Задание 4

Сложность вычислений: классы P, NP и co-NP

Литература:

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.
   Алгоритмы. Построение и анализ.
   2-е изд. М.: Вильямс, 2005.

2. Sipser M.
   Introduction to the theory of computation

3. Hopcroft J., Ullman J.
   Introduction to Automata Theory, Languages, and Computation.
   1-st edition, 1979.

1 Недетерминированные машины Тьюринга

Определение недетерминированной машины Тьюринга получается из определения детерминированной подобно тому как из определения ДКА получается определение НКА: достаточно заменить функцию переходов на отношение переходов и добавить квантор существования в условие приёма слова. То есть, недетеринированная машина Тьюринга M принимает слово x, если существует последовательность переходов, которая приводит машину M на слове x в принимающее состояние.

Упражнение 1. Покажите, что детереминированные и недетерминированные машины Тьюринга распознают один и тот же класс языков.

Если недетерминированная машина Тьюринга M распознаёт язык L, причём для каждого слова x из L она делает не более чем $O(n^c)$ тактов, то язык L лежит в классе $\mathsf{NTIME}(O(n^c))$. Такую машину Тьюринга мы будем называть недетерминированной полиномиальной.

Таким образом, аналогично определению класса P, получим определение класса NP:

$$\mathsf{NP} = \bigcup_{c \geqslant 0} \mathsf{NTIME}(O(n^c))$$

2 Класс NP

Приведём более наглядное эквивалентное определение класса NP. Под записью $M(x, y)$ мы понимаем, что на вход машине M подали строки x и y, записанные через разделитель, например $M(x, y) = M(x\#y)$.

Напомним, что в случае когда мы рассматриваем соответствующую машине Тьюринга M вычислимую функцию $M(\cdot)$ одного аргумента, запись $M(x) = 1$ означает, что машина M принимает слово x или что то же самое, что слово x принадлежит языку, распознаваемому машиной M: $x \in L(M)$.

Определение 1. Язык L лежит в классе NP, если существуют полином $p(n) : \mathbb{N} \to \mathbb{N}$ и (детерминированная) полиномиальная МТ M, такие что

$$x \in L \Leftrightarrow \exists y \in \Sigma^{p(|x|)}\ M(x, y) = 1.$$

Слово y мы будем называть *сертификатом* для слова x.

Упражнение 2. Покажите, что два определения класса NP эквивалентны.

Задача 1. Пусть мы не накладываем полиномиального ограничения на сертификат, но при этом машина M является полиномиальной по длине входного слова x.

То есть, для языка L есть машина $M(x, y)$ полиномиальная по длине входа $|x|$ и

$$x \in L \Leftrightarrow \exists y \in \Sigma^*\ M(x, y) = 1.$$

Верно ли, что $L \in$ NP? Если да, то как найти полиномиальный по x сертификат y?

3 Сводимости

Довольно часто в сложности вычислений мы сталкиваемся со следующей ситуацией: оказывается, что мы умеем решать задачу A, если мы уже умеем решать задачу B. Или наоборот, задача B является сложной (или даже неразрешимой), если задача A является сложной (неразрешимой).

Для описания таких отношений между языками мы пользуемся сводимостью. В этом задании мы будем говорить об m-сводимости или сводимости по Карпу.

Определение 2. Пусть для языка $A \subseteq \Sigma_1^*$ существует такая вычислимая функция $f : \Sigma_1^* \to \Sigma_2^*$, что слово x принадлежит A тогда и только тогда, когда слово $f(x)$ принадлежит языку B. Будем говорить, что язык A сводится к языку B m-сводимостью и обозначать это как $A \leqslant_m B$. В формулах это выглядит как

Определение 3. Пусть для языка $A \subseteq \Sigma_1^*$ существует такая полиномиально-вычислимая функция $f : \Sigma_1^* \to \Sigma_2^*$, что слово x принадлежит A тогда и только тогда, когда слово $f(x)$ принадлежит языку B. Будем говорить, что язык A сводится к языку B полиномиальной m-сводимостью (сводимостью по Карпу) и обозначать это как $A \leqslant_m^p B$.

Для краткости, мы будем говорить вместо «сводится m-сводимостью» и «сводится полиномиальной m-сводимостью» «сводится» и «полиномиально сводится».

Упражнение 3. Покажите, что если задача HALT сводится к задаче A, то задача A является неразрешимой.

Упражнение 4. Покажите, что если задача A сводится полиномиально к задаче B и задача B лежит в классе P, то и задача A лежит в классе P.

Задача A является NP-полной, если задача A лежит в NP и любая задача $B \in \mathsf{NP}$ полиномиально сводится к A Класс NP-полных задач мы будем обозначать NP-с. Формально

$$L \in \mathsf{NP}\text{-с} \Leftrightarrow L \in \mathsf{NP},\ \forall A \in \mathsf{NP} : A \leqslant_m^p L.$$

Факт существования $\mathsf{NP}-$полных задач установили независимо друг от друга Левин и Кук. В ближайшее время изучение $\mathsf{NP}-$полных задач будет нашим основным полем деятельности.

Определение NP-полного языка состоит из двух частей. Первую из них, о том, что любой язык из класса NP сводится к языку A выделяют в отдельное определение – такие языки называются NP-трудными. Таким образом, язык A является NP-полным, если он является NP-трудным и принадлежит классу NP.

Помимо классов NP и NP-с нас также будет интересовать класс co-NP, состоящий из языков, являющихся дополнением к языкам из NP. То есть, если язык L лежит в классе NP, то язык $\bar{L}$ лежит в классе co-NP

4 NP-полные задачи

Приведём пример NP-полной задачи.

Пример 1. Язык SAT состоит из всех выполнимых булевых формул ϕ, заданных в конъюнктивной нормальной форме.

$$\text{SAT} = \{\phi \mid \exists y_1, \ldots, y_n : \phi(y_1, \ldots, y_n) = 1\}$$

Теорема (Кук, Левин)**.** *Язык* SAT *является* NP*-полным.*

Упражнение 5. Изучить доказательство теоремы Кука-Левина. Ознакомиться с ним я рекомендую в книжке Сипсера.

На семинаре мы говорили о языке 3-SAT, который состоит из выполнимых булевых формул, каждый дизъюнкт которых содержит ровно три переменных. Пример такой формулы:

$$\phi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_4 \vee x_5 \vee x_1).$$

Задача 2. Показать, что из теоремы Кука-Левина следует, что 3-SAT $\in$ NP-с

Задача 3. Показать, что 2-SAT $\in$ P.

Также нам интересен класс co-NP, состоящий из языков, дополнение которых лежит в NP.

Определим язык UNSAT как язык состоящий из невыполнимых булевых формул, заданных КНФ. То есть

$$\text{UNSAT} = \{\phi \mid \forall y_1, \ldots, y_n : \phi(y_1, \ldots, y_n) = 0\}$$

Упражнение 6. Показать, что язык UNSAT лежит в классе co-NP.

Полнота языка в классе относительно сводимостей осмыслена не только в случае NP-полных языков. В учебных целях мы будем пользоваться полиномиальной m-сводимостью также в классе P и co-NP. Разумеется,

полнота относительно полиномиальной сводимости в классе P выглядит нелепо – в реальной жизни класс P исследуют относительно более осмысленной сводимости – сводимости на логарифмической памяти. Относительно этой сводимости, например, полна задача о проверки пустоты языка, заданного КС-грамматикой. То есть, любая задача из класса P сводится к задаче о пустоте языка, заданного КС-грамматикой на логарифмической памяти. Подробнее об этом можно почитать, например, в книжке Хопкрофта-Ульмана «Introduction to Automata Theory, Languages, and Computation» 1979-го года – второе здание этой книги, выпущенное совместно с Мотвани многие из вас изучали в курсе ТРЯП, но второе издание сильно упрощено и сжато.

Задача 4. Верно ли, что любой язык из класса P является полным относительно полиномиальной m-сводимости?

Задача 5†. Докажите, что язык UNSAT является полным в классе co-NP относительно полиномиальной m-сводимости.

5 О сложности вычислений

Соотношение между классами P, NP и NP-с co-NP представляет собой центральный вопрос сложности вычислений. Задача $\mathsf{P} \overset{?}{\neq} \mathsf{NP}$ является основной в сложности вычислений и, по-видимому, безнадёжно трудной. Эта задача стала своего рода наследником теоремы Ферма, в том плане, что регулярно (хотя и не столь часто как в случае теоремы Ферма), находятся люди, которые «доказывают», что $\mathsf{P} = \mathsf{NP}$ или обратное. Интерес к проблеме подогревается институтом Клэя, обещавшим за решение этой задачи \$1 000 000. Тем не менее, недавно была предпринята первая серьёзная попытка действительно доказать, что $\mathsf{P} \neq \mathsf{NP}$. Так что возможно мы всё же доживём до решения этого вопроса.

Сообщество верит, что $\mathsf{P} \neq \mathsf{NP}$ и довольно многие результаты доказываются по модулю этой гипотезы. Более того, сложность этого вопроса используется на практике, в частности в криптографии. В этом курсе мы непосредственно с этим столкнёмся при изучении RSA-алгоритма шифрования. Тем не менее, то что задача лежит в NP и даже в NP-с ещё вовсе не означает что её частный случай, *экземпляр* трудно решить. Это означает лишь, что задача трудна только для некоторых входов.

6 Домашнее задание

Задачи из канонического задания № 17.3, 23, задачи №1-4 из данного текста.

Большинство задач, начиная с 21, я включу в следующее задание, последнее по теме «сложность вычислений». Оставшиеся задачи по этой теме из задавальника будут заданы при изучении чисел. При желании можно присылать решение задач из задавальника по этой теме не дожидаясь того, как я их задам.