροισμα (disjoint sum)

$$
\begin{array}{l}
D + E \;=\; \{ \langle 0, d \rangle \mid d \in D \} \cup \{ \langle 1, e \rangle \mid e \in E \} \\
\langle 0, d_1 \rangle \sqsubseteq \langle 0, d_2 \rangle \;\; \text{αν } d_1 \sqsubseteq_D d_2 \\
\langle 1, e_1 \rangle \sqsubseteq \langle 1, e_2 \rangle \;\; \text{αν } e_1 \sqsubseteq_E e_2
\end{array}
$$

Θεωρία πεδίων (vi)

- Μια συνάρτηση $f : D \to E$ λέγεται μονότονη (monotone) αν για κάθε $x \sqsubseteq_D y$ ισχύει $f(x) \sqsubseteq_E f(y)$
- Μια συνάρτηση $f : D \to E$ λέγεται συνεχής (continuous) αν για κάθε ω-αλυσίδα του D

$$
f\left(\bigsqcup_{i=0}^{\infty} d_i\right) \;=\; \bigsqcup_{i=0}^{\infty} f(d_i)
$$

- Αν η f είναι συνεχής, τότε είναι μονότονη
- Μια συνάρτηση $f : D \to E$ λέγεται αυστηρή (strict) αν $f(\perp_D) = \perp_E$

Θεωρία πεδίων (vii)

- Κατασκευές πεδίων: (συνέχεια)
  - Πεδίο συναρτήσεων (function domain)

$$
\begin{array}{l}
D \to E \;=\; \{ f : D \to E \mid f \text{ συνεχής} \} \\
f \sqsubseteq g \;\; \text{αν } f(x) \sqsubseteq_E g(x) \text{ για κάθε } x \in D
\end{array}
$$

- Θεώρημα ελάχιστου σταθερού σημείου: Έστω D πεδίο με ελάχιστο στοιχείο $\perp$ και $f : D \to D$ συνεχής συνάρτηση. Τότε η f έχει ελάχιστο (ως προς $\sqsubseteq_D$) σταθερό σημείο και αυτό είναι ίσο με

$$
\operatorname{fix} f \;=\; \bigsqcup_{i=0}^{\infty} f^i(\perp)
$$

Προστακτικές γλώσσες ξανά (i)

- Έστω τα πεδία διακριτής διάταξης
  - $\mathbb{N}$: φυσικοί αριθμοί
  - $\mathbb{T}$: λογικές τιμές $\{ \mathit{true}, \mathit{false} \}$
  - S: καταστάσεις μνήμης
- Σημασιολογικές συναρτήσεις

$$
\begin{array}{rcl}
\mathcal{C}[\![\mathbf{C}]\!] & : & S \to S_\perp \\
\mathcal{B}[\![\mathbf{B}]\!] & : & S \to \mathbb{T} \\
\mathcal{N}[\![\mathbf{N}]\!] & : & S \to \mathbb{N}
\end{array}
$$

Προστακτικές γλώσσες ξανά (ii)

- Αν $f : D \to E_\perp$ ορίζουμε $f^+ : D_\perp \to E_\perp$

$$
f^+(x) \;=\; \begin{cases} \perp, & \text{αν } x = \perp \\ f(x), & \text{διαφορετικά} \end{cases}
$$

- Σημασιολογικές εξισώσεις

$$
\begin{array}{lcl}
\mathcal{C}[\![\mathtt{skip}]\!]s & = & s \\
\mathcal{C}[\![\mathbf{C}_0 ; \mathbf{C}_1]\!]s & = & \mathcal{C}[\![\mathbf{C}_1]\!]^+(\mathcal{C}[\![\mathbf{C}_0]\!]s) \\
\mathcal{C}[\![\mathtt{for}\ \mathbf{N}\ \mathtt{do}\ \mathbf{C}]\!]s & = & (\mathcal{C}[\![\mathbf{C}]\!]^+)^n(s), \;\; \text{όπου } n = \mathcal{N}[\![N]\!]s
\end{array}
$$

Προστακτικές γλώσσες ξανά (ii)

- Σημασιολογικές εξισώσεις (συνέχεια)

$$\mathcal{C}[\![\texttt{if } \mathbf{B} \texttt{ then } \mathbf{C}_0 \texttt{ else } \mathbf{C}_1]\!]s \;=\; \begin{cases} \mathcal{C}[\![\mathbf{C}_0]\!]s, \text{ αν } \mathcal{B}[\![\mathbf{B}]\!]s = \mathit{true} \\ \mathcal{C}[\![\mathbf{C}_1]\!]s, \text{ διαφορετικά} \end{cases}$$

- Σημασιολογία της εντολής `while`

$$\mathcal{C}[\![\texttt{while } \mathbf{B} \texttt{ do } \mathbf{C}]\!]s \;=\; \mathrm{fix}\, F\ s$$

$$F\ f\ s \;=\; \begin{cases} f(\mathcal{C}[\![\mathbf{C}]\!]s), & \text{αν } \mathcal{B}[\![\mathbf{B}]\!]s = \mathit{true} \\ s, & \text{αν } \mathcal{B}[\![\mathbf{B}]\!]s = \mathit{false} \end{cases}$$

παράβαλε $F^i(\bot)$ και c_i

Σημασιολογία λ-λογισμού (i)

- Σύνταξη του λ-λογισμού χωρίς τύπους

$$M, N \;::=\; x \mid (\lambda x.\, M) \mid (M\, N)$$

- Σημασιολογική συνάρτηση $[\![M]\!] : D$
- Πρόβλημα: τα στοιχεία του πεδίου D είναι συναρτήσεις $f : D \to D$

$$D \;\simeq\; D \to D$$

- Δεν υπάρχει μη τετριμμένο σύνολο που να ικανοποιεί τον παραπάνω ισομορφισμό
- Υπάρχει όμως τέτοιο πεδίο (Scott)

Σημασιολογία λ-λογισμού (ii)

- Ισομορφισμός $D \;\simeq\; D \to D$

$$\begin{array}{lcl} \phi & : & D \to (D \to D) \\ \psi & : & (D \to D) \to D \end{array} \qquad \begin{array}{lcl} \phi \circ \psi & = & \mathrm{id} \\ \psi \circ \phi & = & \mathrm{id} \end{array}$$

- Περιβάλλον: $\rho \in \mathit{Env} = \mathit{Var} \to D$
- Σημασιολογική συνάρτηση: $[\![M]\!] : \mathit{Env} \to D$
- Σημασιολογικές εξισώσεις

$$\begin{array}{lcl} [\![x]\!]\rho & = & \rho\, x \\ [\![\lambda x.\, M]\!]\rho & = & \psi\,(\lambda v : D.\, [\![M]\!](\rho[x := v])) \\ [\![M\, N]\!]\rho & = & \phi\,([\![M]\!]\rho)\,([\![N]\!]\rho) \end{array}$$

Σημασιολογία λ-λογισμού (iii)

- Ιδιότητες της σημασιολογίας
  - Συνέπεια της β-μετατροπής

$$[\![(\lambda x.\, M)\, N]\!] \;=\; [\![M[x := N]]\!]$$

  - Ομοίως, συνέπεια της η-μετατροπής
  - Αποδεικνύονται (δεδομένων των ϕ και ψ):

$$\begin{array}{rcl} [\![\Omega]\!]\rho & = & \bot \\ [\![\lambda x.\, \Omega]\!]\rho & = & \psi\,\bot \;=\; \bot \\ [\![(\lambda x.\, I)\, \Omega]\!]\rho & = & [\![I]\!]\rho \end{array}$$

Η σημασιολογία αποδίδει τη στρατηγική της αριστερότερης μετατροπής