

Κεφάλαιο 3: Γλώσσες Τύπου 2 (Γλώσσες Χωρίς Συμφραζόμενα)

3.1 Βασικές Έννοιες

Προς το τέλος του κεφαλαίου 2 είδαμε ότι οι κανονικές γλώσσες είναι πολύ περιορισμένες όσον αφορά την παραγωγή/αποδοχή γλωσσών προγραμματισμού. Τώρα κάνουμε ένα βήμα προς πιο εκφραστικές γλώσσες. Έχουμε ήδη δει ότι μία απλή γλώσσα προγραμματισμού μπορεί να παραχθεί από γραμματική τύπου 2 (βλέπε παράδειγμα 7, σελίδα 1-3).

Γραμματική Χωρίς Συμφραζόμενα

$A ::= u$, όπου $A \in N$, $u \in (N \cup T)^*$

Θεώρημα 3.1 Έστω $G=(N, T, P, S)$ μία γραμματική χωρίς συμφραζόμενα. Τότε υπάρχει μία ισοδύναμη γραμματική $G'=(N', T', P', S')$ τέτοια ώστε:

1. $(A ::= \varepsilon) \in P' \Rightarrow A=S'$
2. $(A ::= u) \in P', A \neq S' \Rightarrow u \neq \varepsilon$

Απόδειξη

Έστω $G=(N, T, P, S)$ και $S' \notin N$.

1. Ορίζουμε $G''=(N \cup \{S'\}, T, P'', S')$, όπου:
 $P''=P \cup \{S' ::= S\}$, αν $\varepsilon \notin L(G)$
 $P''=P \cup \{S' ::= S|\varepsilon\}$, αν $\varepsilon \in L(G)$
 Ισχύει: $L(G'')=L(G)$
2. Σημειώνουμε όλα τα μη τερματικά $A \in N \cup \{S'\}$ με $A \rightarrow^* \varepsilon$ ως εξής:
 $\checkmark \Sigma_0 := \{A \in N \cup \{S'\} \mid (A ::= \varepsilon) \in P''\}$
 $\checkmark \Sigma_{n+1} := \Sigma_n \cup \{A \in N \cup \{S'\} \mid (A ::= A_1 A_2 \dots A_n) \in P, A \notin \Sigma_n, A_i \in \Sigma_n, i=1,2,\dots,n\}$
 $\checkmark \Sigma_\varepsilon = \bigcup_{i \geq 0} \Sigma_i$
3. Ορίζουμε $G'=(N \cup \{S'\}, T, P', S')$ όπου
 $P'=(P'' \cup \{A ::= uv \mid A ::= uBv \in P'', B \in \Sigma_\varepsilon, uv \neq \varepsilon\}) - \{A ::= \varepsilon \mid A \neq S'\}$
 Ισχύει: $L(G')=L(G'')=L(G)$ και G' έχει την σωστή μορφή

Συντακτικά Δέντρα

Κάθε παραγωγή $A \rightarrow^* u$ μπορεί να παρασταθεί γραφικά σαν ένα δέντρο.

Παράδειγμα

$S ::= A$

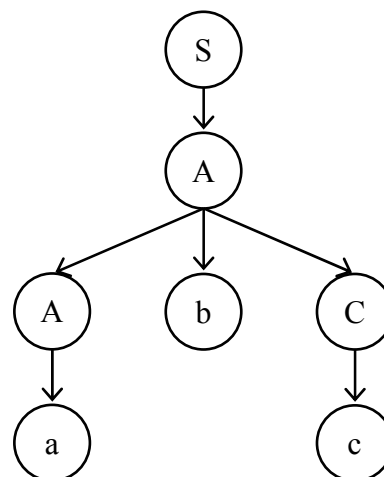
$A ::= AbC \mid a$

$C ::= c$

Παραγωγές:

- $\checkmark S \rightarrow A \rightarrow AbC \rightarrow abC \rightarrow abc$
- $\checkmark S \rightarrow A \rightarrow AbC \rightarrow Abc \rightarrow abc$

Δέντρο:



- Ήδη αυτό το παράδειγμα δείχνει μία αφαιρετική ιδιότητα των συντακτικών δέντρων: ένα δέντρο μπορεί να αναπαραστήσει περισσότερες παραγωγές.

Έστω $A \rightarrow u_1 \rightarrow u_2 \rightarrow \dots \rightarrow u_n = u$

- Η ρίζα του συντακτικού δέντρου είναι το A
- Κάθε «εσωτερικός» κόμβος είναι ένα μη τερματικό σύμβολο B
- Τα παιδιά του B προκύπτουν από τον κανόνα $B ::= v$ που εφαρμόστηκε
- Τα φύλλα, από αριστερά προς τα δεξιά, μας δίνουν το u

Μαθηματικά

Επαγωγή ως προς το μήκος της παραγωγής.

1. Έστω $A \rightarrow^0 A$, $A \in N \cup T$.

Συντακτικό δέντρο:

$dt(A \rightarrow^0 A)$

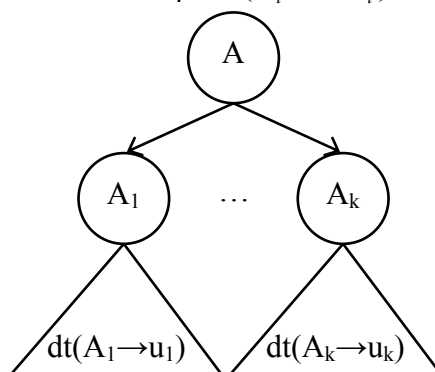


2. $A \rightarrow A_1 \dots A_k \rightarrow^n u$, $A_1, \dots, A_k \in N \cup T$.

✓ $u = u_1 u_2 \dots u_k$, $A_i \rightarrow^{n_i} u_i$, $n_i \leq n$ (*)

✓ Υπάρχουν ήδη συντακτικά δέντρα $dt(A_i \rightarrow^{n_i} u_i)$

Τότε $dt(A \rightarrow^* u)$:



Έστω $height(dt)$ το ύψος ενός συντακτικού δέντρου.

Έστω $res(dt)$ το αποτέλεσμα ενός συντακτικού δέντρου, δηλαδή η λέξη που σχηματίζουν τα φύλλα του.

Λήμμα 3.2 $|res(dt)| \leq b^{height(dt)}$, όπου b είναι το μέγιστο μήκος μίας δεξιάς πλευράς ενός κανόνα.

Στην παραπάνω κατασκευή χρησιμοποιήσαμε το βήμα (*).

Θεώρημα 3.2 Έστω $G = (N, T, P, S)$ γραμματική χωρίς συμφραζόμενα, $u \rightarrow^n v$ παραγωγή μήκους n , $u = u_1 \dots u_k$. Τότε υπάρχουν k παραγωγές $u_i \rightarrow^{n_i} v_i$ έτσι ώστε $v = v_1 \dots v_k$ και $n = \sum n_i$.

Η απόδειξη προκύπτει με επαγωγή στο n και παραλείπεται.

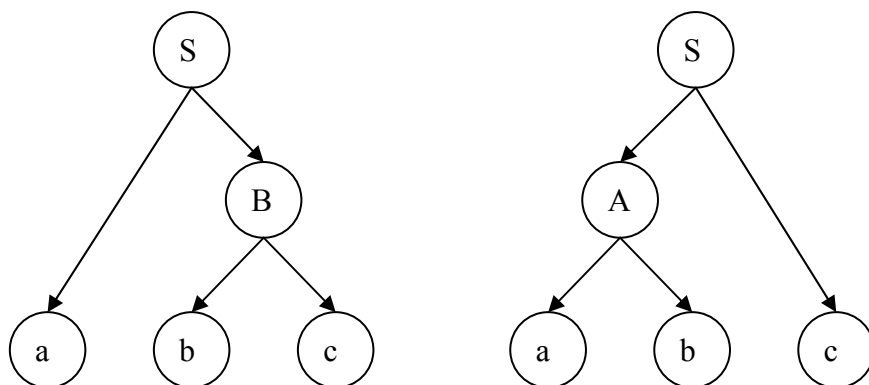
Για μία λέξη είναι δυνατόν να υπάρχουν περισσότερα του ενός συντακτικά δέντρα.

Παράδειγμα:

$S ::= aB \mid Ac$

$A ::= ab$

$B ::= bc$



Γενικά τέτοιες γραμματικές είναι πιο δύσκολες στην πράξη (από αλγοριθμική πλευρά).

Αυτό μπορεί να συμβαίνει επειδή σχηματίσαμε μία όχι καλή γραμματική. Στο παράδειγμά μας μία ισοδύναμη γραμματική είναι: $S ::= abc$. Αλλά υπάρχουν και γλώσσες για τις οποίες όλες οι γραμματικές έχουν αυτή την αρνητική ιδιότητα.

Παράδειγμα:

$L = \{a^i b^j c^k \mid i=j \text{ ή } j=k\}$

(χωρίς απόδειξη)

3.2 Η Κανονική Μορφή του Chomsky

Έστω $G=(N, T, P, S)$ μία γραμματική χωρίς συμφραζόμενα.

Η G έχει την κανονική μορφή του Chomsky αν όλοι οι κανόνες έχουν μία από τις επόμενες δύο μορφές:

1. $A ::= a, a \in T$
2. $A ::= BC, B, C \in N$

Θεώρημα 3.3 Για κάθε γραμματική G τύπου 2 με $\varepsilon \notin L(G)$ υπάρχει ισοδύναμη γραμματική που έχει την κανονική μορφή του Chomsky.

Απόδειξη – Κατασκευή

1. Διαγραφή κανόνων της μορφής $A ::= B, A, B \in N$
 - 1.1. Διαγραφή κύκλων
Για κάθε κύκλο $A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_n \rightarrow A_1$, όπου $A_i \in N$:
 - ✓ Τα A_2, \dots, A_n αντικαθίστώνται από το A_1
 - ✓ Διαγράφονται κανόνες της μορφής $A_1 ::= A_1$
 - 1.2. Αρίθμηση των υπολοίπων μη τερματικών A_1, \dots, A_k έτσι ώστε $A_i ::= A_j \Rightarrow i < j$, δηλαδή όσο πιο μακριά από το τέλος, τόσο πιο μικρό το νούμερό του
 - 1.3. Διαγραφή των λοιπών κανόνων της μορφής $A ::= B$

```

FOR i=k, k-1, ..., 1 DO
  FOR j=i+1, ..., k DO
    IF  $A_i ::= A_j$  THEN
      DELETE  $A_i ::= A_j$ 
      FOR ALL  $A_j ::= x$  DO
        ADD  $A_i ::= x$ 
      END
    END
  END
END
END
END

```

2. Αντικατάσταση $a \in T$ με A_a
 - ✓ Για κάθε $a \in T$ πρόσθεσε νέο μη τερματικό A_a και κανόνες $A_a ::= a$
 - ✓ Αντικατέστησε κάθε a σε δεξιά πλευρά κανόνων μήκους ≥ 2 με A_a (τώρα οι δεξιές πλευρές μήκους ≥ 2 έχουν μόνο μη τερματικά)
3. Διαγραφή κανόνων της μορφής $A ::= B_1 B_2 \dots B_k, k \geq 3$
 Αντικατέστησε κάθε κανόνα $A ::= B_1 B_2 \dots B_k, k \geq 3$ με:
 - ✓ $A ::= B_1 C_1$
 - ✓ $C_1 ::= B_2 C_2$
 - ✓ ...
 - ✓ $C_{k-2} ::= B_{k-1} B_k$
 όπου C_1, \dots, C_{k-2} είναι νέα μη τερματικά σύμβολα

Παράδειγμα 1

$S ::= C$

$C ::= aCb \mid ab$

1. Βήμα 1
 - 1.1. Δεν υπάρχουν κύκλοι
 - 1.2. Κρατάμε την αρίθμηση-συμβολισμό ως έχει
 - 1.3. $S ::= aCb \mid ab$
 $C ::= aCb \mid ab$
 2. $S ::= ACB \mid AB$
 $C ::= ACB \mid AB$
 $A ::= a$
 $B ::= b$
 3. $S ::= AD \mid AB$
 $D ::= CB$
 $C ::= AE \mid AB$
 $E ::= CB$
 $A ::= a$
 $B ::= b$
- (φυσικά απλοποιήσεις είναι δυνατές, πχ $D=E, C=S$)

Παράδειγμα 2

$S ::= A \mid aB \mid aC$

$A ::= B \mid C \mid cAd$

$B ::= S \mid Ba$

$C ::= D \mid c$

$D ::= d \mid dDD$

1. Βήμα 1
 - 1.1. Κύκλος: $S \rightarrow A \rightarrow B \rightarrow S$
 Άρα σβήνουμε A, B :
 $S ::= aS \mid aC \mid C \mid cSd \mid Sa$
 $C ::= D \mid c$
 $D ::= d \mid dDD$
 - 1.2. Κρατάμε την αρίθμηση-συμβολισμό ως έχει
 - 1.3. Πρώτα αντικατάσταση της $C ::= D$ από $C ::= d \mid dDD$
 Μετά αντικατάσταση της $S ::= C$ από $S ::= d \mid dDD$
 Οπότε:

- $$\begin{aligned} S &::= aS \mid aC \mid d \mid dDD \mid cSd \mid Sa \\ C &::= d \mid dDD \mid c \\ D &::= d \mid dDD \end{aligned}$$
2.
$$\begin{aligned} S &::= A_a S \mid A_a C \mid A_d \mid A_d DD \mid A_c S A_d \mid S A_a \\ C &::= A_d \mid A_d DD \mid A_c \\ D &::= A_d \mid A_d DD \\ A_a &::= a \\ A_c &::= c \\ A_d &::= d \end{aligned}$$

3.
$$\begin{aligned} S &::= A_a S \mid A_a C \mid A_d \mid A_d E \mid A_c F \mid S A_a \\ E &::= DD \\ F &::= S A_d \\ C &::= A_d \mid A_d G \mid A_c \\ G &::= DD \\ D &::= A_d \mid A_d H \\ H &::= DD \\ A_a &::= a \\ A_c &::= c \\ A_d &::= d \end{aligned}$$

➤ Παρατήρηση: Στο 1.3 η σειρά που ακολουθείται είναι σημαντική!

Παράδειγμα

Αν η γλώσσα G είναι η:

$S ::= A$

$A ::= B$

$B ::= b$

Οπότε $L(G) = \{b\}$.

Εάν εφαρμόσουμε την κατασκευή πρώτα στο $S ::= A$ προκύπτει:

$S ::= B, A ::= b, B ::= b$, που πάλι περιέχει $S ::= B$.

Αν διαγράψουμε αυτόν τον καινούριο κανόνα τότε μένουν:

$A ::= b, B ::= b$.

Άρα $b \notin L(G)$.

3.3 Λήμμα Αντλησης

Θεώρημα 3.4 (Λήμμα Αντλησης για γλώσσες χωρίς συμφραζόμενα)

Έστω L γλώσσα τύπου 2. Τότε υπάρχει $n \in \mathbb{N}$ έτσι ώστε κάθε $z \in L$ μήκους $\geq n$ μπορεί να γραφεί σαν $z = uvwxy$ με τις εξής ιδιότητες:

1. $|vx| \geq 1$
2. $|vwx| \leq n$
3. $uv^iwx^iy \in L, \forall i \in \mathbb{N}$

Απόδειξη

Έστω $e \notin L$ (αλλιώς εξετάζουμε την γλώσσα $L - \{e\}$).

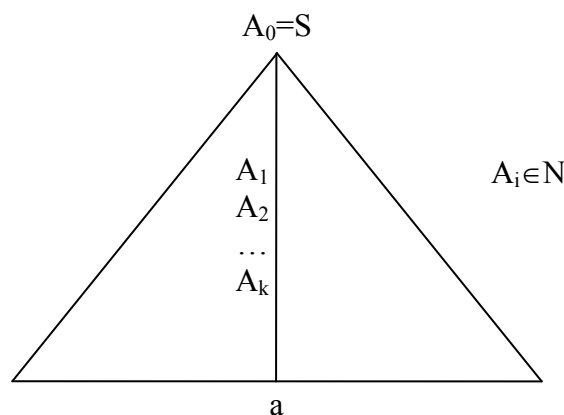
Έστω $L = L(G)$, όπου $G = (N, T, P, S)$ γραμματική χωρίς συμφραζόμενα στην κανονική μορφή Chomsky.

Ορίζουμε: $n := 2^{\#N}$.

Έστω $z \in L$ με $|z| \geq n$. Εξετάζουμε ένα συντακτικό δέντρο για το z . Αυτό πρέπει να έχει $|z| \geq n$ φύλλα.

Λόγω της μορφής της G , το ύψος k του δέντρου πρέπει να είναι $k > \#N$ (διότι ο πατέρας κάθε φύλλου είναι ένα μη τερματικό σύμβολο, το οποίο αναγκαστικά έχει ένα μόνο παιδί: $A ::= a$).

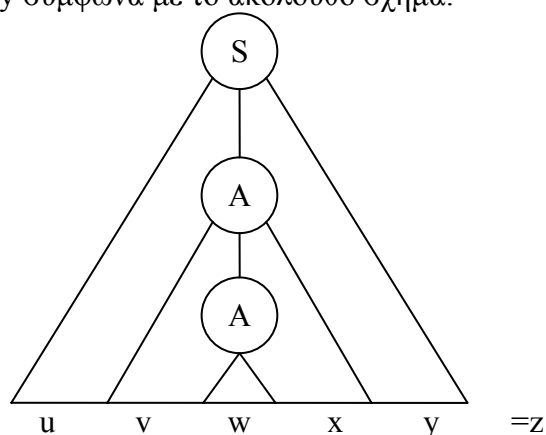
Εξετάζουμε ένα δρόμο από την ρίζα μήκους k :



$k > \#N$, οπότε από την αρχή του Περιστερώνα, υπάρχουν A_i, A_j ώστε: $A_i = A_j$, $i < j$. Διαλέγουμε το τελευταίο τέτοιο ζευγάρι. Άρα:

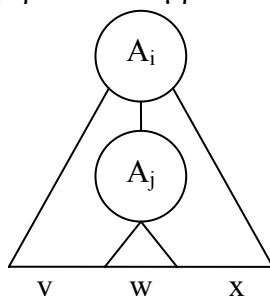
- $A_i = A_j = A$
- $i < j$
- A_{i+1}, \dots, A_k είναι διαφορετικά μη τερματικά

Ορίζουμε u, v, w, x, y σύμφωνα με το ακόλουθο σχήμα:



Τώρα έχουμε:

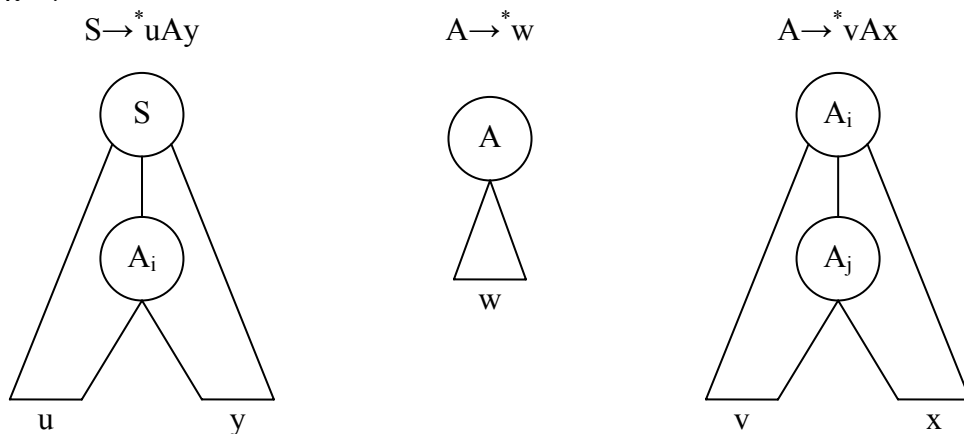
1. $|vx| \geq 1$, λόγω της μορφής της G :
 - ✓ Είτε $A \rightarrow^1 BC$ με $B \neq A$, $C \neq A$, οπότε $v \neq \epsilon$, $x \neq \epsilon$, αφού δεν υπάρχει κανόνας της μορφής $A ::= \epsilon$ στην G λόγω της κανονικής μορφής Chomsky.
 - ✓ Είτε $A \rightarrow^1 BA$, οπότε $v \neq \epsilon$, αφού δεν υπάρχει κανόνας της μορφής $A ::= \epsilon$ στην G λόγω της κανονικής μορφής Chomsky.
 - ✓ Είτε $A \rightarrow^1 AC$, οπότε $x \neq \epsilon$, αφού δεν υπάρχει κανόνας της μορφής $A ::= \epsilon$ στην G λόγω της κανονικής μορφής Chomsky.
2. Διαλέξαμε το τελευταίο ζευγάρι ίδιων συμβόλων $A_i = A_j$. Άρα στο δέντρο:



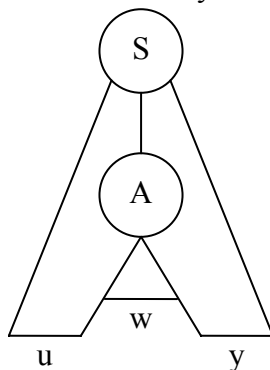
το μήκος του δρόμου από το A_i στο A_j είναι το πολύ $\#N$.

Άρα $|vwx| \leq 2^{\#N} = n$

3. Έχουμε:

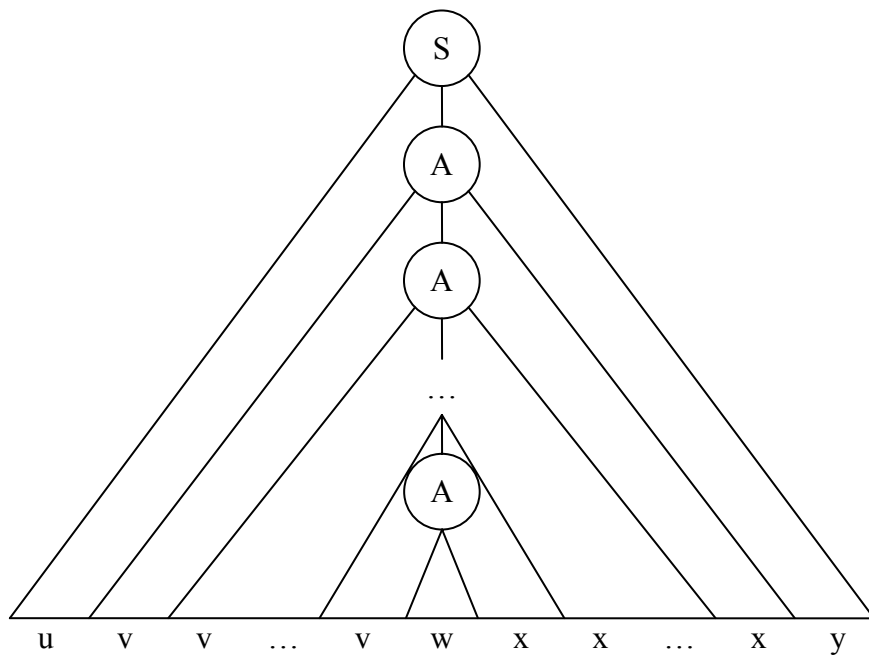


Τότε: $S \rightarrow^* uA_iy \rightarrow^* uwy$



Και επιπλέον:

$S \rightarrow^* uA_iy \rightarrow^* uvAxy \rightarrow^* uv^2Ax^2y \rightarrow^* \dots \rightarrow^* uv^iAx^iy \rightarrow^* uv^iwx^iy$.



Παράδειγμα

$L = \{a^k b^k c^k \mid k \in \mathbb{N}\}$ δεν είναι γλώσσα τύπου 2.

Έστω ότι L είναι τύπου 2.

Έστω $n \in \mathbb{N}$ ο φυσικός αριθμός από το θεώρημα.

$z := a^n b^n c^n$. Προφανώς $|z| \geq n$.

Έστω $z = uvwxy$, με τις ιδιότητες 1-3.

Τι δυνατότητες υπάρχουν;

a^n	b^n	c^n
-------	-------	-------

$|vwx| \leq n \Rightarrow vwx$ μέρος του $a^n b^n$ ή του $b^n c^n$.

Περίπτωση 1: vwx περιέχει μόνο a και/ή b .

Ας δούμε το $uwy \in L$ (3)

Έχουμε $|vx| \geq 1$ (2), άρα στο uwy έχουμε a και/ή b λιγότερα του n . Αλλά ο αριθμός των c είναι n , άρα ο αριθμός των a, b, c δεν είναι ο ίδιος, άρα $uwy \notin L$, άτοπο.

Περίπτωση 2: vwx περιέχει μόνο b και/ή c .

Προκύπτει με όμοιο συλλογισμό.

3.4 Ιδιότητες Κλειστότητας

Ένωση

Έστω $L_1, L_2 \subseteq \Sigma^*$ τύπου 2.

$L_i = L(G_i), G_i = (N_i, \Sigma, P_i, S_i), i=1,2, N_1 \cap N_2 = \emptyset$.

Ορίζουμε:

$G = (N_1 \cup N_2 \cup \{S_0\}, \Sigma, P_1 \cup P_2 \cup \{S_0 ::= S_1 \mid S_2\}, S_0), S_0 \notin N_1 \cup N_2$.

Προφανώς:

1. $L(G) = L(G_1) \cup L(G_2) = L_1 \cup L_2$

2. G : γραμματική τύπου 2

Άρα $L_1 \cup L_2$: γλώσσα τύπου 2.

Παράθεση

Έστω $L_1, L_2 \subseteq \Sigma^*$ τύπου 2.

$L_i = L(G_i), G_i = (N_i, \Sigma, P_i, S_i), i=1,2, N_1 \cap N_2 = \emptyset$.

Ορίζουμε:

$G = (N_1 \cup N_2 \cup \{S_0\}, \Sigma, P_1 \cup P_2 \cup \{S_0 ::= S_1 S_2\}, S_0), S_0 \notin N_1 \cup N_2$.

Προφανώς:

1. $L(G) = L(G_1) L(G_2) = L_1 L_2$

2. G : γραμματική τύπου 2

Άρα $L_1 L_2$: γλώσσα τύπου 2.

Kleene Star

Έστω $L \subseteq \Sigma^*$ τύπου 2.

$L = L(G), G = (N, \Sigma, P, S)$.

Ορίζουμε:

$G^* = (N \cup \{S_0\}, \Sigma, P \cup \{S_0 ::= \varepsilon \mid SS_0\}, S_0), S_0 \notin N$.

Προφανώς:

1. $L(G^*) = L(G)^* = L^*$

2. G^* : γραμματική τύπου 2

Άρα L^* : γλώσσα τύπου 2.

Τομή

Θέτουμε:

$L_1 = \{a^n b^n c^m \mid n, m \geq 0\}$

$$L_2 = \{a^n b^m c^m \mid n, m \geq 0\}$$

L_1, L_2 είναι γλώσσες τύπου 2, για παράδειγμα για την L_1 :

$S ::= DC$

$C ::= \varepsilon \mid cC$

$D ::= aDb \mid \varepsilon$

Αλλά $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$, που δείξαμε ότι δεν είναι τύπου 2!

Συμπλήρωση

Έστω ότι η συμπλήρωση κάθε γλώσσας τύπου 2 είναι γλώσσα τύπου 2.

Έστω L_1, L_2 : γλώσσες τύπου 2.

Τότε L_1^-, L_2^- : γλώσσες τύπου 2.

Οπότε και $L_1^- \cup L_2^-$: γλώσσα τύπου 2.

Άρα $(L_1^- \cup L_2^-)^- = L_1 \cap L_2$: γλώσσα τύπου 2, που είναι άτοπο γιατί είδαμε ότι η τομή δύο γλωσσών τύπου 2 δεν είναι πάντα γλώσσα τύπου 2.

Θεώρημα 3.5 Οι γλώσσες χωρίς συμφραζόμενα είναι κλειστές ως προς την ένωση, παράθεση και Kleene Star, αλλά όχι ως προς την τομή και την συμπλήρωση.

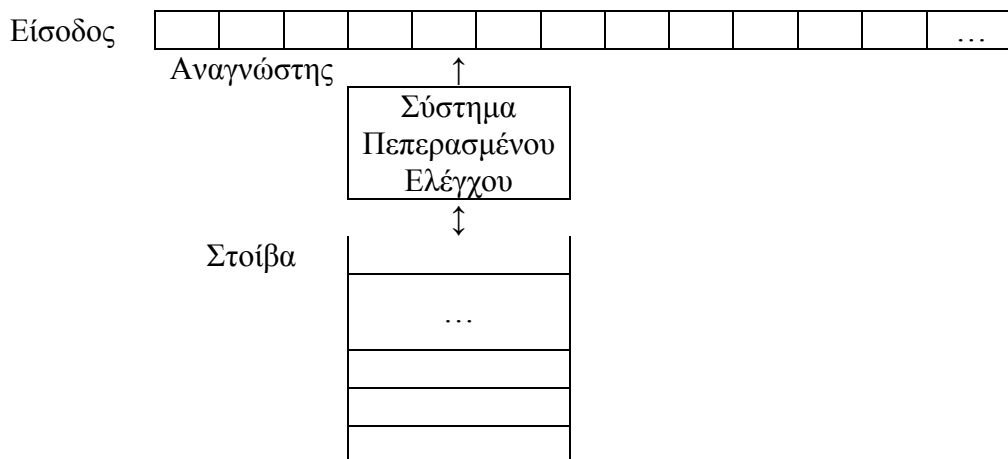
3.5 Αυτόματα με Στοιίβα

Ας εξετάσουμε μία γλώσσα που είναι τύπου 2 αλλά όχι τύπου 3:

$$L = \{a^n b^n \mid n \geq 0\}.$$

Ο λόγος που δεν μπορεί να αναγνωριστεί από ένα πεπερασμένο αυτόματο είναι ότι το αυτόματο δεν μπορεί να αποθηκεύσει την πληροφορία του πόσα a διαβάστηκαν ήδη (αν το αυτόματο A έχει k καταστάσεις δεν μπορεί να διακρίνει το $a^k b^k$ από το $a^m b^k$ αν $m < k$).

Η κεντρική ιδέα για έναν τύπο αυτομάτων που μπορεί να αναγνωρίσει γλώσσες σαν την $\{a^n b^n\}$ είναι να προσθέσουμε μία *στοίβα* (stack), στην οποία μπορεί να αποθηκευτεί απεριόριστος αριθμός πληροφοριών (πχ a^n για οποιοδήποτε n).



➤ Τι είναι στοίβα;

- ✓ push
- ✓ pop
- ✓ top

- ✓ $\text{top}(\emptyset) = \perp$
- ✓ $\text{top}(\text{push}(X, S)) = X$

- ✓ $\text{pop}(\emptyset) = \perp$
- ✓ $\text{pop}(\text{push}(X, S)) = S$
- Abstract Data Type

Ένα αυτόματο με στοίβα είναι ένα σύστημα $K = (Q, \Sigma, \Gamma, \delta, q_0, F, \gamma_0)$ όπου:

- Q : πεπερασμένο σύνολο καταστάσεων
- Σ : πεπερασμένο αλφάβητο εισόδου
- Γ : πεπερασμένο αλφάβητο στοίβας
- $q_0 \in Q$: αρχική κατάσταση
- $\gamma_0 \in \Gamma$: αρχικό σύμβολο της στοίβας
- $F \subseteq Q$: σύνολο των τελικών καταστάσεων
- δ : σχέση μετάβασης:
 - ✓ Έστω $q \in Q, a \in \Sigma \cup \{\epsilon\}, c \in \Gamma$, τότε $\delta(q, a, c) \subseteq Q \times \Gamma^*$

Γραφική Αναπαράσταση:



Εάν το δ είναι συνάρτηση, τότε το K είναι ντετερμινιστικό (ή αλλιώς αιτιοκρατικό).

- Συνολική Κατάσταση:
 - ✓ Κατάσταση $q \in Q$
 - ✓ $w \in \Sigma^*$: το μέρος της εισαγωγής που θα διαβαστεί
 - ✓ $\gamma \in \Gamma^*$: το περιεχόμενο της στοίβας
- Αρχική Συνολική Κατάσταση για $w \in \Sigma^*$: (q_0, w, c_0)
- Σχέση μετάβασης \vdash
 Έστω $(q', a) \in \delta(q, x, c)$.
 Τότε: $(q, xv, c\gamma) \vdash (q', v, a\gamma)$.
 Δηλαδή:
 - ✓ Διαβάζουμε το x (αν $x \neq \epsilon$)
 - ✓ Διαγράφουμε το κορυφαίο στοιχείο της στοίβας (pop)
 - ✓ Τοποθετούμε το a στην κορυφή της στοίβας (push)

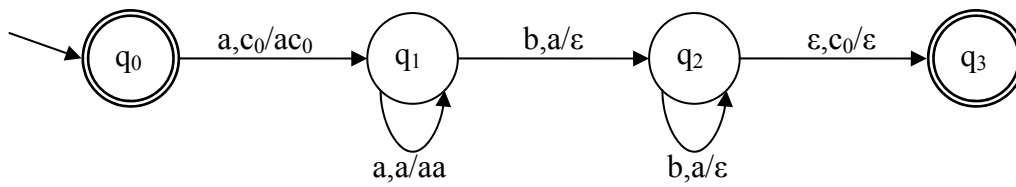
Απόδειξη

$\Sigma K_0 \vdash \Sigma K_1 \vdash \Sigma K_2 \vdash \dots \vdash \Sigma K_n$, άρα $\Sigma K_0 \vdash^* \Sigma K_n$.

- Μία λέξη $w \in \Sigma^*$ γίνεται δεκτή αν και μόνο αν: $(q_0, w, c_0) \vdash^* (q_n, \epsilon, \gamma)$, για ένα $q_n \in F$ και ένα τυχαίο $\gamma \in \Gamma^*$.
- Η γλώσσα που αναγνωρίζεται από το K :
 $L(K) = \{w \in \Sigma^* \mid w \text{ γίνεται δεκτό από το } K\}$

Παράδειγμα 1

$L = \{a^n b^n \mid n \geq 0\}$



Παράδειγμα 2

$L = \{ww^R \mid w \in \{a,b\}^*\}$

