ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΙΚΑ
ΜΑΘΗΜΑΤΑ

Σχεδίαση Γλωσσών & Μεταγλωττιστές

Ενότητα 8: Πίνακες LR Ανάλυσης

Επ. Καθ. Π. Κατσαρός
Τμήμα Πληροφορικής

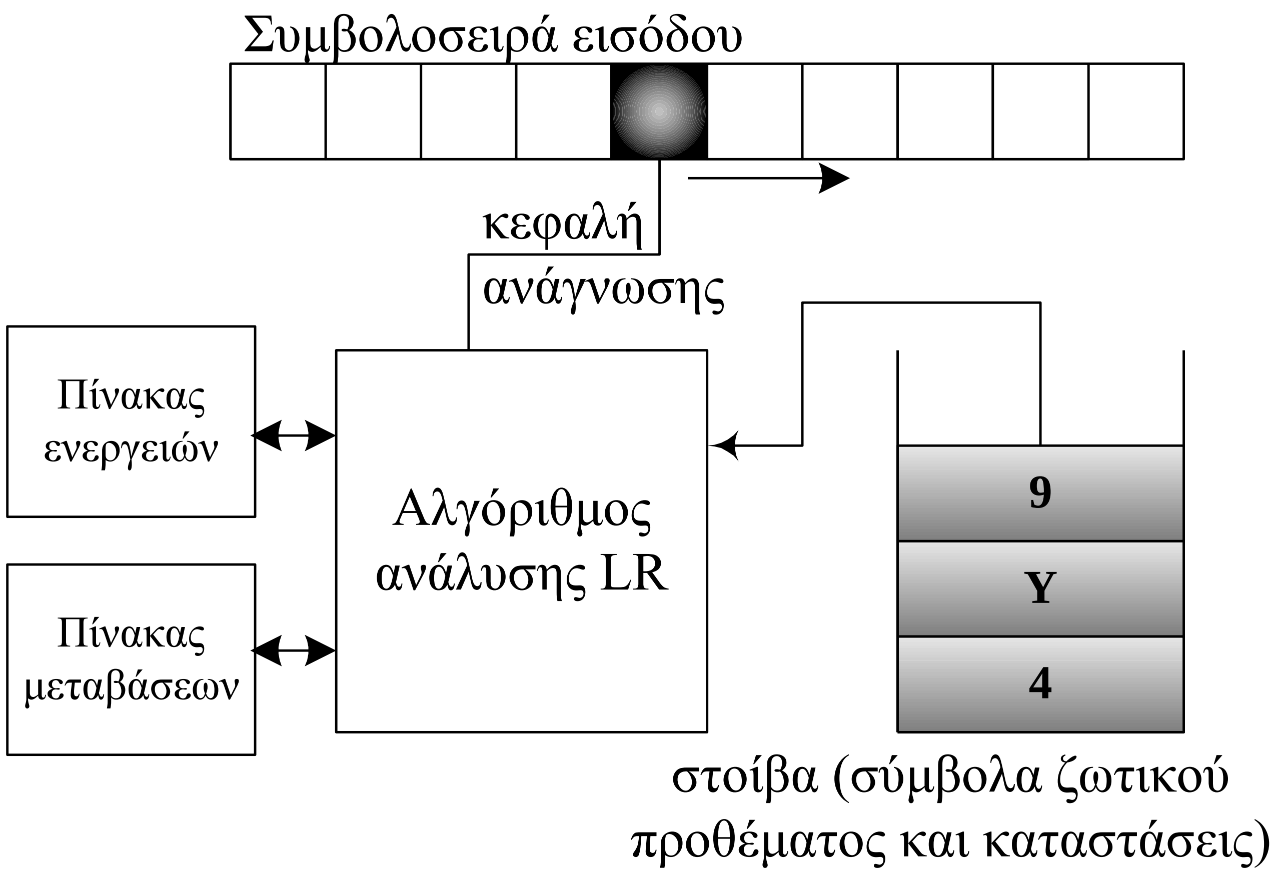
Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

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.

Αναλυτής LR ώθησης - απλοποίησης I



Αναλυτής LR ώθησης - απλοποίησης II

```
push($);                          // $ το σύμβολο τέλους συμβολοσειράς
push(s0);                         // s0 η αρχική κατάσταση
lookahead = get_next_token();
repeat forever
  s = top_of_stack();             // ACTION είναι ο πίνακας ενεργειών
  if ( ACTION[s,lookahead] = = απλοποίηση α→β ) then
      pop 2*|β| σύμβολα;          //αφαιρούνται τα σύμβολα & οι καταστάσεις
      s = top_of_stack();
      push(α);
      push(GOTO[s,α]);            // GOTO είναι ο πίνακας μεταβάσεων
  else if ( ACTION[s,lookahead] = = ώθηση si ) then
      push(lookahead);
      push(si);
      lookahead = get_next_token();
  else if ( ACTION[s,lookahead] = = αποδοχή and lookahead = = $ )
     then return (επιτυχής αναγνώριση);
  else error();
```

Αναλυτής LR ώθησης - απλοποίησης III

Μια **γραμματική είναι LR(k)** αν, δοθείσης μιας δεξιάς παραγωγής

$$S \Rightarrow \gamma_0 \Rightarrow \gamma_1 \Rightarrow \gamma_2 \Rightarrow \dots \Rightarrow \gamma_{n-1} \Rightarrow \gamma_n \Rightarrow \textit{πρόταση}$$

μπορούμε:

1. *Να αναγνωρίσουμε μια λαβή απλοποίησης για κάθε δεξιά προτασιακή μορφή* γ_i, και

2. *Να καθορίσουμε τον κανόνα της απλοποίησης*

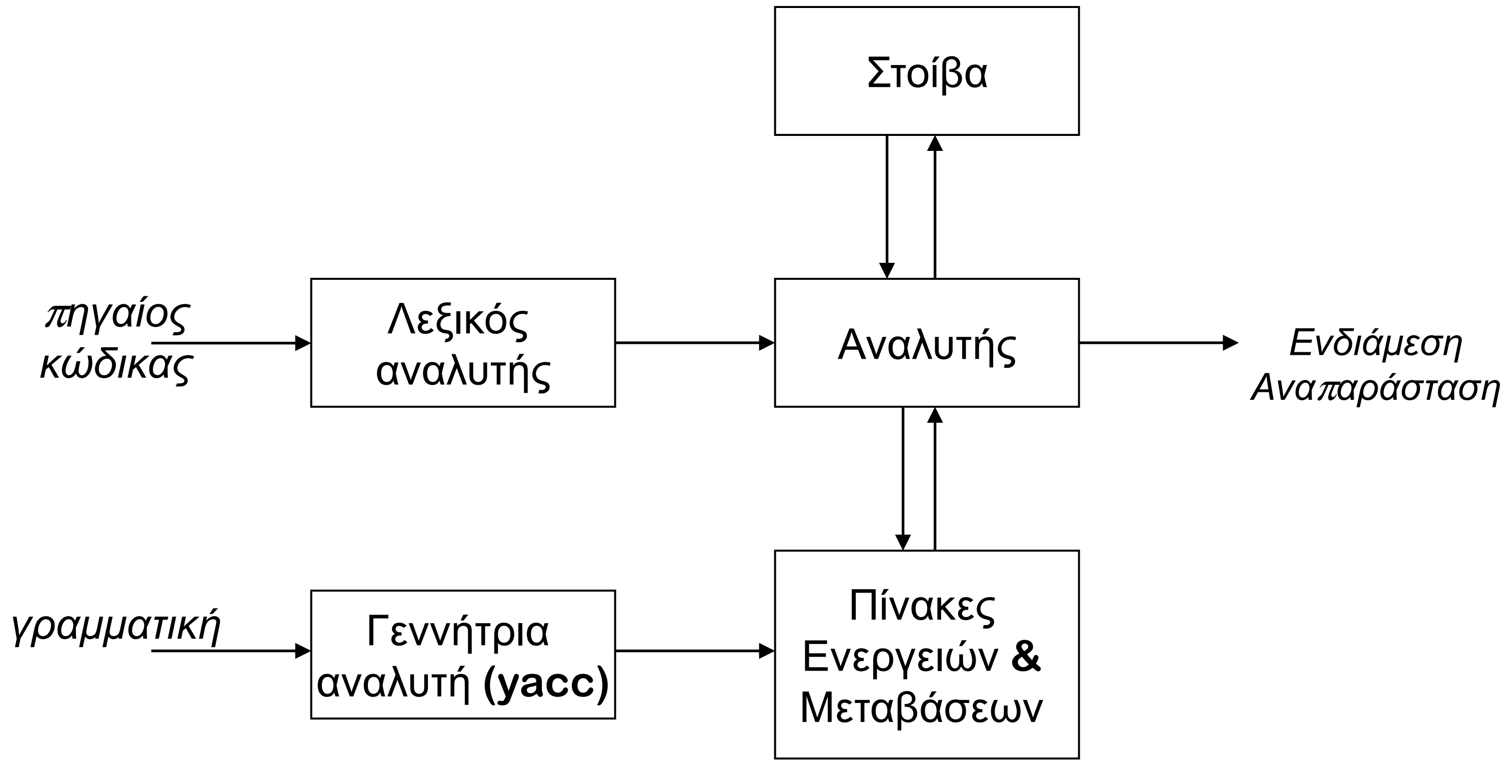
με την ανάγνωση της γ_i από αριστερά προς τα δεξιά, προχωρώντας το πολύ k σύμβολα πιο δεξιά από το τέλος της λαβής απλοποίησης της γ_i .

Αναλυτής LR ώθησης - απλοποίησης IV

- Αν μία γραμματική είναι LR(?) ⇒ υπάρχει μία και μόνο μία δεξιά παραγωγή της κάθε πρότασης
- Στο πάνω μέρος της στοίβας είτε
  - υπάρχουν όλες οι ενεργές λαβές απλοποίησης που περιλαμβάνουν την κορυφή της στοίβας, είτε
  - ωθούνται σύμβολα μέχρι να αποτελεί η κορυφή το δεξί άκρο μιας λαβής απλοποίησης
- Η γλώσσα των λαβών απλοποίησης είναι κανονική και άρα
  - μπορεί να κατασκευασθεί ένα προσδιοριστικό πεπερασμένο αυτόματο που να εκτελεί αναγνώριση λαβών απλοποίησης
  - οι πίνακες ενεργειών και μετάβασης ουσιαστικά εκφράζουν το συγκεκριμένο αυτόματο
- Κάθε νέα κατάσταση του αυτόματου την εισάγουμε στη στοίβα
- Οι τελικές καταστάσεις του αυτόματου αντιστοιχούν σε ενέργειες απλοποίησης
  - νέα κατάσταση είναι η GOTO[*αριστερό σύμβολο κανόνα* , κατάσταση κορυφής]

Αναλυτής LR ώθησης - απλοποίησης V

Πως φτιάχνουμε τους πίνακες ενεργειών & μεταβάσεων;



Αναλυτής LR ώθησης - απλοποίησης VI

Πως φτιάχνουμε τους πίνακες ενεργειών & μεταβάσεων (χωρίς το yacc);

- κατασκευάζουμε από τη γραμματική το αυτόματο αναγνώρισης λαβών απλοποίησης
- ορίζουμε τους πίνακες ενεργειών & μεταβάσεων
- αν η κατασκευή του αυτομάτου πετύχει τότε η γραμματική είναι γραμματική LR(?)
- αλλιώς είναι πιθανό να προκύψουν συγκρούσεις ώθησης – απλοποίησης ή συγκρούσεις απλοποίησης – απλοποίησης
- τότε ή θα πρέπει να εφαρμοσθεί μία άλλη κατασκευή LR(?) ή να κατασκευάσουμε τον αναλυτή έτσι ώστε σε μία περίπτωση σύγκρουσης να επιλέγει κάθε φορά μία συγκεκριμένη ενέργεια
- **οι αναλυτές του byacc όταν έχουμε σύγκρουση ώθησης – απλοποίησης εκτελούν κάθε φορά την ώθηση**

Αναλυτής LR ώθησης - απλοποίησης VII

Πως φτιάχνουμε το αυτόματο αναγνώρισης λαβών απλοποίησης (χωρίς το yacc);

- χρησιμοποιούμε στοιχεία LR(k) για να κωδικοποιήσουμε το σύνολο των κανόνων που μπορεί το δεξί τους μέρος να αποτελέσει λαβή απλοποίησης σε μία δεδομένη κατάσταση
- υπολογίζουμε για κάθε κατάσταση τα *goto(s,* α *)* και *closure(s),* δηλαδή το κλείσιμο (βλ. συνέχεια)
- καταγράφουμε τις καταστάσεις και τις μεταβάσεις του αυτόματου
- συμπληρώνουμε τους πίνακες ενεργειών & μεταβάσεων

Αναλυτής LR ώθησης - απλοποίησης VIII

Στοιχείο LR(*k*) είναι ένα ζεύγος [*A* , *B*], όπου

A είναι μία παραγωγή $\alpha \rightarrow \beta\gamma\delta$ με μία • κάπου στο δεξί μέρος

B σύμβολα εισόδου με μήκος ≤ *k* (τερματικά ή $)

Παραδείγματα:

[$\alpha \rightarrow \bullet\beta\gamma\delta$, a]

[$\alpha \rightarrow \beta\bullet\gamma\delta$, a]

[$\alpha \rightarrow \beta\gamma\bullet\delta$, a]

[$\alpha \rightarrow \beta\gamma\delta\bullet$, a]

Η • δείχνει την τρέχουσα θέση που μπορούμε να θεωρήσουμε ότι αναπαριστά η κορυφή της στοίβας

- στοιχεία LR(0) [$\alpha \rightarrow \beta \bullet \gamma\delta$] (δεν υπάρχουν σύμβολα εισόδου)
- στοιχεία LR(1) [$\alpha \rightarrow \beta \bullet \gamma\delta$, a] (ένα μόνο σύμβολο εισόδου)
- στοιχεία LR(2) [$\alpha \rightarrow \beta \bullet \gamma\delta$, a b] (δύο σύμβολα εισόδου) ...

Πίνακες ανάλυσης LR(1) I

Η παραγωγή $\alpha \to \beta\gamma\delta$, με σύμβολο εισόδου **a**, παράγει 4 στοιχεία LR(1)

[$\alpha \to \bullet\beta\gamma\delta$, **a**], [$\alpha \to \beta\bullet\gamma\delta$, **a**], [$\alpha \to \beta\gamma\bullet\delta$, **a**], & [$\alpha \to \beta\gamma\delta\bullet$, **a**]

Το σύνολο των LR(1) στοιχείων μιας γραμματικής είναι πεπερασμένο.

Ποια είναι η σημασία των συμβόλων εισόδου στα στοιχεία;

- τα καταγράφουμε έτσι ώστε να οδηγούμαστε στη σωστή απλοποίηση
- αν η • βρίσκεται σε ενδιάμεση θέση σε ένα μόνο στοιχείο δεν παίζει κάποιο ιδιαίτερο ρόλο
  - π.χ. στο [$\alpha \to \beta\gamma\bullet\delta$, **a**], το σύμβολο εισόδου δεν έχει ιδιαίτερη σημασία
  - στο [$\alpha \to \beta\gamma\delta\bullet$, **a**], το σύμβολο **a** οδηγεί στην απλοποίηση $\alpha \to \beta\gamma\delta$
  - στο σύνολο { [$\alpha \to \gamma\bullet$, **a**],[$\beta \to \gamma\bullet\delta$, **b**] }

    συμβόλο εισόδου = **a** $\Rightarrow$ ***απλοποίηση*** σε α;

    σύμβολο εισόδου $\in$ FIRST(δ) $\Rightarrow$ ***ώθηση***

$\Rightarrow$ σε μία LR(1) γραμματική ένα σύμβολο εισόδου είναι αρκετό για να καθορίσει την επόμενη ενέργεια

Πίνακες ανάλυσης LR(1) II

1 Οι καταστάσεις του προσδιοριστικού αυτόματου αναγνώρισης λαβών απλοποίησης ορίζονται ως σύνολα στοιχείων, $I_0, I_1, \ldots, I_n$

a. Εισάγουμε ένα νέο σύμβολο *S′* και μία παραγωγή

$S' \rightarrow S$ *όπου S η αρχή της γραμματικής*

b. Η αρχική κατάσταση, I_0 περιλαμβάνει τα στοιχεία

- $[S' \rightarrow \bullet S, \$]$ και όλα τα ισοδύναμα που δίνονται ως
- *closure*(I_0)

c. Για κάθε σύνολο I_k και κάθε σύμβολο α, βρίσκουμε το *goto*(I_k, α)

- αν το σύνολο της *goto*(I_k, α) δεν έχει ήδη δημιουργηθεί τότε το δημιουργούμε
- καταγράφουμε όλες τις μεταβάσεις *goto( )*

2 Συμπληρώνουμε τους πίνακες ενεργειών και μεταβάσεων της ανάλυσης

Πίνακες ανάλυσης LR(1) III

ΥΠΟΛΟΓΙΣΜΟΣ closure(I)

Το *closure(I)* προσθέτει στο σύνολο *I* όλα τα στοιχεία που προκύπτουν από τα ήδη υπάρχοντα στο *I*

- κάθε στοιχείο [$\alpha \rightarrow \beta \bullet \gamma\delta$, a] δημιουργεί επίσης όλα τα [$\gamma \rightarrow \bullet\ \tau$, *x*] για κάθε παραγωγή με γ στα αριστερά και για όλα τα $x \in$ FIRST(δa)
- εφόσον συμπεριληφθούν όλα τα στοιχεία της παραγωγής γ περιλαμβάνονται και όσα προκύπτουν από αυτά εφαρμόζοντας την ίδια πράξη

```
Closure( I )
  while ( το σύνολο I μεγαλώνει )
    for κάθε [α → β • γδ , a] ∈ I
      for κάθε παραγωγή γ → τ ∈ P
        for κάθε τερματικό b ∈ FIRST(δa)
          if [γ → • τ , b] ∉ I
            then προσέθεσε το [γ → • τ , b] στο I
```

Πίνακες ανάλυσης LR(1) IV

ΥΠΟΛΟΓΙΣΜΟΣ goto(I,x)

Υπολογίζει τη νέα κατάσταση που ο αναλυτής προσεγγίζει αν αυτός βρίσκεται στην κατάσταση *I* και αναγνωρίζει το τερματικό *x*

- *goto(* { [$\alpha \to \beta \bullet \gamma\delta$, a] }, γ *)* παράγει κατάσταση που περιλαμβάνει το [$\alpha \to \beta\gamma \bullet \delta$, a]
- αν δεν υπάρχει δημιουργείται και περιλαμβάνεται σε αυτή και το *closure(* [$\alpha \to \beta\gamma \bullet \delta$, a] *)*

```
Goto( I, x )
   new = Ø
    for κάθε [α → β • x δ , a] ∈ I
           new = new ∪ [α → βx • δ , a]
    return closure(new)
```

Πίνακες ανάλυσης LR(1) V

ΑΛΓΟΡΙΘΜΟΣ ΚΑΤΑΣΚΕΥΗΣ ΑΥΤΟΜΑΤΟΥ

Ξεκινάμε από το $I_0 = closure([S' \rightarrow \bullet S , \$])$

Δημιουργούμε συνέχεια νέες καταστάσεις μέχρι τη στιγμή που δεν θα μπορεί να δημιουργηθούν άλλες

I_0 = closure([$S' \rightarrow \bullet S$, $])
$C = \{ I_0 \}$

while (*προστίθενται στο C νέα σύνολα στοιχείων*)
  for κάθε $I_i \in C$ και κάθε $x \in (T \cup NT)$
    I_{new} = goto(I_i, x)
    if $I_{new} \notin C$ then
      $C = C \cup I_{new}$
      κατέγραψε τη μετάβαση $I_i \rightarrow I_{new}$ με αναγνώριση του x

Πίνακες ανάλυσης LR(1) VI

ΑΛΓΟΡΙΘΜΟΙ ΚΑΤΑΣΚΕΥΗΣ ΠΙΝΑΚΩΝ LR(1)

Closure(I)
while (*το σύνολο* I μεγαλώνει)
for κάθε $[\alpha \rightarrow \beta \bullet \gamma\delta \text{ , a}] \in I$
for κάθε παραγωγή $\gamma \rightarrow \tau \in P$
for κάθε τερματικό b $\in$ FIRST(δa)
if $[\gamma \rightarrow \bullet \tau \text{ , b}] \notin I$
then προσέθεσε το $[\gamma \rightarrow \bullet \tau \text{ , b}]$ στο I

Goto(I, x)
new = Ø
for κάθε $[\alpha \rightarrow \beta \bullet x\, \delta \text{ , a}] \in I$
new = new $\cup$ $[\alpha \rightarrow \beta x \bullet \delta \text{ , a}]$
return closure(new)

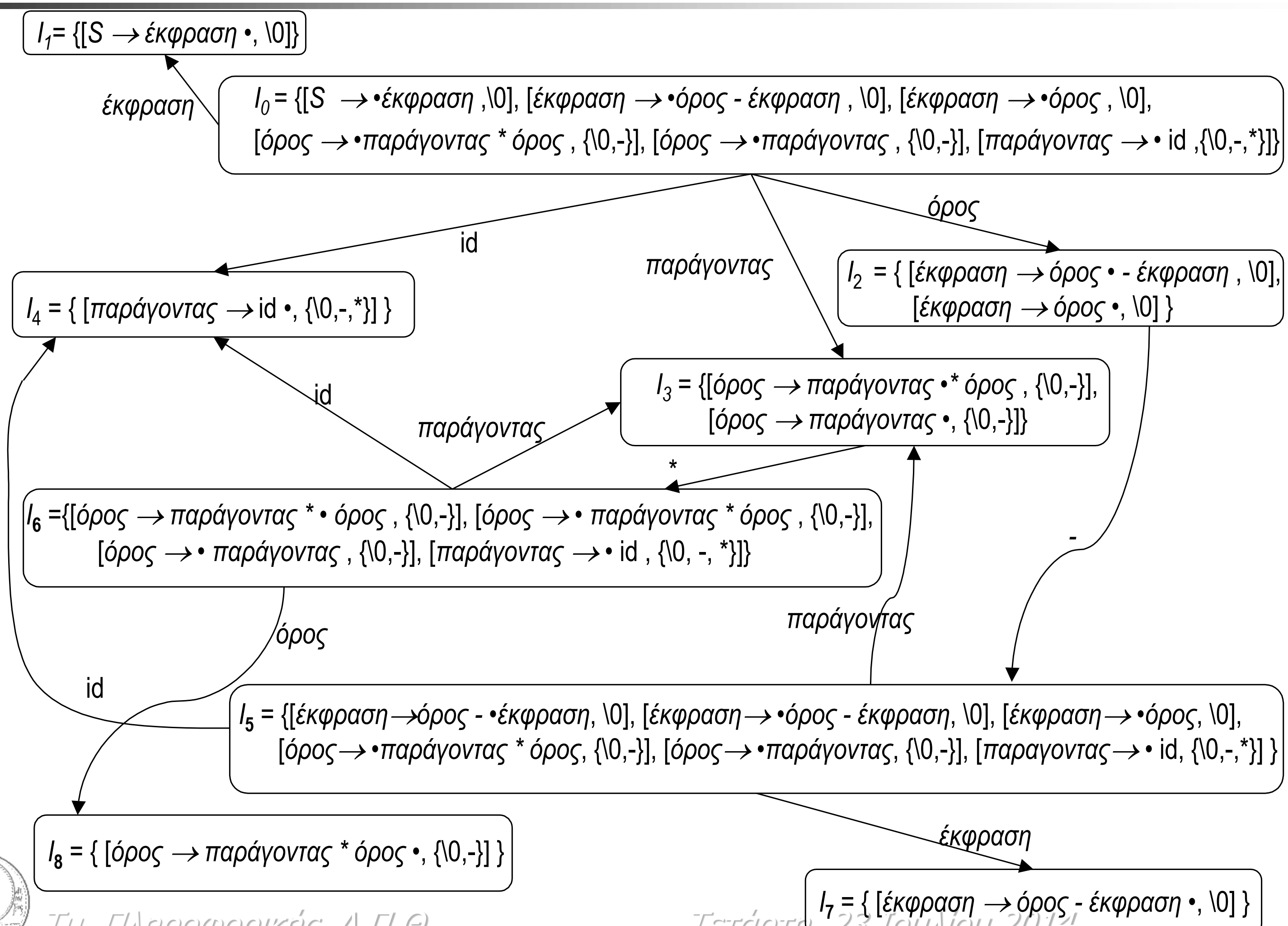
I_0 = closure($[S' \rightarrow \bullet S \text{ , \$}]$)
C = { I_0 }
while (*προστίθενται στο C νέα σύνολα στοιχείων*)
for κάθε $I_i \in C$ και κάθε $x \in (T \cup NT)$
I_{new} = goto(I_i, x)
if $I_{new} \notin C$ then
$C = C \cup I_{new}$
κατέγραψε τη μετάβαση $I_i \rightarrow I_{new}$ με αναγνώριση του x

Παράδειγμα ανάλυσης LR(1) I

$S \rightarrow$ *έκφραση*
έκφραση $\rightarrow$ *όρος* - *έκφραση*
έκφραση $\rightarrow$ *όρος*
όρος $\rightarrow$ *παράγοντας* * *όρος*
όρος $\rightarrow$ *παράγοντας*
παράγοντας $\rightarrow$ **id**

Σύμβολο	**FIRST**
S	**{ id }**
έκφραση	**{ id}**
όρος	**{ id }**
παράγοντας	**{ id}**
-	**{ - }**
*****	**{ * }**
id	**{ id}**

Παράδειγμα ανάλυσης LR(1) II



Παράδειγμα ανάλυσης LR(1) III

ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΩΝ

for *κάθε σύνολο στοιχείων* $I_x \in C$
  for κάθε στοιχείο $\in I_x$
    if *στοιχείο είναι* **[**$\alpha \to \beta$ •**a**γ,**b]** και **a** $\in T$ και **goto(**I_x,**a) =** I_k **,**
      then ACTION[x**,a]** ← "ώθηση k"
    else if *στοιχείο είναι* **[**$S' \to S$ **•,$]**
      then ACTION[x **,$]** ← "αποδοχή"
    else if *στοιχείο είναι* **[**$\alpha \to \beta$ **•,a]**
      then ACTION[x**,a]** ← "απλοποίηση $\alpha \to \beta$"
  for κάθε $n \in NT$
    if goto(I_x **,**n**) ==** I_k
      then GOTO[x**,n]** ← k

Παράδειγμα ανάλυσης LR(1) IV

	ΕΝΕΡΓΕΙΕΣ				ΜΕΤΑΒΑΣΕΙΣ		
	id	-	*	$	*έκφραση*	*όρος*	*παράγοντας*
0	ωθ 4				1	2	3
1				αποδ			
2		ωθ 5		απ 3			
3		απ 5	ωθ 6	απ 5			
4		απ 6	απ 6	απ 6			
5	ωθ 4				7	2	3
6	ωθ 4					8	3
7				απ 2			
8		απ 4		απ 4			

Συγκρούσεις ενεργειών

Τι γίνεται αν η κατάσταση *s* περιέχει [$\alpha \rightarrow \beta \bullet a\gamma$, b] και [$\alpha \rightarrow \beta \bullet$, a] ;

- Το πρώτο στοιχείο παράγει "ώθηση" και το δεύτερο "απλοποίηση"
- Δεν μπορούμε στο ACTION[*s*,a] να συμπληρώσουμε δύο ενέργειες
- **Σύγκρουση ώθησης – απλοποίησης** (shift – reduce conflict)
- Μπορούμε ή να αλλάξουμε τη γραμματική ή
- αυθαίρετα να επιλέξουμε την εκτέλεση ώθησης, κάτι που έχει αποδειχθεί ότι συνήθως αποδίδει τη σωστή σημασία (π.χ. Πρόβλημα μετέωρου else) ή
- να επιλέξουμε κάποια άλλη ανάλυση LR

Τι γίνεται όταν η κατάσταση s περιέχει [$\alpha \rightarrow \gamma \bullet$, a] και [$\beta \rightarrow \gamma \bullet$, a] ;

- Κάθε στοιχείο παράγει "απλοποίηση", αλλά με διαφορετικό κανόνα
- Δεν μπορούμε στο ACTION[*s*,a] να συμπληρώσουμε δύο ενέργειες
- **Σύγκρουση απλοποίησης – απλοποίησης** (reduce – reduce conflict)
- Ένδειξη λάθους στη γραμματική

και στις δύο περιπτώσεις λέμε ότι η γραμματική δεν είναι LR(1)

Παραλλαγές ανάλυσης LR: LR(0)

κλείσιμο συνόλου στοιχείων LR(0):

```
Closure( I )
   while ( το σύνολο I μεγαλώνει )
      for κάθε στοιχείο [α → β • γδ ] ∈ I
         for κάθε παραγωγή γ → τ ∈ P
            if [γ → • τ ] ∉ I
               then προσέθεσε το [γ → • τ ] στο I
```

κατασκευή συνόλων στοιχείων LR(0):

```
I_0 = closure( [S' → • S ] )
C = { I_0 }
while ( προστίθενται στο C νέα σύνολα στοιχείων)
   for κάθε I_i ∈ C και κάθε x ∈ ( T∪NT )
      I_new = goto(I_i , x)
      if I_new ∉ C then
         C = C ∪ I_new
         κατέγραψε τη μετάβαση I_i → I_new με
               αναγνώριση του x
```

μεταβάσεις συνόλων στοιχείων LR(0):

```
Goto( I, x )
   new = Ø
   for κάθε [α → β • x δ ] ∈ I
         new = new ∪ [α → βx • δ ]
   return closure(new)
```

- Βασικά είναι οι ίδιοι αλγόριθμοι όπως και για την ανάλυση LR(1)
- Η μόνη διαφορά είναι ότι στα στοιχεία LR(0) δεν υπάρχει σύμβολο εισόδου και γι΄ αυτό δεν παράγονται πολλά στοιχεία σε κάθε σύνολο
- Η ανάλυση LR(0) παράγει λιγότερες καταστάσεις
- Δεν μπορεί να αναγνωρίσει όλες τις γραμματικές που αναγνωρίζει η LR(1)
- Είναι πιο πιθανό να οδηγήσει σε συγκρούσεις ανάλυσης

Παραλλαγές ανάλυσης LR: SLR(1)

- Ο αλγόριθμος SLR(1) για να καθορίσει πότε θα γίνει απλοποίηση χρησιμοποιεί και τα σύνολα FOLLOW
- Οι αναλυτές SLR(1) έχουν λιγότερες καταστάσεις από ότι οι LR(1)

```
for κάθε σύνολο στοιχείων Ix ∈ C
   for κάθε στοιχείο ∈ Ix
      if  στοιχείο είναι [α→β •aγ] και a ∈ T και goto(Ix,a) = Ik ,
          then ACTION[x,a] ← "ώθηση k"
      else if  στοιχείο είναι [S'→S •]
          then ACTION[x ,\0] ← "αποδοχή"
      else if  στοιχείο είναι [α→β •]
        then for κάθε a ∈ FOLLOW(α)
           then ACTION[x,a] ← "απλοποίηση α→β"

   for each n ∈ NT
      if  goto(Ix ,n) = Ik
         then GOTO[x,n] ← k
```

Παραλλαγές ανάλυσης LR: LALR(1)

- Βασική ιδέα: συνένωση καταστάσεων LR(1)
  - κρατάμε τον πυρήνα LR(0) των στοιχείων LR(1) (αγνοούμε τα σύμβολα εισόδου)
  - αν δύο σύνολα LR(1) έχουν τον ίδιο πυρήνα, τότε τα συνενώνουμε και ενημερώνουμε τους πίνακες ACTION και GOTO
- Οι αναλυτές LALR(1) μπορούν να φτιαχτούν με δύο τρόπους
  1. Κατασκευή των συνόλων στοιχείων LR(1) και συνένωση
  2. Αγνοούμε τα στοιχεία με τελεία στην αρχή του δεξιού μέρους και κατασκευάζουμε πυρήνες συνόλων στοιχείων LR(0). Στη συνέχεια χρησιμοποιούμε έναν αλγόριθμο προώθησης συμβόλων εισόδου για να υπολογίσουμε τα σύμβολα εισόδου.

Η δεύτερη προσέγγιση είναι πιο αποδοτική επειδή αποφεύγει τη δημιουργία μεγάλων ενδιάμεσων LR(1) πινάκων

Παραλλαγές ανάλυσης LR: LALR(1)

- Ένας αναλυτής LALR(1) για μια γραμματική G έχει τον ίδιο αριθμό καταστάσεων με τον αντίστοιχο αναλυτή SLR(1)
- Αν ένας αναλυτής LR(1) για μια γραμματική G δεν παρουσιάζει συγκρούσεις ώθησης - απλοποίησης, τότε δεν θα έχει συγκρούσεις ώθησης – απλοποίησης και ο αναλυτής LALR(1)
- Ένας αναλυτής LALR(1) μπορεί να εμφανίζει σύγκρουση απλοποίησης - απλοποίησης χωρίς αυτό να συμβαίνει και στον αντίστοιχο LR(1)
- Οι αναλυτές LALR(1) είναι πιο γενικοί από τους SLR(1) και συγχρόνως έχουν το ίδιο μικρό χώρο καταστάσεων
- Ανάλυση LALR(1) παράγουν οι γεννήτριες yacc και byacc

Γλώσσες χωρίς συμφραζόμενα I

Γλώσσες χωρίς συμφραζόμενα

↓

Προσδιοριστικές γλώσσες (**LR(*k*)**)

LR(*k*) ≡ LR(1)

↙ ↘

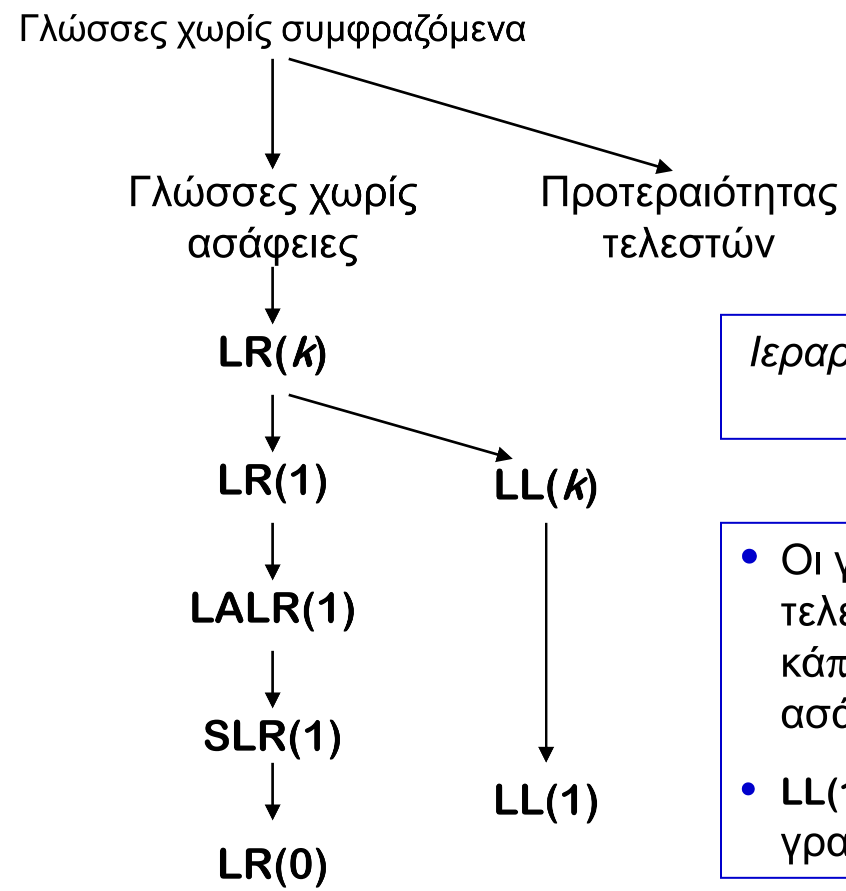
Γλώσσες **LL(*k*)** | Γλώσσες απλής προήγησης

↓ ↓

Γλώσσες **LL(1)** | Γλώσσες προτεραιότητας τελεστών

Η ιεραρχία γλωσσών χωρίς συμφραζόμενα

Γλώσσες χωρίς συμφραζόμενα II



Ιεραρχία γλωσσών χωρίς συμφραζόμενα

- Οι γλώσσες προτεραιότητας τελεστών περιλαμβάνουν και κάποιες γλώσσες με ασάφειες
- **LL(1)** είναι υποσύνολο των γραμματικών **SLR(1)**

Τέλος ενότητας

Επεξεργασία: Εμμανουέλα Στάχτιαρη
Θεσσαλονίκη, 21/07/2014

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης