Полиномиальные сводимости и $\mathcal{NP}$-полнота

Н. Н. Кузюрин С. А. Фомин

13 декабря 2011 г.

Труднорешаемые задачи

Определение

*Алгоритмическая задача называется **труднорешаемой**, если для нее не существует полиномиального алгоритма.*

Существуют ли *разрешимые* задачи, которые тем не менее не принадлежат классу $\mathcal{P}$?
Ответ — «теорема об иерархии».

Теорема

Существует алгоритмическая задача, разрешимая некоторым алгоритмом сложности $n^{O(\log n)}$, но не принадлежащая классу $\mathcal{P}$.

Сводимость по Куку

Определение

Алгоритмическая задача P_1 полиномиально сводится к задаче P_2, если существует полиномиальный алгоритм для решения задачи P_1, возможно, вызывающий в ходе своей работы процедуру для решения задачи P_2.

Проблема — переполнение памяти.

Вычисление 2^{2^t}

Вход: Натуральное t

$R \leftarrow 2$
for all $i \in 1..t$ **do**
  $R \leftarrow R \times R$
end for

$\mathcal{P}$ получается незамкнут при сводимости по Куку.

Выход: рассматривать только «задачи разрешения».

Дискретные задачи оптимизации

Задача

«Коммивояжер», «TSP[a]». Заданы неориентированный граф из n вершин-городов, и $d_{ij} \equiv d(v_i, v_j)$ — положительные целые расстояния между городами.
Чему равна наименьшая возможная длина гамильтонова цикла (кольцевого маршрута, проходящего по одному разу через все города)? т. е. нужно найти минимально возможное значение суммы

$$\min_{p \in \begin{pmatrix} 1 & 2 & \dots & n \\ . & . & . & . \end{pmatrix}} \sum_{i=1}^{n-1} d_{p_i, p_{i+1}} + d_{p_n, p_1}, \tag{1}$$

где минимум берется по всем перестановкам p чисел $1, \dots, n$.

[a] В англоязычной литературе — Traveling Salesman Problem.

Переборные задачи разрешения

Задача

«TSP-разрешимость». Заданы:

- *n городов* $c_1, \ldots, c_n$*;*
- $d_{ij} \equiv d(c_i, c_j) \in Z^+$ *— расстояния между ними;*
- B *— положительное целое.*

Верно ли, что минимально возможное значение суммы (1) *не больше* B*?*

Сведение бинарным поиском

Вход: $n > 0$, $d_{ij} > 0$, $\forall i, j \in (1, \ldots, n)$.
Выход: Перестановка π_{opt}, такая, что сумма (1) минимальна.

{Используем процедуру $TSP_{bool}(n, D, B)$ для задачи 2 «TSP-разрешимость»}
$B_{\min} \leftarrow 0$
$B_{\max} \leftarrow n \cdot \max_{1 \le i < j \le n} d(c_i, c_j)$
while $B_{\max} - B_{\min} > 1$ **do**
  $B \leftarrow \lfloor (B_{\min} + B_{\max})/2 \rfloor$
  if $TSP_{bool}(n, D, B)$ **then**
    $B_{\max} \leftarrow B$
  else
    $B_{\min} \leftarrow B$
  end if
end while
if $TSP_{bool}(n, D, B_{min})$ **then**
  return $B_{\min}$
else
  return $B_{\max}$
end if

Определение

Недетерминированная машина Тьюринга (НМТ) *— это набор* $\mathsf{T} = \langle k, \Sigma, \Gamma, \Phi \rangle$*, где*

- $k \geq 1$ *— натуральное число (число лент),*
- Σ *— «алфавит лент», конечное множество,*
- $\star \in \Sigma$ *— алфавит содержит «пробел»,*
- Γ *— конечное множество состояний,* $START, STOP \in \Gamma$*,*
- Φ *— произвольное отношение:*

$$\Phi \subset (\Gamma \times \Sigma^k) \times (\Gamma \times \Sigma^k \times \{-1, 0, 1\}^k).$$

Переход из состояния g*, с символами на лентах* $h_1, \ \ldots, h_k$ *будет допустим, если новое состояние* g'*, записанные символы* $h'_1, \ \ldots, h'_k$ *и смещения головок* $\varepsilon_1, \ \ldots, \varepsilon_k$ *удовлетворяют соотношению*

$$(g, h_1, \ \ldots, h_k, g', h'_1, \ \ldots, h'_k, \varepsilon_1, \ \ldots, \varepsilon_k) \in \Phi.$$

Недетерминированная машина Тьюринга

Определение

Недетерминированный алгоритм выдает окончательный ответ «1», если существует хотя бы один путь развития вычисления, на котором выдается ответ 1, и «0» — в противном случае.

Временная сложность для НМТ

$\mathcal{NTIME}(f(n)), \mathcal{NSPACE}(f(n))$ определяются аналогично $\mathcal{DTIME}(f(n)), \mathcal{DSPACE}(f(n))$.

Временная сложность для НМТ

$\mathcal{NTIME}(f(n)), \mathcal{NSPACE}(f(n))$ определяются аналогично $\mathcal{DTIME}(f(n)), \mathcal{DSPACE}(f(n))$.

Но для детерминированной МТ классы сложности языков замкнуты относительно дополнения.

$$\begin{aligned} \text{co}\mathcal{DTIME}(f(n)) &\equiv \{L | \bar{L} \in \mathcal{DTIME}(f(n))\} \\ \text{co}\mathcal{DTIME}(f(n)) &= \mathcal{DTIME}(f(n)) \end{aligned}$$

Временная сложность для НМТ

$\mathcal{NTIME}(f(n)), \mathcal{NSPACE}(f(n))$ определяются аналогично $\mathcal{DTIME}(f(n)), \mathcal{DSPACE}(f(n))$.

Но для детерминированной МТ классы сложности языков замкнуты относительно дополнения.

$$\begin{aligned} \text{co}\mathcal{DTIME}(f(n)) &\equiv \{L | \overline{L} \in \mathcal{DTIME}(f(n))\} \\ \text{co}\mathcal{DTIME}(f(n)) &= \mathcal{DTIME}(f(n)) \end{aligned}$$

А классы сложности для НМТ — скорее всего не замкнуты.

$$\begin{aligned} \text{co}\mathcal{NTIME}(f(n)) &\stackrel{?}{=} \mathcal{NTIME}(f(n)) \\ \text{co}\mathcal{NSPACE}(f(n)) &\stackrel{?}{=} \mathcal{NSPACE}(f(n)) \end{aligned}$$

Классы $\mathcal{NP}$ и co$\mathcal{NP}$

Определение

$$\begin{aligned} \mathcal{NP} &= \cup_{k\geq 0}\mathcal{NTIME}(n^k), \\ \text{co}\mathcal{NP} &= \{L|\bar{L} \in \mathcal{NP}\}. \end{aligned}$$

Упражнение

Покажите, что $\mathcal{P} \subseteq \mathcal{NP} \cap \text{co}\mathcal{NP}$.

Упражнение

Покажите, что $\mathcal{NP} \subseteq \mathcal{PSPACE}$

Определение $\mathcal{NP}$ через ДМТ

Определение

Язык $L \subseteq \Sigma^*$ принадлежит классу $\mathcal{NP}$, если существуют полиномиальная детерминированная машина Тьюринга M *и полином $p(\cdot)$, такие, что*

$$L = \{x \in \Sigma^* : \exists\, y,\ |y| < p(|x|) \& \,\mathrm{M}(x, y) = 1\}.$$

Слово y называется обычно «подсказкой», «свидетелем» (*witness*), «доказательством» (*proof*).

Определение $\mathcal{NP}$ через ДМТ

Определение

Язык $L \subseteq \Sigma^*$ принадлежит классу $\mathcal{NP}$, если существуют полиномиальная детерминированная машина Тьюринга M *и полином $p(\cdot)$, такие, что*

$$L = \{x \in \Sigma^* : \exists\, y,\ |y| < p(|x|) \& \,\mathrm{M}(x, y) = 1\}.$$

Слово y называется обычно «подсказкой», «свидетелем» (*witness*), «доказательством» (*proof*).

Теорема

Определения 0.5 «$\mathcal{NP}$/НМТ» и 0.6 «$\mathcal{NP}$/ДМТ» эквивалентны.

0.5 «$\mathcal{NP}$/НМТ» $\Leftrightarrow$ 0.6 «$\mathcal{NP}$/ДМТ»

$\Rightarrow$: $\forall x \in L \rightarrow$ подсказка y: закодированный кратчайший протокол выполнения-подтверждения НМТ T из определения 0.5 «$\mathcal{NP}$/НМТ».

0.5 «$\mathcal{NP}$/НМТ» $\Leftrightarrow$ 0.6 «$\mathcal{NP}$/ДМТ»

$\Rightarrow$: $\forall x \in L \rightarrow$ подсказка y: закодированный кратчайший протокол выполнения-подтверждения НМТ T из определения 0.5 «$\mathcal{NP}$/НМТ».

- Длина кратчайшего пути-подтверждения $\mathrm{T}(x)$ полиномиально ограничена $\Rightarrow$ $y(x)$ — полиномиально ограничена.

0.5 «$\mathcal{NP}$/НМТ» $\Leftrightarrow$ 0.6 «$\mathcal{NP}$/ДМТ»

$\Rightarrow$: $\forall x \in L \rightarrow$ подсказка y: закодированный кратчайший протокол выполнения-подтверждения НМТ T из определения 0.5 «$\mathcal{NP}$/НМТ».

- Длина кратчайшего пути-подтверждения $\mathrm{T}(x)$ полиномиально ограничена $\Rightarrow$ $y(x)$ — полиномиально ограничена.
- Проверка протокола $y(x)$ занимает полиномиальное время.

0.5 «$\mathcal{NP}$/НМТ» $\Leftrightarrow$ 0.6 «$\mathcal{NP}$/ДМТ»

$\Rightarrow$: $\forall x \in L \rightarrow$ подсказка y: закодированный кратчайший протокол выполнения-подтверждения НМТ T из определения 0.5 «$\mathcal{NP}$/НМТ».

- Длина кратчайшего пути-подтверждения $\mathrm{T}(x)$ полиномиально ограничена $\Rightarrow$ $y(x)$ — полиномиально ограничена.
- Проверка протокола $y(x)$ занимает полиномиальное время.

$\Leftarrow$: Пусть НМТ T:

1. недетерминированно дописывает # и некоторое слово y к входному слову x,
2. работает над словом $x\#y$, как полиномиальная ДМТ из определения 0.6 «$\mathcal{NP}$/ДМТ».

Сводимость по Куку: незамкнутость $\mathcal{NP}$ и co$\mathcal{NP}$

Задача разрешения «Comm$(m, d) > B$»
верно ли, что *любой* маршрут коммивояжера имеет длину по крайней мере $(B + 1)$?

- принадлежит co$\mathcal{NP}$
- не принадлежит классу $\mathcal{NP}$

при общепринятой гипотезе $\mathcal{P} \neq \mathcal{NP}$.

В то же время она очевидным образом сводится по Куку к переборной задаче 2 «TSP-разрешимость» из $\mathcal{NP}$.

Сводимость по Карпу

Определение

Задача разрешения P_1 ***полиномиально сводится*** *к задаче разрешения P_2, если*

- *существует полиномиально вычислимая функция $f : I_1 \to I_2$, (отображает входные данные I_1 для P_1 во входные данные $I_2 \equiv f(I_1)$ для задачи P_2),*
- $\forall I_1$ *совпадают ответы на вопросы «$P_1(I_1)$?» и «$P_2(f(I_1))$?».*

$\mathcal{NP}$-полные задачи

Определение

Задача разрешения называется $\mathcal{NP}$-полной[a], если

- *она принадлежит классу $\mathcal{NP}$,*
- *произвольная задача из $\mathcal{NP}$ сводится к ней полиномиально (См. определение 0.7 «Сводимость по Карпу»).*

Класс $\mathcal{NP}$-полных задач обозначается $\mathcal{NPC}$.

---

[a]Чтобы не перегружать лекции излишней терминологией, мы будем называть в дальнейшем оптимизационную задачу *$\mathcal{NP}$-полной*, если $\mathcal{NP}$-полна соответствующая задача разрешения.

В определение $\mathcal{NPC}$ входит принадлежность классу $\mathcal{NP}$!

Если опустить это условие, получится класс $\mathcal{NP}$-трудных (*$\mathcal{NP}$-hard*) задач, включающих $\mathcal{NPC}$, но выходящих за границы класса $\mathcal{NP}$ (при гипотезе $\mathcal{P} \neq \mathcal{NP}$).

Первая $\mathcal{NP}$-полная задача

Задача

«Выполнимость/SAT»[a]. Дано булевское выражение, являющееся ***конъюнктивной нормальной формой*** *(КНФ):*

$$CNF = \bigwedge_{i=1}^{m} C_i, \tag{2}$$

где C_i — элементарные дизъюнкции вида

$$x_{j_1}^{\sigma_1} \vee \ \ldots \vee x_{j_k}^{\sigma_k}, \tag{3}$$

$1 \leq k \leq n$, $\sigma_j \in \{0, 1\}$, $x^1 = x$ *и* $x^0 = (\neg x)$.
Существует ли (булевский) набор переменных x_j, обращающий эту форму в 1 *(т. е. в «Истину»)?*

---

[a] В англоязычной литературе — Satisfiability или просто SAT.

Задача 3 «SAT» — $\mathcal{NP}$-полна.

$\forall x$ — выполнимой входной КНФ:

- $\exists y$ — значения переменных, выполняющих x,
- $x(y) \stackrel{?}{=} 1$ — проверяется за полиномиальное время.

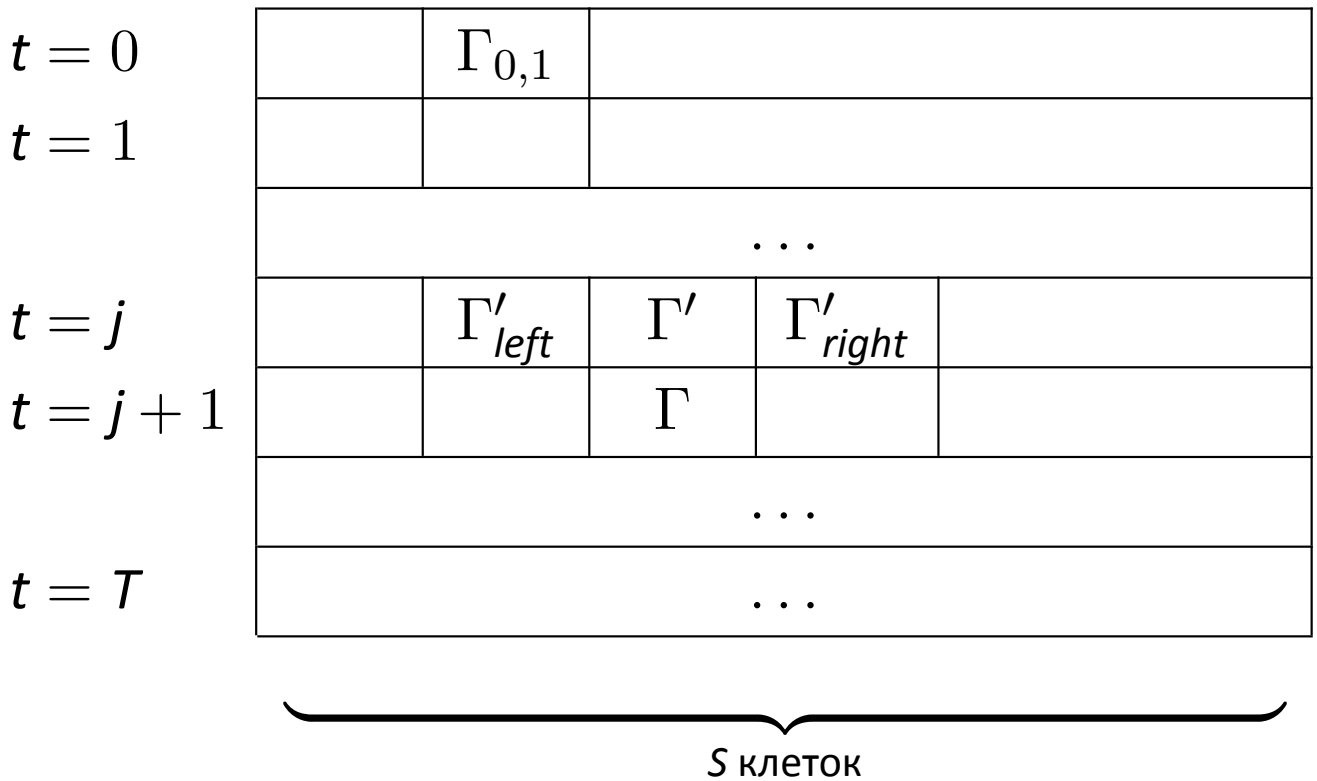
$\Rightarrow$ задача 3 «SAT»$\in \mathcal{NP}$.

Произвольный $L \in \mathcal{NP}$: $\Rightarrow$

1. $\forall x \in L, \exists y(x) : |y(x)| < poly(|x|)$,
2. $\exists$ полиномиальная МТ M, распознающая $L_y = \{x\#y(x) | x \in L\}$.

Процесс вычисления на x длины n можно представить *таблицей вычисления* размера $T \times S$, где $T = \mathrm{poly}(n)$, $S = \mathrm{poly}(n)$.

Задача 3 «SAT» — $\mathcal{NP}$-полна.



В ячейке (i, j) состояние МТ, если головка на i-й позиции, иначе $\emptyset$.
Ячейки, кодирующие часть клеток нулевой строки, определяются x,
Результат вычисления в ячейке $(t, 0)$.

Задача 3 «SAT» — $\mathcal{NP}$-полна.

Ячейки $\equiv$ переменные, вся таблица $\equiv$ формула (в форме КНФ).

В таблице правильное вычисление с ответом 1:

- должны выполняться локальные правила согласования для каждой четвёрки клеток вида $\boxplus$
- результат должен быть 1.

- φ_x — конъюнкция всех этих формул
- первая строка — $x\#y$
  - $x\#$ – константы
- остальные ячейки — переменные y и z.

- M распознает $x\#y \Rightarrow \exists z(x,y) : \varphi_x(y, z(x,y)) = 1$.
- M не распознает $x\#y \Rightarrow \varphi_x(y,z) \equiv 0$.

Доказательство через сводимость

Если $\mathcal{NP}$-полная задача P_1 полиномиально сводится к переборной задаче P_2, то P_2 также $\mathcal{NP}$-полна.

Задача

«3-Выполнимость/3SAT».
Вариант задачи 3 «SAT», где каждая элементарная дизъюнкция (3) *имеет длину* $k \leq 3$. *Соответствующие КНФ называются 3-КНФ.*

Заменим каждую элементарную дизъюнкцию с $k > 3$ на:

$$\left(y_{i2} \equiv (x_{j_1}^{\sigma_1} \vee x_{j_2}^{\sigma_2})\right) \wedge \left(y_{i3} \equiv (y_{i2} \vee x_{j_3}^{\sigma_3})\right) \wedge \ \ldots \wedge \left(y_{ik} \equiv (y_{i,k-1} \vee x_{j_k}^{\sigma_k})\right) \wedge y_{ik},$$

где $y_{i2}, \ \ldots, y_{ik}$ — новые булевы переменные, и трансформируем $y_{i\nu} \equiv (y_{i,\nu-1} \vee x_{j_\nu}^{\sigma_\nu})$ в 3-КНФ.

Задача 4 «3SAT» — $\mathcal{NP}$-полна.

«2SAT» $\in \mathcal{P}$

Задача

«2-Выполнимость»(2SAT). Частный случай задачи 3 «SAT», в котором каждая элементарная дизъюнкция имеет длину $k \leq 2$. Соответствующие КНФ называются 2-КНФ.

Можно исключить все дизъюнкции, состоящие из одного терма:
Если $(x_i) \Rightarrow x_i \equiv 1 \Rightarrow$ выкидываем все дизъюнкты с x_i в положительной степени, а из дизъюнктов содержащих $\neg x_i$, выкидываем $\neg x_i$ (т.к. $\neg x_i \equiv 0$). Если есть дизъюнкт $(\neg x_i)$ — формула неразрешима.

Аналогично для $(\neg x_i)$.

$\Rightarrow$ все дизъюнкты в КНФ из двух термов.

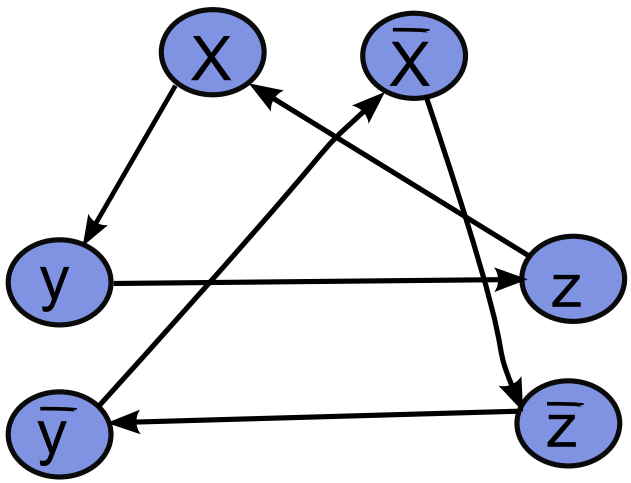
«2SAT» $\in \mathcal{P}$

$$x \vee y \equiv (\neg x \to y) \wedge (\neg y \to x)$$

2SAT-формула, с n переменных $x_i \Rightarrow$ ориентированный граф:

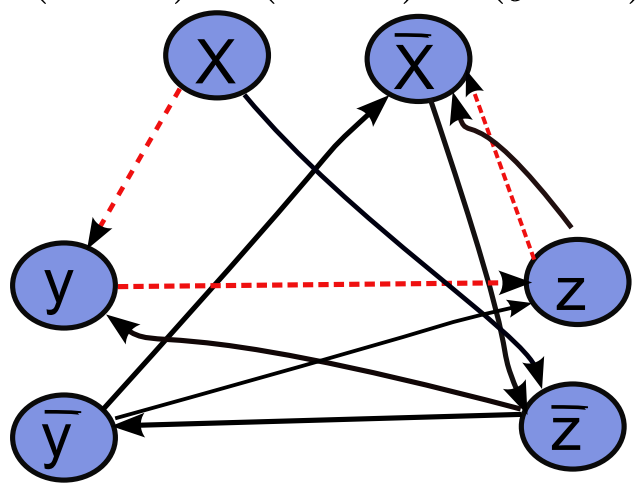
- $2n$ узлов: $\forall i\ x_i,\ \neg x_i$
- $2m$ дуг: дизъюнкция $(x \vee y) \to$ дуги $(\neg x \to y)$ и $(\neg y \to x)$.

$$(\overline{x} \vee y) \wedge (\overline{y} \vee z) \wedge (x \vee \overline{z})$$



Выполнимая 2КНФ

$$(\overline{x} \vee y) \wedge (\overline{y} \vee z) \wedge (x \vee \overline{z}) \wedge (\overline{x} \vee \overline{z}) \wedge (y \vee z)$$



Невыполнимая 2КНФ

«Вершинное покрытие»

Задача

«Вершинное покрытие»[a].
Дан граф $G = (V, E)$ *и положительное целое число* K, $K \leq |V|$.
Имеется ли в графе G ***вершинное покрытие*** *не более чем из* K *элементов, т. е. такое подмножество* $V' \subseteq V$, *что* $|V'| \leq K$ *и каждое ребро из* E *содержит хотя бы одну вершину из* V'?

[a] В англоязычной литературе — *Vertex Covering*.

Лемма

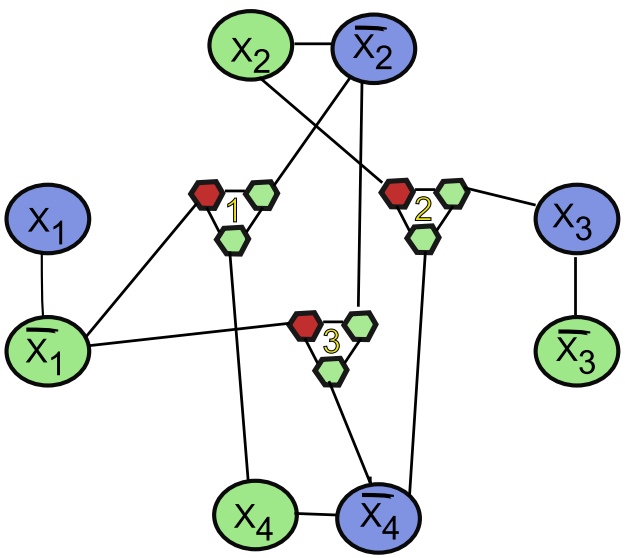
Задача 6 «Vertex Covering» лежит в $\mathcal{NPC}$.

Задача 6 «Vertex Covering» $\in \mathcal{NPC}$

3-КНФ от $x_1, \ldots, x_n$, $k = 3$ для всех (m) ЭД.

- $\forall j\ x_j \Rightarrow$ дуга $(x_j, \neg x_j)$
- $\forall$ ЭД $\Rightarrow$ треугольник (v_{i1}, v_{i2}, v_{i3})
- соединяем дугами v_{i1} с $x_{j_1}^{\sigma_1}$, v_{i2} — с $x_{j_2}^{\sigma_2}$ и v_{i3} — с $x_{j_3}^{\sigma_3}$.

$(\overline{x_1} \vee \overline{x_2} \vee x_4) \wedge (x_2 \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_4})$



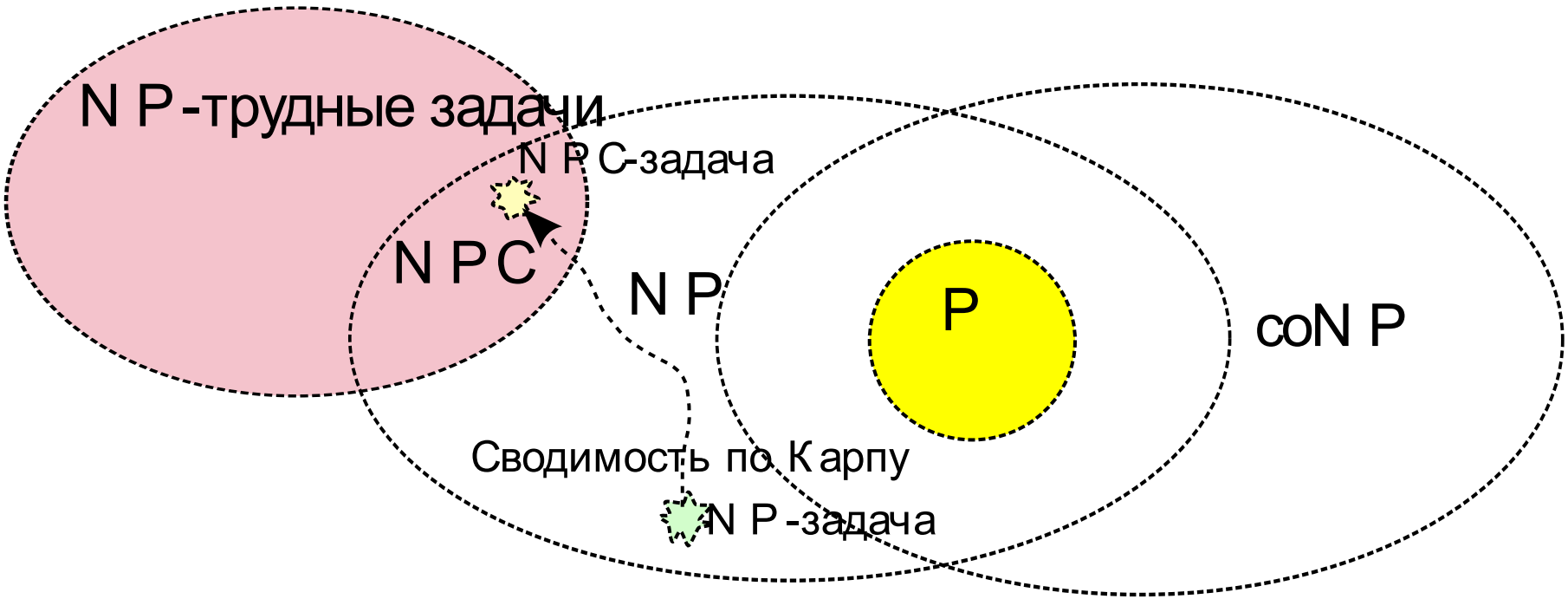
Выполнимо при $x_1 = 0, x_2 = 1$

(x_3, x_4 — любые).

Вершинное покрытие должно иметь размер не менее $(n + 2m)$ (n вершин на $(x_j, \neg x_j)$ и $2m$ на (v_{i1}, v_{i2}, v_{i3})).

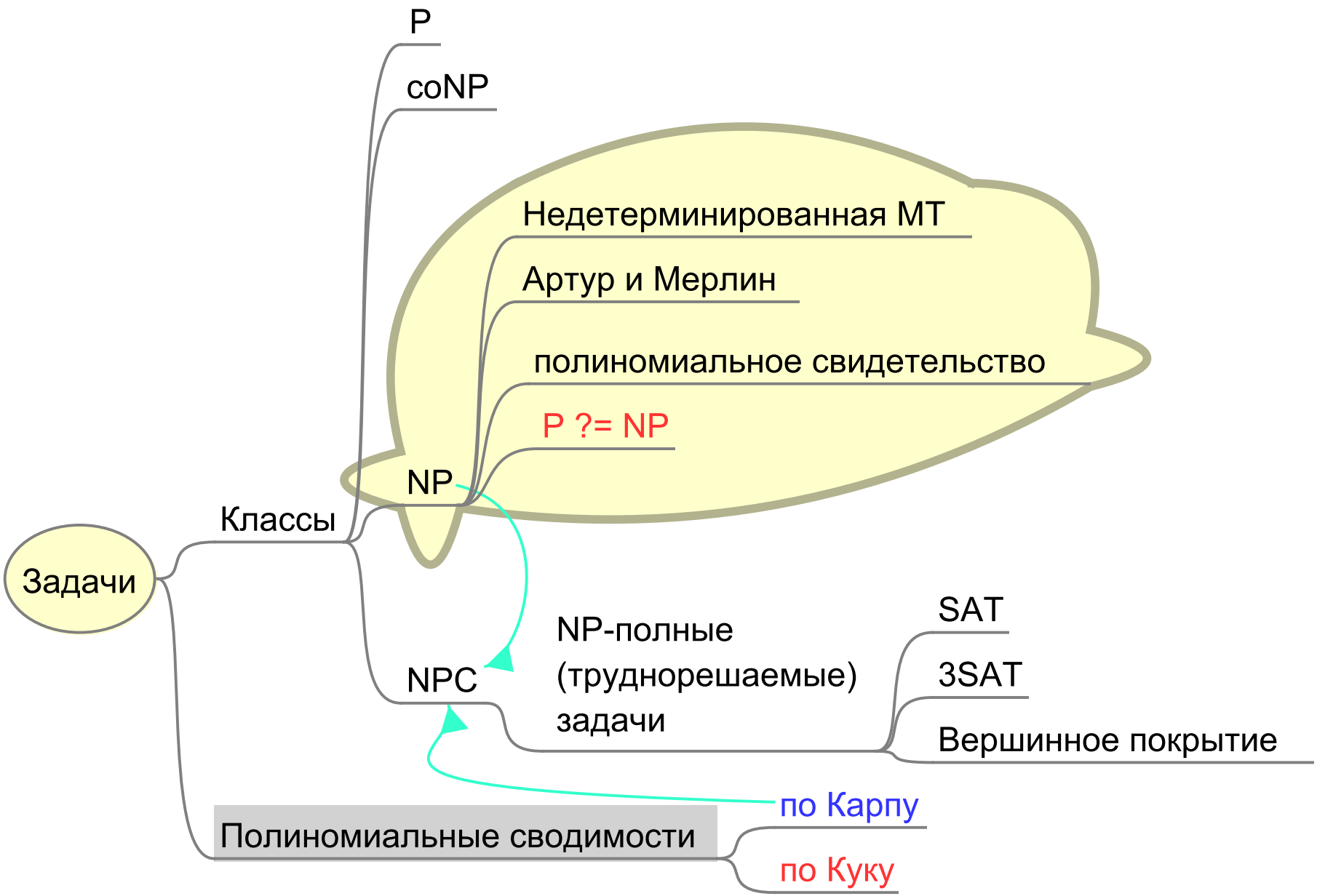
Если есть выполняющий набор для 3КНФ, то существует вершинное покрытие размера $(n + 2m)$: x_j если $x_j = 1$ (иначе $\neg x_j$), этим «покрываем» по одной вершине в каждом треугольнике. Чтобы покрыть остальные, нужно + две вершины, чтобы покрыть треугольник.

$\mathcal{P} \stackrel{?}{=} \mathcal{NP}$



«Экспертное мнение» считает, что $\mathcal{P} \neq \mathcal{NP}$.

«Карта памяти» лекции



Интернет поддержка курса

http://discopal.ispras.ru/

Вопросы?