ΗΥ - 280 Θεωρία Υπολογισμού

Διάλεξη 21

Κωνσταντίνος Βάρσος

Τμήμα Επιστήμης Υπολογιστών – Πανεπιστήμιο Κρήτης

Χειμερινό εξάμηνο 2025 - 26

Πολυπλοκότητα

Τι είδαμε στην Διάλεξη 20

- **NP**-πληρότητα.
- $SAT \in$ **NP**-πλήρες.
- 3-$SAT \in$ **NP**-πλήρες.
- $HAMPATH \in$ **NP**-πλήρες.

Χωρική Πολυπλοκότητα (SPACE)

- Μέχρι τώρα ασχοληθήκαμε με την διαχείριση του χρόνου.
- Τώρα θα μιλήσουμε για την πολυπλοκότητα των υπολογισμών συναρτήσει το χώρου που απαιτούν.
- Βάσει της χωρικής πολυπλοκότητας θα ταξινομήσουμε διάφορα προβλήματα.

Χωρική Πολυπλοκότητα (SPACE)

- Έστω $f : \mathbb{N} \to \mathbb{N}$. Λέμε ότι η ντετερμινιστική TM M **απαιτεί χώρο** $f(n)$ αν η ταινία της M χρησιμοποιεί **το πολύ** $f(n)$ θέσεις σε κάθε είσοδο μεγέθους n.

- Λέμε επίσης ότι η M λειτουργεί σε χώρο $f(n)$ (ή) ότι έχει χώρο εκτέλεσης $f(n)$.

- Έστω $f : \mathbb{N} \to \mathbb{R}$ οποιαδήποτε συνάρτηση. Η **χωρική πολυπλοκότητα** SPACE($f(n)$) ορίζεται ως εξής:

$$SPACE(f(n)) := \left\{ A \mid \begin{array}{l} A = L(M) \text{ η γλώσσα } A \text{ διαγιγνώσκεται από κάποια} \\ \text{ντετερμινιστική TM } M \text{ η οποία λειτουργεί σε} \\ \text{χώρο } O(f(n)) \end{array} \right\}.$$

Χωρική Πολυπλοκότητα (NSPACE)

■ Έστω $f : \mathbb{N} \to \mathbb{N}$. Λέμε ότι η μη-ντετερμινιστική TM M **απαιτεί χώρο** $f(n)$ αν η ταινία της M χρησιμοποιεί **το πολύ** $f(n)$ θέσεις σε οποιονδήποτε κλάδο του υπολογισμού, σε κάθε είσοδο μεγέθους n.

■ Έστω $f : \mathbb{N} \to \mathbb{R}$ οποιαδήποτε συνάρτηση. Η **χωρική πολυπλοκότητα** NSPACE($f(n)$) ορίζεται ως εξής:

$$NSPACE(f(n)) := \left\{ A \mid \begin{array}{l} A = L(M) \text{ η γλώσσα } A \text{ διαγιγνώσκεται από κάποια} \\ \text{μη-ντετερμινιστική TM } M \text{ η οποία λειτουργεί σε} \\ \text{χώρο } O(f(n)) \end{array} \right\}.$$

■ Ο χώρος που καταναλώνει μια ντετερμινιστική TM είναι ο αριθμός των κελιών της ταινίας εργασίας που χρησιμοποιεί.

■ Ο χώρος που καταναλώνει μια μη-ντετερμινιστική TM είναι ο μέγιστος αριθμός κελιών που χρησιμοποιεί σε όλα τα υπολογιστικά μονοπάτια.

Χωρική Πολυπλοκότητα

- $SAT \in \text{SPACE}(f(n))$.

Κατασκευάζουμε την εξής μηχανή TM M.

$M =$ "Για είσοδο $\langle \phi \rangle$:

1. Για κάθε αποτίμηση στις μεταβλητές $x_1, \ldots, x_m$ του ϕ.
2. Υπολογίζει την τιμή του ϕ για την δεδομένη αποτίμηση.
3. Εάν ο ϕ λαβει την τιμή 1, αποδέχεται.
   Αλλιώς, απορρίπτει."

Η M έχει γραμμικό χώρο εκτέλεσης. Κάθε επανάληψη του βήματος 1. μπορεί να χρησιμοποιεί ξανά το ίδιο τμήμα της τανίας. Το πλήθος των μεταβλητών είναι $m <$ από το μήκος n της εισόδου. Άρα η M χρειάζεται $O(n)$ χώρο. □

Χωρική Πολυπλοκότητα

- Έστω η γλώσσα

$$EQ_{\text{REX}} = \{\langle R_1, R_2\rangle \mid R_1, R_2 \text{ κανονικές εκφράσεις τ.ω. } L(R_1) = L(R_2)\}.$$

Να δείξετε ότι $EQ_{\text{REX}} \in \text{SPACE}(f(n))$.

Απόδειξη. Μπορούμε να κατασκευάσουμε έναν *SPACE* διαγνώστη για την EQ_{REX} ως εξής,

$Q =$ "Για είσοδο $\langle R_1, R_2\rangle$:

1. Μετατρέπουμε τις R_1, R_2 σε ισοδύναμα NFAs N_1, N_2.
2. Μετατρέπουμε N_1, N_2 σε ισοδύναμα DFAs M_1, M_2.
3. Ελαχιστοποιούμε.
4. Ελέγχουμε $EQ_{\text{DFA}}(M_1, M_2)$.
5. Εάν αποδέχεται, αποδέχεται.
   Αλλιώς, απορρίπτει."

...

Χωρική Πολυπλοκότητα

...

- Η μνήμη που χρησιμοποιείται κατά την ελαχιστοποίηση είναι ανάλογη με τον αριθμό των καταστάσεων στα αρχικά NFAs.
- Τα βήματα 2., 3. και 4. απαιτούν πολυωνυμικό χώρο.
- Άρα η Q διαγιγνώσκει χρησιμοποιώντας πολυωνυμικό χώρο, $EQ_{\mathrm{REX}} \in \mathrm{SPACE}(f(n))$.

□

Χωρική Πολυπλοκότητα

- Έστω η γλώσσα,
$$ALL_{\text{NFA}} = \{\langle M \rangle \mid M \text{ NFA} \wedge L(M) = \Sigma^*\}$$
- Να δείξτε ότι η $\overline{ALL}_{\text{NFA}}$ είναι διαγνώσιμη.

Έχουμε λοιπόν να εξετάσουμε το πρόβλημα ελέγχου σε μια μη-ντετερμινιστική TM $N = (K, \Sigma, \delta, q_0, F)$ για το $\overline{ALL}_{\text{NFA}}$.

Ιδέα. Να χρησιμοποιήσουμε τον μη-ντετερμινισμό ώστε να 'μαντέψουμε' αν μια συμβολοσειρά απορρίπτεται από το NFA.

Επίσης, θα πρέπει να έχουμε μια εκτίμηση για τον χώρο ώστε να ξέρουμε σε ποια κατάσταση είναι ο υπολογισμός.

Χωρική Πολυπλοκότητα

Κατασκευάζουμε την εξής μηχανή TM N.

$N =$ "Για είσοδο $\langle M \rangle$, όπου M NFA :

1. Τοποθετούμε ένα σημάδι στην αρχική κατάσταση M.
2. Επαναλαμβάνουμε 2^q φορές, $q = |K|$:
3. Επιλέγουμε μη-ντετερμινιστικά ένα σύμβολο εισόδου και προσομοιώνουμε την ανάγνωση του από το M.
   Μεταβάλλουμε τις θέσεις των σημαδιών πάνω στις καταστάσεις.
4. Εάν υπάρξει επανάληψη στην οποία οι καταστάσεις αποδοχής.
   δεν έχουν σημάδι, αποδέχεται. Αλλιώς, απορρίπτει."

Η N χρειάζεται εκθετικό χρόνο, αλλά πολυωνυμικό χώρο.

Άρα, $\overline{ALL}_{\text{NFA}} \in \textbf{NPSPACE}$. □

Χωρική Πολυπλοκότητα

- Για την χωρική πολυπλοκότητα, ο αλγόρθμος απαιτεί: *i*) χώρο για την αποθήκευση των θέσεων των σημαδιών, και *ii*) τον μετρητή βημάτων.

Το *i*) απαιτεί $O(q)$.

Το *ii*) απαιτεί $O(q)$.

Άρα ο N τρέχει χρησιμοποιώντας μη-ντετερμινιστικά πολυωνυμικό χώρο. Συνεπώς, $\overline{ALL}_{\text{NFA}} \in \textbf{NPSPACE}$.

Χωρική Πολυπλοκότητα

Μερικά σχόλια,

- $SPACE(f(n))$, αναφέρεται στην κλάση των προβλημάτων που μπορούν να λυθούν από υπολογιστές με πεπερασμένη μνήμη.

- $SPACE(f(n))$, ενδεχομένως τα προβλήματα σε αυτή την κλάση να χρειαστούν αρκετό χρόνο ώστε να λυθούν.

- Επειδή ένα πρόβλημα χρειάζεται γραμμικό χώρο, δεν σημαίνει ότι χρειάζεται και γραμμικό χρόνο.

Χωρική Πολυπλοκότητα

Συγκεντρωτικά, για $f : \mathbb{N} \to \mathbb{R}_{>0}$,

Γλώσσες/Προβλήματα

$$SPACE(f(n)) := \left\{ A \mid \begin{array}{l} A = L(M) \text{ η γλώσσα } A \text{ διαγιγνώσκεται από κάποια} \\ \text{ντετερμινιστική TM } M \text{ η οποία λειτουργεί σε χώρο } O(f(n)) \end{array} \right\}$$

$$NSPACE(f(n)) := \left\{ A \mid \begin{array}{l} A = L(M) \text{η γλώσσα } A \text{ διαγιγνώσκεται από κάποια} \\ \text{μη-ντετερμινιστική TM } M \text{ η οποία λειτουργεί σε χώρο } O(f(n)) \end{array} \right\}$$

Κλάσεις

$$\textbf{PSPACE} := \bigcup_k SPACE(n^k), \text{ πολυωνυμικός χώρος.}$$

$$\textbf{NPSPACE} := \bigcup_k NSPACE(n^k), \text{ μη-ντετερμινιστικά πολυωνυμικός χώρος.}$$

Θεώρημα Savitch

- Αφορά την προσομοίωση μη-ντετερμινστικών μηχανών από ντετερμινιστικές.
- Η αντίστοιχη διαδικασία στην χρονική πολυπλοκότητα επέφερε εκθετική αύξηση. [Για γλώσσα A και $A \in O(n^k)$ για N NTM, τότε $A \in O(2^{n^k})$ για M TM]

Θεώρημα. Για οποιαδήποτε συνάρτηση $f : \mathbb{N} \to \mathbb{R}_{>0}$, για την οποία ισχύει ότι $f(n) \geq \log(n)$,

$$NSPACE(f(n)) \subseteq SPACE(f^2(n)).$$

Θεώρημα Savitch

Απόδειξη. Έστω ότι η γλώσσα $L \in NSPACE(f(n))$ διαγιγνώσκεται από μία μη-ντετερμινιστική TM N.

Δεδομένης μιας εισόδου w και N, χρησιμοποιεί χώρο $f(n)$. Η κάθε φάση της N περιγράφεται από: την τρέχουσα κατάσταση, τη θέση της κεφαλής, το περιεχόμενο της ταινίας (μέχρι $f(n)$ σύμβολα). Ο συνολικός αριθμός διαφορετικών φάσεων είναι $2^{O(f(n))}$.

Εξετάζουμε τον γράφο των φάσεων της N επί της x, $G_{N,x}$.

Ορίζουμε τότε τον γράφο $G_{N,x}$ όπου:

- κάθε κορυφή είναι μία φάση της μηχανής,
- υπάρχει ακμή από C_1 σε C_2 αν η N μπορεί να μεταβεί από το C_1 στο C_2 σε ένα βήμα.

Άρα το ερώτημα *'αναγνωρίζει η N την είσοδο x;'* μετατρέπεται στο:

Υπάρχει μονοπάτι από την αρχική φάση s στην τελική φάση t στον $G_{N,x}$;

Θεώρημα Savitch

Απόδειξη. (συνέχεια) **Ιδέα**. Θέλουμε έναν ντετερμινιστικό αλγόριθμο που αποφασίζει αν υπάρχει μονοπάτι από το s στο t με μήκος το πολύ 2^i.

Παρατήρηση. Υπάρχει μονοπάτι από s σε t μήκους $\leq 2^i$ ανν υπάρχει κορυφή u στον γράφο τέτοια ώστε:

$$s \rightsquigarrow u \text{ με μήκος } \leq 2^{i-1} \qquad \text{και} \qquad u \rightsquigarrow t \text{ με μήκος } \leq 2^{i-1}.$$

Δηλαδή, 'χωρίζουμε' το πιθανό μονοπάτι στη μέση και εξετάζουμε όλους τους πιθανούς ενδιάμεσους κόμβους u.

...

Θεώρημα Savitch

Απόδειξη. (συνέχεια) Ορίζουμε τον αναδρομικό αλγόριθμο $RecReach(G, s, t, i)$ ως εξής:

$$RecReach(G, s, t, i) = \begin{cases} 1, & \text{αν υπάρχει μονοπάτι μήκους το πολύ } 2^i \text{ από το } s \text{ στο } t \\ 0, & \text{διαφορετικά.} \end{cases}$$

Περιγραφή:

1. Αν $i = 0$, επιστρέφει 1 ανν $s = t$.
2. Επιστρέφει 1 αν υπάρχει ακμή από το s στο t.
3. Επιστρέφει 1 αν υπάρχει κορυφή $u \in G$ τ.ω. $RecReach(G, s, u, i-1) = 1$ και $RecReach(G, u, t, i-1) = 1$.
4. Διαφορετικά επιστρέφει 0.

...

Θεώρημα Savitch

Απόδειξη. (συνέχεια) Έστω ότι $S(n, i)$ είναι ο χώρος που χρησιμοποιεί ο $RecReach(G, s, t, i)$, όπου ο G σε έναν γράφο με n κορυφές.
Σε κάθε επίπεδο της αναδρομής:

- αποθηκεύουμε μόνο μία κορυφή u κάθε φορά, που απαιτεί $O(\log n)$ bits,
- οι δύο αναδρομικές κλήσεις δεν χρειάζεται να συνυπάρχουν, άρα ο χώρος *δεν* διπλασιάζεται.

Άρα προκύπτει η αναδρομή:

$$S(n, i) = S(n, i-1) + O(\log n), \qquad S(n, 0) = O(1).$$

Εφαρμόζοντας την,

$$S(n, \log n) = O(\log n) + O(\log n) + \cdots + O(\log n) = (\log n)O(\log n) = O(\log^2 n).$$

[(βάθος αναδρομής) $\times$ (χώρος ανά επίπεδο)]

...

Θεώρημα Savitch

Απόδειξη. (συνέχεια)

Ο γράφος των φάσεων έχει μέγεθος $n = 2^{O(f(n))}$, άρα $\log n = O(f(n))$.

Συνεπώς, $O(\log^2 n) = O(f(n)^2)$.

Άρα κάθε γλώσσα στο $NSPACE(f(n))$ μπορεί να αναγνωριστεί ντετερμινιστικά με χώρο $f(n)^2$, όπως ακριβώς ισχυρίζεται το Θεώρημα Savitch. □

$$\boxed{NSPACE(f(n)) \subseteq SPACE(f(n)^2)}$$

Χωρική Πολυπλοκότητα

- Η **PSPACE** είναι κλειστή ως προς την ένωση, το συμπλήρωμα και την Kleene star.

Απόδειξη. [Ένωση] Έστω γλώσσες L_1, L_2 που μπορούν να διαγνωστούν σε πολυωνυμικό χώρο από τις TM M_1 και M_2 αντίστοιχα.

Κατασκευάζουμε έναν διαγνώστη M' για $L_1 \cup L_2$. Ο M' εκτελεί τον M_1 στην είσοδο. Αν ο M_1 δεχτεί, τότε δέχεται την είσοδο. Διαφορετικά, εκτελεί τον M_2. Αν ο M_2 δεχτεί, τότε αποδέχεται την είσοδο, αλλιώς την απορρίπτει.

Ανάλυση: Αυτός ο αλγόριθμος ανήκει στην **PSPACE**. Αν ο M_1 χρειάζεται το πολύ $p_1(n)$ χώρο για οποιαδήποτε είσοδο μήκους n και ο M_2 χρειάζεται το πολύ $p_2(n)$ χώρο, τότε για είσοδο μήκους n, ο M' θα χρειάζεται $p_1(n) + p_2(n)$ χώρο, καθώς ο M_1 τρέχει μια φορά και ο M_2 το πολύ μια φορά. Άρα $L_1 \cup L_2 \in$ PSPACE.

...

Χωρική Πολυπλοκότητα

Απόδειξη. [Συμπλήρωμα] Έστω γλώσσα L που μπορεί να διαγνωστεί σε πολυωνυμικό χώρο από κάποια TM M.

Κατασκευάζουμε διαγνώστη M' για $\overline{L}$ που απλά αντιστρέφει την έξοδο του M. Δηλαδή, αν ο M δεχτεί την είσοδο, ο M' απορρίπτει. Διαφορετικά, ο M' αποδέχεται.

Ανάλυση: Αυτός ο αλγόριθμος είναι σε **PSPACE**. Αν ο M χρειάζεται το πολύ $p(n)$ χώρο, τότε ο M' θα χρειαστεί $p(n) + c$ χώρο, όπου c είναι μια σταθερά για την αναστροφή του αποτελέσματος. Άρα $\overline{L} \in$ PSPACE.

...

Χωρική Πολυπλοκότητα

Απόδειξη. [Kleene star] Δεδομένης μιας γλώσσας L που μπορεί να δαγνωστεί σε πολυωνυμικό χώρο από κάποια TM M.

Κατασκευάζουμε διαγνώστη M' για L^* που επιλέγει μη-ντετερμινιστικά έναν τρόπο διαίρεσης της εισόδου σε μη κενές υποσυμβολοσειρές και εκτελεί τον M σε κάθε υποσυμβολοσειρά. Αν ο M δεχτεί όλες τις υποσυμβολοσειρές, τότε αποδέχεται. Διαφορετικά απορρίπτει.

Ανάλυση: Αυτός ο αλγόριθμος είναι σε **NPSPACE**. Αν ο M χρειάζεται το πολύ $p(n)$ χώρο για είσοδο μήκους n, τότε σε μια είσοδο μήκους n, ο M' θα χρειαστεί το πολύ $p(n)$ χώρο, επιλέγοντας τη διαίρεση με μια συμβολοσειρά μήκους n. Κάθε διαίρεση με τη μεγαλύτερη συμβολοσειρά μήκους m θα χρειαστεί $p(m)$ χώρο, καθώς μπορούμε να επαναχρησιμοποιήσουμε το χώρο αφού ο M τελειώσει με κάθε συμβολοσειρά.
Τώρα, αν $L \in$ **PSPACE**, τότε $L^* \in$ **NPSPACE**. Σύμφωνα με το θεώρημα του Savitch, **NPSPACE** $=$ **PSPACE**, άρα $L^* \in$ **PSPACE**. $\square$

Χωρική Πολυπλοκότητα

Θεώρημα. **PSPACE** = **NSPACE**.

Απόδειξη. Από Θεώρημα Savitch. □

■ Δηλαδή ο μη-ντετερμινισμός δεν προσθέτει (σχεδόν) καθόλου υπολογιστική ισχύ στις χωρικά φραγμένες TM!

Χωρική Πολυπλοκότητα

- Έστω η γλώσσα

$$\text{co-}\mathbf{NSPACE} = \{L \mid \overline{L} \in \mathbf{SPACE}\}$$

Θεώρημα. $\mathbf{PSPACE} = \text{co-}\mathbf{SPACE}$.

Απόδειξη.

Έστω μια γλώσσα $L \in \mathbf{PSPACE}$. Κατασκευάζουμε έναν διαγνώστη για την $\overline{L}$.

Πρώτα διαγιγνώσκουμε για $L \in \mathbf{PSPACE}$.

Έπειτα αντιστρέφουμε το αποτέλεσμα. Άρα διαγιγνώσκουμε για $\overline{L}$ [θα χρειαστεί πολυωνυμικό χώρο]

Δουλεύουμε ανάλογα για $L \in \text{co-}\mathbf{NSPACE}$. □

- Ίδια επιχειρηματολογία για $\mathbf{NPSPACE} = \text{co-}\mathbf{NSPACE}$.

Χρονική vs Χωρική Πολυπλοκότητα

- **P** $\subseteq$ **PSPACE**.

- 3-*SAT* $\in$ **PSPACE**.

Απόδειξη

- Απαριθμήστε όλες τις 2^n πιθανές αναθέσεις αληθείας χρησιμοποιώντας μετρητή.
- Χρειάζεται το πολύ $\log 2^n$ συν κάποια επιπλέον bits μνήμης (σταθερά).
- Ελέγξτε κάθε ανάθεση για να δείτε αν ικανοποιεί όλες τις προτάσεις.

- **NP** $\subseteq$ **PSPACE**.

Απόδειξη. Έστω αυθαίρετο πρόβλημα $Y \in$ **NP** .

Επειδή $Y \leq$ 3-SAT, υπάρχει αλγόριθμος που επιλύει το Y σε πολυωνυμικό χρόνο συν έναν πολυωνυμικό αριθμό κλήσεων στο 3-SAT. Το 3-SAT χρειάζεται πολυωνυμικό χώρο. □

Χρονική vs Χωρική Πολυπλοκότητα

Συγκεντρωτικά, για $f : \mathbb{N} \to \mathbb{R}_{>0}$,

$TIME(f(n)) \subseteq SPACE(f(n))$.

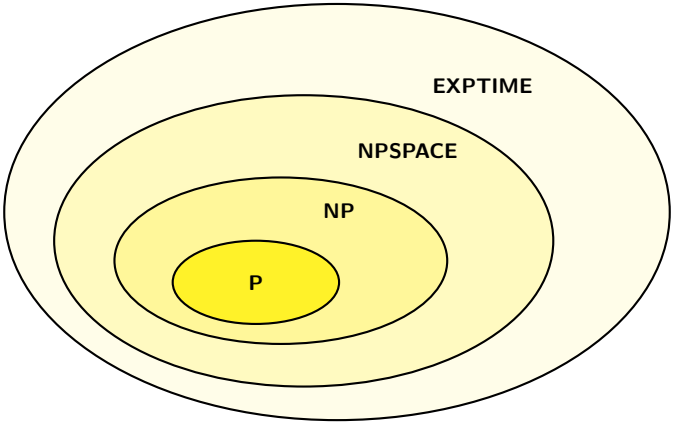
$NTIME(f(n)) \subseteq NSPACE(f(n))$.

$SPACE(f(n)) \subseteq NSPACE(f(n)) \subseteq TIME(2^{O(f(n))})$.

$NTIME(f(n)) \subseteq SPACE(f(n))$.

Χρονική vs Χωρική Πολυπλοκότητα

Και μια ιεραρχία, [επικρατούσα άποψη]



PSPACE-πληρότητα

- Μια γλώσσα A είναι **PSPACE**-πλήρης ανν
  1. $A \in$ **PSPACE**, και
  2. για κάθε $A \in$ **PSPACE** έχουμε $B \leq_P A$ (**PSPACE**-hard).

- Αν B είναι **PSPACE**-πλήρης, $B \leq_P C$, και $C \in$ **PSPACE**, τότε C είναι **PSPACE**-πλήρης.

Απόδειξη: Επειδή το $\leq_P$ είναι μεταβατικό. □

PSPACE-πληρότητα

- Έστω

$$L = \{\langle M, x, 1^n\rangle \mid M \text{ δέχεται το } x \text{ χρησιμοποιώντας το πολύ } O(n) \text{ χώρο}\}.$$

Είναι το L **PSPACE**-πλήρες;

Απόδειξη. [L **PSPACE**] Για είσοδο y της μορφής $\langle M, x, 1^n\rangle$ υπάρχει καθολική μηχανή Turing M_U η οποία προσομοιώνει την M στο x.

Καθώς η M_U χρησιμοποιεί μόνο σταθερή επιπλέον μνήμη. ισχύει:

$$y \in L \iff M_U \text{ χρησιμοποιεί } O(m) \text{ χώρο.}$$

Άρα:

$$L \in \textbf{PSPACE}.$$

...

PSPACE-πληρότητα

Απόδειξη. (συνέχεια) [$L \in$ **PSPACE**-hard] Έστω L' μια γλώσσα στην **PSPACE**. Τότε υπάρχει μηχανή Turing M που διαγιγνώσκει την L' χρησιμοποιώντας χώρο $p(n)$, όπου $p(n)$ είναι πολυωνυμική συνάρτηση.

Έστω x το ινπυτ για τη μηχανή M. Ορίζουμε τη συνάρτηση αναγωγής:

$$x \mapsto (M, x, 1^{p(|x|)}),$$

η οποία υπολογίζεται σε χρόνο $O(p(|x|))$. Από τον ορισμό του L, έχουμε:

$$(M, x, 1^{p(|x|)}) \in L \iff x \in L'.$$

Άρα το L είναι **PSPACE**-πλήρες. □