ΚΕΦΑΛΑΙΟ

Μοντελοποίηση Υπολογισμού

Οι υπολογιστές εκτελούν πολλές εργασίες. Οταν δίνεται μια εργασία, εμφανίζονται δύο ερωτήματα. Το πρώτο είναι: Μήπως αυτή μπορεί να εκτελεστεί με χρήση υπολογιστή; Από την στιγμή που γνωρίζουμε ότι αυτό το πρώτο ερώτημα έχει καταφατική απάντηση, μπορούμε να κάνουμε το δεύτερο ερώτημα: Με ποιό τρόπο μπορεί να εκτελεστεί η εργασία; Για βοήθεια στην απάντηση αυτών των ερωτημάτων χρησιμοποιούνται μοντέλα υπολογισμού.

Θα μελετήσουμε τρία είδη δομών, οι οποίες χρησιμοποιούνται σε μοντέλα υπολογισμών, τα οποία είναι οι γραμματικές, οι μηχανές πεπερασμένης κατάστασης, και οι μηχανές Turing. Οι γραμματικές χρησιμοποιούνται στην δημιουργία των λέξεων γλώσσας και τον προσδιορισμό τού αν μια λέξη βρίσκεται σε γλώσσα. Οι τυπικές γλώσσες, οι οποίες δημιουργούνται από γραμματικές, προσφέρουν μοντέλα και για φυσικές γλώσσες, όπως τα Αγγλικά, και για γλώσσες προγραμματισμού, όπως είναι η Pascal, η Fortran, η Prolog, η C, και η Java. Ειδικότερα, οι γραμματικές είναι εξαιρετικά σημαντικές στην κατασκευή και στην θεωρία των μεταγλωττιστών. Οι γραμματικές που θα εξετάσουμε εδώ χρησιμοποιήθηκαν για πρώτη φορά από τον Αμερικανό γλωσσολόγο Noam Chomsky την δεκαετία του 1950.

Στην κατασκευή μοντέλων χρησιμοποιούνται διάφορα είδη μηχανών πεπερασμένης κατάστασης. Ολες οι μηχανές πεπερασμένης κατάστασης έχουν ένα σύνολο καταστάσεων, όπου περιλαμβάνονται και μια κατάσταση εκκίνησης, ένα αλφάβητο εισόδου, και μια συνάρτηση μετάβασης η οποία αναθέτει μια επόμενη κατάσταση σε κάθε ζεύγος καταστάσεων και μια είσοδο. Οι καταστάσεις μηχανής πεπερασμένης κατάστασης της δίνουν περιορισμένες ικανότητες μνήμης. Κάποιες μηχανές πεπερασμένης κατάστασης δίνουν ένα σύμβολο εξόδου για κάθε μετάβαση. Οι μηχανές αυτές χρησιμοποιούνται για κατασκευή μοντέλων πολλών ειδών μηχανών, όπως μηχανών πωλήσεων, μηχανών καθυστέρησης, δυαδικών μηχανών πρόσθεσης, και μηχανών αναγνώρισης γλωσσών. Θα μελετήσουμε και μηχανές πεπερασμένης κατάστασης που δεν έχουν έξοδο, αλλά έχουν τελικές καταστάσεις. Οι μηχανές αυτές χρησιμοποιούνται σε μεγάλο βαθμό στην αναγνώριση γλωσσών. Οι συμβολοσειρές που αναγνωρίζονται είναι αυτές που μετα-

φέρουν την κατάσταση εκκίνησης σε τελική κατάσταση. Η έννοιες των γραμματικών και των μηχανών πεπερασμένης κατάστασης μπορούν να συνδεθούν. Θα χαρακτηρίσουμε τα σύνολα που αναγνωρίζονται από μηχανή πεπερασμένης κατάστασης και θα δείξουμε ότι είναι ακριβώς τα σύνολα που γεννιούνται από ορισμένο είδος γραμματικής.

Τέλος, θα παρουσιάσουμε την έννοια της μηχανής Turing. Θα δείξουμε τον τρόπο με τον οποίο χρησιμοποιούνται οι μηχανές Turing για αναγνώριση συνόλων. Θα δείξουμε, ακόμη, τον τρόπο χρήσης των μηχανών Turing για υπολογισμό αριθμοθεωρητικών συναρτήσεων. Θα εξετάσουμε την θέση Church-Turing, η οποία αναφέρει ότι κάθε αποτελεσματικός υπολογισμός μπορεί να εκτελεστεί με χρήση μηχανής Turing.

11.1 Γλώσσες και Γραμματικές

ΕΙΣΑΓΩΓΗ

Οι λέξεις της Αγγλικής γλώσσας (Σημ. Μετ.: και οποιασδήποτε γλώσσας) συνδυάζονται με διάφορους τρόπους. Η γραμματική της Αγγλικής μας λέει αν συνδυασμός λέξεων αποτελεί πρόταση που ισχύει. Για παράδειγμα, η φράση *the frog writes neatly* (*ο βάτραχος γράφει καθαρά*) είναι φράση που ισχύει, επειδή σχηματίζεται από μια φράση ουσιαστικού, *the frog*, η οποία αποτελείται από το άρθρο *the* και από το ουσιαστικό *frog*, που ακολουθείται από μια φράση ρήματος, *writes neatly*, η οποία αποτελείται από το ρήμα *writes*, και από το επίρρημα *neatly*. Δεν δίνουμε σημασία στο ότι πρόκειται για δήλωση που δεν έχει έννοια, επειδή εξετάζουμε μόνο την **σύνταξη**, ή μορφή, της πρότασης, και όχι την **σημασιολογία** της, ή έννοια. Παρατηρούμε, ακόμη, ότι ο συνδυασμός των λέξεων *swims quickly mathematics* (*κολυμπάει γρήγορα μαθηματικά*) αποτελεί πρόταση που δεν ισχύει επειδή δεν ακολουθεί τους κανόνες της Αγγλικής γραμματικής.

Η σύνταξη μιάς **φυσικής γλώσσας**, δηλ., μιας ομιλούμενης γλώσσας, όπως είναι τα Αγγλικά, τα Γαλλικά, τα Γερμανικά, ή τα Ισπανικά, είναι εξαιρετικά πολύπλοκη. Μάλιστα, φαίνεται ότι δεν είναι δυνατό να καθορίσουμε όλους τους κανόνες σύνταξης φυσικής γλώσσας. Η έρευνα για αυτόματη μετάφραση μιάς γλώσσας σε άλλη έχει οδηγήσει στην έννοια της **τυπικής γλώσσας**, η οποία, αντίθετα από την φυσική γλώσσα, καθορίζεται από καλά διατυπωμένο σύνολο κανόνων σύνταξης. Οι κανόνες σύνταξης είναι σημαντικοί όχι μόνο στην γλωσσολογία, δηλ., την μελέτη των φυσικών γλωσσών, αλλά και στην μελέτη των γλωσσών προγραμματισμού.

Θα περιγράψουμε τις προτάσεις μιάς τυπικής γλώσσας με χρήση μιάς γραμματικής. Η χρήση των γραμματικών βοηθάει όταν εξετάζουμε τις δύο κλάσεις προβλημάτων που εμφανίζονται συχνότερα σε εφαρμογές γλωσσών προγραμματισμού: (1) Με ποιό τρόπο προσδιορίζουμε αν ένας συνδυασμός λέξεων αποτελεί πρόταση που ισχύει σε τυπική γλώσσα; (2) Με ποιό τρόπο παράγονται οι έγκυρες προτάσεις τυπικής γλώσσας που ισχύουν;

Πριν δώσουμε τεχνικό ορισμό μιάς γραμματικής, θα περιγράψουμε ένα παράδειγμα γραμματικής, η οποία γεννάει ένα υποσύνολο της Αγγλικής γλώσσας. Αυτό το υποσύνολο ορίζεται με χρήση λίστας κανόνων που περιγράφουν τον τρόπο με τον οποίο δημιουργείται μια πρόταση που ισχύει. Καθορίζουμε ότι

1. πρόταση αποτελείται από **φράση ουσιαστικού** που ακολουθείται από **φράση ρήματος**
2. **πρόταση ουσιαστικού** αποτελείται από **άρθρο** που ακολουθείται από **επίθετο** που ακολουθείται από **ουσιαστικό**, ή
3. **πρόταση ουσιαστικού** αποτελείται από **άρθρο** που ακολουθείται από **ουσιαστικό**
4. **πρόταση ρήματος** αποτελείται από **ρήμα** που ακολουθείται από **επίρρημα**, ή
5. **πρόταση ρήματος** αποτελείται από **ρήμα**
6. άρθρο είναι το *a* (*ένα*)
7. άρθρο είναι το *the* (*το*)
8. επίθετο είναι το *large* (*μεγάλο*), ή
9. επίθετο είναι το *hungry* (*πεινασμένος*)
10. ουσιαστικό είναι το *rabbit* (*λαγός*), ή
11. ουσιαστικό είναι το *mathematician* (*μαθηματικός*)
12. ρήμα είναι το *eats* (*τρώει*), ή
13. ρήμα είναι το *hops* (*πηδάει*)
14. επίρρημα είναι το *quickly* (*γρήγορα*), ή
15. επίρρημα είναι το *wildly* (*τρελλά*).

Από τους κανόνες αυτούς μπορούμε να σχηματίσουμε έγκυρες προτάσεις, με χρήση μιάς σειράς αντικαταστάσεων, μέχρι που να μη μπορούν να χρησιμοποιηθούν κανόνες. Για παράδειγμα, ακολουθούμε την διαδοχή αντικαταστάσεων:

πρόταση
φράση ουσιαστικού φράση ρήματος
άρθρο επίθετο ουσιαστικό φράση ρήματος
άρθρο επίθετο ουσιαστικό ρήμα επίρρημα
the **επίθετο ουσιαστικό ρήμα επίρρημα**
the lar **ουσιαστικό ρήμα επίρρημα**
the large rabbit **ρήμα επίρρημα**
the large rabbit hops **επίρρημα**
the large rabbit hops quickly

έτσι ώστε να πάρουμε μια πρόταση που ισχύει. Εύκολα βλέπουμε ότι κάποιες άλλες προτάσεις που ισχύουν είναι οι: *a hungry mathematician eats wildly*, *a large mathematician hops*, *the rabbit eats quickly*, κ.ο.κ. Βλέπουμε, ακόμη, ότι η *the quickly eats mathematician* είναι πρόταση που δεν ισχύει.

ΓΡΑΜΜΑΤΙΚΕΣ ΔΟΜΗΣ ΦΡΑΣΕΩΝ

Πριν δώσουμε τον τυπικό ορισμό γραμματικής, θα παρουσιάσουμε κάποια ορολογία.

ΟΡΙΣΜΟΣ 1 *Λεξιλόγιο* (ή *αλφάβητο*) V είναι πεπερασμένο, μη κενό σύνολο στοιχείων που ονομάζονται *σύμβολα*. *Λέξη* (ή *πρόταση*) επί του V είναι συμβολοσειρά πεπερασμένου μήκους στοιχείων του V. Η *κενή συμβολοσειρά* ή *μηδενική συμβολοσειρά*, που συμβολίζεται με λ, είναι η συμβολοσειρά που δεν περιέχει σύμβολα. Το σύνολο όλων των λέξεων επί του V συμβολίζεται με V^*. *Γλώσσα επί* του V είναι υποσύνολο του V^*.

Σημειώνουμε ότι η λ, η κενή συμβολοσειρά, είναι η συμβολοσειρά που δεν περιέχει σύμβολα. Είναι διαφορετική από το $\emptyset$, το κενό σύνολο. Επεται ότι το $\{\lambda\}$ είναι το σύνολο που περιέχει μόνο μια συμβολοσειρά, την κενή συμβολοσειρά.

Οι γλώσσες καθορίζονται με διάφορους τρόπους. Ενας τρόπος είναι η καταγραφή όλων των λέξεων της γλώσσας. Ενας άλλος είναι να δοθούν κάποια κριτήρια τα οποία θα πρέπει να ικανοποιεί μια λέξη για να είναι στην γλώσσα. Στην παράγραφο αυτή θα περιγράψουμε έναν άλλο σημαντικό τρόπο καθορισμού γλώσσας, που είναι μέσω της χρήσης γραμματικής, όπως του συνόλου κανόνων που δώσαμε στην εισαγωγή αυτής της παραγράφου. Μια γραμματική παρέχει ένα σύνολο συμβόλων διαφόρων ειδών και ένα σύνολο κανόνων δημιουργίας λέξεων. Ακριβέστερα, μια γραμματική έχει ένα **λεξιλόγιο** V, το οποίο είναι σύνολο συμβόλων που χρησιμοποιείται για να δίνει μέλη της γλώσσας. Κάποια απο τα στοιχεία του λεξιλογίου δεν μπορούν να αντικατασταθούν από άλλα σύμβολα. Αυτά ονομάζονται **τερματικά**, και τα άλλα μέλη του λεξιλογίου, τα οποία μπορούν να αντικατασταθούν από άλλα σύμβολα, ονομάζονται **μη τερματικά**. Τα σύνολα των τερματικών και των μη τερματικών συνήθως συμβολίζονται με T και N, αντίστοιχα. Στο παράδειγμα της εισαγωγής της παραγράφου, το σύνολο των τερματικών είναι το $\{a, the, rabbit, mathematician, hops, eats, quickly, wildly\}$, και το σύνολο των μη τερματικών είναι το {**πρόταση**, **φράση ουσιαστικού**, **φράση ρήματος**, **επίθετο**, **άρθρο**, **ουσιαστικό**, **ρήμα**, **επίρρημα**}.Υπάρχει ένα ειδικό μέλος του λεξιλογίου, επονομαζόμενο **σύμβολο εκκίνησης**, που συμβολίζεται με S, το οποίο είναι το στοιχείο του λεξιλογίου με το οποίο πάντοτε ξεκινούμε. Στο παράδειγμα της εισαγωγής, το σύμβολο εκκίνησης είναι **πρόταση**. Οι κανόνες που καθορίζουν πότε μπορούμε να αντικαταστήσουμε συμβολοσειρά από το V^*, το σύνολο όλων των συμβολοσειρών στοιχείων στο λεξιλόγιο, με άλλη συμβολοσειρά ονομάζονται **αρχές παραγωγής** της γραμματικής. Συμβολίζουμε με $z_0 \rightarrow z_1$ την δημιουργία που καθορίζει ότι το z_0 μπορεί να αντικατασταθεί από το z_1 μέσα σε συμβολοσειρά. Εχουν καταγραφεί οι αρχές παραγωγής στην γραμματική που δίνεται στην εισαγωγή αὐτής της παραγράφου. Η πρώτη αρχή παραγωγής, που είναι γραμμένη με τον συμβολισμό αυτό, είναι η **πρόταση** $\rightarrow$ **φράση ουσιαστικού φράση ρήματος**. Ανακεφαλαιώνουμε με τον παρακάτω ορισμό.

ΟΡΙΣΜΟΣ 2 Η *γραμματική δομής φράσεων* $G = (V, T, S, P)$ αποτελείται από λεξιλόγιο V, υποσύνολο T του V που αποτελείται από τερματικά στοιχεία, σύμβολο εκκίνησης S από το V, και σύνολο αρχών παραγωγών P. Το σύνολο V–T

συμβολίζεται με N. Τα στοιχεία του N ονομάζονται *μη τερματικά σύμβολα*. Κάθε δημιουργία του P θα πρέπει να περιέχει τουλάχιστο ένα μη τερματικό στην αριστερή του πλευρά.

ΠΑΡΑΔΕΙΓΜΑ 1

Εστω ότι $G=(V,T,S,P)$ όπου $V=(a,b,A,B,S\}$, $T=\{a,b\}$, S είναι το σύμβολο εκκίνησης, και $P=\{S\to ABa, A\to BB, B\to ab, AB\to b\}$. Το G αποτελεί παράδειγμα γραμματικής δομής φράσεων.

Θα ενδιαφερθούμε για τις λέξεις που γεννιούνται με τις αρχές παραγωγής γραμματικής δομής φράσεων.

ΟΡΙΣΜΟΣ 3 Εστω ότι η $G=(V,T,S,P)$ είναι γραμματική δομής φράσεων. Εστω ότι η $w_0=lz_0r$ (δηλ., η αλληλουχία των l, z_0, και r) και η $w_1=lz_1r$ είναι συμβολοσειρές επί του V. Αν η $z_0\to z_1$ είναι αρχή παραγωγής της G, λέμε ότι η w_1 *παράγεται άμεσα* από την w_0 και γράφουμε $w_0\Rightarrow w_1$. Αν οι $w_0,w_1,\ldots,w_n$ είναι συμβολοσειρές επί του V, έτσι ώστε να είναι $w_0\Rightarrow w_1$, $w_1\Rightarrow w_2$, . . . , $w_{n-1}\Rightarrow w_n$, τότε λέμε ότι η w_n *παράγεται από την* w_0 και γράφουμε $w_0\overset{*}{\Rightarrow} w_n$. Η ακολουθία βημάτων που χρησιμοποιούμε για να πάρουμε την w_n από την w_0 ονομάζεται *παραγωγή*.

ΠΑΡΑΔΕΙΓΜΑ 2

Η συμβολοσειρά $Aaba$ παράγεται άμεσα από την ABa στην γραμματική του Παραδείγματος 1 επειδή η $B\to ab$ είναι αρχή παραγωγής της γραμματικής. Η συμβολοσειρά $abababa$ παράγεται από την ABa επειδή $ABa\Rightarrow Aaba\Rightarrow BBaba\Rightarrow Bababa\Rightarrow abababa$, με διαδοχική χρήση των δημιουργιών $B\to ab$, $A\to BB$, $B\to ab$, και $B\to ab$.

ΟΡΙΣΜΟΣ 4 Εστω ότι η $G=(V,T,S,P)$ είναι γραμματική δομής φράσεων. Η *γλώσσα που γεννιέται* από την G (ή η *γλώσσα της* G), που συμβολίζεται με $L(G)$, είναι το σύνολο όλων των συμβολοσειρών των τερματικών που παράγονται από την κατάσταση εκκίνησης S. Με άλλα λόγια,

$$L(G)=\{w\in T^* \mid S\overset{*}{\Rightarrow} w\}.$$

Στα Παραδείγματα 3 και 4 βρίσκουμε την γλώσσα που δημιουργείται από γραμματική δομής φράσεων.

ΠΑΡΑΔΕΙΓΜΑ 3

Εστω ότι G είναι η γραμματική με λεξιλόγιο $V=\{S,A,a,b\}$, σύνολο τερματικών $T=\{a,b\}$, σύμβολο εκκίνησης S, και αρχές παραγωγής $P=\{S\to aA, S\to b, A\to aa\}$. Ποιά είναι η $L(G)$, η γλώσσα αυτής της γραμματικής;

Λύση: Από την κατάσταση εκκίνησης S παράγουμε την aA με χρήση της αρχής παραγωγής $S \to aA$. Για να παράξουμε την b μπορούμε να χρησιμοποιήσουμε και την αρχή παραγωγής $S \to b$. Για να παράξουμε την aaa, από την aA μπορούμε να χρησιμοποιήσουμε την αρχή παραγωγής $A \to aa$. Δεν μπορούν να παραχθούν επιπλέον λέξεις. Αρα $L(G) = \{b, aaa\}$.

ΠΑΡΑΔΕΙΓΜΑ 4

Εστω ότι G είναι η γραμματική με λεξιλόγιο $V = \{S, 0, 1\}$, σύνολο τερματικών $T = \{0, 1\}$, σύμβολο εκκίνησης S, και αρχές παραγωγής $P = \{S \to 11S, S \to 0\}$. Ποιά είναι η $L(G)$, η γλώσσα αυτής της γραμματικής;

Λύση: Από το S μπορούμε να παράξουμε την 0 με χρήση της $S \to 0$, ή την $11S$ με χρήση της $S \to 11S$. Από την $11S$ μπορούμε να παράξουμε είτε την 110 είτε την $1111S$. Από την $1111S$ μπορούμε να παράξουμε την 11110 και την $111111S$. Σε οποιοδήποτε στάδιο παραγωγής μπορούμε είτε να προσθέσουμε δύο 1 στο τέλος της συμβολοσειράς ή να τερματίσουμε την παραγωγή με πρόσθεση ενός 0 στο τέλος της συμβολοσειράς. Υποθέτουμε ότι $L(G) = \{0, 110, 11110, 1111110, \ldots\}$, δηλ., το σύνολο όλων των στοιχειοσειρών που αρχίζουν με άρτιο πλήθος 1 και τελειώνουν με 0. Αυτό αποδεικνύεται με επαγωγικό επιχείρημα που δείχνει ότι μετά από χρήση n δημιουργιών, οι μόνες συμβολοσειρές τερματικών που γεννιούνται είναι οι συμβολοσειρές που αποτελούνται από $n-1$ ή λιγότερες ενώσεις του 11 που ακολουθείται από 0. (Η απόδειξη αφήνεται σαν άσκηση για τον αναγνώστη.)

Συχνά παρουσιάζεται το πρόβλημα κατασκευής γραμματικής που γεννά δεδομένη γλώσσα. Τα Παραδείγματα 5, 5, και 7 περιγράφουν προβλήματα αυτού του είδους.

ΠΑΡΑΔΕΙΓΜΑ 5

Να δοθεί γραμματική δομής φράσεων που γεννάει το σύνολο $\{0^n 1^n \mid n = 0, 1, 2, \ldots\}$.

Λύση: Χρησιμοποιούμε δύο αρχές παραγωγής για να γεννήσουν όλες τις συμβολοσειρές που αποτελούνται από συμβολοσειρά με 0 που ακολουθείται από συμβολοσειρά με το ίδιο πλήθος με 1, όπου περιλαμβάνεται και η μηδενική συμβολοσειρά. Η πρώτη συμβολοσειρά κατασκευάζει διαδοχικά στην γλώσσα συμβολοσειρές με μεγαλύτερο μήκος με πρόσθεση ενός 0 στην αρχή της συμβολοσειράς και ενός 1 στο τέλος. Η δεύτερη δημιουργία αντικαθιστά το S με την κενή συμβολοσειρά. Η λύση είναι η γραμματική $G = (V, T, S, P)$, όπου $V = \{0, 1, S\}$, $T = \{0, 1\}$, S είναι το σύμβολο εκκίνησης, και οι αρχές παραγωγής είναι

$$S \to 0S1$$
$$S \to \lambda.$$

Η επαλήθευση ότι αυτή η γραμματική γεννάει το σωστό σύνολο αφήνεται σαν άσκηση για τον αναγνώστη.

Το Παράδειγμα 5 αφορούσε στο σύνολο στοιχειοσειρών που αποτελούνται από 0 που ακολουθούνται από 1, όπου το πλήθος των 0 και 1 είναι το ίδιο. Το Παράδειγμα 6 εξετάζει το σύνολο στοιχειοσειρών που αποτελούνται από 0 που ακολουθούνται από 1, όπου διαφέρει το πλήθος των 0 και 1.

ΠΑΡΑΔΕΙΓΜΑ 6

Να βρεθεί η γραμματική δομής φράσεων που γεννάει το σύνολο $\{0^m1^n \mid$ οι m και n είναι μη αρνητικοί ακέραιοι$\}$.

Λύση: Θα δώσουμε δύο γραμματικές G_1 και G_2 που γεννούν αυτό το σύνολο. Το γεγονός αυτό θα δείξει ότι δύο γραμματικές μπορούν να γεννήσουν την ίδια γλώσσα.

Η γραμματική G_1 έχει αλφάβητο $V = \{S, 0, 1\}$, τερματικά $T = \{0, 1\}$, και αρχές παραγωγής $S \to 0S$, $S \to S1$, και $S \to \lambda$. Η G_1 γεννάει το σωστό σύνολο, επειδή η χρήση της πρώτης δημιουργίας m φορές θέτει m 0 στην αρχή της συμβολοσειράς, και η χρήση της δεύτερης δημιουργίας n φορές θέτει n 1 στο τέλος της συμβολοσειράς. Οι λεπτομέρειες αυτής της επαλήθευσης αφήνονται στον αναγνώστη.

Η γραμματική G_2 έχει αλφάβητο $V = \{S, A, 0, 1\}$, τερματικά $T = \{0, 1\}$, και αρχές παραγωγής $S \to 0S$, $S \to 1A$, $S \to 1$, $A \to 1A$, $A \to 1$, $S \to \lambda$. Οι λεπτομέρειες ότι αυτή η γραμματική γεννάει το σωστό σύνολο αφήνονται σαν άσκηση για τον αναγνώστη.

Μερικές φορές σύνολο που περιγράφεται εύκολα γεννιέται μόνο με πολύπλοκη γραμματική. Αυτό δείχνεται με το Παράδειγμα 7.

ΠΑΡΑΔΕΙΓΜΑ 7

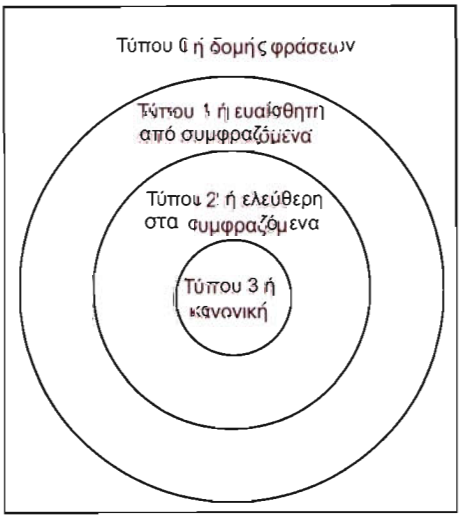
Μια γραμματική που γεννάει το σύνολο $\{0^n1^n2^n \mid n = 0, 1, 2, 3, \ldots\}$ είναι η $G = (V, T, S, P)$ με $V = \{0, 1, 2, S, A, B\}$, $T = \{0, 1, 2\}$, κατάσταση εκκίνησης S, και αρχές παραγωγής $S \to 0SAB$, $S \to \lambda$, $BA \to AB$, $0A \to 01$, $1A \to 11$, $1B \to 12$, $2B \to 22$. Αφήνουμε στον αναγνώστη σαν άσκηση να δείξει ότι αυτή η δήλωση είναι σωστή. Η γραμματική που δίνεται είναι το απλούστερο είδος γραμματικής που γεννάει αυτό το σύνολο, από άποψη που θα γίνει φανερή παρακάτω στην παράγραφο αυτή. Ισως ο αναγνώστης να απορεί από πού ήλθε αυτή η γραμματική, επειδή φαίνεται δύσκολο αυτή η γραμματική να εμφανιστεί από το "πουθενά". Ισως βοηθάει να γνωρίζουμε ότι αυτή η γραμματική μπορεί να κατασκευαστεί με συστηματικό τρόπο με χρήση τεχνικών από την θεωρία υπολογισμού, η οποία είναι πέρα από το αντικείμενο αυτού του βιβλίου.

ΕΙΔΗ ΓΡΑΜΜΑΤΙΚΩΝ ΔΟΜΗΣ ΦΡΑΣΕΩΝ

Οι γραμματικές δομής φράσεων ταξινομούνται σύμφωνα με τους τύπους των αρχών παραγωγής που επιτρέπονται. Θα περιγράψουμε τον τρόπο ταξινόμησης που παρουσιάστηκε από τον Noam Chomsky. Στην Παράγραφο 11.4 θα δούμε ότι τα διαφορετικά είδη γλωσσών που ορίζονται στο τρόπο αυτό αντιστοιχούν στις κλάσεις γλωσσών που αναγνωρίζονται με χρήση διαφορετικών μοντέλων υπολογιστικών μηχανών.

Η γραμματική **τύπου 0** δεν έχει περιορισμούς στις αρχές παραγωγής της. Η γραμματική **τύπου 1** μπορεί να έχει αρχές παραγωγής με την μορφή $w_1 \rightarrow w_2$ όπου το μήκος της w_2 είναι μεγαλύτερο από το μήκος της w_1 ή της μορφής $w_1 \rightarrow \lambda$. Η γραμματική **τύπου 2** μπορεί να έχει αρχές παραγωγής μόνο με την μορφή $w_1 \rightarrow w_2$, όπου w_1 είναι ένα σύμβολο που δεν είναι τερματικό σύμβολο. Η γραμματική **τύπου 3** μπορεί να έχει αρχές παραγωγής μόνο με την μορφή $w_1 \rightarrow w_2$ με $w_1 = A$ και είτε $w_2 = aB$ ή $w_2 = a$, όπου τα A και B είναι μη τερματικά σύμβολα και το a είναι τερματικό σύμβολο, ή με $w_1 = S$ και $w_2 \rightarrow \lambda$.

Από τους ορισμούς αυτούς βλέπουμε ότι κάθε γραμματική τύπου 3 είναι γραμματική τύπου 2, κάθε γραμματική τύπου 2 είναι γραμματική τύπου 1, και κάθε γραμματική τύπου 1 είναι γραμματική τύπου 0. Οι γραμματικές τύπου 2 ονομάζονται **γραμματικές ελεύθερες από συμφραζόμενα,** επειδή μη τερματικό σύμβολο που είναι το αριστερό μέλος δημιουργίας μπορεί να αντικατασταθεί σε συμβολοσειρά όταν εμφανίζεται, ανεξάρτητα από το τι άλλο βρίσκεται στην συμβολοσειρά. Γλώσσα που γεννιέται από γραμματική τύπου 2 ονομάζεται **γλώσσα ελεύθερη από συμφραζόμενα ή ασυμφραστική**. Οταν υπάρχει δημιουργία με την μορφή $lw_1r \rightarrow lw_2r$ (αλλά όχι με την μορφή $w_1 \rightarrow w_2$), η γραμματική ονομάζεται τύπου 1 ή **ευαίσθητη στα συμφραζόμενα ή συμφραστική** επειδή το w_1 μπορεί να αντικατασταθεί από το w_2 μόνο όταν περιβάλλεται από τις συμβολοσειρές l και r. Οι γραμματικές τύπου 3 ονομάζονται και **κανονικές γραμματικές**. Γλώσσα που γεννιέται από κανονική γραμματική ονομάζεται **κανονική**. Η Παράγραφος 11.4 ασχολείται με την σχέση μεταξύ κανονικών γλωσσών και μηχανών πεπερασμένης κατάστασης. Στο διάγραμμα Venn του Σχήματος 1 φαίνεται η σχέση μεταξύ των διαφόρων τύπων γραμματικών.



ΣΧΗΜΑ 1 Τύποι Γραμματικών

ΠΑΡΑΔΕΙΓΜΑ 8

Από το Παράδειγμα 6 γνωρίζουμε ότι η $\{0^m1^n \mid m, n = 0, 1, 2, \dots\}$ είναι κανονική γλώσσα, επειδή μπορεί να γεννηθεί από κανονική γραμματική, που είναι η γραμματική G_2 του Παραδείγματος 6.

Οι ελεύθερες από συμφραζόμενα γραμματικές και οι κανονικές γραμματικές παίζουν σημαντικό ρόλο στις γλώσσες προγραμματισμού. Οι ελεύθερες από συμφραζόμενα γραμματικές χρησιμοποιούνται για να ορίζουν την σύνταξη σχεδόν όλων των γλωσσών προγραμματισμού. Αυτές οι γραμματικές είναι αρκετά δυνατές για να ορίζουν μια μεγάλη περιοχή γλωσσών. Επιπλέον, μπορούν να επινοηθούν αποτελεσματικοί αλγόριθμοι για να προσδιοριστεί αν μπορεί να γεννηθεί μια συμβολοσειρά και με ποιό τρόπο.Οι κανονικές γραμματικές χρησιμοποιούνται για αναζήτηση σε κείμενο ορισμένων προτύπων και σε λεξικογραφική ανάλυση, η οποία είναι η διαδικασία μετασχηματισμού ροής εισόδου σε ροή συμβόλων για χρήση από αναλυτή.

ΠΑΡΑΔΕΙΓΜΑ 9

Από το Παράδειγμα 5 έπεται ότι η $\{0^n1^n \mid n = 0, 1, 2, \dots\}$ είναι ευαίσθητη στα συμφραζόμενα γλώσσα, επειδή οι αρχές παραγωγής στην γραμματική αυτή είναι $S \to 0S1$ και $S \to \lambda$. Ωστόσο, δεν είναι κανονική γλώσσα. Αυτό θα δειχτεί στην παράγραφο 11.4.

ΠΑΡΑΔΕΙΓΜΑ 10

Το σύνολο $\{0^n1^n2^n \mid n = 0, 1, 2, \dots\}$ είναι ευαίσθητη στα συμφραζόμενα γλώσσα, επειδή μπορεί να γεννηθεί από γραμματική τύπου 1, όπως δείχνει το Παράδειγμα 7, αλλά όχι από οποιαδήποτε γλώσσα τύπου 2. (Αυτό φαίνεται στην Ασκηση 28, στις συμπληρωματικές ασκήσεις στο τέλος του κεφάλαιου.)

Στον Πίνακα 1 δίνεται περιληπτικά η ορολογία που χρησιμοποιείται για την ταξινόμηση γραμματικών με δομή φράσεων.

ΠΙΝΑΚΑΣ 1 Τύποι Γραμματικών

Τύπος	*Περιορισμοί στις Δημιουργίες* $w_1 \to w_2$
0	Δεν υπάρχουν περιορισμοί
1	$l(w_1) < l(w_2)$, ή $w_2 = \lambda$
2	$w_1 = A$ όπου A είναι μη τερματικό σύμβολο
3	$w_1 = A$ και $w_2 = aB$ ή $w_2 = a$, όπου $A \in N$, $B \in N$, και $a \in T$, ή $S \to \lambda$

ΔΕΝΔΡΑ ΠΑΡΑΓΩΓΗΣ

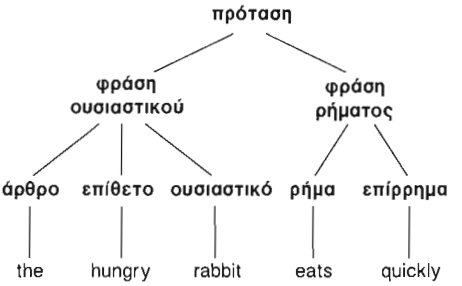
Παραγωγή στην γλώσσα που γεννιέται από ελεύθερη από συμφραζόμενα γραμματική παριστάνεται με γραφικό τρόπο με χρήση διατεταγμένου ριζωμένου δένδρου, που ονομάζεται **δένδρο παραγωγής**. Η ρίζα αυτού του δένδρου παριστά-

νει το σύμβολο εκκίνησης. Οι εσωτερικές κορυφές του δένδρου παριστάνουν τα μη τερματικά σύμβολα που εμφανίζονται στην παραγωγή. Τα φύλλα του δένδρου παριστάνουν τα τερματικά σύμβολα που εμφανίζονται. Αν στην παραγωγή εμφανίζεται η αρχή παραγωγής $A \rightarrow w$, όπου w είναι λέξη, η κορυφή που παριστάνει την A έχει σαν παιδιά κορυφές που παριστάνουν κάθε σύμβολο στην w, με την σειρά από τα αριστερά προς τα δεξιά.

ΠΑΡΑΔΕΙΓΜΑ 11

Να κατασκευαστεί δένδρο παραγωγής για την παραγωγή της *the hungry rabbit eats quickly*, που δίνεται στην εισαγωγή της παραγράφου.

Λύση: Στο Σχήμα 2 φαίνεται το δένδρο παραγωγής.



Σχήμα 2 Δένδρο Παραγωγής

Το πρόβλημα του προσδιορισμού αν συμβολοσειρά βρίσκεται στην γλώσσα που γεννιέται από γραμματική ελεύθερη συμφραζομένων εμφανίζεται σε πολλές εφαρμογές, όπως στην κατασκευή μεταγλωττιστών. Στο παρακάτω παράδειγμα δείχνονται δύο προσεγγίσεις στο πρόβλημα αυτό.

ΠΑΡΑΔΕΙΓΜΑ 12

Να προσδιοριστεί αν η λέξη *cbab* ανήκει στην γλώσσα που γεννιέται από την γραμματική $G=(V,T,S,P)$ όπου $V = \{a,b,c,A,B,C,S\}$, $T = \{a,b,c\}$, S είναι το σύμβολο εκκίνησης, και οι αρχές παραγωγής είναι

$S \rightarrow AB$
$A \rightarrow Ca$
$B \rightarrow Ba$
$B \rightarrow Cb$
$B \rightarrow b$
$C \rightarrow cb$
$C \rightarrow b$.

AVRAM NOAM CHOMSKY (1928 -) Ο Noam Chomsky, που γεννήθηκε στην Φιλαδέλφεια της Πολιτείας Pennsylvania των ΗΠΑ, είναι γιός Εβραίου λόγιου. Πήρε τα πτυχία του B.A., M.A., και διδακτορικό στην γλωσσολογία, όλα από το Πανεπιστήμιο της Pennsylvania. Ανήκε στο προσωπικό του Πανεπιστημίου της Pennsylvania από το 1950 μέχρι το 1951. Το 1955 πήγε στο προσωπικό του M.I.T., όπου ξεκίνησε την σταδιοδρομία του στο M.I.T. με διδασκαλία Γαλλικής και Γερμανικής γλώσσας στους μηχανικούς. Σήμερα ο Chomsky είναι Καθηγητής ξένων γλωσσών και γλωσσολογίας στην έδρα Ferrari P. Ward στο M.I.T. Είναι γνωστός για τις πολλές θεμελιώδεις συνεισφορές του στην γλωσσολογία, μαζί με την μελέτη γραμματικών.

Ο Chomsky είναι, ακόμη, πολύ γνωστός για την ειλικρινή του πολιτική δραστηριότητα.

Λύση: Ενας τρόπος προσέγγισης αυτού του προβλήματος είναι να ξεκινήσουμε με το S και να προσπαθήσουμε να παράξουμε την *cbab* με χρήση σειρά αρχών παραγωγής. Επειδή υπάρχει μόνο μια δημιουργία με το S στην αριστερή της πλευρά, θα πρέπει να ξεκινήσουμε με την $S \Rightarrow AB$. Στη συνέχεια θα χρησιμοποιήσουμε την μόνη αρχή παραγωγής που έχει το A στην αριστερή της πλευρά, δηλαδή την $A \rightarrow Ca$, για να πάρουμε την $S \Rightarrow AB \Rightarrow CaB$. Επειδή η *cbab* αρχίζει με τα σύμβολα *cb*, χρησιμοποιούμε την αρχή παραγωγής $C \rightarrow cb$. Αυτή μας δίνει $S \Rightarrow Ab \Rightarrow CaB \Rightarrow cbaB$. Τελειώνουμε αν χρησιμοποιήσουμε την αρχή παραγωγής $B \rightarrow b$, έτσι ώστε να πάρουμε την $S \Rightarrow AB \Rightarrow CaB \Rightarrow cbaB \Rightarrow cbab$. Η προσέγγιση που χρησιμοποιήσαμε ονομάζεται **συντακτική ανάλυση από επάνω προς τα κάτω**, επειδή ξεκινάει με το σύμβολο εκκίνησης και προχωράει με διαδοχική εφαρμογή παραγωγών.

Υπάρχει και μια άλλη προσέγγιση στο πρόβλημα αυτό, που ονομάζεται **συντακτική ανάλυση από κάτω προς τα επάνω**. Στην προσέγγιση αυτή εργαζόμαστε προς τα πίσω. Επειδή πρόκειται να παραχθεί η συμβολοσειρά *cbab*, μπορούμε να χρησιμοποιήσουμε την αρχή παραγωγής $C \rightarrow cb$, έτσι ώστε να είναι $Cab \Rightarrow cbab$. Υστερα, μπορούμε να χρησιμοποιήσουμε την δημιουργία $A \rightarrow Ca$, έτσι ώστε να είναι $Ab \Rightarrow Cab \Rightarrow cbab$. Η χρήση της δημιουργίας $B \rightarrow b$ δίνει $AB \Rightarrow Ab \Rightarrow Cab \Rightarrow cbab$. Τέλος, η χρήση της $S \rightarrow AB$ δείχνει ότι μια πλήρης παραγωγή για την *cbab* είναι η $S = AB \Rightarrow Ab \Rightarrow Cab \Rightarrow cbab$.

ΜΟΡΦΗ BACKUS-NAUR

Για τον καθορισμό γραμματικής τύπου 2 υπάρχει και ένας άλλος συμβολισμός που χρησιμοποιείται μερικές φορές, και ονομάζεται **μορφή Backus-Naur** (Backus-Naur form, **BNF**), από τον John Backus, που την ανακάλυψε, και από τον Peter Naur, που την βελτίωσε για να χρησιμοποιηθεί στις προδιαγραφές της γλώσσας προγραμματισμού ALGOL. (Κατά παράδοξο τρόπο, ένας συμβολισμός που έμοιαζε αρκετά με τον συμβολισμό Backus-Naur χρησιμοποιούνταν πριν από περίπου 2500 χρόνια για να περιγράφει την γραμματική στα Σανσκριτικά.) Η μορφή Backus-Naur χρησιμοποιείται για να καθορίζει τους συντακτικούς κανόνες πολλών γλωσσών υπολογιστών, περιλαμβανομένης της γλώσσας Java. Οι αρχές παραγωγής σε γραμματική τύπου 2 έχουν σαν αριστερή τους πλευρά ένα μόνο μη τερματικό σύμβολο. Αντί να καταγράφουμε ξεχωριστά όλες τις αρχές παραγωγής, μπορούμε να τις συνδυάσουμε όλες με το ίδιο μη τερματικό σύμβολο στην αριστερή πλευρά σε μια δήλωση. Αντί να χρησιμοποιούμε σε παραγωγή το σύμβολο $\rightarrow$ χρησιμοποιούμε το σύμβολο ::=. Περικλείουμε όλα τα μη τερματικά σύμβολα σε αγκύλες, $\langle\ \rangle$, και καταγράφουμε όλες τις δεξιές πλευρές των αρχών παραγωγής στην ίδια δήλωση, διαχωρίζοντάς τις με κατακόρυφες ευθείες.

Για παράδειγμα, οι παραγωγές $A \to Aa$, $A \to a$, και $A \to AB$ συνδυάζονται στην $\langle A \rangle ::= \langle A \rangle a \mid a \mid \langle A \rangle \langle B \rangle$.

Στο Παράδειγμα 13 φαίνεται ο τρόπος χρήσης της μορφής Backus-Naur για την περιγραφή της σύνταξης γλωσσών προγραμματισμού. Το παράδειγμά μας προέρχεται από την αρχική χρήση της μορφής Backus-Naur στην περιγραφή της ALGOL 60.

🕮 *ΠΑΡΑΔΕΙΓΜΑ 13*

Στην ALGOL 60 ο identifier (αναγνωριστικό, που είναι το όνομα μιάς οντότητας όπως η μεταβλητή) αποτελείται από μια συμβολοσειρά αλφαριθμητικών χαρακτήρων (δηλαδή, γραμμάτων και ψηφίων) και θα πρέπει να αρχίζει με γράμμα. Χρησιμοποιούμε αυτούς τους κανόνες στην μορφή Backus-Naur για να περιγράψουμε το σύνολο των identifier που επιτρέπονται:

$\langle identifier \rangle ::= \langle \gamma\rho\acute{\alpha}\mu\mu\alpha \rangle \mid \langle identifier \rangle \langle \gamma\rho\acute{\alpha}\mu\mu\alpha \rangle \mid \langle identifier \rangle \langle \psi\eta\phi\acute{\iota}o \rangle$

$\langle \gamma\rho\acute{\alpha}\mu\mu\alpha \rangle ::= a \mid b \mid \cdots \mid y \mid z$ οι τρεις τελείες σημαίνουν ότι περιλαμβάνονται και τα 26 γράμματα του Λατινικού αλφάβητου

$\langle \psi\eta\phi\acute{\iota}o \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

JOHN BACKUS (1924 -) Ο John Backus γεννήθηκε στην Φιλαδέλφεια της Πολιτείας Pennsylvania των ΗΠΑ και μεγάλωσε στο Wilmington της Πολιτείας Delaware. Παρακολούθησε το Hill School στο Pottstown της Πολιτείας Pennsylvania. Κάθε χρόνο έπρεπε να παρακολουθεί θερινό σχολείο επειδή δεν του άρεσε η μελέτη και δεν ήταν σοβαρός μαθητής. Αλλά του άρεσε να περνά τα καλοκαίρια του στην Πολιτεία New Hamshire όπου παρακολουθούσε το θερινό σχολείο και διασκέδαζε με τις καλοκαιρινές δραστηριότητες, όπως η ιστιοπλοΐα. Εκανε μεγάλη χάρη στον πατέρα του με το να εγγραφεί στο Πανεπιστήμιο της Virginia για να σπουδάσει Χημεία. Γρήγορα, όμως, αποφάσισε ότι η Χημεία δεν ήταν γι' αυτόν, και το 1943 κατατάγηκε στον στρατό, όπου πήρε ιατρική εκπαίδευση και εργάστηκε σε θάλαμο νευροχειρουργικής σε στρατιωτικό νοσοκομείο. Σύντομα του έγινε διάγνωση για έναν όγκο στα οστά του κρανίου όπου του τοποθετήθηκε μια μεταλλική πλάκα. Η ιατρική του εργασία στον στρατό τον έπεισε να δοκιμάσει την Ιατρική Σχολή, αλλά την εγκατέλειψε μετά από εννέα μήνες επειδή δεν του άρεσε την μηχανική αποστήθιση που χρειάζονταν. Μετά την παραίτησή του από την Ιατρική Σχολή, μπήκε σε ένα σχολείο για τεχνικούς ραδιοφώνου επειδή ήθελε να κατασκευάσει ο ίδιος το δικό του στερεοφωνικό συγκρότημα. Ενας καθηγητής στο σχολείο αυτό αναγνώρισε τις δυνατότητές του και του ζήτησε βοήθεια για κάποιους μαθηματικούς υπολογισμούς που χρειάζονταν για ένα άρθρο σε περιοδικό. Τελικά, ο Backus ανακάλυψε ότι ενδιαφερόταν για τα μαθηματικά και τις εφαρμογές τους. Γράφτηκε στο Πανεπιστήμιο Columbia, απ' όπου πήρε πτυχίο και μεταπτυχιακό δίπλωμα στα μαθηματικά. Προσλήφθηκε στην IBM σαν προγραμματιστής το 1950. Συμμετείχε στην σχεδίαση και ανάπτυξη σε δύο από τους πρώτους υπολογιστές της IBM. Από το 1954 μέχρι το 1958 ήταν επικεφαλής της ομάδας της IBM που ανέπτυξε την γλώσσα FORTRAN. Το 1958 έγινε μέλος του προσωπικού στο Watson Research Center της IBM. Ηταν μέλος της επιτροπής που σχεδίασε την γλώσσα προγραμματισμού ALGOL, και χρησιμοποίησε αυτό που σήμερα ονομάζουμε μορφή Backus-Naur για την περιγραφή της σύνταξης αυτής της γλώσσας. Αργότερα, ο Backus εργάστηκε στα μαθηματικά των οικογενειών συνόλων και σε ένα λειτουργικό στύλ του προγραμματισμού. Έγινε εταίρος της IBM το 1963, και το 1974 πήρε το Εθνικό Μετάλλιο Επιστημών των ΗΠΑ, και το 1977 το περίφημο Βραβείο Turing από τον Σύνδεσμο Υπολογιστικών Μηχανών των ΗΠΑ.

Για παράδειγμα, δημιουργούμε τον έγκυρο identifier $x99a$ με χρήση του πρώτου κανόνα αντικατάστασης του $\langle identifier \rangle$ από το $\langle identifier \rangle \langle \gamma\rho\acute{\alpha}\mu\mu\alpha \rangle$, του δεύτερου κανόνα ώστε να πάρουμε το $\langle identifier \rangle a$, του πρώτου κανόνα δύο φορές για να πάρουμε το $\langle identifier \rangle \langle \psi\eta\phi\acute{\iota}o \rangle \langle \psi\eta\phi\acute{\iota}o \rangle a$, του τρίτου κανόνα δύο φορές για να πάρουμε το $\langle identifier \rangle 99a$, του πρώτου κανόνα για να πάρουμε το $\langle \gamma\rho\acute{\alpha}\mu\mu\alpha \rangle 99a$, και τέλος του δεύτερου κανόνα για να πάρουμε το $x99a$.

ΠΑΡΑΔΕΙΓΜΑ 14

Ποιά είναι η μορφή Backus-Naur της γραμματικής για το υποσύνολο στα Αγγλικά που περιγράφεται στην εισαγωγή αυτής της παραγράφου;

Λύση: Η μορφή Backus-Naur αυτής της γραμματικής είναι

$\langle \pi\rho\acute{o}\tau\alpha\sigma\eta \rangle ::= \langle \varphi\rho\acute{\alpha}\sigma\eta\ o\upsilon\sigma\iota\alpha\sigma\tau\iota\kappa o\acute{\upsilon} \rangle \langle \varphi\rho\acute{\alpha}\sigma\eta\ \rho\acute{\eta}\mu\alpha\tau o\varsigma \rangle$

$\langle \varphi\rho\acute{\alpha}\sigma\eta\ o\upsilon\sigma\iota\alpha\sigma\tau\iota\kappa o\acute{\upsilon} \rangle ::=$

$\langle \acute{\alpha}\rho\theta\rho o \rangle \langle \epsilon\pi\acute{\iota}\theta\epsilon\tau o \rangle \langle o\upsilon\sigma\iota\alpha\sigma\tau\iota\kappa\acute{o} \rangle \mid \langle \acute{\alpha}\rho\theta\rho o \rangle \langle o\upsilon\sigma\iota\alpha\sigma\tau\iota\kappa\acute{o} \rangle$

$\langle \varphi\rho\acute{\alpha}\sigma\eta\ \rho\acute{\eta}\mu\alpha\tau o\varsigma \rangle ::= \langle \rho\acute{\eta}\mu\alpha \rangle \langle \epsilon\pi\acute{\iota}\rho\rho\eta\mu\alpha \rangle \mid \langle \rho\acute{\eta}\mu\alpha \rangle$

$\langle \acute{\alpha}\rho\theta\rho o \rangle ::= a \mid the$

$\langle \epsilon\pi\acute{\iota}\theta\epsilon\tau o \rangle ::= large \mid hungry$

$\langle o\upsilon\sigma\iota\alpha\sigma\tau\iota\kappa\acute{o} \rangle ::= rabbit \mid mathematician$

$\langle \rho\acute{\eta}\mu\alpha \rangle ::= eats \mid hops$

$\langle \epsilon\pi\acute{\iota}\rho\rho\eta\mu\alpha \rangle ::= quickly \mid wildly$

ΠΑΡΑΔΕΙΓΜΑ 15

Να δοθεί η μορφή Backus-Naur της δημιουργίας ακεραίων με πρόσημο σε

PETER NAUR (1928 -) Ο Peter Naur γεννήθηκε στο Frederiksberg κοντά στην Κοπεγχάγη. Σαν παιδί τον ενδιέφερε η αστρονομία. Οχι μόνο παρατηρούσε τα ουράνια σώματα, αλλά υπολόγιζε και τις τροχιές των κομητών και των αστεροειδών. Παρακολούθησε το Πανεπιστήμιο της Κοπεγχάγης, απ' όπου πήρε το πτυχίο του το 1949. Πέρασε το 1950 και το 1951 στο Cambridge, όπου χρησιμοποιούσε έναν πρώϊμο υπολογιστή για να υπολογίζει τις κινήσεις των κομητών και των πλανητών. Μετά την επιστροφή του στην Δανία, συνέχισε να εργάζεται στην αστρονομία, αλλά διατήρησε του δεσμούς του με τους υπολογιστές. Το 1955 υπηρέτησε σαν σύμβουλος στην κατασκευή του πρώτου Δανέζικου υπολογιστή. Το 1959 ο Naur έκανε την στροφή από την αστρονομία στους υπολογιστές σαν πλήρη δραστηριότητα. Η πρώτη του εργασία σαν επιστήμονα υπολογιστών ήταν η συμμετοχή στην ανάπτυξη της γλώσσας προγραμματισμού ALGOL. Από το 1960 μέχρι το 1967 εργάστηκε στην ανάπτυξη μεταγλωττιστών για την ALGOL και για την COBOL. Το 1960 έγινε καθηγητής της επιστήμης υπολογιστών στο Πανεπιστήμιο της Κοπεγχάγης, όπου εργάστηκε στον τομέα της μεθοδολογίας προγραμματισμού.

δεκαδική παράσταση. (**Ακέραιος με πρόσημο** είναι ο μη αρνητικός ακέραιος πριν από τον οποίο υπάρχει θετικό ή αρνητικό πρόσημο.)

Λύση: Η μορφή Backus-Naur για γραμματική που δημιουργεί ακέραιους με πρόσημο είναι η

$$\langle \textit{ακέραιος με πρόσημο} \rangle ::= \langle \textit{πρόσημο} \rangle \langle \textit{ακέραιος} \rangle$$

$$\langle \textit{πρόσημο} \rangle ::= + \mid -$$

$$\langle \textit{ακέραιος} \rangle ::= \langle \textit{ψηφίο} \rangle \mid \langle \textit{ψηφίο} \rangle \langle \textit{ακέραιος} \rangle$$

$$\langle \textit{ψηφίο} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

Η μορφή Backus-Naur, με ποικιλία προεκτάσεων, χρησιμοποιείται σε μεγάλο βαθμό στον καθορισμό της σύνταξης γλωσσών προγραμματισμού, όπως η γλώσσα Java και η γλώσσα LISP, και γλωσσών όπως η XML. Κάποιες προεκτάσεις της μορφής Backus-Naur που χρησιμοποιούνται συνήθως στην περιγραφή γλωσσών προγραμματισμού παρουσιάζονται στον πρόλογο της Ασκησης 28 στο τέλος αυτής της παραγράφου.

Ασκήσεις

Οι Ασκήσεις 1-3 αναφέρονται στην γραμματική με σύμβολο εκκίνησης **πρόταση**, σύνολο τερματικών $T = \{o, \textit{κοιμισμένος}, \textit{ευτυχισμένος}, \textit{χελώνα}, \textit{λαγός}, \textit{ξεπερνάει}, \textit{τρέχει}, \textit{γρήγορα}, \textit{αργά}\}$, σύνολο μη τερματικών $N = \{$**φράση ουσιαστικών**, **μεταβατική φράση ρήματος**, **μη μεταβατική (αμετάβατη) φράση ρήματος**, **άρθρο**, **επίθετο**, **ουσιαστικό**, **ρήμα**, **επίρρημα**$\}$, και αρχές παραγωγής:

πρόταση → **φράση ουσιαστικού** **μεταβατική φράση ρήματος** **φράση ουσιαστικού**
πρόταση → **φράση ουσιαστικού** **μη μεταβατική φράση ρήματος**
φράση ουσιαστικού → **άρθρο** **επίθετο** **ουσιαστικό**
φράση ουσιαστικού → **άρθρο** **ουσιαστικό**
μεταβατική φράση ρήματος → **μεταβατικό ρήμα**
μη μεταβατική φράση ρήματος → **μη μεταβατικό ρήμα** **επίρρημα**
μη μεταβατική φράση ρήματος → **μη μεταβατικό ρήμα**
άρθρο → *o*
επίθετο → *κοιμισμένος*
επίθετο → *ευτυχισμένος*
ουσιαστικό → *χελώνα*
ουσιαστικό → *λαγός*
μεταβατικό ρήμα → *ξεπερνάει*

μη μεταβατικό ρήμα → *τρέχει*
επίρρημα → *γρήγορα*
επίρρημα → *αργά*

1. Με χρήση του συνόλου των δημιουργιών να δειχτεί ότι οι παρακάτω προτάσεις είναι έγκυρες.
a) *ο ευτυχισμένος λαγός τρέχει* **b)** *η κοιμισμένη χελώνα τρέχει γρήγορα*
c) *η χελώνα ξεπερνάει τον λαγό*
d) *ο κοιμισμένος λαγός ξεπερνάει την ευτυχισμένη χελώνα*

2. Να βρεθούν ακόμη άλλες πέντε έγκυρες προτάσεις, εκτός από τις προτάσεις της Ασκησης 1.

3. Να δειχτεί ότι η *ο λαγός τρέχει την κοιμισμένη χελώνα* δεν είναι έγκυρη πρόταση.

***4.** Εστω ότι $V = \{S, A, B, a, b\}$ και $T = \{a, b\}$. Να βρεθεί η γλώσσα που γεννιέται από την γραμματική $\{V, T, S, P\}$ όταν το σύνολο P των αρχών παραγωγής αποτελείται από τις
a) $S \to AB$, $A \to ab$, $B \to bb$.
b) $S \to AB$, $S \to aA$, $A \to a$, $B \to ba$.
c) $S \to AB$, $S \to AA$, $A \to aB$, $A \to ab$, $B \to b$.
d) $S \to AA$, $S \to B$, $A \to aaA$, $A \to aa$, $B \to bB$, $B \to b$.
e) $S \to AB$, $A \to aAb$, $B \to bBa$, $A \to \lambda$, $B \to \lambda$.

5. Να κατασκευαστεί παραγωγή του $0^3 1^3$ με χρήση της γραμματικής του Παραδείγματος 5.

6. Να δειχτεί ότι η γραμματική του Παραδείγματος 5 γεννάει το σύνολο $\{0^n 1^n \mid n = 0, 1, 2, \ldots\}$.

7. a) Να κατασκευαστεί παραγωγή του $0^2 1^4$ με χρήση της γραμματικής G_1 του Παραδείγματος 6.
b) Να κατασκευαστεί παραγωγή του $0^2 1^4$ με χρήση της γραμματικής G_2 του Παραδείγματος 6.

8. a) Να δειχτεί ότι η γραμματική G_1 του Παραδείγματος 6 γεννάει το σύνολο $\{0^m 1^n \mid m, n = 0, 1, 2, \ldots\}$.
b) Να δειχτεί ότι η γραμματική G_2 του Παραδείγματος 6 γεννάει το ίδιο σύνολο.

9. Να κατασκευαστεί παραγωγή του $0^2 1^2 2^2$ με χρήση της γραμματικής του Παραδείγματος 7.

***10.** Να δειχτεί ότι η γραμματική του Παραδείγματος 7 γεννάει το σύνολο $\{0^n 1^n 2^n \mid m, n = 0, 1, 2, \ldots\}$.

***11.** Να βρεθεί γραμματική δομής φράσεων για τις παρακάτω γλώσσες.
a) το σύνολο όλων των συμβολοσειρών bit που περιέχουν άρτιο πλήθος από 0 και κανένα 1.
b) το σύνολο όλων των συμβολοσειρών bit που αποτελούνται από ένα 1 που ακολουθείται από περιττό πλήθος 0.

c) το σύνολο όλων των συμβολοσειρών bit που περιέχουν άρτιο πλήθος από 0 και άρτιο πλήθος από 1.

d) το σύνολο όλων των συμβολοσειρών που περιέχουν 10 ή περισσότερα 0 και κανένα 1.

e) το σύνολο όλων των συμβολοσειρών που περιέχουν περισσότερα 0 από 1.

f) το σύνολο όλων των συμβολοσειρών που περιέχουν ίσο πλήθος 0 από 1.

g) το σύνολο όλων των συμβολοσειρών που περιέχουν άνισο πλήθος 0 από 1.

12. Να κατασκευαστούν γραμματικές δομής φράσεων που γεννούν τα παρακάτω σύνολα.

a) $\{01^{2^n} \mid n \geq 0\}$ **b)** $\{0^n 1^{2^n} \mid n \geq 0\}$ **c)** $\{0^n 1^m 0^n \mid m \geq 0, n \geq 0\}$

13. Εστω ότι $V = \{S, A, B, a, b\}$ και $T = \{a, b\}$. Να προσδιοριστεί αν η $G = \{V, T, S, P\}$ είναι γραμματική τύπου 0 αλλά όχι γραμματική τύπου 1, γραμματική τύπου 1 αλλά όχι γραμματική τύπου 2, ή γραμματική τύπου 2 αλλά όχι γραμματική τύπου 3, αν το P, το σύνολο των αρχών παραγωγής, είναι

a) $S \to aAB$, $A \to Bb$, $B \to \lambda$.

b) $S \to aA$, $A \to a$, $A \to b$.

c) $S \to ABa$, $AB \to a$.

d) $S \to ABa$, $A \to aB$, $B \to ab$.

e) $S \to bA$, $A \to B$, $B \to a$.

f) $S \to Aa$, $aA \to B$, $B \to aA$, $A \to b$.

g) $S \to ba$, $A \to b$, $S \to \lambda$.

h) $S \to AB$, $B \to aAb$, $aAb \to b$.

i) $S \to aA$, $A \to bB$, $B \to b$, $B \to \lambda$.

j) $S \to A$, $A \to B$, $B \to \lambda$.

14. **Παλίνδρομη** είναι η συμβολοσειρά που διαβάζεται με τον ίδιο τρόπο από δεξιά προς τα αριστερά, όπως και από αριστερά προς τα δεξιά, δηλ., συμβολοσειρά w όπου $w=w^R$, όπου w^R είναι η αντιστροφή της συμβολοσειράς w. Να βρεθεί γραμματική ελεύθερη συμφραζόμενων που γεννάει το σύνολο όλων παλίνδρομων συμβολοσειρών επί του αλφάβητου $\{0,1\}$.

***15.** Εστω ότι οι G_1 και G_2 είναι γραμματικές ελεύθερες συμφραζόμενων, που γεννούν τις γλώσσες $L(G_1)$ και $L(G_2)$, αντίστοιχα. Να δειχτεί ότι υπάρχει γραμματική ελεύθερη συμφραζόμενων που γεννάει τα παρακάτω σύνολα.

a) $L(G_1) \cup L(G_2)$ **b)** $L(G_1)L(G_2)$ **c)** $L(G_1)^*$

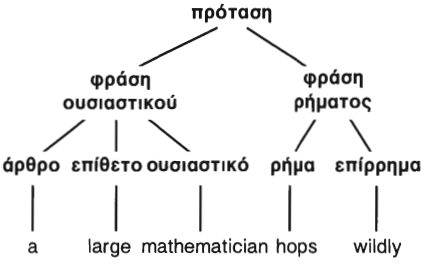
16. Να βρεθούν οι συμβολοσειρές που κατασκευάζονται με χρήση των δένδρων παραγωγής που φαίνονται στα σχήματα της επόμενης σελίδας.

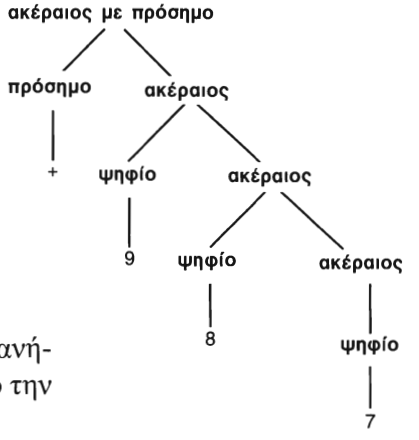
17. Να κατασκευαστούν δένδρα παραγωγής για τις προτάσεις της Ασκησης 1.

18. Εστω ότι G είναι η γραμματική με $V = \{a, b, c, S\}$, $T = \{a, b, c\}$, σύμβολο εκκίνησης S, και αρχές παραγωγής $S \to abS$, $S \to bcS$, $S \to bbS$, $S \to a$, $S \to cB$. Να κατασκευαστούν δένδρα παραγωγής για τις

a) *bcbba* **b)** *bbbcbba* **c)** *bcabbbbcb*.

***19.** Με χρήση συντακτικής ανάλυσης από επάνω προς τα κάτω να προσδιορι-





στεί αν οι παρακάτω συμβολοσειρές ανήκουν στην γλώσσα που γεννιέται από την γραμματική του Παραδείγματος 12.

a) *baba* **b)** *abab* **c)** *cbaba* **d)** *bbbcba*

***20.** Με χρήση συντακτικής ανάλυσης από κάτω προς τα επάνω να προσδιοριστεί αν οι συμβολοσειρές της Ασκησης 19 ανήκουν στην γλώσσα που γεννιέται από την γραμματική του Παραδείγματος 12.

21. Να κατασκευαστεί δένδρο παραγωγής για την −109 με χρήση της γραμματικής του Παραδείγματος 15.

22. a) Να εξηγηθεί ποιές είναι οι αρχές παραγωγής σε γραμματική αν η μορφή Backus-Naur για αρχές παραγωγής είναι:

$$\begin{aligned}\langle \textit{έκφραση} \rangle \quad &::= (\langle \textit{έκφραση} \rangle) \mid \\ &\langle \textit{έκφραση} \rangle + \langle \textit{έκφραση} \rangle \mid \\ &\langle \textit{έκφραση} \rangle * \langle \textit{έκφραση} \rangle \mid \\ &\langle \textit{μεταβλητή} \rangle \\ \langle \textit{μεταβλητή} \rangle \quad &::= x \mid y\end{aligned}$$

b) Να βρεθεί δένδρο παραγωγής για την $(x * y) + x$ στην γραμματική αυτή.

23. a) Να κατασκευαστεί γραμματική δομής φράσεων που γεννάει όλους του δεκαδικούς αριθμούς με πρόσημο, οι οποίοι αποτελούνται από πρόσημο, είτε το + είτε το −, από μη αρνητικό ακέραιο, και από δεκαδικό κλάσμα που είναι είτε η κενή συμβολοσειρά είτε υποδιαστολή που ακολουθείται από θετικό ακέραιο, όπου επιτρέπονται τα μηδενικά στην αρχή ακέραιου.

b) Να δοθεί η μορφή Backus-Naur αυτής της γραμματικής.

c) Να κατασκευαστεί δένδρο παραγωγής για τον −31,4 στην γραμματική αυτή.

24. a) Να κατασκευαστεί γραμματική δομής φράσεων για το σύνολο όλων των κλασμάτων που έχουν την μορφή *a*/*b*, όπου ο *a* είναι ακέραιος με πρόσημο σε δεκαδική παράσταση και ο *b* είναι θετικός ακέραιος.

b) Ποιά είναι η μορφή Backus-Naur αυτής της γραμματικής.

c) Να κατασκευαστεί δένδρο παραγωγής για τον +311/17 στην γραμματική αυτή.

25. Να δοθούν κανόνες δημιουργίας σε μορφή Backus-Baur για identifier αν αυτός μπορεί να αποτελείται από

a) ένα ή περισσότερα πεζά γράμματα.

b) τουλάχιστο τρία αλλά όχι περισσότερα από έξη πεζά γράμματα.

c) ένα μέχρι έξη κεφαλαία ή πεζά γράμματα που αρχίζουν με κεφαλαίο γράμμα.

d) πεζό γράμμα, που ακολουθείται από ψηφίο ή υπογράμμιση, που ακολουθείται από τρεις ή τέσσερεις αλφαριθμητικούς χαρακτήρες (πεζά ή κεφαλαία γράμματα και ψηφία).

26. Να δοθούν κανόνες δημιουργίας σε μορφή Backus-Naur για το όνομα ατόμου, αν αυτό το όνομα αποτελείται από μικρό όνομα, που είναι συμβολοσειρά γραμμάτων, όπου μόνο το πρώτο γράμμα είναι κεφαλαίο, από το αρχικό του πατρικού μικρού ονόματος, και επώνυμο, το οποίο μπορεί να είναι οποιαδήποτε συμβολοσειρά γραμμάτων.

27. Να δοθούν κανόνες δημιουργίας σε μορφή Backus-Naur οι οποίοι γεννούν όλους τους identifier στην γλώσσα προγραμματισμού C. Στην C ο identifier αρχίζει με γράμμα ή με υπογράμμιση (_) που ακολουθείται από ένα ή περισσότερα πεζά γράμματα, κεφαλαία γράμματα, υπογραμμίσεις, και ψηφία.

Συνήθως χρησιμοποιούνται αρκετές επεκτάσεις της μορφής Backus-Naur για τον ορισμό γραμματικών δομής φράσεων. Σε μια τέτοια επέκταση, το ερωτηματικό (?) δείχνει ότι το σύμβολο, ή η ομάδα συμβόλων μέσα στην παρένθεση, που βρίσκεται στα αριστερά του μπορεί να εμφανίζεται καμμία ή μια φορά (είναι δηλαδή προαιρετικό), ο αστερίσκος (*) δείχνει ότι το σύμβολο στα αριστερά του μπορεί να εμφανίζεται καμμία ή περισσότερες φορές, και το συν (+) δείχνει ότι το σύμβολο στα αριστερά του μπορεί να εμφανίζεται καμία ή περισσότερες φορές. Οι επεκτάσεις αυτές αποτελούν μέρος της **επεκταμένης μορφής Backus-Naur** (extended BNF, **EBNF**), και τα σύμβολα ?, *, και + ονομάζονται **μεταχαρακτήρες**. Στην EBNF συνήθως δεν φαίνονται οι αγκύλες που συμβολίζουν μη τερματικά.

28. Να περιγραφεί το σύνολο συμβολοσειρών που ορίζονται από τα παρακάτω σύνολα αρχών παραγωγής σε EBNF.

a) *συμβολοσειρά* ::= $L+D?L+$

$L ::= a \mid b \mid c$

$D ::= 0 \mid 1$

b) *συμβολοσειρά* ::= *πρόσημο* $D+ \mid D+$

πρόσημο ::= $+ \mid -$

$D ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

c) *συμβολοσειρά* ::= $L*(D+)?L*$

$L ::= x \mid y$

$D ::= 0 \mid 1$

29. Να δοθούν κανόνες αρχών παραγωγής σε επεκταμένη μορφή Backus-Naur, οι οποίοι γεννούν όλους τους δεκαδικούς αριθμούς που αποτελούνται από προαιρετικό πρόσημο, από μη αρνητικό ακέραιο, και από δεκαδικό κλάσμα που είναι είτε η κενή συμβολοσειρά είτε υποδιαστολή που ακολουθείται από προαιρετικό θετικό ακέραιο, πριν από τον οποίο υπάρχει προαιρετικά κάποιο πλήθος από μηδενικά.

30. Να δοθούν κανόνες αρχών παραγωγής σε επεκταμένη μορφή Backus-Naur οι οποίοι γεννούν ένα σάντουιτς, αν το σάντουιτς αποτελείται από κάτω φέτα ψωμιού, από μουστάρδα ή μαγιονέζα, από προαιρετικό μαρούλι, από προαιρετική φέτα ντομάτας, από μια ή περισσότερες φέτες γαλοπούλας, κοτόπουλου, ή ψημμένου κρέατος (με οποιονδήποτε συνδυασμό), από μερικές προαιρετικές φέτες τυριού, και από επάνω φέτα ψωμιού.

31. Να δοθούν κανόνες αρχών παραγωγής σε επεκταμένη μορφή Backus-Naur για identifier στην γλώσσα προγραμματισμού C (βλ. Ασκηση 27).

32. Να εξηγηθεί ο τρόπος με τον οποίο μεταφράζονται αρχές παραγωγής για γραμματική σε EBNF, σε σύνολο δημιουργιών για την γραμματική σε BNF.

Παρακάτω δίνεται η μορφή Backus-Naur που περιγράφει την σύνταξη εκφράσεων σε συμβολισμό μεταθέματος (ή ανάστροφο Πολωνικό συμβολισμό).

$\langle \textit{έκφραση} \rangle ::= \langle \textit{όρος} \rangle \mid \langle \textit{όρος} \rangle \ \langle \textit{όρος} \rangle \ \langle \textit{τελεστής πρόσθεσης} \rangle$

$\langle \textit{τελεστής πρόσθεσης} \rangle ::= + \mid -$

$\langle \textit{όρος} \rangle ::=$

$\langle \textit{παράγοντας} \rangle \mid \langle \textit{παράγοντας} \rangle \ \langle \textit{παράγοντας} \rangle \ \langle \textit{τελεστής πολλαπλασιασμού} \rangle$

$\langle \textit{τελεστής πολλαπλασιασμού} \rangle ::= * \mid /$

$\langle \textit{παράγοντας} \rangle ::= \langle \textit{identifier} \rangle \mid \langle \textit{έκφραση} \rangle$

$\langle \textit{identifier} \rangle ::= a \mid b \mid \cdots \mid z$

33. Στις παρακάτω συμβολοσειρές, να προσδιοριστεί αν αυτές γεννιούνται με την γραμματική που δίνεται για συμβολισμό μεταθέματος. Αν ναι, να βρεθούν τα βήματα γέννησης της συμβολοσειράς.

a) $abc*+$ **b)** $xy++$ **c)** $xy-z*$ **d)** $wxyz-*/$ **e)** $ade-*$

34. Με χρήση μορφής Backus-Naur να περιγραφεί η σύνταξη εκφράσεων με συμβολισμό ενθέματος, όπου το σύνολο των τελεστών και identifier είναι το ίδιο όπως στην BNF για συμβολισμό μετάθεσης που δίνεται στον πρόλογο της Ασκησης 33, αλλά όπου θα πρέπει να υπάρχουν παρενθέσεις σε εκφράσεις που χρησιμοποιούνται σαν παράγοντες.

35. Στις παρακάτω συμβολοσειρές, να προσδιοριστεί αν αυτές γεννιούνται με την γραμματική που δίνεται για συμβολισμό ενθέματος από την Ασκηση 34. Αν ναι, να βρεθούν τα βήματα γέννησης της συμβολοσειράς.