Листок 4

Пространственная сложность

Разумная модель вычислений для определения классов сложности по памяти – многоленточная машина Тьюринга. Мы будем без ограничения общности рассматривать трёхленточную МТ, на одной из лент написан вход машины – с этой ленты МТ может только читать, на одной из лент будет написан выход машины – один бит для распознавателей или больше для МТ, вычисляющих некоторые функции, – на эту ленту МТ может только писать, **в частности на этой ленте движение головки происходит только вправо**. Третья лента (или же все оставшиеся ленты) рабочая. Именно число ячеек на рабочей ленте, используемое МТ во время работы, есть сложность МТ по памяти на конкретном входе.

Естественным образом определяются классы $\mathrm{DSPACE}(f)$ и $\mathrm{NSPACE}(f)$.

Полиномиальная память

По аналогии с полиномиальными классами $\mathcal{P}$ и $\mathcal{NP}$ вводятся $\mathcal{PSPACE} = \bigcup\limits_{i=0}^{\infty} \mathrm{DSPACE}(O(n^i))$ и $\mathcal{NPSPACE} = \bigcup\limits_{i=0}^{\infty} \mathrm{NSPACE}(O(n^i))$.

Логарифмическая память

Одними из особенно важных пространственных классов языков являются замкнутые относительно стандартных операций подмножества $\mathcal{P}$ – множества языков, распознаваемые на логарифмической памяти.

Используемые нами классы: $\mathcal{L} = \mathrm{DSPACE}(O(\log n))$, $\mathcal{NL} = \mathrm{NSPACE}(O(\log n))$

Как и класс $\mathcal{NP}$ класс $\mathcal{NL}$ допускает альтернативное определение в терминах сертификатов и верификаторов.

$L \in \mathcal{NL}$ тогда и только тогда, когда

для некоторой ДМТ M выполнено $x \in L \Longleftrightarrow \exists y M(x,y) = 1$. При этом

– M логарифмическая по памяти,

– длина сертификата y должна быть полиномиальна от длины x,

– машина M получает y на отдельной ленте, по которой может двигаться только слева направо (т.е. сертификат читается только один раз).

По аналогии с $\mathcal{NP}$-полными определяются $\mathcal{NL}$-полные языки. Полиномиальная сводимость для такого класса не подходит, нужна более тонкая, логарифмическая по памяти сводимость.

Формально $L_1 \leqslant_{\log} L_2$ если:

– существует функция f, вычислимая некоторой МТ за логарифмическую память: т.е. на входе $y \in \{0,1\}^*$ МТ останавливается в состоянии q_{halt}, на выходной ленте после останова написано $f(y)$, запись на выходную ленту ведётся слева направо, выходная лента не читается, на рабочей ленте используется $O(\log |y|)$ ячеек;

– $x \in L_1$ тогда и только тогда, когда $f(x) \in L_2$.

Язык L называется $\mathcal{NL}$-полным, если

– $L \in \mathcal{NL}$,

– $\forall L' \in \mathcal{NL} \quad L' \leqslant_{\log} L$.

Основные теоремы пространственной сложности

Сложность по памяти существенно отличается от сложности по времени благодаря тому, что память можно использовать повторно. Следующие теоремы показывают, что сложность по памяти в некотором смысле устроена “проще”, чем сложность по времени: аналогичные гипотезы для временной сложности не доказаны и, более того, предполагаются неверными.

Хорошей функцией мы назовём конструируемую по памяти s такую, что $s(n) = \Omega(\log n)$

Savitch theorem

Для любой хорошей s выполняется

$$\mathrm{NSPACE}\,(s) \subset \mathrm{DSPACE}\left(s^2\right)$$

Immerman–Szelepcsényi theorem

Для любой хорошей s выполняется

$$\mathrm{NSPACE}\,(s) = \mathrm{co}\,\mathrm{NSPACE}\,(s)$$

Из этих теорем следует, в том числе, что

$$\mathcal{NL} = \mathrm{co}\mathcal{NL}, \quad \mathcal{PSPACE} = \mathcal{NPSPACE} = \mathrm{co}\mathcal{NPSPACE}$$

Кроме того выполняется следующее.

Пусть $\mathrm{stCON} = (G, s, t)$ – язык **ориентированных** графов G вместе с двумя вершинами s и t таких, что в G есть путь из s в t.

$\mathrm{stCON} \in \mathcal{NL}$-complete.

Его аналог для неориентированных графов UstCON лежит в $\mathcal{L}$. Это утверждение называется **Reingold’s theorem**.

Задачи

Задача 4.1:

Доказать, что для любой "хорошей" f выполняется

$$\mathrm{DTIME}(f) \subset \mathrm{NTIME}(f) \subset \mathrm{DSPACE}(f) \subset \mathrm{NSPACE}(f) \subset \mathrm{DTIME}\left(2^{O(f)}\right)$$

Задача 4.2:

1) Доказать, что два определения класса $\mathcal{NL}$ эквивалентны.

2) Какой класс получится, если заменить полиномиальную длину сертификата на логарифмическую и убрать ограничение на чтение сертификата единожды?

3) Какой класс получится, если заменить полиномиальную длину сертификата на логарифмическую, но оставить ограничение на чтение сертификата единожды?

4) Какой класс получится, если оставить полиномиальную длину сертификата и убрать ограничение на чтение сертификата единожды?

Задача 4.3:

Доказать, что языки лежат в $\mathcal{L}$

– $\mathrm{ADD} = \{a\#b\#c \mid c = a + b\}$,

– $\mathrm{MULT} = \{a\#b\#c \mid c = a \cdot b\}$,

– язык всех палиндромов,

– язык всех неориентированных графов, содержащих цикл,

– язык всех неориентированных графов, не содержащих цикл.

Задача 4.4:

Пусть $L_1 \leqslant_{\log} L_2$, $L_2 \leqslant_{\log} L_3$

1) доказать, что если $L_2 \in \mathcal{L}$, то $L_1 \in \mathcal{L}$,

2) доказать, что если $L_2 \in \mathcal{NL}$, то $L_1 \in \mathcal{NL}$,

3) доказать, что $L_1 \leqslant_{\log} L_3$.

Задача 4.5:

UstCON $\in \mathcal{L}$, для каждого из языков доказать, что он лежит в $\mathcal{L}$ и что он эквивалентен по сложности UstCON (язык сводится к UstCON и обратно UstCON сводится к языку)

– язык всех не двудольных графов,

– язык всех двудольных графов,

– язык пар граф G и ребро графа e, таких, что в G есть цикл, проходящий через e,

– $\mathrm{XOR2SAT} = \{\varphi \mid \varphi$ есть выполнимая конъюнкция выражений вида $x_i \oplus x_j\}$.

Задача 4.6:

Доказать $\mathcal{NL}$-полноту языков

– $\textsc{ShortestPath} = \{(G, k, s, t) |\ G$ – орграф, кратчайший путь из s в t в графе G имеет длину ровно $k\}$

здесь лучше доказать двумя способами – с помощью свойств языков $\mathcal{NL}$ и co$\mathcal{NL}$ и построением явной НМТ (или ДМТ с сертификатом), распознающей язык,

– всех сильносвязных орграфов,

– всех орграфов, содержащих цикл,

– всех орграфов, не содержащих цикла (DAG),

– 2-SAT,

– всех пар (A, ω), где НКА A принимает слово ω,

– всех таких ДКА, что язык распознаваемый ими, непуст.