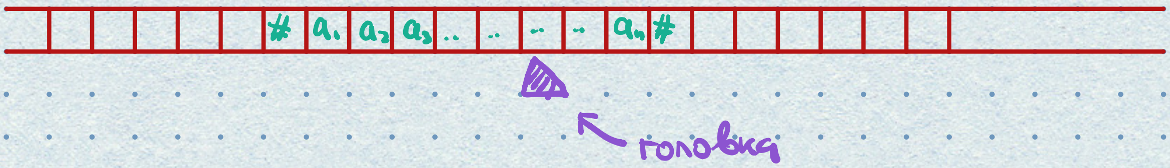
Машина Тьюринга:



1) Бесконечная лента
2) Головка - указатель на какую-то ячейку.
3) Нет произвольного доступа к памяти
4) Есть алфавит символов. В том числе пробельный (# или λ)
5) Есть множество состояний головки и правила переходов.
6) На каждом шаге принимается решение:
   1. Что написать в текущую ячейку.
   2. Куда сдвинуться: L, R, N ← остаться
   3. В какое состояние перейти.
7) головка перед первым символом.
8) Есть допускающие и отвергающие состояния.

Программа - функция: $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$

какое сейчас состояние — Q; какой сейчас символ — Γ; в какое состояние перейти — Q; какой символ написать — Γ; куда сдвинуться — $\{L, R, N\}$

функция перехода — δ

машина останавливается, если она пришла в заверш. состояние: q_f

Конфигурация: состояние + то, что написано на ленте + положение головки

Пример: $\Sigma = \{0,1\}$.   010110

хотим функцию, которая определяет четность длины входа:

$q_0\# \to q_0\#R$ — плохо, если пустой вход. лучше: $q_0\# \to q_2\#R$

$q_0 1 \to q_1\#R$

$\to q_1 1 R$

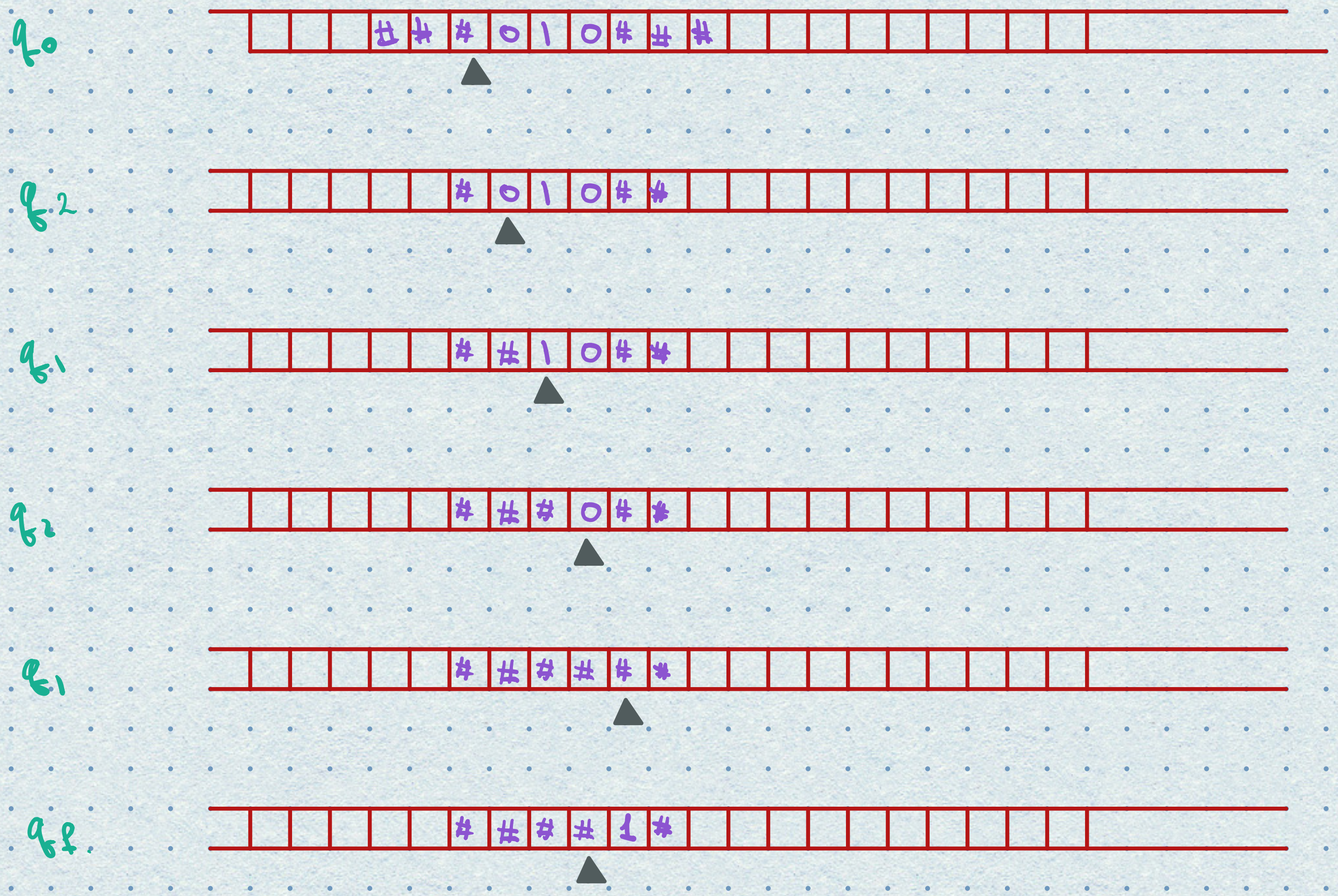
$q_0 0 \to q_1\#R$

$q_1 1 \to q_2\#R$ $q_2 0 \to q_1\#R$

$q_1 0 \to q_2\#R$ $q_2 1 \to q_1\#R$

$q_1\# \to q_f 1 L$ ← пишем 1 (нечет.) сдвигаем головку перед началом ответа

$q_2\# \to q_f 0 L$ ← пишем 0 - четное. — " —



ответ: 1.

цифры с числа: 1 → 1; 2 → 11; 3 → 111 .... n → $\underbrace{111...111}_{n}$

реализовать: +1; ×2 (понадобится доп символ).

Проблема остановки: дана конфиг. машины Тьюринга, остановится ли она?

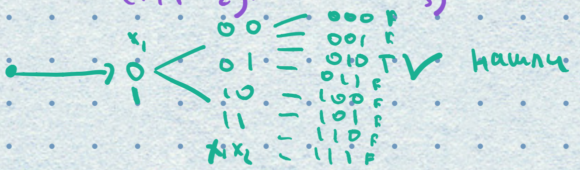
Недетерминированная машина Тьюринга

из каждого состояния есть несколько переходов по одному символу.

Как? или машина всегда угадывает нужный переход или делится на много копий. Если хоть одна копия допускает слово, то машина его допускает.

Можно повторил НМТ на МТ (решаем просто перебором)

$!x_1 \& (x_1 | x_2) \& (!x_2 | !x_3)$



Теперь не решаем задачи, а про ерем слова на принадлежность языку.

1. Принадлежит ли граф языку двудольных графов.

Сложностные классы

- T(p, x) — время работы программы p на входе x
- S(p, x) — объём памяти для работы программы p на входе x
- DTIME(f(n)) — класс языков L, для которых существует программа p такая, что T(p, x) = O(f(n)) для любого x∈L, |x| = n
- DSPACE(g(n)) — то же про память
- TS(f, g): T(p, x) = O(f(n)), S(p, x) = O(g(n))

Класс P.

класс языков разрешимых на МТ за P время.

$$P = \cup DTIME(p(n))$$

1. Отсортированность массива
2. Связность графа

...

Сведение по Карп

Сведе́ние по Карпу

Язык L_1 **сводится по Карпу** к языку L_2 ($L_1 \leqslant L_2$), если $x \in L_1 \Leftrightarrow f(x) \in L_2$ для некоторой f, работающей за полином

- Проверка отсортированности по убыванию сводится к проверке отсортированности по возрастанию
- Поиск подстроки в строке сводится к проверке максимума z-функции
- Поиск клики размера k сводится к поиску независимого множества размера k, но в дополнении графа

Хотим, зная решение одной задачи, решать другие задачи не сильно теряя в производит (полиномиально).

Свойства класса P

P замкнут относительно:

- сведения по Карпу: $L \in P$, $M \leqslant L \Rightarrow M \in P$;
- объединения языков;
- пересечения;
- дополнения;
- конкатенации;
- замыкания Клини (L^*)

Класс NP:

Класс языков разрешимых на $\boxed{\text{НМТ}}$ за полином. время

$$P = \cup \, NTIME(p(n))$$

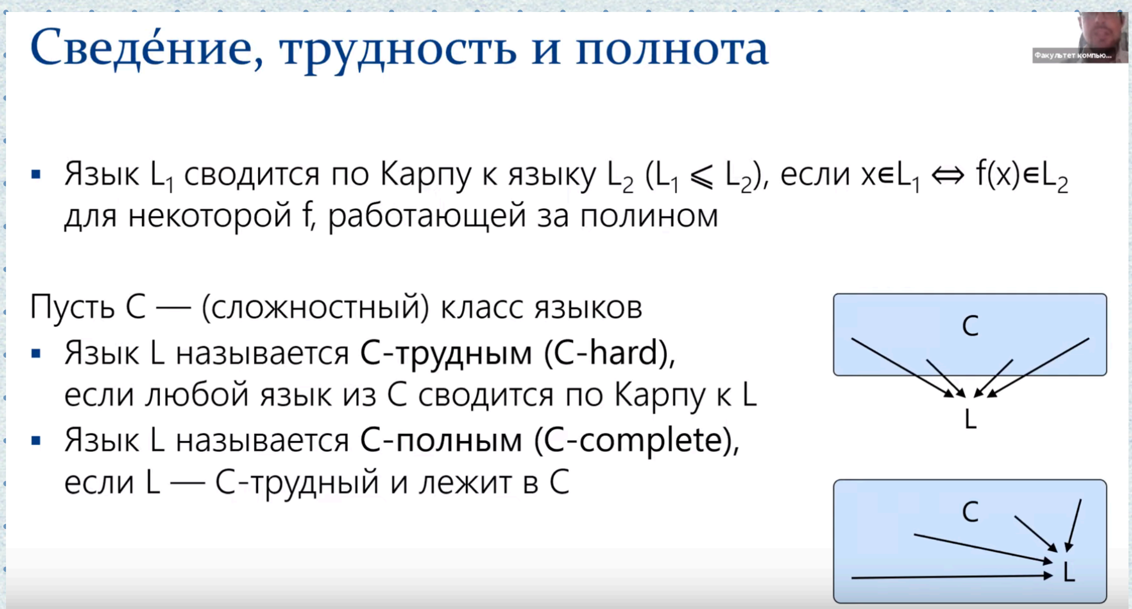
$P \subseteq NP$

Пример: SAT, VERTEX COVER

Класс из NP если для сертификата сущ. полиномиальный алгоритм проверки сертификата.

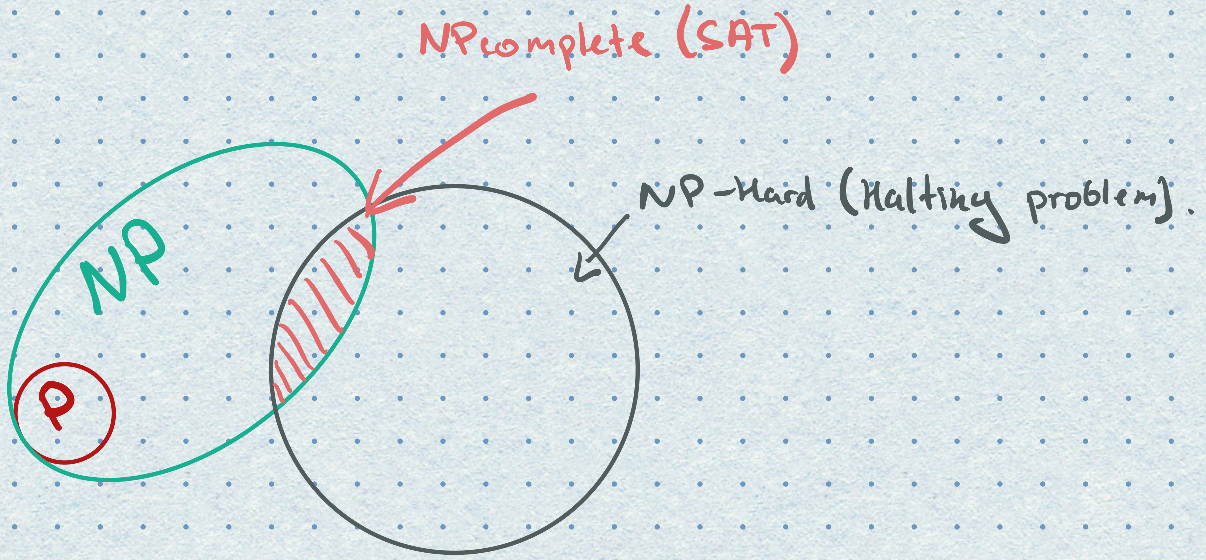
ответ на задачу.

Задача SAT, сертификат — набор значений переменных. легко проверить за полином.



Любой язык в P — p-complete

$P \overset{?}{=} NP$

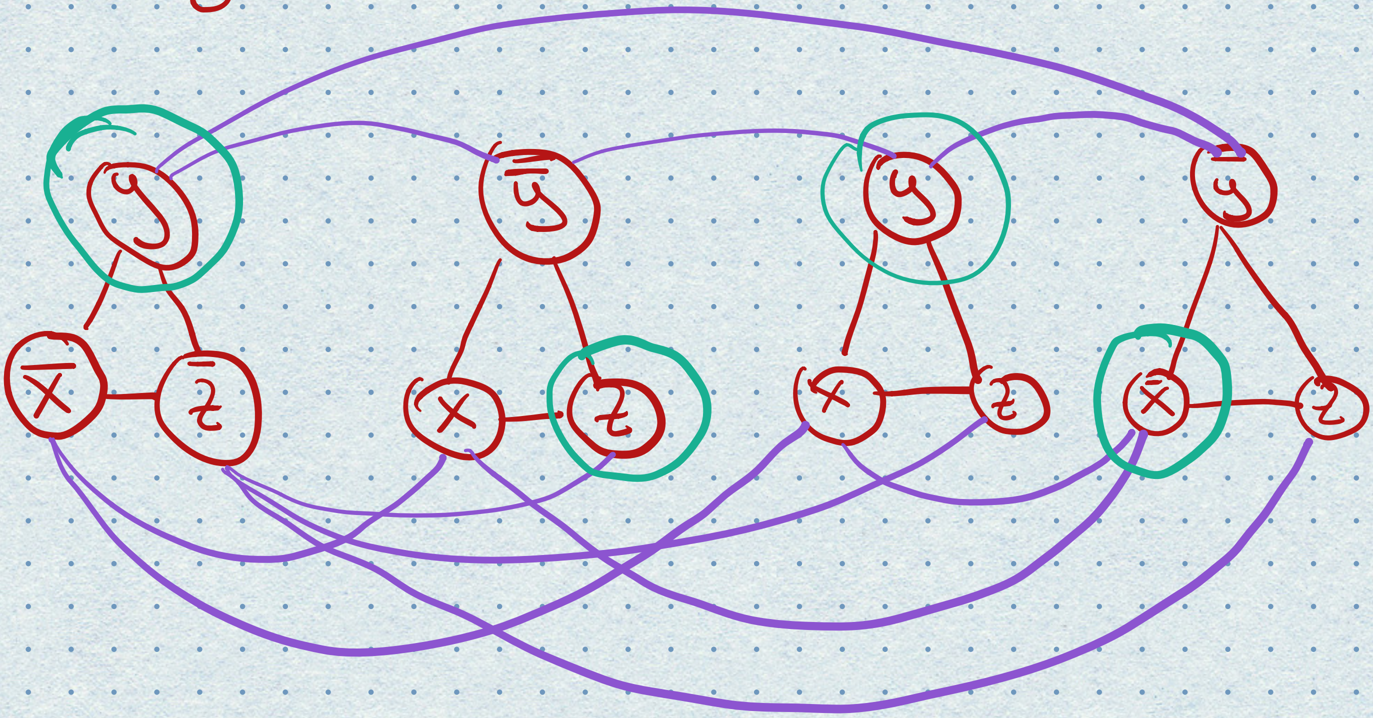


3SAT INDEPEND. SET

$(a_1 \vee b_1 \vee c_1) \wedge (a_1 \vee b_2 \vee c_2) \dots (a_n \vee b_n \vee c_n)$

a_1 может быть отрицанием

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



$y=1$

$x=0$

$z=1$

Сведение полиномиально.

Если есть решение для 3SAT, то и для IS

тоже есть решение.

3SAT – 3COL