

Личные записи по теоринфе ^{β}

@keba4ok

18 октября 2021г.

Некоторые материалы пока что с практик в рамках подготовки к ближайшим контрольным.

Содержание

1 половина Гирша.	2
Что мы решаем?	2
Две ленты.	2
НМТ.	3
Различные классы.	3
Сведения	4
Трудные и полные задачи.	4
Немного про оракулы и дополнения.	5
Полиномиальная иерархия.	5
QBF_k	5
Коллапсы и классы.	5
Опять QBF.	6

1 половина Гирша.



Что мы решаем?

Мы будем работать на бинарных строках.

Определение 1. *Индивидуальная задача* - пара (условие, решение) $\in \{0, 1\}^* \times \{0, 1\}^*$. *Массовая задача* - некоторое множество индивидуальных задач, то есть, бинарное отношение на $\{0, 1\}^*$.

Определение 2. Алгоритм решает *задачу поиска* для массовой задачи R , если для условия x он находит решение w , удовлетворяющее $(x, w) \in R$.

Определение 3. *Язык* (задача распознавания): $L \subseteq \{0, 1\}^*$. Массовой задаче, заданной отношением R , соответствует язык

$$L(R) = \{x \mid \exists w (x, w) \in R\}.$$

В задачах мы будем рассматривать ДМТ, которые почти ничем не отличаются от машин Тьюринга, которые мы рассматривали в первой половине семестра, разве что лент может быть несколько. Одна из них может быть лентой-input, а ещё одна - лентой-output, остальные ленты рабочие, они независимы друг от друга.

Определение 4. ДМТ M *распознаёт* язык A , если принимает все $x \in A$, отвергает все $x \notin A$. Записывается это как $A = L(M)$.

Определение 5. *Время* работы машины M на входе x - количество шагов до достижения принимающего или отвергающего состояния. Используемая *память* - суммарное крайнее правое положение всех головок на рабочих лентах.

Две ленты.

Определение 6. *Универсальная машина Тьюринга*: $M(a, x)$ берёт описание (номер) a машины M_a и её вход x , и выдаёт тот же результат, что и $M_a(x)$.

Теорема 1. Существует универсальная ДМТ $M(a, x)$, использующая лишь две рабочих ленты и выдающая результат за время $O(t \log t)$, где t - время работы M_a на x .

Примечание 1. Универсальную машину Тьюринга можно построить и с одной лентой, но с квадратичным замедлением.

Доказательство. Моделируем через пропуски по модулю k , k лент на одной. Затем заметим, что можно совместить головки и двигать сами смоделированные ленты. Затем происходит какая-то магия с разбиением ленты на кусочки по степеням двойки, а затем внутри них двигаются элементы. \square

НМТ.

Определение 7. *Недетерминированная машина Тьюринга* допускает больше одной инструкции для данных $q \in Q$ и $c_1, \dots, c_k \in \Sigma$, то есть, δ для неё - многозначная функция. Так появляется *дерево вычислений*. Нмт принимает вход, если существует путь в дереве вычислений, который принимается.

Определение 8. В машины с заведомо ограниченным временем работы можно встроить *будильник* и считать время вычислений на входах одной длины всегда одним и тем же.

Определение 9. *Недетерминированная машина Тьюринга* - это просто ДМТ, у которой есть дополнительный аргумент (подсказка w на второй ленте). В рамках такого определения, НМТ M принимает вход x , если существует w , для которой вычисление принимается (пишем $M(x, w) = 1$).

Различные классы.

Определение 10. $t : \mathbb{N} \rightarrow \mathbb{N}$ называется *конструируемой по времени*, если

- $t(n)$ не убывает;
- $t(n) \geq n$;
- двоичную запись $t(|x|)$ можно найти по входу x на ДМТ за $t(|x|)$ шагов.

Язык принадлежит $DTime[t(n)]$, если есть ДМТ M , принимающая L за время $O(t(n))$. Аналогично, $L \in NTime[t(n)]$, если есть НМТ M , принимающая L за время $O(t(n))$.

Определение 11. Массовая задача R *полиномиально ограничена*, если существует полином p , ограничивающий длину кратчайшего решения:

$$\forall x (\exists u(x, u)) \in R \Rightarrow \exists w ((x, w) \in R \wedge |w| \leq p(|x|)).$$

Массовая задача R *полиномиально проверяема*, если существует полином q , ограничивающий время проверки решения: для любой пары (x, w) можно проверить принадлежность $(x, w) \in R$ за время $q(|(x, w)|)$.

\widetilde{NP} - класс задач поиска, задаваемых полиномиально ограниченными полиномиально проверяемыми массовыми задачами.

\tilde{P} - класс задач поиска из \widetilde{NP} , разрешимых за полиномиальное время, то есть, задаваемых отношениями R , такими, что $\forall x \in \{0, 1\}^*$ за полиномиальное время можно найти w , для которого $(x, w) \in R$.

Определение 12. NP - класс языков (задач распознавания), задаваемых полиномиально ограниченными полиномиально проверяемыми массовыми задачами, то есть $NP = \{L(R) | R \in \widetilde{NP}\}$. Иначе говоря, $L \in NP$, если имеется полиномиально ограниченная полиномиально проверяемая R , такая, что

$$\forall x \in \{0, 1\}^* x \in L \iff \exists w(x, w) \in R.$$

P - класс языков (задача распознавания), распознаваемых за полиномиальное время; ясно, что $P = \{L(R) | R \in \tilde{P}\}$.

Сведёния

Определение 13. Сведение языков *по Карпу*: $L_1 \rightarrow L_2$, если имеется полиномиально вычислимая f :

$$\forall x x \in L_1 \iff f(x) \in L_2.$$

Определение 14. Сведение задач поиска *по Левину*: $R_1 \rightarrow R_2$, если существуют f, g, h такие, что для любых x_1, y_1 и y_2 :

- $R_1(x_1, y_1) \iff R_2(f(x_1), g(x_1, y_1))$;
- $R_1(x_1, h(x_1, y_2)) \iff R_2(f(x_1), y_2)$;
- f, g и h полиномиально вычислимы.

Определение 15. *Оракульная МТ* имеет доступ к оракулу, который за 1 шаг даёт ей ответ на вопрос. M^B - оракульная машина M , которой дали конкретный оракул B .

Определение 16. Сведение чего угодно *по Тьюрингу*: $A \rightarrow B$, если имеется оракульная полиномиальная по времени машина M^\bullet , такая, что M^B решает A (например, если A - язык, то $A = L(M^B)$).

Примечание 2. Классы P и \tilde{P} замкнуты относительно всех этих сведений. Классы же неполиномиальные могут быть незамкнуты относительно сведений по Тьюрингу.

Трудные и полные задачи.

Определение 17. Задача A - *трудная* для класса C , если $\forall D \in C, D \rightarrow A$. Задача - *полная* для C , если она трудная и принадлежит C .

Теорема 2. Если A - NP -трудная и $A \in P$, то $P = NP$.

Следствие 1. Если A - NP -полная, то $A \in P$ тогда и только тогда, когда $P = NP$.

Определение 18. *Задача об ограниченной остановке*: $\widetilde{BH}(\langle M, x, 1^t \rangle, w) = \text{НМТ } M \text{ с под- сказкой } w \text{ принимает вход } x \text{ за } \leq t \text{ шагов.}$

Теорема 3. *Задача об ограниченной остановке* - \widetilde{NP} -полная, а соответствующий язык - NP -полный.

Определение 19. $\widetilde{CIRCUIT_SAT} = \{(C, w) | C - \text{схема}, C(w) = 1\}$. Эта задача также NP -полная.

Определение 20. $3 - \widetilde{SAT} = \{(F, A) | F - \text{в } 3\text{-КНФ}, F(A) = 1\}$. Очередная NP -полная задача.

Теорема 4. $R \in \widetilde{NP}$, язык $L(R)$ - NP -полон, тогда $R \rightarrow L(R)$ (поиск и распознавание).

Теорема 5. Если $P \neq NP$, то существует язык $L \in NP \setminus P$, не являющийся NP -полным.

Немного про оракулы и дополнения.

Определение 21. Для классов \mathcal{C} , (D) новый класс \mathcal{C}^D состоит из языков вида C^D , где $D \in \mathcal{D}$, C - машина для языка из \mathcal{C} .

Определение 22. $co - C = \{L | \bar{L} \in C\}$.

Полиномиальная иерархия.

Теорема 6. $L \in \Sigma^k P$ тогда и только тогда, когда существует полиномиально ограниченное отношение $R \in \Pi^{k-1} P$, такое, что для любого x

$$x \in L \Leftrightarrow \exists y R(x, y).$$

Следствие 2. $L \in \Pi^k P$ тогда и только тогда, когда существует полиномиально ограниченное отношение $R \in \Sigma^{k-1} P$, такое, что для любого x

$$x \in L \Leftrightarrow \forall y R(x, y).$$

Следствие 3. А значит, мы можем расписать такие длинные цепочки, например, $L \in \Sigma^k P$ тогда и только тогда, когда существует полиномиально ограниченное $R \in P$, такое, что для любого x

$$x \in L \Leftrightarrow \exists y_1 \forall y_2 \exists y_3 \dots R(x, y_1, y_2, y_3, \dots),$$

и аналогично для другого класса, только чередование другое.

QBF_k

Определение 23. Язык QBF_k состоит из замкнутых истинных формул вида

$$\exists X_1 \forall X_2 \exists X_3 \dots X_k \varphi,$$

где каждый квантор состоит из большого количества утверждений для малых переменных с соответствующим квантором, а φ - формула в КНФ или ДНФ. $\{X_i\}_{i=1}^k$ - в свою очередь, разбиение множества переменных этой формулы на непустые непересекающиеся подмножества.

Следствие 4. QBF_k - $\Sigma^k P$ -полна.

Коллапсы и классы.

Теорема 7. Если $\Sigma^k P = \Pi^k P$, то $PH = \sigma^k P$ (PH - объединение всех таких классов по всем степеням).

Следствие 5. Если существует PH -полная задача, то полиномиальная иерархия конечна.

Определение 24.

$$DTime[f(n)] = \{L | L \text{ принимается ДМТ, работающей время } O(f(n))\};$$

$$DSpace[f(n)] = \{L | L \text{ принимается ДМТ с памятью } O(f(n))\};$$

$$PSPACE = \bigcup_{k \geq 0} DSpace[n^k].$$

Опять QBF.

Определение 25. Язык QBF состоит из замкнутых истинных формул вида

$$q_1x_1q_2x_2\ldots\varphi,$$

где φ - формула в КНФ, $q_i = \forall$ или $q_i = \exists$.

Теорема 8. QBF $PSPACE$ -полна.

Следствие 6. $PH = PSPACE \Rightarrow PH$ коллапсирует.

Предметный указатель

Будильник, 3

Задача

индивидуальная, 2

массовая, 2

об ограниченной остановке, 4

поиска, 2

полная, 4

трудная, 4

Оракул, 4

Сведение

по Карпу, 4

по Левину, 4

по Тьюрингу, 4

Язык, 2