



# Σχεδίαση Γλωσσών & Μεταγλωττιστές

Ενότητα 12: Σχήματα Μετάφρασης & Παραδείγματα  
Σημασιολογικής Ανάλυσης

Επ. Καθ. Π. Κατσαρός  
Τμήμα Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδεια χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



# Σχήματα μετάφρασης I

- Στις γραμματικές ιδιοτήτων δε διατυπώνεται η σειρά υπολογισμού τιμών ιδιοτήτων των συμβόλων.
  - Οι εξισώσεις ιδιοτήτων επισυνάπτονται στα δεξιά των κανόνων παραγωγής
  - Στην πιο γενική περίπτωση, για τον καθορισμό της σειράς υπολογισμού των τιμών απαιτείται η δημιουργία ενός γράφου εξάρτησης τιμών
- Σε ένα σχήμα μετάφρασης η σειρά υπολογισμού τιμών καθορίζεται με την απευθείας τοποθέτηση των ενεργειών στις επιλεγείσες θέσεις του δεξιού μέρους των κανόνων παραγωγής
  - Μία κληρονομήσιμη ιδιότητα ενός συμβόλου του δεξιού μέρους πρέπει να έχει υπολογισθεί σε ενέργεια πριν από αυτό το σύμβολο
  - Μία ενέργεια δεν πρέπει να αναφέρεται σε συνθέσιμη ιδιότητα συμβόλου που βρίσκεται δεξιά της ενέργειας
  - Μία συνθέσιμη ιδιότητα για το μη τερματικό σύμβολο του αριστερού μέρους μπορεί να υπολογισθεί μόνο μετά τον υπολογισμό όλων των ιδιοτήτων στις οποίες αναφέρεται (την τοποθετούμε στο τέλος του δεξιού μέρους του κανόνα)



## Σχήματα μετάφρασης II

Παραγωγές

$$D \rightarrow T L$$

$$T \rightarrow \text{int}$$

$$\quad / \text{ float}$$

$$L \rightarrow L_1, \text{id}$$

$$\quad | \text{ id}$$

Σημασιολογικοί κανόνες

$$L.in \leftarrow T.type$$

$$T.type \leftarrow \text{integer}$$

$$T.type \leftarrow \text{float}$$

$$L_1.in \leftarrow L.in, \text{enter}(\text{id.spelling}, L.in)$$

$$\text{enter}(\text{id.spelling}, L.in)$$

Καταχώρηση μεταβλητών στον πίνακα συμβόλων.  
Από θεωρητικής απόψεως αυτός ο κανόνας δεν επιτρέπεται σε μία γραμματική ιδιοτήτων. Μιλάμε για μία ad hoc μετάφραση.

Σχήμα μετάφρασης: Οι σημασιολογικές ενέργειες τοποθετούνται στα δεξιά μέρη των παραγωγών καθορίζοντας τη σειρά εκτέλεσής τους.

$$D \rightarrow T \{ L.in \leftarrow T.type \} L$$

$$T \rightarrow \text{int} \{ T.type \leftarrow \text{integer} \}$$

$$\quad / \text{ float} \{ T.type \leftarrow \text{float} \}$$

$$L \rightarrow \{ L_1.in \leftarrow L.in \} L_1, \text{id} \{ \text{enter}(\text{id.spelling}, L.in) \}$$

$$\quad | \text{ id} \{ \text{enter}(\text{id.spelling}, L.in) \}$$


## Σχήματα μετάφρασης III

ΠΑΡΑΔΕΙΓΜΑ:

Το σχήμα μετάφρασης, που ακολουθεί, μετατρέπει αριθμητικές εκφράσεις ένθετης μορφής στην αντίστοιχη επιθεματική μορφή. Έτσι, η μετάφραση της έκφρασης,

$$((a+b) * c) / (d-e*f) + 3$$

δίνει ως αποτέλεσμα την

$$ab+c*def*- / 3+$$

έκφραση =	όρος υπ_όροι.
υπ_όροι =	“+” όρος {print(“+”.λεξ_μονάδα)} υπ_όροι <sub>R</sub>   “-” όρος {print(“-”.λεξ_μονάδα)} υπ_όροι <sub>R</sub>   ε.
όρος =	παράγοντας υπ_παραγ.
υπ_παραγ =	“*” παράγοντας {print(“*” .λεξ_μονάδα)} υπ_παραγ <sub>R</sub>   “/” παράγοντας {print(“/” .λεξ_μονάδα)} υπ_παραγ <sub>R</sub>   ε.
παράγοντας =	“(” έκφραση “)”   “αριθμός” {print(“αριθμός ”.τιμή)}.



## Παράδειγμα γραμματικής ιδιοτήτων: συντακτικό δένδρο

Γραμματική ιδιοτήτων για την ανάπτυξη του συντακτικού δένδρου των εκφράσεων.

ιδιότητα <i>nptr</i>	δείκτης σε κόμβο δένδρου
ιδιότητα <b>ID.entry</b>	η εγγραφή ενός ονόματος στον πίνακα συμβόλων
<i>mknode</i>	δημιουργία νέου κόμβου
<i>mkleaf</i>	δημιουργία φύλλου στο δένδρο

$E_{n+1} = E_{n-1} \text{ “+” } E_n.$

$[E_{n+1}.nptr = mknode(‘+’, E_{n-1}.nptr, E_n.nptr) ]$

$E_{n+1} = E_{n-1} \text{ “*” } E_n.$

$[E_{n+1}.nptr = mknode(‘*’, E_{n-1}.nptr, E_n.nptr) ]$

$E_{n+1} = \text{“(” } E_n \text{ “)”}.$

$[E_{n+1}.nptr = E_n.nptr ]$

$E_{n+1} = \text{“ID”}.$

$[E_{n+1}.nptr = mkleaf(\mathbf{ID}, \mathbf{ID}.entry) ]$



# Παράδειγμα γραμματικής ιδιοτήτων: έλεγχος τύπων I

- βασικοί τύποι (λογικός τύπος, ακέραιοι, πραγματικοί κ.α.) και οριζόμενοι από τον προγραμματιστή, που μπορούν να έχουν το δικό τους όνομα
- ο τύπος ενός δομικού στοιχείου της γλώσσας περιγράφεται από μία *έκφραση τύπων*, όπου συνδυάζονται ένας ή περισσότεροι απλοί τύποι με *δομητές τύπων*
- τύπος πίνακα: `array(index_type, component_type)`
- εγγραφή από δύο πεδία: `type1 x type2`
- δείκτης: `pointer(type)`
- συνάρτηση πρόσθεσης δύο αριθμών: `int x int → int`





## Παράδειγμα γραμματικής ιδιοτήτων: έλεγχος τύπων ΙΙ

- έλεγχος συνέπειας τύπων:  
υλοποιείται με μία ανοδική διάσχιση του συντακτικού δένδρου όπου οι τύποι των φύλλων (ονόματα, σταθερές) εξετάζονται με βάση τις αντίστοιχες δηλώσεις τους στον πίνακα συμβόλων για κάθε κόμβο που αντιστοιχεί σε τελεστή γίνεται έλεγχος αν οι τύποι των υποδένδρων είναι επιτρεπτοί για τη συγκεκριμένη πράξη
- *ονομαστική ισοδυναμία τύπων*: πολύ περιοριστική

```
type my_int = integer;  
var x: integer;  
y: my_int;  
x:=y;
```

- *δομικά ισοδύναμες εκφράσεις*: αν είναι του ίδιου βασικού τύπου ή οι τύποι τους προκύπτουν μετά από εφαρμογή του ίδιου δομητή σε δομικά ισοδύναμους τύπους



# Παράδειγμα γραμματικής ιδιοτήτων: δηλώσεις τύπων

δηλώσεις μεταβλητών – δείκτη και πινάκων

decl = type varlist.

type = “INT”.

type = “FLOAT”.

varlist = varlist<sub>R</sub> “,” “ID”.

varlist = varlist<sub>R</sub> “,” “ID” “[” “Num” “]”. [varlist<sub>R</sub>.in = varlist.in

varlist = varlist<sub>R</sub> “,” “\*” “ID”.

varlist = “ID”.

varlist = “ID” “[” “Num” “]”.

varlist = “\*” “ID”.

[varlist.in = type.name ]

[type.name = “integer” ]

[type.name = “float” ]

[varlist<sub>R</sub>.in = varlist.in

addtype(**ID**.entry, varlist.in) ]

addtype(**ID**.entry, array(0... “Num”.val – 1,  
varlist.in))]

[varlist<sub>R</sub>.in = varlist.in

addtype(**ID**.entry, pointer(varlist.in)) ]

[addtype(**ID**.entry, varlist.in) ]

[ addtype(**ID**.entry, array(0... “Num”.val – 1,  
varlist.in))]

[ addtype(**ID**.entry, pointer(varlist.in)) ]



## Παράδειγμα σχήματος μετάφρασης: έλεγχος τύπων αριθμητικών εκφράσεων

$E = \text{"literal"}$ .       $\{E.type = \text{"char"}\}$   
 $E = \text{"Num"}$ .       $\{E.type = \text{"integer"}\}$   
 $E = \text{"ID"}$ .       $\{E.type = \text{lookup}(\mathbf{ID}.entry)\}$   
 $E = E_1 \text{ "mod" } E_2$ .  $\{ \text{αν } (E_1.type == \text{"integer"} \text{ και } E_2.type == \text{"integer"}) \text{ τότε}$   
     $E.type = \text{"integer"};$   
    αλλιώς  
     $E.type = \text{"type error"}; \}$   
 $E = E_1 \text{ "[" } E_2 \text{ "]"}$ .  $\{ \text{αν } (E_2.type == \text{"integer"} \text{ και}$   
     $E_1.type == \text{array}(i\_type, b\_type)) \text{ τότε}$   
     $E.type = b\_type;$   
    αλλιώς  
     $E.type = \text{"type error"}; \}$   
 $E = \text{"*"} E_R$ .       $\{ \text{αν } (E_R.type == \text{pointer}(b\_type)) \text{ τότε}$   
     $E.type = b\_type;$   
    αλλιώς  
     $E.type = \text{"type error"}; \}$   
 $E = E_1 \text{ "==" } E_2$ .  $\{ \text{αν } (E_1.type == E_2.type) \text{ τότε}$   
     $E.type = \text{"boolean"};$   
    αλλιώς  
     $E.type = \text{"type error"}; \}$





# Τέλος ενότητας

Επεξεργασία: Εμμανουέλα Στάχτιαρη  
Θεσσαλονίκη, 21/07/2014



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ