Семинар 16

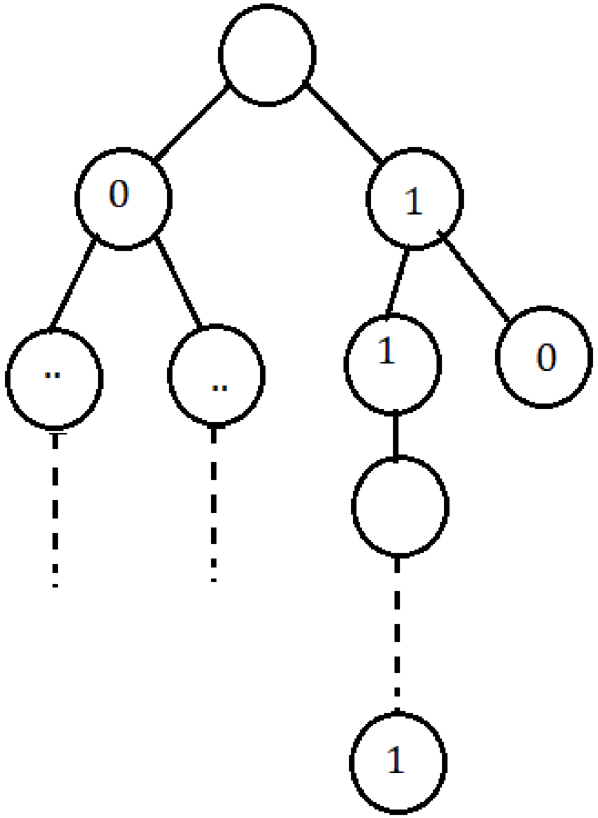
Класс языков NP. Сводимости.

NP

- Будем рассматривать задачи, для которых пока не придуманы эффективные алгоритмы. Среди этих задач рассмотрим такие, которые можно быстро проверить, если предложен вариант ответа (сертификат). Для таких задач используют не ДМТ (детерминированные машины Тьюринга), а НМТ (недетерминированные).
- Пусть поставлена некоторая задача распознавания (ответ ->{true, false}). Если x – исходные данные, $\mathrm{X} = |x|$ -длина входа. Сначала запускаем МТ в недетерминированном режиме. Она записывает случайные биты, из которых составится сертификат y. Затем НМТ выполняет вычисление в детерминированном режиме с полученным сертификатом.
- Рассмотрим языки типа L_A: (V – функция, вычисляемая МТ)

$\forall x \in L_A \Leftrightarrow \exists y\colon V(x, y) = true, \text{причем } |y| = X^n$ (полиномиально зависит от длины x и время работы на МТ, реализующей функцию $V, \text{является полиномиальным})$.

Схема работы НМТ



недетерминированный этап

детерминированный этап

Если одна из ветвей приведет к true(1) – значит слово принадлежит языку

Если длина сертификата и время работы полиномиально зависит от *X*, то такой язык принадлежит классу NP

- Например, рассмотрим следующую задачу: верно ли, что среди чисел некоторого множества М есть такие, что их сумма равна 0?

M={-2, -3, 15, 14, 7, -10, …} -2+(-3)+15+(-10)=0 легко проверить.

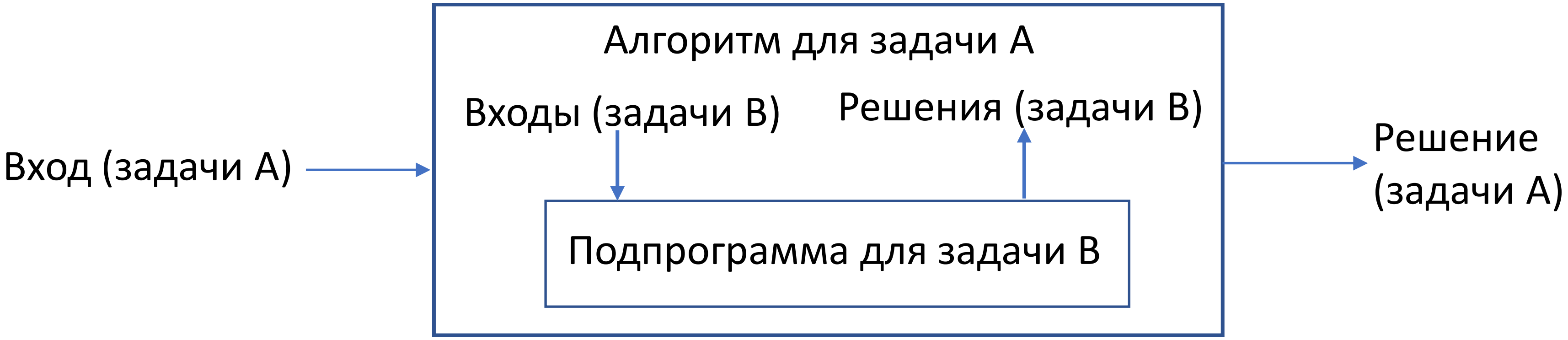
Сложность проверки не больше $n = |M|$ - полиномиальна. Но найти такие числа кроме как полным перебором пока не представляется возможным. Т.е. время очного решения $O(2^n)$ – экпоненциально, а время решения с сертификатом – полиномиально $\Rightarrow$ это задача из NP.

Полиномиальная сводимость

Язык L_A полиномиально сводится к языку L_B ($L_A \leqslant_P L_A$), если существует такая полиномиальная функция f, что $\forall x \ (x \in L_A \Leftrightarrow f(x) \in L_B)$.

В терминах задач:

Задача A полиномиально сводится к задаче B, если алгоритм, решающий задачу B, может быть легко переведен в алгоритм, решающий задачу A.



Полиномиальное число вызовов B и полиномиальный объем дополнительной работы

Примеры сводимости задач класса NP (см. семинар 15)

1. SAT $\leqslant_P$ SAT

Идея: любой дизъюнкт (клоз) $(a_1 \vee a_2 \vee \cdots \vee a_k)$

$(a_1 \vee a_2 \vee \cdots \vee a_k) = (a_1 \vee a_2 \vee y)\ (\overline{y} \vee a_3 \vee \cdots \vee a_k)$

y-новая переменная, значение которой можно подобрать $\Leftrightarrow$ разбиваемый дизъюнкт истинный.

$(a_1 \vee a_2 \vee y_1)\ (\overline{y_1} \vee a_3 \vee y_2)\ (\overline{y_2} \vee a_4 \vee y_3) \ldots (\overline{y_{k-3}} \vee a_{k-1} \vee a_k)$

Пример: $K = (p \vee q \vee \overline{r})(\overline{p} \vee q \vee r \vee \overline{s})(\overline{q} \vee \mathrm{s}) =$

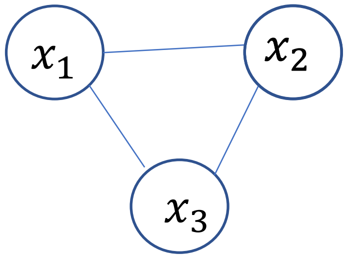
$=(p \vee q \vee \overline{r})\ (\overline{p} \vee q \vee y_1)(\overline{y_1} \vee r \vee \overline{s})(\overline{q} \vee \mathrm{s})$

Если некоторая переменная входит в КНФ больше, чем в 3 дизъюнкта, переобозначим ее столько раз, сколько вхождений. $\left(x \longrightarrow x_1, x_{2,\ldots}\right)$и добавим к КНФ $(\overline{x_1} \vee x_2)(\overline{x}_2 \vee x_3) \ldots (\overline{x}_k \vee x_1)$. Все преобразования линейно зависят от количества переменных.

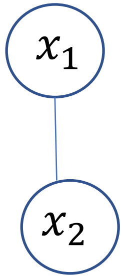
2. 3SAT $\leqslant_P$ INDSET

- дизъюнкт -> 2(3) вершины графа

$(x_1 \vee x_2 \vee x_3)$



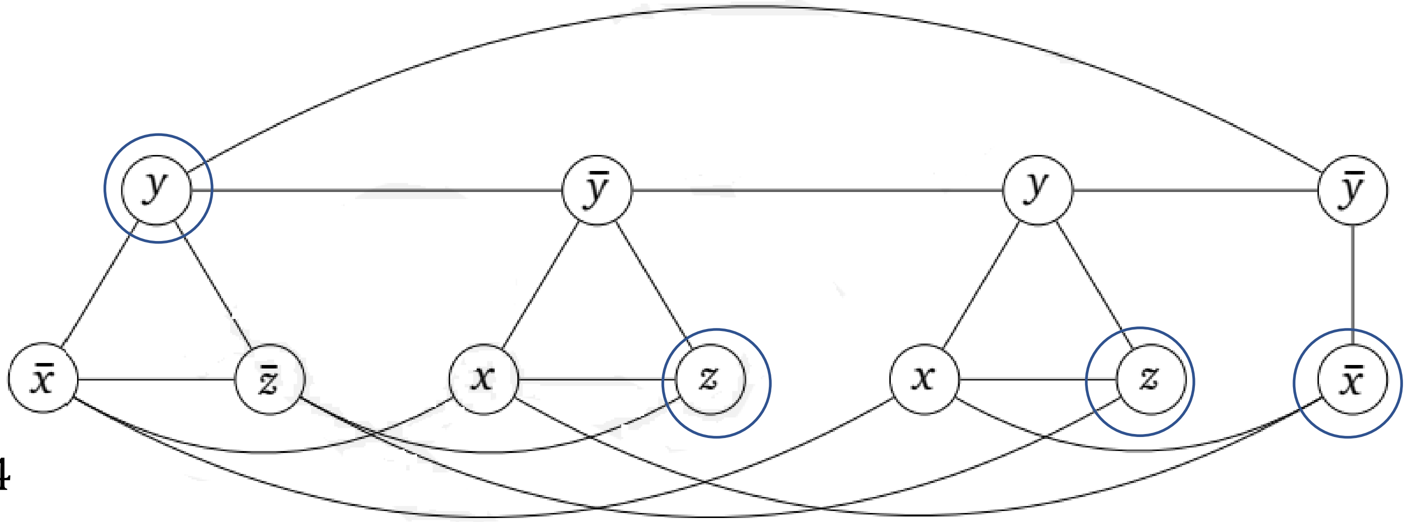
$(x_1 \vee x_2)$



Противоположные литералы соединим ребрами.

Пример:

$(\bar{x} \vee y \vee \bar{z})\,(x \vee \bar{y} \vee z)\,(x \vee y \vee z)\,(\bar{x} \vee \bar{y})$.



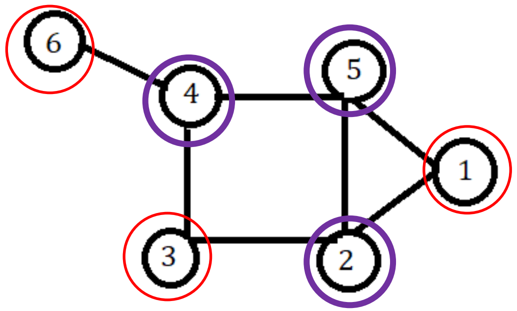
Найти независимое множество размера ≤ 4

0, 1, 1 – выполняющий набор $\leftrightarrow$ НЕЗАВИСИМОЕ МНОЖЕСТВО

3. INDSET $\leqslant_P$ VERTEX-COVER

(S -VERTEX-COVER -любое ребро графа инцидентно какой-либо вершине из S)
Пусть S – INDSET. Тогда V-S =VERTEX-COVER

Пример:

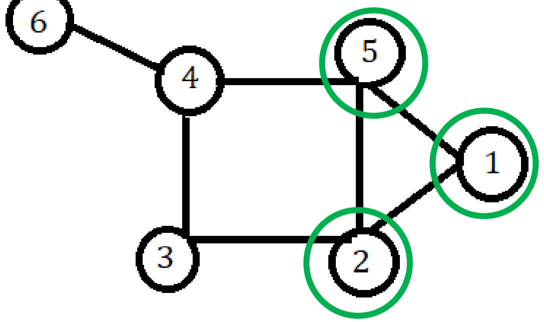


4. Клика ↔ INDSET,

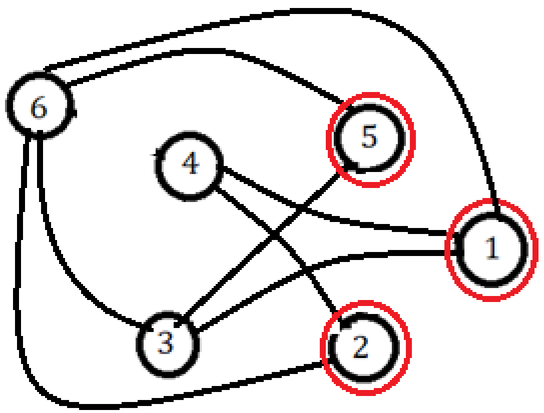
- $G = (V, E) \longrightarrow \overline{G} = (V, \overline{E})$, где $\overline{E}$ - ребра, отсутствующие в G.

Тогда, если S – клика размера k в G, то S – независимое множество размера k в $\overline{G}$.

Пример:



Клика размера 3 (1, 2, 5)



INDSET 1, 2, 5

Свойства сводимости

- Основной принцип: композиция полиномиально вычислимых функций полиномиально вычислима
- Свойство 1: если $A \leqslant_p B$ и $B \leqslant_p C$, то $A \leqslant_p C$
- Доказательство: $x \in A \Leftrightarrow f(x) \in B \Leftrightarrow g(f(x)) \in C$
- Свойство 2: если $A \leqslant_p B$ и $B \in \mathrm{P}$, то $A \in \mathrm{P}$
- Доказательство: $x \in A \Leftrightarrow f(x) \in B \Leftrightarrow M(f(x)) = 1$
- Свойство 3: если $A \leqslant_p B$ и $B \in \mathrm{NP}$, то $A \in \mathrm{NP}$
- Доказательство: $x \in A \Leftrightarrow f(x) \in B \Leftrightarrow \exists y \ V(f(x), y) = 1$

NP-трудность

Определение. Язык B называется NP-трудным, если $\forall A \in NP \;\; A \leqslant_P B$

Утверждение:

$\boxed{\text{Если какой-то NP-трудный язык B} \in P \text{, то } P = NP}$

NP-полнота

Определение. Язык B называется NP-полным (NPC), если он NP-трудный и B$\in NP$

Следствие: $\boxed{\text{если какой-то NP-полный язык B} \in P \text{, то } P = NP}$

Получение NP-трудности и NP-полноты

- Если B является NP-трудным и B $\leqslant_P$ C , то C – NP-трудный.

- Если B является NP-полным и B $\leqslant_P$ C , то C – NP-полный.

Для доказательства:

- доказать, что C $\in NP$
- свести к нему какой-либо известный NPC-язык