



# HY340 : ΓΛΩΣΣΕΣ ΚΑΙ ΜΕΤΑΦΡΑΣΤΕΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ,  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ,  
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
VAR i:Integer;  
FUNCTION(Symbol) replicate  
    x = (function(x,y){return x+y;});  
    class DelFunctor: public std::unary_function<
```

ΔΙΔΑΣΚΩΝ

Αντώνιος Σαββίδης



# **HY340 : ΓΛΩΣΣΕΣ ΚΑΙ ΜΕΤΑΦΡΑΣΤΕΣ**

## **Διάλεξη 7η ΣΥΝΤΑΚΤΙΚΗ ΑΝΑΛΥΣΗ – IV (τελευταία)**



# Περιεχόμενα

---

- *Κατασκευή SLR parse table*
- Χρήση του SLR table
- Οι βελτιώσεις του SLR parser





## Κατασκευή SLR parse table (1/16)

- Για την κατασκευή θα ακολουθήσουμε δύο βήματα
  - Χρειαζόμαστε ένα DFA το οποίο να αναγνωρίζει τα εφικτά προθέματα της γραμματικής
    - ◆ Ουσιαστικά το DFA θα ξεκινά από ένα viable prefix
    - ◆ Θα καταναλώνει είσοδο (shift, transitions)
    - ◆ Οδηγώντας στο προηγούμενο viable prefix (reduction)
  - Θα φτιάξουμε το DFA (δύσκολο)
  - Μετά θα το μετατρέψουμε σε έναν parse table (εύκολο)



## Κατασκευή SLR parse table (2/16)

**Ορισμός.** Ένα LR(0) item, ή απλώς item, μία γραμματικής  $G$  είναι μία παραγωγή της  $G$  με μία τελεία σε κάποιο σημείο στο δεξί τμήμα της παραγωγής. Π.χ. η παραγωγή  $A \rightarrow XYZ$  οδηγεί στα εξής items:

$A \rightarrow \cdot XYZ$

$A \rightarrow X \cdot YZ$

$A \rightarrow XY \cdot Z$

$A \rightarrow XYZ \cdot$

Το κάθε item μίας παραγωγής αποτυπώνει και μία ξεχωριστή κατάσταση σχετικά με «το κομμάτι της δεδομένης παραγωγής που έχουμε ήδη αναγνωρίσει κατά τη συντακτική ανάλυση»:

- Το  $A \rightarrow \cdot XYZ$  δηλώνει ότι αναμένουμε να συναντήσουμε στην συνέχεια της εισόδου μια λέξη που παράγεται από το  $XYZ$
- Το  $A \rightarrow X \cdot YZ$  δηλώνει ότι έχουμε μόλις δει μία λέξη που παράγεται από το  $X$  και αναμένουμε στην συνέχεια της εισόδου μια λέξη που παράγεται από το  $YZ$



# Κατασκευή SLR parse table (3/16)



Εύκολα παρατηρούμε ότι για κάθε παραγωγή με  $N$  γραμματικά σύμβολα στο RHS προκύπτουν  $N+1$  αντίστοιχα LR(0) items





## Κατασκευή SLR parse table (4/16)

*Συμβολισμοί.* Εάν γράφουμε  $A \rightarrow \alpha c \beta$  ή  $A \rightarrow \alpha B \beta$  τότε:

- $\alpha$  και  $\beta$  είναι ακολουθίες γραμματικών συμβόλων (από τερματικά και μη τερματικά σύμβολα)
- Το  $c$  είναι ένα τερματικό σύμβολο
- Το  $B$  είναι ένα γραμματικό σύμβολο (τερματικό ή μη τερματικό)

- Οι λατινικοί lower-case χαρακτήρες αντιστοιχούν σε τερματικά σύμβολα.
- Τα Ελληνικά μικρά γράμματα σε ακολουθίες γραμματικών συμβόλων.
- Τα κεφαλαία γράμματα σε ένα γραμματικό σύμβολο.



# Κατασκευή SLR parse table (5/16)

*Η βασική ιδέα σχετικά με τη χρησιμότητα των items στην κατασκευή του FA*

- Οι καταστάσεις αντιστοιχούν σε σύνολα από items
- 1. Εάν μία κατάσταση περιέχει το item  $A \rightarrow \alpha \bullet c \beta$  τότε
  - Ο αναλυτής αναμένει περιμένει τελικά να κάνει αναγωγή με την παραγωγή  $A \rightarrow \alpha c \beta$
  - Ο αναλυτής έχει ήδη κάνει συντακτική ανάλυση «ενός»  $\alpha$
  - Αναμένει η είσοδος να περιέχει το  $c$  και έπειτα «ένα»  $\beta$
- 2. Εάν μία κατάσταση περιέχει το item  $A \rightarrow \alpha \bullet$  τότε
  - Ο αναλυτής έχει ήδη κάνει συντακτική ανάλυση «ενός»  $\alpha$
  - Θα κάνει αναγωγή για την παραγωγή  $A \rightarrow \alpha$
- 3. Εάν μία κατάσταση περιέχει το item  $S \rightarrow \alpha \bullet$  και δεν έχουμε άλλα σύμβολα εισόδου (empty input buffer) τότε
  - Η είσοδος γίνεται δεκτή - accept

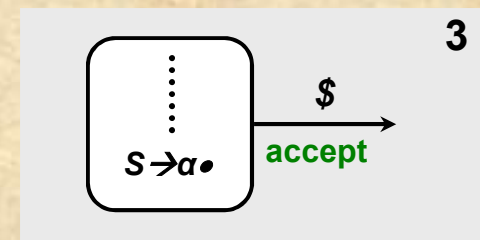
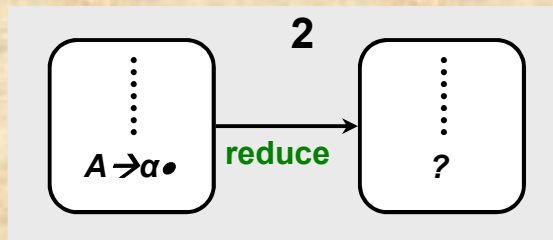
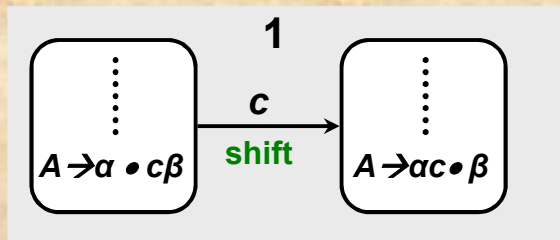




# Κατασκευή SLR parse table (6/16)

Συσχέτιση των *items* με μεταβάσεις του αυτομάτου συντακτικής ανάλυσης

1. Εάν η παρούσα κατάσταση περιέχει το item  $A \rightarrow \alpha \bullet c \beta$  και το σύμβολο στην είσοδο είναι το  $c$  τότε
  - Ο αναλυτής κάνει shift το  $c$  στη στοίβα
  - Η επόμενη κατάσταση θα περιέχει το  $A \rightarrow \alpha c \bullet \beta$
2. Εάν η παρούσα κατάσταση περιέχει το item  $A \rightarrow \alpha \bullet$  τότε
  - Ο αναλυτής κάνει αναγωγή για την παραγωγή  $A \rightarrow \alpha$
3. Εάν η παρούσα κατάσταση περιέχει το item  $S \rightarrow \alpha \bullet$  και δεν έχουμε άλλα σύμβολα εισόδου τότε
  - Ο αναλυτής κάνει accept

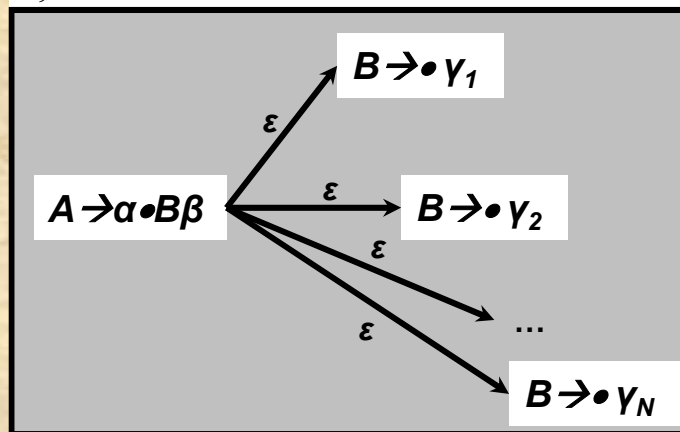




# Κατασκευή SLR parse table (7/16)

## Προσδιορισμός των *items* της ίδιας κατάστασης

```
closure(I) {  
  C = {I};  
  repeat {  
    added = false;  
    for κάθε  $A \rightarrow \alpha \bullet B \beta \in C$  do  
      for κάθε παραγωγή  $B \rightarrow \gamma$  με  $B \rightarrow \bullet \gamma \notin C$  do {  
         $C = C \cup \{ B \rightarrow \bullet \gamma \}$ ;  
        added = true;  
      }  
  }  
  until added = false;  
}
```



Πρακτικά, τα *items* μίας κατάστασης σηματοδοτούν την ίδια θέση αναμονής για επόμενες παραγωγές στη συντακτική ανάλυση

Όταν ο αναλυτής είναι στην κατάσταση  $A \rightarrow \alpha \bullet B \beta$ , θα μπορούσε να είναι σε οποιαδήποτε από αυτές του  $\text{closure}(A \rightarrow \alpha \bullet B \beta)$ , γεγονός που προφανώς καθιστά το αυτόματο σε μη αιτιοκρατικό (NFA).



## Κατασκευή SLR parse table (8/16)

*Επεξηγήσεις.* Το *closure* εμπεριέχει όλα ακριβώς τα items που αντιπροσωπεύουν ταυτόχρονα την ίδια «κατάσταση» - δηλ. έχουμε ένα NFA. Η ύπαρξη ενός item  $A \rightarrow \alpha \bullet B\beta$  στο  $closure(I)$  υποδεικνύει ότι σε κάποιο σημείο της ανάλυσης αναμένουμε την ενδεχόμενη παρουσία στην είσοδο ακολουθίας συμβόλων που παράγεται από το  $B\beta$ . Εάν  $B \rightarrow \gamma$  είναι μία παραγωγή, επίσης αναμένουμε την παρουσία λέξης που παράγεται από το  $\gamma$ , επαναλαμβάνοντας αναδρομικά.

Θα προσθέσουμε στη γραμματική μία νέα παραγωγή  $S' \rightarrow S$ , με  $S'$  το νέο αρχικό σύμβολο, ονομάζοντας μία τέτοια γραμματική επαυξημένη. Ο ρόλος αυτής της νέας παραγωγής είναι να υποδεικνύει στον parser πότε να σταματήσει επιτυχώς - accept, δηλ. ακριβώς μόλις γίνει αναγωγή για την παραγωγή  $S' \rightarrow S$ . Η επιλογή της ονομασίας για το  $S'$  είναι ελεύθερη.





# Κατασκευή SLR parse table (9/16)

Γραμματική αριθμητικών εκφράσεων  
με αριστερή προσηταιριστικότητα και  
προτεραιότητα τελεστών

## Παραδείγματα για το closure

Γραμματική ισορροπημένων  
φωλιασμένων παρενθέσεων

$S$	$\rightarrow$	$expr$
$expr$	$\rightarrow$	$expr + term \mid term$
$term$	$\rightarrow$	$term * prim \mid prim$
$prim$	$\rightarrow$	$(expr) \mid num$

$S$	$\rightarrow$	$X$
$X$	$\rightarrow$	$(X)$
$X$	$\rightarrow$	$( )$

### $closure(S \rightarrow \bullet expr)$

$S$	$\rightarrow$	$\bullet expr$
$expr$	$\rightarrow$	$\bullet expr + term$
$expr$	$\rightarrow$	$\bullet term$
$term$	$\rightarrow$	$\bullet term * prim$
$term$	$\rightarrow$	$\bullet prim$
$prim$	$\rightarrow$	$\bullet ( expr )$
$prim$	$\rightarrow$	$\bullet num$

### $closure(X \rightarrow (\bullet X))$

$X$	$\rightarrow$	$(\bullet X)$
$X$	$\rightarrow$	$\bullet (X)$
$X$	$\rightarrow$	$\bullet ( )$

### $closure(S \rightarrow \bullet X)$

$S$	$\rightarrow$	$\bullet X$
$X$	$\rightarrow$	$\bullet (X)$
$X$	$\rightarrow$	$\bullet ( )$

Αναμενόμενα καθώς περιμένουμε  
παραγωγές από όλους τους κανόνες!



# Κατασκευή SLR parse table (10/16)

$goto(I, X)$ . Αυτή είναι η επόμενη πολύ χρήσιμη συνάρτηση η οποία προσδιορίζει την νέα κατάσταση μετά την «κατανάλωση» ενός γραμματικού συμβόλου  $X$  (ουσιαστικά την μετακίνηση του  $\bullet$  «πάνω» από το  $X$ ). Θεωρούμε ότι η παρούσα κατάσταση αντιστοιχεί στο σύνολο από items  $I$ . Υπολογίζεται ως εξής:

$$goto(I, X) = closure(\text{Όλων των } A \rightarrow \alpha X \bullet \beta \text{ για } A \rightarrow \alpha \bullet X \beta \in I)$$

$I = \{ S \rightarrow expr \bullet, expr \rightarrow expr \bullet + term \}$

$X = +$

το  $goto(I, X)$  είναι  $closure(\{expr \rightarrow expr + \bullet term\})$

$expr \rightarrow expr + \bullet term$

$term \rightarrow \bullet term * prim$

$term \rightarrow \bullet prim$

$prim \rightarrow \bullet ( expr )$

$prim \rightarrow \bullet num$

$goto(\{X \rightarrow (\bullet X)\}, X)$

$X \rightarrow (X \bullet)$

$goto(\{X \rightarrow \bullet (X)\}, ( )$

$X \rightarrow (\bullet X)$

$X \rightarrow \bullet (X)$

$X \rightarrow \bullet ( )$



# Κατασκευή SLR parse table (11/16)

Ο αλγόριθμος υπολογισμού του DFA αναγνώρισης των εφικτών προθεμάτων της  $G$ . Ο τρόπος υπολογισμού είναι πρακτικά με subset construction πάνω στο αρχικό NFA, για το οποίο είχαμε ήδη όλες τις καταστάσεις και τη λογική μετάβασης (αλλά και πολλές κενές μεταβάσεις).

```
dfa_states( $G$  επανζημένη γραμματική) {  
    states = { closure( $S' \rightarrow S$ ) };  
    repeat {  
        added = false;  
        for κάθε ένα  $I \in states$  do  
            for κάθε γραμματικό σύμβολο  $X$  do {  
                 $P = goto(I, X)$ ;  
                if  $P \neq \emptyset$  και  $P \notin states$  then {  
                    states = states  $\cup$   $P$ ;  
                    added = true;  
                    Πρόσθεσε ακμή  $I \rightarrow P$  με label  $X$ ;  
                }  
            }  
        }  
    }  
    until added = false;  
}
```

Το σύνολο των DFA states που παράγεται από τον αλγόριθμο αυτό λέγεται και «κανονικοποιημένη συλλογή LR(0) items γραμματικής»

Κάθε  $P$  που παράγεται είναι ένα σύνολο από items της ίδιας κατάστασης





# Κατασκευή SLR parse table (12/16)

$I_1$		
$S$	$\rightarrow$	$expr \bullet$
$expr$	$\rightarrow$	$expr \bullet + term$

$I_2$		
$expr$	$\rightarrow$	$term \bullet$
$term$	$\rightarrow$	$term \bullet * prim$

$I_3$		
$term$	$\rightarrow$	$prim \bullet$

$I_4$		
$prim$	$\rightarrow$	$(\bullet expr)$
$expr$	$\rightarrow$	$\bullet expr + term$
$expr$	$\rightarrow$	$\bullet term$
$term$	$\rightarrow$	$\bullet term * prim$
$term$	$\rightarrow$	$\bullet prim$
$prim$	$\rightarrow$	$\bullet (expr)$
$prim$	$\rightarrow$	$\bullet num$

$I_5$		
$prim$	$\rightarrow$	$num \bullet$

$I_6$		
$expr$	$\rightarrow$	$expr + \bullet term$
$term$	$\rightarrow$	$\bullet term * prim$
$term$	$\rightarrow$	$\bullet prim$
$prim$	$\rightarrow$	$\bullet (expr)$
$prim$	$\rightarrow$	$\bullet num$

$I_7$		
$term$	$\rightarrow$	$term * \bullet prim$
$prim$	$\rightarrow$	$\bullet (expr)$
$prim$	$\rightarrow$	$\bullet num$

$I_8$		
$prim$	$\rightarrow$	$(expr \bullet)$
$expr$	$\rightarrow$	$expr \bullet + term$

$I_9$		
$expr$	$\rightarrow$	$expr + term \bullet$
$term$	$\rightarrow$	$term \bullet * prim$

$I_{10}$		
$term$	$\rightarrow$	$term * prim \bullet$

$I_{11}$		
$prim$	$\rightarrow$	$(expr) \bullet$

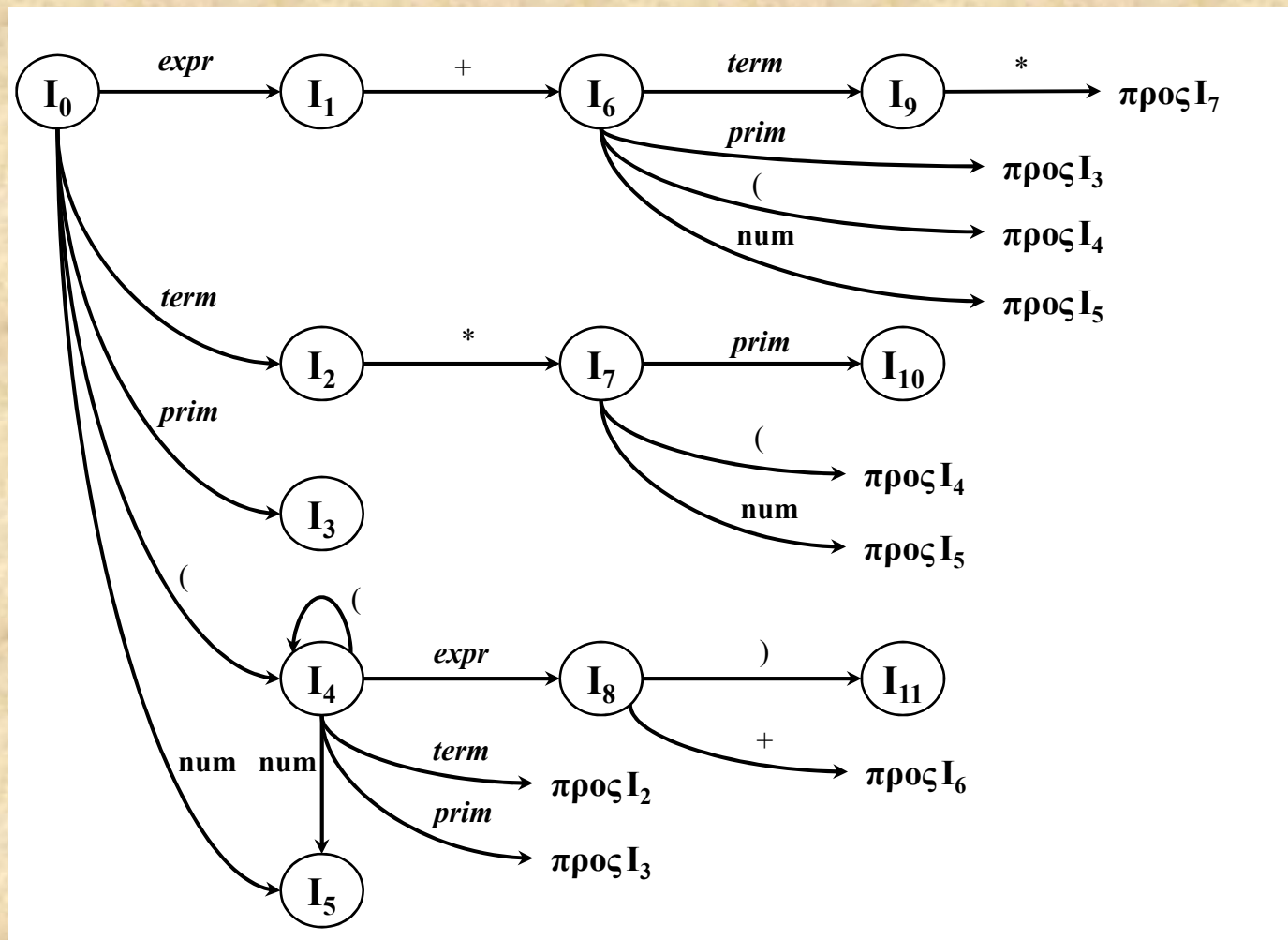
Παράδειγμα κατασκευής του κανονικοποιημένου LR(0) item set (δηλ. των καταστάσεων του DFA αναγνώρισης των εφικτών προθεμάτων).

$S$	$\rightarrow expr$
$expr$	$\rightarrow expr + term \mid term$
$term$	$\rightarrow term * prim \mid prim$
$prim$	$\rightarrow (expr) \mid num$



# Κατασκευή SLR parse table (13/16)

Το DFA για το προηγούμενο παράδειγμα





# Κατασκευή SLR parse table (14/16)

Κατασκευή του πίνακα ανάλυσης (*parse table*). Εφαρμόζουμε την παρακάτω διαδικασία δεδομένου του DFA που έχει ήδη παραχθεί, για κάθε ξεχωριστή κατάσταση  $s$ :

- Μετάβαση σε άλλη κατάσταση  $s'$  με τη χρήση ενός τερματικού συμβόλου  $a$  ισοδυναμεί με **shift** στην νέα κατάσταση. Δηλ.
  - $action[s, a] = shift\ s'$
- Μετάβαση σε άλλη κατάσταση  $s'$  με τη χρήση ενός μη τερματικού συμβόλου  $X$  ισοδυναμεί με **goto** στην νέα κατάσταση. Δηλ.
  - $action[s, X] = goto\ s'$  (αυτό συμβαίνει μόνο μετά από reduction)
- Εάν υπάρχει κάποιο item  $A \rightarrow a \bullet$  στην  $s$ , τότε κάνουμε αναγωγή – reduction με αυτή την παραγωγή  $k$  για κάθε τερματικό σύμβολο. Δηλ.
  - $action[s, a] = reduce\ k, \forall a$  (μετά το reduce ο parser θα κάνει  $action[s', A]$ , για  $s'$  στην κορυφή της στοίβας, που θα είναι πάντα goto)





## Κατασκευή SLR parse table (15/16)

Η παρούσα κατάσταση του parser. Ορίζεται στην πράξη από το current configuration το οποίο περιλαμβάνει:

- Την ακολουθία συμβόλων εισόδου, που τερματίζει με \$
- Την παρούσα κατάσταση  $s$  του DFA στην κορυφή της στοίβας
- Το υπόλοιπο τμήμα το οποίο είναι καταστάσεις κι σύμβολα.

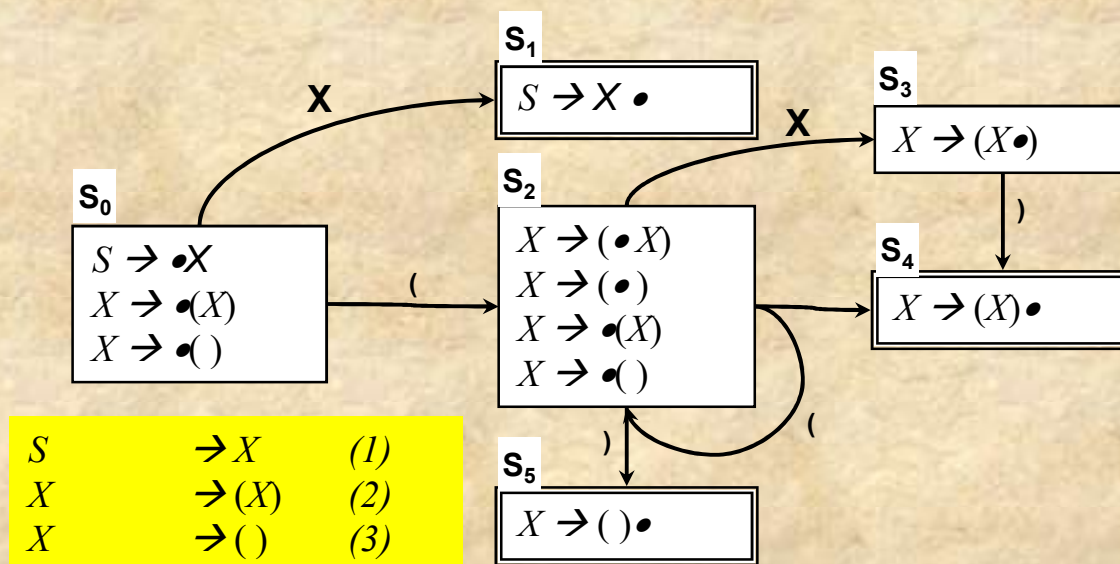
Για λόγους ευκολίας θα ξεχωρίσουμε τις καταστάσεις από τα σύμβολα σε δύο διαφορετικές στοίβες, εφαρμόζοντας όμως πάντα τις ίδιες ενέργειες. Έτσι έχουμε:

- Την ακολουθία των συμβόλων εισόδου, με \$ end-of-input.
- Τη στοίβα καταστάσεων με την παρούσα κατάσταση  $s$  στην κορυφή.
- Τη στοίβα γραμματικών συμβόλων (τερματικά και μη τερματικά, τα τελευταία από αναγωγές).



# Κατασκευή SLR parse table (16/16)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	



- Οι τελικές καταστάσεις είναι αυτές με items που έχουν  $\bullet$  μετά το RHS
- Μεταβάσεις για  $X$  μη τερματικά ορίζονται πάντα μέσω της *goto*
- Μεταβάσεις με tokens ορίζονται πάντα ως *shift*
- Σε items τελικών καταστάσεων, για κάθε token, κάνουμε *reduce*,
- ...εκτός και εάν έχουμε το αρχικό σύμβολο της γραμματικής στο LHS οπότε μόνο με \$ κάνουμε *accept* αλλιώς πάντοτε *error*.



# Περιεχόμενα

---

- Κατασκευή SLR parse table
- *Χρήση του SLR table*
- Οι βελτιώσεις του SLR parser





# Χρήση του SLR table (1/24)

Ανακεφαλαιώνουμε τις ενέργειες που εκτελεί ενός SLR parser:

- *shift s*
  - *push* το παρόν σύμβολο εισόδου στη στοίβα συμβόλων
  - *push* το *s* στη στοίβα καταστάσεων
  - προχώρησε στο επόμενο σύμβολο εισόδου
- *reduce (k)*
  - *pop* από τη στοίβα καταστάσεων και από τη στοίβα συμβόλων τόσα στοιχεία όσα και τα γραμματικά σύμβολα στο δεξί τμήμα (RHS) του γραμματικού κανόνα (*k*) *πρακτικά κάνουμε roll-back to αυτόματο!*
  - *push* το RHS του κανόνα (*k*) στη στοίβα συμβόλων
  - *push goto*[*top*(στοίβα καταστάσεων), *top*(στοίβα συμβόλων)] στη στοίβα καταστάσεων
- *accept*
  - Τερμάτισε την ανάλυση και ανέφερε επιτυχής περάτωση



## Χρήση του SLR table (2/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοιβα  
καταστάσεων


Στοιβα  
συμβόλων


Διαθέσιμη  
είσοδος

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)



## Χρήση του SLR table (3/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοίβα  
καταστάσεων

$S_0$

Στοίβα  
συμβόλων


Διαθέσιμη  
είσοδος

( ( ) ) \$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)





## Χρήση του SLR table (4/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοίβα  
καταστάσεων

$S_0$

Στοίβα  
συμβόλων


Διαθέσιμη  
είσοδος

( ( ) ) \$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)



## Χρήση του SLR table (5/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοίβα  
καταστάσεων

$S_2$
$S_0$

Στοίβα  
συμβόλων

(

Διαθέσιμη  
είσοδος

( ) ) \$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)



# Χρήση του SLR table (6/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοίβα  
καταστάσεων

$S_2$
$S_0$

Στοίβα  
συμβόλων

(

Διαθέσιμη  
είσοδος

( ) ) \$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)





# Χρήση του SLR table (7/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοίβα  
καταστάσεων

$S_2$
$S_2$
$S_0$

Στοίβα  
συμβόλων

(
(

Διαθέσιμη  
είσοδος

) ) \$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)



## Χρήση του SLR table (8/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοίβα  
καταστάσεων

$S_2$
$S_2$
$S_0$

Στοίβα  
συμβόλων

(
(

Διαθέσιμη  
είσοδος

) ) \$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)



# Χρήση του SLR table (9/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοίβα  
καταστάσεων

$S_5$
$S_2$
$S_2$
$S_0$

Στοίβα  
συμβόλων

)
(
(

Διαθέσιμη  
είσοδος

) \$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)





# Χρήση του SLR table (10/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοιβά  
καταστάσεων

$S_5$
$S_2$
$S_2$
$S_0$

Στοιβά  
συμβόλων

)
(
(

Διαθέσιμη  
είσοδος

) \$

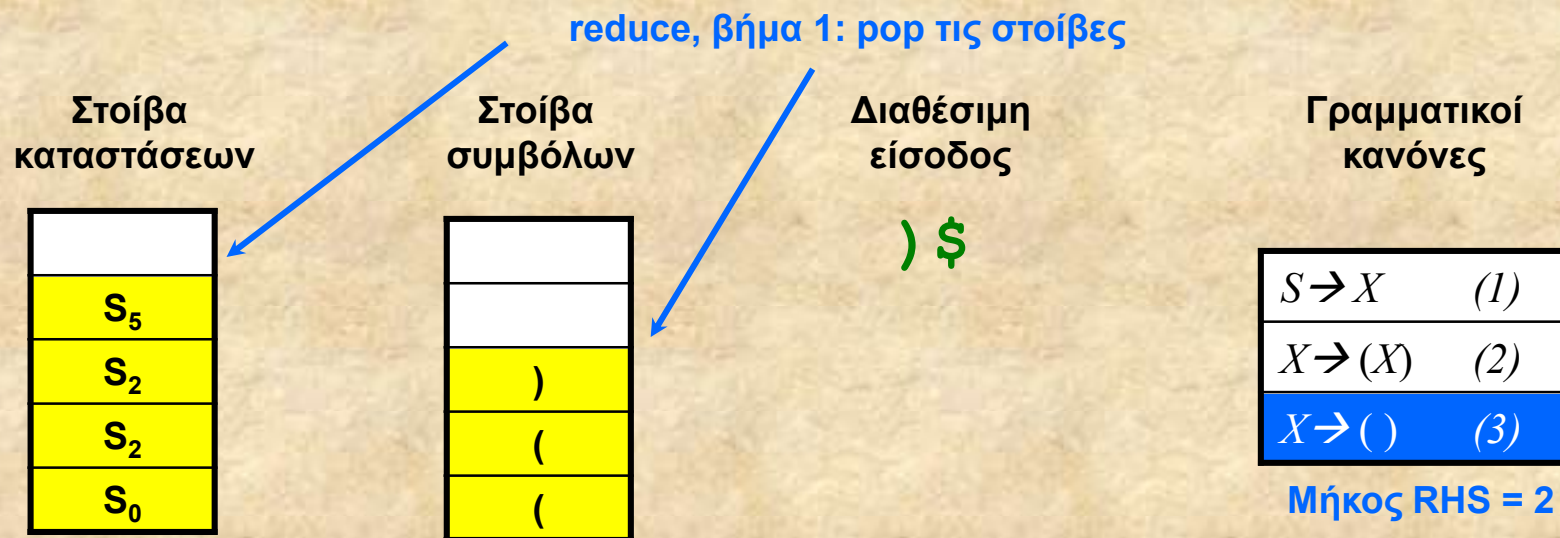
Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)



# Χρήση του SLR table (11/24)

	action			goto
State	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	





# Χρήση του SLR table (12/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

reduce, βήμα 1: pop τις στοίβες

Στοίβα  
καταστάσεων

$S_2$
$S_0$

Στοίβα  
συμβόλων

(

Διαθέσιμη  
είσοδος

) \$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)

Μήκος RHS = 2





# Χρήση του SLR table (13/24)

	action			goto
State	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

reduce, βήμα 2: push το μη τερματικό σύμβολο

Στοίβα  
καταστάσεων

$S_2$
$S_0$

Στοίβα  
συμβόλων

(

Διαθέσιμη  
είσοδος

) \$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)

Σύμβολο X



# Χρήση του SLR table (14/24)

	action			goto
State	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

reduce, βήμα 2: push το μη τερματικό σύμβολο

Στοίβα  
καταστάσεων

$S_2$
$S_0$

Στοίβα  
συμβόλων

$X$
(

Διαθέσιμη  
είσοδος

) \$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)

Σύμβολο  $X$



# Χρήση του SLR table (15/24)

	action			goto
State	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

reduce, βήμα 3: goto στο νέο state

Στοίβα  
καταστάσεων

$S_2$
$S_0$

Στοίβα  
συμβόλων

$X$
(

Διαθέσιμη  
είσοδος

) \$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)



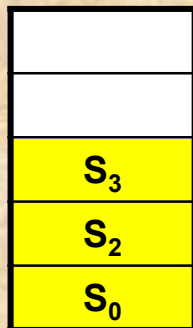


# Χρήση του SLR table (16/24)

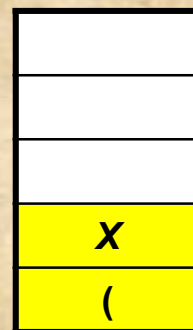
	action			goto
State	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

reduce, βήμα 3: goto στο νέο state

Στοίβα  
καταστάσεων



Στοίβα  
συμβόλων



Διαθέσιμη  
είσοδος

) \$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ($	(3)



# Χρήση του SLR table (17/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοίβα  
καταστάσεων

$S_3$
$S_2$
$S_0$

Στοίβα  
συμβόλων

X
(

Διαθέσιμη  
είσοδος

) \$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)



# Χρήση του SLR table (18/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοίβα  
καταστάσεων

$S_4$
$S_3$
$S_2$
$S_0$

Στοίβα  
συμβόλων

)
X
(

Διαθέσιμη  
είσοδος

\$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)





# Χρήση του SLR table (19/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοιβα  
καταστάσεων

$S_4$
$S_3$
$S_2$
$S_0$

Στοιβα  
συμβόλων

)
X
(

Διαθέσιμη  
είσοδος

\$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)



# Χρήση του SLR table (20/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

reduce, βήμα 1: pop τις στοίβες

Στοίβα  
καταστάσεων

$S_4$
$S_3$
$S_2$
$S_0$

Στοίβα  
συμβόλων

)
X
(

Διαθέσιμη  
είσοδος

\$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)



# Χρήση του SLR table (21/24)

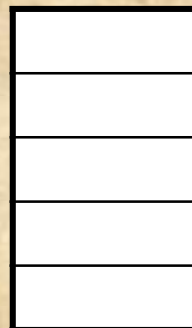
State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

reduce, βήμα 2: push το μη τερματικό σύμβολο

Στοίβα  
καταστάσεων



Στοίβα  
συμβόλων



Διαθέσιμη  
είσοδος

\$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)





# Χρήση του SLR table (22/24)

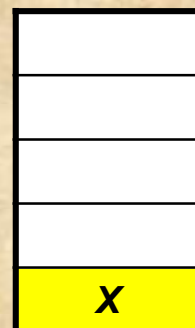
State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

reduce, βήμα 3: goto στο νέο state

Στοίβα  
καταστάσεων



Στοίβα  
συμβόλων



Διαθέσιμη  
είσοδος

\$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)



## Χρήση του SLR table (23/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοιβα  
καταστάσεων

$S_1$
$S_0$

Στοιβα  
συμβόλων

X

Διαθέσιμη  
είσοδος

\$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)



# Χρήση του SLR table (24/24)

State	action			goto
	(	)	\$	X
$S_0$	shift $S_2$	error	error	$S_1$
$S_1$	error	error	accept	
$S_2$	shift $S_2$	shift $S_5$	error	$S_3$
$S_3$	error	shift $S_4$	error	
$S_4$	reduce (2)	reduce (2)	reduce (2)	
$S_5$	reduce (3)	reduce (3)	reduce (3)	

Στοίβα  
καταστάσεων

$S_1$
$S_0$

Στοίβα  
συμβόλων

X

Διαθέσιμη  
είσοδος

\$

Γραμματικοί  
κανόνες

$S \rightarrow X$	(1)
$X \rightarrow (X)$	(2)
$X \rightarrow ()$	(3)





# Περιεχόμενα

---

- Κατασκευή SLR parse table
- Χρήση του SLR table
- *Οι βελτιώσεις του SLR parser*



## Οι βελτιώσεις του SLR parser (1/5)

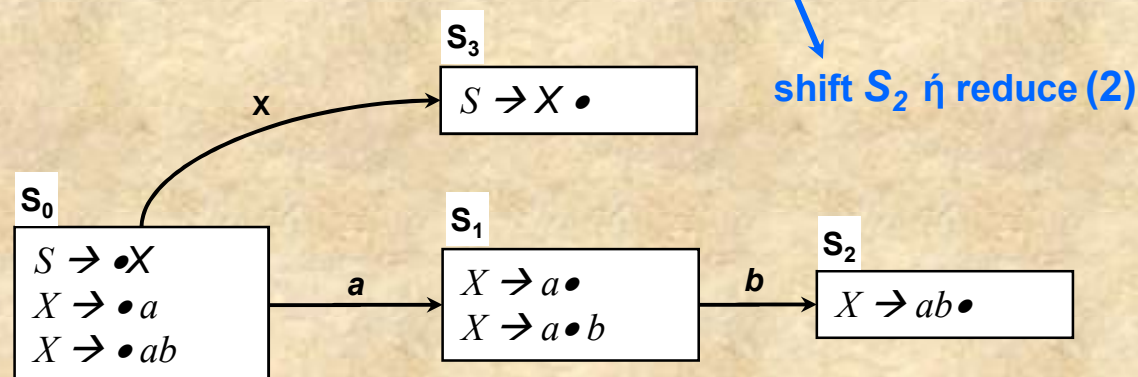
- *Τα κυριότερα προβλήματα του SLR*
  - Δεν χρησιμοποιεί look-ahead
  - Ευάλωτο σε ασυμφωνίες που μπορούν να αποφευχθούν
    - ◆ shift/reduce
    - ◆ reduce/reduce
  - Η λύση είναι να χρησιμοποιηθεί look-ahead
    - ◆ Αρκεί για τις αναγωγές
      - να κάνουμε reduce  $A \rightarrow \beta$  μόνο εάν το επόμενο σύμβολο είναι εφικτό να εμφανιστεί μετά το  $A$  σε κάποια παραγωγή
    - ◆ Συστηματική εφαρμογή, επηρεάζει το σύνολο των καταστάσεων, οι ενέργειες γίνονται τώρα πάντα συνάρτηση του επόμενου συμβόλου (LA)
    - ◆ Μπορεί η τεχνική να γενικευτεί και στις περιπτώσεις πολλαπλών LA, δηλ. ( $k$ )





## Οι βελτιώσεις του SLR parser (2/5)

State	action			goto
	a	b	\$	X
$S_0$	shift $S_1$	error	error	$S_3$
$S_1$	reduce(2)	S/R conflict	reduce(2)	
$S_2$	reduce(3)	reduce(3)	reduce(3)	
$S_3$	error	error	accept	



$S \rightarrow X$  (1)  
 $X \rightarrow a$  (2)  
 $X \rightarrow ab$  (3)





## Οι βελτιώσεις του SLR parser (3/5)

Κατασκευή του πίνακα ανάλυσης (*parse table*) για την υποστήριξη *look-ahead* σε αναγωγές. Εφαρμόζουμε την παρακάτω διαδικασία δεδομένου του DFA που έχει ήδη παραχθεί, για κάθε ξεχωριστή κατάσταση  $s$ :

- Μετάβαση σε άλλη κατάσταση  $s'$  με τη χρήση ενός τερματικού συμβόλου  $a$  ισοδυναμεί με **shift** στην νέα κατάσταση. Δηλ.
  - $action[s, a] = shift\ s'$  - όπως και πριν
- Μετάβαση σε άλλη κατάσταση  $s'$  με τη χρήση ενός μη τερματικού συμβόλου  $X$  ισοδυναμεί με **goto** στην νέα κατάσταση. Δηλ.
  - $action[s, X] = goto\ s'$  - όπως και πριν
- Εάν υπάρχει κάποιο item  $A \rightarrow \alpha \bullet$  στην  $s$ , τότε κάνουμε αναγωγή – reduction με αυτή την παραγωγή  $k$  **μόνο εάν το παρόν σύμβολο εισόδου  $T$  μπορεί να έπεται του  $A$  σε κάποια παραγωγή**. Δηλ.
  - $action[s, T] = reduce$  κ εάν  $\exists$  παραγωγή  $B \rightarrow \beta A T \gamma$

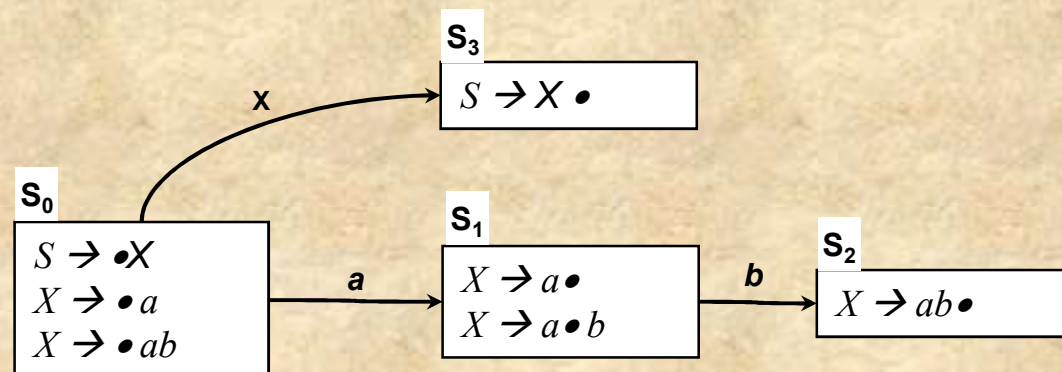
Δηλ.  $T \in FOLLOW(A)$



## Οι βελτιώσεις του SLR parser (4/5)

Ο νέος parse table με look-ahead για τις αναγωγές. Καθώς το  $b$  δεν έπεται του  $X$  σε καμία παραγωγή, η ενέργεια που επιλέγεται είναι το shift.

State	action			goto
	a	b	\$	
$S_0$	shift $S_1$	error	error	$S_3$
$S_1$	reduce(2)	shift $S_2$	reduce(2)	
$S_2$	reduce(3)	reduce(3)	reduce(3)	
$S_3$	error	error	accept	



$S \rightarrow X$  (1)  
 $X \rightarrow a$  (2)  
 $X \rightarrow ab$  (3)



## Οι βελτιώσεις του SLR parser (5/5)

- Συνήθως τα conflicts σε LR(1) parser οφείλονται σε διαφορούμενη εγγενώς γραμματική.
- Τα shift/reduce conflicts στην πράξη σημαίνουν την ύπαρξη καταστάσεων με items  $A \rightarrow \alpha \bullet$  και  $A \rightarrow \alpha \bullet \beta$  ταυτόχρονα. Όπως π.χ. και στην περίπτωση του if then else. Η επίλυση του conflict γίνεται με μεταβάσεις στον δεύτερο κανόνα  $\forall c \notin \text{FOLLOW}(A)$ .
- Αντίστοιχα, τα reduce/reduce ακόμη conflicts οφείλονται στην ύπαρξη κατάστασης η οποία περιέχει δύο items  $A \rightarrow \alpha$  και  $B \rightarrow \beta$  και επίσης να  $\exists c: c \in \text{FOLLOW}(A) \wedge c \in \text{FOLLOW}(B)$ . Τότε στην κατάσταση αυτή με είσοδο  $c$  δεν ξέρουμε τι να κάνουμε.