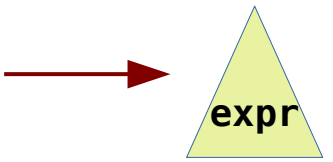
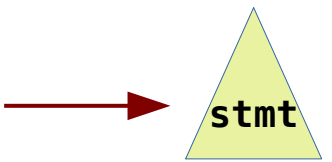
Μεταγλωττιστές 2023-24

Συντακτική ανάλυση και δημιουργία AST

Ο στόχος

- Δημιουργία AST από τη γραμματική των αριθμητικών εκφράσεων & εντολών ανάθεσης/εκτύπωσης
  - Μέρος του AST αναπαριστά την αριθμητική έκφραση
  - Το υπόλοιπο AST αναπαριστά την ακολουθία εντολών (ανάθεση/εκτύπωση), με τη σειρά που δίνονται στον πηγαίο κώδικα





Η κλάση ASTNode

- θα τη βρείτε στο module compilerlabs
- Βοηθητική κλάση για τη δημιουργία AST

```
from compilerlabs import ASTNode
```

- Παρέχει μεθόδους για την κατασκευή κόμβων ενός δένδρου AST (AST nodes)
  - Με τα επιλεγμένα χαρακτηριστικά (atrributes)
  - Και προσδιορίζει τη σύνδεση κάθε κόμβου με τα επιθυμητά υποδένδρα

Παράδειγμα

- Κατασκευή 2 «φύλλων» του AST (a και b)

```
a = ASTNode(attributes={'name':'a'})
b = ASTNode(attributes={'name':'b'})
```

- Κατασκευή κόμβου c και διασύνδεση των a και b ως «παιδιά» του

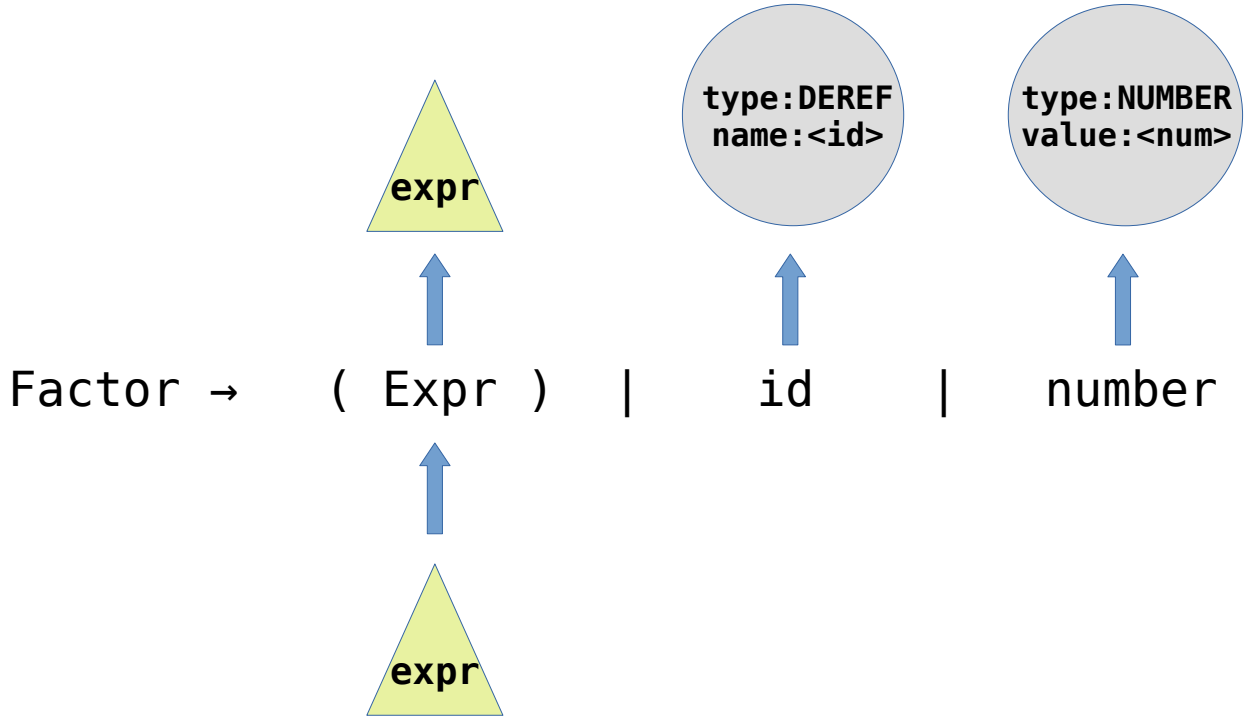
```
c = ASTNode(subnodes=[a,b],attributes={'name':'c'})
```

- Εμφάνιση δένδρου που έχει το c ως ρίζα (για debugging)

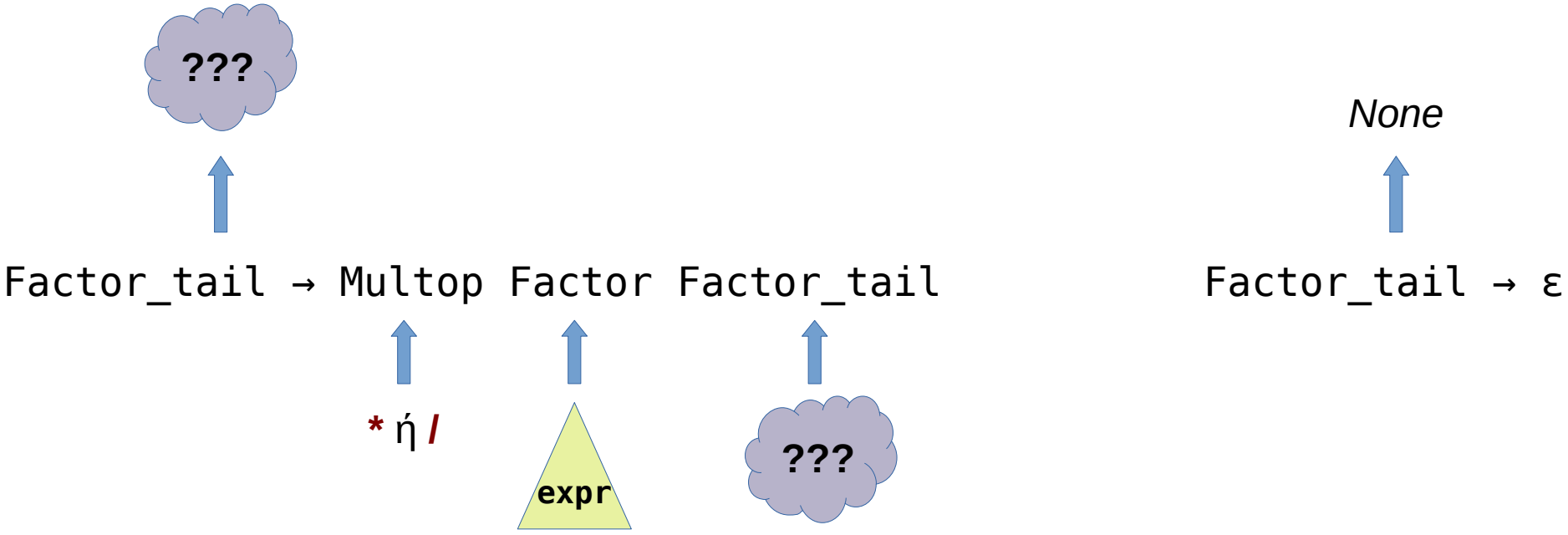
```
print(c)
ASTNode{'name': 'c'}
├── ASTNode{'name': 'a'}
└── ASTNode{'name': 'b'}
```

AST αριθμητικής έκφρασης

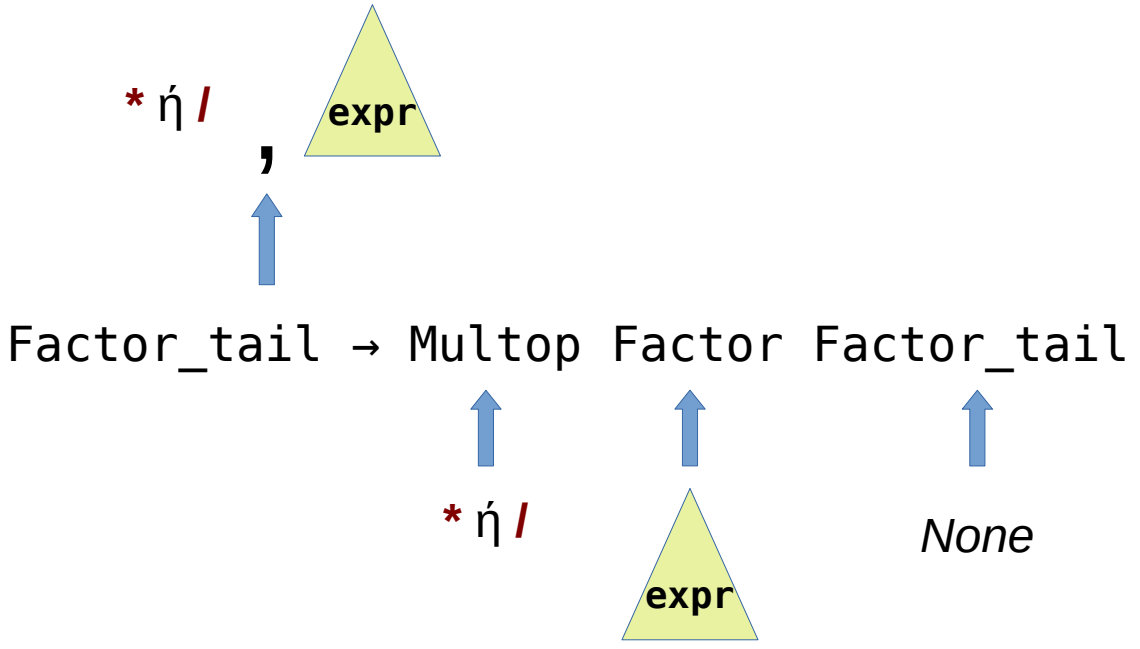
- Τα φύλλα είναι οι τιμές (σταθερές και μεταβλητές)
- Οι ενδιάμεσοι κόμβοι είναι αριθμητικοί τελεστές που συνδυάζουν τις τιμές του αριστερού και δεξιού υποδένδρου
- Κάθε μέθοδος που υλοποιεί κανόνες μη τερματικού συμβόλου της γραμματικής
  - Δέχεται μέρος του AST από τις υπο-κλήσεις μεθόδων των μη τερματικών συμβόλων που περιέχει
  - Συνδυάζει τα υποδένδρα AST και επιστρέφει όποιο νέο AST που προκύπτει
  - Τα Addop και Multop απλά επιστρέφουν τον τελεστή



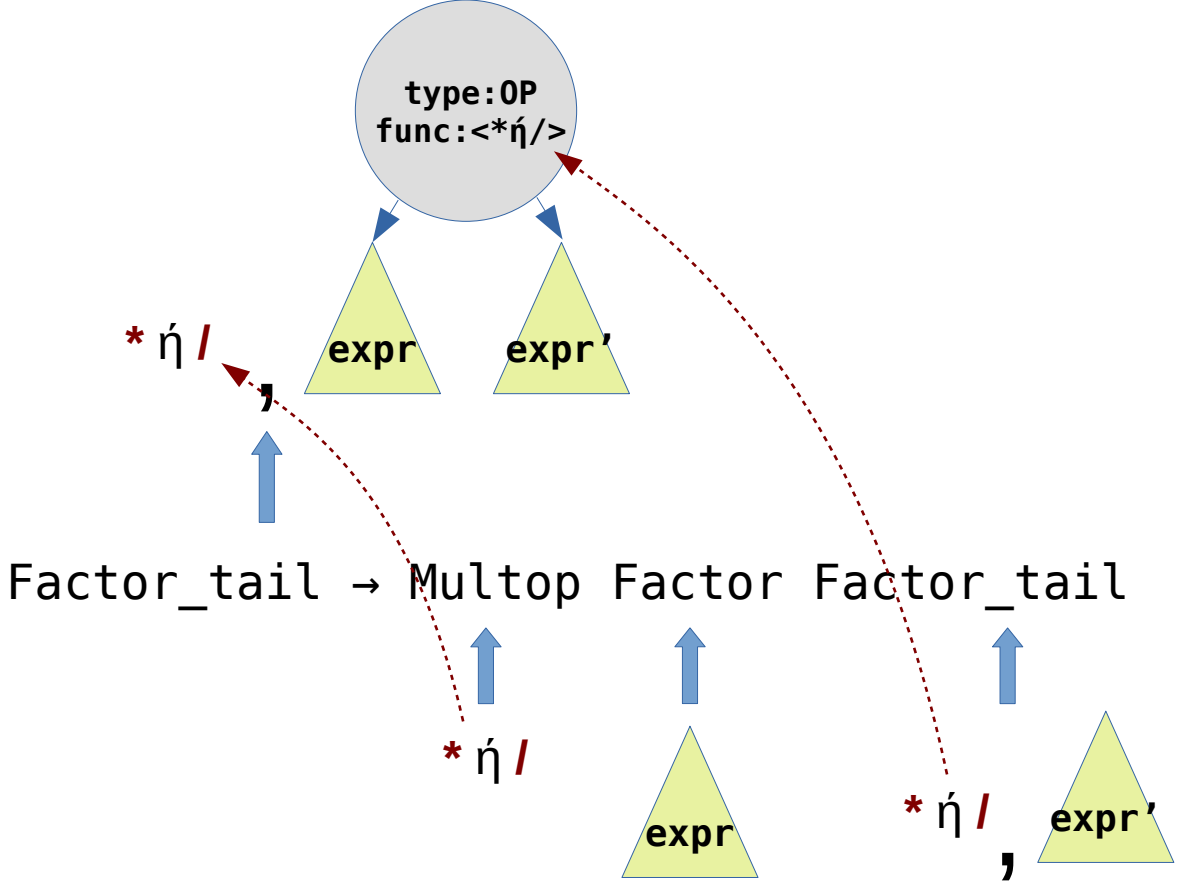
- Η μέθοδος του `Factor` κατασκευάζει τα φύλλα του AST
  - Μεταβλητές και σταθερούς αριθμούς float
  - Ή προωθεί ένα ήδη κατασκευασμένο υπο-AST από τη συνάρτηση του `Expr` στην περίπτωση του `( Expr )`



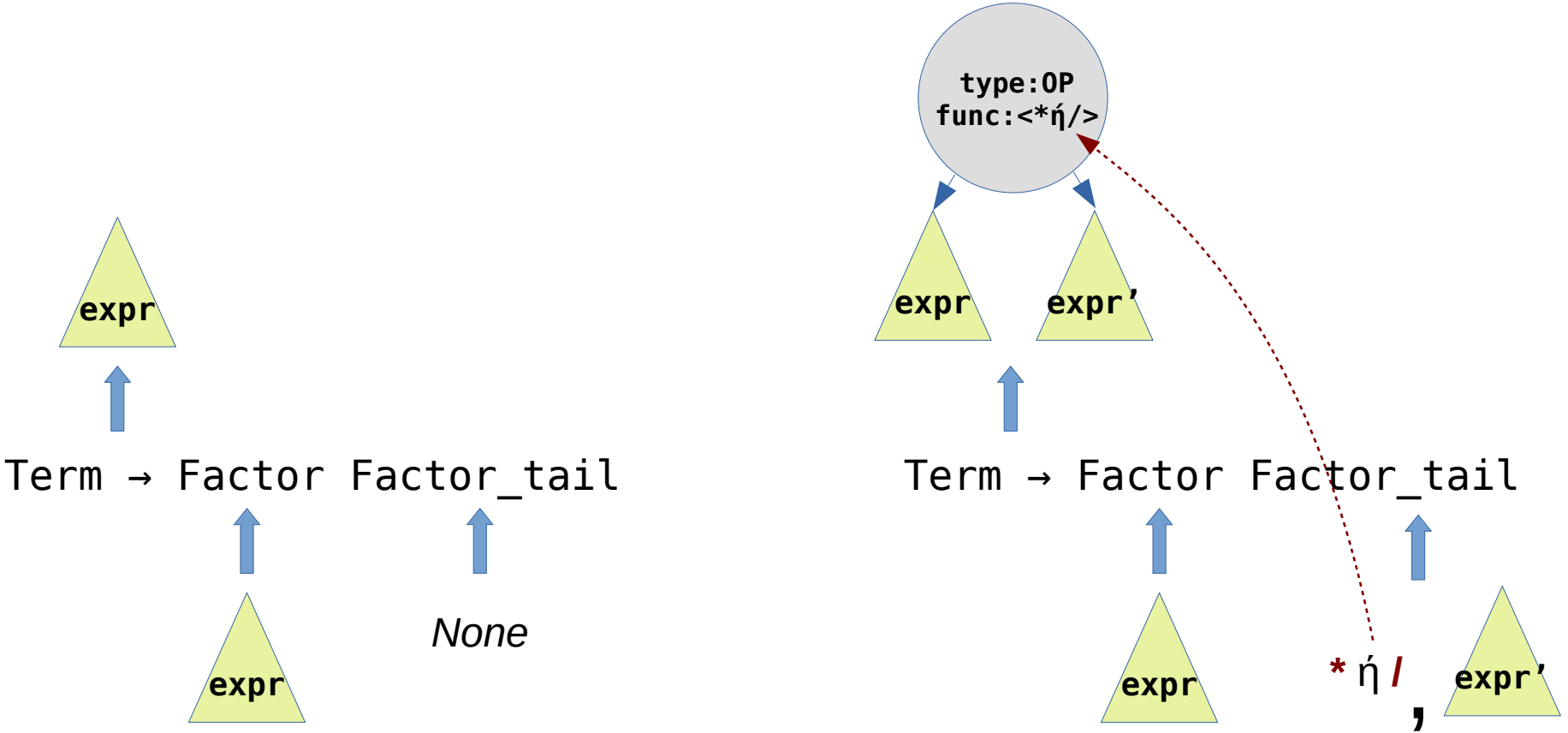
- Η `Factor_tail` επιστρέφει `None` στην περίπτωση της κενής παραγωγής
- Τι επιστρέφει όμως στην αντίθετη περίπτωση;
  - Εξαρτάται από το τι επιστρέφει η αναδρομική κλήση της `Factor_tail`



- Περίπτωση 1: Η αναδρομική κλήση της `Factor_tail` επιστρέφει `None`
- Η πληροφορία που υπάρχει είναι ο τελεστής (* ή /) και το δεξιό μέρος της έκφρασης που εφαρμόζεται στον τελεστή
  - Λείπει το αριστερό μέρος
- Η Factor_tail δεν μπορεί να κατασκευάσει άλλο μέρος του AST
  - Απλά προωθεί την πληροφορία (τελεστής,δεξιό μέρος) στο επόμενο στάδιο



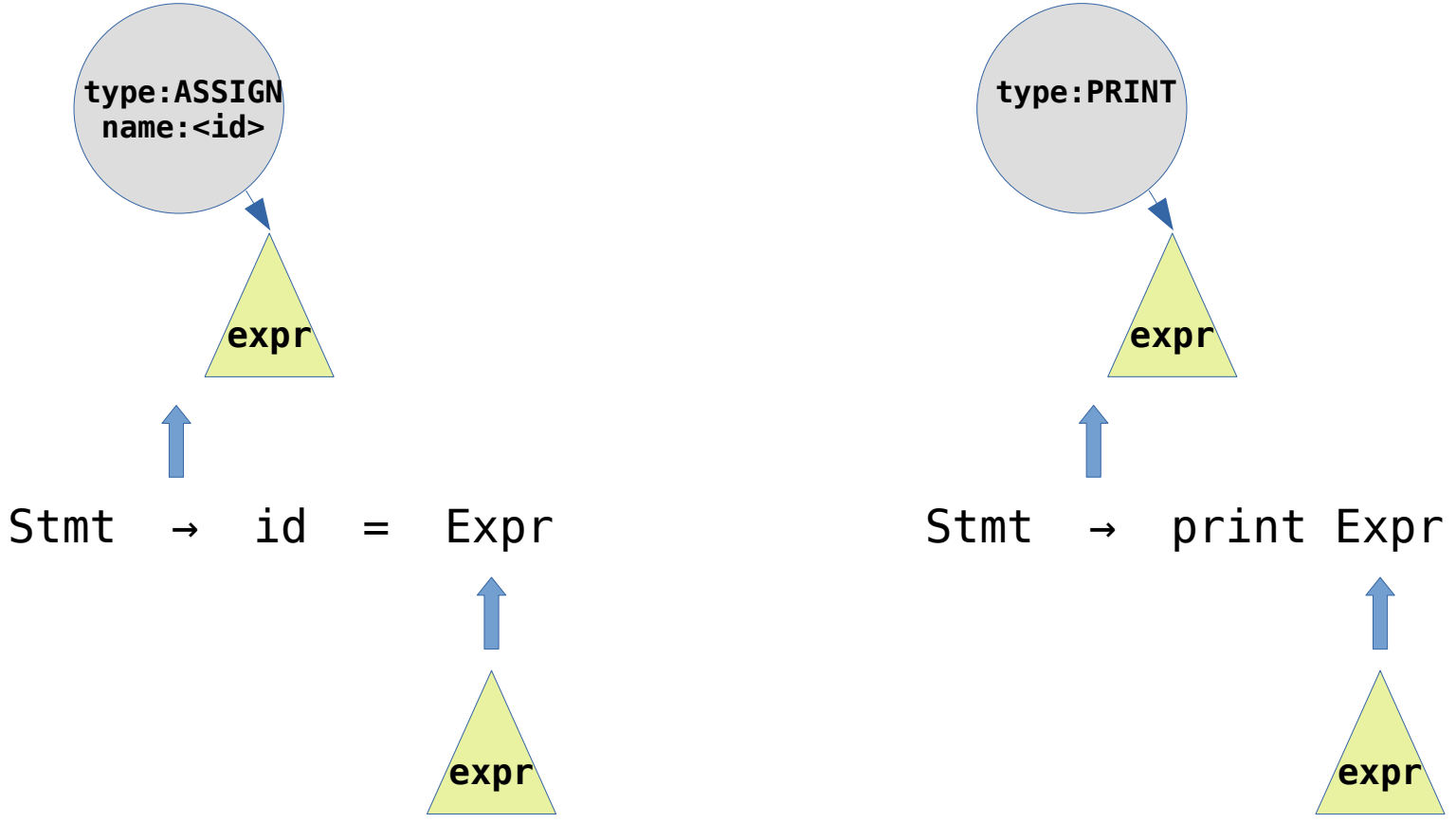
- Περίπτωση 2: Η αναδρομική κλήση της `Factor_tail` επιστρέφει το ζευγάρι (τελεστής,δεξιό μέρος)
  - Συνδυάζεται σε νέο AST με το αριστερό μέρος που επιστρέφει η `Factor`
  - Το νέο AST επιστρέφεται ως δεξιό μέρος μαζί με τον τελεστή της `Multop`



- Η `Term`, ανάλογα με το τι επιστρέφει η κλήση της `Factor_tail`
  - Είτε προωθεί το AST που επιστρέφει η `Factor`
  - Είτε το συνδυάζει με το ζευγάρι (τελεστής,δεξιό μέρος) της `Factor_tail`

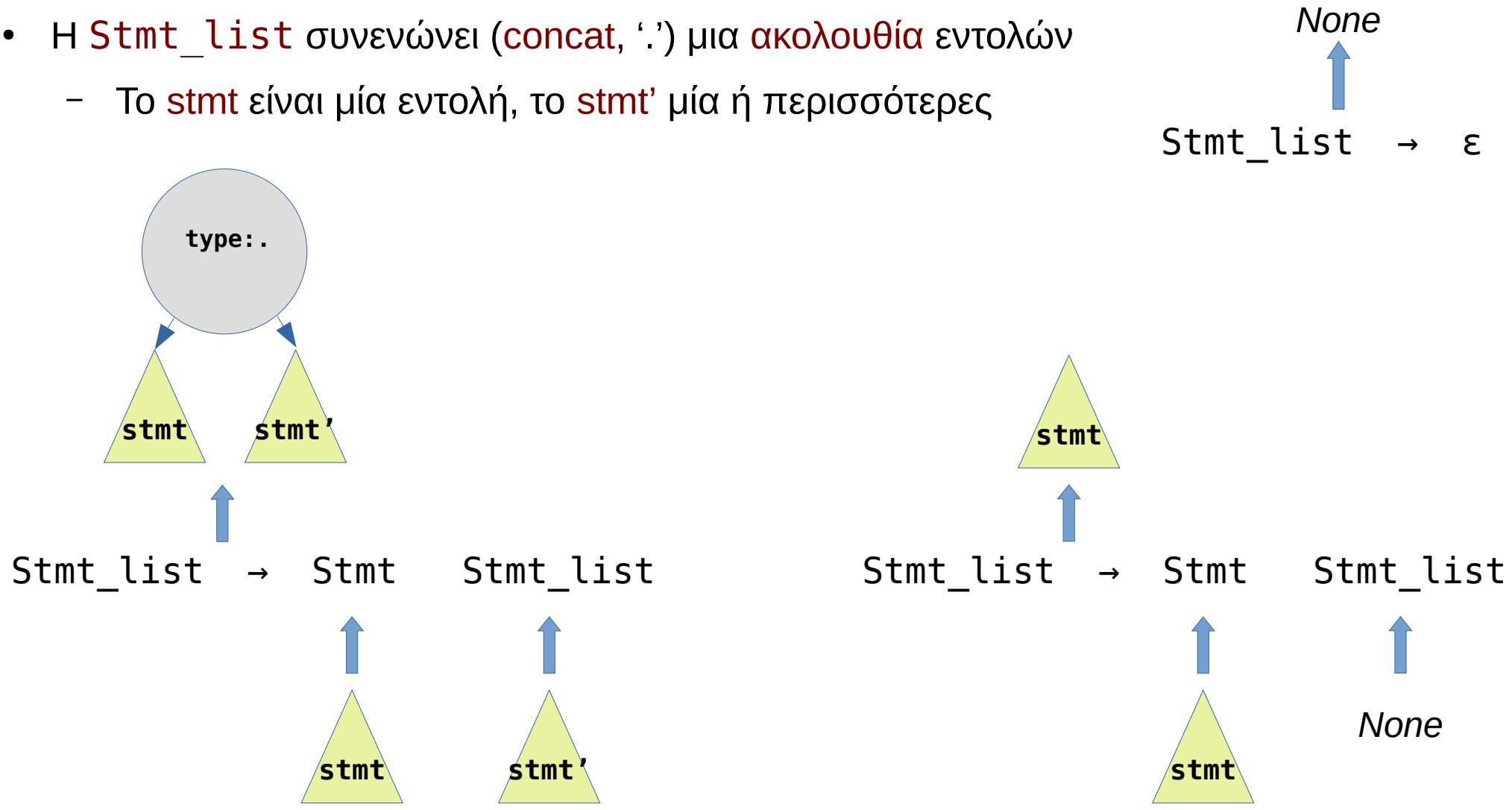
```
Expr        → Term Term_tail

Term_tail   → Addop Term Term_tail   |   ε
```

- Η υλοποίηση των `Expr` και `Term_tail` είναι ακριβώς ίδια με την περίπτωση των `Term` και `Factor_tail`
  - Εδώ όμως ο τελεστής είναι `+` ή `-`



- Η `Stmt` επιστρέφει κόμβους του AST που αντιστοιχούν στις εντολές (statements) της γλώσσας (ανάθεση ή εκτύπωση)

- Η Stmt_list συνενώνει (concat, ‘.’) μια ακολουθία εντολών
  - Το stmt είναι μία εντολή, το stmt’ μία ή περισσότερες



Στο τέλος της συντακτικής ανάλυσης

- Η μέθοδος parse() επιστρέφει το πλήρες δέντρο AST που προέκυψε κατά την συντακτική ανάλυση
- Το AST θα χρησιμοποιηθεί σε επόμενο βήμα του μεταγλωττιστή
  - Ή θα εκτελεστεί απευθείας σε έναν διερμηνευτή (interpreter)
  - Ή θα χρησιμοποιηθεί σε οποιονδήποτε άλλον μετασχηματισμό του κώδικα εισόδου