Класи складності за пам'яттю

Андрій Фесенко

Використання пам'ятті МТ

Означення

Просторовою складністю обчислень багатострічкової ДМТ M з вхідним словом $x \in \{0,1\}^*$, на якому вона зупиняється, називають кількість різних комірок, які зчитують зчитувальні пристрої на всіх робочих стрічках ДМТ M під час роботи з вхідним словом x.
Просторовою складністю обчислень багатострічкової НДМТ M на вхідному слові $x \in \{0,1\}^*$, на якому вона зупиняється для будь-якої гілки розгалуження чи будь-якої підказки, називають максимальну кількість різних комірок, які зчитують зчитувальні пристрої на всіх робочих стрічках НДМТ M під час роботи з вхідним словом x.

Використання пам'ятті МТ

Означення

Просторовою складністю обчислень багатострічкової ДМТ M з вхідним словом $x \in \{0, 1\}^*$, на якому вона зупиняється, називають кількість різних комірок, які зчитують зчитувальні пристрої на всіх робочих стрічках ДМТ M під час роботи з вхідним словом x.
Просторовою складністю обчислень багатострічкової НДМТ M на вхідному слові $x \in \{0, 1\}^*$, на якому вона зупиняється для будь-якої гілки розгалуження чи будь-якої підказки, називають максимальну кількість різних комірок, які зчитують зчитувальні пристрої на всіх робочих стрічках НДМТ M під час роботи з вхідним словом x.

- комірки можна використовувати повторно;

Використання пам'ятті МТ

Означення

Просторовою складністю обчислень багатострічкової ДМТ M з вхідним словом $x \in \{0,1\}^*$, на якому вона зупиняється, називають кількість різних комірок, які зчитують зчитувальні пристрої на всіх робочих стрічках ДМТ M під час роботи з вхідним словом x.
Просторовою складністю обчислень багатострічкової НДМТ M на вхідному слові $x \in \{0,1\}^*$, на якому вона зупиняється для будь-якої гілки розгалуження чи будь-якої підказки, називають максимальну кількість різних комірок, які зчитують зчитувальні пристрої на всіх робочих стрічках НДМТ M під час роботи з вхідним словом x.

- комірки можна використовувати повторно;
- не враховуються комірки вхідної стрічки;

Використання пам'яті МТ

Означення

Просторовою складністю обчислень багатострічкової ДМТ M з вхідним словом $x \in \{0,1\}^*$, на якому вона зупиняється, називають кількість різних комірок, які зчитують зчитувальні пристрої на всіх робочих стрічках ДМТ M під час роботи з вхідним словом x.
Просторовою складністю обчислень багатострічкової НДМТ M на вхідному слові $x \in \{0,1\}^*$, на якому вона зупиняється для будь-якої гілки розгалуження чи будь-якої підказки, називають максимальну кількість різних комірок, які зчитують зчитувальні пристрої на всіх робочих стрічках НДМТ M під час роботи з вхідним словом x.

- комірки можна використовувати повторно;
- не враховуються комірки вхідної стрічки;
- використані = зчитані комірки;

Використання пам'ятті МТ

Означення

Просторовою складністю обчислень багатострічкової ДМТ M з вхідним словом $x \in \{0,1\}^*$, на якому вона зупиняється, називають кількість різних комірок, які зчитують зчитувальні пристрої на всіх робочих стрічках ДМТ M під час роботи з вхідним словом x.
Просторовою складністю обчислень багатострічкової НДМТ M на вхідному слові $x \in \{0,1\}^*$, на якому вона зупиняється для будь-якої гілки розгалуження чи будь-якої підказки, називають максимальну кількість різних комірок, які зчитують зчитувальні пристрої на всіх робочих стрічках НДМТ M під час роботи з вхідним словом x.

- комірки можна використовувати повторно;
- не враховуються комірки вхідної стрічки;
- використані $=$ зчитані комірки;
- іноді вихідна стрічка не враховується з рухом тільки праворуч.

Обмеження кількості конфігурацій

Твердження

Якщо (Н)ДМТ M використовує пам'ять $\mathcal{O}(s(n))$, $s : \mathbb{N} \to \mathbb{N}$, то кількість різних конфігурацій (Н)ДМТ M з вхідним словом довжини n дорівнює $n 2^n (s(n))^{\mathcal{O}(1)} 2^{\mathcal{O}(s(n))}$.

Обмеження кількості конфігурацій

Твердження

Якщо (Н)ДМТ M використовує пам'ять $\mathcal{O}(s(n))$, $s : \mathbb{N} \to \mathbb{N}$, то кількість різних конфігурацій (Н)ДМТ M з вхідним словом довжини n дорівнює $n2^n(s(n))^{\mathcal{O}(1)}2^{\mathcal{O}(s(n))}$.

Доведення.

- кількість внутрішніх станів — $\mathcal{O}(1)$

$\square$

Обмеження кількості конфігурацій

Твердження

Якщо (Н)ДМТ M використовує пам'ять $\mathcal{O}(s(n))$, $s : \mathbb{N} \to \mathbb{N}$, то кількість різних конфігурацій (Н)ДМТ M з вхідним словом довжини n дорівнює $n2^n(s(n))^{\mathcal{O}(1)}2^{\mathcal{O}(s(n))}$.

Доведення.

- кількість внутрішніх станів — $\mathcal{O}(1)$
- вміст вхідної стрічки — 2^n

$\square$

Обмеження кількості конфігурацій

Твердження

Якщо (Н)ДМТ M використовує пам'ять $\mathcal{O}(s(n))$, $s : \mathbb{N} \to \mathbb{N}$, то кількість різних конфігурацій (Н)ДМТ M з вхідним словом довжини n дорівнює $n2^n(s(n))^{\mathcal{O}(1)}2^{\mathcal{O}(s(n))}$.

Доведення.

- кількість внутрішніх станів — $\mathcal{O}(1)$
- вміст вхідної стрічки — 2^n
- позиція на вхідній стрічці — n

$\square$

Обмеження кількості конфігурацій

Твердження

Якщо (Н)ДМТ M використовує пам'ять $\mathcal{O}(s(n))$, $s : \mathbb{N} \to \mathbb{N}$, то кількість різних конфігурацій (Н)ДМТ M з вхідним словом довжини n дорівнює $n2^n(s(n))^{\mathcal{O}(1)}2^{\mathcal{O}(s(n))}$.

Доведення.

- кількість внутрішніх станів — $\mathcal{O}(1)$
- вміст вхідної стрічки — 2^n
- позиція на вхідній стрічці — n
- вміст робочих стрічок — $(\mathcal{O}(1))^{s(n)}$

$\square$

Обмеження кількості конфігурацій

Твердження

Якщо (Н)ДМТ M використовує пам'ять $\mathcal{O}(s(n))$, $s : \mathbb{N} \to \mathbb{N}$, то кількість різних конфігурацій (Н)ДМТ M з вхідним словом довжини n дорівнює $n2^n(s(n))^{\mathcal{O}(1)}2^{\mathcal{O}(s(n))}$.

Доведення.

- кількість внутрішніх станів — $\mathcal{O}(1)$
- вміст вхідної стрічки — 2^n
- позиція на вхідній стрічці — n
- вміст робочих стрічок — $(\mathcal{O}(1))^{s(n)}$
- позиції на робочих стрічках — $(s(n))^{\mathcal{O}(1)}$

$\Rightarrow n2^n(s(n))^{\mathcal{O}(1)}2^{\mathcal{O}(s(n))}$ $\square$

Обмеження кількості конфігурацій

Твердження

Якщо (Н)ДМТ M використовує пам'ять $\mathcal{O}(s(n))$, $s : \mathbb{N} \to \mathbb{N}$, то кількість різних конфігурацій (Н)ДМТ M з вхідним словом довжини n дорівнює $n2^n(s(n))^{\mathcal{O}(1)}2^{\mathcal{O}(s(n))}$.

Доведення.

- кількість внутрішніх станів — $\mathcal{O}(1)$
- вміст вхідної стрічки — 2^n
- позиція на вхідній стрічці — n
- вміст робочих стрічок — $(\mathcal{O}(1))^{s(n)}$
- позиції на робочих стрічках — $(s(n))^{\mathcal{O}(1)}$

$\Rightarrow n2^n(s(n))^{\mathcal{O}(1)}2^{\mathcal{O}(s(n))}$ □

Наслідок

(Н)ДМТ M використовує пам'ять $\mathcal{O}(s(n))$ — зупиняється

Класи складності *DSPACE* та *NSPACE*

Означення

Для довільної функції $s : \mathbb{N} \to \mathbb{N}$ класом складності $DSPACE(s(n))$ (або $SPACE(s(n))$) є множина мов $\{L_1 \subseteq \{0,1\}^* \mid$ існує ДМТ M, яка розв'язує мову L_1, використовуючи пам'ять $O(s(n))\}$. Для довільної функції $s : \mathbb{N} \to \mathbb{N}$ класом складності $NSPACE(s(n))$ є множина мов $\{L_1 \subseteq \{0,1\}^* \mid$ існує НДМТ M, яка розв'язує мову L_1, використовуючи пам'ять $O(s(n))\}$.

Класи складності *DSPACE* та *NSPACE*

Твердження

Нехай $s : \mathbb{N} \to \mathbb{N}$ є довільною конструктивною за пам'яттю функцією. Тоді
$DTIME(s(n)) \subseteq DSPACE(s(n)) \subseteq NSPACE(s(n)) \subseteq DTIME(2^{O(S(n))})$.

Доведення.

$DTIME(s(n)) \subseteq DSPACE(s(n))$ — кількість тактів завжди є меншою
$DSPACE(s(n)) \subseteq NSPACE(s(n))$ — детермінована машина завжди є частковим випадком недетермінованої
$NSPACE(s(n)) \subseteq DTIME(2^{\mathcal{O}(S(n))})$ — перебір підказки та обмеження конфігурацій $\square$

Класи складності за пам'яттю

Означення

$$PSPACE = \bigcup_{k\in\mathbb{N}} DSPACE(n^k)$$
$$NPSPACE = \bigcup_{k\in\mathbb{N}} NSPACE(n^k)$$
$$EXPSPACE = \bigcup_{k\in\mathbb{N}} DSPACE(2^{n^k})$$
$$NEXPSPACE = \bigcup_{k\in\mathbb{N}} NSPACE(2^{n^k})$$
$$L = DSPACE(\log n)$$
$$L^2 = DSPACE((\log n)^2)$$
$$NL = NSPACE(\log n)$$
$$REG = SPACE(\mathcal{O}(1)) = NSPACE(\mathcal{O}(1))$$
$$POLYLOGSPACE = \bigcup_{k\in\mathbb{N}, k\geq 1} DSPACE(\log n^k)$$
$$DLBA = \bigcup_{k\in\mathbb{N}, k\geq 1} DSPACE(kn)$$
$$LBA = \bigcup_{k\in\mathbb{N}, k\geq 1} NSPACE(kn)$$

Класи складності за пам'яттю

Наслідок

$REG \subseteq L \subseteq L^2 \subseteq POLYLOGSPACE \subseteq DLBA \subseteq LBA$.
$L \subseteq NL \subseteq P \subseteq PSPACE \subseteq NPSPACE \subseteq EXP \subseteq EXPSPACE \subseteq NEXPSPACE$

Мова *PATH*

Означення

Мову $PATH \subseteq \{0,1\}^*$ визначають як $PATH = \{(G, s, t) \mid$ існує шлях з вершини s у вершину t орієнтованого графу $G\}$.

Мова *PATH*

Твердження

$PATH \in L^2$.

Доведення.

- $PATH\ LIM = \{(G, s, t, k) \mid$ існує шлях довжини не більше k з вершини s у вершину t орієнтованого графу $G\}$
- $(G, s, t) \in PATH \Leftrightarrow (G, s, t, n-1) \in PATH\ LIM$
- $(G, s, t, 0) \in PATH\ LIM \Leftrightarrow s = t$
- $(G, s, t, 1) \in PATH\ LIM \Leftrightarrow s = t$ або $(s, t) \in E$
- $(G, s, t, k) \in PATH\ LIM \Leftrightarrow$ існує така вершина u графу G, що $\left(G, s, u, \left\lfloor \frac{k}{2} \right\rfloor\right) \in PATH\ LIM$ і $\left(G, u, t, \left\lceil \frac{k}{2} \right\rceil\right) \in PATH\ LIM$
- перебрати всі вершини чи не є вони серединою (рекурсивний пошук в глибину)
- глибина рекурсії — $\mathcal{O}(\log n)$
- збереження локальних параметрів — $\mathcal{O}(\log n)$ комірок пам'яті.

$\square$

Теорема Севіча

Walter J. Savitch "Relationships between nondeterministic and deterministic tape complexities— Journal of Computer and System Sciences, 4 (2): 177–192, 1970

Теорема Севіча

Для довільної конструктивної за пам'яттю функції $s : \mathbb{N} \to \mathbb{N}$, $s(n) \geq \log n$, $NSPACE(s(n)) \subseteq DSPACE\left((s(n))^2\right)$.

Теорема Севіча

Walter J. Savitch "Relationships between nondeterministic and deterministic tape complexities— Journal of Computer and System Sciences, 4 (2): 177–192, 1970

Теорема Севіча

Для довільної конструктивної за пам'яттю функції $s : \mathbb{N} \to \mathbb{N}$, $s(n) \geq \log n$, $NSPACE(s(n)) \subseteq DSPACE\left((s(n))^2\right)$.

Доведення.

$\forall x \in \{0, 1\}^*$, $|x| = n$, і НДМТ M граф конфігурацій містить $n(s(n))^{\mathcal{O}(1)} 2^{\mathcal{O}(s(n))}$ вершин

□

Теорема Севіча

Walter J. Savitch "Relationships between nondeterministic and deterministic tape complexities— Journal of Computer and System Sciences, 4 (2): 177–192, 1970

Теорема Севіча

Для довільної конструктивної за пам'яттю функції $s : \mathbb{N} \to \mathbb{N}$, $s(n) \geq \log n$, $NSPACE(s(n)) \subseteq DSPACE\left((s(n))^2\right)$.

Доведення.

$\forall x \in \{0, 1\}^*$, $|x| = n$, і НДМТ M граф конфігурацій містить $n(s(n))^{\mathcal{O}(1)} 2^{\mathcal{O}(s(n))}$ вершин
для такого графу $PATH$ розв'язується ДМТ з $\mathcal{O}((s(n))^2)$ пам'яті

$\square$

Теорема Севіча

Walter J. Savitch "Relationships between nondeterministic and deterministic tape complexities— Journal of Computer and System Sciences, 4 (2): 177–192, 1970

Теорема Севіча

Для довільної конструктивної за пам'яттю функції $s : \mathbb{N} \to \mathbb{N}$, $s(n) \geq \log n$, $NSPACE(s(n)) \subseteq DSPACE\left((s(n))^2\right)$.

Доведення.

$\forall x \in \{0,1\}^*$, $|x| = n$, і НДМТ M граф конфігурацій містить $n(s(n))^{\mathcal{O}(1)} 2^{\mathcal{O}(s(n))}$ вершин

для такого графу *PATH* розв'язується ДМТ з $\mathcal{O}((s(n))^2)$ пам'яті

одна початкова конфігурація і не більше $n(s(n))^{\mathcal{O}(1)} 2^{\mathcal{O}(s(n))}$ заключних з q_{acc}

$\square$

Теорема Севіча

Walter J. Savitch "Relationships between nondeterministic and deterministic tape complexities— Journal of Computer and System Sciences, 4 (2): 177–192, 1970

Теорема Севіча

Для довільної конструктивної за пам'яттю функції $s : \mathbb{N} \to \mathbb{N}$, $s(n) \geq \log n$, $NSPACE(s(n)) \subseteq DSPACE\left((s(n))^2\right)$.

Доведення.

$\forall x \in \{0,1\}^*$, $|x| = n$, і НДМТ M граф конфігурацій містить $n(s(n))^{\mathcal{O}(1)} 2^{\mathcal{O}(s(n))}$ вершин

для такого графу $PATH$ розв'язується ДМТ з $\mathcal{O}((s(n))^2)$ пам'яті

одна початкова конфігурація і не більше $n(s(n))^{\mathcal{O}(1)} 2^{\mathcal{O}(s(n))}$ заключних з q_{acc}

перебираємо всі $n(s(n))^{\mathcal{O}(1)} 2^{\mathcal{O}(s(n))}$ екземплярів задачі $PATH$ □

Теорема Севіча (наслідки)

Наслідок

1. $PSPACE = NPSPACE$.
2. $EXPSPACE = NEXPSPACE$.
3. $NP \subseteq PSPACE$.
4. $NL \subseteq L^2$.

Клас складності *PSPACE*

Клас складності *PSPACE* є замкненим відносно операцій

- об'єднання мов
- конкатенації мов
- доповнення мови
- замикання Кліні

Клас складності *PSPACE* є замкненим відносно поліноміального зведення
$L_1 \leq_p L_2$, $L_2 \in PSPACE \Rightarrow L_1 \in PSPACE$

PSPACE-повні мови

$SPACETM = \{(M, x, 1^n) \mid$ ДМТ M приймає слово x, використовуючи n комірок пам'яті $\}$

PSPACE-повні мови

$SPACETM = \{(M, x, 1^n) \mid$ ДМТ M приймає слово x, використовуючи n комірок пам'яті $\}$

Твердження

SPACETM є *PSPACE*-повною мовою

PSPACE-повні мови

$SPACETM = \{(M, x, 1^n) \mid$ ДМТ M приймає слово x, використовуючи n комірок пам'яті $\}$

Твердження

$SPACETM$ є $PSPACE$-повною мовою

Булева формула з кванторами
$Q_1 x_1 \dots Q_n x_n \; \varphi(x_1, \dots, x_n)$, $Q_i \in \{\forall, \exists\}$.

$PSPACE$-повні мови

$SPACETM = \{(M, x, 1^n) \mid$ ДМТ M приймає слово x, використовуючи n комірок пам'яті $\}$

Твердження

$SPACETM$ є $PSPACE$-повною мовою

Булева формула з кванторами
$Q_1 x_1 \dots Q_n x_n \ \varphi(x_1, \dots, x_n)$, $Q_i \in \{\forall, \exists\}$.

Приклади

1. $\forall x \exists y (x \wedge y) \vee (\neg x \wedge \neg y)$
2. $\forall x \forall y (x \wedge y) \vee (\neg x \wedge \neg y)$
3. SAT

PSPACE-повні мови

$SPACETM = \{(M, x, 1^n) \mid$ ДМТ M приймає слово x, використовуючи n комірок пам'яті $\}$

Твердження

SPACETM є *PSPACE*-повною мовою

Булева формула з кванторами
$Q_1 x_1 \dots Q_n x_n\ \varphi(x_1, \dots, x_n)$, $Q_i \in \{\forall, \exists\}$.

Приклади

1. $\forall x \exists y (x \wedge y) \vee (\neg x \wedge \neg y)$
2. $\forall x \forall y (x \wedge y) \vee (\neg x \wedge \neg y)$
3. *SAT*

Мова *TQBF* — всі істинні булеві формули з кванторами

PSPACE-повні мови

Твердження

TQBF є *PSPACE*-повною мовою

PSPACE-повні мови

Твердження

TQBF є *PSPACE*-повною мовою

QBF гра є *PSPACE*-повною мовою

PSPACE-повні мови

Твердження

TQBF є *PSPACE*-повною мовою

QBF гра є *PSPACE*-повною мовою

сертифікат в *NP* = виграшна стратегія в грі для 2 гравців з повною інформацією (шахи, го, навколишнє середовище і тд.)

Клас складності *NL*

Клас складності *NL* є замкненим відносно операцій

- об'єднання мов
- перетину мов
- конкатенації мов
- замикання Кліні

Клас складності NL

Клас складності NL є замкненим відносно операцій

- об'єднання мов
- перетину мов
- конкатенації мов
- замикання Кліні

Поліноміальне зведення не має сенсу використовувати — $NL \subseteq P$, L *vs* NL

Клас складності *NL*

Означення

Довільну функцію $f : \{0,1\}^* \to \{0,1\}^*$ називають *поліноміально обмеженою*, якщо існує така константа $c \in \mathbb{N}$, що для довільного значення $x \in \{0,1\}^*$ виконується нерівність $|f(x)| < |x|^c$.
Довільну поліноміально обмежену функцію $f : \{0,1\}^* \to \{0,1\}^*$ називають *(неявно) обчислюваною з використанням логарифмічно обмеженої пам'яті*, якщо мови $L_{f,b} = \{(x,i) \mid f(x)_i = 1\}$ і $L_{f,l} = \{(x,i) \mid i \leq |f(x)|\}$ належать класу складності L.

Клас складності NL

Означення

Довільну функцію $f : \{0,1\}^* \to \{0,1\}^*$ називають *поліноміально обмеженою*, якщо існує така константа $c \in \mathbb{N}$, що для довільного значення $x \in \{0,1\}^*$ виконується нерівність $|f(x)| < |x|^c$.
Довільну поліноміально обмежену функцію $f : \{0,1\}^* \to \{0,1\}^*$ називають *(неявно) обчислюваною з використанням логарифмічно обмеженої пам'яті*, якщо мови $L_{f,b} = \{(x,i) \mid f(x)_i = 1\}$ і $L_{f,l} = \{(x,i) \mid i \leq |f(x)|\}$ належать класу складності L.

Неформально

ДМТ з $\mathcal{O}(\log n)$ пам'яттю обчислює будь-який біт результату

Клас складності NL

Означення

Довільну функцію $f : \{0,1\}^* \to \{0,1\}^*$ називають *поліноміально обмеженою*, якщо існує така константа $c \in \mathbb{N}$, що для довільного значення $x \in \{0,1\}^*$ виконується нерівність $|f(x)| < |x|^c$.
Довільну поліноміально обмежену функцію $f : \{0,1\}^* \to \{0,1\}^*$ називають *(неявно) обчислюваною з використанням логарифмічно обмеженої пам'яті*, якщо мови $L_{f,b} = \{(x,i) \mid f(x)_i = 1\}$ і $L_{f,l} = \{(x,i) \mid i \leq |f(x)|\}$ належать класу складності L.

Неформально

ДМТ з $\mathcal{O}(\log n)$ пам'яттю обчислює будь-який біт результату

Твердження

Множина обчислюваних з використанням логарифмічно обмеженої пам'яті функцій є замкненою відносно операції композиції функцій.

Зведенням з використанням логарифмічно обмеженої пам'яті

Означення

Зведенням з використанням логарифмічно обмеженої пам'яті (logspace reduction) називають зведення функціонального типу відносно множини обчислюваних з використанням логарифмічно обмеженої пам'яті функцій. $L_1 \leq_l L_2$ $\forall x \in \{0,1\}^*$ $x \in L_1 \Leftrightarrow f(x) \in L_2$.

Зведенням з використанням логарифмічно обмеженої пам'яті

Означення

Зведенням з використанням логарифмічно обмеженої пам'яті (logspace reduction) називають зведення функціонального типу відносно множини обчислюваних з використанням логарифмічно обмеженої пам'яті функцій. $L_1 \leq_l L_2$ $\forall x \in \{0,1\}^*$ $x \in L_1 \Leftrightarrow f(x) \in L_2$.

1. класи L, NL, P, NP, $PSPACE$ $EXPTIME$ є замкненими відносно зведення $\leq_l$
2. довільна нетривіальна мова з класу L є повною відносно зведення $\leq_l$
3. $L_1 \leq_l L_2 \Rightarrow L_1 \leq_p L_2$ ($\leq_l$ є сильнішим)

Клас складності *NL*

Мови *PATH* та 2-*SAT* належать класу складності *NL*.

Доведення.

Для мови *PATH* НДМТ може недетерміновано вгадувати наступну вершину суміжну з поточною вершиною у шляху,обмежуючи довжину шляху кількістю вершин у графі або, якщо досягли вершину t.

Необхідно зберігати лише довжину поточного шляху та поточну вершину, куди вже дійшли. $\mathcal{O}(\log n)$ $\square$

Клас складності NL

Мови $PATH$ та 2-SAT належать класу складності NL.

Доведення.

Для мови $PATH$ НДМТ може недетерміновано вгадувати наступну вершину суміжну з поточною вершиною у шляху,обмежуючи довжину шляху кількістю вершин у графі або, якщо досягли вершину t.

Необхідно зберігати лише довжину поточного шляху та поточну вершину, куди вже дійшли. $\mathcal{O}(\log n)$ $\square$

$PATH$ є NL повною відносно зведення $\leq_l$

Сертифікати в класі NL

Означення

Мова L_1 належить класу складності NL, якщо існує ДМТ M з додатковою стрічкою, яка є доступною тільки для одноразового зчитування, і поліном $p : \mathbb{N} \to \mathbb{N}$ такі, що для довільного слова $x \in \{0, 1\}^*$ виконується співвідношення $x \in L_1 \Leftrightarrow \exists u \in \{0, 1\}^{p(|x|)}$ таке, що $M(x, u) = 1$, при цьому, при обчисленні $M(x, u)$, слово x записується на вхідній стрічці, яка є доступною тільки для зчитування; слово u записується на додатковій стрічці, яка є доступною тільки для одноразового зчитування, а ДМТ M використовує при обчисленні не більше $\mathcal{O}(\log |x|)$ комірок робочих стрічок для всіх слів x.

Сертифікати в класі NL

Означення

Мова L_1 належить класу складності NL, якщо існує ДМТ M з додатковою стрічкою, яка є доступною тільки для одноразового зчитування, і поліном $p : \mathbb{N} \to \mathbb{N}$ такі, що для довільного слова $x \in \{0,1\}^*$ виконується співвідношення $x \in L_1 \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}$ таке, що $M(x,u) = 1$, при цьому, при обчисленні $M(x,u)$, слово x записується на вхідній стрічці, яка є доступною тільки для зчитування; слово u записується на додатковій стрічці, яка є доступною тільки для одноразового зчитування, а ДМТ M використовує при обчисленні не більше $\mathcal{O}(\log|x|)$ комірок робочих стрічок для всіх слів x.

- $\Rightarrow$ сертифікатом є послідовність вибору

Сертифікати в класі NL

Означення

Мова L_1 належить класу складності NL, якщо існує ДМТ M з додатковою стрічкою, яка є доступною тільки для одноразового зчитування, і поліном $p : \mathbb{N} \to \mathbb{N}$ такі, що для довільного слова $x \in \{0,1\}^*$ виконується співвідношення $x \in L_1 \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}$ таке, що $M(x,u) = 1$, при цьому, при обчисленні $M(x,u)$, слово x записується на вхідній стрічці, яка є доступною тільки для зчитування; слово u записується на додатковій стрічці, яка є доступною тільки для одноразового зчитування, а ДМТ M використовує при обчисленні не більше $\mathcal{O}(\log|x|)$ комірок робочих стрічок для всіх слів x.

- $\Rightarrow$ сертифікатом є послідовність вибору
- $\Leftarrow$ сертифікат обирає функцію переходів

Сертифікати в класі NL

Означення

Мова L_1 належить класу складності NL, якщо існує ДМТ M з додатковою стрічкою, яка є доступною тільки для одноразового зчитування, і поліном $p : \mathbb{N} \to \mathbb{N}$ такі, що для довільного слова $x \in \{0,1\}^*$ виконується співвідношення $x \in L_1 \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}$ таке, що $M(x,u) = 1$, при цьому, при обчисленні $M(x,u)$, слово x записується на вхідній стрічці, яка є доступною тільки для зчитування; слово u записується на додатковій стрічці, яка є доступною тільки для одноразового зчитування, а ДМТ M використовує при обчисленні не більше $\mathcal{O}(\log |x|)$ комірок робочих стрічок для всіх слів x.

- $\Rightarrow$ сертифікатом є послідовність вибору
- $\Leftarrow$ сертифікат обирає функцію переходів
- без одноразового зчитування буде сертифікат для класу NP

Теорема Іммермана-Селепчені

для $PATH$ і $(G = (V, E), s, t)$ сертифікатом є послідовність вершин $v_1, \ldots, v_n \in V$, $n < |V|$, така, що
$v_1 = s$, $v_n = t$ і $(v_i, v_{i+1}) \in E$, $i \in \{1, \ldots, n-1\}$
розмір сертифікату $n \log n$

Теорема Іммермана-Селепчені

для $PATH$ і $(G = (V, E), s, t)$ сертифікатом є послідовність вершин
$v_1, \ldots, v_n \in V$, $n < |V|$, така, що
$v_1 = s$, $v_n = t$ і $(v_i, v_{i+1}) \in E$, $i \in \{1, \ldots, n-1\}$
розмір сертифікату $n \log n$

Теорема Іммермана-Селепчені, (Immerman-Szlepcsenyi), 1987

$\overline{PATH} \in NL$

Теорема Іммермана-Селепчені

Доведення.

- C_i — множина вершин графу G, досяжних з вершини s не більш ніж за i ребер

$\square$

Теорема Іммермана-Селепчені

Доведення.

- C_i — множина вершин графу G, досяжних з вершини s не більш ніж за i ребер
- $\forall i \in \{1, .., n\}$ і $\forall v \in V$ існує сертифікат, що $v \in C_i$: $u = v_0, v_1, \ldots, v_k$, де $v_0 = s$, $(v_i, v_{i+1}) \in E$, $v_k = v$, $k \leq i$

$\square$

Теорема Іммермана-Селепчені

Доведення.

- C_i — множина вершин графу G, досяжних з вершини s не більш ніж за i ребер
- $\forall i \in \{1, .., n\}$ і $\forall v \in V$ існує сертифікат, що $v \in C_i$: $u = v_0, v_1, \ldots, v_k$, де $v_0 = s$, $(v_i, v_{i+1}) \in E$, $v_k = v$, $k \leq i$
- дві додаткові процедури:
  - перевірити, що вершина v не належить множині C_i за значенням $|C_i|$

$\square$

Теорема Іммермана-Селепчені

Доведення.

- C_i — множина вершин графу G, досяжних з вершини s не більш ніж за i ребер
- $\forall i \in \{1, .., n\}$ і $\forall v \in V$ існує сертифікат, що $v \in C_i$: $u = v_0, v_1, \ldots, v_k$, де $v_0 = s$, $(v_i, v_{i+1}) \in E$, $v_k = v$, $k \leq i$
- дві додаткові процедури:
  - перевірити, що вершина v не належить множині C_i за значенням $|C_i|$
  - перевірити, що $|C_i| = c$ за значенням $|C_{i-1}|$

$\square$

Теорема Іммермана-Селепчені

Доведення.

- C_i — множина вершин графу G, досяжних з вершини s не більш ніж за i ребер
- $\forall i \in \{1, .., n\}$ і $\forall v \in V$ існує сертифікат, що $v \in C_i$:
  $u = v_0, v_1, \ldots, v_k$, де $v_0 = s$, $(v_i, v_{i+1}) \in E$, $v_k = v$, $k \le i$
- дві додаткові процедури:
  - перевірити, що вершина v не належить множині C_i за значенням $|C_i|$
  - перевірити, що $|C_i| = c$ за значенням $|C_{i-1}|$
- $C_0 = \{s\}$, C_n — всі досяжні вершини

$\square$

Теорема Іммермана-Селепчені

Доведення.

- C_i — множина вершин графу G, досяжних з вершини s не більш ніж за i ребер
- $\forall i \in \{1, .., n\}$ і $\forall v \in V$ існує сертифікат, що $v \in C_i$:
  $u = v_0, v_1, \ldots, v_k$, де $v_0 = s$, $(v_i, v_{i+1}) \in E$, $v_k = v$, $k \leq i$
- дві додаткові процедури:
  - перевірити, що вершина v не належить множині C_i за значенням $|C_i|$
  - перевірити, що $|C_i| = c$ за значенням $|C_{i-1}|$
- $C_0 = \{s\}$, C_n — всі досяжні вершини
- $\Rightarrow$ друга процедура до C_n, а потім перша відносно $|C_n|$ і t

□

Теорема Іммермана-Селепчені

Доведення.

1) вершина v не належить множині C_i за значенням $|C_i|$
Сертифікат — сертифікати всіх вершин з множини C_i, впорядковані за зростанням
Перевірка:

- коректність кожного сертифікату
- зростання нумерації
- немає сертифіката для вершини v
- загальна кількість сертифікатів дорівнює $|C_i|$

$\square$

Теорема Іммермана-Селепчені

Доведення.

1) вершина v не належить множині C_i за значенням $|C_i|$

Сертифікат — сертифікати всіх вершин з множини C_i, впорядковані за зростанням

Перевірка:

- коректність кожного сертифікату
- зростання нумерації
- немає сертифіката для вершини v
- загальна кількість сертифікатів дорівнює $|C_i|$

Розмір сертифікату — $n^2 \log n$

Перевірка — $\mathcal{O}(\log n)$ □

Теорема Іммермана-Селепчені

Доведення.

1) вершина v не належить множині C_i за значенням $|C_i|$

Сертифікат — сертифікати всіх вершин з множини C_i, впорядковані за зростанням

Перевірка:

- коректність кожного сертифікату
- зростання нумерації
- немає сертифіката для вершини v
- загальна кількість сертифікатів дорівнює $|C_i|$

Розмір сертифікату — $n^2 \log n$

Перевірка — $\mathcal{O}(\log n)$

вершина v не належить множині C_i за значенням $|C_{i-1}|$ $\square$

Теорема Іммермана-Селепчені

Доведення.

1) вершина v не належить множині C_i за значенням $|C_i|$
Сертифікат — сертифікати всіх вершин з множини C_i, впорядковані за зростанням
Перевірка:

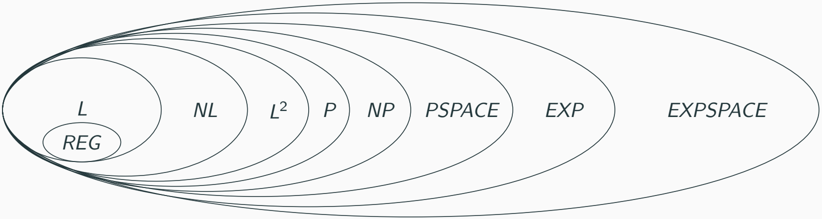
- коректність кожного сертифікату
- зростання нумерації
- немає сертифіката для вершини v
- загальна кількість сертифікатів дорівнює $|C_i|$

Розмір сертифікату — $n^2 \log n$
Перевірка — $\mathcal{O}(\log n)$
вершина v не належить множині C_i за значенням $|C_{i-1}|$
2) $|C_i| = c$ за значенням $|C_{i-1}|$ – відповідні сертифікати для кожної вершини в порядку зростання □

Наслідок

- $\forall s : \mathbb{N} \to \mathbb{N},\ s(n) \geq \log n,\ NSPACE(s(n)) = coNSPACE(s(n))$ (метод доповнення)
- $NL = coNL$

Діаграма класів складності



Відкриті питання

1. $NL \subseteq P$, але чи $NL = P$?
2. $L \subseteq NL$, але чи $L = NL$?
3. $NL \subsetneq PSPACE$, але чи $P = PSPACE$?