



Σχεδίαση Γλωσσών & Μεταγλωττιστές

Ενότητα 15: Παραγωγή Κώδικα για Αριθμητικές και
Λογικές Εκφράσεις

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



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
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.

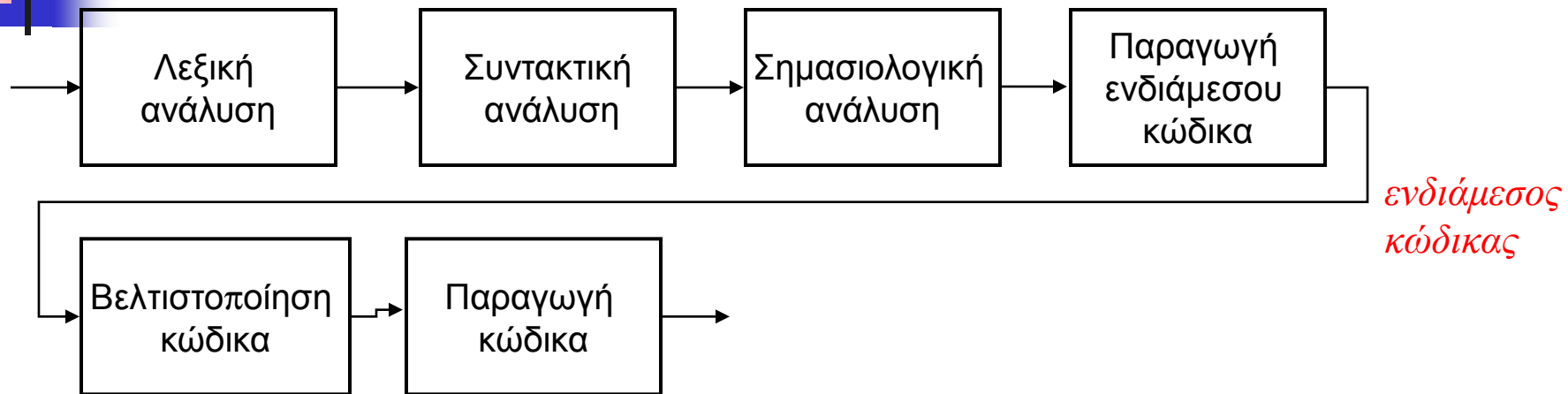


Χρηματοδότηση

- 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.



Δομή μεταγλωττιστή: backend



- Η κατασκευή του πρώτου σταδίου επεξεργασίας (front end) είναι σε μεγάλο βαθμό αυτοματοποιημένη
- Για το τελικό στάδιο επεξεργασίας (back end) δε διαθέτουμε γενικά αυτοματοποιημένες τεχνικές κατασκευής, γιατί εξάλλου ο παραγόμενος κώδικας εξαρτάται από τα χαρακτηριστικά της μηχανής στην οποία θα εκτελείται και οι δυνατότητες βελτιστοποιήσεων ποικίλλουν
- Η δημιουργία ενδιάμεσου κώδικα επηρεάζει καθοριστικά την απόδοση του back end



Μορφές ενδιάμεσης αναπαράστασης

γραφική
ενδιάμεση
αναπαράσταση

- Συντακτικά δένδρα
- Κατευθυνόμενοι άκυκλοι γράφοι
- Γράφοι ροής ελέγχου

γραμμική
ενδιάμεση
αναπαράσταση

- Κώδικας μηχανής στοίβας
- Κώδικας τριών διευθύνσεων

υψηλό επίπεδο



χαμηλό επίπεδο
(πιο κοντά στη μηχανή)

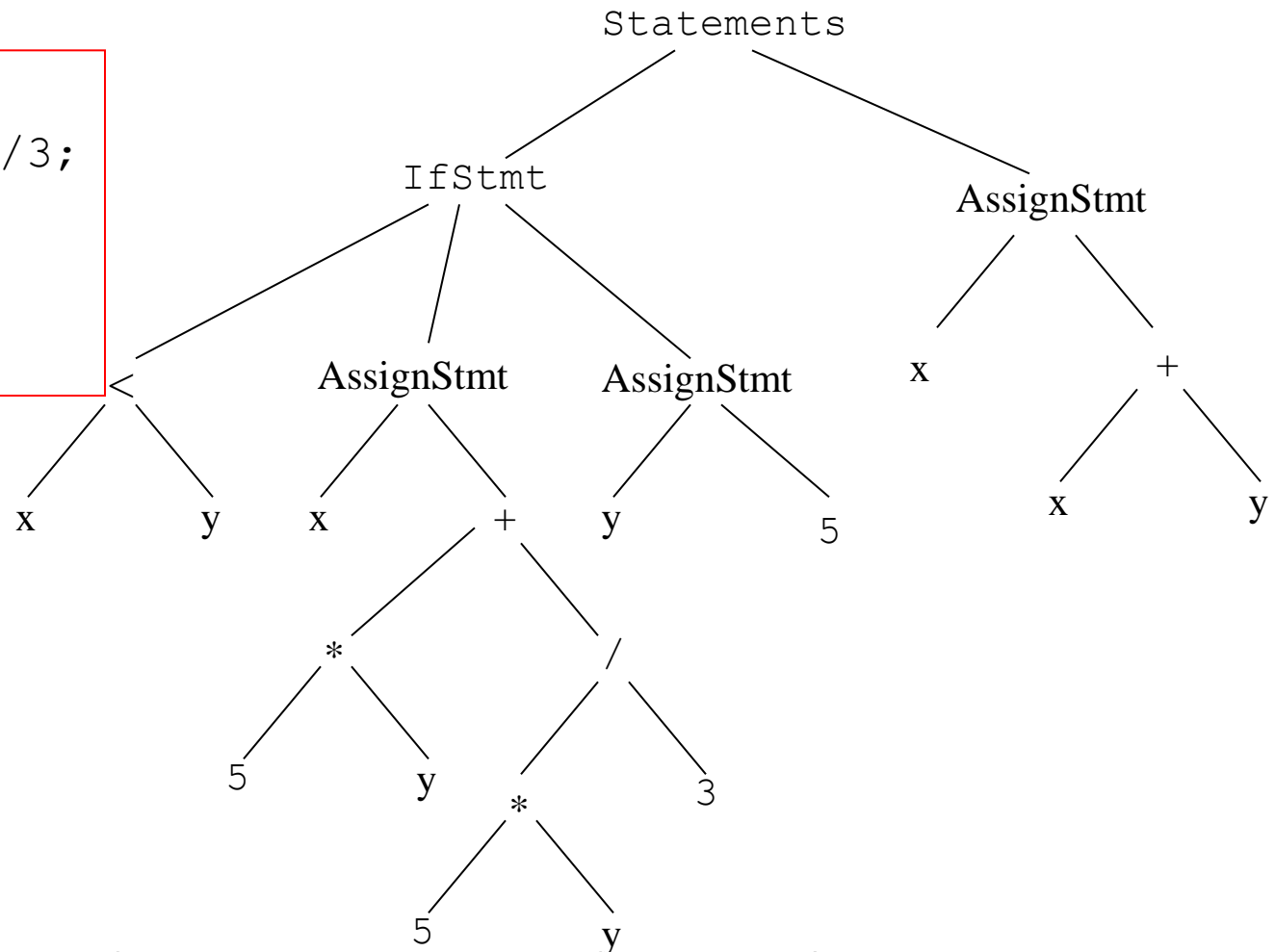
- Υβριδικές προσεγγίσεις που χρησιμοποιούν τεχνικές από τις δύο προαναφερόμενες οικογένειες:
 - Κάποιοι μεταγλωττιστές χρησιμοποιούν κώδικα τριών διευθύνσεων και για επαναληπτικούς βρόχους, εντολές if και αναφορές σε array χρησιμοποιούν συντακτικά δένδρα



ΣΥΝΤΑΚΤΙΚΑ Δένδρα

```

if (x < y)
  x = 5*y + 5*y/3;
else
  y = 5;
  x = x+y;
    
```

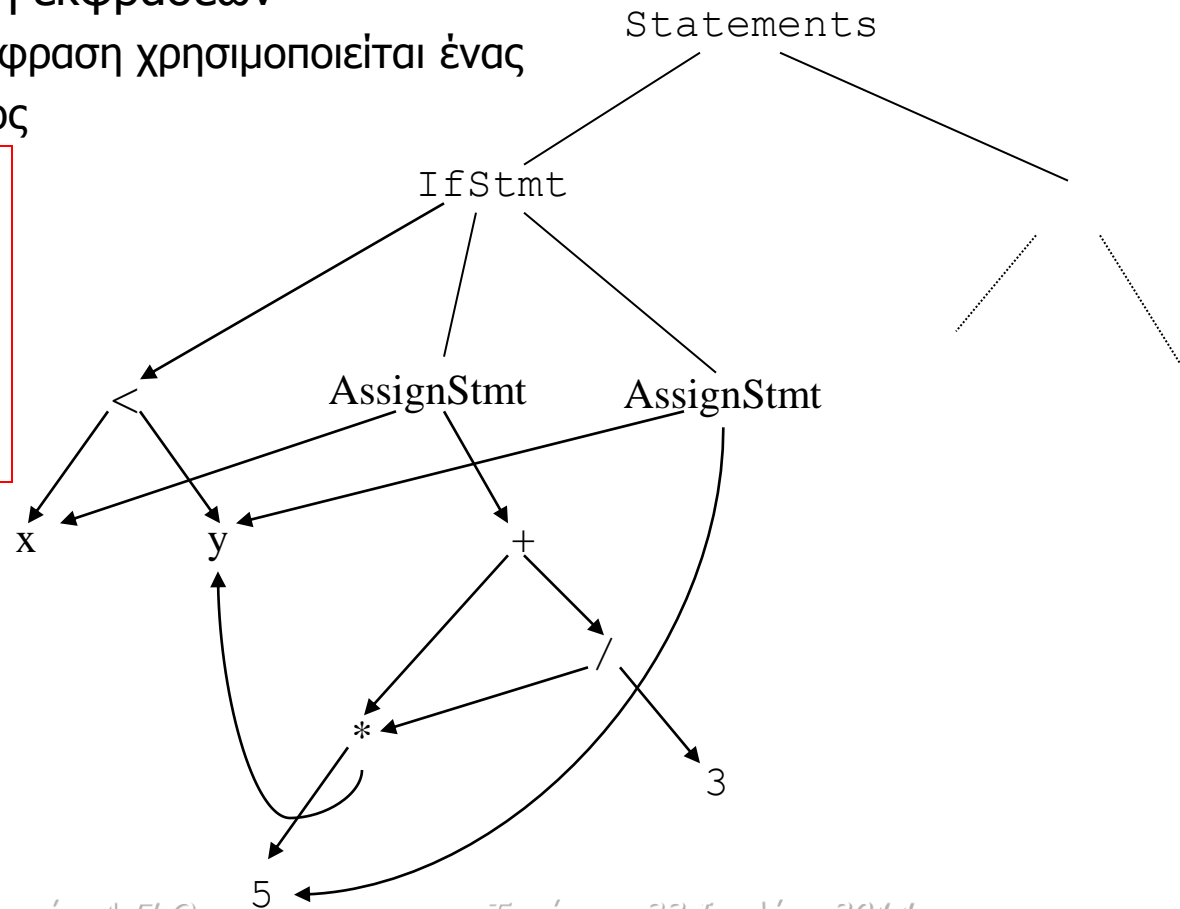


Κατευθυνόμενοι άκυκλοι γράφοι

Όταν γίνεται χρήση κατευθυνόμενων άκυκλων γράφων για την αναπαράσταση εκφράσεων

- Για κάθε έκφραση χρησιμοποιείται ένας μόνο κόμβος

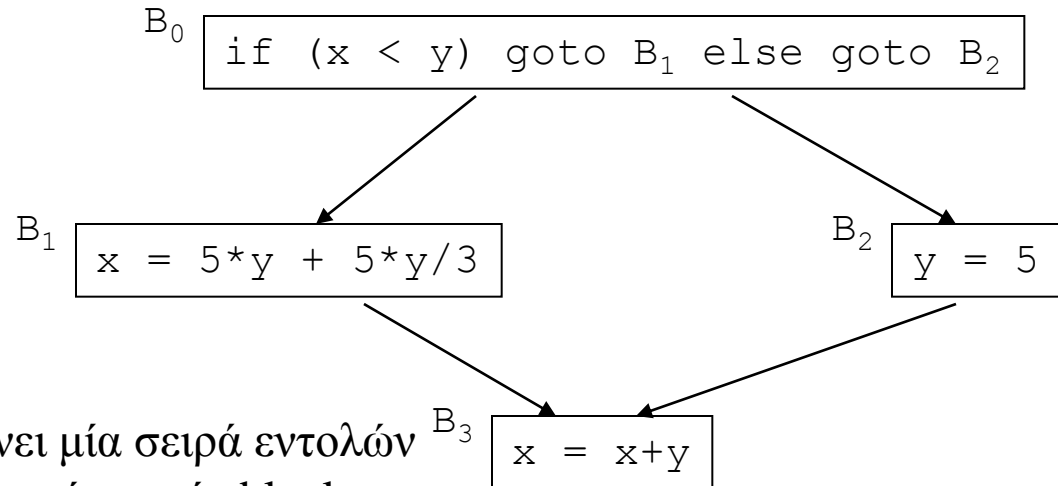
```
if (x < y)
    x = 5*y + 5*y/3;
else
    y = 5;
x = x+y;
```



Γράφοι ροής ελέγχου

- Οι κόμβοι ενός γράφου ροής ελέγχου είναι **βασικά block**
 - Βασικό block είναι μία σειρά εντολών που πάντα η εκτέλεσή τους αρχίζει από την πρώτη και τελειώνει στην τελευταία απ' αυτές
 - Τα τόξα σε ένα γράφο ροής ελέγχου αναπαριστούν τη ροή ελέγχου

```
if (x < y)
    x = 5*y + 5*y/3;
else
    y = 5;
x = x+y;
```



- Κάθε βασικό block περιλαμβάνει μία σειρά εντολών
- Δεν επιτρέπεται διαφυγή από το μέσο ενός block
- Όταν γίνεται εκκίνηση της εκτέλεσης ενός block τότε η εκτέλεση ολοκληρώνεται πάντα στην τελευταία εντολή του



Μηχανή στοίβας

```
if (x < y)
    x = 5*y + 5*y/3;
else
    y = 5;
x = x+y;
```

JVM: Μία μηχανή στοίβας

- Δεξιά φαίνεται συμβολικός κώδικας για την JVM, που μετατρέπεται σε bytecode με τον κατάλληλο συμβο/στή
- Ο JVM διερμηνευτής εκτελεί τον bytecode σε διαφορετικές μηχανές
- Η JVM διαθέτει μία στοίβα που χρησιμοποιείται στην αποτίμηση εκφράσεων
- Η JVM χρησιμοποιεί 65535 «τοπικές μεταβλητές»
- Οι «τοπικές μεταβλητές» λειτουργούν ως καταχωρητές
- Κάθε «τοπική μεταβλητή» στην JVM συμβολίζεται με έναν αριθμό μεταξύ 0 και 65535 (στο παράδειγμα αποδίδονται στις x και y μοναδικοί αριθμοί)

```
load x
load y
iflt L1
goto L2
L1: push 5
load y
multiply
push 5
load y
multiply
push 3
divide
add
store x
goto L3
L2: push 5
store y
L3: load x
load y
add
store x
```

ωθεί την τιμή της θέσης x στην κορυφή της στοίβας

αφαιρεί δύο στοιχεία από την κορυφή και τα συγκρίνει

αφαιρεί δύο στοιχεία από την κορυφή, τα πολλαπλασιάζει και τοποθετεί το αποτέλεσμα στην κορυφή

αποθηκεύει την τιμή της κορυφής της στοίβας στη θέση x



Κώδικας τριών διευθύνσεων I

- Κάθε εντολή μπορεί να έχει το πολύ τρεις τελεστές
- Εκχώρηση τιμών
 - $x := y$
 - $x := y \text{ op } z$ op: δυαδικοί αριθμητικοί ή λογικοί τελεστές
 - $x := \text{op } y$ op: μοναδιαίοι τελεστές (-, NOT κ.α.)
- Διακλάδωση
 - `goto L` εκτέλεση της εντολής με την ετικέτα L
- Υπό συνθήκη διακλάδωση
 - `if x relop y goto L` relop: <, =, <=, >=, ==, !=
 - αν η συνθήκη είναι αληθής τότε εκτελείται η εντολή με την ετικέτα L
 - αν η συνθήκη δεν είναι αληθής τότε συνεχίζεται η εκτέλεση με την επόμενη εντολή



Κώδικας τριών διευθύνσεων ΙΙ

```
if (x < y)
  x = 5*y + 5*y/3;
else
  y = 5;
x = x+y;
```

οι προσωρινές μεταβλητές αντιστοιχούν σε εσωτερικούς κόμβους του συντακτικού δένδρου

Οι μεταβλητές αναπαριστώνται από τη θέση τους στον πίνακα συμβόλων

```

if x < y goto L1
goto L2
L1: t1 := 5 * y
    t2 := 5 * y
    t3 := t2 / 3
    x := t1 + t2
    goto L3
L2: y := 5
L3: x := x + y
```

Οι εντολές του κώδικα τριών διευθύνσεων μπορούν να αναπαρασταθούν
σε πίνακα από τετράδες: πράξη, παράμετρος 1, παράμετρος 2, αποτέλεσμα
σε πίνακα από τριάδες: πράξη, παράμετρος 1, παράμετρος 2
(κάθε τριάδα θεωρείται ότι αντιστοιχεί σε μία προσωρινή μεταβλητή με πρόσβαση στην τιμή της μέσω αναφοράς στον αριθμό της εντολής)



Παραγωγή κώδικα τριών διευθύνσεων

Ιδιότητες:	$E.place$: θέση όπου βρίσκεται η τιμή για την έκφραση E $E.code$: σειρά εντολών που παράγονται για την E
Διαδικασίες:	$newtemp()$: επιστρέφει κάθε φορά μία νέα προσωρινή μεταβλητή $gen()$: δημιουργεί κώδικα (καλείται με τις κατάλληλες παραμέτρους) $lookup(id.name)$: επιστρέφει τη θέση του id στον πίνακα συμβόλων

Παραγωγές

$S \rightarrow id := E$

$E \rightarrow E_1 + E_2$

$E \rightarrow E_1 * E_2$

$E \rightarrow (E_1)$

$E \rightarrow - E_1$

$E \rightarrow id$

Σημασιολογικοί κανόνες

$id.place \leftarrow lookup(id.name);$

$S.code \leftarrow E.code \parallel gen(id.place \text{ ':=' } E.place);$

$E.place \leftarrow newtemp();$

$E.code \leftarrow E_1.code \parallel E_2.code \parallel gen(E.place \text{ ':=' } E_1.place \text{ '+' } E_2.place);$

$E.place \leftarrow newtemp();$

$E.code \leftarrow E_1.code \parallel E_2.code \parallel gen(E.place \text{ ':=' } E_1.place \text{ '*' } E_2.place);$

$E.code \leftarrow E_1.code;$

$E.place \leftarrow E_1.place;$

$E.place \leftarrow newtemp();$

$E.code \leftarrow E_1.code \parallel gen(E.place \text{ ':=' } \text{'uminus'} E_1.place);$

$E.place \leftarrow lookup(id.name);$

$E.code \leftarrow \text{''}$

(κενή συμβολοσειρά)



Παραγωγή κώδικα για μηχανή στοίβας

Ιδιότητες:

$E.code$: σειρά εντολών που παράγονται για την E

(*δε χρειάζεται ιδιότητα θέσης για την έκφραση, αφού το αποτέλεσμα της έκφρασης αποθηκεύεται στη στοίβα*)

Διαδικασίες:

$newtemp()$: επιστρέφει κάθε φορά μία νέα προσωρινή μεταβλητή

$gen()$: δημιουργεί κώδικα (*καλείται με τις κατάλληλες παραμέτρους*)

$lookup(id.name)$: επιστρέφει τη θέση του id στον πίνακα συμβόλων

Παραγωγές

$S \rightarrow id := E$

$E \rightarrow E_1 + E_2$

$E \rightarrow E_1 * E_2$

$E \rightarrow (E_1)$

$E \rightarrow - E_1$

$E \rightarrow id$

Σημασιολογικοί κανόνες

$id.place \leftarrow lookup(id.name);$

$S.code \leftarrow E.code \parallel gen('store' id.place);$

$E.code \leftarrow E_1.code \parallel E_2.code \parallel gen('add');$

(*οι παράμετροι της εντολής add βρίσκονται στην κορυφή της στοίβας*)

$E.code \leftarrow E_1.code \parallel E_2.code \parallel gen('multiply');$

$E.code \leftarrow E_1.code;$

$E.code \leftarrow E_1.code \parallel gen('negate');$

$E.code \leftarrow gen('load' id.place)$



Παραγωγή κώδικα για λογικές εκφράσεις

- Δύο προσεγγίσεις:
 - Αριθμητική αναπαράσταση
 - Συνεπαγόμενη αναπαράσταση
- Αριθμητική αναπαράσταση
 - Χρησιμοποιούμε 1 για το true και 0 για το false
 - Σε κώδικα τριών διευθύνσεων αποθηκεύεται το αποτέλεσμα σε προσωρινή μεταβλητή
 - Σε κώδικα μηχανής στοίβας το αποτέλεσμα αποθηκεύεται στη στοίβα
- Συνεπαγόμενη αναπαράσταση
 - Για λογικές εκφράσεις που χρησιμοποιούνται σε έλεγχο ροής (π.χ. εντολές if, while κλπ.) και δε χρησιμοποιούνται για τον υπολογισμό ενός αποτελέσματος,
 - δημιουργούμε κώδικα που απλά διακλαδώνει στην κατάλληλη εντολή ανάλογα με το αποτέλεσμα



Λογικές εκφράσεις: αριθμητική αναπαράσταση

Ιδιότητες : $E.place$: θέση όπου βρίσκεται η τιμή για την έκφραση E
 $E.code$: σειρά εντολών που παράγονται για την E
 $id.place$: θέση του id

Καθ. μεταβλητή: $nextstat$: επιστρέφει τη θέση της νέας εντολής που θα δημιουργηθεί
 (κάθε κλήση στη $gen()$ αυξάνει τη $nextstat$ κατά 1)

Παραγωγές

$E \rightarrow id_1 \text{ relop } id_2$

Σημασιολογικοί κανόνες

$E.place \leftarrow newtemp();$
 $E.code \leftarrow gen('if' id_1.place \text{ relop.op } id_2.place \text{ 'goto' } nextstat+3);$
 $\quad \quad \quad \parallel gen(E.place \text{ ':=' } '0')$
 $\quad \quad \quad \parallel gen('goto' nextstat+2)$
 $\quad \quad \quad \parallel gen(E.place \text{ ':=' } '1');$

$E \rightarrow E_1 \text{ and } E_2$

$E.place \leftarrow newtemp();$
 $E.code \leftarrow E_1.code \parallel E_2.code$
 $\quad \quad \quad \parallel gen(E.place \text{ ':=' } E_1.place \text{ 'and' } E_2.place);$



Λογικές εκφράσεις: συνεπαγόμενη αναπαράσταση

Ιδιότητες :

- $E.code$: σειρά εντολών που παράγονται για την E
- $E.false$: ετικέτα διακλάδωσης αν η E είναι false
- $E.true$: ετικέτα διακλάδωσης αν η E είναι true
- ($E.code$ είναι συνθέσιμη ενώ οι $E.true$ και $E.false$ είναι κληρονομήσιμες)
- $id.place$: θέση του id

Παραγωγές

$E \rightarrow id_1 \text{ relop } id_2$

Σημασιολογικοί κανόνες

$E.code \leftarrow \text{gen}(\text{'if' } id_1.place \text{ relop.op } id_2.place \text{ 'goto' } E.true) \parallel \text{gen}(\text{'goto' } E.false);$

οποιοσδήποτε τελεστής συσχέτισης: =, <=, >= !=

$E \rightarrow E_1 \text{ and } E_2$

$E_1.true \leftarrow \text{newlabel}();$
 $E_1.false \leftarrow E.false;$
 $E_2.true \leftarrow E.true;$
 $E_2.false \leftarrow E.false;$
 $E.code \leftarrow E_1.code \parallel \text{gen}(E_1.true \text{ ':'}) \parallel E_2.code ;$

Τα συγκεκριμένα σημεία του κώδικα συμπληρώνονται με τις κατάλληλες ετικέτες όταν αυτές γίνουν διαθέσιμες

Η δημιουργηθείσα ετικέτα τοποθετείται στη θέση της $E_1.true$ στον κώδικα της E_1



Παράδειγμα λογικών εκφράσεων

Θέσεις εντολών τριών
διευθύνσεων και όχι ετικέτες

Λογική έκφραση εισόδου:
 $x < y$ and $a == b$

Οι ετικέτες αυτές παράγονται
σε μεταγενέστερη φάση και
τοποθετούνται στη σωστή
θέση του κώδικα

Αριθμητική αναπαράσταση:

```

100    if x < y goto 103
101    t1 := 0
102    goto 104
103    t1 := 1
104    if a = b goto 107
105    t2 := 0
106    goto 108
107    t2 := 1
108    t3 := t1 and t2
    
```

Συνεπαγόμενη αναπαράσταση:

```

        if x < y goto L1
        goto LFalse
L1:     if a = b goto LTrue
        goto LFalse
        ...
LTrue:
LFalse:
    
```





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