Abstract Syntax Trees (AST)

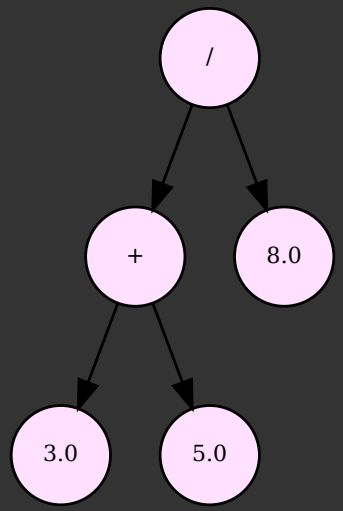
(Οπτικοποίηση μέσω Python)

Abstract Syntax Tree (AST)

- Αφαιρετικό δένδρο συντακτικής ανάλυσης
  - Από το οποίο έχουν αφαιρεθεί όλα τα περιττά στάδια παραγωγής, keywords, παρενθέσεις κ.ο.κ
  - Κρατώντας μόνο τη χρήσιμη (λειτουργική) πληροφορία
- Ενδιάμεσο στάδιο μετασχηματισμού του πηγαίου κώδικα
  - Θα χρησιμοποιηθεί στα επόμενα στάδια της μεταγλώττισης
  - Η μορφή του εξαρτάται από την απαιτούμενη λειτουργικότητα του μεταγλωττιστή

Παράδειγμα AST

Η λειτουργική πληροφορία της έκφρασης `(3.0+5.0)/8.0` είναι μόνο:



(εάν το ζητούμενο είναι ο υπολογισμός εκφράσεων!)

AST: γραμματική των αριθμητικών εκφράσεων

- Τροποποιήστε τον κώδικα του συντακτικού αναλυτή για τις αριθμητικές εκφράσεις
  - Αναπαράσταση AST με πλειάδες (tuples)
- Κάθε κόμβος του AST θα είναι tuple $(parent, kidlist)$
  - όπου *kidlist* θα είναι tuple $(kid_1, kid_2, ..., kid_n)$
  - Τα φύλλα θα έχουν τη μορφή $(parent,)$
  - (tuple ενός στοιχείου)
- Στο τέλος της ανάλυσης θα έχετε ένα πλήρες δένδρο AST
  - Χρησιμοποιήστε τις επιστρεφόμενες τιμές από τις συναρτήσεις

Παράδειγμα

Για είσοδο:

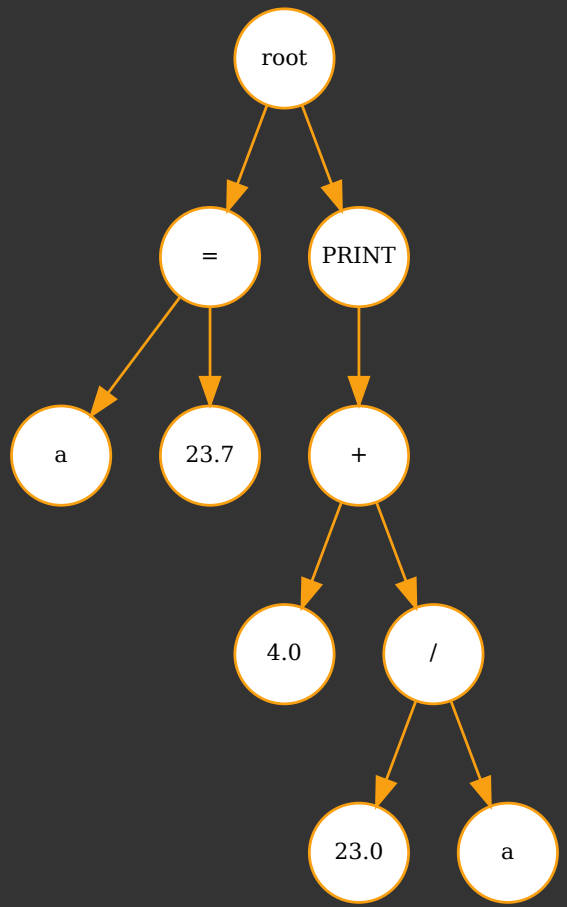
```
a = 23.7
print 4+23/a
```

θα πρέπει να πάρετε το AST

```
ast = ('root',
       (
        ('=',('a',(23.7,))),
        ('PRINT',(('+',(('4.0',),('/',((23.0,),('a',))))),))
       )
      )
```

Οπτικοποίηση AST

Θα χρησιμοποιήσετε το module pydot της Python



(για το προηγούμενο παράδειγμα AST)

pydot: παράδειγμα κώδικα

```
import pydot

# create the pydot directed graph
g = pydot.Dot(graph_type='digraph',prog='dot',splines='true',
                    overlap='false',size='80,80')

# add a graph node
node = pydot.Node("n1",shape='circle',style='filled',
                    fillcolor='#FFFFFF',fontsize='8',margin='0')
node.set_label('"1"')
g.add_node(node)

# add a second node
node = pydot.Node("n2",shape='circle',style='filled',
                    fillcolor='#FFFFFF',fontsize='8',margin='0')
node.set_label('"2"')
g.add_node(node)

# add an edge between nodes
g.add_edge(pydot.Edge("n1","n2",color="#f89f12"))

# output graph (svg format)
g.write("test.svg",format='svg')
```

Δοκιμάστε κι εσείς!

- Τροποποιήστε τον κώδικα παραγωγής του AST
  - Αρχικοποιήστε έναν γράφο του pydot
  - Προσθέστε συνάρτηση για τη διάσχιση του δένδρου
  - Κατά τη διάσχιση δημιουργήστε τους κόμβους του γράφου
  - Και διασυνδέστε τους κόμβους μεταξύ τους
  - Στο τέλος, αποθηκεύστε τον γράφο σε μορφή svg