



Σχεδίαση Γλωσσών & Μεταγλωττιστές

Ενότητα 6: Αλγόριθμος Προβλέπουσας
Αναδρομικής Κατάβασης

Επ. Καθ. Π. Κατσαρός
Τμήμα Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδεια χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ανάλυση προβλέπουσας αναδρομικής κατάβασης I

Στην ανάλυση αναδρομικής κατάβασης ο κάθε κανόνας, που αναφέρεται σε κάποιο μη τερματικό σύμβολο, εκφράζεται από τον ορισμό μιας διαδικασίας, που θα το αναγνωρίζει.

Ένας αναλυτής προβλέπουσας αναδρομικής κατάβασης αποτελείται:

- από μία καθολική μεταβλητή, που περιέχει την τιμή της τρέχουσας λεξικής μονάδας
- από μία βοηθητική διαδικασία αναγνώρισης, που ελέγχει αν η τρέχουσα λεξική μονάδα είναι η αναμενόμενη και καλεί τη διαδικασία λεξικής ανάλυσης, για την ανάγνωση της επόμενης λεξικής μονάδας και την ενημέρωση της καθολικής μεταβλητής
- από τις διαδικασίες ανάλυσης, που αντιστοιχούν στα μη τερματικά σύμβολα της γραμματικής
- από μια διαδικασία εκκίνησης, που αφού διαβάσει την πρώτη λεξική μονάδα καλεί τη διαδικασία, που αντιστοιχεί στο μη τερματικό σύμβολο της αρχής.



Ανάλυση προβλέπουσας αναδρομικής κατάβασης VI

Δοθείσης μιας γραμματικής που έχει την ιδιότητα $LL(1)$

- μπορούμε να γράψουμε διαδικασίες που να αναγνωρίζουν το αριστερό μέρος της κάθε παραγωγής
- ο κώδικας της ανάλυσης είναι απλός και γρήγορος

Οι γραμματικές με την ιδιότητα $LL(1)$ ονομάζονται προγνώσιμες γραμματικές γιατί ο αναλυτής μπορεί να «προγνώσει» τη σωστή ανάπτυξη σε κάθε σημείο της ανάλυσης.

Οι αναλυτές που εκμεταλλεύονται την ιδιότητα $LL(1)$ ονομάζονται αναλυτές πρόγνωσης.

Μία περίπτωση ανάλυσης πρόγνωσης είναι η ανάλυση προβλέπουσας αναδρομικής κατάβασης.



Ανάλυση προβλέπουσας αναδρομικής κατάβασης VII

Παράδειγμα διορθωμένης γραμματικής αριθμητικών εκφράσεων

(βλ. διαφάνεια Ανάλυση προβλέπουσας αναδρομικής κατάβασης III)

```

int PLUS=1, MINUS=2, ...
int lookahead = getNextToken(); //καθολική μεταβλητή ΔΕΕ_MON
void advance() { lookahead = getNextToken(); }
void match(int token) { if (lookahead == token) advance(); else error(); }

void S() { Expr(); }

void Expr() { Term(); ExprPrime(); }

void ExprPrime() {
    switch(lookahead) {
        case PLUS : match(PLUS); Term();
                    ExprPrime(); break;
        case MINUS : match(MINUS); Term();
                    ExprPrime(); break;
        default: return;
    }
}

void Term() { Factor(); TermPrime(); }

void TermPrime() {
    switch(lookahead) {
        case TIMES: match(TIMES); Factor();
                    TermPrime(); break;
        case DIV: match(DIV); Factor();
                TermPrime(); break;
        default: return;
    }
}

void Factor() {
    switch(lookahead) {
        case LPAR : match(LPAR); Expr();
                    match(RPAR); break;
        case NUMBER: match(NUMBER); break;
        default: error();
    }
}

```



Ανάλυση προβλέπουσας αναδρομικής κατάβασης VIII

Παράδειγμα διορθωμένης γραμματικής αριθμητικών εκφράσεων
(βλ. διαφάνεια Ανάλυση προβλέπουσας αναδρομικής κατάβασης III)

```
Expr() {  
    Term(); ExprPrime();  
/*  
    δημιουργία κόμβου Expr;  
    εξαγωγή κόμβου ExprPrime από στοίβα;  
    εξαγωγή κόμβου Term από στοίβα;  
    καθιστούμε ExprPrime και Term  
        απογόνους του Expr;  
    εισαγωγή κόμβου Expr στη στοίβα;  
*/  
}
```



Ανάλυση προβλέπουσας αναδρομικής κατάβασης IX

Άλλο παράδειγμα

S	=	“if” E “then” S “else” S
		“begin” $S L$
		“print” E .
L	=	“end”
		“;” $S L$.
E	=	“num” “=” “num”

```
void S() {
    switch(lookahead) {
        case IF: match(IF); E(); match(THEN); S();
                match(ELSE); S(); break;
        case BEGIN: matvh(BEGIN); S(); L(); break;
        case PRINT: match(PRINT); E(); break;
        default: error();
    }
}
```

```
void L() {
    switch(lookahead) {
        case END: match(END); break;
        case SEMI: match(SEMI); S();
                  L(); break;
        default: error();
    }
}
```

```
void E() { match(NUM); match(EQ); match(NUM); }
```





Ανάλυση προβλέπουσας αναδρομικής κατάβασης X

Άλλο παράδειγμα (συνέχεια)

πρόταση: if 2=2 then print 5=5 else print 1=1



Ανάλυση προβλέπουσας αναδρομικής κατάβασης III

Η προβλέπουσα αναδρομική κατάβαση στηρίζεται στην **πρόγνωση** του κατάλληλου κάθε φορά κανόνα, που οδηγεί στην παραγωγή του δένδρου της πρότασης.

Δε μπορεί να εφαρμοσθεί σε αριστερά αναδρομικές γραμματικές, όπως αυτή του προηγούμενου παραδείγματος.

Ας θεωρήσουμε τη μη αριστερά αναδρομική γραμματική των αριθμητικών εκφράσεων:

$S = \text{έκφραση.}$

$\text{έκφραση} = \text{όρος υπ_όροι.}$

$\text{υπ_όροι} = "+" \text{όρος υπ_όροι} \mid "-" \text{όρος υπ_όροι} \mid \epsilon.$

$\text{όρος} = \text{παράγοντας υπ_παραγ.}$

$\text{υπ_παραγ} = "*" \text{παράγοντας υπ_παραγ} \mid "/" \text{παράγοντας υπ_παραγ} \mid \epsilon.$

$\text{παράγοντας} = "(" \text{έκφραση} ")" \mid \text{"αριθμός"}.$



Ανάλυση προβλέπουσας αναδρομικής κατάβασης IV

Για το μη τερματικό σύμβολο «έκφραση», μπορεί εναλλακτικά να έχουμε την παραγωγή,

έκφραση \Rightarrow όρος υπ_όροι \Rightarrow παράγοντας υπ_παραγ υπ_όροι \Rightarrow (έκφραση) υπ_παραγ υπ_όροι

ή την παραγωγή

έκφραση \Rightarrow όρος υπ_όροι \Rightarrow παράγοντας υπ_παραγ υπ_όροι \Rightarrow **αριθμός** υπ_παραγ υπ_όροι

Γενικά, για να στηριχθεί η ανάλυση στην τεχνική της πρόγνωσης χρειάζεται για κάθε μη τερματικό σύμβολο, να είναι εκ των προτέρων γνωστό το σύνολο των τερματικών, που είναι δυνατό να εμφανισθούν στην αρχή των συμβολοσειρών, που παράγονται από αυτό (σύνολο **FIRST**). Επιπλέον, αν η γραμματική περιλαμβάνει κανόνες-ε, τότε για κάθε μη τερματικό σύμβολο που βρίσκεται στο αριστερό μέρος ενός τέτοιου κανόνα, επιβάλλεται και ο υπολογισμός του συνόλου των τερματικών, που μπορεί να εμφανισθούν αμέσως μετά από αυτό (σύνολο **FOLLOW**).



Ανάλυση προβλέπουσας αναδρομικής κατάβασης V

Ανάλυση πρόγνωσης: κεντρική ιδέα

Δοθείσης μιας παραγωγής $A \rightarrow \alpha \mid \beta$, ο αναλυτής πρέπει να είναι σε θέση να επιλέξει μεταξύ α & β

Σύνολα FIRST

Για ένα δεξί μέρος παραγωγής $\alpha \in G$, ορίζουμε ως $FIRST(\alpha)$ το σύνολο των αναγνωριστικών που εμφανίζονται πρώτα στις συμβολοσειρές που παράγονται από το α .

Δηλαδή ισχύει, $x \in FIRST(\alpha)$ αν και μόνο αν $\alpha \Rightarrow^* x \gamma$, για κάποια συμβολοσειρά γ

Γραμματική LL(1)

Αν $A \rightarrow \alpha$ και $A \rightarrow \beta$ εμφανίζονται στη γραμματική πρέπει

$$FIRST(\alpha) \cap FIRST(\beta) = \emptyset$$

Αυτό επιτρέπει στον αναλυτή να προγνώσει με ασφάλεια την παραγωγή που θα χρησιμοποιεί σε κάθε βήμα!



Recursive descent με σύνολα FIRST & Follow

Για την ανάλυση προβλέπουσας αναδρομικής κατάβασης (recursive descent) η γενική μορφή μιας διαδικασίας για ένα μη τερματικό σύμβολο A με κανόνες

$$A \rightarrow a_1 | a_2 | \dots | a_n \text{ (υποθέτουμε ότι } \varepsilon \in \text{FIRST}(a_i) \text{ για κάποιο } a_i)$$

είναι:

A:

switch *lookahead*

case $k \in \text{FIRST}(a_1)$:

κλήση διαδικασιών που αντιστοιχούν στα σύμβολα a_1
(match για τα τερματικά και κλήση αντίστοιχων διαδικασιών
για τα μη τερματικά)

case $k \in \text{FIRST}(a_2)$:

κλήση διαδικασιών που αντιστοιχούν στα σύμβολα του a_2
(ομοίως)

.....

case $m \in \text{FIRST}(a_i) \cup \text{FOLLOW}(A)$:

κλήση διαδικασιών που αντιστοιχούν στα σύμβολα του a_i ή
στα σύμβολα που έπονται του A
(ομοίως)

else

syntax error

end switch;



Ανάλυση προβλέπουσας αναδρομικής κατάβασης XVII

- **Δεν μπορούν να χρησιμοποιηθούν αριστερά αναδρομικές γραμματικές.**
- **Δεν μπορούν να χρησιμοποιηθούν γραμματικές, που δεν είναι LL(1), όπως π.χ.:**
σειρά_εντολών = εντολή ";" σειρά_εντολών | εντολή.
εντολή = "τερματικό".





Τέλος ενότητας

Επεξεργασία: Εμμανουέλα Στάχτιαρη
Θεσσαλονίκη, 21/07/2014



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ