|  |  |
| --- | --- |
|  | **Πανεπιστήμιο Αιγαίου**  **Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων** |

# Προηγμένα Θέματα Γλωσσών Προγραμματισμού

**Εργασία 4η (Ομαδική)**

**321/2014134 Μπόνης Αθανάσιος**

**321/2011039 Δημόπουλος Γεώργιος**

**321/2014190 Σταύρου Γιάννης**

**Σάββατο 19/05/2018**

Περιεχόμενα

[Προηγμένα Θέματα Γλωσσών Προγραμματισμού 1](#_Toc514519088)

[Ερώτηση 1η 3](#_Toc514519089)

# Ερώτηση 1η

Η περιγραφή των λεκτικών στοιχείων με Κανονικές Εκφράσεις(Regular Expressions) δίνεται παρακάτω:

PROGRAM → 'PROGRAM';

INT\_TYPE → 'INTEGER';

BOOL\_TYPE → 'BOOLEAN';

STRING\_TYPE → 'STRING';

ARRAY → 'ARRAY';

OF → 'OF';

READ → 'READ';

WRITE → 'WRITE';

IF → 'IF';

THEN → 'THEN';

ELSE → 'ELSE';

WHILE → 'WHILE';

DO → 'DO';

EXIT → 'EXIT';

VAR → 'VAR';

BODY → 'BODY';

BEGIN → 'BEGIN';

END → 'END';

AND → 'AND';

OR → 'OR';

NOT → 'NOT';

TRUE → 'TRUE';

FALSE → 'FALSE';

PLUS → '+';

MINUS → '-';

MULTIPLE → '\*';

DIV → '/';

MOD → '%';

EQUALS → '=';

GREATER → '>';

LESS → '<';

LFPAR → '(';

RTPAR → ')';

COMMA → ',';

COLON→ ':';

LBRA → '[';

RBRA → ']';

S\_CONC → '|';

UNDERSCORE → '\_';

SEMICOLON → ';';

ASSIGN → ':=';

GTEQ → '>=';

LTEQ → '<=';

NEQ → '<>';

PNAME → LETTER+ DIGIT\*;

STRING → '"' (~('"' | '\\' | '\r' | '\n') | '\\' ('"' | '\\'))\* '"';

COMMENT → '{' .\*? '}' -> skip ;

WS → [ \t\r\n]+ -> skip ;

# Ερώτηση 2η

Στη συγκεκριμένη περίπτωση γραμματικής, για να είναι recursive-descent η ανίχνευση πρέπει να γίνει απαλοιφή της αριστερής αναδρομής. Παρακάτω βλέπουμε την μετατροπή που έχουμε κάνει ώστε να είναι recursive-descent ανίχνευση.

stmtlist 🡪 statement SEMICOLON nStatement

nStatement 🡪 statement SEMICOLON nStatement

|ε

statement 🡪 lvalue ASSIGN expr

| IF expr THEN statement

| IF expr THEN statement ELSE statement

| WHILE expr DO statement

| EXIT

lvalue 🡪 ID LBRACK index RBRACK

| ID

Παρακάτω παρατίθεται ο ψευδοκώδικας για τις recursive-descent ρουτίνες για τους κανόνες statement και lvalue.

