

# **ΜΕΡΟΣ Β-2**

## **FLEX & BISON**



Μέλη Ομάδας :

**Τμήμα Β2**

**Ομάδα 6**

1. ΗΛΙΑΣ ΔΗΜΑΣ (71347267)
2. ΝΙΚΟΛΑΟΣ ΠΑΠΠΑΣ (47422)
3. ΡΟΜΑΝΙΟΥΚ ΒΙΚΤΩΡ (713242017024)
4. ΤΖΩΡΤΖΗΣ ΑΓΓΕΛΟΣ (18390094)

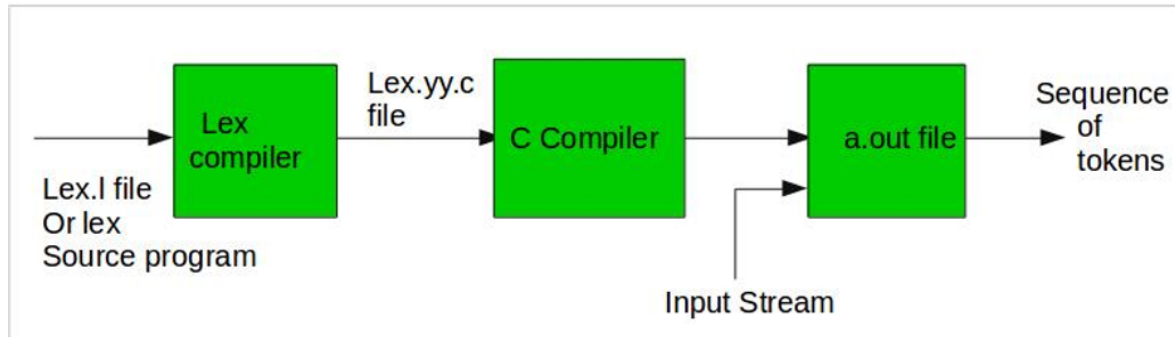
# Πίνακας Περιεχομένων

Τι είναι το FLEX;	2
Κανονικές εκφράσεις FLEX	3
Τι είναι το BISON:	3
Αρχείο Εισόδου(in.txt)	5
Σχόλια και Αποτελέσματα του αρχείου in.txt	6
Προβλήματα που προέκυψαν και πώς αντιμετωπίστηκαν:	6
Συμπέρασμα:	7
Αρμοδιότητες των μελών της ομάδας:	7

## Τι είναι το FLEX;

Το flex είναι ένα εργαλείο για τη δημιουργία σαρωτών: προγράμματα που αναγνωρίζουν λεξικά μοτίβα στο κείμενο. Το flex διαβάζει τα δεδομένα αρχεία εισόδου ή την τυπική του είσοδο εάν δεν δίνονται ονόματα αρχείων, για να δημιουργήσει μια περιγραφή ενός σαρωτή.

Σχηματική Απεικόνιση της λειτουργίας του Flex:



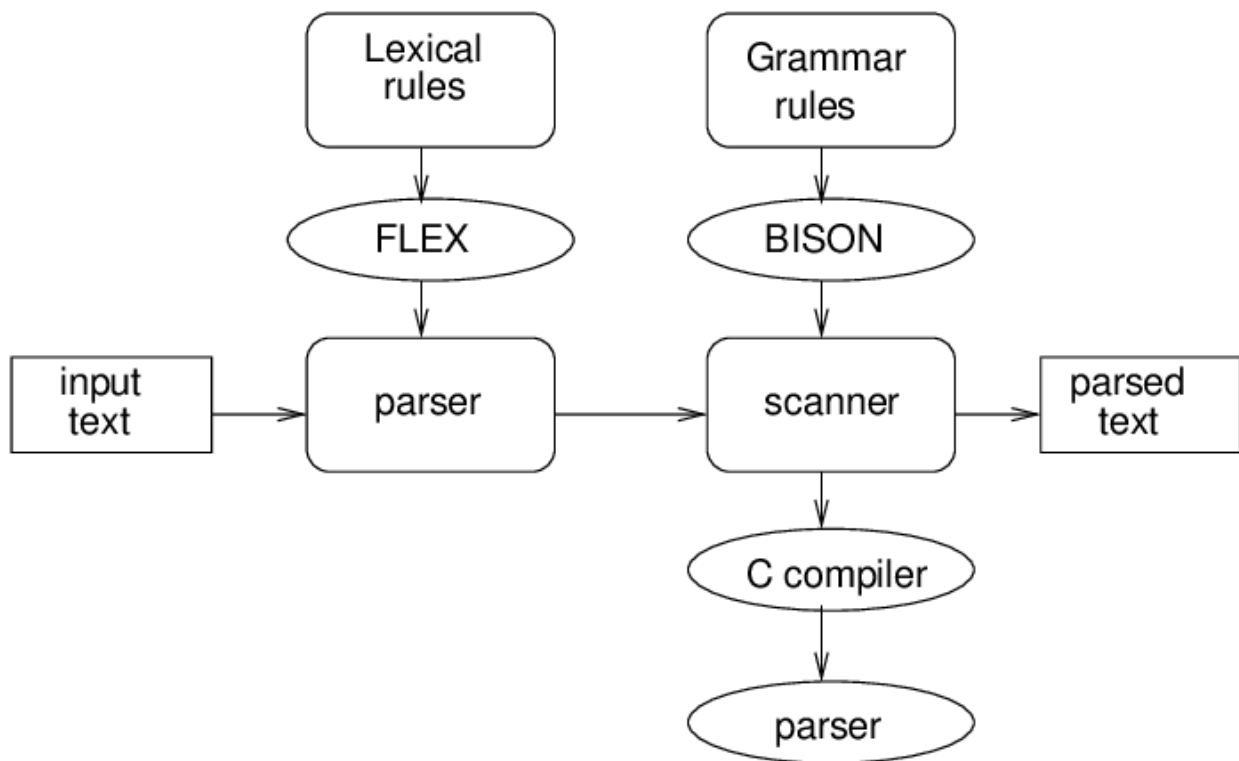
## Κανονικές εκφράσεις FLEX

DELIMITER	<code>[ \t ]+</code>
INTEGERS	<code>[ \s \t ]* ( 0 [ + - ]? [ 1 - 9 ]+ [ 0 - 9 ]* ) [ \s \t ]*</code>
FLOATS	<code>[ \s \t ]* [ - + ]? ( 0 [ 1 - 9 ]+ [ 0 - 9 ]* ) ( . [ 0 - 9 ]+ ( [ e E ] [ + - ] ( 0 [ 1 - 9 ]+ [ 0 - 9 ]* ) )?   [ e E ] [ + - ]? ( 0 [ 1 - 9 ]+ [ 0 - 9 ]* ) ) [ \s \t ]*</code>
NAMES	<code>[ \s \t ]* [ a - z A - Z ]+ ( ( -   _ ) * [ 0 - 9 ] * [ a - z A - Z ] * ) * [ \s \t ]*</code>
VARIABLES	<code>[ \s \t ]* \? ( [ 0 - 9 ] * [ a - z A - Z ] * ) + [ \s \t ]*</code>
STRINGS	<code>[ \s \t ]* " ( [ \\\\ \\n ]   [ ^ \\ \\n ] ) * " [ \s \t ]*</code>
COMMENTS	<code>;.*\n</code>
BREAKS	<code>[ \s ]+</code>

## Τι είναι το BISON:

Το Bison είναι μια γεννήτρια ανάλυσης γενικής χρήσης που μετατρέπει μια σχολιασμένη γραμματική χωρίς πλαίσιο σε ντετερμινιστικό LR ή γενικευμένο αναλυτή LR (GLR) που χρησιμοποιεί πίνακες ανάλυσης LALR. Το Bison είναι συμβατό με το Yacc: όλες οι σωστά γραμμένες γραμματικές Yacc πρέπει να λειτουργούν με το Bison χωρίς καμία αλλαγή.

Στην παρακάτω σχηματική απεικόνιση , βλέπουμε , πως συνδυάζεται το Flex με το Bison , καθώς και πιο είναι το αποτέλεσμα του συνδυασμού αυτού:



## Αρχείο Εισόδου (input.txt)

Παρακάτω παρουσιάζονται οι δοκιμές σε όλες τις δοθούσες δομές γραμματικής , για την σύνταξη προτάσεων της UniClips:

1. ;Comment Test.
- 2.
3. ;Events.
4. (color A is blue)
5. (cube A on table)
6. (color B is white)
7. (cube B on A)
8. (number\_of\_cubes 2)
9. (Extra bracket))
10. (color C is !)
- 11.
12. ;Deffacts.
13. (deffacts CUBES
14.     (color A is blue)
15.     (cube A on table)
16.     (color B is white)
17.     (cube B on A)
18.     (number\_of\_cubes 2))
- 19.
20. (deffacts wrong example
21.     (color A is blue)
22.     (cube A on table)
23.     (color B is white)
24.     (cube B on A)
25.     (number\_of\_cubes 2))
- 26.

27. ;Defrule.

28. (defrule move-up

29. (cube A on ?something)(cube B on A)(number\_of\_cubes ?num)

30. (test (= ?num 2))

31. ->

32. (printout t(?something " is under A") ("A is under B") ("there are 2 cubes"))(printout t1("hello")))

33.

34. ;Maths.

35. (- +3 -20 ?x1)

36. (- + 3 4)

37. (+ 4)

38. (\* ?id1 ?id2)

39. (/ 10 -2 -1)

40.

41.;Comp

42. (= ?num 2)

43. (= 2 ?num ?3-numbers)

44. (= 100 (+ ?num 2))

45. (= (+ ?num 2) ?x)

46.

47. ;Test

48. (test (= ?num 2))

49. (test =( ?num 2))

50.

51. (bind ?x (read))

52. (bind ?x 4)

53. (bind ?x (+ 1 ?num))

54. (bind 2 ?x)

55.

```

56. (deffacts static-facts
57.   ;;; food declarations
58.   (food-at 4 2)
59.   (food-at 5 2)
60. )
61.
62. (deffacts dynamic-facts
63.   ;;;pacman declaration
64.   (pacman-at 6 2)
65. )
66.
67. (defrule move-UP
68.   (pacman-at ?x ?y)
69.   (food-at ?z ?y)
70.   (test (= ?z (- ?x 1))))
71. ->
72   ;;; prints just a message
73   (printout t ("pacman has reached food"))
74. )
75.
76. ;No brackets.
77. ->
78. printout
79. test
80. 12345
81. (
82. )
83. ?var
84. nAmE-tEsT

```

85. (Κενή γραμμή στο τέλος του input.txt)

### Σχόλια και Αποτελέσματα του αρχείου input.txt

Τα αποτελέσματα που εμφανίζουν την ένδειξη correct σημαίνει ότι είναι σωστά καθώς ακολουθούν τους συντακτικούς κανόνες που έχουμε ορίσει. Επίσης βλέπουμε πώς αγνοούνται τα σχόλια όπως θέλουμε καθώς δεν επηρεάζουν τον κώδικα μας. Η γραμμή που εμφανίζεται στο είναι για

Line: 4 type: event. Correct.

Line: 5 type: event. Correct.

Line: 6 type: event. Correct.

Line: 7 type: event. Correct.

Line: 8 type: event. Correct.

Line: 9 type: ?. Error.

Η γραμμή 9 μας εμφανίζει error καθώς υπάρχει μια επιπλέον παρένθεση μετά το event.

Line: 10 type: event. Correct.

Η γραμμή 10 αν και σωστά συντακτικά περιέχει λέξη που δεν υπάρχει στα tokens μας οπότε αυξάνεται ο μετρητής λάθος λέξεων.

Line: 18 type: deffacts. Correct.

Line: 20 type: ?. Error.

Η γραμμή 20 μας εμφανίζει error καθώς υπάρχουν δύο ονόματα ορισμών και στοιχείων γεγονότων μετά το deffacts.

Line: 21 type: event. Correct.

Line: 22 type: event. Correct.

Line: 23 type: event. Correct.

Line: 24 type: event. Correct.

Line: 25 type: ?. Error.

Εφόσον διάβασε την γραμμή 20 σαν error στην συνέχεια οι γραμμές 21-24 διαβάζονται σωστά ως events και στην συνέχεια η γραμμή 25 έχει μία επιπλέον παρένθεση στο τέλος από την προσπάθεια να οριστεί το defrule οπότε η γραμμή είναι λάθος

Line: 32 type: defrule. Correct.

Να σημειωθεί ότι η defrule που ορίσαμε δέχεται πολλαπλές εντολές printout.

Line: 35 type: maths. Correct.



Line: 36 type: ?. Error.

Λάθος στην γραμμή 36 επειδή βάλαμε 2 τελεστές για πράξεις και δέχεται μόνο έναν.

Line: 37 type: ?. Error.

Λάθος στην γραμμή 37 επειδή υπάρχει μόνο ένα στοιχείο για να κάνουμε πράξη και απαιτούνται τουλάχιστον 2.

Line: 38 type: maths. Correct.

Line: 39 type: maths. Correct.

Line: 42 type: compare. Correct.

Line: 43 type: ?. Error.

Λάθος στην γραμμή 43 επειδή απαιτούνται αυτήρα 2 στοιχεία για σύγκριση και όχι 3 που βάλαμε.

Line: 44 type: compare. Correct.

Line: 45 type: compare. Correct.

Line: 48 type: test. Correct.

Line: 49 type: ?. Error.

Λάθος στην γραμμή 49 επειδή το “=” βρίσκεται εκτός το 2ο σετ παρενθέσεων και θα έπρεπε να είναι μέσα.

Line: 51 type: bind. Correct.

Line: 52 type: bind. Correct.

Line: 53 type: bind. Correct.

Line: 54 type: ?. Error.

Λάθος στην γραμμή 54 καθώς μετά το “bind” πρέπει να ακολουθεί μεταβλητή.

Line: 60 type: ?. Error.

Line: 65 type: ?. Error.

Line: 69 type: ?. Error.

Οι προσπάθειες στις γραμμές 56-69 να ορίσουμε deffacts και defrule βγάζουν error καθώς δεν βρήκαμε τρόπο να επιτρέπεται οι παρενθέσεις να τοποθετούνται σε διαφορετικές γραμμές καθώς και δεν βρήκαμε τρόπο να υπάρχουν τα events και το test όλα σε ξεχωριστές γραμμές. Οπότε ενώ οι παραπάνω γραμμές είναι θεωρητικά σωστές για την δική μας γραμματική θεωρούνται λάθος.

Line: 70 type: test. Correct.

Line: 71 type: ?. Error.

Λάθος στην γραμμή 71 επειδή το βέλος δεν μπορεί να υπάρξει μόνο του συντακτικά.

Line: 73 type: printout. Correct.

Line: 74 type: ?. Error.

Λάθος στην γραμμή 74 έχει απομείνει παρένθεση από την προσπάθεια ορισμού του defrule.

Line: 77 type: ?. Error.

Line: 78 type: ?. Error.

Line: 79 type: ?. Error.

Line: 80 type: ?. Error.

Line: 81 type: ?. Error.

Line: 82 type: ?. Error.

Line: 83 type: ?. Error.

Line: 84 type: ?. Error.

Στις γραμμες 77-84 υπάρχουν λέξεις αναγνωρισμένες οι οποίες όμως δεν είναι σωστά συντακτικά οπότε θα μετρηθούν ως λάθος εκφράσεις αλλά σωστές λέξεις.

Finished parsing.

Εδώ εμφανίζονται τα στατιστικά για την είσοδο μας.

Lines: 85

Correct words: 311

Correct syntax: 23

Wrong words: 1

Wrong syntax: 22

## Προβλήματα που προέκυψαν, πώς αντιμετωπίστηκαν:

Αρχικά είχαμε πρόβλημα στο διάβασμα των κρατημένων λέξεων καθώς το πρόγραμμα μας τις διάβαζε σαν NAMES. Το πρόβλημα λύθηκε αφαιρώντας γύρω από τα παλιά μας regex τις εκφράσεις που διαβάζουν το white space. Επίσης υπάρχει το πρόβλημα με τα deffacts και defrule που δεν μπορεί να έχει ελεύθερη μορφή (δηλαδή να επεκτείνεται είτε σε 1 είτε σε πολλές γραμμές) το οποίο με αναζήτηση στο βιβλίο του μαθήματος, στο eclass και σε πηγές στο διαδίκτυο δεν μπόρεσε να λυθεί παρά τις προσπάθειες της ομάδας.

## Συμπεράσματα:

Έπειτα από αρκετές προσπάθειες και δυσκολίες που αντιμετωπίσαμε στο συγκεκριμένο κομμάτι της άσκησης, καταφέραμε να έχουμε έναν λειτουργικό λεκτικό και συντακτικό αναλυτή. Αρχικά ελέγξαμε ξανά τον κώδικα flex μας και διορθώσαμε οποιες τυχόν regex εκφράσεις μας δημιουργούν κάποιο πρόβλημα, παρόλο που ήταν σωστές στο προηγούμενο κομμάτι της άσκησης. Έπειτα ξεκινήσαμε την συγγραφή του κώδικα bison με βάση τις σημειώσεις του εργαστηρίου. Τα αποτελέσματα που πήραμε από τις δοκιμές μας δείχνουν την λειτουργία του αναλυτή μας και δεν υπήρχε κανένα πρόβλημα στο compile. Όσον αφορά την συνεργασία της ομάδας, για άλλη μια φορά κύλησε ομαλά, με τον κάθε συμμετέχοντα να αναλαμβάνει από ένα τμήμα της εργασίας και στο τέλος, να γίνεται ολική επανάλυση ολόκληρης της εργασίας, ώστε να καταλάβουν όλα τα μέλη, όλα τα μέρη της εργασίας.

## Αρμοδιότητες των μελών της ομάδας:

ΗΛΙΑΣ ΔΗΜΑΣ: Συγγραφή κώδικα - δοκιμές για την είσοδο/έξοδο του κώδικα μας.

ΝΙΚΟΛΑΟΣ ΠΑΠΠΑΣ: Συγγραφή κώδικα - εγγράφου τεκμηρίωσης - δοκιμές για την είσοδο/έξοδο του κώδικα μας.

ΡΟΜΑΝΙΟΥΚ ΒΙΚΤΩΡ: Συγγραφή κώδικα και δοκιμές για την είσοδο/έξοδο του κώδικα μας.

ΤΖΩΡΤΖΗΣ ΑΓΓΕΛΟΣ: Συγγραφή κώδικα - εγγράφου τεκμηρίωσης - δοκιμές για την είσοδο/έξοδο του κώδικα μας.