Άσκηση 1
Χαράλαμπος Παπαδόπουλος
AM: 1115200800169

Ερώτηση 1)

Η αρχική γραμματική που μας δόθηκε ήταν:

```
exp   -> num | exp op exp | (exp)
op    -> + | - | * | /
num   -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Για να υποστηρίζεται προτεραιότητα του πολλαπλασιασμού και της διαίρεσης έναντη της πρόσθεσης και της αφαίρεσης θα πρέπει οι πρώτοι δυο τελεστές (*, /) να βρίσκονται πιο βαθιά στο δέντρο που προκύπτει από τους άλλους δυο (+, -). Αυτό εξασφαλίζεται αν τους βάλουμε και "βαθύτερα" στη γραμματική. Η νέα γραμματική είναι:

```
exp   -> exp + term | exp - term | term
term  -> term * factor | term / factor | factor
factor-> num | (exp)
num   -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Η παραπάνω γραμματική έχει αριστερή αναδρομή. Αν θέλουμε να την εξαλείψουμε δεν θα πρέπει να υπάρχει κανόνας που το σώμα του να ξεκινάει με την κεφαλή του. Η καινούρια γραμματική είναι:

```
exp   -> term exp2
exp2  -> + term exp2 | - term exp2 | ε
term  -> factor term2
term2 -> * factor term2 | / factor term2 | ε
factor-> num | (exp)
num   -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

FIRST(num) = FIRST(num -> 0) U FIRST(num -> 1) U FIRST(num -> 2) U FIRST(num -> 3) U FIRST(num -> 4) U FIRST(num -> 5) U FIRST(num -> 6) U FIRST(num -> 7) U FIRST(num -> 8) U FIRST(num -> 9) = {1} U {2} U {3} U {4} U {5} U {6} U {7} U {8} U {9} = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
FIRST(factor) = FIRST(factor -> num) U FIRST(factor -> (exp)) = FIRST(num) U FIRST(() = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} U {(} = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (}
FIRST(term2) = FIRST(term2 -> * factor term2) U FIRST(term2 -> / factor term2) U FIRST(term2 -> ε) = {*} U {/} U {ε} = {*, /, ε}
FIRST(term) = FIRST(term -> factor term2) = FIRST(factor) = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (}
FIRST(exp2) = FIRST(exp2 -> + term exp2) U FIRST(exp2 -> - term exp2) U FIRST(exp2 -> ε) = {+} U {-} U {ε} = {+, -, ε}
FIRST(exp) = FIRST(exp -> term exp2) = FIRST(term) = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (}

FOLLOW(exp) = {EOF} U {)} = {EOF,)}
FOLLOW(exp2) = FOLLOW(exp) U FOLLOW(exp2) = FOLLOW(exp) = {EOF,)}
FOLLOW(term) = FIRST(exp2) U FOLLOW(exp2) = {+, -, ε} U {EOF,)} = {+, -, ε, EOF,)} = {+, -, EOF,)}
FOLLOW(term2) = FOLLOW(term) U FOLLOW(term2) = FOLLOW(term) = {+, -, EOF,)}
FOLLOW(factor) = FIRST(term2) U FOLLOW(term) U FOLLOW(term2) = {*, /, ε} U {+, -, EOF,)} U {+, -, EOF,)} = {+, -, *, /, EOF,)}
FOLLOW(num) = FOLLOW(factor) = {+, -, *, /, EOF,)}

FIRST+(exp -> term exp2) = FIRST(term) = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (}

FIRST+(exp2 -> + term exp2) = FIRST(+) = {+}
FIRST+(exp2 -> - term exp2) = FIRST(-) = {-}
FIRST+(exp2 -> ε) = FIRST(ε) U FOLLOW(exp2) = {ε} U {EOF,)} = {ε, EOF,)}

FIRST+(term -> factor term2) = FIRST(factor) = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (}

FIRST+(term2 -> * factor term2) = FIRST(*) = {*}
FIRST+(term2 -> / factor term2) = FIRST(/) = {/}
FIRST+(term2 -> ε) = FIRST(ε) U FOLLOW(term2) = {ε} U {+, -, EOF,)} = {ε, +, -, EOF,)}

FIRST+(factor -> num) = FIRST(num) = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
FIRST+(factor -> (exp)) = FIRST(() = {(}

FIRST+(num -> 0) = FIRST(0) = {0}
FIRST+(num -> 1) = FIRST(1) = {1}
FIRST+(num -> 2) = FIRST(2) = {2}
FIRST+(num -> 3) = FIRST(3) = {3}
FIRST+(num -> 4) = FIRST(4) = {4}
FIRST+(num -> 5) = FIRST(5) = {5}
FIRST+(num -> 6) = FIRST(6) = {6}
FIRST+(num -> 7) = FIRST(7) = {7}
FIRST+(num -> 8) = FIRST(8) = {8}
FIRST+(num -> 9) = FIRST(9) = {9}

Παρατηρώ οτι για κάθε ομάδα κανόνων με κοινή κεφαλή, τα αντίστοιχα FIRST+ σύνολα είναι εντελώς ξένα μεταξύ τους.
Άρα η γραμματική είναι LL(1).

Lookahead πίνακας:

	+, -	*, /	num = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}	(	)	EOF
exp	error	error	term2 exp2	exp) term2 exp2	error	error
exp2	term exp2	error	error	error	ε	ε
term	error	error	term2	exp) term2	error	error
term2	ε	factor term2	error	error	ε	ε
factor	error	error	(do nothing)	exp)	error	error

Στο project parser είναι το 2ο ερώτημα και στο project sableCCparser είναι το 3ο ερώτημα