

ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ

Ασκήσεις

2019-2020

Κανονικές Εκφράσεις και FLEX / BISON Συντακτική Ανάλυση

Ποσοστό στη τελική βαθμολογία: 10% (1 Μονάδα)

1. A NetLogo like Language

Έστω η ακόλουθη γραμματική, η οποία αποτελεί μέρος της γραμματικής μιας γλώσσας:

```
PDefinitions ::= PDef PDefinitions | ε
PDef ::= to ProcFunN Body "end" | "to-report" ProcFunN Body "end"
ProcFunN ::= "name" | "name" "[" Vars "]"
Vars ::= "var" | Vars "var"
Body ::= "if" Call "[" Body "]" | "report" "var" | "report" "num" |
        Call | "var" "=" "num"
Call ::= "name" | "name" Args
Args ::= "var" | "num" | Args "var" | Args "num"
```

Όπου μπορείτε να υποθέσετε την ύπαρξη ενός λεκτικού αναλυτή, ο οποίος επιστρέφει όλα τα keywords ("to", "to-report", κλπ) και τα σύμβολα "num" "var" και "name".

Οι περιγραφές των λεκτικών μονάδων, εκτός εκείνων που είναι προφανείς από την παραπάνω γραμματική, είναι οι ακόλουθες

- **"name"** οποιαδήποτε συμβολοσειρά πεζών χαρακτήρων
- **"var"** οποιαδήποτε συμβολοσειρά γραμμάτων (πεζών ή κεφαλαίων) η οποία ξεκινά με "?"
- **"num"** οποιοσδήποτε ακέραιος (ακολουθία ψηφίων).

Να μετατρέψετε την γραμματική σε LL, και να υλοποιήσετε τον LL συντακτικό αναλυτή χρησιμοποιώντας το εργαλείο Flex. Να αιτιολογήσετε τα βήματά σας, δίνοντας όλη την διαδικασία (μετατροπή σε LL, εύρεση FIRST/FOLLOW, σχολιασμένα παραδείγματα υλοποίησης 2 κανόνων της γραμματικής). Στο corpus θα βρείτε τα απαραίτητα αρχεία για να δοκιμάσετε το συντακτικό σας αναλυτή. Σε περίπτωση σφάλματος ο συντακτικός αναλυτής απλώς σταματά ενημερώνοντας για το σφάλμα. Για παράδειγμα:

```
./netlogo test1
Success!
```

```
./netlogo test1_error
```

```
Parsing Rule Match Failed Syntax Error reading ?e at line 3 of input.
```

2. AgentSpeak

Να γραφεί πρόγραμμα σε FLEX/BISON το οποίο υλοποιεί ένα απλό συντακτικό αναλυτή για την μια γλώσσα προγραμματισμού πρακτόρων. Η γραμματική της γλώσσας φαίνεται παρακάτω:

```

agent ::= beliefs plans
beliefs ::= beliefs belief | ε
belief ::= predicate "."
predicate ::= Atom "(" terms ")"
plans ::= plans plan | ε
plan ::= triggering_event ":" context "<-" body "."
triggering_event ::= "+" predicate | "-" predicate
                | "+" goal | "-" goal
context ::= "true" | cliterals
cliterals ::= literal | literal "&" cliterals
literal ::= predicate | "not" "(" predicate ")"
goal ::= "!" predicate | "?" predicate
body ::= "true" | actions
actions ::= action | action ";" actions
action ::= predicate | goal | belief_update
belief_update ::= "+" predicate | "-" predicate

terms ::= term | term "," terms
term ::= Var | Atom | Number | Atom "(" terms ")"

```

Ο ορισμός των λεκτικών μονάδων είναι εκείνος της Prolog.

- **Var** : (μεταβλητή) ακολουθία χαρακτήρων, αριθμών ή `_`, που ξεκινά με Κεφαλαίο χαρακτήρα.
- **Atom**: (άτομο) ακολουθία χαρακτήρων, αριθμών ή `_`, που ξεκινά με πεζό χαρακτήρα.
- **Num**: αριθμός ακέραιος ή υποδιαστολής (απλή μορφή).

Ο συντακτικός αναλυτής που θα υλοποιήσετε θα πρέπει να μπορεί να ανιχνεύσει αν ένα πρόγραμμα της γλώσσας είναι συντακτικά ορθό. Παραδείγματα αρχείων στη γλώσσα θα βρείτε στο compus (test1 και test2). Στον ίδιο κατάλογο θα βρείτε και το αρχείο test_error το οποίο έχει συντακτικά σφάλματα. Το όνομα του παραγόμενου εκτελέσιμου αρχείου συντακτικού αναλυτή θα είναι agentSpeak (.exe αν χρησιμοποιήσετε windows). Το αποτέλεσμα στα τρία αρχεία είναι το ακόλουθο:

```

agentSpeak test1
Syntax OK!

```

```

agentSpeak test2
Syntax OK!

```

```

./agentSpeak test_error
Error(line 3) : syntax error, unexpected ';'
Error(line 5) : syntax error, unexpected '&', expecting '.'
There were 2 errors in code. Failure!

```

ΠΑΡΑΔΟΣΗ

Θα πρέπει να παραδώσετε ένα αρχείο coursework1_2020.zip, που θα περιέχει:

- Μια μικρή αναφορά σε μορφή pdf (**report.pdf**), σχολιασμένο τον κώδικα (θα περιλάβετε και τον κώδικα) που αφορά τα εργαλεία FLEX/BISON. **ΣΤΗΝ ΠΡΩΤΗ ΣΕΛΙΔΑ Η ΑΝΑΦΟΡΑ ΣΑΣ ΝΑ ΠΕΡΙΕΧΕΙ ΤΟ ΟΝΟΜΑ ΚΑΙ ΤΟΝ ΑΜ ΣΑΣ.**
- Τα αρχεία που αντιστοιχούν στις λύσεις των ασκήσεων, **οργανωμένα σε καταλόγους όπως στο compus.**
- **ΟΛΑ ΤΑ ΟΝΟΜΑΤΑ ΤΩΝ ΑΡΧΕΙΩΝ ΠΟΥ ΘΑ ΠΑΡΑΔΟΣΕΤΕ ΝΑ ΕΙΝΑΙ ΜΕ ΑΓΓΛΙΚΟΥΣ ΧΑΡΑΚΤΗΡΕΣ.**

Σημειώσεις:

- Οι χαρακτήρες νέας γραμμής διαφέρουν από unix σε windows. Για να μην έχετε προβλήματα που θα σας ταλανίσουν πολύ, δώστε και χρησιμοποιήστε το ακόλουθο κανονικό ορισμό στο αρχείο flex που θα δημιουργήσετε:

```
newline \n|\x0A|\x0D\x0A
```

- Το FLEX σε περίπτωση που δεν ταυτοποιήσει με ένα από τους διαθέσιμους κανόνες ένα μέρος της εισόδου, τότε το "αντιγράφει" (τυπώνει) στην έξοδο. Αν θέλετε να αποφύγετε κάτι τέτοιο θα πρέπει να βάλετε στο αρχείο σας στο τέλος των κανόνων που θα περιγράψετε τους ακόλουθους δύο κανόνες:

```
. { /* Do Nothing */}  Η κάποιο μήνυμα σφάλματος  
{newline} { /* Do nothing */} Η αύξηση του αριθμού γραμμής
```

Στον πρώτο κανόνα ο χαρακτήρας "." (τελεία) ταυτοποιείται με οποιονδήποτε χαρακτήρα εκτός εκείνου της νέας γραμμής. Το "{newline}" ταυτοποιείται με τον χαρακτήρα της νέας γραμμής, όπως περιγράφεται παραπάνω.

- Στο σύστημα compus, θα βρείτε χρήσιμα αρχεία για την υλοποίηση της άσκησης.
- **Καλή επιτυχία!**