

HY-280 Θεωρία Υπολογισμού

Φροντιστήριο 11

12/12/2025

Άσκηση 1. Δοθέντος μη-κατευθυνόμενου γράφου $G, G = (V, E)$, λέμε το υποσύνολο κορυφών $V' \subseteq V$ κάλυμμα κορυφών αν κάθε ακμή $(u, v) \in E$ είναι προσπίπτουσα σε τουλάχιστον μια κορυφή του $a \in V'$. Δίνεται η γλώσσα

$$\text{ΚΑΛΥΜΜΑ ΚΟΡΥΦΩΝ} = \left\{ \langle G, k \rangle \mid \begin{array}{l} G \text{ μη-κατευθυνόμενος γράφος με κάλυμμα} \\ \text{κορυφών μεγέθους το πολύ } k \end{array} \right\}.$$

Να δείξετε ότι $\text{ΚΑΛΥΜΜΑ ΚΟΡΥΦΩΝ} \in \text{NP}$.

Λύση. Θα κατασκευάσουμε έναν πολυωνυμικό επαληθευτή V με πιστοποιητικό c .

$V =$ “Στην είσοδο $\langle G, k, c \rangle$:

1. Ελέγχει αν το c αποτελείται από κορυφές του G .
2. Ελέγχει αν το $|c| \leq k$.
3. Αν 1. και 2. εντάξει, ελέγχει αν κάθε ακμή στο G είναι προσπίπτουσα στο c .
4. Αν ο έλεγχος είναι επιτυχής, αποδέχεται.
5. Αλλιώς, απορρίπτει.”

Στο βήμα 3. χρειαζόμαστε $O(n^2)$ χρόνο. Συνεπώς, το ΚΑΛΥΜΜΑ ΚΟΡΥΦΩΝ επαληθεύεται σε πολυωνυμικό χρόνο. Άρα, $\text{ΚΑΛΥΜΜΑ ΚΟΡΥΦΩΝ} \in \text{NP}$. $\square$

Τώρα παρουσιάζουμε μια άλλη γλώσσα η οποία θα μας χρησιμεύσει στην επόμενη άσκηση. Έστω μη-κατευθυνόμενος γράφος $G, G = (V, E)$, λέμε το υποσύνολο κορυφών $\hat{V} \subseteq V$ ανεξάρτητο σύνολο αν δεν έχει ζεύγη κορυφών οι οποίες να συνδέονται μέσω ακμής στο G . Ορίζουμε την γλώσσα ως εξής,

$$\text{ΑΝΕΞΑΡΤΗΤΟ ΣΥΝΟΛΟ} = \left\{ \langle G, k \rangle \mid \begin{array}{l} G \text{ μη-κατευθυνόμενος γράφος και } k \text{ το} \\ \text{πλήθος των κορυφών που δεν συνδέονται} \\ \text{ανά δύο μέσω ακμής του } G \end{array} \right\}^1.$$

¹Στη Διάλεξη 20 ονομάσαμε αυτή τη γλώσσα INDSET .

Άσκηση 2. Έστω ότι ΑΝΕΞΑΡΤΗΤΟ ΣΥΝΟΛΟ $\in$ NP-πλήρες. Δίνεται η γλώσσα

$$\text{ΚΑΛΥΜΜΑ ΚΟΡΥΦΩΝ} = \left\{ \langle G, k \rangle \mid \begin{array}{l} G \text{ μη-κατευθυνόμενος γράφος με κάλυμμα} \\ \text{κορυφών μεγέθους το πολύ } k \end{array} \right\}.$$

Να δείξετε ότι ΚΑΛΥΜΜΑ ΚΟΡΥΦΩΝ $\in$ NP-πλήρες.

Λύση. Η πρώτη συνθήκη για την συμμετοχή του ΚΑΛΥΜΜΑ ΚΟΡΥΦΩΝ στην κλάση των NP-πλήρων δίνεται από την Άσκηση 1.

Σύμφωνα με την εκφώνηση το ΑΝΕΞΑΡΤΗΤΟ ΣΥΝΟΛΟ είναι NP-πλήρες πρόβλημα. Θα το χρησιμοποιήσουμε για να αποδείξουμε ότι και το ΚΑΛΥΜΜΑ ΚΟΡΥΦΩΝ είναι επίσης NP-πλήρες πρόβλημα. Δηλαδή, θα κάνουμε αναγωγή **από το** πρόβλημα ΑΝΕΞΑΡΤΗΤΟ ΣΥΝΟΛΟ **στο** πρόβλημα ΚΑΛΥΜΜΑ ΚΟΡΥΦΩΝ. Η αναγωγή αυτή θέλουμε να είναι πολυωνυμική, δηλαδή να υπάρχει μια πολυωνυμικού χρόνου² υπολογίσιμη συνάρτηση για την οποία να ισχύει ΑΝΕΞΑΡΤΗΤΟ ΣΥΝΟΛΟ $\leq_P$ ΚΑΛΥΜΜΑ ΚΟΡΥΦΩΝ. Κατασκευάζουμε την πολυωνυμική αναγωγή,

Αναγωγή f :

Είσοδος: $\langle G, k \rangle$ [Στιγμιότυπο του ΑΝΕΞΑΡΤΗΤΟ ΣΥΝΟΛΟ]

Περιγραφή: Κατασκευάζουμε το στιγμιότυπο $\langle G', k' \rangle$ ως εξής

$$V' = V, E' = E, k' = |V| - k.$$

Έξοδος: $\langle G', k' \rangle$

Δοθέντος του $\langle G, k \rangle \in$ ΑΝΕΞΑΡΤΗΤΟ ΣΥΝΟΛΟ η f κατασκευάζει $\langle G', k' \rangle$ και ο χρόνος που απαιτεί είναι $O(|V'|)$, ο οποίος είναι ο χρόνος που απαιτείται ώστε να καταμετρήσει τον αριθμό των κορυφών. Επομένως, η f χρειάζεται πολυωνυμικό χρόνο στο μέγεθος της εισόδου.

Για να ολοκληρωθεί η αναγωγή θα πρέπει να δείξουμε ότι

$$\langle G, k \rangle \in \text{ΑΝΕΞΑΡΤΗΤΟ ΣΥΝΟΛΟ} \iff \langle G', k' \rangle \in \text{ΚΑΛΥΜΜΑ ΚΟΡΥΦΩΝ}. \quad (1)$$

Δηλαδή θα αποδείξουμε ότι το G έχει ένα ανεξάρτητο σύνολο V μεγέθους k αν και μόνο αν το G' έχει κάλυμμα κορυφών μεγέθους k' .

Θεωρούμε δύο σύνολο I και J τέτοια ώστε $I \cap J = \emptyset$ και $I \cup J = V = V'$, δηλαδή τα I και J διαμερίζουν τα σύνολα V και V' (όχι σε ίσα μέρη αναγκαστικά). Αν πάρουμε μια οποιαδήποτε ακμή (u, v) τότε ισχύει μια από τις περιπτώσεις

²ή απλά πολυωνυμικά.

i) $u, v \in I$, ii) $u \in I, v \in J$, iii) $u \in J, v \in I$, iv) $u, v \in J$

[Απόδειξη $\Rightarrow$ στην (1)] Έστω ότι το σύνολο I είναι ένα ανεξάρτητο σύνολο στο G , τότε αποκλείεται $u, v \in I$, περίπτωση i., εφόσον οι κορυφές που ανήκουν στο I δεν είναι γειτονικές. Στις περιπτώσεις ii. και iii. η ακμή (u, v) έχει ακριβώς μια κορυφή στο J , ενώ στην περίπτωση iv. η ακμή (u, v) έχει και τις δύο κορυφές στο J . Άρα το σύνολο J καλύπτει το G . Επομένως, αν το σύνολο I είναι ένα ανεξάρτητο σύνολο στο G , τότε σύνολο J είναι ένα κάλυμμα κορυφών στο G' , καθώς από την f έχουμε $G' = G$.

[Απόδειξη $\Leftarrow$ στην (1)] Έστω ότι το σύνολο J είναι ένα κάλυμμα κορυφών στο G' , δηλαδή στο G . Αν το σύνολο I δεν είναι ένα ανεξάρτητο σύνολο στο G τότε υπάρχει ακμή (w, r) με $w, r \in I$. Καθώς $I = V \setminus J$, η ακμή (w, r) δεν θα έχει καμία κορυφή στο J . Καταλήξαμε σε άτοπο καθώς υποθέσαμε ότι το σύνολο J είναι ένα κάλυμμα κορυφών στο G . Άρα, το σύνολο I είναι ένα ανεξάρτητο σύνολο στο G . $\square$

Άσκηση 3. Έστω η γλώσσα,

$$\text{DOUBLE-SAT} = \{ \langle \phi \rangle \mid \phi \text{ έχει το λιγότερο δύο ικανοποιήσιμες αναθέσεις} \}$$

Να δείξετε ότι $\text{DOUBLE-SAT} \in \text{NP-πλήρες}$.

Λύση. Θα αποδείξουμε τις δύο συνθήκες. Πρώτα δείχνουμε $\text{DOUBLE-SAT} \in \text{NP}$. Κατασκευάζουμε έναν μη-ντετερμινιστικό διαγνώστη N για την DOUBLE-SAT ως εξής:

$N =$ “ Στην είσοδο $\langle \phi \rangle$:

1. Μη-ντετερμινιστικά ελέγχει αν δύο boolean αναθέσεις t_1, t_2 διαφέρουν.
2. Αν τόσο η t_1 όσο και η t_2 ικανοποιούν την ϕ , αποδέχεται.
3. Αλλιώς, απορρίπτει.”

Άρα, $\text{DOUBLE-SAT} \in \text{NP}$.

Έπειτα δείχνουμε $\text{SAT} \leq_P \text{DOUBLE-SAT}$. Η συνάρτηση f απεικονίζει το $\langle \phi \rangle \in \text{SAT}$, στο $\langle \phi' \rangle \in \text{DOUBLE-SAT}$, όπου $\phi' = \phi \wedge (x_1 \vee x_2)$, και x_1, x_2 νέες μεταβλητές οι οποίες δεν υπάρχουν στον ϕ . Η αναγωγή γίνεται σε πολυωνυμικό χρόνο.

Αν ϕ μη-ικανοποιήσιμος, τότε ϕ' επίσης μη-ικανοποιήσιμος. Αν ο ϕ έχει αναθέσεις που τον καθιστούν ικανοποιήσιμο, τότε η ϕ' έχει το λιγότερο 3 αναθέσεις που τον ικανοποιούν ($\{x_1 = \text{Αληθές}, x_2 = \text{Αληθές}\}$, $\{x_1 = \text{Αληθές}, x_2 = \text{Ψευδές}\}$, $\{x_1 = \text{Ψευδές}, x_2 = \text{Αληθές}\}$). Συνεπώς, $\text{SAT} \leq_P \text{DOUBLE-SAT}$. Εφόσον αποδείξαμε και τα δύο κριτήρια, $\text{DOUBLE-SAT} \in \text{NP-πλήρες}$. $\square$

Άσκηση 4. Έστω η γλώσσα

$$A_{\text{LBA}} = \{ \langle M, w \rangle \mid M \text{ ντετερμινιστική LBA και } w \in L(M) \}$$

Δείξτε ότι $A_{\text{LBA}} \in \text{PSPACE}$.

Λύση. Έστω ντετερμινιστική LBA M (χρησιμοποιούμε επίσης τα E_L και E_R ως αριστερά και δεξιά σημάδια τερματισμού αντίστοιχα) μπορεί να προσπελάσει το πολύ $|K| \cdot |w| \cdot |\Gamma|^{|w|}$ διαφορετικές υπολογιστικές φάσεις. Επομένως για να διαγνώσουμε αν η M τερματίζει για w , προσομοιώνουμε την M για w για $|K| \cdot |w| \cdot |\Gamma|^{|w|}$ βήματα. Αν η προσομοίωση αποδεχτεί, αποδεχόμαστε. Αλλιώς, απορρίπτουμε.

Για να ανιχνεύσουμε τον αριθμό των βημάτων, χρησιμοποιούμε δυαδικό μετρητή ο οποίος παίρνει το πολύ $\log |K| + \log |w| + |w| \cdot \log |\Gamma|$ θέσεις. Άρα, $A_{LBA} \in \mathbf{PSPACE}$.

Η ανίχνευση μπορεί αν γίνει ως εξής: Για κάθε $(q, a) \mapsto (q', b, D) \in \delta$ (όπου $q \in K$, $a, b \in \Gamma$, $D \in \{R, L\}$):

- Αν $a = E_L$, τότε $b = E_L$ και $D = R$, και
- Αν $a = E_R$, τότε $b = E_R$ και $D = L$.

Επίσης, οι είσοδοι έχουν τη μορφή $E_L, w_1, \dots, w_n, E_R$ με $w_i \notin \{E_L, E_R\}$ για όλα τα $i \in \{1, \dots, n\}$. $\square$

Άσκηση 5. Ονομάζουμε ως σκάλα (ladder) μια ακολουθία από συμβολισμούς $s_1, s_2, \dots, s_k$, όπου κάθε συμβολισμός διαφέρει από την προηγούμενη κατά ακριβώς έναν χαρακτήρα. Για παράδειγμα, η παρακάτω είναι μια σκάλα από αγγλικές λέξεις, που ξεκινάει με την *head* και τελειώνει με την *free*: *head, hear, near, fear, bear, beer, deer, deed, feed, feet, fret, free*. Έστω η γλώσσα

$$\Sigma\text{KALA}_{\text{DFA}} = \left\{ \langle M, s, t \rangle \mid \begin{array}{l} M \text{ είναι DFA και υπάρχουν συμβολισμοί στο } L(M) \\ \text{που μπορούν να χρησιμοποιηθούν για τη δημιουργία} \\ \text{σκάλας που ξεκινά από το } s \text{ και καταλήγει στο } t \end{array} \right\}.$$

Να δείξετε ότι $\Sigma\text{KALA}_{\text{DFA}} \in \mathbf{PSPACE}$.

Λύση. Αρκεί να δείξουμε ότι $\Sigma\text{KALA}_{\text{DFA}} \in \mathbf{NPSPACE}$.

Έστω Σ το αλφάβητο του αυτομάτου M θα ξεκινήσουμε την επίλυση με κάποιες παρατηρήσεις. Αν $|s| \neq |t|$ τότε σαφώς δεν υπάρχει σκάλα, καθώς δεν διατηρείται το μήκος της λέξης, άρα θα πρέπει η TM να απορρίψει.

Αν $|s| = |t| = n$, θεωρούμε τον γράφο G με κορυφές όλες τις λέξεις μήκους n , δηλαδή $V = \Sigma^n$. Ως ακμές στο G θεωρούμε όλα τα ζεύγη κορυφών u, v που διαφέρουν σε ακριβώς ένα σύμβολο, δηλαδή

$$u \leftrightarrow v \iff u \text{ και } v \text{ διαφέρουν σε ακριβώς ένα σύμβολο και } u, v \in L(M).$$

Τότε αν $\langle M, s, t \rangle \in \Sigma\text{KALA}_{\text{DFA}}$, υπάρχει μονοπάτι από s σε t στον G . Ο γράφος G έχει $N = |\Sigma|^n$ πλήθος κορυφών, δηλαδή εκθετικό ως προς n . Παρ' όλα αυτά μπορούμε να επαληθεύσουμε την ύπαρξη μονοπατιού με μη-ντετερμινιστικό αλγόριθμο που χρησιμοποιεί χώρο μόνο $O(n)$.

Κατασκευάζουμε την μη-ντετερμινιστική μηχανή Turing N η οποία διαγιγνώσκει την γλώσσα $\Sigma\text{KALL}_{\text{DFA}}$.

$N =$ “ Στην είσοδο $\langle M, s, t \rangle$:

1. Ελέγχει το μήκος των s και t . Αν $|s| \neq |t|$ απορρίπτει.
2. Αλλιώς, ελέγχει αν $s \in L(M)$. Αν ναι αποδέχεται, αλλιώς απορρίπτει.
3. Καλεί την μηχανή B σε είσοδο $\langle M, s, t \rangle$.
4. Αν αποδεχθεί η B , αποδέχεται.
5. Αλλιώς, απορρίπτει.”

Η βοηθητική μηχανή B έχει την εξής λειτουργία,

$B =$ “ Στην είσοδο $\langle M, s, t \rangle$:

0. Θέτουμε $w = s$ και $c = 0$.
1. Επαναλαμβάνει.
 - i. Μη-ντετερμινιστικά επιλέγει μία θέση $i \in \{1, \dots, n\}$ και ένα σύμβολο $a \in \Sigma$ με $a \neq w[i]$.
 - ii. Κατασκευάζει την νέα συμβολοσειρά w' που είναι ίδια με την w σε όλες τις θέσεις εκτός της i , όπου $w'[i] = a$.
 - iii. Ελέγχει αν $w' \in L(M)$.
 - iv. Αν $w' \notin L(M)$, απορρίπτει αυτή την επανάληψη.
 - v. Αν $w' \in L(M)$, τότε ενημερώνει $w := w'$.
 - vi. Αν $w = t$, αποδέχεται.
 - vii. Αυξάνει τον μετρητή $c := c + 1$.
 - viii. Αν $c > |\Sigma|^n$ τότε απορρίπτει.”

Το βήμα 1i. εξασφαλίζει ότι κάθε επανάληψη αλλάζει ακριβώς ένα σύμβολο, όπως απαιτεί ο ορισμός της σκάλας. Στο βήμα 1ii., ο έλεγχος $w' \in L(M)$ γίνεται με απλή προσομοίωση του DFA πάνω στο w' . Το DFA έχει σταθερό πλήθος καταστάσεων, οπότε η προσομοίωση χρειάζεται σταθερό επιπλέον χώρο για την κάθε κατάσταση. Η B αποθηκεύει στην ταινία την w' το οποίο απαιτεί χώρο $O(n)$. Έπειτα, ο μετρητής c χρειάζεται $\lceil n \log |\Sigma| \rceil = O(n)$ bits, δηλαδή χώρο $O(n)$. Στο βήμα 1viii., το όριο $|\Sigma|^n$ εξασφαλίζει τερματισμό, επομένως οποιοδήποτε απλό μονοπάτι έχει μήκος το πολύ $|\Sigma|^n$. Αν υπάρχει μονοπάτι από s σε t , υπάρχει επίσης και απλό μονοπάτι μήκους το πολύ $|\Sigma|^n$. Ο έλεγχος του μετρητή απορρίπτει αν ξεπεραστεί αυτό το όριο, άρα η μηχανή δεν μπορεί να τρέχει επ' αόριστον.

Ορθότητα Αν υπάρχει σκάλα $s = w_1, \dots, w_k = t$ με κάθε $w_i \in L(M)$, τότε υπάρχει και μία μη-ντετερμινιστική διαδρομή της N που επιλέγει διαδοχικά τα σωστά $w_2, w_3, \dots$ και τελικά αποδέχεται. Ο μετρητής δεν ξεπερνάει το $|\Sigma|^n$ επειδή μπορούμε να αφαιρέσουμε κύκλους ώστε το μονοπάτι να έχει μήκος το πολύ $|\Sigma|^n$.

Αν η N αποδεχτεί σε κάποια μη-ντετερμινιστική διαδρομή, τότε από τον τρόπο λειτουργίας της προκύπτει μία ακολουθία συμβολοσειρών $s = w_1, w_2, \dots, w_\ell = t$ όπου κάθε διπλανή συμβολοσειρά διαφέρει σε ένα σύμβολο και κάθε $w_i \in L(M)$, επομένως υπάρχει σκάλα.

Ας αναλύσουμε τον χώρο που χρησιμοποιεί η N σε μια τυπική μη-ντετερμινιστική διαδρομή:

1. Αποθήκευση της τρέχουσας λέξης w , δηλαδή n σύμβολα και άρα $\implies O(n)$ χώρο.
2. Αποθήκευση της θέσης i και του νέου σύμβολου a , δηλαδή $O(\log n) + O(1) = O(\log n) \subseteq O(n)$.
3. Αποθήκευση του μετρητή c , δηλαδή $\lceil \log |\Sigma|^n \rceil = O(n)$ bits.
4. Χώρος για την προσομοίωση του DFA M , $O(1)$ επιπλέον χώρος, το οποίο είναι η κατάσταση του DFA.

Συνολικά ο χώρος είναι $O(n)$, δηλαδή γραμμικός στο μήκος της συμβολοσειράς s . Το μέγεθος της εισόδου $\langle M, s, t \rangle$ είναι $|M| + n + n$ (περίπου), επομένως $O(n)$ είναι πολυωνυμικό στο μέγεθος της εισόδου. Άρα η μη-ντετερμινιστική μηχανή χρησιμοποιεί πολυωνυμικό χώρο. Άρα, η γλώσσα ανήκει στην $\Sigma\text{KALL}_{\text{DFA}} \in \text{NPSPACE}$. Τέλος, από το Θεώρημα Savitch γνωρίζουμε ότι $\text{NPSPACE} = \text{PSPACE}$ και άρα καταλήγουμε στο ζητούμενο, $\Sigma\text{KALL}_{\text{DFA}} \in \text{PSPACE}$.

□

Σχόλιο. Ένας τρόπος να αποδεικνύεται ότι ένα πρόβλημα $A \in \text{NP}$ -πλήρης.

Για να αποδείξουμε ότι ένα πρόβλημα $A \in \text{NP}$ -πλήρης, πρέπει να δείξουμε ότι ανήκει στο NP και ότι είναι τουλάχιστον τόσο δύσκολο όσο κάθε άλλο πρόβλημα στο NP . Το τελευταίο βήμα γίνεται συνήθως δείχνοντας ότι $B \leq_P A$ για κάποιο πρόβλημα B που ήδη γνωρίζουμε ότι είναι NP -πλήρες. Αυτό περιγράφεται στα παρακάτω βήματα.

1. **Δείξτε ότι το $A \in \text{NP}$.** Θέλουμε να αποδείξουμε ότι υπάρχει ένας αποδοτικός επαληθευτής για το A . Με άλλα λόγια, υπάρχει επαληθευτής τέτοιος ώστε

για κάθε στοιχείο του A , υπάρχει ένα πιστοποιητικό που ο επαληθευτής θα αποδεχθεί, και για κάθε μη-στοιχείο του A , δεν υπάρχει πιστοποιητικό που θα αποδεχθεί ο επαληθευτής. Ο χρόνος εκτέλεσης του επαληθευτή (και κατά συνέπεια το μέγεθος του πιστοποιητικού) πρέπει να είναι πολυωνυμικός. Συνήθως, μια λύση στο δοθέν πρόβλημα αποτελεί ένα επαρκές πιστοποιητικό. Αυτό το βήμα είναι σύντομο, αλλά απαραίτητο.

2. **Επιλέξτε ένα γνωστό πρόβλημα NP-πλήρες.** Δηλώστε ποιο πρόβλημα B ανάγεται στο A . Πρέπει να δείξετε ότι $B \leq_P A$. Μπορείτε να χρησιμοποιήσετε οποιοδήποτε πρόβλημα B έχουμε αποδείξει στις διαλέξεις ή στα φροντιστήρια ότι είναι NP-πλήρες. Με βάση τον ορισμό της NP-πληρότητας, αν το $A \in \text{NP-πλήρες}$, τότε τεχνικά μπορείτε να χρησιμοποιήσετε οποιοδήποτε άλλο NP-πλήρες πρόβλημα B για να το δείξετε. Ωστόσο, κάποια προβλήματα είναι πολύ πιο εύκολα στη χρήση από άλλα στην απόδειξη. Συχνά είναι χρήσιμο να σκεφτείτε ποιο από τα προβλήματα που γνωρίζετε ότι είναι NP-πλήρες θα ήταν πιο φυσικό να χρησιμοποιήσετε για μια δοθείσα αναγωγή.
3. **Κατασκευάστε έναν αλγόριθμο για να λύσετε το B δεδομένου ότι υπάρχει αλγόριθμος για το A .** Πρέπει να δείξετε ότι οποιοδήποτε παράδειγμα του B μπορεί να λυθεί χρησιμοποιώντας έναν πολυωνυμικό αριθμό πράξεων και έναν πολυωνυμικό αριθμό κλήσεων σε έναν αλγόριθμο που μπορεί να λύσει το A . Σημείωση: Είναι πολύ εύκολο να μπερδευτείτε και να αποδείξετε ότι $A \leq_P B$. Προσοχή, αυτό δεν είναι αυτό που θέλετε να δείξετε (γνωρίζουμε ήδη ότι το $B \in \text{NP-πλήρες}$). Συνήθως, θα δείξετε πώς να λύσετε το B κατασκευάζοντας μία είσοδο για τον αλγόριθμο του A και ο αλγόριθμός σας θα δίνει συνήθως την ίδια απάντηση με αυτή που δίνει ο αλγόριθμος του A . (Ωστόσο, αυτό δεν ισχύει πάντα.)
4. **Αποδείξτε την ορθότητα του αλγορίθμου σας** Αυτό απαιτεί να αποδείξετε μια δήλωση *αν και μόνο αν*. Θέλετε να δείξετε ότι, δεδομένου ενός στιγμιότυπου του A , ο αλγόριθμός σας επιστρέφει **ΝΑΙ**, και θέλετε να δείξετε ότι αν ο αλγόριθμός σας επιστρέψει **ΝΑΙ**, τότε η δεδομένη είσοδος είναι πράγματι ένα ενός στιγμιότυπου του A . Είναι πάντα εύκολο να δημιουργήσετε έναν αλγόριθμο που ικανοποιεί μόνο μία από αυτές τις δύο συνθήκες. Θέλουμε να ικανοποιούνται και οι δύο.
5. **Πολυωνυμικός χρόνος και συμπέρασμα.** Τέλος, πρώτα θα πρέπει να καταλήξετε στο συμπέρασμα ότι ο αλγόριθμός σας τρέχει σε πολυωνυμικό χρόνο, $B \leq_P A$. Εφόσον το $B \in \text{NP-πλήρες}$ και εφόσον έχει αποδειχθεί ότι το $A \in \text{NP}$, το $A \in \text{NP-πλήρες}$.