


---

# Εισαγωγή

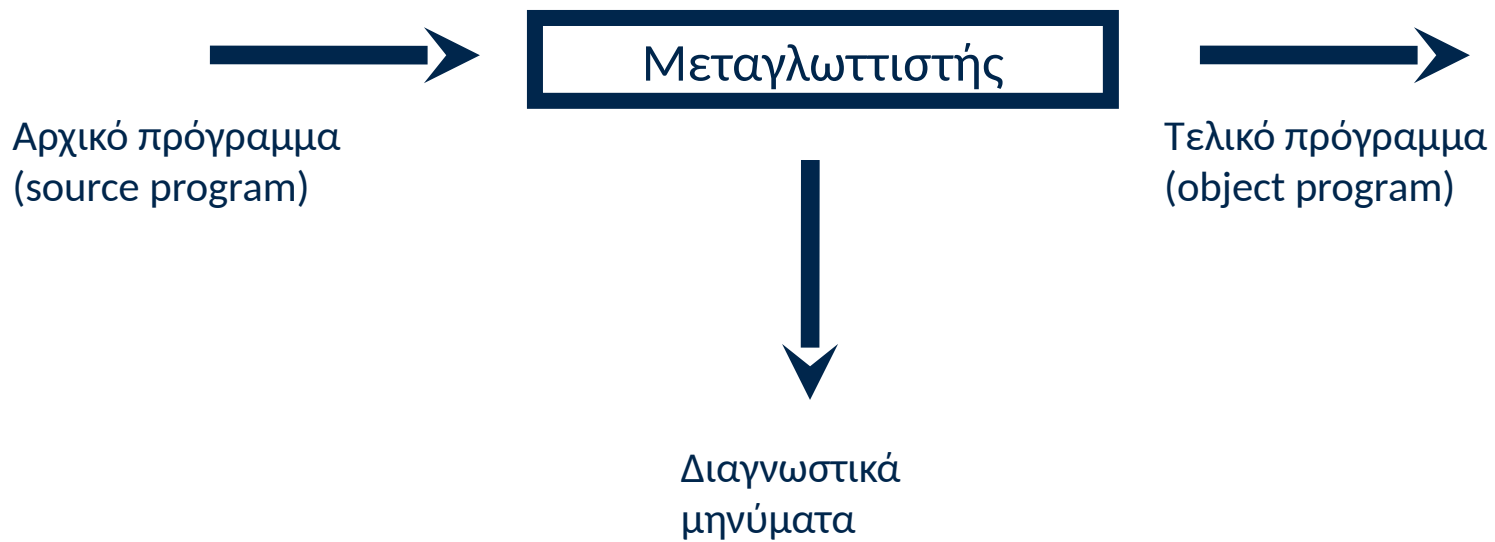
Διαλέξεις στο μάθημα: Μεταφραστές  
Γεώργιος Μανής

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ  
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
UNIVERSITY OF IOANNINA



# Μεταγλωττιστής

---



# Μεταγλωττιστής

---

- ✦ Παίρνει σαν είσοδο το **αρχικό πρόγραμμα**
  - Είναι το πρόγραμμα το οποίο θέλουμε να μεταγλωττίσουμε
- ✦ Παράγει σαν έξοδο το ισοδύναμο **τελικό πρόγραμμα**
  - Είναι το πρόγραμμα στο οποίο μετασχηματίζουμε το αρχικό πρόγραμμα
  - Συνήθως είναι πρόγραμμα σε γλώσσα μηχανής κάποιου επεξεργαστή
- ✦ Εμφανίζει **διαγνωστικά μηνύματα**
  - Συνήθως μηνύματα λάθους
  - Μερικές φορές και μηνύματα προειδοποίησης

# Γλώσσες

---

- ✦ Το αρχικό πρόγραμμα γράφεται σε αυτό που ονομάζουμε **αρχική γλώσσα**
  - Είναι η γλώσσα την οποία θέλουμε να υλοποιήσουμε
- ✦ Το τελικό πρόγραμμα παράγεται στην **τελική γλώσσα**
  - Είναι η γλώσσα στην οποία θέλουμε να μεταγλωττιστεί το πρόγραμμα
  - Συνήθως είναι η γλώσσα μηχανής κάποιου επεξεργαστή
- ✦ Ο μεταγλωττιστής είναι υλοποιημένος και αυτός σε κάποια γλώσσα προγραμματισμού, τη **γλώσσα υλοποίησης**



# Ανάπτυξη Μεταγλωττιστών

---

- # Έχουμε τρεις βασικές εναλλακτικές στην ανάπτυξη ενός μεταγλωττιστή
  - Υλοποίηση **εξ' αρχής** σε κάποια **γλώσσα προγραμματισμού**
  - Υλοποίηση με τη **χρήση εργαλείων**
    - Η υλοποίηση τους έχει γίνει χρησιμοποιώντας κατάλληλα εργαλεία που αυτοματοποιούν τη διαδικασία και την παραγωγή κώδικα
  - **Σύνθεση από μικρά τμήματα λογισμικού** που συνδυάζονται ανάλογα με τις ανάγκες

## Υλοποίηση με χρήση εργαλείων

---



## Παράδειγμα Εργαλείων Ανάπτυξης

---

### # lex (flex)

- παίρνει έως είσοδο την **περιγραφή λεκτικών μονάδων**
- από την περιγραφή αυτή **παράγεται αυτόματα λεκτικός αναλυτής** που αναγνωρίζει αυτές τις λεκτικές μονάδες
- με την αναγνώριση μίας λεκτικής μονάδας δίνεται η δυνατότητα **εκτέλεσης κώδικα** που αντιστοιχεί σε ενέργειες που θέλουμε να γίνουν μετά την αναγνώριση
- η λεκτική μονάδα που αναγνωρίστηκε μπορεί να επιστραφεί ως **αποτέλεσμα στον bison**

## Παράδειγμα Εργαλείων Ανάπτυξης

---

### # **bison (yacc)**

- παίρνει έως είσοδο την περιγραφή των **κανόνων** μίας γλώσσας σε μορφή γραμματικής
- επίσης παίρνει ως είσοδο το **αποτέλεσμα** της λεκτικής ανάλυσης από τον lex
- από τη γραμματική **παράγεται** αυτόματα **συντακτικός αναλυτής** για τη γλώσσα αυτή
- κατά διάσχιση της γραμματικής δίνεται η **εκτέλεσης κώδικα**, σε όποια σημεία επιλέξουμε



# Παράδειγμα Εργαλείων Ανάπτυξης

---

## # Antlr

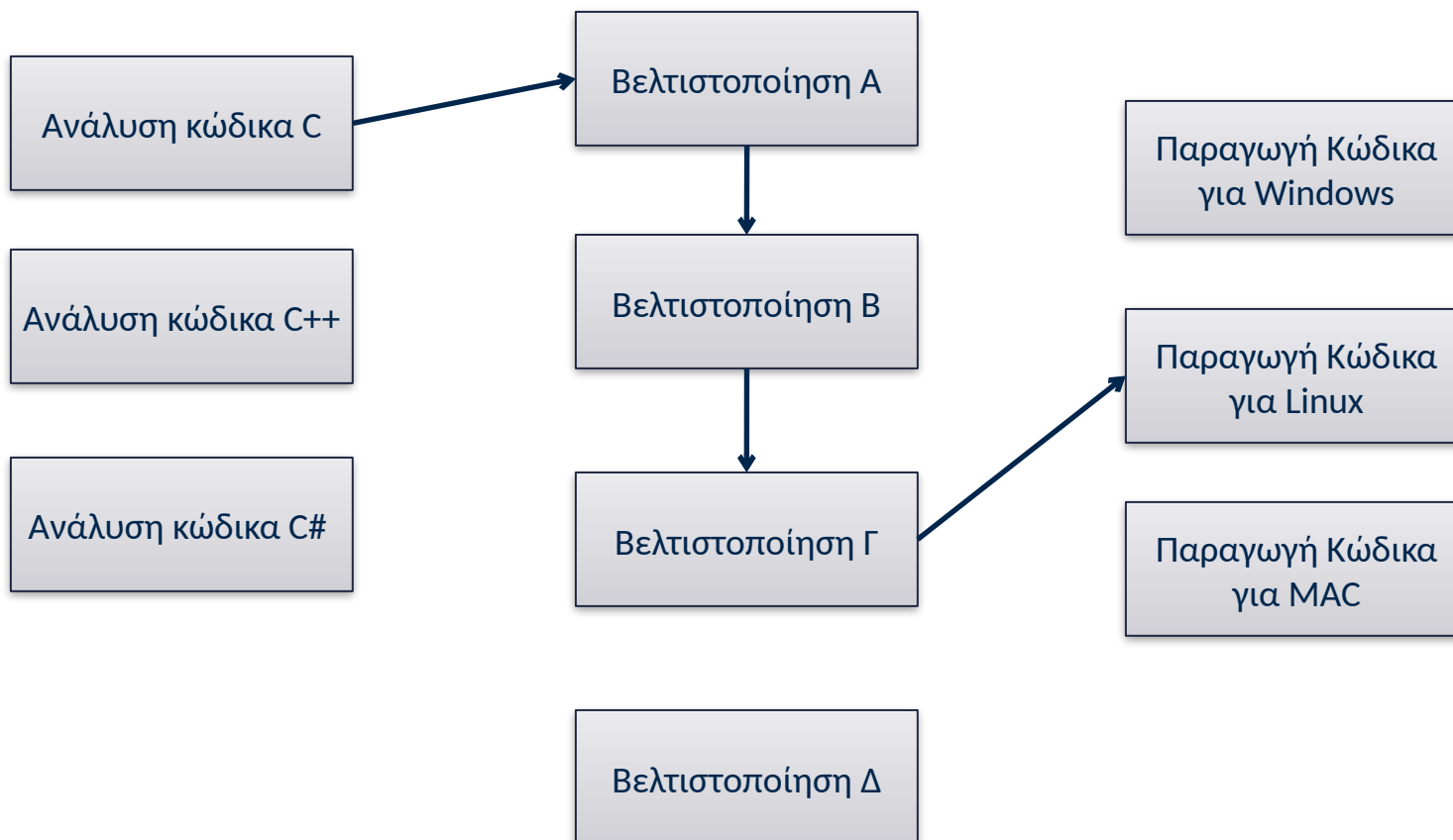
- ακολουθεί **παρόμοια φιλοσοφία** με τα δύο προηγούμενα εργαλεία
  - είναι όμως ένα **ενιαίο** εργαλείο, δεν κάνει τόσο σαφή διαχωρισμό ανάμεσα σε λεκτική και συντακτική ανάλυση, αφού συντάσσονται και τα δύο στην ίδια γραμματική
  - είναι πιο **διαδεδομένο** (πια) από τα lex-bison και **εξελίσσεται ταχύτερα**
  - είναι μάλλον η πιο **προφανής επιλογή** σήμερα
-

## Σύνθεση από Τμήματα Λογισμικού

---

- ✦ Έχουμε ένα **σύνολο από λειτουργίες** λογισμικού καλά ορισμένες, κάθε μία από τις οποίες κάνει μία συγκεκριμένη λειτουργία
  - ✦ Ανάλογα με τις ανάγκες μας **επιλέγουμε** ποιες από αυτές θα χρησιμοποιηθούν
  - ✦ Κάνουμε μικρές **μετατροπές** αν αυτό είναι αναγκαίο ή φτιάχνουμε καινούργιες με βάση τις ήδη υπάρχουσες ή εξαρχής
  - ✦ **Συνδυάζουμε** τα τμήματα που χρειαζόμαστε συνθέτοντας ένα καινούργιο μεταγλωττιστή
-

## Παράδειγμα Σύνθεσης από Τμήματα Λογισμικού



# Χρήσιμη Ορολογία

---

## # Διαμεταγλωττιστές (cross-compilers)

- Ο υπολογιστής στον οποίο γίνεται η μετάφραση είναι διαφορετική αρχιτεκτονική από τον υπολογιστή στον οποίο θα εκτελεστεί το τελικό πρόγραμμα

## # Προεπεξεργαστές (preprocessors)

- Από αρχική γλώσσα πάλι σε αρχική γλώσσα αφού έχει κάνει κάποιες τροποποιήσεις, π.χ. αντικατάσταση συμβολικών ονομάτων των σταθερών με τις πραγματικές τιμές τους

## # Συμβολομεταφραστές (assemblers)

- Από συμβολική γλώσσα μηχανής σε γλώσσα μηχανής – μία προς μία αντιστοίχιση εντολών
-

# Χρήσιμη Ορολογία

---

- # **Βιβλιοθήκες χρόνου εκτέλεσης** (run time libraries)
  - Υποπρογράμματα και συναρτήσεις σε μεταγλωττισμένη μορφή
- # **Εντοπιστές σφαλμάτων** (debuggers)
  - Βηματική εκτέλεση προγραμμάτων
  - Παρακολούθηση τιμών μεταβλητών
  - Τοποθέτηση σημείων διακοπής εκτέλεσης
- # **Στατιστικοί αναλυτές** (profiles)
  - Δίνουν πληροφορίες σχετικά με την εκτέλεση ενός προγράμματος, π.χ. απασχόληση των επεξεργαστών όταν χρησιμοποιούμε ένα υπολογιστικό σύστημα πολλών επεξεργαστών

# (προ)Ιστορική Ορολογία

---

## ■ Συνδέτες (linkers)

- Έχουν πια ενσωματωθεί στη διαδικασία μεταγλώττισης, παλαιότερα ήταν χωριστά προγράμματα
- Δέχονται σαν είσοδο μεταγλωττισμένα αρχεία και βιβλιοθήκες χρόνου εκτέλεσης και παράγουν ως έξοδο ένα εκτελέσιμο πρόγραμμα
- Αναλύουν τις σχετικές διευθύνσεις των μεταγλωττισμένων αρχείων και των προγραμμάτων βιβλιοθήκης και τα ενοποιούν ώστε να είναι όλες με βάση μία σχετική διεύθυνση

## ■ Φορτωτές (loaders)

- Δέχονται σαν είσοδο ένα εκτελέσιμο πρόγραμμα και το τοποθετούν στη μνήμη για να είναι έτοιμο προς εκτέλεση
  - Μετατρέπουν τις σχετικές διευθύνσεις σε απόλυτες διευθύνσεις
  - Αποτελεί μέρος του λειτουργικού συστήματος
-

## Διερμηνείς (*interpreters*)

---

- ✦ Μετάφραση και εκτέλεση **εντολή προς εντολή** σε αντίθεση με τους μεταγλωττιστές που μεταφράζουν μία φορά το πηγαίο πρόγραμμα το αποθηκεύουν στο δίσκο και το εκτελούν από εκεί
- ✦ **Ευκολότερη ανάπτυξη** λογισμικού από ότι με τους μεταγλωττιστές
- ✦ **Ευκολότερη** η βήμα προς βήμα εκτέλεσή άρα και η **αποσφαλμάτωση**
- ✦ Σημαντικά πιο **αργοί** από τους μεταγλωττιστές
- ✦ Σήμερα πολλές γλώσσες αξιοποιούν υβριδικές τεχνικές ώστε να συνδυάσουν ταχύτητα και μεταφερισιμότητα

## Απαιτήσεις από ένα Μεταγλωττιστή

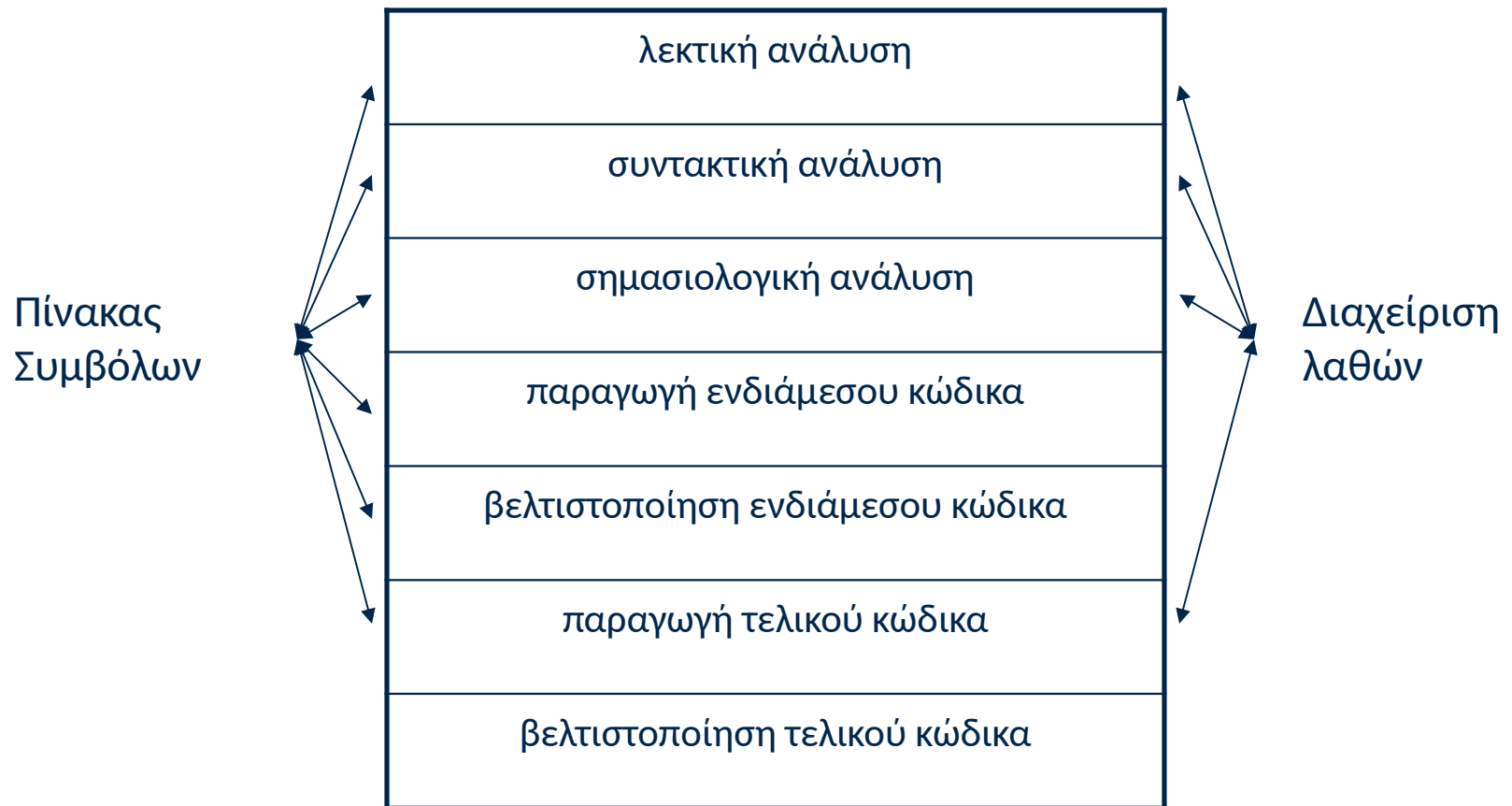
---

- # Σωστή λειτουργία
  - # Να συμμορφώνεται με τις **προδιαγραφές** αρχικής και τελικής γλώσσας
  - # Να μεταφράζει προγράμματα αυθαίρετου **μεγάλου μήκους**
  - # Να παράγει **αποδοτικό κώδικα**
  - # Να έχει **μικρό χρόνο εκτέλεσης**
  - # Να έχει **μικρές απαιτήσεις μνήμης** κατά τη μεταγλώττιση
  - # Να δίνει **καλά διαγνωστικά μηνύματα**
  - # Να έχει τη δυνατότητα **συνέχισης** ύστερα από τον **εντοπισμό σφαλμάτων**
  - # Να είναι **μεταφέρισμος**
-



## Οι Φάσεις της Μεταγλώττισης

---



# Λεκτική Ανάλυση

---



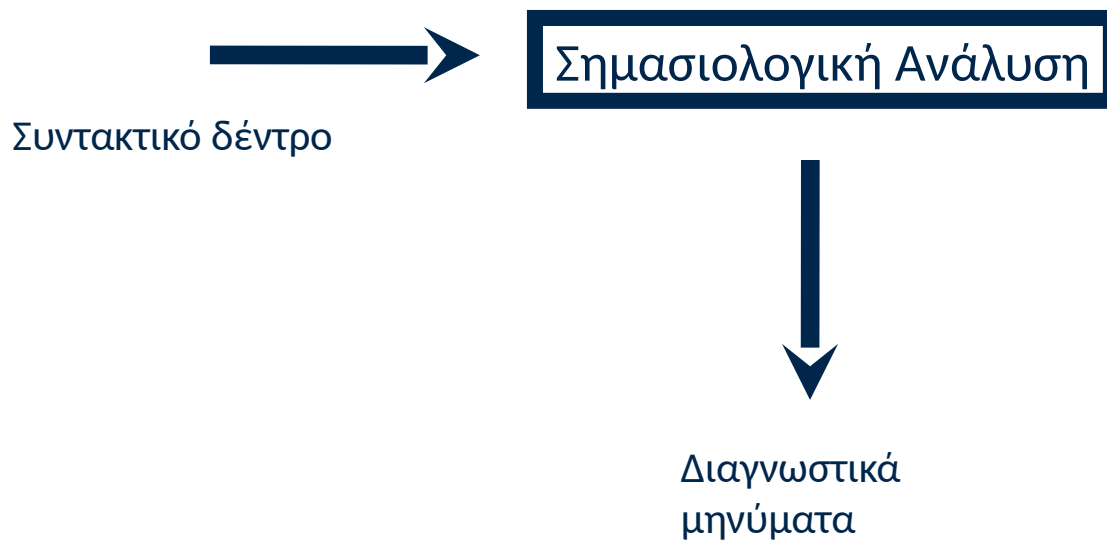
# Συντακτική Ανάλυση

---



# Σημασιολογική Ανάλυση

---



## Παραγωγή Ενδιάμεσου Κώδικα

---



## Βελτιστοποίηση Ενδιάμεσου Κώδικα

---



# Πίνακας Συμβόλων

---



## Παραγωγή Τελικού Κώδικα

---



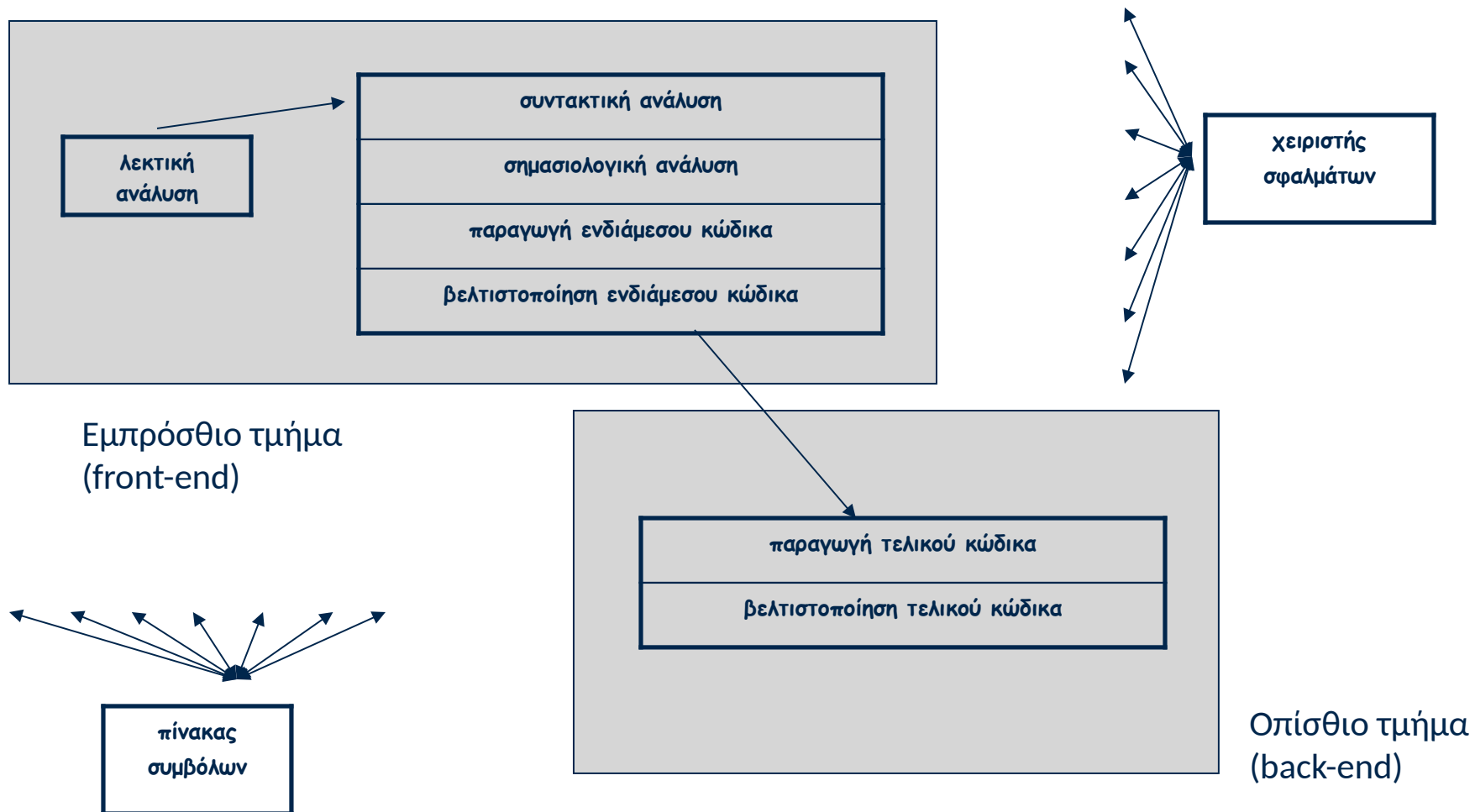


## Βελτιστοποίηση Τελικού Κώδικα

---



# Οργάνωση Μεταγλωττιστή



## Βιβλιογραφία

---

A. Aho, M. Lam, R. Sethi, J. Ullman, **“Μεταγλωττιστές: Αρχές, Τεχνικές και Εργαλεία”**, Εκδόσεις Νέων Τεχνολογιών

Keith Cooper, Linda Torczon, Πανεπιστημιακές Εκδόσεις Κρήτης, **“Σχεδίαση και Κατασκευή Μεταγλωττιστών”**

Νικόλαος Παπασπύρου, Εμμανουήλ Σκορδαλάκης, **“Μεταγλωττιστές”**, Εκδόσεις Συμμετρία

Κωνσταντίνος Λάζος, Παναγιώτης Κατσαρός, Ζαφείρης Καραϊσκος, **“Μεταγλωττιστές Γλωσσών Προγραμματισμού: Θεωρία & Πράξη”**, Εκδότης: Παναγιώτης Κατσαρός

---

# Ηλεκτρονικό Σύγγραμμα

Γεώργιος Μανής  
Αναπληρωτής Καθηγητής  
Τμήματος Μηχανικών Η/Υ και Πληροφορικής  
Πολυτεχνικής Σχολής, Πανεπιστημίου Ιωαννίνων

## ΕΓΧΕΙΡΙΔΙΟ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΑΝΑΠΤΥΞΗΣ ΜΕΤΑΓΛΩΤΤΙΣΤΩΝ

