**ΑΝΑΦΟΡΑ PROJECT ΑΡΧΩΝ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΚΑΙ ΜΕΤΑΦΡΑΣΤΩΝ**

*Μέλη:*

Ζαροκανέλλος Γεώργιος 1072495 3ο έτος up1072495@upnet.gr

Τριανταφυλλόπουλος Παναγιώτης 1072570 3ο έτος up1072570@upnet.gr

Σάμαρης Γεώργιος 1072493 3ο έτος up1072493@upnet.gr

**Εξήγηση γραμματικής BNF**

**Program**: τα 2 αρχεία που δοκιμάσαμε τον μεταγλωττιστή αποτελούνται από μέρη που περιέχουν μέσα πολλά στοιχεία. Συγκεκριμένα το last\_result αποτελείται από τα last και active που είναι χωρισμένα με κόμμα συνεπώς το πρώτο μέρος του κανόνα θα είναι «T\_L\_BRACE({), [last](#last)(κανόνας) T\_COMMA [active](#active)(κανόνας) T\_R\_BRACE(})». Αντιθέτως το range\_result αποτελείται μόνο από το order(χρησιμοποιείται για να ελεγχθεί η σειρά των στοιχείων θα αναλυθεί παρακάτω) συνεπώς το δεύτερο μέρος του κανόνα είναι «T\_L\_BRACE({) order(κανόνας) T\_R\_BRACE(})»

**expr**: Ως expr ορίζουμε τα στοιχεία που μπορούν να βρεθούν μέσα στο last, active και content των δύο αρχείων που δοκιμάζουμε. Όλα ακολουθούν το format «T\_QUOTATIONS T\_(το ανάλογο token) T\_QUOTATIONS T\_ASSIGNMENT και έπειτα το ανάλογο token σχετικά με το τι τύπου στοιχεία είναι».

**\*****last**: σε αυτόν τον κανόνα ορίζουμε το πρώτο μεγάλο μέρος του last\_result το οποίο στο αρχείο βρίσκεται στην μορφή **“last” :** (τα στοιχεία που περιέχει θα εξηγηθούν στο last\_scope) άρα έχουμε «Τ\_QUOTATIONS (“) T\_LAST Τ\_QUOTATIONS (“) T\_ASSIGNMENT(:) last scope(κανόνας)». Ωστόσο στο range\_result το στοιχείο last είναι διαφορετικά ορισμένο για αυτό και ο κανόνας έχει δεύτερο μέρος. Μέχρι το T\_ASSIGNMENT είναι ίδιος ο κανόνας απλά αλλάζει το τελευταίο token που στην συγκεκριμένη περίπτωση είναι T\_BOOLEAN διότι το last στο αρχείο range\_result παίρνει boolean τιμή.

**last\_scope**: αφορά τον ορισμό των εσωτερικών στοιχείων του last για το αρχείο last\_result . Όπως περιγράψαμε παραπάνω τα στοιχεία που έχει το last τα έχουμε ορίσει ως expr. Τo last αποτελείται από expr που χωρίζονται με κόμμα συνεπώς ο κανόνας θα πρέπει να καλεί τον εαυτό του ξανά κάθε φορά που αναγνωρίζει κάποιο expr ακολουθούμενο από κόμμα και να τερματίζει αν διαβάσει μόνο expr όπου θα είναι και το τελευταίο στοιχείο του last. Συνεπώς ο κανόνας είναι «last\_scope : [expr](#expression) T\_COMMA last\_scope | expr |expr error last\_scope»(το τελευταίο μέρος του κανόνα είναι για error recovery σε περίπτωση που λείπει κάποιο κόμμα)

**\*****active**: σε αυτόν τον κανόνα ορίζεται το 2ο μεγάλο μέρος του last\_result το οποίο στο αρχείο βρίσκεται στο αρχείο στην μορφή “active” : (active\_scope) άρα ο κανόνας θα είναι «Τ\_QUOTATIONS (“) T\_ACTIVE Τ\_QUOTATIONS (“) T\_ASSIGNMENT(:) active scope(κανόνας)».

**active\_scope**: Αφορά τον ορισμό των εσωτερικών στοιχείων του active. Όπως και το last έτσι και το active αποτελείται από expr χωρισμένες με κόμμα συνεπώς ο κανόνας θα είναι «active\_scope: [expr](#expression) T\_COMMA active\_scope | expr|expr error active\_scope;»(το τελευταίο μέρος του κανόνα είναι για error recovery σε περίπτωση που λείπει κάποιο κόμμα)

**\*u\_int\_array** : Οι πίνακες μη αρνητικών ακεραίων στα αρχεία που δοκιμάσαμε βρίσκεται στο αρχείο στην μορφή [u\_int\_scope] συνεπώς ο κανόνας θα είναι «T\_L\_BRACKET([) u\_int\_scope T\_R\_BRACKET(])»

**u\_int\_scope** : Οι πίνακες μη αρνητικών ακεραίων περιέχουν ακεραίους χωρισμένους με κόμμα συνεπώς ο κανόνας θα είναι «T\_U\_INT T\_COMMA(,) u\_int\_scope | T\_U\_INT | T\_U\_INT error u\_int\_scope»(ομοίως με τους παραπάνω κανόνες το τελευταίο μέρος του κανόνα είναι για error recovery στην έλλειψη κόμματος)

**json\_string**: τo json\_string βρίσκεται στο αρχείο στην μορφή “(όνομα του string)”. Επιπλέον θέλουμε το string active να το περάσουμε σαν τιμή στην μεταβλητή status συνεπώς ο κανόνας θα είναι «T\_QUOTATIONS T\_JSON\_STRING T\_QUOTATIONS | T\_QUOTATIONS T\_ACTIVE T\_QUOTATIONS | error».

**\*json\_object**: το json αντικείμενο περιέχει πολλά στοιχεία μέσα του και βρίσκεται στο αρχείο στην μορφή ({) (json\_obj\_scope) (}) συνεπώς ο κανόνας θα είναι «T\_L\_BRACE({) json\_obj\_scope T\_R\_BRACE(}) | error»

**Json\_obj\_scope:** το json αντικείμενο περιέχει expressions χωρισμένα με κόμμα συνεπώς ο κανόνας θα είναι «[expr](#expression) T\_COMMA json\_obj\_scope | expr | expr error json\_obj\_scope» (ομοίως με τους παραπάνω κανόνες το τελευταίο μέρος του κανόνα είναι για error recovery στην έλλειψη κόμματος)

**\*json\_array:** το json array είναι της μορφής ([) (json αντικείμενα) (]) συνεπώς ο κανόνας θα είναι «T\_L\_BRACKET([) json\_arr\_scope T\_R\_BRACKET(]) | error»

**Json\_arr\_scope**: το json array αποτελείται από json objects χωρισμένα με κόμμα συνεπώς ο κανόνας θα είναι : «json\_object T\_COMMA json\_arr\_scope | json\_object | json\_object error json\_arr\_scope(ομοίως με πριν το τελευταίο μέρος του κανόνα είναι για error recovery)»

**\*Content**: το content βρίσκεται στο αρχείο στην μορφή “content”: [json obj] άρα ο κανόνας θα είναι «T\_QUOTATIONS(“) T\_CONTENT T\_QUOTATIONS(“) T\_ASSIGNMENT(:) T\_L\_BRACKET([) content\_scope T\_R\_BRACKET(])»

**content\_scope:** το content περιέχει json objects χωρισμένα με κόμμα συνεπώς ο κανόνας θα είναι «json\_object T\_COMMA content\_scope | json\_object | json\_object error content\_scope(ομοίως το τελευταίο μέρος είναι για error recovery)»

**wager\_objects:** το wager\_objects περιλαμβάνει τα στοιχεία που περιέχονται στο wager\_statistics δηλαδή τα columns, wagers και add\_on χωρισμένα με κόμμα και σε αυτήν την σειρά. Συνεπώς ο κανόνας θα είναι «columns T\_COMMA wager\_objects | wagers T\_COMMA wager\_objects | add\_on | error». Τα columns, wagers και add on τα ορίσαμε όλα σε διαφορετικούς κανόνες διότι μας ενδιαφέρει η σειρά

**columns:** το columns είναι θετικός ακέραιος και βρίσκεται στο αρχείο στην μορφή “columns”:(u\_int) συνεπώς ο κανόνας θα είναι «T\_QUOTATIONS T\_COLUMNS T\_QUOTATIONS T\_ASSIGNMENT T\_U\_INT | error»

**wagers:** το wagers είναι και αυτό θετικός ακέραιος και βρίσκεται στο αρχείο στην μορφή “wagers”(u\_int) συνεπώς θα είναι «T\_QUOTATIONS T\_WAGERS T\_QUOTATIONS T\_ASSIGNMENT T\_U\_INT»

**add\_on:** το add on είναι είναι json array

**price\_points:** περιγράφει το τι περιλαμβάνει το pricePoints το οποίο είναι το στοιχείο amount το οποίο είναι θετικός πραγματικός και βρίσκεται στο αρχείο στην μορφή {“amount”(u\_float)} συνεπώς ο κανόνας θα είναι «T\_L\_BRACE Τ\_QUOTATIONS T\_AMOUNT T\_QUOTATIONS T\_ASSIGNMENT T\_U\_FLOAT T\_R\_BRACE | error»

**\*winning\_numbers:** το winning numbers περιέχει το εσωτερικά στοιχεία του winningNumbers(που έχουμε ορίσει σαν [expr](#expression)) τα οποία είναι τα list και bonus. Το εσωτερικό του winningNumbers είναι στην μορφή {list(κανόνας),bonus(κανόνας)} συνεπώς ο κανόνας θα είναι «T\_L\_BRACE({) list T\_COMMA(,) bonus T\_R\_BRACE(})»

**list:** το List είναι το ένα από τα δύο στοιχεία που περιέχει το winningNumbers και είναι array που περιέχει θετικούς ακεραίους και βρίσκεται στο αρχείο στην μορφή “list”:[u\_int, u\_int..] συνεπώς ο κανόνας θα είναι «T\_QUOTATIONS(“) T\_LIST T\_QUOTATIONS(“) T\_ASSIGNMENT(:) u\_int\_array»

**bonus:** το bonus είναι το δεύτερο στοιχείο που περιέχει το winningNumbers και είναι επίσης array που περιέχει ένα θετικό ακέραιο και βρίσκεται στο αρχείο στην μορφή “bonus” : [u\_int] συνεπώς ο κανόνας θα είναι «T\_QUOTATIONS(“) T\_BONUS T\_QUOTATIONS(“) T\_ASSIGNMENT(:) u\_int\_array»

Τα **total\_pages, total\_elements, number\_of\_elements, sort , first, size** και **number** ακολουθούν όλα το ίδιο format το οποίο είναι : “(totalPages|totalElements|…|size)” : (αντίστοιχη τιμή κάθε στοιχείου) συνεπώς οι κανόνες τους ακολουθούν την μορφή «T\_QUOTATIONS(“) αντίστοιχο token T\_QUOTATIONS(“) T\_ASSIGNMENT(:) αντίστοιχο token για τύπο τιμής» *(Τα στοιχεία αυτά παρόλο που έχουν το ίδιο format με τα* [*expr*](#expression) *επιλέξαμε να μην τα βάλουμε εκεί διότι χρειαζόμαστε να υπάρχουν με κάποια σειρά στο αρχείο)*

**\*(Τα στοιχεία αυτά επιλέξαμε να χρησιμοποιήσουμε 2 κανόνες για να τα περιγράψουμε, έναν κανόνα για το ίδιο το στοιχείο και έναν για τα στοιχεία που περιέχει στο εσωτερικό του και αυτό για να εξαλείψουμε τα shift/reduce και reduce/reduce error)**