

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ, ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Ι, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2010-2011

ΕΡΓΑΣΙΑ ΕΞΑΜΗΝΟΥ

Ποιος πρέπει να ολοκληρώσει αυτή την εργασία?

Φοιτητές έτους ≥ 2 που δεν έχουν ολοκληρώσει επιτυχώς την εργασία του ακαδημαϊκού έτους 2009-2010.

Η εργασία μπορεί να γίνει είτε ατομικά είτε σε ομάδα των δύο ατόμων.

Προθεσμία

Παρασκευή, 7 Ιανουαρίου 2011, 23:59

Διαδικαστικά

Η προθεσμία παράδοσης είναι τελική - εκπρόθεσμες εργασίες δε γίνονται δεκτές.

Για να παραδώσετε την εργασία σας:

Δημιουργήστε ένα φάκελο με όνομα

επώνυμο_όνομα_AEM_email

ή

επώνυμο1_όνομα1_AEM1_email1_επώνυμο2_όνομα2_AEM2_email2

αν η εργασία έχει γίνει σε ομάδα.

Μέσα σε αυτό το φάκελο **αντιγράψτε τα αρχεία C που γράψατε**.

Δημιουργήστε ένα αρχείο ZIP ή TAR ή TGZ το οποίο περιέχει το φάκελλο που κατασκευάσατε.

Η παράδοση γίνεται με email στο ce120lab@gmail.com. Το email πρέπει να συντάσσεται ως εξής:

Θέμα (Subject): CE120 project

Πρέπει να γράψετε το θέμα ακριβώς με αυτό τον τρόπο για να περάσει από τα κατάλληλα φίλτρα.

CC: το email σας

Να στέλνετε πάντα αντίγραφο στον εαυτό σας και να το ελέγχετε για να επιβεβαιώνετε ότι έγιναν όλα σωστά.

Περιεχόμενο (Text): Το όνομα σας και το ΑΕΜ σας

Επισυνάψεις (attachments): Το ZIP ή TAR ή TGZ αρχείο που δημιουργήσατε

Σύντομη περιγραφή

Θα γράψετε ένα πρόγραμμα το οποίο, δεδομένης μιας γραμματικής χωρίς συμφραζόμενα, δημιουργεί μια τυχαία παραγωγή.

Εισαγωγική θεωρία

Μια γραμματική χωρίς συμφραζόμενα είναι ένα σύστημα που περιγράφει μια γλώσσα προσδιορίζοντας πώς μπορεί να παραχθεί μια νόμιμη “πρόταση” αυτής της γλώσσας. Μια γραμματική χωρίς συμφραζόμενα αποτελείται από :

- Μια συλλογή από σύμβολα που λέγονται *τερματικά*, τα οποία είναι οι βασικές μονάδες (“λέξεις”) της γλώσσας.
- Μια συλλογή από σύμβολα που λέγονται *μη-τερματικά*, τα οποία είναι οι συντακτικές μονάδες της γλώσσας. Ένα μη-τερματικό αναπαριστά μια ομάδα από τερματικά και μη-τερματικά σύμβολα.
- Μια συλλογή από κανόνες παραγωγής που περιγράφουν τους συντακτικούς κανόνες της γλώσσας. Ένας κανόνας παραγωγής εκφράζει πώς μπορεί ένα μη-τερματικό να αναπτυχθεί σε μια σειρά από τερματικά και μη τερματικά σύμβολα.
- Ένα ειδικό σύμβολο που λέγεται *αρχικό*, και είναι το σημείο εκκίνησης για την παραγωγή μιας πρότασης της γλώσσας.

Μια σειρά από κανόνες που δείχνουν πώς μπορεί να παραχθεί μια συγκεκριμένη πρόταση λέγονται “παραγωγή”

Παράδειγμα

Τερματικά:

H, γάτα, O, σκύλος, δε(v), βρίσκεται, έχει, κρυφτεί, σπιτάκι, πίσω, τραπέζι, από, το(v), καναπέ, .

Μη-τερματικά:

<αρχικό>, <υποκείμενο>, <ρηματική_φράση>, <ρήμα>, <επιρρηματική_φράση>, <τόπος>

Κανόνες:

<αρχικό> → <υποκείμενο> <ρηματική_φράση> <επιρρηματική_φράση>

<υποκείμενο> → H γάτα | O σκύλος

<ρηματική_φράση> → δε(v) <ρήμα> | <ρήμα>

<ρήμα> → βρίσκεται | έχει κρυφτεί

<επιρρηματική_φράση> → πίσω από το(v) <τόπος>

<τόπος> → τραπέζι. | σπιτάκι. | καναπέ.

Το σύμβολο → σημαίνει “μπορεί να αναπτυχθεί στα...” .

Το σύμβολο | σημαίνει “ή”. Για παράδειγμα η <ρηματική_φράση> μπορεί να είναι **δεν <ρήμα>** ή μπορεί να είναι **<ρήμα>**. Για κάθε μη-τερματικό υπάρχει τουλάχιστον ένας κανόνας. Τα μη-τερματικά περικλείονται σε < και > για να τα ξεχωρίζουμε από τα τερματικά.

Η αντικατάσταση ενός μη-τερματικού από τα σύμβολα που βρίσκονται δεξιά του βέλους λέγεται “παραγωγή”.

Ξεκινώντας από το αρχικό σύμβολο, μπορούμε να δημιουργήσουμε την πρόταση “Η γάτα βρίσκεται πίσω από το(v) καναπέ.” με την ακόλουθη σειρά από παραγωγές:

<αρχικό> → <υποκείμενο> <ρηματική_φράση> <επιρρηματική_φράση>
 → Η γάτα <ρηματική_φράση> <επιρρηματική_φράση>
 → Η γάτα <ρήμα> <επιρρηματική_φράση>
 → Η γάτα βρίσκεται <επιρρηματική_φράση>
 → Η γάτα βρίσκεται πίσω από το(ν) <τόπος>
 → Η γάτα βρίσκεται πίσω από το(ν) καναπέ .

Παρατηρείστε πως σε κάθε βήμα διαλέγουμε να αντικαταστήσουμε το *πιο αριστερό* μη-τερματικό!

Αν κάποιο μη-τερματικό έχει πολλαπλούς κανόνες παραγωγής και κάθε φορά διαλέγουμε τυχαία έναν από αυτούς, τότε μπορούμε να παράγουμε μια τυχαία πρόταση της γλώσσας μας. Αυτή είναι η ιδέα που εκμεταλλευόμαστε στην άσκηση.

Τι πρέπει να κάνετε

Θα σας δοθεί ένα αρχείο που περιέχει μια γραμματική. Το πρόγραμμά σας πρέπει να διαβάσει το αρχείο, να ανιχνεύσει τα τερματικά, μη-τερματικά και τους κανόνες, και να αποθηκεύσει όλες αυτές τις πληροφορίες σε μια κατάλληλη δομή.

Μετά, πρέπει να ακολουθήσει τη διαδικασία παραγωγής μια τυχαίας πρότασης της γλώσσας, όπως περιγράφεται πιο πάνω. Η πρόταση που παράγεται πρέπει να καταγράφεται σε ένα δεύτερο αρχείο.

Ακολουθούν λεπτομέρειες για την υλοποίηση των επιμέρους κομματιών της εργασίας.

Η γραμματική

Η γραμματική είναι γραμμένη στο αρχείο εισόδου με την παρακάτω μορφή:

- Τα μη-τερματικά σύμβολα περικλείονται σε < και >
- Οι παραγωγές για ένα μη-τερματικό περικλείονται σε { και }
 - Η πρώτη γραμμή μετά το { περιέχει το μη-τερματικό στο οποίο αναφέρεται η παραγωγή
 - Κάθε μια από τις επόμενες γραμμές μέχρι το } περιέχει μια παραγωγή για το συγκεκριμένο τερματικό.
 - Κάθε μία παραγωγή τερματίζει σε ; το οποίο δε θεωρείται μέρος της παραγωγής
- Το πρώτο σετ παραγωγών είναι πάντα για το αρχικό σύμβολο, το οποίο λέγεται <start>
- Whitespace καθώς και κείμενο που μπορεί να βρίσκεται πριν το πρώτο { ή μετά το τελευταίο } ή ανάμεσα σε παραγωγές διαφορετικών τερματικών (δηλαδή ανάμεσα σε } και {) πρέπει να αγνοείται.

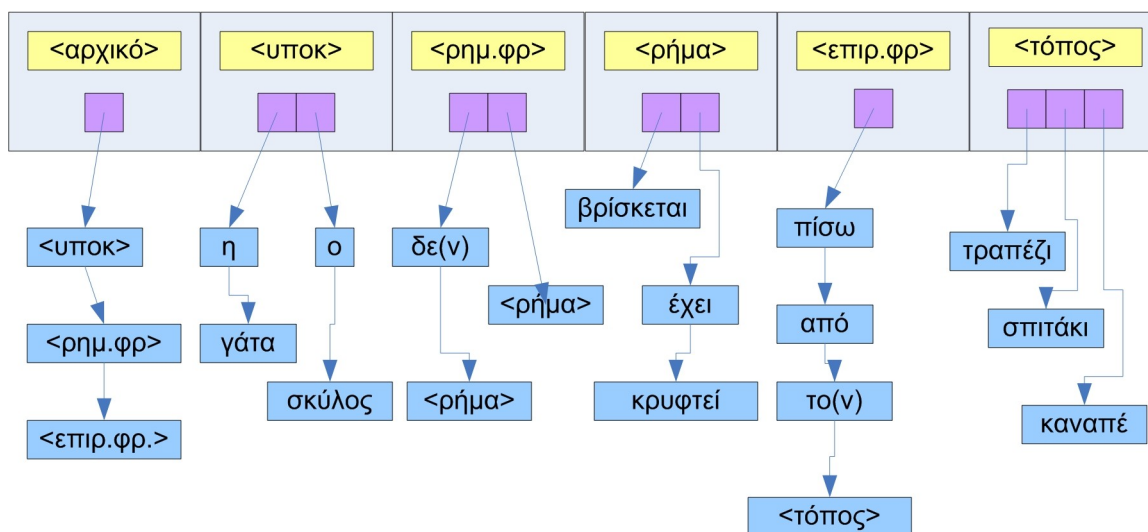
Αποθήκευση γραμματικής

Η γραμματική αποθηκεύεται σε ένα πίνακα από καταχωρίσεις. Κάθε καταχώριση αντιστοιχεί σε ένα μη-τερματικό και τις παραγωγές του. Δεν επιτρέπεται να κάνετε υποθέσεις για το πλήθος των μη-τερματικών.

Μια καταχώριση αποτελείται από ένα μη-τερματικό (συμβολοσειρά μήκους το πολύ 60) και ένα πίνακα από παραγωγές. Μπορείτε να υποθέσετε ότι ένα μη-τερματικό δε μπορεί να έχει περισσότερες από δέκα παραγωγές.

Μια παραγωγή είναι μία διασυνδεδεμένη λίστα από τερματικά. Κάθε τερματικό είναι μια συμβολοσειρά μήκους το πολύ 60.

Η παραπάνω γραμματική θα μπορούσε να έχει αποθηκευτεί ως εξής:



Σημειώστε πως το σχήμα είναι απλά για να σας δώσει μια γενική ιδέα του πώς είναι οργανωμένα τα δεδομένα. Πέρα από τις προδιαγραφές της προηγούμενης παραγράφου οι οποίες πρέπει να ακολουθηθούν αυστηρά, έχετε πολλές επιλογές για το τι και πώς θα βρίσκεται στον πίνακα (για παράδειγμα, μπορείτε να έχετε διπλή λίστα, οι λέξεις μπορεί να αποθηκευτούν με ανάποδη σειρά, μπορεί να αποφασίσετε ότι χρειάζονται περισσότερα δεδομένα ανά καταχώριση, κτλ.)

Παραγωγή πρότασης

Όταν θα έχετε αποθηκεύσει ολόκληρη τη γραμματική, πρέπει να παράγετε μια “πρόταση” τη γλώσσας, ξεκινώντας από το αρχικό σύμβολο. Σε κάθε ενδιάμεσο βήμα θα χρειαστεί να παρακολουθείτε μια ενδιάμεση μορφή της πρότασης η οποία αποτελείται από ανακατεμένα τερματικά και μη-τερματικά, για παράδειγμα, *Η γάτα <ρήμα> <επιρρηματική_φράση>*. Προτείνουμε να χρησιμοποιήσετε μια δομή στοίβας, ή εναλλακτικά, αναδρομή.

Γενικά η παραγωγή της πρότασης προχωρά ως εξής: Διαλέγετε το πιο αριστερό σύμβολο. Αν αυτό είναι τερματικό, μπορείτε να το γράψετε απευθείας στο αρχείο εξόδου. Αν είναι μη-τερματικό, ψάχνετε να το βρείτε στον πίνακα. Όταν το βρείτε, διαλέγετε τυχαία μια από τις διαθέσιμες παραγωγές αυτού του μη-τερματικού. Κατόπιν, αφαιρείτε από την ενδιάμεση μορφή το μη-τερματικό και προσθέτετε στη θέση του τα σύμβολα της επιλεγμένης παραγωγής. Προσοχή στη σειρά με την οποία τα προσθέτετε.

Αρχείο εισόδου

Το αρχείο εισόδου πρέπει να δίνεται στη γραμμή εντολής. Είναι της μορφής *x.y* όπου *x* είναι το όνομά του και *y* η επέκταση. Δε μπορείτε να κάνετε υποθέσεις για το όνομα ή την επέκταση του αρχείου εισόδου. Αν υπάρχει οποιοδήποτε πρόβλημα στη διαχείριση του αρχείου, το πρόγραμμα πρέπει να εκτυπώνει ένα κατάλληλο μήνυμα και να τερματίζει.

Αρχείο εξόδου

Το όνομα του αρχείου εξόδου πρέπει να το δημιουργείτε με βάση το όνομα του αρχείου εισόδου, ως εξής: Αν το αρχείο εισόδου έχει όνομα *x* και επέκταση *y*, τότε το αρχείο εξόδου πρέπει να έχει όνομα *x1* και επέκταση *.out*. Αν όμως το αρχείο *x1.out* υπάρχει ήδη στο φάκελο (από προηγούμενη εκτέλεση της εφαρμογής), τότε το όνομα του αρχείου εξόδου πρέπει να είναι *x2.out*, κ.ο.κ. Με άλλα λόγια, κάθε φορά που εκτελείται το πρόγραμμα, δημιουργεί στο φάκελο και αποθηκεύει την παραχθείσα πρόταση σε ένα νέο αρχείο εξόδου με όνομα *xI.out* όπου *I* είναι ο επόμενος ακέραιος από τα ονόματα των ήδη υπαρχόντων αρχείων εξόδου.

Επιπλέον, όταν δεσμεύετε μνήμη για το όνομα του αρχείου εξόδου, πρέπει να δεσμεύετε ακριβώς όση

χρειάζεται - ούτε περισσότερη ούτε λιγότερη.

Τέλος, αν υπάρχει οποιοδήποτε πρόβλημα στη διαχείριση του αρχείου, τότε πρέπει εναλλακτικά το πρόγραμμά σας να παρουσιάζει την παραχθείσα πρόταση στην οθόνη.

Μορφή εξόδου

Η πρόταση που παράγετε θα περιέχει μόνο τερματικά σύμβολα. Πρέπει να είναι διαμορφωμένη σε μορφή παραγράφου όπου η κάθε γραμμή δεν περιέχει περισσότερους από 60 χαρακτήρες. Επιπλέον, πρέπει να φροντίσετε να μην υπάρχουν περισσότερα από ένα κενό ανάμεσα σε διαδοχικές λέξεις.

Μνήμη

Το πρόγραμμά σας θα χρησιμοποιεί δυναμικά δεσμευμένη μνήμη. Πρέπει να φροντίσετε ώστε όλη η δυναμικά δεσμευμένη μνήμη του προγράμματος απελευθερώνεται πριν τον τερματισμό του.

Άλλες απαιτήσεις

Διασπάστε το πρόγραμμά σας σε κατάλληλα header files και source files. Αν χρησιμοποιήσετε συγκεκριμένες δομές (πχ. στοιβ) κάντε την υλοποίησή τους σε ξεχωριστό αρχείο.

Χρησιμοποιήστε καλά ορισμένες συναρτήσεις. Θα δοθεί ιδιαίτερη σημασία στο πώς έχετε οργανώσει τον κώδικά σας.

Χρησιμοποιήστε αποτελεσματικά σχόλια, περιγραφικά ονόματα μεταβλητών και σωστή στοίχιση.

Έλεγχος ορθότητας

Σας παρέχονται δύο γραμματικές:

Η `cats_and_dogs.grammar` είναι μια μικρή γραμματική παρόμοια με το παράδειγμα αυτού του φυλλαδίου. Χρησιμοποιήστε τη για το βασικό έλεγχο ορθότητας του προγράμματός σας και για debugging.

Η `bond_movie.story` είναι μια μεγάλη γραμματική η οποία παράγει μια φανταστική υπόθεση για την επόμενη ταινία του James Bond. Χρησιμοποιήστε τη για πιο εκτεταμένο έλεγχο.

Αν θέλετε, μπορείτε να γράψετε και δικές σας γραμματικές και να τις μοιραστείτε μαζί μας.