Πανεπιστήμιο Θεσσαλίας - Τμήμα Πληροφορικής

Μεταγλωττιστές

Γιώργος Δημητρίου

Μάθημα 3ο

Λεκτική Ανάλυση και Λεκτικοί Αναλυτές

- Γενικά για τη λεκτική ανάλυση
  - Έννοιες που χρειαζόμαστε
  - Τεχνικές λεκτικής ανάλυσης
  - Πίνακας συμβόλων και διαχείριση λαθών
- Σχεδίαση λεκτικού αναλυτή
  - Καταγραφή χαρακτηριστικών
  - Προγραμματισμός του λεκτικού αναλυτή

Λεκτική Ανάλυση

- Η αναγνώριση των λεκτικών μονάδων ενός προγράμματος
- Η πρώτη φάση της μετάφρασης
- Η μόνη φάση που έχει άμεση επαφή με την είσοδο του μεταγλωττιστή
  - Προεπεξεργασία
  - Μηνύματα ελέγχου λαθών
  - Εισαγωγή ονομάτων στον πίνακα συμβόλων

Αλφάβητο και Λεκτικές Μονάδες

- Αλφάβητο είναι συνήθως το σύνολο των ASCII χαρακτήρων
- Οι λέξεις που αναγνωρίζει ο Λεκτικός Αναλυτής (ΛΑ) κωδικοποιούνται στις λεκτικές μονάδες της γλώσσας
- Επειδή κάποιες από τις λέξεις δε συμμετέχουν στη σύνταξη της γλώσσας (πχ. σχόλια), αυτές δεν παράγουν λεκτικές μονάδες

Αναγνώριση Λεκτικών Μονάδων

- Η κανονική έκφραση
  - πχ letter (letter | digit | ‘_’) *
- Η λεκτική μονάδα (token)
  - πχ T_ELSE, ID, FCONST
- Η λέξη (lexeme)
  - η ακριβής συμβολοσειρά που αναγνωρίστηκε ως η λεκτική μονάδα
  - πχ “else”, “parse_tree_root”, “100.325E-2”

Αναγνώριση με ΜΠΑ-ε

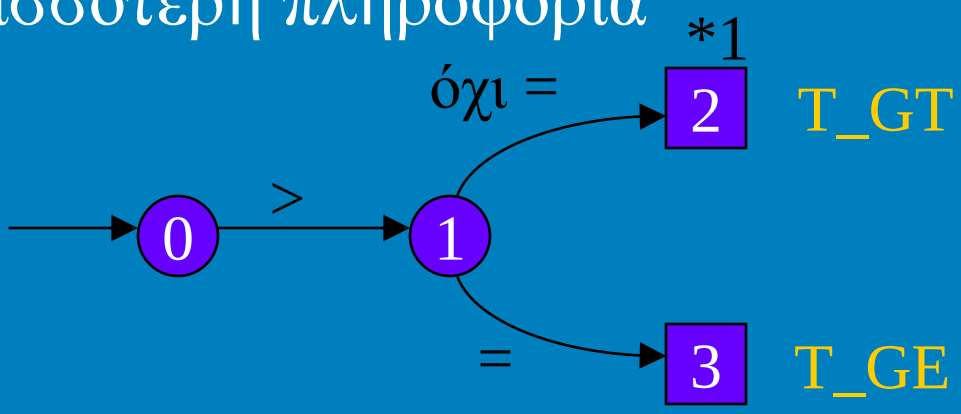
- Κατασκευάζουμε ένα ΜΠΑ-ε για κάθε κανονική έκφραση
- Ενοποίηση όλων των ΜΠΑ-ε σε ένα
- Ξεκινάμε από την αρχική κατάσταση και προχωράμε με σύνολα καταστάσεων
- Σταματάμε όταν δεν ορίζεται μετάβαση
- Βρίσκουμε το *επιτυχημένο σύνολο*
  - Μέγιστη αναγνωρισμένη λεκτική μονάδα

Συμφραζόμενα / Οπισθοδρόμηση

- Πόσοι χαρακτήρες πρέπει να διαβαστούν για να αναγνωριστεί μια λεκτική μονάδα;
- Είναι όλοι οι χαρακτήρες μέρος της λεκτικής μονάδας;
  - Συμφραζόμενα!?
- Τι κάνουμε, εάν έχουμε διαβάσει περισσότερους χαρακτήρες από όσους αποτελούν τη λεκτική μονάδα;

Διάγραμμα Μετάβασης (ΔΜ)

- Αναγνώριση με πεπερασμένο αυτόματο
- Επαυξημένος γράφος μετάβασης
  - Δυνατότητα οπισθοδρόμησης
  - Περισσότερη πληροφορία



Λειτουργία ΛΑ από το ΔΜ

- Για κάθε λεκτική μονάδα ξεκινάμε από την αρχική κατάσταση του ΔΜ
- Απορρίπτουμε όταν δε βρίσκουμε ορισμένη μετάβαση
- Σε κάθε τελική κατάσταση:
  - Αναγνωρίζουμε την αντίστοιχη λεκτική μονάδα
  - Πιθανά οπισθοδρομούμε στην είσοδο
  - Επιστρέφουμε την πληροφορία που χρειάζεται ο Συντακτικός Αναλυτής (ΣΑ)

Κατασκευή ΔΜ

- Εμπειρική κατασκευή:
  - Κατασκευάζουμε ένα ΝΠΑ για κάθε κανονική έκφραση
  - Μετατροπή κάθε ΝΠΑ σε ΔΜ με προσαύξηση για οπισθοδρόμηση και επιστροφή πληροφορίας στο ΣΑ
  - Ενοποίηση όλων των ΔΜ σε ένα
  - Το τελικό ΔΜ πρέπει να είναι ντετερμινιστικό

Πίνακας Συμβόλων (ΠΣ)

- Στη φάση αυτή μπορεί να γίνει εισαγωγή στον ΠΣ των ονομάτων που αναγνωρίζει ο ΛΑ
- Οργάνωση ΠΣ
  - Γραμμικές λίστες
  - Δέντρα δυαδικής αναζήτησης
  - Πίνακες κατακερματισμού
- Στοιχεία σημασιολογίας θα προστεθούν στις επόμενες φάσεις!

Διαχείριση Λαθών (ΔΛ)

- Εκτύπωση μηνυμάτων
- Ανάνηψη από σφάλματα
  - Μέθοδος πανικού
  - Διαγραφή μη επιτρεπτού χαρακτήρα
  - Εισαγωγή χαρακτήρα
  - Αντικατάσταση χαρακτήρα
  - Αντιμετάθεση χαρακτήρων
- Πολυπλοκότητα / Κόστος / Επιτυχία μεθόδου

Σχεδίαση ΛΑ

- Ορισμός αλφαβήτου
- Καταγραφή των λεκτικών μονάδων
- Κατασκευή ΔΜ ή πίνακα μεταβάσεων ΜΠΑ-ε
- Υλοποίηση ΛΑ
  - Υλοποίηση λογισμικού προσομοίωσης λειτουργίας του ΔΜ ή του ΜΠΑ-ε
  - Αρχική υλοποίηση ΠΣ
  - Υλοποίηση λογισμικού ΔΛ του ΛΑ

Κατηγορίες Λεκτικών Μονάδων

- Λέξεις κλειδιά (πχ: and if while)
- Αναγνωριστικά (Ονόματα) (πχ: iff point_1_3)
- Σταθερές χωρίς πρόσημο (πχ: 2 2.3 0.1e3 ‘a’)
- Τελεστές (πχ: ‘+’ ‘*’ ‘/’ “<>” “:=”)
- Διαχωριστές (πχ: ‘[’ ‘(’ ‘,’ ‘.’ “::”)
- Τέλος αρχείου (<<EOF>> ή $)
- Λέξεις ΛΑ που δεν είναι λεκτικές μονάδες
  - Σχόλια, κενά

Ανάγνωση Προγράμματος

- Ανάγνωση με δυνατότητα οπισθοδρόμησης
  - Εισαγωγή όλου του αρχείου στη μνήμη!
- Αρίθμηση γραμμών για ΔΛ
  - Θέση λεκτικών μονάδων στη γραμμή
- Ρόλος κενών στο ΛΑ
- Πεζά / Κεφαλαία γράμματα
- Φώλιασμα σχολίων

Υλοποίηση ΛΑ

- Με κατ' ευθείαν προγραμματισμό - προσομοίωση της μηχανής που χρησιμοποιούμε
  - ΜΠΑ-ε
  - ΔΜ
- Με τη βοήθεια μετα-εργαλείου γεννήτορα ΛΑ, όπως το Lex ή το Flex

Κατ' ευθείαν Προγραμματισμός

- Προσομοίωση του ΔΜ με κώδικα περιγραφής μεταβάσεων, όπως με C:

```
while(true){
    ch = get_next_character();
    switch(state){
    case 10: if (isalnum(ch))
                {...; state = 11;}
             else
                {...; unget_character(); return TOKEN;}
    ...
    }
}
```

- Χρήση πίνακα μετάβασης
  - Απλή υλοποίηση με ένα βρόχο διαδοχικών μεταβάσεων κατάστασης

Χρήση Lex / Flex

- Πρώτο μέρος:
  - Κώδικας (πχ ορισμοί κωδικών λεκτικών μονάδων)
  - Συντομογραφίες βασικών κανονικών εκφράσεων
- Δεύτερο μέρος:
  - Κανόνες αναγνώρισης λεκτικών μονάδων
  - Κώδικας που εκτελείται με κάθε αναγνώριση
- Τρίτο μέρος:
  - Συμπληρωματικός κώδικας (πχ ορισμός main())

ΛΑ και ΣΑ

- Ομοιότητες και διαφορές
- Γιατί υλοποιούμε ξεχωριστά το ΛΑ από το ΣΑ;
  - Απλουστεύουμε και μειώνουμε την ήδη μεγάλη πολυπλοκότητα του ΣΑ
  - Πιο εξειδικευμένη σχεδίαση ΛΑ
  - Μη εξάρτηση των υπόλοιπων φάσεων από τη μορφή του προγράμματος