

# Κεφάλαιο 4

## Σημασιολογία μιας Απλής Προστακτικής Γλώσσας

Π. Ροντογιάννης

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών  
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Μία κλασσική γλώσσα προγραμματισμού αποτελείται από:

- Εκφράσεις
  - των οποίων η τιμή πρέπει να αποτιμηθεί
- Εντολές
  - οι οποίες εκτελούνται και αλλάζουν την κατάσταση του υπολογιστή

## Βασική σημασιολογική έννοια

- Κατάσταση
  - Τιμές, που έχουν οι μεταβλητές στη μνήμη του υπολογιστή
  - Η τιμή μίας έκφρασης του προγράμματος εξαρτάται από την παρούσα κατάσταση του υπολογιστή
  - Αλλαγή κατάστασης (π.χ. με καταχώρηση νέας τιμής σε μία από τις μεταβλητές του προγράμματος)

- Δομές ελέγχου
  - Καθορίζουν την πορεία εκτέλεσης εντολών ενός προγράμματος
  - Π.χ. If-then-else, for, while
  - Υποστηρίζονται από προστακτικές γλώσσες

- Απλή προστακτική γλώσσα
  - Υποστήριξη δύο τύπων δεδομένων
    - Τιμές αληθείας (Boolean)
    - Φυσικούς αριθμούς (Natural)
  - Εισαγωγή και άλλων τύπων χωρίς πρόβλημα
    - `int`, `real`, `char`
  - Τρεις κατηγορίες συντακτικών στοιχείων
    - Λογικές εκφράσεις
    - Αριθμητικές εκφράσεις
    - Εντολές

# Βασικές Αριθμητικές και Λογικές Εκφράσεις

- Αριθμητικές
  - Σταθερά (0)
  - Τελεστής (succ)
    - Παράγει τον επόμενο φυσικό αριθμό
- Λογικές
  - Σταθερά true
  - Εκφράσεις, που παράγονται με χρήση τελεστών (not, and, κλπ)
- Προσθήκη επιπλέον συμβόλων (πχ false, 1, 2 κλπ) και τελεστών χωρίς πρόβλημα

- «Κενή» εντολή (skip)
  - Δεν έχει κανένα αποτέλεσμα
- Σύνθεση εντολών
  - Συμβολισμός με ';'
- Μηχανισμός βρόγχου
  - Ο βρόγχος 'for N do C' εκτελείται υπολογίζοντας την τιμή της αριθμητικής έκφρασης N και μετά εκτελώντας το σώμα C τόσες φορές όσες είναι η τιμή της αριθμητικής έκφρασης. Ο βρόγχος ονομάζεται οριστικός, γιατί ο αριθμός των επαναλήψεων καθορίζεται πριν την εκτέλεση του.

- Η γλώσσα δεν διαθέτει (αρχικά) μηχανισμούς αλλαγής της κατάστασης ούτε μεταβλητές.



- Δίνεται με τη βοήθεια context-free γραμματικής
  - Το σύμβολο
    - $C$  αναπαριστά μία εντολή
    - $B$  μία λογική έκφραση
    - $N$  μία αριθμητική έκφραση

# Κανόνες Παραγωγής Εντολών

## Κανόνας

```
C ::= skip
   | C0 ; C1
   | for N do C
   | if B then C0 else C1
   | (C)
```

# Κανόνες Παραγωγής Λογικών Εκφράσεων

## Κανόνας

```
 $B ::= \text{true}$   
|  $\text{not } B$   
|  $B_0 \text{ and } B_1$   
|  $N_0 < N_1$   
|  $N_0 = N_1$   
|  $B_0 = B_1$   
|  $\text{if } B \text{ then } B_0 \text{ else } B_1$   
|  $(B)$ 
```

# Κανόνες Παραγωγής Αριθμητικών Εκφράσεων

## Κανόνας

```
 $N ::= 0$   
|  $\text{succ } N$   
|  $\text{if } B \text{ then } N_0 \text{ else } N_1$   
|  $(N)$ 
```

- Βασική είναι η έννοια της κατάστασης (state)
  - Αναπαριστά τη μνήμη υπολογιστή
  - Η τιμή μιας έκφρασης σε μία προστακτική γλώσσα προγραμματισμού εξαρτάται από τις μεταβλητές της έκφρασης.
  - Οι τιμές των μεταβλητών φυλάσσονται στη μνήμη του υπολογιστή
- Η γλώσσα όπως ορίσθηκε δεν διαθέτει μεταβλητές και εντολές ανάθεσης

- Η γλώσσα όμως θα επεκταθεί και έχουν ήδη εισαχθεί απλές εντολές
- Η σημασιολογία της γλώσσας δίνεται με βάση την έννοια της κατάστασης

- Έστω σύνολο καταστάσεων  $S$
- Το νόημα μιας
  - Έκφρασης της γλώσσας είναι μία συνάρτηση η οποία για κάθε κατάσταση του υπολογιστή δίνει την τιμή της έκφρασης σε σχέση με την κατάσταση αυτή
  - Εντολής είναι μία συνάρτηση η οποία μετασχηματίζει την τρέχουσα κατάσταση του υπολογιστή σε μία νέα κατάσταση

- $\mathbb{C}[[\cdot]]$ 
  - Δίνει νόημα στις εντολές της γλώσσας
- $\mathbb{B}[[\cdot]]$ 
  - Δίνει νόημα στις λογικές εκφράσεις
- $\mathbb{N}[[\cdot]]$ 
  - Δίνει νόημα στις αριθμητικές εκφράσεις
- Οι συναρτήσεις αυτές ορίζονται αναδρομικά και ο ορισμός καθεμιάς χρησιμοποιεί τις υπόλοιπες



# Ορισμός συνάρτησης $\mathbb{B}[[\cdot]]$

$$\mathbb{B}[[\text{true}]]s = \text{true}$$

$$\mathbb{B}[[\text{not } B]]s = \begin{cases} \text{true,} & \text{if } \mathbb{B}[[B]]s = \text{false} \\ \text{false,} & \text{if } \mathbb{B}[[B]]s = \text{true} \end{cases}$$

$$\mathbb{B}[[B_0 \text{ and } B_1]]s = \begin{cases} \text{true,} & \text{if } \mathbb{B}[[B_0]]s = \text{true and} \\ & \mathbb{B}[[B_1]]s = \text{true} \\ \text{false,} & \text{otherwise} \end{cases}$$

# Ορισμός συνάρτησης $\mathbb{B}[[\cdot]]$

$$\mathbb{B}[[N_0 < N_1]]s = \begin{cases} \text{true,} & \text{if } \mathbb{N}[[N_0]]s < \mathbb{N}[[N_1]]s \\ \text{false,} & \text{otherwise} \end{cases}$$

$$\mathbb{B}[[N_0 = N_1]]s = \begin{cases} \text{true,} & \text{if } \mathbb{N}[[N_0]]s = \mathbb{N}[[N_1]]s \\ \text{false,} & \text{otherwise} \end{cases}$$

# Ορισμός συνάρτησης $\mathbb{B}[[\cdot]]$

$$\mathbb{B}[[B_0 = B_1]]s = \begin{cases} \text{true,} & \text{if } \mathbb{B}[[B_0]]s = \mathbb{B}[[B_1]]s \\ \text{false,} & \text{otherwise} \end{cases}$$

$$\mathbb{B}[[\text{if } B \text{ then } B_0 \text{ else } B_1]]s = \begin{cases} \mathbb{B}[[B_0]]s, & \text{if } \mathbb{B}[[B]]s = \text{true} \\ \mathbb{B}[[B_1]]s, & \text{otherwise} \end{cases}$$

$$\mathbb{B}[[ (B) ]]s = \mathbb{B}[[B]]s$$

# Ορισμός συνάρτησης $\mathbb{N}[[\cdot]]$

$$\mathbb{N}[[0]]s = 0$$

$$\mathbb{N}[[\text{succ } M]]s = \mathbb{N}[[M]]s + 1$$

$$\mathbb{N}[[\text{if } B \text{ then } N_0 \text{ else } N_1]]s = \begin{cases} \mathbb{N}[[N_0]]s, & \text{if } \mathbb{B}[[B]]s = \text{true} \\ \mathbb{N}[[N_1]]s, & \text{otherwise} \end{cases}$$

$$\mathbb{N}[[\lambda (N)]]s = \mathbb{N}[[M]]s$$

# Ορισμός συνάρτησης $\mathbb{C}[[\cdot]]$

$$\mathbb{C}[[\text{skip}]]s = s$$

$$\mathbb{C}[[C_0; C_1]]s = \mathbb{C}[[C_1]](\mathbb{C}[[C_0]]s)$$

$$\mathbb{C}[[\text{for } N \text{ do } C]]s = \mathbb{C}[[C]]^n s \text{ where } n = \mathbb{N}[[N]]s$$

$$\mathbb{C}[[\text{if } B \text{ then } C_0 \text{ else } C_1]]s = \begin{cases} \mathbb{C}[[C_0]]s, & \text{if } \mathbb{B}[[B]]s = \text{true} \\ \mathbb{C}[[C_1]]s, & \text{otherwise} \end{cases}$$

$$\mathbb{C}[[\langle C \rangle]]s = \mathbb{C}[[C]]s$$

- Τεχνική για αποδείξεις στη σημασιολογία
- Εφαρμογή της συνηθισμένης επαγωγής στην συντακτική δομή των εκφράσεων μιας γλώσσας (πιο συγκεκριμένα στο βάθος του συντακτικού δέντρου των εκφράσεων της γλώσσας)

- Μία ιδιότητα αποδεικνύεται για κάθε
  - ένα από τα πρωτόγονα στοιχεία του συντακτικού της γλώσσας
  - σύνθετη έκφραση της γλώσσας με βάση την υπόθεση ότι οι εκφράσεις που την αποτελούν έχουν την ιδιότητα

## Παράδειγμα 4.1

Με τη χρήση δομικής επαγωγής μπορεί ναδειχθεί ότι για την δεδομένη γλώσσα (που δεν έχει εντολές ανάθεσης) ισχύει:

$$\mathbb{C}[[C]]s = s$$

για κάθε εντολή  $C$  και για κάθε κατάσταση  $s$ .



## Παράδειγμα 4.1

Για την επαγωγική βάση αρκεί να εξετασθεί η περίπτωση  $C = \text{skip}$ , για την οποία ισχύει το ζητούμενο, καθώς

$$C[[\text{skip}]]s = s$$

Μένει να δειχθεί ότι ισχύει και για πιο σύνθετες εντολές με βάση ότι ισχύει για τις υποεντολές, που τις συνθέτουν. Αυτό απαιτεί την εξέταση τεσσάρων περιπτώσεων μιας σύνθετης εντολής (σύνθεση εντολών, for, if και παρενθετοποίηση).

$$\begin{aligned}\mathbb{C}[[C_0; C_1]]s &= \mathbb{C}[[C_1]](\mathbb{C}[[C_0]]s) && \text{(Semantics)} \\ &= \mathbb{C}[[C_1]](s) && \text{(Hypothesis)} \\ &= s && \text{(Hypothesis)}\end{aligned}$$

Οι υπόλοιπες περιπτώσεις σύνθετων εντολών μπορούν να δειχθούν με ανάλογο τρόπο.

- Σημαντική έλλειψη της γλώσσας η απουσία μεθόδων για προσπέλαση και αλλαγή της κατάστασης (δηλαδή της μνήμης)
- Για το σκοπό αυτό ορίζεται ένα σύνολο ονομάτων μεταβλητών (variable-identifiers)
  - Συμβολίζεται με *Var*
  - Χωρίζεται σε δύο υποσύνολα μεταβλητών
    - Λογικές (Boolean)
    - Αριθμητικές (Natural)

- Προσδιορίζει τον τύπο της κάθε μεταβλητής
- Παράδειγμα:
  - Αν  $i$  είναι μία λογική μεταβλητή, τότε  $\pi(i) = \text{boolean}$ ,  
ενώ αν  $j$  είναι μία αριθμητική μεταβλητή, τότε  
 $\pi(j) = \text{natural}$

# Επέκταση Συντακτικού

- Επιτρέπει χρήση μεταβλητών και εντολών ανάθεσης
- Επεκτείνουμε κανόνες παραγωγής εκφράσεων και εντολών της γλώσσας, ως εξής:

## Κανόνας

$$\begin{array}{l} B ::= \dots \\ \quad | \quad i, \text{ where } i \in \text{Var}, \pi(i) = \text{boolean} \\ N ::= \dots \\ \quad | \quad i, \text{ where } i \in \text{Var}, \pi(i) = \text{natural} \\ C ::= \dots \\ \quad | \quad i := B, \text{ where } i \in \text{Var}, \pi(i) = \text{boolean} \\ \quad | \quad i := N, \text{ where } i \in \text{Var}, \pi(i) = \text{natural} \end{array}$$

# Μαθηματικός Ορισμός Συνόλου Καταστάσεων

- Διαισθητικά, μία κατάσταση  $s \in S$  είναι μία συνάρτηση, που
  - αντιστοιχίζει σε κάθε μεταβλητή της γλώσσας μία τιμή
  - πρέπει να «σέβεται» τον τύπο κάθε μεταβλητής
  - αντιστοιχίζει στις
    - Λογικές μεταβλητές, λογικές τιμές
    - Αριθμητικές μεταβλητές, αριθμητικές τιμές

- Με βάση τις παρατηρήσεις, το σύνολο  $S$  των καταστάσεων ορίζεται ως:

$$S = \left\{ \begin{array}{l} s : Var \rightarrow Nat \cup \{true, false\} \mid \\ s(i) \in Nat, \text{ if } \pi(i) = \textit{natural} \text{ and} \\ s(i) \in \{true, false\}, \text{ if } \pi(i) = \textit{boolean} \end{array} \right\}$$

- Η σημασιολογία της γλώσσας μπορεί να επεκταθεί, για να καλύψει μεταβλητές και εντολές ανάθεσης

Στην δεύτερη εξίσωση, το σύμβολο  $E$  χρησιμοποιείται για αναπαράσταση οποιασδήποτε έκφρασης της γλώσσας (λογικής ή αριθμητικής)

$$[[i]]s = s(i)$$

$$[[i := E]]s = (s \mid i \mapsto [[E]]s)$$

Όπου  $(s \mid i \mapsto [[E]]s)$  είναι  
μία νέα κατάσταση  $s'$  έτσι ώστε  $s'(i) = [[E]]s$   
και για κάθε άλλη μεταβλητή  $i' \neq i$ ,  $s'(i') = s(i')$ .



- Η δεύτερη εξίσωση λέει ότι η εκτέλεση μίας εντολής ανάθεσης σε μία δεδομένη κατάσταση έχει ως αποτέλεσμα την παραγωγή μιας «νέας» κατάστασης στην οποία άλλαξε μόνο η τιμή της μεταβλητής, που σχετίζεται με την ανάθεση.
- Η νέα τιμή είναι η τιμή του δεξιού μέλους της εντολής ανάθεσης υπολογισμένη στην «παλαιά» κατάσταση.

# Απόδειξη Ισοδυναμίας Εντολών

Η εντολή

$$i := (\text{if } B \text{ then } E_0 \text{ else } E_1)$$

και η εντολή

$$\text{if } B \text{ then } (i := E_0) \text{ else } (i := E_1)$$

μπορούν εύκολα να αποδειχθούν ισοδύναμες με τη χρήση των παραπάνω σημασιολογικών εξισώσεων.