Личные записи по теоринфе$^{\beta}$

@keba4ok

18 октября 2021г.

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

Содержание

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 выход 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. □

НМТ.

Определение 7. *Недетерминированная машина Тьюринга* допускает больше одной инструкции для данных $q \in Q$ и $c_1, \ldots, c_k \in \Sigma$, то есть, δ для неё - многозначная функция. Так появляется *дерево вычислений*. Нмт принимает вход, если существет 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. $t : \mathbb{N} \to \mathbb{N}$ 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.

$\widetilde{P}$ - класс задач поиска из $\widetilde{NP}$, разрешимих 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

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

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

Сведе́ния

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

$$\forall x \; x \in L_1 \Longleftrightarrow f(x) \in L_2.$$

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

- $R_1(x_1, y_1) \Leftrightarrow R_2(f(x_1), g(x_1, y_1))$;
- $R_1(x_1, h(x_1, y_2)) \Leftarrow R_2(f(x_1), y_2)$;
- 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*: $A \to B$, 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 $\widetilde{P}$ замкнуты относительно всех этих сведений. Классы же неплоиномиальные могут быть незамкнуты относительно сведений по Тьюрингу.

Трудные и полные задачи.

Определение 17. Задача A - *трудная* для класса C, если $\forall D \in C$, $D \to A$. 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*: $\widetilde{BH}(\langle M, x, 1^t \rangle, w) =$ НМТ M с подсказкой w принимает входд x за $\leq t$ шагов.

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

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

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

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

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

Немного про оракулы и дополнения.

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

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

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

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

и аналогично для другого класса, только чередование другое.

QBF_k

Определение 23. Язык QBF_k состоит из замкнутых истинных формул вида

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

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

$$q_1 x_1 q_2 x_2 \dots \varphi,$$

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

Теорема 8. *QBF $PSPACE$-полна.*

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

2 половина Гирша.

Иерархии.

Теорема 9. $DSpace[s(n)] \neq DSpace[S(n)]$, *где* $s(n) = o(S(n))$ *и* $\forall n > n_0$, $S(n) \geq \log n$.

Теорема 10. $DSpace[\log \log n] \neq DSpace[O(1)]$.

Теорема 11. $\forall \varepsilon > 0$, $DSpace[(\log \log n)^{1-\varepsilon}] = DSpace[O(1)]$.

Теорема 12. $DTime[t(n)] \neq DTime[T(n)]$, *где* $t(n) \log t(n) = o(T(n))$, $T(n) = \Omega(n)$.

Следствие 7. $P \neq EXP$.

Теорема 13. $NTime[t(n)] \neq NTime[T(n)]$, *где* $t(n+1) = o(T(n))$ *конструируема по времени.*

Схемочки.

Определение 26. $L \in Size[f(n)]$, если существует семейство булевых схем $\{C_n\}_{n \in \mathbb{N}}$ таких, что

- $\forall n \ |C_n| \leq f(n)$;
- $\forall x \ (x \in L \Leftrightarrow C_{|x|(x)=1})$.

$P/poly = \bigcup_{k \in \mathbb{N}} Size[n^k]$.

Теорема 14 (*Теорема Карпа-Левина*)**.** $NP \subseteq P/poly \Rightarrow PH = \Sigma^2 P$.

Теорема 15. $\forall k\ \Sigma^4 P \not\subseteq Size[n^k]$,
Следствие 8. $\forall k\ \Sigma^2 P \cap \Pi^2 P \not\subseteq Size[n^k]$.

Классы.

Определение 27. $NSpace[f(n)]$ - множество языков, принимаемых НМТ с памятью $O(f(n))$. $NPSpace = \bigcup_{k\geq 0} NSpace[n^k]$. $L = DSpace(\log n)$, $NL = NSpace(\log n)$.

Определение 28. $STCON$ - задача нахождения пути из одной вершины в другую в связном графе.

Лемма 1. $STCON \in DSpace[\log^2 n]$.

Лемма 2. *$STCON$ является NP-полной задачей (относительно logspace-сведений).*

Определение 29. Семейство схем $\{C_n\}_{n\in\mathbb{N}}$ *равномерно*, если имеется полиномиальный алгоритм A такой, что $A(1^n) = C_n$.

Примечание 3. Равномерные полиномиальные схемы задают P.

Определение 30. *Logspace-равномерные схемы*: A использует память $O(\log n)$. Глубина съемы $\sim$ время параллельного вычисления. LC -множество языков таких, что для них есть *logspace*-равномерные схемы глубины $O(\log^i n)$. NC - объединение их всех по i, лежит в P.

Лемма 3. *Композиция двух logspace функций лежит в logspace.*

Теорема 16. *Если L - P-полный, то*

- $L \in \boldsymbol{L} \Leftrightarrow \boldsymbol{P} = \boldsymbol{L}$;
- $L \in \boldsymbol{NC} \Leftrightarrow \boldsymbol{P} = \boldsymbol{NC}$.

Параллельные вычисления.

Теорема 17.

$$NC^1 \subseteq L \subseteq NL \subset NC^2.$$

Теорема 18. *Если L - P-полный, то $L \in NC$ тогда и только тогда, когда $P = NC$.*

Теорема 19. $STCON \in co - NL$.
Следствие 9. Если $s(n) = \Omega(\log n)$, то $NSpace[s(n)] = co - NSpace[s(n)]$.

Вероятности.

Определение 31. $L \in RP$, если имеется полиномиально ограниченное и полиномиально проверяемое отношение R такое, что для любой строчки из нулей и единиц, если она не лежит в языке, то для любой подсказки оно не пройдёт проверочку, а иначе пройдёт с хотя бы половиной ото всех подсказок. $ZPP = RP \cap co - RP$ - безошибочная хуйня. Двусторонняя ошибка же - это когда имеется по пп отношение такое, что если x не в языке, то подсказок, которые это выявят не более трети, а иначе подходящих подсказочек более $\frac{2}{3}$.

Утверждение 1. Неравенство Чернова? Нахуя оно тут? Ищите сами

Теорема 20. $BPP \subseteq \Sigma^2 P$.

Теорема 21 (*Теорема Тода*)**.** $PH \subseteq P^{PP}$.

Дальше параша какая-то, займёмся этим на экзамене.

Предметный указатель