

Μεταγλωττιστές 2019

Προγραμματιστική Εργασία #2

(Προσοχή: η παράδοση της άσκησης θα γίνει μέσω *github*. Διαβάστε τις οδηγίες στο τέλος της εκφώνησης)

Ζητούμενο

Ο στόχος της άσκησης είναι να κατασκευάσετε top-down συντακτικό αναλυτή με τη μέθοδο της αναδρομικής κατάβασης, ο οποίος θα αναγνωρίζει εντολές (statements) για την **ανάθεση σε μεταβλητές** και **εκτύπωση** δυαδικών αριθμητικών εκφράσεων που θα περιέχουν:

- δυαδικές αριθμητικές τιμές (σειρές από **0** και **1**)
- και αναγνωριστικά ονόματα μεταβλητών
- σε συνδυασμούς με τελεστές δυαδικής λογικής σε επίπεδο bit : **and**, **or** και **xor** (ακριβώς αυτά τα keywords).
- Επιπλέον οι εκφράσεις θα μπορούν να συνδυαστούν με παρενθέσεις, σε οποιοδήποτε βάθος.

Διαδικασία

α) Σχεδιάστε τη ζητούμενη γραμματική. Βασιστείτε στη γραμματική των αριθμητικών εκφράσεων που χρησιμοποιήθηκε στο εργαστήριο:

<http://mixstef.github.io/courses/compilers/lecturedoc/unit5/module2.html#id1>

Θυμηθείτε ότι η σειρά προτεραιότητας των τελεστών πράξεων σε επίπεδο bits (από την υψηλότερη προς τη χαμηλότερη) είναι: and, or, xor. Συνεπώς θα πρέπει να έχετε 3 επίπεδα κανόνων (π.χ. Term, Factor, Atom και τα αντίστοιχα `_tail`) αντί για τα 2 της αρχικής γραμματικής.

β) **Βεβαιωθείτε ότι η γραμματική σας είναι LL(1)!** Μπορείτε να χρησιμοποιήσετε το on-line εργαλείο για να ελέγξετε τη συμβατότητα LL(1) και να βρείτε τα FIRST και FOLLOW sets στη διεύθυνση:

<http://smlweb.cpsc.ucalgary.ca/start.html>

γ) Κατασκευάστε πρόγραμμα συντακτικού αναλυτή, **αναγνωριστή της γλώσσας**, σύμφωνα με τη μέθοδο της αναδρομικής κατάβασης σε Python3, χρησιμοποιώντας το module Plex. Το πρόγραμμα αυτό θα αναγνωρίζει χωρίς μήνυμα λάθους κάθε συντακτικά σωστή είσοδο.

Μπορείτε να χρησιμοποιήσετε ως βάση τον κώδικα που έχετε ήδη έτοιμο από το εργαστήριο ή το εξής παράδειγμα:

<https://gist.github.com/mixstef/946fce67f49f147991719bfa4d0101fa>

Θυμηθείτε να τροποποιήσετε κατάλληλα το λεξικό του Plex για να αναγνωρίζετε δυαδικούς αριθμούς (σειρές από 0 και 1).

δ) Σε δεύτερο αντίγραφο του προγράμματός σας προσθέστε εντολές έτσι ώστε να **διερμηνεύετε** τον πηγαίο κώδικα του αρχείου εισόδου (να εκτελείτε τις αναθέσεις και εκτυπώσεις).

Υποδείξεις:

- Χρησιμοποιήστε ακεραίους για τις πράξεις και την αποθήκευση των τιμών των μεταβλητών.
- Μετατρέψτε ένα δυαδικό string σε ακέραιο ως εξής:

```
a = int('1010',2)
```

- Τυπώστε έναν αριθμό σε δυαδική μορφή ως εξής:

```
print('{:b}'.format(a))
```

ε) Δοκιμάστε το πρόγραμμά σας με διάφορους συνδυασμούς εισόδου (αποδεκτούς ή όχι) και αποθηκεύστε την είσοδο και έξοδο της κονσόλας (log εκτέλεσης).

Παραδοτέο

Η παράδοση θα γίνει μέσω github. Οδηγίες:

1. Αντιγράψτε (**fork**) το repository <https://github.com/mixstef/compiler1819a2> στο δικό σας repository. Βεβαιωθείτε ότι δουλεύετε αποκλειστικά στο **master branch**.
2. Τροποποιήστε κατάλληλα τα αρχεία που περιέχονται στο repository σας με το δικό σας περιεχόμενο:
 - Συμπληρώστε τα στοιχεία σας στο αρχείο **README.md**.
 - Συμπληρώστε τον κώδικά σας στα αρχεία **parser.py** (για τον συντακτικό αναλυτή – αναγνωριστή της γλώσσας) και **runner.py** (για τον συντακτικό αναλυτή – διερμηνευτή).
 - Προσθέστε την αναφορά σας ως **report.pdf**.
 - **Προσοχή: πρέπει να διατηρήσετε τα ονόματα των παραπάνω αρχείων!**
3. Ενημερώστε το repository σας στο github εντός προθεσμίας. **Μην κάνετε pull request!**

Η εργασία είναι αυστηρά ατομική. Για την εγκυρότητα της υποβολής σας θα χρησιμοποιηθεί η χρονοσήμανση των αλλαγών (commits) των αρχείων σας.

Προθεσμία παράδοσης: Σάββατο 1/6/2019 23:59.