

Μεταγλωττιστές

ΕΡΓΑΣΙΑ 2019-2020

Αργυροπούλου Μαρία | ΑΜ: Π18011

Δημητρίου Δημήτρης | ΑΜ:

Στεργίου Χρήστος | ΑΜ: Π18147

27/1/2020

Άσκηση (Γ)

Για να αποδειχθεί εάν είναι η γραμματική LL(1), αρχικά θα υπολογισθούν τα σύνολα FIRST, FOLLOW, EMPTY και LOOKAHEAD και έπειτα θα γίνει παράθεση του συντακτικού πίνακα.

Αναλύοντας τους κανόνες παραγωγής της γραμματικής της εκφώνησης έχουμε τους παρακάτω κανόνες:

- 1) $S \rightarrow (X)$
- 2) $X \rightarrow YZ$
- 3) $Y \rightarrow \alpha$
- 4) $Y \rightarrow \beta$
- 5) $Y \rightarrow S$
- 6) $Z \rightarrow *X$
- 7) $Z \rightarrow -X$
- 8) $Z \rightarrow +X$
- 9) $Z \rightarrow \epsilon$

Σύνολα **FIRST**:

$$\text{FIRST}(S) = \{ (\}$$

- I. $\text{FIRST}(Y) - \{\epsilon\} \in \text{FIRST}(X)$
- II. $\text{FIRST}(Y) = \{\alpha, \beta, (\}$

$$\text{Από I και II} \Rightarrow \text{FIRST}(X) = \{\alpha, \beta, (\}$$

$$\text{FIRST}(Z) = \{ *, -, +, \epsilon \}$$

Σύνολα **FOLLOW**:

Σύμφωνα με τους κανόνες υπολογισμού του FOLLOW (που βρίσκονται στο βιβλίο ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ, Μ. Κ. ΒΙΡΒΟΥ, ΕΚΔΟΣΕΙΣ ΒΑΡΒΑΡΗΓΟΥ, ΚΕΦ. 5.2, ΣΕΛ. 177).

Από τον κανόνα 1) υπολογισμού του FOLLOW

$$\text{I. } \$ \in \text{FOLLOW}(S)$$

Από τον κανόνα 2) υπολογισμού του FOLLOW

- II. 1) $\Rightarrow \text{FIRST}() - \{\epsilon\} \subseteq \text{FOLLOW}(X) \Rightarrow \{) \} \subseteq \text{FOLLOW}(X)$
- III. 2) $\Rightarrow \text{FIRST}(Z) - \{\epsilon\} \subseteq \text{FOLLOW}(Y) \Rightarrow \{ *, -, + \} \subseteq \text{FOLLOW}(Y)$

Από τον κανόνα 3i) υπολογισμού του FOLLOW

$$\text{IV. } 2) \Rightarrow \text{FOLLOW}(X) \subseteq \text{FOLLOW}(Z)$$

- V. $5) \Rightarrow \text{FOLLOW}(Y) \subseteq \text{FOLLOW}(S)$
 VI. $6) + 7) + 8) \Rightarrow \text{FOLLOW}(Z) \subseteq \text{FOLLOW}(X)$

Από τον κανόνα 3ii) υπολογισμού του FOLLOW

- VII. $2) \Rightarrow \text{FOLLOW}(X) \subseteq \text{FOLLOW}(Y)$

Από τις παραπάνω σχέσεις προκύπτουν τα εξής:

$$\text{II} \Rightarrow \text{FOLLOW}(X) = \{ \) \}$$

$$\text{II} + \text{IV} + \text{VI} \Rightarrow \text{FOLLOW}(Z) = \{ \) \}$$

$$\text{III} + \text{IV} \Rightarrow \text{FOLLOW}(Y) = \{ *, -, +,) \}$$

$$\text{I} + \text{V} \Rightarrow \text{FOLLOW}(S) = \{ *, -, +,), \$ \}$$

Σύνολα **EMPTY**:

$$\text{EMPTY}(S) = \text{FALSE}$$

$$\text{EMPTY}(X) = \text{FALSE}$$

$$\text{EMPTY}(Y) = \text{FALSE}$$

$$\text{EMPTY}(Z) = \text{TRUE}$$

Σύνολα **LOOKAHEAD**:

$$\text{LOOKAHEAD}(S \rightarrow (X)) = \text{FIRST}(\) = \{ \) \}$$

$$\text{LOOKAHEAD}(X \rightarrow YZ) = \text{FIRST}(Y) = \{ \alpha, \beta, \) \}$$

$$\text{LOOKAHEAD}(Y \rightarrow \alpha) = \text{FIRST}(\alpha) = \{ \alpha \}$$

$$\text{LOOKAHEAD}(Y \rightarrow \beta) = \text{FIRST}(\beta) = \{ \beta \}$$

$$\text{LOOKAHEAD}(Y \rightarrow S) = \text{FIRST}(S) = \{ (\}$$

$$\text{LOOKAHEAD}(Z \rightarrow *X) = \text{FIRST}(*) = \{ * \}$$

$$\text{LOOKAHEAD}(Z \rightarrow -X) = \text{FIRST}(-) = \{ - \}$$

$$\text{LOOKAHEAD}(Z \rightarrow +X) = \text{FIRST}(+) = \{ + \}$$

$$\text{LOOKAHEAD}(Z \rightarrow \epsilon) = \text{FOLLOW}(Z) = \{ \) \}$$

Επομένως η γραμματική είναι LL(1) διότι:

$$\text{LOOKAHEAD}(Y \rightarrow \alpha) \cap \text{LOOKAHEAD}(Y \rightarrow \beta) \cap \text{LOOKAHEAD}(Y \rightarrow S) = \emptyset$$

$$\text{LOOKAHEAD}(Z \rightarrow *X) \cap \text{LOOKAHEAD}(Z \rightarrow -X) \cap \text{LOOKAHEAD}(Z \rightarrow +X) \cap \text{LOOKAHEAD}(Z \rightarrow \epsilon) = \emptyset$$

V/T	()	α	β	*	-	+	\$
S	$S \rightarrow (X)$							
X	$X \rightarrow YZ$		$X \rightarrow YZ$	$X \rightarrow YZ$				
Y	$Y \rightarrow S$		$Y \rightarrow \alpha$	$Y \rightarrow \beta$				
Z		$Z \rightarrow \epsilon$			$Z \rightarrow *X$	$Z \rightarrow -X$	$Z \rightarrow +X$	

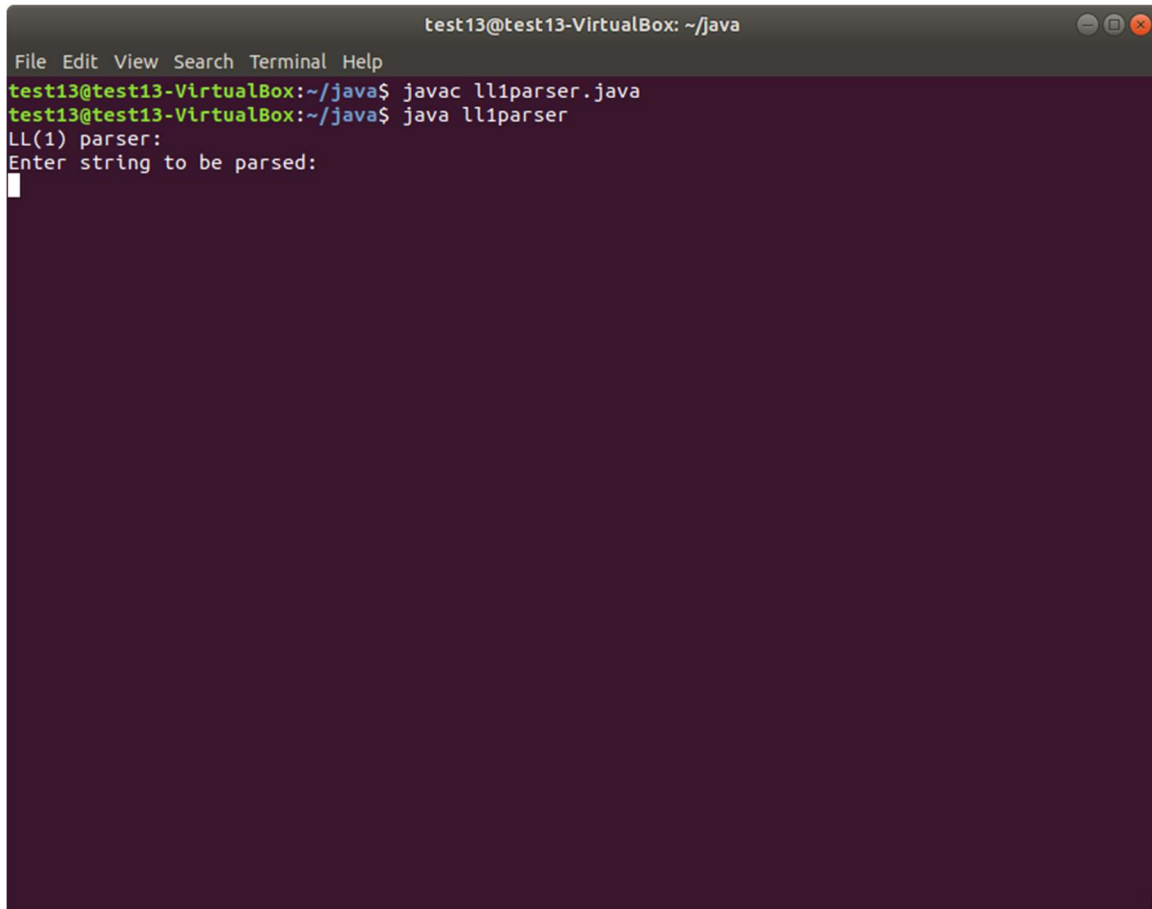
ΛΕΙΤΟΥΡΓΙΑ ΠΡΟΓΡΑΜΜΑΤΟΣ

- Η υλοποίηση του προγράμματος έχει γίνει σε γλώσσα JAVA.
- Έχει αναπτυχθεί με βάση τους κανόνες παραγωγής της συγκεκριμένης εκφώνησης.
- Το πρόγραμμα ονομάζεται ll1parser.java.
- Για το compilation χρησιμοποιείται η εντολή “javac ll1parser.java” καθώς και για την εκτέλεση η εντολή “java ll1parser”. **Σημείωση:** Η ανάπτυξη του προγράμματος προϋποθέτει την ικανότητα αναγνώρισης και εμφάνισης των ελληνικών χαρακτήρων ‘α’ και ‘β’ στο command line.

Το πρόγραμμα κατά την εκτέλεσή του παραπέμπει τον χρήστη να εισάγει ένα σύνολο συμβόλων το οποίο επιθυμεί να ελέγξει εάν αναγνωρίζεται από την συγκεκριμένη γλώσσα. Η είσοδος αποθηκεύεται σε μία μεταβλητή input τύπου String. Έπειτα γίνεται η αρχικοποίηση των πινάκων. Ο δισδιάστατος πίνακας ptable είναι ο συντακτικός πίνακας του προγράμματος με κάθε γραμμή να αντιστοιχεί σε ένα μη-τερματικό σύμβολο και με κάθε στήλη σε ένα τερματικό. Όπου δεν αντιστοιχίζεται κανόνας μεταξύ μη-τερματικού και τερματικού συμβόλου υπάρχει null. Στη συνέχεια γίνεται η εκτύπωση των πινάκων με τα σύμβολα της γραμματικής. Ο αλγόριθμος που χρησιμοποιείται βρίσκεται στη σελίδα 202 του κεφαλαίου 5.2.3 του βιβλίου ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ, Μ. Κ. ΒΙΡΒΟΥ, ΕΚΔΟΣΕΙΣ ΒΑΡΒΑΡΗΓΟΥ. Υπάρχει υλοποίηση της στοίβας που χρησιμοποιεί τις λειτουργίες: stack.peek() για την ανάγνωση της κορυφής της στοίβας, stack.pop() για την αφαίρεση του στοιχείου που βρίσκεται στην κορυφή της και της stack.push() για την εισαγωγή στοιχείου. Η αρχικοποίηση της στοίβας γίνεται με την είσοδο του “\$” και του αρχικού συμβόλου S. Σε κάθε επανάληψη εμφανίζονται στην οθόνη ο αριθμός της επανάληψης, ποιο σύμβολο αφαιρείται (εάν αυτό απαιτείται από το βήμα του αλγορίθμου), ο κανόνας που χρησιμοποιείται, το σύνολο των συμβόλων που εισέρχεται στη στοίβα μετά την αντικατάσταση με βάση τους κανόνες, η κορυφή της στοίβας καθώς και το σύμβολο το οποίο βρίσκεται στο input προς πιθανή αντικατάσταση. Σε περίπτωση σφάλματος

εμφανίζεται το αντίστοιχο μήνυμα λάθους. Εάν το input αναγνωριστεί τότε στην οθόνη εμφανίζονται επιτυχώς τα βήματα που έγιναν και οι pointers της στοίβας και του input ταυτίζονται περιέχοντας και οι δύο το “\$”. Στην περίπτωση που το input δεν αναγνωριστεί ο χρήστης θα λάβει στην οθόνη του το μήνυμα “ Input not recognised by grammar!!!”.

ΕΝΔΕΙΚΤΙΚΑ SCREENSHOTS



```
test13@test13-VirtualBox: ~/java
File Edit View Search Terminal Help
test13@test13-VirtualBox:~/java$ javac ll1parser.java
test13@test13-VirtualBox:~/java$ java ll1parser
LL(1) parser:
Enter string to be parsed:

```

Εικόνα 1. Compilation και prompt για input.

```

test13@test13-VirtualBox: ~/java
File Edit View Search Terminal Help
LL(1) parser:
Enter string to be parsed:
((a-β)*(β+a))
String to be parsed is:((a-β)*(β+a))$
Displaying terminal characters: ( ) α β * - + $
Displaying non-terminal characters: S X Y Z
Displaying parsing table:
-----
| (X) | null | null | null | null | null | null |
-----
| YZ | null | YZ | YZ | null | null | null |
-----
| S | null | α | β | null | null | null |
-----
| null | | null | null | *X | -X | +X |
-----
Stack pointer: S
Input pointer: (
-----
Repetition number: 1
Rule is: S->(X)
Revprod is: )
Revprod is: )X
Revprod is: )X(

```

Εικόνα 2. Εμφάνιση των πινάκων.

```

test13@test13-VirtualBox: ~/java
File Edit View Search Terminal Help
-----
Repetition number: 11
- is popped out of the stack
-----
Repetition number: 12
Rule is: X->YZ
Revprod is: Z
Revprod is: ZY
top on stack: Y
Input pointer: β
-----
Repetition number: 13
Rule is: Y->β
Revprod is: β
top on stack: β
Input pointer: β
-----
Repetition number: 14
β is popped out of the stack
-----
Repetition number: 15
Rule is: Z->
top on stack: )
Input pointer: )

```

Εικόνα 3. Εμφάνιση ενδεικτικών βημάτων.

```
File Edit View Search Terminal Help
-----
Repetition number: 30
a is popped out of the stack
-----
Repetition number: 31
Rule is: Z->
top on stack: )
Input pointer: )
-----
Repetition number: 32
) is popped out of the stack
-----
Repetition number: 33
Rule is: Z->
top on stack: )
Input pointer: )
-----
Repetition number: 34
) is popped out of the stack
-----
top on stack: $
Input pointer: $
LL1 parser shutting down ...
test13@test13-VirtualBox:~/java$
```

Εικόνα 4. Παράδειγμα τερματισμού επιτυχημένης αναγνώρισης.

```
File Edit View Search Terminal Help
LL(1) parser:
Enter string to be parsed:
(a+β()-a())$
String to be parsed is:(a+β()-a())$
Displaying terminal characters: ( ) a β * - + $
Displaying non-terminal characters: S X Y Z
Displaying parsing table:
-----
| (X) | null | null | null | null | null | null |
-----
| YZ | null | YZ | YZ | null | null | null |
-----
| S | null | a | β | null | null | null |
-----
| null | | null | null | *X | -X | +X |
-----
Stack pointer: S
Input pointer: (
-----
Repetition number: 1
Rule is: S->(X)
Revprod is: )
Revprod is: )X
Revprod is: )X(
```

Εικόνα 5. Παράδειγμα λειτουργίας με μη αναγνωρίσιμη είσοδο.(1)

```
File Edit View Search Terminal Help
+ is popped out of the stack
-----
Repetition number: 8
Rule is: X->YZ
Revprod is: Z
Revprod is: ZY
top on stack: Y
Input pointer: β
-----
Repetition number: 9
Rule is: Y->β
Revprod is: β
top on stack: β
Input pointer: β
-----
Repetition number: 10
β is popped out of the stack
-----
Repetition number: 11
Input not recognised by grammar!!!
top on stack: Z
Input pointer: (
LL1 parser shutting down ...
test13@test13-VirtualBox:~/java$
```

Εικόνα 6. Παράδειγμα λειτουργίας με μη αναγνωρίσιμη είσοδο.(2)

```
File Edit View Search Terminal Help
Enter string to be parsed:
@)β+)α((-β)
String to be parsed is:@)β+)α((-β)$
Displaying terminal characters: ( ) α β * - + $
Displaying non-terminal characters: S X Y Z
Displaying parsing table:
-----
| (X) | null | null | null | null | null | null |
-----
| YZ | null | YZ | YZ | null | null | null |
-----
| S | null | α | β | null | null | null |
-----
| null | | null | null | *X | -X | +X |
-----
Stack pointer: S
Input pointer: @
-----
Repetition number: 1
No terminal character
top on stack: S
Input pointer: @
LL1 parser shutting down ...
test13@test13-VirtualBox:~/java$
```

Εικόνα 7. Παράδειγμα διακοπής λειτουργίας λόγω μη αποδεκτού χαρακτήρα σύμφωνα με το αλφάβητο της γλώσσας.