

# Μεταγλωττιστές 2024-25

Τελεστές αριστερής προσηταιριστικότητας  
και  
συντακτική ανάλυση LL(1)

# Πού βρισκόμαστε τώρα

- Έχουμε κατασκευάσει έναν **συντακτικό αναλυτή LL(1)** με τη μέθοδο της «αναδρομικής κατάβασης» και έναν **διερμηνευτή** (interpreter)
- Για την **ανάλυση** της γραμματικής και την **εκτέλεση** της «γλώσσας υπολογισμού αριθμητικών εκφράσεων»

```
a = 2 + 7.55*44
print a
b = 3*(a-99.01)
print b*0.23
c = 5-3-2
print c
```

**Πρόβλημα!**  
Η έκφραση 5-3-2 δεν υπολογίζεται σωστά

# Το πρόβλημα της προσεταιριστικότητας (associativity) των τελεστών

- Η κλασσική υλοποίηση αναδρομικής κατάβασης **δεν χειρίζεται σωστά** εκφράσεις όπως **5 - 3 - 2** ή **20 / 4 / 2**
  - Οι τελεστές - και / είναι αριστερά προσεταιριστικοί (left associative)
  - Οι πιο πάνω εκφράσεις **πρέπει να υπολογιστούν ως (5 - 3) - 2 και (20 / 4) / 2**
- Αντιθέτως, ο συντακτικός αναλυτής τις χειρίζεται ως **5 - (3 - 2)** και **20 / (4 / 2)**
  - λόγω της μεθόδου ανάλυσης (recursive descent LL(1) με δεξιές αναδρομές)

5 - 3 - 2

Expr → Term Term\_tail

5 - 3 - 2

Term\_tail → Addop Term Term\_tail | ε

- 3 - 2

Term\_tail → Addop Term Term\_tail | ε

- 2

Term\_tail → Addop Term Term\_tail | ε



5 - 3 - 2

Expr → Term Term\_tail

5 - 3 - 2

Term\_tail → Addop Term Term\_tail | ε

- 3 - 2

Term\_tail → Addop Term Term\_tail | ε

- 2

Term\_tail → Addop Term Term\_tail | ε



Επιστρέφει  
None

5 - 3 - 2

Expr → Term Term\_tail

5 - 3 - 2

Term\_tail → Addop Term Term\_tail | ε

- 3 - 2

Term\_tail → Addop Term Term\_tail | ε

- 2

Επιστρέφει  
-, 2

Term\_tail → Addop Term Term\_tail | ε

■

Επιστρέφει  
None

5 - 3 - 2

Expr → Term Term\_tail

5 - 3 - 2

Term\_tail → Addop Term Term\_tail | ε

- 3 - 2

Επιστρέφει  
-, 3-2 (=1)

Term\_tail → Addop Term Term\_tail | ε

- 2

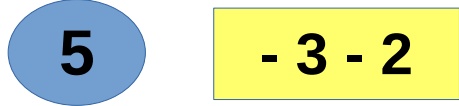



Επιστρέφει  
-, 2

Term\_tail → Addop Term Term\_tail | ε

■

Επιστρέφει  
None

5 - 3 - 2

Expr	→ Term Term_tail		Επιστρέφει 5-1 (=4)
			
Term_tail	→ Addop Term Term_tail   ε		Επιστρέφει -, 3-2 (=1)
			
Term_tail	→ Addop Term Term_tail   ε		Επιστρέφει -, 2
			
Term_tail	→ Addop Term Term_tail   ε		Επιστρέφει None



# Διόρθωση σειράς υπολογισμού

- **Πλεονέκτημα** της μεθόδου «αναδρομικής κατάβασης»
  - Μεγάλη ευελιξία για τροποποιήσεις του κώδικα
    - Για να εξυπηρετούνται οι ανάγκες μας κατά περίπτωση
- Μπορεί να αποτελέσει και **μειονέκτημα** όμως...
  - Αποκλίσεις από τη «στάνταρντ» μέθοδο δυσκολεύουν την **τεκμηρίωση**
    - Άρα και τη **συντήρηση** του κώδικα

# Τροποποίηση της μεθόδου ανάλυσης

```
Stmt_list  → Stmt Stmt_list | ε
Stmt       → id = Expr | print Expr
Expr       → Term (Addop Term)*
Term_tail → Addop Term Term_tail | ε
Term       → Factor (Multop Factor)*
Factor_tail → Multop Factor Factor_tail | ε
Factor     → (Expr) | id | number
Addop      → + | -
Multop     → * | /
```

- Όσο (*while*) υπάρχουν τελεστές ίδιου επιπέδου προτεραιότητας, συνεχίζουμε την κατανάλωση εισόδου → επιτυγχάνεται επεξεργασία από αριστερά προς τα δεξιά
  - Επεκταμένη σύνταξη γραμματικής,  $(...)^*$  = «μηδέν ή περισσότερες φορές»

# Ο νέος κώδικας για το Expr()

```
def Expr(self):
    if self.next_token in ('(', 'id', 'number'):
        # Expr → Term (Addop Term)*
        self.Term()
        while self.next_token in ('+', '-'):
            self.Addop()
            self.Term()
    else:
        raise ParseError
```

Αντίστοιχα για το Term(), με \*,/

- Ο τροποποιημένος κώδικας μπορεί να εξυπηρετήσει τελεστές με **αριστερή μόνο** (-, /) ή με **αριστερή/δεξιά** προσηταιριστικότητα (+, \*)
- Αν υπάρχουν τελεστές με **δεξιά μόνο** προσηταιριστικότητα (π.χ. η ύψωση σε δύναμη) θα πρέπει να χρησιμοποιηθεί **ο αρχικός κώδικας**

# Για εξάσκηση

- Ζητούμενο: να προστεθούν οι τελεστές:
  - Ύψωση σε δύναμη, συμβολίζεται με \*\*
  - Μοναδιαίο + και -
  - Modulo, συμβολίζεται με %
- Οι προτεραιότητες και προσηταιριστικότητες θα είναι ως εξής:

Τελεστής (υψηλότερη προτεραιότητα πρώτα)	Προσηταιριστικότητα
** (ύψωση σε δύναμη)	δεξιά
μοναδιαία + και -	
* / %	αριστερή
+ -	αριστερή

# Βήματα υλοποίησης

- Βρείτε τα νέα tokens και προσθέστε τα σχετικά patterns στον Tokenizer
- Προσθέστε στη γραμματική τους νέους κανόνες
- Βρείτε τα νέα first/follow sets, π.χ. με τη βοήθεια του εργαλείου
  - <https://smlweb.cpsc.ucalgary.ca/start.html>
- Προσθέστε τις συναρτήσεις των νέων μη τερματικών συμβόλων
  - Ή τροποποιήστε τις συναρτήσεις που ήδη υπάρχουν
- Δοκιμάστε με τη νέα είσοδο (source):

```
print 35 % 6 % 2
print -(+27.32/-3.5/---8.0)
print -4**3**-2
```