



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

**Προηγμένα Θέματα Γλωσσών Προγραμματισμού**  
**Διδάσκων: Χρήστος Γκουμόπουλος**  
**Εργαστηριακοί Συνεργάτες: Γιώργος Χρυσολωράς, Αλέξιος Γκίκας**

**3η Ατομική Εργασία**  
**Ημερομηνία παράδοσης μέχρι: 11/05/2020 23:59**

**Οδηγίες για την παράδοση της εργασίας**

Η παράδοση των εργασιών θα πρέπει να γίνει ηλεκτρονικά μέσω του περιβάλλοντος ηλεκτρονικής μάθησης e-class (<https://eclass.icsd.aegean.gr/modules/work/?course=ICSD410>) μέχρι την **Δευτέρα 11/05/2020 στις 23:59**. Εκπρόθεσμη παράδοση της εργασίας μέχρι τις 15/05/2020 23:59, επιφέρει απώλεια 25% του βαθμού. Μετά την παραπάνω ημερομηνία, η εργασία δεν γίνεται δεκτή.

Θα πρέπει να στείλετε ένα συμπιεσμένο αρχείο **.zip** με όνομα **login\_name\_atpl03.zip** (π.χ. icsd18255\_atpl03.zip) το οποίο να περιέχει:

- Φάκελο με τα αρχεία pythοn του πηγαίου κώδικα της άσκησης 1.
- Ένα αρχείο pdf με όνομα **login\_name\_atpl03.pdf** με τις απαντήσεις σας στην άσκηση 2 και στα 3 πρώτα ερωτήματα της άσκησης 3.
- Φάκελο με όλα τα αρχεία που απαρτίζουν τον λεκτικό και συντακτικό αναλυτή (4ο ερώτημα της άσκησης 3).

Καμία εργασία δε θα γίνεται δεκτή εάν η παράδοσή της δεν γίνεται σύμφωνα με τις παραπάνω οδηγίες.

Η αντιγραφή πηγαίου κώδικα απαγορεύεται και θα οδηγεί σε μηδενισμό ολόκληρης της εργασίας και όλων των συμμετεχόντων σε αυτή.



## 1<sup>η</sup> άσκηση (30%)

Για την παρακολούθηση της ασθένειας COVID-19, ζητείται να υλοποιηθεί ένα πρόγραμμα σε Python 3 με αντικειμενοστρεφή σχεδιασμό.

Μας ενδιαφέρει για μια χώρα να καταγράφεται το όνομά της, μια λίστα με τα γεωγραφικά διαμερίσματά της ή τις πολιτείες της και μια λίστα η οποία περιέχει τα κρούσματα ασθένειας COVID-19 για κάθε μέρα αρχής γενομένου από 22/01/2020.

Πηγή δεδομένων είναι το αποθετήριο του Πανεπιστημίου John Hopkins - Center for Systems Science and Engineering (JHU CSSE) στο GitHub:

[https://github.com/CSSEGISandData/COVID-19/blob/master/csse\\_covid\\_19\\_data/csse\\_covid\\_19\\_time\\_series/time\\_series\\_covid19\\_confirmed\\_global.csv](https://github.com/CSSEGISandData/COVID-19/blob/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv)

Τα δεδομένα είναι σε comma-separated values (CSV) αρχείο στο οποίο κάθε στήλη χωρίζεται με ','.

Το πρωτογενές αρχείο το οποίο θα χρησιμοποιήσετε για να τραβήξετε δεδομένα βρίσκεται στο: [https://github.com/CSSEGISandData/COVID-19/raw/master/csse\\_covid\\_19\\_data/csse\\_covid\\_19\\_time\\_series/time\\_series\\_covid19\\_confirmed\\_global.csv](https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv)

Στο αρχείο καταγραφής, η πρώτη γραμμή είναι οι επικεφαλίδες ως εξής: Province/State, Country/Region, Lat, Long, 1/22/20, 1/23/20, ... μέχρι την τελευταία ημέρα που έχει καταχωρηθεί. Κάθε επόμενη μέρα ανάκτησης του αρχείου, έχει προστεθεί και μια νέα στήλη με τις καταγραφές τις προηγούμενης ημέρας.

Από την δεύτερη γραμμή έως στο τέλος του αρχείου είναι οι τιμές των δεδομένων. Αν υπάρχουν χαρακτήρες στην 1η στήλη, Province/State, σημαίνει ότι τα στοιχεία συλλέγονται από διάφορες περιοχές της χώρας (2η στήλη) και κατά πάσα πιθανότητα υπάρχουν και άλλες γραμμές για την ίδια χώρα. Οι συντεταγμένες Lat/Long δε μας ενδιαφέρουν. Οι ημερομηνίες που αναγράφονται είναι της μορφής MM/DD/YY (μήνας/ημέρα/έτος).

Επίσης, υπάρχουν τουλάχιστον 2 περιπτώσεις όπου περιοχές ή χώρες έχουν τον χαρακτήρα ',' στην ονομασία τους. Για να τις ξεχωρίζουμε, είναι μέσα σε διπλά εισαγωγικά. Για παράδειγμα η χώρα "Korea, South" και η περιοχή "Bonaire, Sint Eustatius and Saba". Μπορούμε να αντικαταστήσουμε μια γραμμή που περιέχει το παραπάνω σχέδιο με την συνάρτηση `re.sub()` της `python`. Για παράδειγμα αν θέλουμε να αντικαταστήσουμε σε μια γραμμή του αρχείου `line` το ',' με ' - ' μπορούμε να χρησιμοποιήσουμε την παρακάτω κλήση:

```
re.sub(r'"([\w\s]+),([\w\s]+)"', r'\1 -\2', line)
```

Δείτε πληροφορίες για τη συνάρτηση [εδώ](#) και [εδώ](#).

Η κλάση να περιέχει μια λίστα με το πλήθος κρουσμάτων για κάθε ημέρα αρχής γενομένης αυτής που ξεκινάει και το αρχείο, 1/22/20 (22/01/2020).

Το πρόγραμμα απαιτείται να:

A1. Ανακτά το αρχείο από την διεύθυνση:

[https://github.com/CSSEGISandData/COVID-19/raw/master/csse\\_covid\\_19\\_data/csse\\_covid\\_19\\_time\\_series/time\\_series\\_covid19\\_confirmed\\_global.csv](https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv)



A2. Να δημιουργεί τόσα αντικείμενα χωρών, όσα και οι χώρες του αρχείου. Μπορείτε να χρησιμοποιήσετε μια λίστα από αντικείμενα ή κάποια άλλη δομή της επιλογή σας για το σκοπό αυτό.

A3. Να δημιουργεί για κάθε αντικείμενο χώρας τα δεδομένα μέλη που απαιτούνται.

A4α. Ανάλογα με την προσέγγισή σας, θα πρέπει να καθαρίζονται και οι ονομασίες περιοχών και χωρών που περιέχουν ' , ' ανάμεσα σε διπλά εισαγωγικά.

A4β. Μπορείτε να χρησιμοποιήσετε και το [module csv](#) αντί να καθαρίσετε τα δεδομένα. Θα χρειαστεί να μελετήσετε τις λειτουργίες του.

A5. Να υλοποιήσετε επιπλέον μεθόδους κλάσης για λειτουργίες που θα απαιτηθούν κατά την δημιουργία ενός αντικειμένου χώρας καθώς και συναρτήσεις εκτός κλάσης που μπορεί να χρησιμεύσουν.

A6. Όταν γίνεται ανάγνωση γραμμής που αφορά περιοχή μιας χώρας, να προστίθεται η περιοχή στην λίστα με τις περιοχές της χώρας και τα στοιχεία κρουσμάτων ανά μήνα αντίστοιχα, σε υπάρχον αντικείμενο χώρας. Αν δεν υπάρχει αντικείμενο της συγκεκριμένης χώρας, να δημιουργείται.

Εφόσον φορτωθούν τα δεδομένα σε αντικείμενα της κλάσης, το πρόγραμμα να υποστηρίζει τις παρακάτω λειτουργίες (ορίστε επιλογές για τον χρήστη με δική σας κωδικοποίηση):

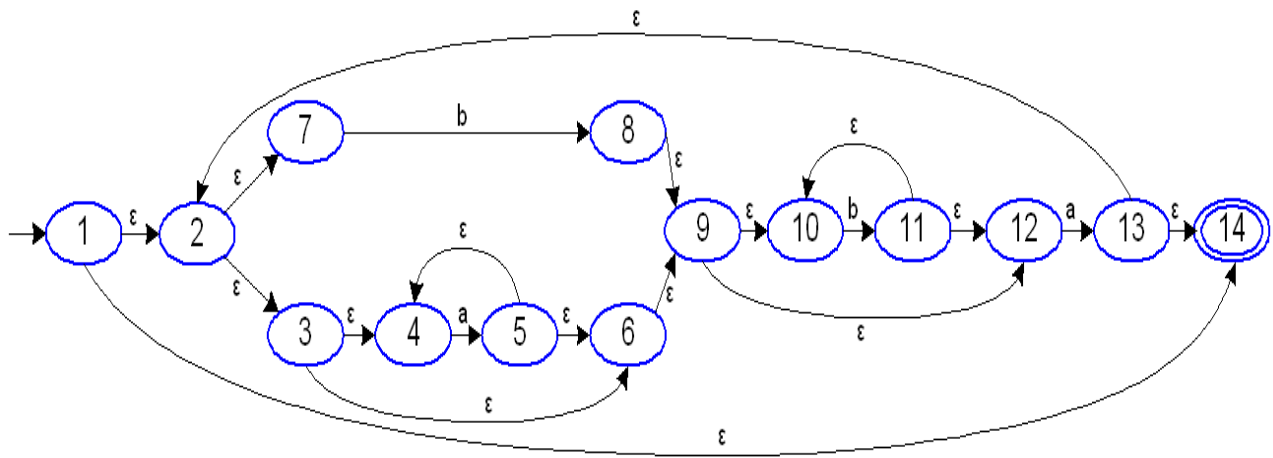
1. Εμφάνιση συνολικών κρουσμάτων για μια χώρα που πληκτρολογεί ο χρήστης.
2. Εμφάνιση λίστας χωρών και κρουσμάτων για μια συγκεκριμένη ημερομηνία που πληκτρολογεί ο χρήστης.
3. Εμφάνιση χώρας που αυτή τη στιγμή έχει τα περισσότερα κρούσματα.



## 2<sup>η</sup> άσκηση (10%)

Να μετατρέψετε το ακόλουθο NFA σε ένα ισοδύναμο DFA χρησιμοποιώντας τον αλγόριθμο subset construction και ορίζοντας τα κατάλληλα σύνολα καταστάσεων. Θα πρέπει να δείξετε τους ενδιάμεσους πίνακες που σχετίζονται με την εφαρμογή του αλγόριθμου και το σύνολο καταστάσεων που αντιστοιχεί σε κάθε κατάσταση του DFA.

Μπορείτε να βοηθηθείτε από το παράδειγμα που περιέχεται στην παρουσίαση 02\_ATPL\_Lexical\_Analysis της θεωρίας (διαφάνειες 84-86).





### **3<sup>η</sup> άσκηση (60%)**

Δίνεται η παρακάτω γραμματική. Αρχικό σύμβολο της γραμματικής είναι το *Expression*. Μη τερματικά είναι τα σύμβολα σε italics.

*Expression* → ID | Access | Create  
*Access* → *Expression* [ *Expression* ]  
*Create* → NEW ID List  
*List* → Dimension List | ε  
*Dimension* → [ *Expression* ]

#### **Ερώτημα 1 (10%)**

Δείξτε ότι η παραπάνω γραμματική είναι διαφορούμενη. Για κάθε παραγωγή που θα χρησιμοποιήσετε να σχεδιάσετε το αντίστοιχο δένδρο ανίχνευσης. Να χρησιμοποιήσετε για το σκοπό αυτό τη συμβολοσειρά: **NEW ID [ ID ]**

#### **Ερώτημα 2 (5%)**

Μια ισοδύναμη γραμματική που δεν είναι διαφορούμενη είναι η παρακάτω:

*Expression* → Access | Create  
*Access* → ID List  
*Create* → NEW ID List  
*List* → Dimension List | ε  
*Dimension* → [ *Expression* ]

Να σχεδιάσετε το μοναδικό δένδρο ανίχνευσης που παράγει τη συμβολοσειρά που χρησιμοποιήσατε στο ερώτημα 1.

#### **Ερώτημα 3 (15%)**

Να εξετάσετε αν η μη-διαφορούμενη γραμματική που προέκυψε στο ερώτημα 2 είναι κατάλληλη για top-down ανίχνευση. Αν δεν είναι να κάνετε τους κατάλληλους μετασχηματισμούς ώστε να την μετασχηματίσετε σε μορφή κατάλληλη για top-down ανίχνευση.

Υπόδειξη: Εφαρμόστε τον γενικό αλγόριθμο απαλοιφής αριστερής αναδρομής (δείτε σχετικά την παρουσίαση 05 Τεχνικές Συντακτικής Ανάλυσης (Μέρος I), slide 23).

#### **Ερώτημα 4 (30%)**

Για τη γραμματική που θα προκύψει από το ερώτημα 3 να κατασκευάσετε έναν συντακτικό αναλυτή που υλοποιεί ανίχνευση μιας πρότασης που δίνεται ως είσοδος με την τεχνική της Top-Down Ανίχνευσης με Οπισθοδρόμηση (η δομή μιας τέτοιας υλοποίησης περιγράφεται στην παρουσίαση της θεωρίας 05\_ATPL\_Syntax\_Analysis\_Techniques στην ενότητα «Παράδειγμα Υλοποίησης Top-Down Ανίχνευσης με Οπισθοδρόμηση»). Ο συντακτικός αναλυτής θα δέχεται την είσοδο προς αναγνώριση ως όρισμα κατά την εκτέλεσή του. Να ονομάσετε τη main class του συντακτικού αναλυτή με το όνομα **ExpressionChecker**. Ο συντακτικός αναλυτής θα πρέπει να διαθέτει μια ρουτίνα/μέθοδο error() η οποία θα αναφέρει όταν καλείται ότι η είσοδος δεν ακολουθεί την πέπουσα δομή ή ότι εντοπίστηκε μη αναγνωρίσιμο token και θα τερματίζει τον συντακτικό αναλυτή.

Ο συντακτικός αναλυτής κάθε φορά που αναγνωρίζει ένα token στη συμβολοσειρά εισόδου θα τυπώνει σε νέα γραμμή το token που αναγνωρίστηκε. Για παράδειγμα:



Token NEW recognized

Token ID recognized

Token [ recognized

...

Για τον έλεγχο του συντακτικού αναλυτή σας δίνονται οι ακόλουθες συμβολοσειρές:

1. ID
2. ID [ ID ] [ ID ]
3. ID [ NEW ID ] [ NEW ID ]
4. NEW ID [ NEW ID [NEW ID]]
5. NEW ID [ ID [NEW ID]]
6. ID NEW ID
7. NEW ID ID
8. ID [ID, ID]

Για την υλοποίηση του συντακτικού αναλυτή μπορείτε να χρησιμοποιήσετε μια γλώσσα προγραμματισμού που έχετε διδαχθεί μέχρι το Δ' εξάμηνο, C, C++, Java, Python.

Να μην χρησιμοποιηθεί εργαλείο παραγωγής συντακτικού αναλυτή (π.χ. ANTLR). Μπορείτε να χρησιμοποιήσετε το ANTLR **μόνο** για αναγνώριση tokens (λεκτική ανάλυση).

Να επισυνάψετε screenshots για την επιτυχή ή όχι αναγνώριση κάθε μιας από τις παραπάνω συμβολοσειρές ως συνέπεια της εκτέλεσης του συντακτικού αναλυτή.

**ΚΑΛΗ ΕΠΙΤΥΧΙΑ!**