Практика №6 по курсу «Теория алгоритмов»
«Иерархия классов сложности»

Группы ФТ-301, ФТ-302

1 Иерархия временных классов сложности

На теории была доказана теорема об иерархии временных классов сложности:

$$\mathtt{DTIME}(f(n)) \subsetneq \mathtt{DTIME}(f(2n+1)^3)$$

Отсюда следует, что $\cup_{c\geq 1}^{k}\mathtt{DTIME}(n^c) \subsetneq P$ для любого k.

На самом деле можно доказать, что $\mathtt{DTIME}(f(n)) \subsetneq \mathtt{DTIME}(g(n))$, если $g(n) = \omega(f(n)\log f(n))$, однако данное доказательство требует более хитрых приёмов ускорения машин Тьюринга.

1.1 Упражнение. «Правильные» функции

Попробуем с помощью результата теоремы об иерархии времени ответить на следующий вопрос: существует ли вычислимая «неправильная» функция? (то есть функция, для которой $f(n)$ нельзя вычислить на машине Тьюринга за время $O(n + f(n))$).

Для ответа на этот вопрос попробуем построить функцию $I(n) : \mathcal{N} \mapsto \{0, 1\}$. Для этого зафиксируем некоторый язык $\mathcal{L}$ над бинарным алфавитом. Также занумеруем все биарные строки натуральными числами так, чтобы для любой строки x длины ℓ её номер $\mathtt{enc}(x)$ не превосходил значения $2^{\ell+1}$ — для этого достаточно рассмотреть числа $1x$ в двоичной системе счисления. Тогда несложно показать, что перевод бинарной строки x длины ℓ в соответствующую унарную строку $1^{\mathtt{enc}(x)}$, а также обратную трансформацию можно произвести с помощью машины Тьюринга, работающей за время $O(\ell 2^\ell) = O(\mathtt{enc}(x)\log\mathtt{enc}(x))$. Тогда построим функцию I следующим образом: $I(n) = [\mathtt{dec}(n) \in \mathcal{L}]$. Если для языка $\mathcal{L}$ существует машина Тьюринга, разрешающая его за время $T(\ell)$, то функция $I(n)$ вычисляется за время $O(T(\log n) + n\log n)$. Чтобы время на вычисление принадлежности слова языку доминировало над временем конвертации числа в двоичное представление выберем $\mathcal{L}$ из разности множеств $\mathtt{DTIME}(2^{(2\ell+1)^6}) \setminus \mathtt{DTIME}(2^{\ell^2})$. Заметим, что для такого языка время вычисления функции $I(n)$ ограничено снизу величиной $\Omega(n^2)$, так как если существует машина Тьюринга, вычисляющая данную функцию за время $f(n) = O(n^2)$, то исходный язык $\mathcal{L}$ можно также разрешить за время $O(n^2) = O(2^{\ell^2})$, а значит $\mathcal{L} \in \mathtt{DTIME}(2^{\ell^2})$. Таким образом $I(n)$ — вычислимая функция, которая не является «правильной».

2 Иерархия классов

На лекциях были рассмотрены следующие соотношения между разными классами сложности:

$$\mathtt{DTIME}(f(n)) \subset_1 \mathtt{NTIME}(f(n)) \subset_2 \mathtt{SPACE}(f(n)) \subset_3 \mathtt{NSPACE}(f(n)) \subset_4 \mathtt{TIME}(k^{f(n)+\log n})$$

Нетривиальным отношениями в этой цепочке являются вложения между разными классами, то есть $\mathtt{NTIME}(f(n)) \subset_2 \mathtt{SPACE}(f(n))$ и $\mathtt{NSPACE}(f(n)) \subset_4 \mathtt{TIME}(k^{f(n)+\log n})$.

$\subset_2$ Для доказательства данного включения достаточно просимулировать работу недетерминированной машины Тьюринга, использую $O(f(n))$ ячеек памяти. Так как время работы машины Тьюиринга ограничено функцией $f(n)$, то каждая ветка исполнения имеет глубину не более чем $f(n)$, а значит можно явно сохранить все решения вдоль данного пути в дереве вычислений машины Тьюринга, используя $O(f(n))$ ячеек памяти

$\subset_4$ Для доказательства данного включения необходимо рассмотреть граф всевозможных конфигураций машины Тьюринга. Так как память МТ ограничена значением $f(n)$, то количество возможных конфигураций не превосходит величины $O(k^{f(n)} \cdot n) = O(k^{f(n)+\log n})$ для некоторой константы k, где множитель n появляться из-за присутствия в конфигурации указателя не исходную ленту, которая имеет n непустых ячеек. В данном графе нужно проверить достижимость одной из терминальных вершин из вершины, соответствующей начальной конфигурации МТ. Данную задачу можно решить за полиномиальное время от размера графа, то есть за время $O((k^{f(n)+\log n})^c) = O((k^c)^{f(n)+\log n}) = O(k'^{f(n)+\log n})$.