Πεπερασμένα Αυτόματα

Myhill-Nerode θεώρημα · Ελαχιστοποίηση DFA

Χαράλαμπος Ε. Τσουρακάκης

Διαλέξεις 12 και 13: ΗΥ-280 · Θεωρία Υπολογισμού

Ανακοίνωση: Το Project του ΗΥ-280 είναι διαθέσιμο!

Συμπίεση DFA και Τυπική Επαλήθευση με Lean 4

Τι θα υλοποιήσετε;

1. **Python:** Κατασκευή DFA και συμπίεση με τον αλγόριθμο Kumar et al.
2. **Lean 4:** Τυπική πιστοποίηση ότι το αρχικό και το συμπιεσμένο αυτόματο αναγνωρίζουν την ίδια γλώσσα.
3. **Πειράματα:** Έλεγχοι ορθότητας, μετρήσεις απόδοσης και τεχνική αναφορά.

Δείτε την εκφώνηση και τα παραδοτέα στο GitHub

Υποχρεωτική προφορική εξέταση

Η παρουσία στην προφορική εξέταση και η **ουσιαστική κατανόηση της υλοποίησης και των αποδείξεων** είναι απαραίτητες.
Χωρίς προφορική εξέταση ή επαρκή κατανόηση, ο βαθμός της εργασίας είναι 0.

4. Θεώρημα Myhill-Nerode

Η ισοδυναμία Myhill-Nerode

Για $L \subseteq \Sigma^*$ και $x, y \in \Sigma^*$ ορίζουμε

$$x \equiv_L y \iff \forall z \in \Sigma^* : \; xz \in L \iff yz \in L.$$

Ισοδύναμα προθέματα

Κανένα κοινό επίθημα δεν μπορεί να τα ξεχωρίσει.

Διακριτικό επίθημα

Αν υπάρχει z ώστε ακριβώς μία από xz, yz ανήκει στην L, τότε $x \not\equiv_L y$.

Συγκρατούμε τα εξής:

Δεν συγκρίνουμε απλώς αν $x, y \in L$. Συγκρίνουμε **όλη τη μελλοντική συμπεριφορά** τους.

- Η κλάση $[x] = \{y \in \Sigma^* : y \sim_L x\}$ περιέχει όλες τις συμβολοσειρές με την ίδια μελλοντική συμπεριφορά.

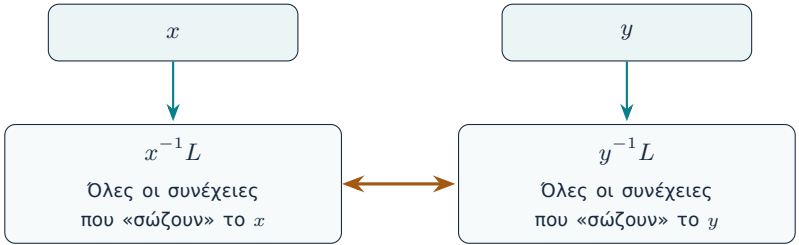
Η πιο κομψή οπτική: residual languages

Για κάθε πρόθεμα x ορίζουμε το **υπόλοιπο** (left quotient)

$$x^{-1}L = \{z \in \Sigma^* : xz \in L\}.$$

Τότε

$$\boxed{x \equiv_L y \iff x^{-1}L = y^{-1}L.}$$



Ίσα υπόλοιπα $\iff$ ίδια κατάσταση στο ελάχιστο DFA.

Το πρόβλημα

Γλώσσα

$$\Sigma = \{a, b\}, \qquad L = (ab \cup ba)^*.$$

- Οι λέξεις της L αποτελούνται από διαδοχικά ζεύγη ab ή ba.
- Παραδείγματα που ανήκουν: $\varepsilon,\ ab,\ ba,\ abba,\ baab,\ abab$.
- Παραδείγματα που δεν ανήκουν: $a,\ b,\ aa,\ bb,\ aabb$.

Πόσες διαφορετικές «καταστάσεις» μπορεί να έχει ένα πρόθεμα;

Η διαίσθηση: τι περιμένω να διαβάσω;

Για τη γλώσσα

$$L = (ab \cup ba)^*$$

φανταζόμαστε ότι διαβάζουμε τη λέξη **από αριστερά προς τα δεξιά**.

Κλάση	**Τι έχει συμβεί;**	**Τι περιμένω;**
$[\varepsilon]$	Ολοκλήρωσα σωστό ζεύγος	Τίποτα! Μπορώ να σταματήσω.
$[a]$	Διάβασα το πρώτο a	**Περιμένω** b
$[b]$	Διάβασα το πρώτο b	**Περιμένω** a
$[aa]$	Διάβασα λανθασμένο ζεύγος	**Είναι πλέον αργά!**

Η διαίσθηση: τι περιμένω να διαβάσω;

Παραδείγματα

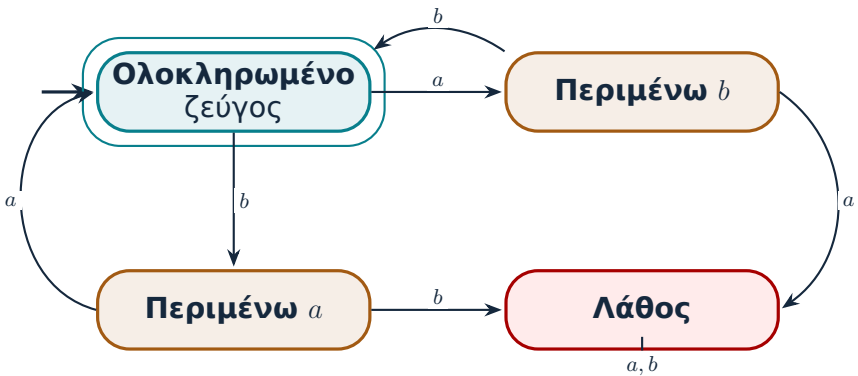
$$abba \longrightarrow [\varepsilon] \quad \text{(ολοκληρωμένα ζεύγη)}$$
$$aba \longrightarrow [a] \quad \text{(περιμένω } b\text{)}$$
$$abb \longrightarrow [b] \quad \text{(περιμένω } a\text{)}$$
$$abaa \longrightarrow [aa] \quad \text{(καμία συνέχεια δεν βοηθά)}$$

Συγκρατούμε τα εξής:

Κάθε κλάση περιγράφει **τι πρέπει να θυμόμαστε για το παρελθόν**, ώστε να αποφασίσουμε σωστά για οποιαδήποτε συνέχεια.

Η διαίσθηση: τι περιμένει το αυτόματο;

$$L = (ab \cup ba)^*$$



Ανακαλύπτουμε τις τέσσερις κλάσεις

$$[\varepsilon] = L,$$
$$[a] = La = \{wa : w \in L\},$$
$$[b] = Lb = \{wb : w \in L\},$$
$$[aa] = L(aa \cup bb)\Sigma^*.$$

Ερμηνεία

- $[\varepsilon]$: μόλις ολοκληρώσαμε σωστά ένα ζεύγος (ή δεν διαβάσαμε τίποτα).
- $[a]$, $[b]$: περιμένουμε αντίστοιχα b ή a για να κλείσει το ζεύγος.
- $[aa]$: εμφανίστηκε λανθασμένο ζεύγος· καμία συνέχεια δεν διορθώνει τη λέξη.

Γιατί οι λέξεις κάθε κλάσης είναι ισοδύναμες;

Ορίζουμε το σύνολο επιθεμάτων που οδηγούν σε αποδοχή:

$$R(x) = \{z \in \Sigma^* : xz \in L\}.$$

Πρόθεμα x	$R(x)$	**Τι απαιτείται;**
$x \in L$	L	Νέα ολοκληρωμένα ζεύγη
$x \in La$	bL	Πρώτα b, μετά ζεύγη
$x \in Lb$	aL	Πρώτα a, μετά ζεύγη
$x \in L(aa \cup bb)\Sigma^*$	$\varnothing$	Αδύνατη η αποδοχή

Κεντρική παρατήρηση

Αν δύο προθέματα ανήκουν στην ίδια γραμμή, έχουν ακριβώς τα ίδια αποδεκτά επιθέματα. Άρα είναι ισοδύναμα.

Γιατί οι τέσσερις κλάσεις είναι διαφορετικές;

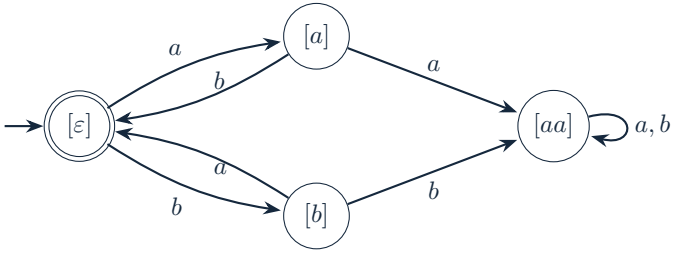
Για κάθε ζεύγος εκπροσώπων βρίσκουμε ένα **διακριτικό επίθεμα** z.

Εκπρόσωποι	z	Πρώτη συνέχεια	Δεύτερη συνέχεια
$\varepsilon,\ a$	ε	$\varepsilon \in L$	$a \notin L$
$\varepsilon,\ b$	ε	$\varepsilon \in L$	$b \notin L$
$\varepsilon,\ aa$	ε	$\varepsilon \in L$	$aa \notin L$
$a,\ b$	b	$ab \in L$	$bb \notin L$
$a,\ aa$	b	$ab \in L$	$aab \notin L$
$b,\ aa$	a	$ba \in L$	$aaa \notin L$

Συμπέρασμα

Οι τέσσερις εκπρόσωποι ε, a, b, aa είναι ανά δύο μη ισοδύναμοι.

Το ελάχιστο DFA από τις κλάσεις



- Οι καταστάσεις είναι οι κλάσεις $[\varepsilon], [a], [b], [aa]$.
- Αρχική και μοναδική τελική κατάσταση είναι η $[\varepsilon]$.
- Η μετάβαση ορίζεται ως $\delta([x], c) = [xc]$.

Τελικό συμπέρασμα: γιατί είναι ελάχιστο;

1. Οι τέσσερις κλάσεις **καλύπτουν** το Σ^* και είναι ξένες ανά δύο.
2. Οι κλάσεις είναι πράγματι κλάσεις ισοδυναμίας, αφού σε καθεμία τα αποδεκτά επιθέματα είναι ίδια.
3. Υπάρχουν **τέσσερις διαφορετικές** κλάσεις, επειδή βρήκαμε διακριτικά επιθέματα για κάθε ζεύγος εκπροσώπων.

Θεώρημα Myhill--Nerode

Ο αριθμός καταστάσεων του ελάχιστου DFA ισούται με τον αριθμό κλάσεων της $\sim_L$.

$$\boxed{\text{Ελάχιστος αριθμός καταστάσεων} = 4.}$$

Το Θεώρημα Myhill-Nerode μας λέει κάτι ακόμα ισχυρότερο

Θεώρημα

Η L είναι κανονική αν και μόνο αν η $\equiv_L$ έχει **πεπερασμένο αριθμό κλάσεων ισοδυναμίας**.

$$\boxed{L \text{ regular} \iff |\Sigma^*/\equiv_L| < \infty.}$$

Και ακόμη ισχυρότερα:

$$\boxed{|\Sigma^*/\equiv_L| = \text{αριθμός καταστάσεων του ελάχιστου DFA.}}$$

Η ιδέα

Το θεώρημα είναι ταυτόχρονα εργαλείο **μη κανονικότητας**, εργαλείο **κάτω φραγμάτων** και θεωρητική βάση της **ελαχιστοποίησης**.

Το πιο απλό δυνατό παράδειγμα: $L = \{a^n : n \geq 0\}$

$$L = \{\varepsilon, a, a^2, a^3, \ldots\} = \{a\}^*.$$

Πάρτε **οποιεσδήποτε** δύο λέξεις

$$x = a^i, \qquad y = a^j.$$

Κάθε πιθανή συνέχεια έχει τη μορφή $z = a^k$. Τότε

$$xz = a^{i+k} \in L, \qquad yz = a^{j+k} \in L.$$

Άρα

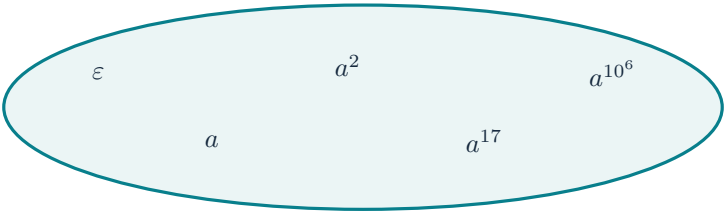
$$\boxed{a^i \equiv_L a^j \quad \text{για όλα } i, j \geq 0.}$$

Ερώτημα

Πόσες διαφορετικές πληροφορίες για το παρελθόν πρέπει να θυμόμαστε;

Μία κλάση — κυριολεκτικά καμία πληροφορία

$$[\varepsilon]_L = \{a\}^* = \Sigma^*.$$

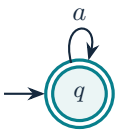


Όλα έχουν ακριβώς το ίδιο μέλλον

Συγκρατούμε τα εξής:

Ο δείκτης είναι 1. Το ελάχιστο DFA πρέπει να έχει ακριβώς **μία κατάσταση**.

Το ελάχιστο DFA για a^*



Ερμηνεία

Αφού κάθε πρόθεμα έχει το ίδιο μέλλον, το αυτόματο δεν έχει λόγο να αποθηκεύσει *οτιδήποτε* για το μήκος που έχει ήδη διαβάσει.

1 κλάση Nerode $\iff$ 1 κατάσταση $\iff$ 0 bits χρήσιμης ιστορίας.

Ένα βήμα παραπέρα: $L = \{a^{2k} : k \geq 0\}$

$$L = \{\varepsilon, a^2, a^4, a^6, \ldots\}.$$

Πάρτε δύο λέξεις

$$x = a^i, \qquad y = a^j.$$

Κάθε πιθανή συνέχεια έχει τη μορφή $z = a^k$.

Περίπτωση 1: Τα i, j έχουν την ίδια ισοτιμία.

$$i + k \equiv j + k \pmod 2.$$

Επομένως, για κάθε $k \geq 0$,

$$a^{i+k} \in L \iff a^{j+k} \in L.$$

Άρα

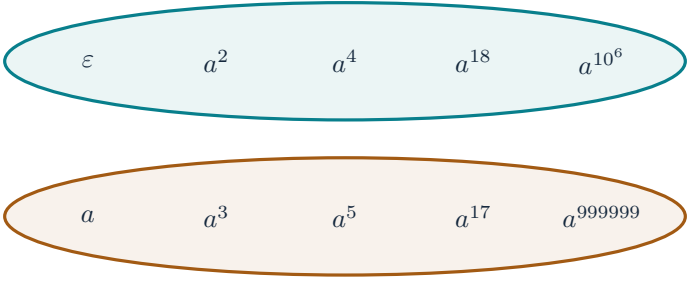
$$\boxed{a^i \equiv_L a^j \iff i \equiv j \pmod 2.}$$

Ερώτημα

Τι συμβαίνει αν τα i, j έχουν διαφορετική ισοτιμία;

Δύο κλάσεις — θυμόμαστε μόνο την ισοτιμία

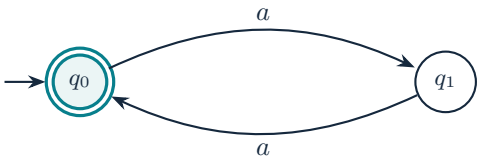
$$[\varepsilon]_L = \{\varepsilon, a^2, a^4, a^6, \ldots\}$$
$$[a]_L = \{a, a^3, a^5, a^7, \ldots\}.$$



Συγκρατούμε τα εξής:

Ο δείκτης είναι 2. Το ελάχιστο DFA έχει ακριβώς **δύο καταστάσεις**.

Το ελάχιστο DFA για $L = \{a^{2k} : k \geq 0\}$



Άρτιο πλήθος a Περιττό πλήθος a

Ερμηνεία

Το αυτόματο δεν χρειάζεται να θυμάται *πόσα* a έχει διαβάσει.
Χρειάζεται να θυμάται μόνο αν το πλήθος είναι **άρτιο ή περιττό**.

$$\underbrace{2 \text{ κλάσεις Nerode}}_{\text{διαφορετικά μέλλοντα}} \iff \underbrace{2 \text{ καταστάσεις}}_{\text{ελάχιστο DFA}} \iff \underbrace{1 \text{ bit}}_{\text{χρήσιμη μνήμη}} .$$

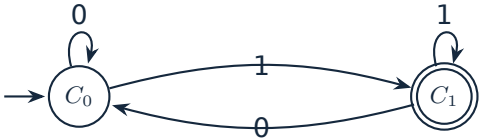
Εναλλακτικός τρόπος του ίδιου ερωτήματος

$$L = \{w \in \{0,1\}^* : w \text{ τελειώνει σε } 1\}.$$

Υπάρχουν δύο είδη μέλλοντος:

$$C_0 = \{\varepsilon\} \cup \{w : w \text{ τελειώνει σε } 0\}, \qquad C_1 = \{w : w \text{ τελειώνει σε } 1\}.$$

Διακρίνονται με $z = \varepsilon$.



Τρεις κλάσεις: μήκος modulo 3

Θεωρούμε τη γλώσσα

$$L = \{w \in \{0,1\}^* : |w| \equiv 0 \pmod 3\}.$$

Δηλαδή, δεχόμαστε τις λέξεις των οποίων το μήκος είναι πολλαπλάσιο του 3.

Πάρτε δύο οποιεσδήποτε λέξεις x, y και μια πιθανή συνέχεια z. Παρατηρούμε οτι $xz \in L \iff |x| + |z| \equiv 0 \pmod 3$. Αντίστοιχα, $yz \in L \iff |y| + |z| \equiv 0 \pmod 3$. Επομένως, αν $|x| \equiv |y| \pmod 3$, κάθε συνέχεια γίνεται αποδεκτή είτε και στις δύο περιπτώσεις είτε σε καμία.

$$\boxed{|x| \equiv |y| \pmod 3 \Longrightarrow x \equiv_L y.}$$

Ερώτημα

Αρκεί αυτό για να αποδείξουμε ότι υπάρχουν ακριβώς τρεις κλάσεις;

Γιατί οι τρεις κλάσεις είναι διαφορετικές;

Αρκεί να εξετάσουμε τις αντιπροσωπευτικές λέξεις

$$\varepsilon, \qquad 0, \qquad 00.$$

Για να αποδείξουμε ότι δύο λέξεις δεν είναι ισοδύναμες, βρίσκουμε μια **διαχωριστική συνέχεια** z.

Λέξεις	**Συνέχεια** z	**Πρώτη λέξη**	**Δεύτερη λέξη**
$\varepsilon,\ 0$	ε	$\varepsilon \in L$	$0 \notin L$
$\varepsilon,\ 00$	ε	$\varepsilon \in L$	$00 \notin L$
$0,\ 00$	0	$00 \notin L$	$000 \in L$

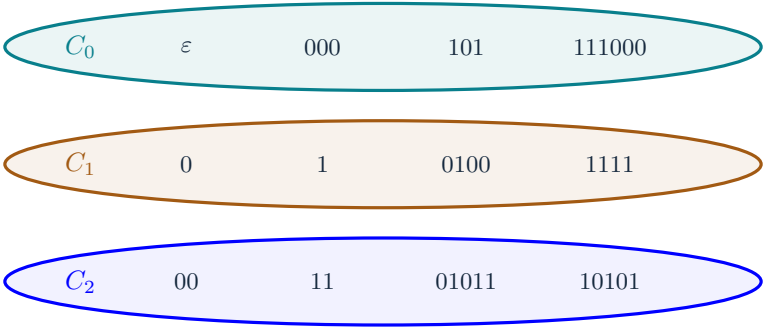
Άρα $\varepsilon \not\equiv_L 0, \qquad \varepsilon \not\equiv_L 00, \qquad 0 \not\equiv_L 00.$

Συγκρατούμε τα εξής:

Υπάρχουν τουλάχιστον **τρεις** διαφορετικές κλάσεις. Επειδή το υπόλοιπο modulo 3 καθορίζει το μέλλον, υπάρχουν και το πολύ τρεις. Άρα ακριβώς τρεις.

Τρεις κλάσεις — τρία διαφορετικά μέλλοντα

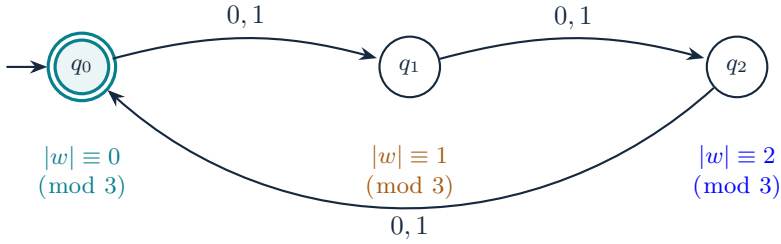
$$C_r = \{w \in \{0,1\}^* : |w| \equiv r \pmod 3\}, \qquad r = 0, 1, 2.$$



Συγκρατούμε τα εξής:

Κάθε κλάση περιέχει άπειρες λέξεις, αλλά όλες απαιτούν την ίδια πληροφορία για να αποφασίσουμε τη μελλοντική αποδοχή.

Το ελάχιστο DFA: θυμόμαστε το μήκος modulo 3



Ερμηνεία

Κάθε σύμβολο, είτε 0 είτε 1, αυξάνει το μήκος κατά ένα. Επομένως, μετακινούμαστε κυκλικά από το υπόλοιπο 0 στο 1, στο 2 και πάλι στο 0.

$|\Sigma^*/\equiv_L| = 3 \iff$ 3 καταστάσεις στο ελάχιστο DFA.

Proving Lower bounds: το n-οστό bit από το τέλος

Ορίζουμε

$$L_n = \{w \in \{0,1\}^* : \text{το } n\text{-οστό σύμβολο από το τέλος είναι } 1\}.$$

Για να αποφασίσουμε στο τέλος, πρέπει να θυμόμαστε τα τελευταία n bits.



Ποιο bit θα γίνει «4ο από το τέλος» εξαρτάται από το πόσα ακόμη bits θα διαβάσουμε.

Ερώτημα

Μπορούμε να αποδείξουμε ότι χρειάζονται πραγματικά 2^n καταστάσεις;

Το 3ο bit από το τέλος: ας δούμε παραδείγματα

Θεωρούμε αρχικά την ειδική περίπτωση $n = 3$:

$$L_3 = \{w \in \{0,1\}^* : \text{το 3ο bit από το τέλος είναι } 1\}.$$

1 0 **1** 1 0	**Αποδοχή**
1 0 **0** 1 0	**Απόρριψη**
0 1 1	**Απόρριψη**
1 1 0	**Αποδοχή**

Συγκρατούμε τα εξής:

Δεν μας ενδιαφέρει πόσα 1 έχει η λέξη. Μας ενδιαφέρει **ένα συγκεκριμένο bit**, μετρώντας από το τέλος.

Γιατί πρέπει να θυμόμαστε τα τελευταία bits;

Έστω ότι μέχρι τώρα έχουμε διαβάσει

$$x = 101.$$

Ας δούμε τι συμβαίνει όταν προσθέτουμε διαφορετικές συνέχειες.

Συνέχεια z	xz	3ο bit από τέλος	Αποδοχή
ε	$\mathbf{1}01$	1	ΝΑΙ
0	$1\mathbf{0}10$	0	ΟΧΙ
00	$10\mathbf{1}00$	1	ΝΑΙ
000	$101\mathbf{0}00$	0	ΟΧΙ

Προσέξτε τι συμβαίνει:

- Χωρίς συνέχεια, εξετάζουμε το πρώτο bit του x.
- Με ένα επιπλέον bit, εξετάζουμε το δεύτερο.
- Με δύο επιπλέον bits, εξετάζουμε το τρίτο.

Δύο λέξεις που μοιάζουν: 101 και 111

Θεωρούμε τα δύο προθέματα

$$x = 101, \qquad y = 111.$$

Χωρίς συνέχεια:

$$101 \in L_3, \qquad 111 \in L_3.$$

Και οι δύο λέξεις γίνονται αποδεκτές!

Αλλά ας προσθέσουμε το ίδιο bit $z = 0$.

1	0	1	0	$1010 \notin L_3$

1	1	1	0	$1110 \in L_3$

$$1010 \notin L_3, \qquad 1110 \in L_3.$$

Οκτώ διαφορετικά προθέματα — οκτώ μέλλοντα

Υπάρχουν $2^3 = 8$ δυαδικές λέξεις μήκους 3.
Για κάθε λέξη $x = x_1x_2x_3$, εξετάζουμε τρεις πιθανές συνέχειες:

$$z \in \{\varepsilon, 0, 00\}.$$

x	$x \in L_3$	$x0 \in L_3$	$x00 \in L_3$	Προφίλ
000	O	O	O	000
001	O	O	N	001
010	O	N	O	010
011	O	N	N	011
100	N	O	O	100
101	N	O	N	101
110	N	N	O	110
111	N	N	N	111

N = αποδοχή, O = απόρριψη. Το προφίλ κωδικοποιεί τις τρεις απαντήσεις.

Η γενική απόδειξη: τουλάχιστον 2^n καταστάσεις

$$L_n = \{w \in \{0,1\}^* : \text{το } n\text{-οστό bit από το τέλος είναι } 1\}.$$

Πάρτε δύο διαφορετικές λέξεις μήκους n:

$$x = x_1 x_2 \cdots x_n, \qquad y = y_1 y_2 \cdots y_n.$$

Επειδή $x \neq y$, υπάρχει θέση $i \in \{1, \ldots, n\}$ με $x_i \neq y_i$.

Επιλέγουμε τη συνέχεια

$$\boxed{z = 0^{i-1}.}$$

Οι λέξεις xz και yz έχουν μήκος $n + i - 1$. Άρα το n-οστό bit από το τέλος βρίσκεται ακριβώς στη θέση i.

Επομένως,

$$xz \in L_n \iff x_i = 1, \qquad yz \in L_n \iff y_i = 1.$$

Άρα (**γιατί;**) $\boxed{x \not\equiv_{L_n} y \implies \operatorname{ind}(L_n) \geq 2^n.}$

Ας δούμε το γιατί σε λεπτομέρεια

Έχουμε δείξει ότι για κάθε δύο διαφορετικές λέξεις $x, y \in \{0,1\}^n$ υπάρχει συνέχεια z τέτοια ώστε

$$xz \in L_n \iff yz \notin L_n.$$

Τι σημαίνει αυτό για ένα DFA $M = (Q, \Sigma, \delta, q_0, F)$;

- Υποθέτουμε ότι το M φτάνει στην **ίδια κατάσταση** μετά την ανάγνωση των x και y:
$$\delta^*(q_0, x) = \delta^*(q_0, y) = q.$$
- Αν διαβάσουμε στη συνέχεια την ίδια λέξη z, το αυτόματο θα καταλήξει πάλι στην ίδια κατάσταση:
$$\delta^*(q_0, xz) = \delta^*(q, z) = \delta^*(q_0, yz).$$
- Άρα θα πρέπει είτε να αποδέχεται και τις δύο λέξεις xz, yz, είτε να απορρίπτει και τις δύο. **Άτοπο!**

Επομένως, οι x, y οδηγούν σε διαφορετικές καταστάσεις, δηλαδή $x \not\equiv_{L_n} y$ και άρα κάθε DFA για τη L_n χρειάζεται τουλάχιστον 2^n καταστάσεις.

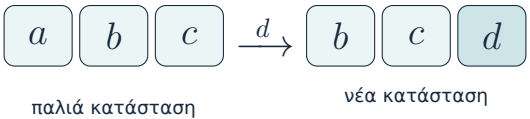
Γιατί αρκούν 2^n καταστάσεις;

Μπορούμε να κατασκευάσουμε DFA που θυμάται ακριβώς τα τελευταία n bits. Για παράδειγμα, όταν $n = 3$, η κατάσταση είναι μια τριάδα

$$q_{abc}, \qquad a, b, c \in \{0, 1\}.$$

Κάθε νέο bit προκαλεί ολίσθηση:

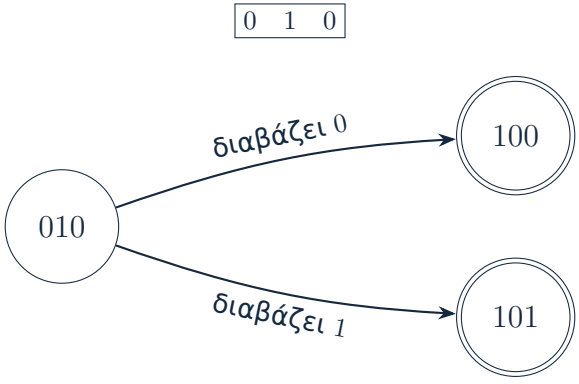
$$\boxed{\delta(q_{abc}, d) = q_{bcd}.}$$



Πώς θυμάται το DFA τα τελευταία 3 bits;

Παράδειγμα: Βρισκόμαστε στην κατάσταση 010.

Το αυτόματο θυμάται ότι τα τελευταία τρία bits που διάβασε είναι:



- Σχεδιάστε όλο το αυτόματο για εξάσκηση.

Συνταγή για απόδειξη μη κανονικότητας

Για να δείξουμε ότι L δεν είναι κανονική:

1. Επιλέγουμε άπειρη οικογένεια προθεμάτων $x_0, x_1, x_2, \ldots$.
2. Για κάθε $i \neq j$, βρίσκουμε **ένα** επίθημα z που διακρίνει τα x_i, x_j.
3. Συμπεραίνουμε ότι υπάρχουν άπειρες κλάσεις Nerode.

άπειρα pairwise distinguishable prefixes $\Rightarrow$ μη κανονική γλώσσα.

Η τέχνη

Δεν ψάχνουμε ένα δύσκολο DFA. Ψάχνουμε **τι πληροφορία θα έπρεπε να θυμάται για πάντα**.

Μη κανονική: $\{0^n 1^n : n \geq 0\}$

$$L = \{0^n 1^n : n \geq 0\}.$$

Εξετάζουμε $x_i = 0^i$ για $i = 0, 1, 2, \ldots$. Για $i \neq j$, επιλέγουμε

$$z = 1^i.$$

Τότε

$$0^i 1^i \in L, \qquad 0^j 1^i \notin L.$$

Άρα όλα τα 0^i ανήκουν σε διαφορετικές κλάσεις:

$$\boxed{\mathrm{ind}(L) = \infty.}$$

Η ιδέα

Ένα DFA θα έπρεπε να θυμάται **ακριβώς πόσα** μηδενικά είδε — απεριόριστη πληροφορία.

Μια ενδιαφέρουσα γλώσσα: οι πρώτοι αριθμοί

$$L_{\text{prime}} = \{0^p \mid p \text{ είναι πρώτος αριθμός}\}$$

Τι περιέχει αυτή η γλώσσα;

$$L_{\text{prime}} = \{00,\ 000,\ 00000,\ 0000000,\ 00000000000, \ldots\}$$



Ερώτημα

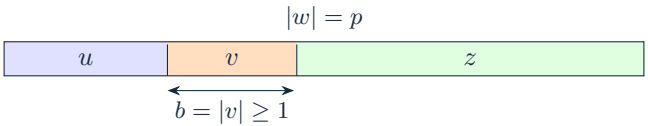
Μπορεί ένα DFA με πεπερασμένες καταστάσεις να αναγνωρίζει ακριβώς τις λέξεις των οποίων το μήκος είναι πρώτος αριθμός;

Θα αποδείξουμε ότι η γλώσσα δεν είναι κανονική με δύο τρόπους.

Απόδειξη 1: Λήμμα άντλησης

Υποθέτουμε προς άτοπο ότι η L_{prime} είναι κανονική. Έστω $s \geq 1$ το μήκος άντλησης.

- Επιλέγουμε έναν **πρώτο αριθμό** $p > s$.
- Η λέξη $w = 0^p \in L_{\text{prime}}$ έχει μήκος μεγαλύτερο από s.
- Από το λήμμα άντλησης υπάρχει διάσπαση $w = uvz$ με $|uv| \leq s, \qquad |v| \geq 1,$ τέτοια ώστε $uv^k z \in L_{\text{prime}}$ για κάθε $k \geq 0$.



Στόχος: Να επιλέξουμε k ώστε το μήκος $|uv^k z|$ να είναι **σύνθετος αριθμός**.

Απόδειξη 1: Πώς προκύπτει η αντίφαση;

Έστω $b = |v| \geq 1$.

Επειδή $|uvz| = p$, για κάθε $k \geq 0$ έχουμε $|uv^k z| = p + (k-1)b$.

Πώς επιλέγουμε το k ώστε το μήκος να μην είναι πρώτος;

Η έξυπνη επιλογή

Θέτουμε $\boxed{k = p + 1.}$ Τότε

$$|uv^{p+1}z| = p + pb = \boxed{p(1+b)}.$$

- Ο p είναι πρώτος, άρα $p \geq 2$. Επειδή $b \geq 1$, έχουμε $1 + b \geq 2$.
- Άρα ο $p(1+b)$ είναι **σύνθετος αριθμός**. ¸ **Άτοπο!** Το λήμμα απαιτεί $uv^k z \in L_{\text{prime}}$ για κάθε $k \geq 0$.
- Άρα $\boxed{L_{\text{prime}} \text{ δεν είναι κανονική.}}$

Απόδειξη 2 Myhill--Nerode: Άπειρες κλάσεις ισοδυναμίας

$$L = \{0^p : p \text{ πρώτος αριθμός}\}.$$

Ισχυρισμός

Κάθε λέξη 0^n αποτελεί μόνη της μία κλάση ισοδυναμίας:

$$\boxed{[0^n]_L = \{0^n\}, \qquad n \geq 0.}$$

Τι πρέπει να αποδείξουμε;

Ότι οποιεσδήποτε δύο διαφορετικές λέξεις 0^a και 0^b είναι διακρίσιμες.

Επιλέγουμε αυθαίρετα

$$0 \leq a < b$$

και θέτουμε

$$\boxed{d = b - a \geq 1.}$$

Υποθέτουμε, προς άτοπο, ότι $\boxed{0^a \equiv_L 0^b.}$

Απόδειξη 2 Myhill--Nerode: Τι συνεπάγεται η ισοδυναμία;

Η υπόθεσή μας είναι

$$0^a \equiv_L 0^b, \qquad b = a + d.$$

Από τον ορισμό Myhill--Nerode:
Για κάθε κοινή συνέχεια z,

$$0^a z \in L \iff 0^b z \in L.$$

Επιλέγουμε μια αυθαίρετη συνέχεια

$$\boxed{z = 0^t, \qquad t \geq 0.}$$

Τότε

$$0^a 0^t \in L \iff 0^b 0^t \in L.$$

Επομένως,

$$\boxed{a + t \text{ πρώτος} \iff a + t + d \text{ πρώτος}, \quad \forall t \geq 0.}$$

Απόδειξη 2 Myhill--Nerode: Η υποτιθέμενη περιοδικότητα

Από την προηγούμενη διαφάνεια γνωρίζουμε ότι

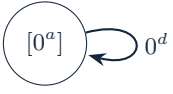
$$a + t \text{ πρώτος} \iff a + t + d \text{ πρώτος}, \qquad \forall t \geq 0.$$

Θέτοντας $n = a + t$, προκύπτει

$$\boxed{n \text{ πρώτος} \iff n + d \text{ πρώτος}, \quad \forall n \geq a.}$$

Τι σημαίνει αυτό;

Από το μήκος a και μετά, η ιδιότητα «είναι πρώτος αριθμός» θα έπρεπε να επαναλαμβάνεται με περίοδο d.



Μετά από d επιπλέον μηδενικά επιστρέφουμε στην ίδια κλάση.

Απόδειξη 2 Myhill--Nerode: Πώς εισάγουμε τον πρώτο p;

Επιλέγουμε έναν πρώτο αριθμό $p > a$.
Αυτό είναι πάντα δυνατό, επειδή υπάρχουν άπειροι πρώτοι αριθμοί.

Προσοχή

Δεν υποθέτουμε ότι

$$0^p \equiv_L 0^a.$$

Η μόνη υπόθεσή μας είναι $0^a \equiv_L 0^b$.

Θέλουμε να χρησιμοποιήσουμε την ισοδυναμία των 0^a και 0^b, ώστε η πρώτη λέξη, μετά από μια κοινή συνέχεια, να γίνει 0^p.

Επιλέγουμε λοιπόν

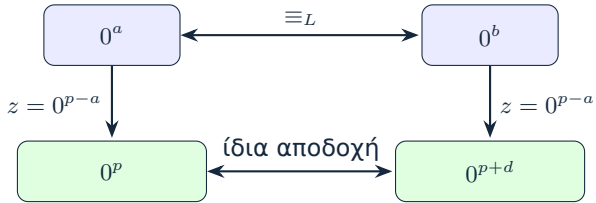
$$\boxed{z = 0^{p-a}.}$$

Επειδή $p > a$, το $p - a$ είναι θετικός ακέραιος. Άρα η συνέχεια z είναι έγκυρη.

Απόδειξη 2 Myhill--Nerode: Η κρίσιμη κοινή συνέχεια

Έχουμε $0^a \equiv_L 0^b$ και επιλέγουμε την κοινή συνέχεια

$$\boxed{z = 0^{p-a}.}$$



Πράγματι,

$$\underbrace{0^a 0^{p-a}}_{0^p} \in L \iff \underbrace{0^b 0^{p-a}}_{0^{p+d}} \in L.$$

Επειδή ο p είναι πρώτος, $0^p \in L$. Επομένως,

$$\boxed{p+d \text{ είναι επίσης πρώτος.}}$$

Απόδειξη 2 Myhill--Nerode: Η τελική αντίφαση

Η υπόθεσή μας συνεπάγεται ότι

$$n \text{ πρώτος} \iff n + d \text{ πρώτος}, \qquad \forall n \geq a.$$

Ξεκινώντας από τον πρώτο $p > a$ και εφαρμόζοντας επανειλημμένα την ισοδυναμία, όλοι οι αριθμοί

$$p,\ p + d,\ p + 2d, \ldots, p + kd, \ldots$$

θα έπρεπε να είναι πρώτοι.



Για $k = p$, όμως, $\boxed{p + pd = p(1 + d),}$ που είναι σύνθετος, αφού $p \geq 2$ και $1 + d \geq 2$.

Άτοπο!

Άρα, για κάθε $0 \leq a < b$, $\boxed{0^a \not\equiv_L 0^b.}$

Δύο αποδείξεις, μία κοινή ιδέα

1. Λήμμα άντλησης

Αν η γλώσσα ήταν κανονική, θα μπορούσαμε να επαναλαμβάνουμε ένα μη κενό τμήμα μιας αρκετά μεγάλης λέξης χωρίς να βγαίνουμε από τη γλώσσα.

Όμως, αν ξεκινήσουμε από μήκος p πρώτο και αντλήσουμε κατάλληλα, παίρνουμε μήκος

$$\boxed{p(1+b)},$$

που είναι σύνθετος αριθμός.

2. Θεώρημα Myhill-Nerode

Αν η γλώσσα ήταν κανονική, δύο διαφορετικά μήκη a και $a+d$ θα ήταν ισοδύναμα.

Αυτό θα επέβαλλε περιοδική συμπεριφορά για όλα τα αρκετά μεγάλα μήκη.

Όμως, ξεκινώντας από έναν πρώτο p, η περιοδικότητα θα ανάγκαζε και το

$$\boxed{p(1+d)}$$

Απόδειξη Θεωρήματος Myhill-Nerode

2

Θεώρημα Myhill--Nerode: η κατασκευή

Θεώρημα

Έστω $L \subseteq \Sigma^*$ κανονική γλώσσα. Τότε υπάρχει DFA που αναγνωρίζει τη L και έχει ακριβώς τόσες καταστάσεις όσες είναι οι κλάσεις ισοδυναμίας της $\sim_L$.

Θα κατασκευάσουμε το **κανονικό αυτόματο** της L:

$$M = (K, \Sigma, \delta, s, F).$$

1. Ορίζουμε καταστάσεις, αρχική κατάσταση και τελικές καταστάσεις.
2. Αποδεικνύουμε ότι η κατασκευή είναι έγκυρη.
3. Αποδεικνύουμε ότι $L(M) = L$.

Κατασκευή του κανονικού αυτομάτου

Ορίζουμε $M = (K, \Sigma, \delta, s, F)$ ως εξής:

$$K = \{[x] : x \in \Sigma^*\},$$
$$s = [\varepsilon],$$
$$F = \{[x] : x \in L\},$$
$$\delta([x], a) = [xa] \qquad (a \in \Sigma).$$

Τι πρέπει να ελέγξουμε;

Το K είναι πεπερασμένο; Η δ δεν εξαρτάται από τον αντιπρόσωπο x της κλάσης $[x]$;

Γιατί το σύνολο καταστάσεων K είναι πεπερασμένο;

Επειδή η L είναι κανονική, υπάρχει DFA

$$M' = (Q', \Sigma, \delta', q_0', F') \quad \text{με} \quad L(M') = L.$$

Ορίζουμε $x \sim_{M'} y$ αν

$$\delta'^*(q_0', x) = \delta'^*(q_0', y).$$

Αν $x \sim_{M'} y$, τότε για κάθε $z \in \Sigma^*$ οι xz, yz οδηγούν στην ίδια κατάσταση. Άρα

$$x \sim_{M'} y \implies x \sim_L y.$$

Συμπέρασμα

Η $\sim_{M'}$ είναι λεπτότερη από την $\sim_L$. Επομένως

$$\#(\Sigma^*/\sim_L) \leq \#(\Sigma^*/\sim_{M'}) \leq |Q'| < \infty.$$

Γιατί η μετάβαση είναι καλά ορισμένη;

Έχουμε ορίσει $\delta([x], a) = [xa]$. Όμως η ίδια κλάση μπορεί να έχει διαφορετικούς αντιπροσώπους. Έστω $[x] = [y]$, δηλαδή $x \sim_L y$. Για κάθε $z \in \Sigma^*$:

$$x(az) \in L \iff y(az) \in L.$$

Επειδή $az \in \Sigma^*$, από τον ορισμό της $\sim_L$ προκύπτει

$$(xa)z \in L \iff (ya)z \in L.$$

Άρα $xa \sim_L ya$, οπότε

$$\boxed{[xa] = [ya].}$$

Συμπέρασμα

Η δ εξαρτάται μόνο από την κλάση $[x]$ και όχι από τον αντιπρόσωπό της.

Βασικό λήμμα: ανάγνωση ολόκληρης λέξης

Λήμμα

Για κάθε $x, y \in \Sigma^*$ ισχύει

$$\boxed{\delta^*([x], y) = [xy].}$$

Απόδειξη με επαγωγή στο $|y|$. Βάση: $y = \varepsilon$. Από τον ορισμό της εκτεταμένης μετάβασης,

$$\delta^*([x], \varepsilon) = [x] = [x\varepsilon].$$

Επαγωγική υπόθεση: Για κάποια λέξη y, υποθέτουμε

$$\delta^*([x], y) = [xy] \quad \text{για κάθε } x \in \Sigma^*.$$

Το επαγωγικό βήμα

Έστω $a \in \Sigma$. Θέλουμε να δείξουμε ότι

$$\delta^*([x], ya) = [xya].$$

Από τον αναδρομικό ορισμό της δ^* και την επαγωγική υπόθεση:

$$\begin{aligned} \delta^*([x], ya) &= \delta\big(\delta^*([x], y), a\big) \\ &= \delta([xy], a) \\ &= [xya]. \end{aligned}$$

Άρα

Το λήμμα ισχύει για κάθε $x, y \in \Sigma^*$.

Ορθότητα: $L(M) = L$

Για κάθε $x \in \Sigma^*$:

$$
\begin{aligned}
x \in L(M) &\iff \delta^*(s, x) \in F \\
&\iff \delta^*([\varepsilon], x) \in F \\
&\iff [x] \in F && \text{(από το λήμμα)} \\
&\iff x \in L && \text{(από τον ορισμό του } F\text{)}.
\end{aligned}
$$

Συμπέρασμα

$L(M) = L$. Το κατασκευασμένο DFA έχει μία κατάσταση για κάθε κλάση της $\sim_L$, όπως απαιτεί το θεώρημα. □

Γιατί είναι το ελάχιστο DFA;

Κάθε DFA A που αναγνωρίζει τη L πρέπει να έχει τουλάχιστον τόσες καταστάσεις όσες οι κλάσεις της $\sim_L$:

$$|Q_A| \geq \#(\Sigma^*/\sim_L).$$

Το κανονικό αυτόματο M που κατασκευάσαμε έχει ακριβώς

$$|K| = \#(\Sigma^*/\sim_L).$$

Τελικό συμπέρασμα

Το M είναι ελάχιστο DFA για τη L. Κάθε ελάχιστο προσβάσιμο DFA που αναγνωρίζει τη L είναι ισόμορφο με αυτό.

Άσκηση : Από τις κλάσεις ισοδυναμίας στο ελάχιστο DFA

Παράδειγμα: η γλώσσα των λέξεων που τελειώνουν σε 01

$$L = (0+1)^*01, \qquad \Sigma = \{0,1\}.$$

Βρείτε τις κλάσεις της $\sim_L$ και κατασκευάστε το ελάχιστο DFA.

Κεντρική ιδέα: δύο λέξεις x, y είναι ισοδύναμες όταν

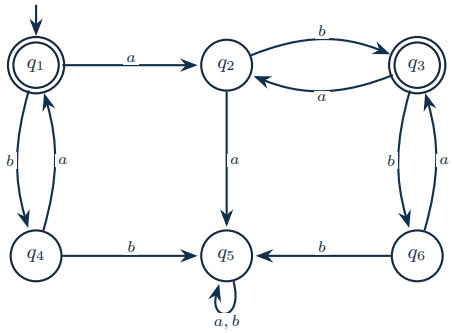
$$\forall z \in \Sigma^* : \quad xz \in L \iff yz \in L.$$

Ερώτηση στην τάξη

Ποια πληροφορία από το x πρέπει να θυμόμαστε ώστε να αποφασίσουμε αν το xz θα τελειώνει σε 01;

5. Ελαχιστοποίηση DFA

Το πρόβλημα: ένα DFA με έξι καταστάσεις



Αλφάβητο: $\Sigma = \{a, b\}$. **Αρχική:** q_1. **Τελικές:** $F = \{q_1, q_3\}$. **Ερώτηση:**

Μπορούν κάποιες καταστάσεις να συγχωνευθούν χωρίς να αλλάξει η γλώσσα;

Συγκρατούμε τα εξής:

Δεν συγχωνεύουμε καταστάσεις επειδή «μοιάζουν» στο σχήμα. Συγκρίνουμε τι αποδέχονται από εδώ και πέρα.

Πρώτα καταλαβαίνουμε τη συμπεριφορά του DFA

Διαβάζουμε από την αρχική κατάσταση q_1. Πού καταλήγουμε;

Λέξη	Διαδρομή	Τελική;	Αποτέλεσμα
ε	q_1	ναι	αποδοχή
ab	$q_1 \xrightarrow{a} q_2 \xrightarrow{b} q_3$	ναι	αποδοχή
ba	$q_1 \xrightarrow{b} q_4 \xrightarrow{a} q_1$	ναι	αποδοχή
aa	$q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_5$	όχι	απόρριψη
bb	$q_1 \xrightarrow{b} q_4 \xrightarrow{b} q_5$	όχι	απόρριψη
$abba$	$q_1 \xrightarrow{a} q_2 \xrightarrow{b} q_3 \xrightarrow{b} q_6 \xrightarrow{a} q_3$	ναι	αποδοχή

Ερώτημα

Τι κοινό έχουν οι λέξεις ab, ba, $abba$; Ποια γλώσσα υποψιαζόμαστε;

Δύο καταστάσεις μπορεί να έχουν το ίδιο μέλλον

Δοκιμάζουμε την ίδια συνέχεια z ξεκινώντας από q_1 και q_3.

Συνέχεια z	από q_1	από q_3
ε	αποδοχή	αποδοχή
a	q_2: απόρριψη	q_2: απόρριψη
b	q_4: απόρριψη	q_6: απόρριψη
ab	q_3: αποδοχή	q_3: αποδοχή
ba	q_1: αποδοχή	q_3: αποδοχή

Υπόθεση: $q_1 \sim q_3$. Όμως λίγες δοκιμές *δεν αποτελούν απόδειξη*.

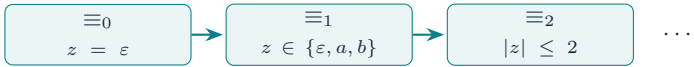
Συγκρατούμε τα εξής:

Για ισοδυναμία απαιτούμε ίδια απάντηση για **κάθε** πιθανή συνέχεια $z \in \Sigma^*$.

Η ιδέα του αλγορίθμου: αυξάνουμε το μήκος της συνέχειας

Ορίζουμε $p \equiv_k q$ όταν οι δύο καταστάσεις συμφωνούν στην αποδοχή για **όλες** τις συνέχειες μήκους το πολύ k:

$$p \equiv_k q \iff \forall z \in \Sigma^*,\ |z| \le k : \quad \big[\delta^*(p,z) \in F \iff \delta^*(q,z) \in F\big].$$



Όσο αυξάνεται το k, οι κλάσεις μπορούν να **διασπαστούν**, ποτέ όμως να ενωθούν ξανά.

Συγκρατούμε τα εξής:

Αν δύο states διακρίνονται με συνέχεια μήκους k, θα διακρίνονται και σε κάθε επόμενο γύρο.

Γύρος 0: μόνο η κενή συνέχεια ε

Με $z = \varepsilon$ βλέπουμε απλώς αν η κατάσταση είναι τελική.

$$P_0 = \Big\{ \underbrace{\{q_1, q_3\}}_{\text{τελικές}}, \quad \underbrace{\{q_2, q_4, q_5, q_6\}}_{\text{μη τελικές}} \Big\}.$$

Τελικές
$q_1 \quad q_3$

Μη τελικές
$q_2 \quad q_4 \quad q_5 \quad q_6$

Για παράδειγμα, q_1 και q_2 διακρίνονται ήδη με $z = \varepsilon$.

Ερώτημα

Μπορούμε ακόμη να ξεχωρίσουμε τα q_2 και q_4; Με ποιο σύμβολο;

Ο πίνακας μεταβάσεων: όλα τα δεδομένα σε μία θέση

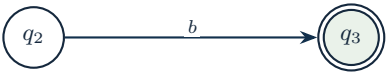
Κατάσταση	$\delta(q, a)$	$\delta(q, b)$	Τελική;
$\rightarrow *q_1$	q_2	q_4	ναι
q_2	q_5	q_3	όχι
$*q_3$	q_2	q_6	ναι
q_4	q_1	q_5	όχι
q_5	q_5	q_5	όχι
q_6	q_3	q_5	όχι

Συγκρατούμε τα εξής:

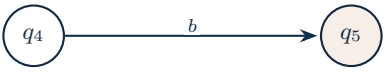
Σε κάθε γύρο δεν χρειάζεται να δοκιμάζουμε όλες τις λέξεις. Κοιτάμε πού οδηγούν τα a, b ως προς τα blocks του προηγούμενου γύρου.

Γύρος 1: τα q_2 και q_4 είναι διαφορετικά

Και τα δύο είναι μη τελικά. Άρα το ε **δεν** τα ξεχωρίζει.



αποδοχή



απόρριψη

$$\delta(q_2, b) = q_3 \in F, \qquad \delta(q_4, b) = q_5 \notin F.$$

Συγκρατούμε τα εξής:

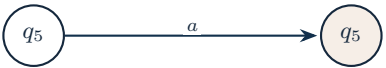
Το **διαχωριστικό επίθημα** $z = b$ αποδεικνύει ότι $q_2 \not\equiv_1 q_4$.

Γύρος 1: γιατί το q_5 πρέπει να μείνει μόνο του;

Εξετάζουμε q_4 έναντι q_5, που ήταν στο ίδιο block του P_0.



αποδοχή



απόρριψη

$$\delta(q_4, a) = q_1 \in F, \qquad \delta(q_5, a) = q_5 \notin F.$$

Ανάλογα, το a διακρίνει q_6 από q_5.

Συγκρατούμε τα εξής:

Άρα το q_5 δεν μπορεί να είναι ισοδύναμο ούτε με q_4 ούτε με q_6.

Γύρος 1: υπολογίζουμε τις υπογραφές

Ονομάζουμε $T = \{q_1, q_3\}$ (τελικές), $N = \{q_2, q_4, q_5, q_6\}$ (μη τελικές). Η **υπογραφή** γράφει το block του state και τα blocks των δύο διαδόχων.

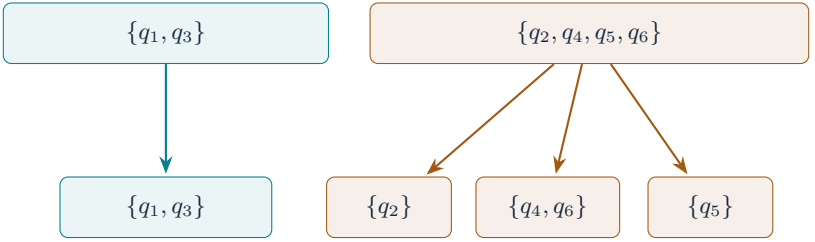
State	ίδιο block	$a \to$	$b \to$	Υπογραφή
q_1	T	N	N	(T, N, N)
q_3	T	N	N	(T, N, N)
q_2	N	N	T	(N, N, T)
q_4	N	T	N	(N, T, N)
q_6	N	T	N	(N, T, N)
q_5	N	N	N	(N, N, N)

Συγκρατούμε τα εξής:

Ίδια υπογραφή $\Rightarrow$ παραμένουν μαζί *σε αυτόν τον γύρο*. Διαφορετική υπογραφή $\Rightarrow$ χωρίζουν.

Το αποτέλεσμα της πρώτης διάσπασης

$$P_1 = \{\{q_1, q_3\},\ \{q_2\},\ \{q_4, q_6\},\ \{q_5\}\}.$$



Ερώτημα

Τελειώσαμε; Πώς ξέρουμε ότι τα q_1, q_3 και τα q_4, q_6 δεν θα χωριστούν στον επόμενο γύρο;

Γύρος 2: δοκιμάζουμε να διασπάσουμε ξανά

Ονομάζουμε τα τέσσερα blocks του P_1:

$$A = \{q_1, q_3\}, \quad B = \{q_2\}, \quad C = \{q_4, q_6\}, \quad D = \{q_5\}.$$

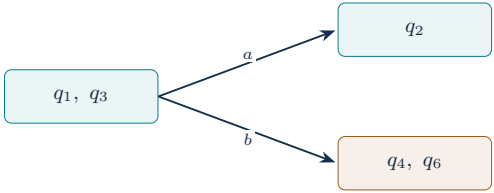
State	δ με a: block	δ με b: block	Υπογραφή
q_1	B	C	(A, B, C)
q_3	B	C	(A, B, C)
q_4	A	D	(C, A, D)
q_6	A	D	(C, A, D)
q_2	D	A	(B, D, A)
q_5	D	D	(D, D, D)

Συγκρατούμε τα εξής:

Μέσα σε κάθε block του P_1, οι υπογραφές είναι ίδιες. **Δεν υπάρχει νέα διάσπαση**: $P_2 = P_1$.

Γιατί το σταθερό partition δίνει ισοδυναμία;

Έχουμε αποδείξει ότι για κάθε block $X \in P_1$ και για κάθε $c \in \{a, b\}$, όλα τα μέλη του X μεταβαίνουν στο **ίδιο block**.



Έτσι, διαβάζοντας *οποιαδήποτε* λέξη z, οι δύο καταστάσεις παραμένουν σε κοινό block. Καθώς τα blocks δεν αναμειγνύουν τελικές με μη τελικές, η τελική απόφαση είναι ίδια.

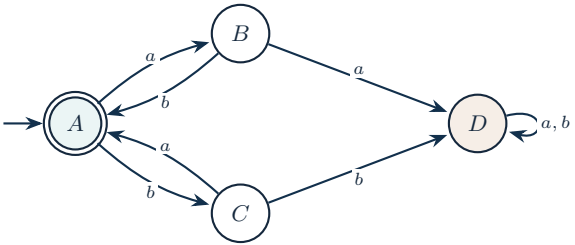
$$q_1 \sim q_3, \qquad q_4 \sim q_6.$$

Συγκρατούμε τα εξής:

Η σταθεροποίηση πιστοποιεί ισοδυναμία για **όλα** τα μήκη, όχι απλώς για 1 ή 2 σύμβολα.

Συγχωνεύουμε καταστάσεις: το quotient DFA

$$A = [q_1] = \{q_1, q_3\}, \quad B = [q_2] = \{q_2\}, \quad C = [q_4] = \{q_4, q_6\}, \quad D = [q_5] = \{q_5\}.$$



	a	b
$\to *A$	B	C
B	D	A
C	A	D
D	D	D

Συγκρατούμε τα εξής:

Το αρχικό DFA έχει 6 καταστάσεις· το quotient έχει **4** και αναγνωρίζει ακριβώς την ίδια γλώσσα.

Αναγνωρίζουμε τη γλώσσα και ελέγχουμε το αποτέλεσμα

Στο quotient, από το A επιστρέφουμε στο A διαβάζοντας **ακριβώς** ab ή ba. Οποιοδήποτε λάθος ζεύγος aa ή bb μας οδηγεί στη νεκρή κατάσταση D.

$$\boxed{L(M) = (ab \cup ba)^*.}$$

Λέξη	Διαδρομή quotient	Απάντηση
$abba$	$A \xrightarrow{a} B \xrightarrow{b} A \xrightarrow{b} C \xrightarrow{a} A$	ναι
$abaa$	$A \xrightarrow{a} B \xrightarrow{b} A \xrightarrow{a} B \xrightarrow{a} D$	όχι
$baba$	$A \xrightarrow{b} C \xrightarrow{a} A \xrightarrow{b} C \xrightarrow{a} A$	ναι

Συγκρατούμε τα εξής:

Η ίδια γλώσσα, με λιγότερες καταστάσεις: κάθε state αποθηκεύει μόνο την πληροφορία που επηρεάζει το μέλλον.

Γιατί δεν μπορεί να ελαχιστοποιηθεί άλλο;

Οι τέσσερις καταστάσεις είναι ανά δύο διακριτές. Αρκούν συγκεκριμένα witnesses:

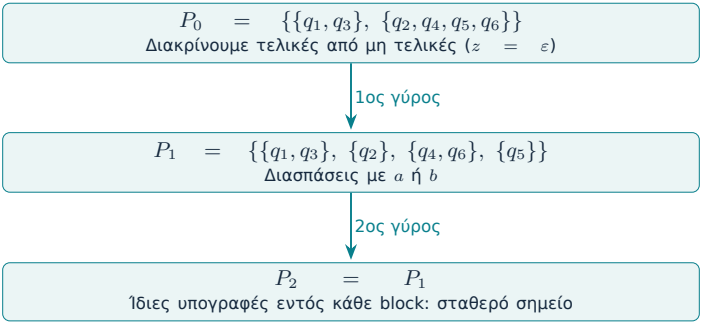
Ζεύγος	Επίθημα z	Τι συμβαίνει;
A, B	ε	μόνο το A είναι τελικό
A, C	ε	μόνο το A είναι τελικό
A, D	ε	μόνο το A είναι τελικό
B, C	b	$B \xrightarrow{b} A$, $C \xrightarrow{b} D$
B, D	b	$B \xrightarrow{b} A$, $D \xrightarrow{b} D$
C, D	a	$C \xrightarrow{a} A$, $D \xrightarrow{a} D$

Ελάχιστος αριθμός καταστάσεων $= 4$.

Συγκρατούμε τα εξής:

Δύο επιχειρήματα: ο αλγόριθμος απέδειξε ισοδυναμία μέσα σε κάθε block και τα διαφορετικά blocks διακρίνονται.

Ο αλγόριθμος σε μία εικόνα



Συγκρατούμε τα εξής:

Αρχική διάκριση → επαναληπτική διάσπαση → σταθερό partition → quotient DFA.

Γιατί μας νοιάζει η ελαχιστοποίηση σε πραγματικά συστήματα;

Μνήμη

Λιγότερες states και μικρότερο transition table.

Ταχύτητα / cache

Μικρότερες δομές έχουν καλύτερη locality.

Verification

Μικρότερο state space για ισοδυναμία, model checking και testing.

Αλλά...

Η ελαχιστοποίηση μειώνει τον **αριθμό states**. Δεν λύνει απαραίτητα το πρόβλημα ότι ένα DFA με μεγάλο αλφάβητο έχει τεράστιο **transition table**.

DFA compression

Από τα regex στις γραμματικές χωρίς συμφραζόμενα (CFG/PDA)

Ένα μικρό πραγματικό αρχείο BibTeX

Αρχείο: `toy_references.bib` (πρώτες δύο εγγραφές)

```
@article{simple2024,
author = {Alice Smith},
title = {Graph Algorithms},
year = {2024}
}

@inproceedings{nested2026,
author = {Alice Smith and Bob Jones},
title = {Graph Algorithms for {{Large Language Models}}},
booktitle = {Proceedings of the Example Conference},
year = {2026}
}
```

Ερώτημα

Μπορούμε με ένα κλασικό regular expression να εντοπίζουμε *πάντα* το τέλος της τιμής του `title`;

Ποιο ακριβώς είναι το πρόβλημα;

Απλή τιμή:

```
title = {Graph Algorithms}
```

Τιμή με εμφώλευση:

```
title = {Graph Algorithms for {{Large Language Models}}}
```

Η κρίσιμη διαφορά

Η πρώτη δεξιά αγκύλη που συναντάμε **δεν** κλείνει απαραίτητα το εξωτερικό `title`.

Για να βρούμε τη σωστή δεξιά αγκύλη, πρέπει να γνωρίζουμε πόσες αριστερές αγκύλες παραμένουν ανοιχτές.

Προσπάθεια 1: Σταματάμε στην πρώτη αγκύλη

Ένα εύλογο μοτίβο για απλές τιμές:

```
\{[^{}]*\}
```

- `\{`: αρχική αριστερή αγκύλη.
- `[^{}]*`: μηδέν ή περισσότεροι χαρακτήρες που δεν είναι αγκύλες.
- `\}`: τελική δεξιά αγκύλη.

```
{Graph Algorithms}                    OK
{Graph Algorithms for {{LLMs}}}       FAIL
```

Γιατί αποτυγχάνει η πρώτη κανονική έκφραση;

Θέλουμε να αναγνωρίσουμε ολόκληρη την τιμή:

```
{Graph Algorithms for {{Large Language Models}}}
```

Πείραμα: Τρέξτε στο terminal

```
echo '{Graph Algorithms for {{Large Language Models}}}' |
grep -oE '\{[^{}]*\}'
```

Τι εμφανίζει το grep;

```
{Large Language Models}
```

Γιατί συμβαίνει αυτό;

Το `[^{}]*` επιτρέπει οποιουσδήποτε χαρακτήρες **εκτός από τις αγκύλες**. Επομένως, η έκφραση δεν αναγνωρίζει ολόκληρη την τιμή, αλλά μόνο το εσωτερικό τμήμα που δεν περιέχει άλλες αγκύλες.

Προσπάθεια 2: Επιτρέπουμε τα πάντα

Μια πιο γενναιόδωρη έκφραση:

```
\{.*\}
```

Αλλά σε μία γραμμή με περισσότερα πεδία:

```
title={A {Nested} Title}, year={2026}
```

Το `.*` είναι άπληστο (greedy): μπορεί να φτάσει μέχρι την **τελευταία** δεξιά αγκύλη της γραμμής, καταναλώνοντας και το `year`.

Γιατί το greedy regex αποτυγχάνει;

Πείραμα

Τρέξτε στο terminal:

```
echo 'title = {Graph {Algorithms}}, year = {2026}' |
grep -oE '\{.*\}'
```

Αποτέλεσμα:

```
{Graph {Algorithms}}, year = {2026}
```

Έστω D η γλώσσα όλων των σωστά ισορροπημένων ακολουθιών από $\{$ και $\}$.
Παραδείγματα μέσα στην D:

$$\varepsilon, \quad \{\}, \quad \{\{\}\}, \quad \{\}\{\}.$$

Παραδείγματα έξω από την D:

$$\}\{, \quad \{\{, \quad \{\}\}.$$

Απόδειξη: Οι ισορροπημένες αγκύλες δεν είναι κανονικές

Ερώτηση

Είναι η D κανονική γλώσσα;

Έστω η **κανονική** γλώσσα

$$R = \{^*\}^*.$$

Δηλαδή, πρώτα όλες οι αριστερές και μετά όλες οι δεξιές αγκύλες. Τότε

$$\boxed{D \cap R = \{\{^n\}^n : n \geq 0\}.}$$

Η γλώσσα στα δεξιά είναι ισομορφική με τη γνωστή μη κανονική $\{a^n b^n : n \geq 0\}$.

Άτοπο μέσω κλειστότητας

Αν η D ήταν κανονική, τότε και η $D \cap R$ θα ήταν κανονική. Όμως δεν είναι. Άρα **η D δεν είναι κανονική**.

Η φυσική συνέχεια: CFG και αυτόματα στοίβας

Για τις ισορροπημένες αγκύλες αρκεί η γραμματική

$$\boxed{S \longrightarrow \{S\}S \mid \varepsilon.}$$

- **DFA / κλασικές regular expressions:** πεπερασμένη μνήμη.
- **CFG+PDA:** Ασυμφραστικές γλώσσες και pushdown automata