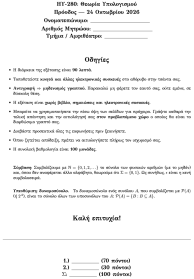
Πρόοδος 24/10

Ασκήσεις: Θεωρία Υπολογισμού · HY-280

Χαράλαμπος Ε. Τσουρακάκης

Υπενθύμιση: Πρόοδος 24/10 (Σάββατο)



ΗΥ-280: Θεωρία Υπολογισμού
Πρόοδος — 24 Οκτωβρίου 2026
Ονοματεπώνυμο: _____
Αριθμός Μητρώου: _____
Τμήμα / Αμφιθέατρο: _____

Οδηγίες

- Η διάρκεια της εξέτασης είναι **90 λεπτά**.
- Τοποθετήστε κινητά και άλλες ηλεκτρονικές συσκευές στο σάκο σας πριν ξεκινήσετε.
- Απαγορεύεται η **μεθοδευμένη** τροποποίηση. Παρακαλούμε μη χάσετε τον χαιρετισμό σας, αυτός χρόνος, σε δύσκολες ώρες.
- Η εξέταση είναι **χωρίς βιβλία, σημειώσεις** και **ηλεκτρονικές συσκευές**.
- Επιτρέπεται να χρησιμοποιήσετε την πλευρά ή τις πλάτες των σελίδων για πρόχειρα. Σημείο σαθρού της κάθε απάντησης για την αντιλαμβάνεστε σας στους **περιβάλλοντα χώρο** ο οποίος θα είναι το διορθώνων χρειαστεί σας.
- Διαβάστε προσεκτικά κάθε ένα ερωτήματος πριν ξεκινήσετε.
- Όταν ζητείται αιτιολόγηση, πρέπει να αιτιολογήσετε πλήρως την απάντησή σας.
- Η συνολική βαθμολογία είναι **100 μονάδες**.

Σύμβαση. Συμβολίζουμε με $\mathbb{N} = \{0, 1, 2, \ldots\}$ το σύνολο των φυσικών αριθμών (και το μηδέν) και, όπου δεν αναφέρεται άλλο αλφάβητο, θεωρούμε ότι $\Sigma = \{0, 1\}$. Ως συνήθως, ε είναι η κενή συμβολοσειρά.

Υποδείξεις διαφορετικούς. Τα διαγωνιοποίηση τις συνάρ. f που συμβολίζονται με $T(A)$ ή T^A, όπου το σύνολο όλων των υποσυνόλων των A. Π.χ. $\mathcal{P}(A) = \{B : B \subseteq A\}$.

Καλή επιτυχία!

1.) _____ **(30 πόντοι)**
2.) _____ **(20 πόντοι)**
Σ.) _____ **(100 πόντοι)**

- Θα ήταν καλό να προσέλθετε μεταξύ 10:30 και 10:50, ώστε να ξεκινήσουμε χωρίς καθυστερήσεις.
- Μην ξεχάσετε να έχετε μαζί σας την ταυτότητά σας!
- Προσπαθήστε να είστε στις θέσεις σας έως τις 10:55, ώστε να είμαστε όλοι έτοιμοι για την έναρξη της εξέτασης στις 11:00.

Προετοιμασία για την Πρόοδο

Τι να διαβάσετε

Επαναλάβετε τη θεωρία, τους ορισμούς, τα παραδείγματα και τις ασκήσεις που έχουμε δουλέψει στις διαλέξεις.

Στόχος δεν είναι η απομνημόνευση λύσεων, αλλά η κατανόηση των βασικών εννοιών και η ικανότητα εφαρμογής τους σε νέα παραδείγματα.

- **Σύνολα, σχέσεις και συναρτήσεις.**
- **Γλώσσες και πράξεις γλωσσών.**
- **Αριθμησιμότητα και υπολογισιμότητα.**
- **Αποφασισιμότητα και αναγωγές.**
- **Κανονικές γλώσσες και πεπερασμένα αυτόματα.**

Σημαντικό

Να μπορείτε να εξηγείτε *γιατί* ισχύει ένας ισχυρισμός, όχι μόνο να αναφέρετε αν είναι σωστός ή λανθασμένος.

Επανάληψη Ι: Μαθηματικά εργαλεία

Σύνολα και σχέσεις

- Βασικές πράξεις συνόλων, δυναμοσύνολο, καρτεσιανό γινόμενο.
- Σχέσεις και βασικές ιδιότητές τους.
- Μεταβατικό κλείσιμο μιας σχέσης.

Συναρτήσεις και αριθμησιμότητα

- Ορισμοί: συνάρτηση 1-1, επί, αμφιμονοσήμαντη αντιστοιχία.
- Πώς αποδεικνύουμε ότι μια συνάρτηση είναι 1-1 ή επί;
- Πεπερασμένα, αριθμήσιμα και μη αριθμήσιμα σύνολα.
- Απαριθμήσεις και τεχνική dovetailing.

Να εξασκηθείτε: στην κατασκευή απλών συναρτήσεων, στην εύρεση αντιπαραδειγμάτων και στη σύντομη αλλά πλήρη αιτιολόγηση.

Επανάληψη II: Γλώσσες και κανονικές εκφράσεις

Βασικές έννοιες

- Αλφάβητο, συμβολοσειρά, κενή συμβολοσειρά ε, γλώσσα, Σ^*.
- Ένωση, συνένωση και επανάληψη Kleene.
- Η διαφορά μεταξύ $\varnothing$ και $\{\varepsilon\}$.
- Πράξεις L^* και L^+.

Κανονικές εκφράσεις

- Πώς ερμηνεύουμε μια κανονική έκφραση.
- Πώς περιγράφουμε μια γλώσσα με κανονική έκφραση.
- Πώς αιτιολογούμε ότι δύο διαφορετικές εκφράσεις περιγράφουν την ίδια γλώσσα.
- Σχέση κανονικών εκφράσεων και πεπερασμένων αυτομάτων.

Συμβουλή: Για κάθε έκφραση, προσπαθήστε να περιγράψετε με λόγια ποιες συμβολοσειρές παράγει.

Επανάληψη III: Υπολογισιμότητα

Θεωρία

- Τι σημαίνει ότι ένα πρόβλημα είναι υπολογίσιμο;
- Τι σημαίνει ότι ένα πρόβλημα είναι αποφασίσιμο;
- Ο ρόλος του τερματισμού ενός αλγορίθμου.
- Το γενικό πρόβλημα τερματισμού (Halting).

Αναγωγές

- Τι σημαίνει ότι ένα πρόβλημα A ανάγεται σε ένα πρόβλημα B;
- Πώς χρησιμοποιούμε έναν υποθετικό αλγόριθμο για να καταλήξουμε σε άτοπο;
- Γιατί έχει σημασία η κατεύθυνση της αναγωγής;
- Γιατί η ίδια η αναγωγή πρέπει να είναι υπολογίσιμη;

Στόχος

Να μπορείτε να παρουσιάσετε μια αναγωγή με σαφή βήματα και να εξηγήσετε ποια υπόθεση οδηγεί στην αντίφαση.

Επανάληψη IV: Πεπερασμένα αυτόματα

DFA, NFA και ε-NFA

- Τυπικοί ορισμοί και διαφορές μεταξύ των μοντέλων.
- Συνάρτηση μετάβασης και επεκτεταμένη συνάρτηση μετάβασης.
- Αρχικές, τελικές και μη προσβάσιμες καταστάσεις.
- Εκτέλεση ενός αυτομάτου σε μια συμβολοσειρά.
- Κατασκευή αυτομάτων για απλές γλώσσες.

Ισοδυναμία και μετατροπές

- Πότε δύο αυτόματα αναγνωρίζουν την ίδια γλώσσα;
- Μετατροπές μεταξύ διαφορετικών μοντέλων.
- Σχέση DFA, NFA και κανονικών εκφράσεων.
- Κατανόηση και εφαρμογή των αλγορίθμων που παρουσιάστηκαν στο μάθημα.

Πώς να προετοιμαστείτε αποτελεσματικά

1. **Μάθετε τους ορισμούς.**
   Να μπορείτε να τους διατυπώσετε με ακρίβεια και να εξηγήσετε τι σημαίνουν.

2. **Λύστε ξανά ασκήσεις χωρίς να βλέπετε τη λύση.**
   Ιδιαίτερα όσες παρουσιάσαμε αναλυτικά στις διαλέξεις.

3. **Εξασκηθείτε σε αποδείξεις και αντιπαραδείγματα.**
   Μια σωστή απάντηση πρέπει να συνοδεύεται από την κατάλληλη αιτιολόγηση.

4. **Ελέγξτε ότι καταλαβαίνετε κάθε βήμα.**
   Μη μαθαίνετε μηχανικά τους αλγορίθμους. Προσπαθήστε να εξηγήσετε γιατί λειτουργούν.

5. **Κάντε επανάληψη υπό χρονικό περιορισμό.**
   Εξασκηθείτε στο να γράφετε καθαρές, σύντομες και ολοκληρωμένες απαντήσεις.

Επισήμανση

Η παρακολούθηση των διαλέξεων και η συστηματική εξάσκηση στις σειρές των ασκήσεων είναι η καλύτερη προετοιμασία.

Άσκηση: Αναγωγή Halting

Γνωρίζουμε ότι το γενικό πρόβλημα τερματισμού είναι **μη αποφασίσιμο**: δεν υπάρχει αλγόριθμος που, για κάθε πρόγραμμα P και είσοδο D, αποφασίζει αν το $P(D)$ τερματίζει.

Κάποιος ισχυρίζεται ότι το πρόβλημα γίνεται αποφασίσιμο αν περιοριστούμε σε **προγράμματα χωρίς είσοδο**.
Υποθέτει ότι υπάρχει ελεγκτής $H_0(Q)$ που τερματίζει πάντοτε και απαντά σωστά αν το $Q()$ τερματίζει.

Ερωτήματα

1. Δοθέντων των P, D, κατασκευάστε πρόγραμμα $Q_{P,D}$ χωρίς είσοδο, τέτοιο ώστε

$$Q_{P,D}() \text{ τερματίζει } \iff P(D) \text{ τερματίζει.}$$

2. Πώς θα χρησιμοποιούσατε το H_0 για να λύσετε το γενικό Halting;
3. Συμπληρώστε την κατεύθυνση της αναγωγής:

$$\underline{\qquad\qquad} \leq \underline{\qquad\qquad}.$$

4. Χρειάζεται να εκτελέσουμε το $P(D)$ για να κατασκευάσουμε το $Q_{P,D}$;

Λύση: Κατασκευή του $Q_{P,D}$

Βήμα 1. Μας δίνονται ένα αυθαίρετο πρόγραμμα P και μια είσοδος D.

Θέλουμε να κατασκευάσουμε ένα νέο πρόγραμμα $Q_{P,D}$ που **δεν δέχεται είσοδο**.

Κατασκευή

```
def Q():
    P(D)
    return
```

Οι περιγραφές των P, D είναι **ενσωματωμένες στον κώδικα** του $Q_{P,D}$.

- Αν το $P(D)$ τερματίσει, η εκτέλεση συνεχίζεται στο `return` και το $Q()$ τερματίζει.
- Αν το $P(D)$ δεν τερματίσει, το $Q()$ παραμένει για πάντα στην εκτέλεση του $P(D)$.

Συμπέρασμα

$$\boxed{Q_{P,D}() \downarrow \iff P(D) \downarrow}$$

όπου $\downarrow$ σημαίνει «τερματίζει».

Λύση: Πώς χρησιμοποιούμε τον H_0;

Βήμα 2. Υποθέτουμε ότι υπάρχει ο ελεγκτής H_0, ο οποίος αποφασίζει αν ένα πρόγραμμα χωρίς είσοδο τερματίζει.

Τότε μπορούμε να κατασκευάσουμε έναν αλγόριθμο για το **γενικό Halting**:

Υποθετικός αλγόριθμος

```
def HALT(P, D):
    Q = construct_Q(P, D)
    return H0(Q)
```



Επειδή ο H_0 τερματίζει πάντοτε και

$$H_0(Q_{P,D}) = \text{ΝΑΙ} \iff P(D) \text{ τερματίζει},$$

ο αλγόριθμος HALT θα αποφάσιζε το γενικό πρόβλημα τερματισμού.

Άτοπο!

Το γενικό Halting είναι μη αποφασίσιμο. Άρα ο υποθετικός ελεγκτής H_0 δεν υπάρχει.

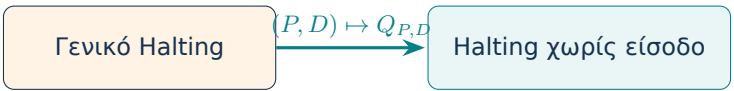
Λύση: Ποια είναι η κατεύθυνση της αναγωγής;

Τι αποδείξαμε;

Αν μπορούσαμε να λύσουμε τον **τερματισμό χωρίς είσοδο**, τότε θα μπορούσαμε να λύσουμε και τον **γενικό τερματισμό**.

Άρα η αναγωγή έχει κατεύθυνση

$$\text{Γενικό Halting} \leq_m \text{Halting χωρίς είσοδο}$$



Προσοχή: Αναγάγουμε το πρόβλημα που *γνωρίζουμε ότι είναι μη αποφασίσιμο* στο πρόβλημα που *θέλουμε να αποδείξουμε ότι είναι μη αποφασίσιμο*.

Λύση: Κατασκευή $\neq$ Εκτέλεση

Κρίσιμο ερώτημα

Για να κατασκευάσουμε το $Q_{P,D}$, χρειάζεται να εκτελέσουμε πρώτα το $P(D)$;

Όχι!

Κατασκευή του $Q_{P,D}$

- Γράφουμε τον κώδικα του Q.
- Ενσωματώνουμε τα P, D.
- Η διαδικασία ολοκληρώνεται σε πεπερασμένο χρόνο.

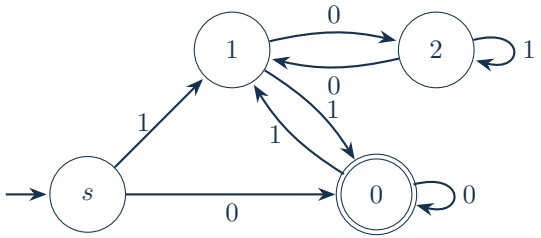
Εκτέλεση του $Q_{P,D}$

- Ξεκινά η προσομοίωση του $P(D)$.
- Μπορεί να τερματίσει.
- Μπορεί να συνεχίζεται για πάντα.

Γιατί είναι απαραίτητο;

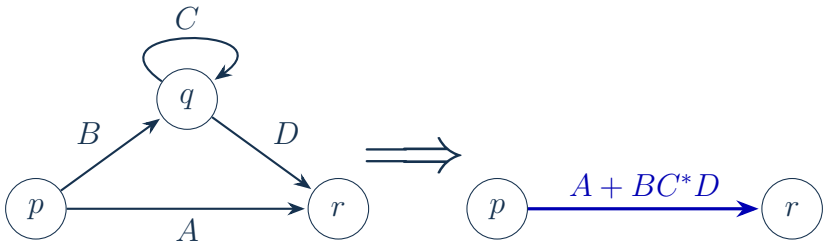
Μια αναγωγή πρέπει να είναι **υπολογίσιμη σε πεπερασμένο χρόνο**.
Αν χρειαζόταν πρώτα να περιμένουμε να τερματίσει το $P(D)$, τότε η ίδια η αναγωγή μπορεί να μην ολοκληρωνόταν ποτέ!

Άσκηση: Ποια η κανονική έκφραση που αντιστοιχεί στο αυτόματο;



Άσκηση: Βρείτε μια κανονική έκφραση που περιγράφει τη γλώσσα που αναγνωρίζει το παραπάνω αυτόματο.

Απαλοιφή μίας κατάστασης: ο βασικός κανόνας



- Θέλουμε να **αφαιρέσουμε την κατάσταση** q χωρίς να αλλάξουμε τη γλώσσα των διαδρομών από p προς r.
- Είτε πηγαίνουμε **απευθείας** με A, είτε περνάμε από την q: B, έπειτα C **μηδέν ή περισσότερες φορές**, και τέλος D.

Νέα ετικέτα: $R'_{pr} = A + BC^*D$

Γιατί είναι σωστός ο κανόνας;

1. Μια διαδρομή από p σε r μπορεί να **μην επισκεφθεί** την q:

$$R(p \to r).$$

2. Ή μπορεί να περάσει από την q, να κάνει **οποιονδήποτε αριθμό** επαναλήψεων στην q, και να φύγει:

$$R(p \to q) \quad \underbrace{R(q \to q)^*}_{0,1,2,\ldots \text{ επαναλήψεις}} \quad R(q \to r).$$

3. Η ένωση $+$ καλύπτει **και τις δύο περιπτώσεις**.

Προσοχή

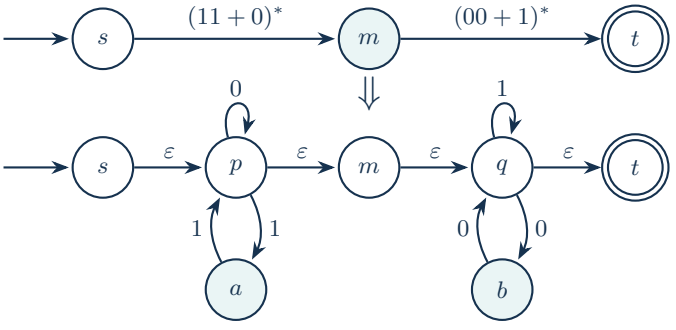
Το * είναι απαραίτητο: η διαδρομή ενδέχεται να περάσει από την q πολλές φορές, ακόμη και καμία επανάληψη του βρόχου.

Θυμηθείτε: Από κανονική έκφραση σε ε-NFA

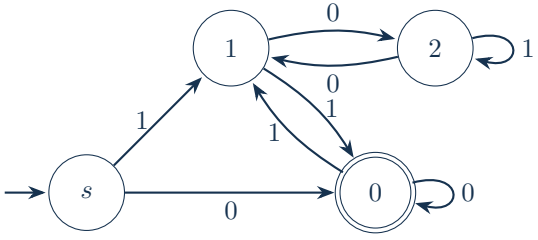
Αντίστροφη διαδικασία: Κανονική έκφραση -> ε-NFA

Κατασκευάστε ένα ε-NFA που αναγνωρίζει τη γλώσσα της κανονικής έκφρασης

$$r = (11 + 0)^*(00 + 1)^*.$$



Το αρχικό αυτόματο



Προετοιμασία για απαλοιφή καταστάσεων (1/2)

Πριν αρχίσουμε να απαλείφουμε καταστάσεις, μετατρέπουμε το αρχικό αυτόματο ώστε να έχει:

1. Μία **νέα αρχική κατάσταση** s, χωρίς εισερχόμενες ακμές.
2. Μία **νέα τελική κατάσταση** t, χωρίς εξερχόμενες ακμές.

Πώς το πετυχαίνουμε;

- Προσθέτουμε νέα αρχική κατάσταση s και μία ε-μετάβαση

$$s \xrightarrow{\varepsilon} q_0,$$

όπου q_0 η παλιά αρχική κατάσταση.

- Προσθέτουμε νέα τελική κατάσταση t και μία ε-μετάβαση

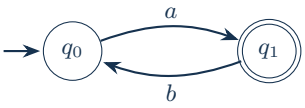
$$f \xrightarrow{\varepsilon} t$$

από κάθε παλιά τελική κατάσταση f. Οι παλιές τελικές παύουν να είναι τελικές.

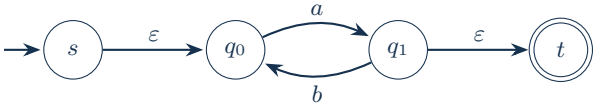
Γιατί ε; Επειδή δεν καταναλώνει κανένα σύμβολο. Επομένως, η γλώσσα δεν αλλάζει.

Προετοιμασία για απαλοιφή καταστάσεων (2/2)

Αρχικό αυτόματο:



Μετά την προσθήκη των s, t:



Γιατί κάνουμε αυτή την προετοιμασία;

Η παλιά αρχική κατάσταση μπορεί να έχει εισερχόμενες ακμές και οι παλιές τελικές μπορεί να έχουν εξερχόμενες ακμές.

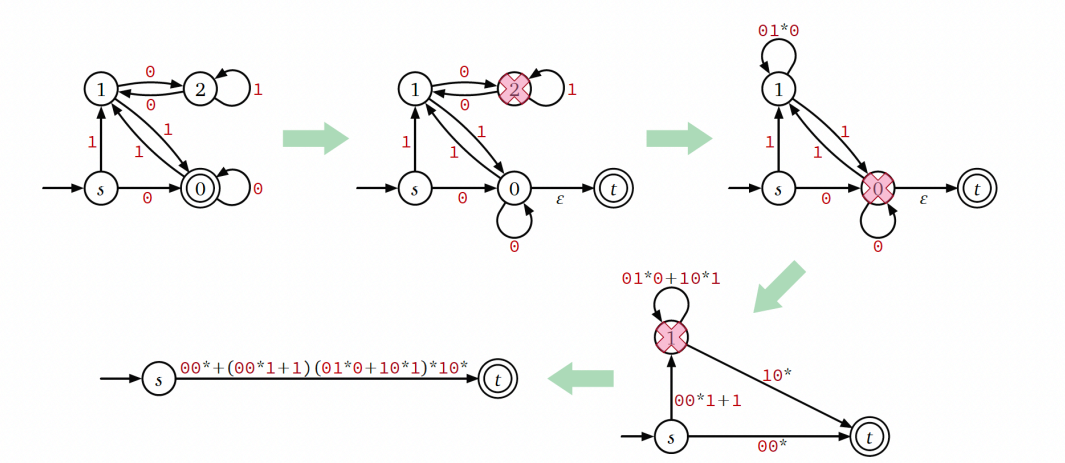
Με την προσθήκη των s, t, μπορούμε να **απαλείψουμε όλες τις παλιές καταστάσεις**, ακόμη και την παλιά αρχική και τις παλιές τελικές.

Ο κανόνας εφαρμόζεται σε ΟΛΑ τα ζεύγη

$$\boxed{R'_{pq} = R_{pq} + R_{pk}(R_{kk})^* R_{kq}}$$

- Για κάθε p, q που παραμένουν, **ακόμη κι αν δεν υπάρχει αρχική ακμή** $p \to q$.
- Αν δεν υπάρχει ακμή: $R_{pq} = \varnothing$.
- $\varnothing + X = X$, $\varnothing X = X\varnothing = \varnothing$, $\varnothing^* = \varepsilon$, $\varepsilon X = X$.
- Όλες οι ενημερώσεις ενός βήματος υπολογίζονται από τις **παλιές** ετικέτες.

Από DFA σε κανονική έκφραση: Παράδειγμα



Πηγή: Jeff Erickson, *Models of Computation*, Lecture 4: Nondeterministic Automata (Spring 2018).

Απαλοιφή καταστάσεων με Python

```python
if __name__ == '__main__':
  states = ['s', 'q0', 'q1', 'q2', 'q3', 'f']

  R = {
    ('s', 'q0'): 'ε',

    ('q0', 'q1'): 'a',
    ('q0', 'q2'): 'b',

    ('q1', 'q1'): 'a',
    ('q1', 'q3'): 'b',

    ('q2', 'q2'): 'b',
    ('q2', 'q3'): 'a',

    ('q3', 'q3'): '(a+b)',
    ('q3', 'f'): 'ε',
  }

  eliminate(
    states, R, 's', 'f',
    ['q1', 'q2', 'q0', 'q3']
  )

  show(
    'Simplified by inspection:',
    r'\boxed{(a+b)^*a(a+b)^*b(a+b)^*'
    r'+(a+b)^*b(a+b)^*a(a+b)^*}'
  )
```

Δοκιμάστε τον αλγόριθμο!

Άνοιγμα του Python notebook στο Google Colab

Απαλοιφή κατάστασης 2 — ζεύγη 1–4 από 16

Κανόνας: $R\prime_{pq} = R_{pq} + R_{p2}(R_{22})^{*}R_{2q}$.

$$
\begin{aligned}
R\prime_{ss} &= \varnothing + \left(\varnothing\right) \cdot 1^{*} \cdot \left(\varnothing\right) \\
&= \varnothing \\
R\prime_{s0} &= 0 + \left(\varnothing\right) \cdot 1^{*} \cdot \left(\varnothing\right) \\
&= 0 \\
R\prime_{s1} &= 1 + \left(\varnothing\right) \cdot 1^{*} \cdot \left(0\right) \\
&= 1 \\
R\prime_{st} &= \varnothing + \left(\varnothing\right) \cdot 1^{*} \cdot \left(\varnothing\right) \\
&= \varnothing
\end{aligned}
$$

Απαλοιφή 2 — ζεύγη 5–8 από 16

Κανόνας: $R\prime_{pq} = R_{pq} + R_{p2}(R_{22})^{*}R_{2q}$.

$$
\begin{aligned}
R\prime_{0s} &= \varnothing + \left(\varnothing\right) \cdot 1^{*} \cdot \left(\varnothing\right) \\
&= \varnothing \\
R\prime_{00} &= 0 + \left(\varnothing\right) \cdot 1^{*} \cdot \left(\varnothing\right) \\
&= 0 \\
R\prime_{01} &= 1 + \left(\varnothing\right) \cdot 1^{*} \cdot \left(0\right) \\
&= 1 \\
R\prime_{0t} &= \varepsilon + \left(\varnothing\right) \cdot 1^{*} \cdot \left(\varnothing\right) \\
&= \varepsilon
\end{aligned}
$$

Απαλοιφή 2 — ζεύγη 9–12 από 16

Κανόνας: $R\prime_{pq} = R_{pq} + R_{p2}(R_{22})^{*}R_{2q}$.

$$
\begin{aligned}
R\prime_{1s} &= \varnothing + \big(0\big)\cdot 1^{*}\cdot \big(\varnothing\big) \\
&= \varnothing \\
R\prime_{10} &= 1 + \big(0\big)\cdot 1^{*}\cdot \big(\varnothing\big) \\
&= 1 \\
R\prime_{11} &= \varnothing + \big(0\big)\cdot 1^{*}\cdot \big(0\big) \\
&= 01^{*}0 \\
R\prime_{1t} &= \varnothing + \big(0\big)\cdot 1^{*}\cdot \big(\varnothing\big) \\
&= \varnothing
\end{aligned}
$$

Απαλοιφή 2 — ζεύγη 13–16 από 16

Κανόνας: $R\prime_{pq} = R_{pq} + R_{p2}(R_{22})^{*}R_{2q}$.

$$
\begin{aligned}
R\prime_{ts} &= \varnothing + \left(\varnothing\right) \cdot 1^{*} \cdot \left(\varnothing\right) \\
&= \varnothing \\
R\prime_{t0} &= \varnothing + \left(\varnothing\right) \cdot 1^{*} \cdot \left(\varnothing\right) \\
&= \varnothing \\
R\prime_{t1} &= \varnothing + \left(\varnothing\right) \cdot 1^{*} \cdot \left(0\right) \\
&= \varnothing \\
R\prime_{tt} &= \varnothing + \left(\varnothing\right) \cdot 1^{*} \cdot \left(\varnothing\right) \\
&= \varnothing
\end{aligned}
$$

Μετά την απαλοιφή της 2 — όλες οι μη κενές ακμές

$$
\begin{aligned}
R_{s0} &= 0 \\
R_{s1} &= 1 \\
R_{00} &= 0 \\
R_{01} &= 1 \\
R_{0t} &= \varepsilon \\
R_{10} &= 1 \\
R_{11} &= 01^*0
\end{aligned}
$$

Καταστάσεις που απομένουν: $s, 0, 1, t$.

Απαλοιφή 0 — ζεύγη 1–4 από 9

Κανόνας: $R\prime_{pq} = R_{pq} + R_{p0}(R_{00})^{*}R_{0q}$.

$$
\begin{aligned}
R\prime_{ss} &= \varnothing + \big(0\big)\cdot 0^{*}\cdot\big(\varnothing\big) \\
&= \varnothing \\
R\prime_{s1} &= 1 + \big(0\big)\cdot 0^{*}\cdot\big(1\big) \\
&= 1 + 00^{*}1 \\
R\prime_{st} &= \varnothing + \big(0\big)\cdot 0^{*}\cdot\big(\varepsilon\big) \\
&= 00^{*} \\
R\prime_{1s} &= \varnothing + \big(1\big)\cdot 0^{*}\cdot\big(\varnothing\big) \\
&= \varnothing
\end{aligned}
$$

Απαλοιφή 0 — ζεύγη 5–8 από 9

Κανόνας: $R\prime_{pq} = R_{pq} + R_{p0}(R_{00})^{*}R_{0q}$.

$$
\begin{aligned}
R\prime_{11} &= 01^{*}0 + \big(1\big) \cdot 0^{*} \cdot \big(1\big) \\
&= 01^{*}0 + 10^{*}1 \\
R\prime_{1t} &= \varnothing + \big(1\big) \cdot 0^{*} \cdot \big(\varepsilon\big) \\
&= 10^{*} \\
R\prime_{ts} &= \varnothing + \big(\varnothing\big) \cdot 0^{*} \cdot \big(\varnothing\big) \\
&= \varnothing \\
R\prime_{t1} &= \varnothing + \big(\varnothing\big) \cdot 0^{*} \cdot \big(1\big) \\
&= \varnothing
\end{aligned}
$$

Απαλοιφή 0 — ζεύγη 9–9 από 9

Κανόνας: $R\prime_{pq} = R_{pq} + R_{p0}(R_{00})^{*}R_{0q}$.

$$\begin{aligned} R\prime_{tt} &= \varnothing + \big(\varnothing\big) \cdot 0^{*} \cdot \big(\varepsilon\big) \\ &= \varnothing \end{aligned}$$

Μετά την απαλοιφή της 0 — όλες οι μη κενές ακμές

$$
\begin{aligned}
R_{s1} &= 1 + 00^*1 \\
R_{st} &= 00^* \\
R_{11} &= 01^*0 + 10^*1 \\
R_{1t} &= 10^*
\end{aligned}
$$

Καταστάσεις που απομένουν: $s, 1, t$.

Απαλοιφή 1 — ζεύγη 1–4 από 4

Κανόνας: $R\prime_{pq} = R_{pq} + R_{p1}(R_{11})^* R_{1q}$.

$$
\begin{aligned}
R\prime_{ss} &= \varnothing + \left(1 + 00^*1\right) \cdot (01^*0 + 10^*1)^* \cdot \left(\varnothing\right) \\
&= \varnothing \\
R\prime_{st} &= 00^* + \left(1 + 00^*1\right) \cdot (01^*0 + 10^*1)^* \cdot \left(10^*\right) \\
&= 00^* + (1 + 00^*1)(01^*0 + 10^*1)^* 10^* \\
R\prime_{ts} &= \varnothing + \left(\varnothing\right) \cdot (01^*0 + 10^*1)^* \cdot \left(\varnothing\right) \\
&= \varnothing \\
R\prime_{tt} &= \varnothing + \left(\varnothing\right) \cdot (01^*0 + 10^*1)^* \cdot \left(10^*\right) \\
&= \varnothing
\end{aligned}
$$

Μετά την απαλοιφή της 1 — όλες οι μη κενές ακμές

$$R_{st} = 00^* + (1 + 00^*1)(01^*0 + 10^*1)^*10^*$$

Καταστάσεις που απομένουν: s, t.

Τελικό αποτέλεσμα

$$\boxed{00^* + (1 + 00^*1)(01^*0 + 10^*1)^*10^*}$$

$$\boxed{00^* + (00^*1 + 1)(01^*0 + 10^*1)^*10^*}$$

Οι δύο γραφές είναι ίδιες λόγω αντιμεταθετικότητας της ένωσης.

Έλεγχος: η γλώσσα περιέχει τις μη κενές δυαδικές αναπαραστάσεις πολλαπλασίων του 3 (με αρχικά μηδενικά).

Άσκηση 3: Dovetailing

Έστω τα σύνολα

$$A = \{0, 2, 4, 6, \ldots\}, \qquad B = \{-1, -2, -3, -4, \ldots\}.$$

Ερωτήματα

1. Γράψτε τους πρώτους 8 όρους της απαρίθμησης του $A \cup B$ που παράγει το **dovetailing**.
2. Δώστε έναν **κλειστό τύπο** για τη συνάρτηση

$$f : \mathbb{N} \longrightarrow A \cup B, \qquad \mathbb{N} = \{0, 1, 2, \ldots\}.$$

3. Αποδείξτε ότι η f είναι **1-1**.
4. Αποδείξτε ότι η f είναι **επί**.

Τι θέλουμε να αποδείξουμε;

Ότι η f είναι μια **αμφιμονοσήμαντη αντιστοιχία** μεταξύ $\mathbb{N}$ και $A \cup B$.

Λύση: Κατασκευή της απαρίθμησης

Απαριθμούμε ξεχωριστά τα δύο σύνολα:

$$a_k = 2k, \qquad b_k = -(k+1), \qquad k \in \mathbb{N}.$$

k	0	1	2	3
a_k	0	2	4	6
b_k	-1	-2	-3	-4

Dovetailing: Εναλλαγή των δύο απαριθμήσεων

Παίρνουμε διαδοχικά έναν όρο από το A και έναν από το B:

$$\boxed{0,\ -1,\ 2,\ -2,\ 4,\ -3,\ 6,\ -4,\ \ldots}$$

Παρατήρηση: Τα A και B είναι ξένα μεταξύ τους:

$$A \cap B = \varnothing.$$

Επομένως, δεν εμφανίζονται επαναλήψεις κατά την εναλλαγή.

Λύση: Κλειστός τύπος της f

Από την απαρίθμηση παρατηρούμε ότι:

- Στις **άρτιες θέσεις** $n = 2k$, επιλέγουμε τον k-οστό όρο του A.
- Στις **περιττές θέσεις** $n = 2k+1$, επιλέγουμε τον k-οστό όρο του B.

Άρα

$$\boxed{f(2k) = 2k, \qquad f(2k+1) = -(k+1)}$$

Ισοδύναμα, ο κλειστός τύπος είναι

$$\boxed{f(n) = \begin{cases} n, & n \text{ άρτιος}, \\ -\dfrac{n+1}{2}, & n \text{ περιττός}. \end{cases}}$$

Έλεγχος

$$f(0) = 0, \quad f(1) = -1,$$
$$f(2) = 2, \quad f(3) = -2,$$
$$f(4) = 4, \quad f(5) = -3.$$

Λύση: Απόδειξη ότι η f είναι 1-1

Ορισμός

Η f είναι 1-1 αν για κάθε $n, m \in \mathbb{N}$,

$$f(n) = f(m) \implies n = m.$$

Έστω $n, m \in \mathbb{N}$ με $f(n) = f(m)$.

Περίπτωση 1: Το n είναι άρτιο.

Τότε $f(n) = n \geq 0$. Επειδή $f(m) = f(n) \geq 0$, το m είναι επίσης άρτιο.

Συνεπώς,

$$n = f(n) = f(m) = m.$$

Περίπτωση 2: Το n είναι περιττό.

Τότε $f(n) < 0$. Άρα και το m είναι περιττό.

Επομένως,

$$-\frac{n+1}{2} = -\frac{m+1}{2} \implies n = m.$$

Συμπέρασμα

Λύση: Απόδειξη ότι η f είναι επί

Ορισμός

Η $f : \mathbb{N} \to A \cup B$ είναι επί αν

$$\forall y \in A \cup B \; \exists n \in \mathbb{N} : f(n) = y.$$

Έστω αυθαίρετο $y \in A \cup B$.

Περίπτωση 1: $y \in A$**.**
Τότε $y = 2k$ για κάποιο $k \in \mathbb{N}$.
Επιλέγουμε $n = 2k$ και έχουμε

$$f(n) = f(2k) = 2k = y.$$

Περίπτωση 2: $y \in B$**.**
Τότε $y = -k$ για κάποιο ακέραιο $k \geq 1$.
Επιλέγουμε $n = 2k - 1 \in \mathbb{N}$ και έχουμε

$$f(n) = f(2k - 1) = -k = y.$$

Συμπέρασμα

Κάθε στοιχείο του $A \cup B$ έχει προεικόνα στο $\mathbb{N}$. Άρα η f είναι **επί**.