ЛЕКЦИЯ 10

ТЕОРЕМА МЕЙЕРА. ПАРАЛЛЕЛЬНОСТЬ. P-ПОЛНОТА

1. Теорема Мейера

На прошлой лекции было рассмотрено понятие класса **P/poly** (класс языков, котрые распознаются семейством схем полиномиального размера) и была доказана **теорема Карпа–Липтона**. Теорема Мейера исходит из более сильных посылок и имеет более сильное утверждение.

Теорема 18 (Мейера) Если EXP $\subset$ P/poly, то EXP $= \sum_2^P$. $*$

Док-во: Пусть $L \in$ EXP. Пусть M — экспоненциальная одноленточная машина Тьюринга, распознающая L. Рассмотрим протокол работы работы M на входе z. Это таблица размера $2^{p(n)} \times 2^{p(n)}$, заполненная $\Gamma \cap Q$.

Задача конкретизации символа в данной клетке протокола будет экспоненциальной.

$$\{(i,j,x) : \text{ в клетке } (i,j) \text{ стоит } x\} \in \text{EXP}.$$

По предположению, что EXP $\subset$ P/poly, можно сделать вывод, что $\{(i,j,x) :$ в клетке (i,j) стоит тоже лежит в P/poly. Можно создать схему, которая по (i,j,x) вычисляет, лежит ли x в (i,j) или нет.

$$\exists\, C : C(z,i,j) = x, \text{ такой что } x \text{ стоит в } (i,j), \text{ если вычисление началось с } z.$$

Предполагаем, что C — правильная схема. Значит, z лежит в языке тогда и только тогда, когда машина переходит в конце в принимающее состояние. Значит, в какой-то из клеток нижнего ряда будет стоять принимающее состояние.

$$z \in L \quad \Leftrightarrow \quad \exists\, j : C(z, 2^{p(n)}, j) = q_{\text{accept}}.$$

Как следует из доказательства теоремы Кука – Левина, корректность протокола равносильна корректности каждого участка размера 2×3.

$$z \in L \quad \Leftrightarrow \quad \exists C : C(z, 2^{p(n)}, j) = q_{\text{accept}} \ \wedge \ \forall\, i \ \forall\, j (C(z,i,j), C(z,i,j+1), ...$$

$$..., C(z, i+1, j+2)) \text{ образуют корректный прямоугольник.}$$

Вышестоящая формула лежит в $\sum_2^P$ (т. к. все кванторы полиномиальны).

2. Оценка схем снизу

Неформально можно утверждать следующее: почти все функции вычисляются схемами размера почти $\dfrac{2^n}{n}$, но предъявить такую функцию с доказательством очень сложно.

Помимо размера существует характеристика формулы под названием **глубина**. Часто возникает вопрос, можно ли вычислить конкретную функцию схемами малой глубины. В связи с этим вопросом возникают сложностные классы внутри P/poly, и вопрос об их глубине связан с вопросом об эффективном параллелизме.

Определение 32: *NC^d — это класс языков, которые распознаются семейством схем полиномиального размера, в которых $\vee$ и $\wedge$ имеют два входа, и глубины $O(\log^d n)$.* ♣

Определение 33: *Класс NC:*

$$NC = \bigcup_{d=0}^{\infty} NC^d.$$

Определение 34: *AC^d — это класс языков, которые распознаются семейством схем полиномиального размера, в которых $\vee$ и $\wedge$ имеют произвольное количество входов, и глубины $O(\log^d n)$.* ♣

Так как можно заменить конъюнкцию и дизъюнкцию k аргументов на двоичное дерево глубины $\log k$, то

$$\mathrm{NC}^d \subset \mathrm{AC}^d \subset \mathrm{NC}^{d+1}.$$

В общем случае непонятно, являются ли эти вложения строгими. В случае $d = 0$:

$$\mathrm{NC}^0 \subsetneq \mathrm{AC}^0.$$

Схема имеет постоянную глубину, и при этом ее элементы имеют два входа. Она может зависеть только от постоянного числа битов входа. Если глубина NC^0 равна l, то схема зависит не более чем от 2^l битов входа.

Конъюнкцию всех входов так вычислить нельзя. С другой стороны, конъюнкция, очевидно, лежит в AC^0.

Трудно доказать следующее утверждение:

$$\mathrm{AC}^0 \subsetneq \mathrm{NC}^1.$$

Следующее утверждение можно доказать, проведя сложение при помощи двоичного дерева:

$$\oplus \in \mathrm{NC}^1.$$

Существует сложная теорема, которая утверждает, что

$$\oplus \notin \mathrm{AC}^0.$$

Остается неизвестным соотношение классов NC^1 и PH.

3. Классы языков с логарифмической памятью

Определение 35: *Uniform-NC^d, Uniform-AC^d — это классы языков, которые распознаются семейством схем полиномиального размера, в которых $\vee$ и $\wedge$ имеют соответственно два входа или произвольное количество входов, и глубины $O(\log^d n)$, причем последовательность схем может вычисляться на логарифмической памяти.* ♣

Справедливы следующие утверждения:

$$\text{Uniform-NC}^d = \mathrm{NC}^d \cap \mathrm{P},$$

$$\text{Uniform-AC}^d = \mathrm{AC}^d \cap \mathrm{P}.$$

Теорема 19 Uniform-$\mathrm{AC}^d \subset \mathrm{L}$. *

Док-во: Приведем идею доказательства.

На логарифмической памяти можно вычислять схему и вычислять ее работу. Схема NC^1 — это двоичное дерево, в узлах которого находятся конъюнкция или дизъюнкция. Поскольку входящая степень у каждой вершины не больше двух, то можно строчкой логарифмической длины задать путь от итоговой точки до входной. Текущий путь можно хранить в памяти вместе с уровнем, на котором находится рекурсия.

4. Параллельность

Параллельная машина — это множество процессоров, соединенных в сеть (например, в сеть в виде гиперкуба). В сети такого типа всего 2^k процессоров, каждый из которых соединен с k соседями, но при этом расстояние между каждым порядка $\log k$. Процессоры могут выполнять вычислительную работу и обмениваться с соседями информацией.

За счет такой сети можно ускорить вычисления. В схеме из NC можно разместить процессор в каждом узле. Процессоры будут вычислять ответы и передавать их дальше. Время вычисления будет равно $(\log k)^{d+1}$. Также можно группировать узлы по разным процессам.

С другой стороны, если имеется эффективное параллельное вычисление, которое выполняет всю работу за полилогарифмическое время, то можно по вычислениям составить схему полилогарифмической глубины.

5. P-полнота

Определение 36: *Язык A называется* ***P-полным****, если $A \in P$, и*

$$\forall\, B \in P \quad B \leqslant_l A \text{ (сводится логарифмически к } A).$$

Теорема 20 Пусть A является P-полным. Тогда:

1. $A \in$ Uniform-NC $\quad\Leftrightarrow\quad$ P = Uniform-NC;
2. $A \in$ L $\quad\Leftrightarrow\quad$ P = L. $*$

Док-во: 2) Пусть A является P-полным, $B \leqslant_l A$, $A \in$ L. Тогда из-за того, что композиция логарифмических вычислений есть логарифмическое вычисление, $B \in$ L.

1) Композиция логарифмических вычислений и NC-вычисления есть NC-вычисление. Нужно доказать, что L $\subset$ NL $\subset$ NC.

В NL есть полная задача о существоании пути (PATH). PATH сводится к умножению матриц. k-кратное произведение матриц смежности даст матрицу путей длины k. Добавление петель даст матрицу путей длины не более k.

Существование пути из s в t равносильно тому, что в клетке (s, t) стоит не ноль. Это можно вычислить за логарифмическое время.