2 Теория на сложността

Колко трудни са проблемите, които разрешаваме?
Как някои машинни модели позволяват по-бързи решения от други?

Пример: Даден е DFA (NFA) A. Вярно ли е, че $L(A) = \Sigma^*$?

Терминология и означения

n: означава големината на входа

Мярката - битове, символи на лентата- трябва да се уточнят.

(разрешаваща процедура) ”проблем” : да се разпознае един език.

Говорим само за разрешими проблеми.

Мерки за сложност

- Time. Времева мярка за сложност - времето необходимо за изчисление. DTIME, NTIME /Ние главно тази мярка ще разгледаме /
- Space -Пространствена мярка за сложност - паметта, необходима за изчисление. DSPACE, NSPACE.
- ...

Примери

$L = \{0^n 1^n \mid n \geq 0\}$. Въпрос: $w \in L$?

M_1- МТ с една лента: вход $w \in \Sigma^*$

- Сканираме надясно и reject, ако има 0 след 1;
- Итерираме, докато на лентата има и 0 и 1;
  - Сканираме и заместваме по една 0 и една 1 с X;
- Ако има 0 и няма 1 или обратно reject, иначе accept.

Да анализираме този алгоритъм-колко време е необходимо в зависимост от дължината на входа.

$O(g(n))$

Деф. Времева сложност на една детерминистична МТ M е функция $f : N \to N$, $f(n)$ е максималният брой стъпки над вход с дължина n.

Деф. Ако $f, g : N \to R^+$, то казваме, че $f(n) = O(g(n))$, ако $\exists n_0 \geq 0$ такова че $\forall n \geq n_0 (f(n) \leq cg(n))$ за някоя константа c.

Например $f(n) = O(n^2)$, $f(n) = O(2^n)$

За M_1: $f(n) = O(n) + O(n^2) + O(n) = O(n^2)$.

$TIME(t(n)) = \{L \mid L$ е език, разпознаван с $O(t(n)) - MT\}$.
$(t : N \to N)$

Примери

$L = \{0^n 1^n \mid n \geq 0\}$. Въпрос: $w \in L$?

M_2- МТ с една лента: вход $w \in \Sigma^*$

- Сканираме надясно и reject, ако има 0 след 1;
- Итерираме, докато на лентата има и 0 и 1
  - Сканираме по лентата, проверявайки дали общият брой на 0 и 1 е четен. Ако е нечетен reject;
  - Сканираме и заместваме всяка нечетна 0 и всяка нечетна 1 с X;
- Ако няма 0 и 1 на лентата accept.

Примери

$L = \{0^n 1^n \mid n \geq 0\}$. $L \in TINE(n \log_2(n))$

M_2- $2O(n) + (1 + \log_2(n))O(n) = O(n \log_2(n))$.

- На всяка стъпка 4. намаляваме броя на 0-те на 2, както и на 1-те, т.е. делим целочислено на 2.
- Напр от 11 нули (нечетен брой)- остават 5 (нечетен брой)-после 2 (четен)- после 1 (нечетен).
- Сега гледаме остатъците при делене на 2 - 1101, но 1011- в обратен ред е точно 11 в двоичен вид.
- На всяка стъпка 3. реално проверяваме дали 0-те и 1-те дават един и същи остатък при делене на 2., т.е. дали са равен брой.

Примери

$L = \{0^n 1^n \mid n \geq 0\}$.

M_3- МТ с две ленти: $O(n)$.

- Сканираме надясно и reject, ако има 0 след 1;
- Копираме 0-те от първата на втората лента до първата 1;
- Сканираме 1-те на първата лента до края на думата и за всяка 1 заместваме по една 0 с X на втората лента;. Ако всички 0 са заместени преди всички 1 да са прочетени reject;
- Ако всички 0 са заместени accept, иначе reject.

Зависимост от модела на изчисление

Факт 1. Нека t е функция и $t(n) \geq n$. Тогава за всяка многолентова МТ с времева сложност $t(n)$ има еквивалентна МТ с една лента с времева сложност $O(t^2(n))$.

Деф. Времева сложност на една недетерминистична МТ M (разрешаваща- yes/no) е функция $f: N \rightarrow N$, $f(n)$ е максималният брой стъпки, която M прави по време на изпълнението си по всеки клон над вход с дължина n.

Факт 2. Нека t е функция и $t(n) \geq n$. Тогава за всяка недетерминистична МТ с времева сложност $t(n)$ има еквивалентна детерминистична МТ с времева сложност $2^{O(t(n))}$. $(= O(t(n)b^{t(n)})$, b е мах брой преходи на δ на МТ.)

Класът P

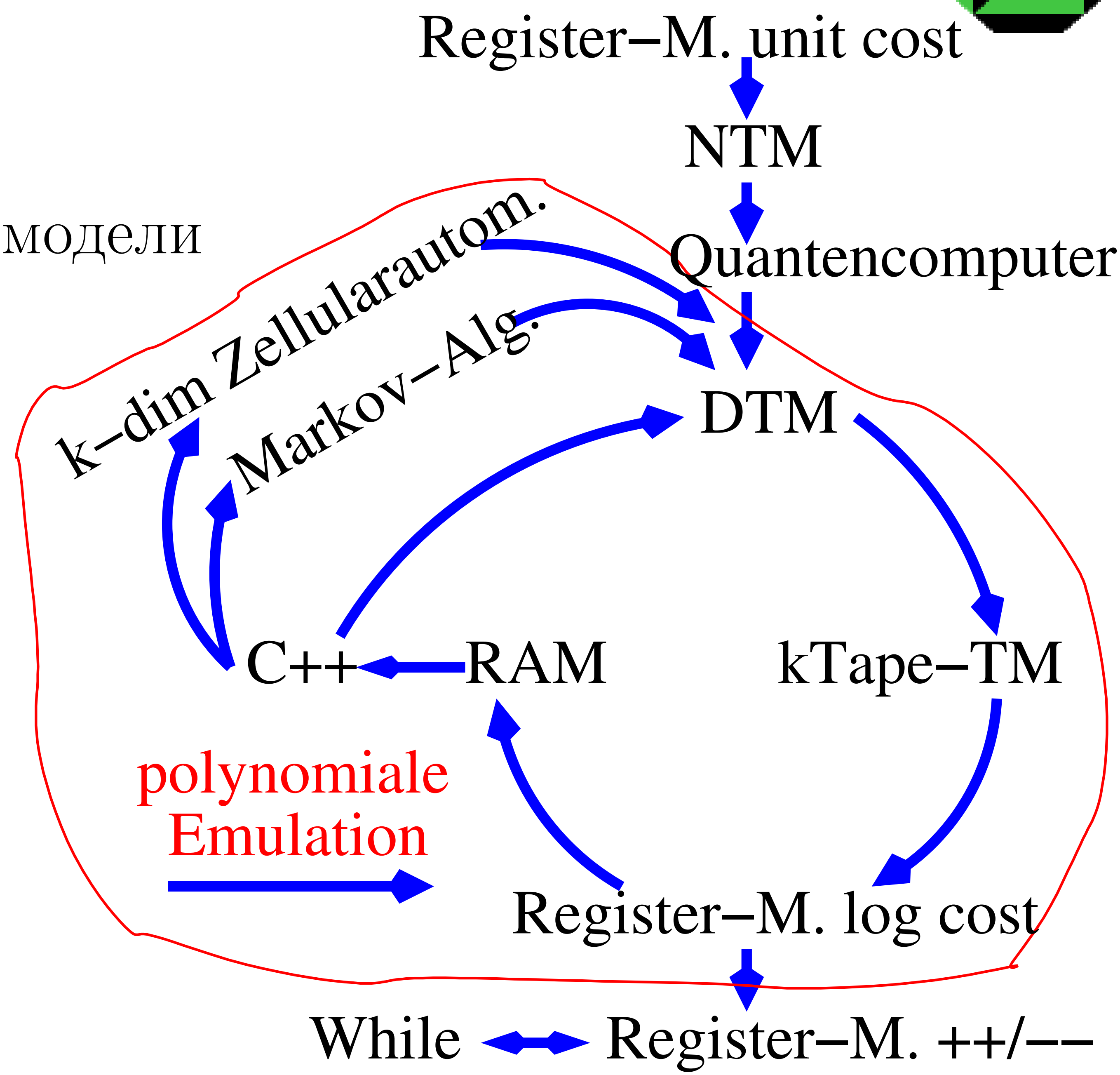
$$\mathrm{P} := \bigcup_k \mathrm{TIME}(n^k)$$

Всички езици разрешими за полиномиално време на детерминистична машина на Тюринг за полиномиално време.

1. P е инвариантен относно всички модели на изчисление, които са полиномиално еквивалентни на някоя детерминистична МТ с 1 лента.

2. P кореспондира със задачите, които са реалистично изчислими с компютър.

P за различни машинни модели



Примери

1. Разумно кодиране на проблема- което позволява полиномиално време за кодиране и декодиране на обектите.

Например унарното кодиране не е разумно- екпоненциално относно разумно кодиране в бройна система с основа k за всяко $k \geq 2$.

Примери $\mathit{PATH} \in \mathrm{P}$

$\mathit{PATH} = \{\langle G, s, t\rangle \mid G$ ориентиран граф, с път от s до $t\}$.

Ориентиран граф $G = (V, E)$

Множество от върхове V ("възли"')

Множество от дъги $E \subseteq V \times V$ ("ребра"')

Път $p = \langle v_0, \ldots, v_k\rangle$

с дължина k свързва върховете v_0 и v_k, ако

в p последователните върхове са свързани с дъги от E,

т.е.,

$e_1 = (v_0, v_1) \in E$, $e_2 = (v_1, v_2) \in E$, $\ldots$, $e_k = (v_{k-1}, v_k) \in E$.

Прост път: $v_0, \ldots, v_k$ са различни.

Представяне на граф

- Матрица на съседство (Adjacency matrix)
- Списъци на съседство (Adjacency arrays)

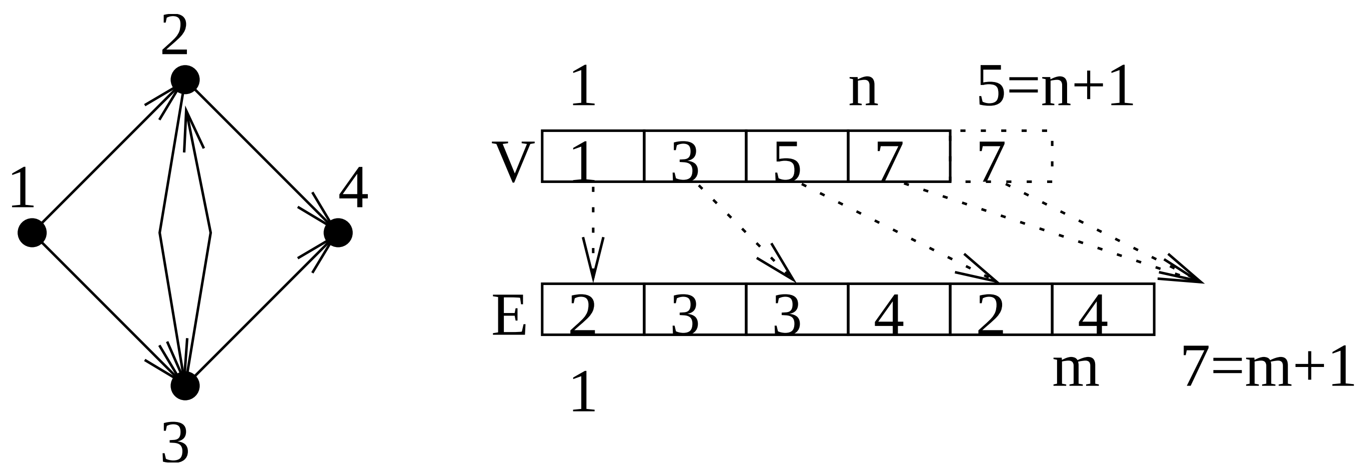
Adjacency arrays

$V = 1..n$

Списък на възлите E пази всички непосредствени наследници на всеки възел (групирани)

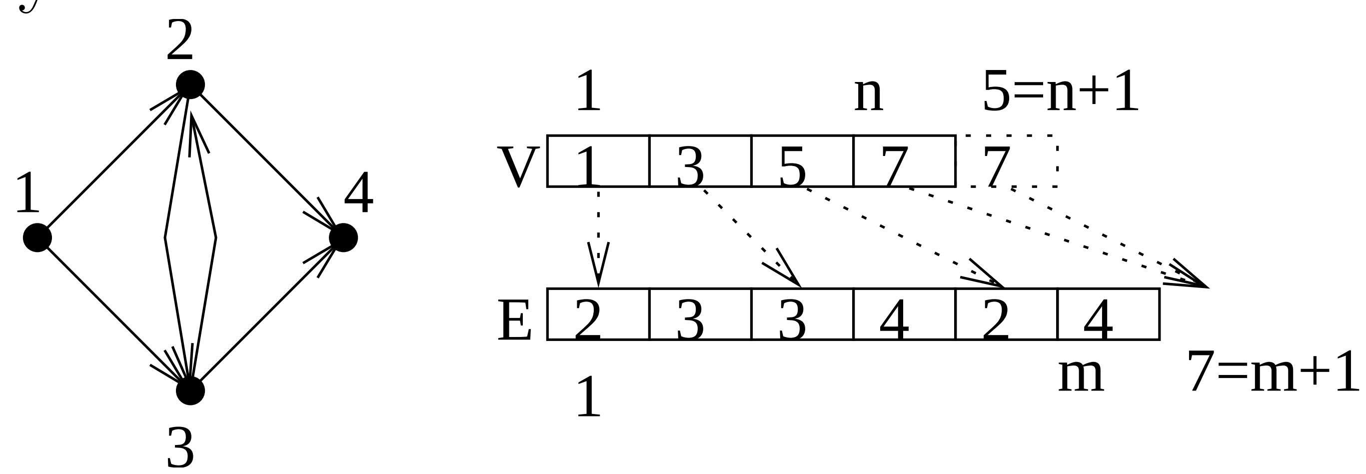
Списък на съседство V пази индекса (позицията) на първата дъга в E от даден връх

Псевдовръх $V[n+1]$ пази $m+1$



Пример: Брой на наследниците $(v) = \mathtt{V}[v+1] - \mathtt{V}[v]$

Adjacency arrays



- Памет $\mathcal{O}(m+n)$
- Бърза навигация (достъп). $\mathcal{O}(1)$.
- Допълнителна информация $\rightsquigarrow$ допълнителни списъци
- За статични графи (непроменящи се)

Но добавяне на нов възел е проблем.

Достижимост Depth First Search

//mark all reachable from s nodes

Procedure reachable(s)

    mark s

    foreach $(s, v) \in E$ do

        if v is not marked then reachable(v)

Реализация с adjacency arrays

$V[1..n+1]$ // Input
$E[1..m]$ // Graph
$\text{mark}[1..n] = \langle 0, \ldots, 0 \rangle$
Function reachable(s)
  $\text{mark}[s] := 1$
  for $e := V[s]$ to $V[s+1]-1$ do // for each $e = (s,v) \in E$
    $v := E[e]$
    if $\text{mark}[v] = 0$ then reachable(v)

Време $\mathcal{O}(m+n)$

Примери $\mathit{RelPrime} \in \mathrm{P}$

$\mathit{RelPrime} = \{\langle x,y\rangle \mid (x,y) = 1\}$.

Алгоритъм на Евклид:

Input:$\langle x,y\rangle$ // Input-binary

Function gcd(x,y)

    Repeat until $y = 0$

        $x := x \bmod y$

        swap(x,y)

        Output x

Function R(x,y) // $\mathcal{O}(n)$

if gcd$(x,y) = 1$ then accept

else reject

Трансформиране на оптимизационен проблем $\longrightarrow$ в разрешаващ проблем

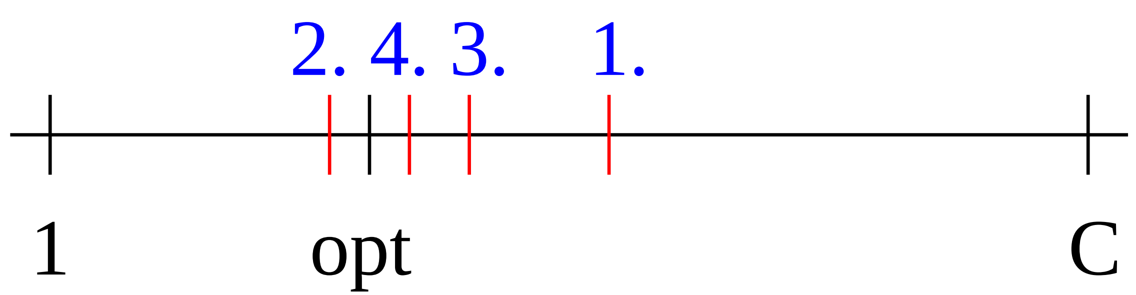
Нека: Оптимизационната функция $\in 1..C$, цели числа.

Двоично търсене $\lceil \log C \rceil$. Разрешаващият проблем е решен.

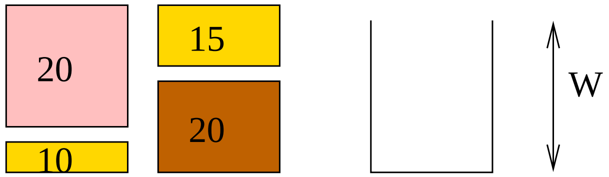
Полиномиално в n, ако $\log C$ полиномиално в n.

Ако C има полиномиално много битове.

Това е цената на output complexity!



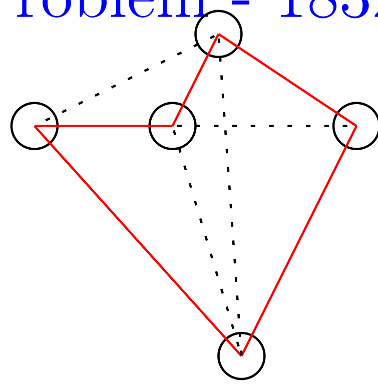
Пример- Проблемът за раниците



- Дадени n обекта с тегла $w_i \in \mathbb{N}$ и печалба p_i
- Има ли подмножество $\mathbf{x}$ на обектите
- така, че $\sum_{i \in \mathbf{x}} w_i \leq W$ и
- да се максимизира печалбата $\sum_{i \in \mathbf{x}} p_i$

Пример: Търговският пътник (TSP)

[Travelling Salesman Problem - 1832].



Даден е граф $G = (V, V \times V)$, да се намери прост цикъл (с различни възли) $C = (v_1, v_2, \ldots, v_n, v_1)$ така, че $n = |V|$ и $\sum_{(u,v) \in C} d(u,v)$ да е минимална.

Формулирайте разрешаващия проблем.

Хамилтонов цикъл

[Уилям Хамилтон, Memorandum respecting a new system of roots of unity. Philosophical Magazine, 12 1856]

$M := \{G = (V,E) : \exists C \subseteq E : |C| = |V|, C$ е прост цикъл$\}$

Кодираме графа $G = (V,E)$ като дума от $\{0,1,\#\}^*$:

$V \subseteq \{0,1\}^{\lceil \log |V| \rceil}$

Кодиране $w_G := \prod_{(u,v) \in E} u\#v\#$

Покритие (Set Cover)

Дадено: Пример:

Основно множество: M $\{1,2,3,4,5\}$

Система от подмножества $\mathscr{T} \subseteq 2^M$ $\{\{1,3\},\{1,2\},\{2,4,5\}\}$

Параметър n 2

Въпрос:

Има ли елементи $T_1 \in \mathscr{T}, \ldots, T_n \in \mathscr{T}$? Да:

$T_1 \cup \cdots \cup T_n = M$? $\{1,3\},\{2,4,5\}$

Клика (CLIQUE)

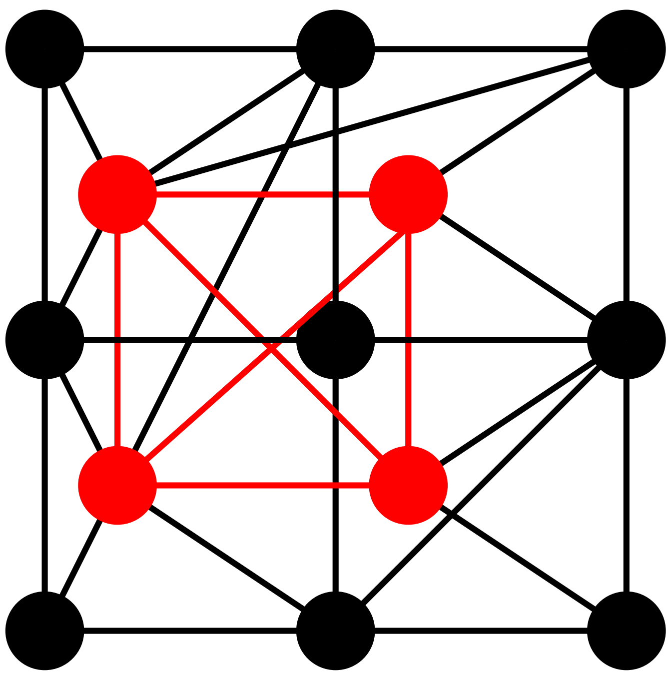
Дадено: Неориентиран граф $G = (V, E)$,

Параметър $k \in \mathbb{N}$.

Въпрос:

Дали G съдържа клика с размерност k?

т.е. $\exists V' \subseteq V : |V'| = k \wedge \forall u \neq v \in V' : \{u, v\} \in E$



Друг клас на сложност

Нека M е недетерминистична машина на Тюринг.

$$\mathrm{ntime}_M(w) := \begin{cases} \min\{|P| : P = (s)w \vdash^* u(f)v, f \in F\} & \text{ако } w \in L(M) \\ 0 & \text{иначе} \end{cases}$$

$NTIME(t(n)) = \{L \mid (\exists NTM\ M)(L(M) = L \wedge (\forall w \in \Sigma_M^* \wedge |w| = n)(ntime_M(w) \leq \mathscr{O}(t(n))))\}$.

$NTIME(t(n)) = \{L \mid L$ е език, разпознаван с $\mathscr{O}(t(n))$ недетерминистична МТ$\}$.

Класът **NP**

$$\mathbf{NP} := \bigcup_k \mathrm{NTIME}(n^k)$$

Класът **NP** се състои от всички езици, които се разпознават с НМТ за полиномиално време.

Пример: Проблемът за раниците (Knapsack problem)

//Is there $x_1 \cdots x_n \in \{0,1\}^n : \sum_i x_i w_i \leq W \wedge \sum_i x_i p_i \geq P$?

Procedure knapsack($\langle w_1, \ldots, w_n \rangle, \langle p_1, \ldots, p_n \rangle, W, P$)

for $i := 1$ to n do nondeterministically guess $x_i \in \{0,1\}$

if $\sum_i x_i w_i > W$ then reject

if $\sum_i x_i p_i < P$ then reject

accept

Knapsack$\in NP$

Класът **NP**

Верификатор за езика L е МТ V със свойството:

$L = \{w \mid V$ приема $\langle w, c\rangle$ за някое $c \in \Sigma^*\}$.

L е полиномиално верифицируем, ако има полиномиален верификатор.

c се нарича сертификат.

Класът NP

Твърдение: $L \in \mathbf{NP} \Leftrightarrow L$ има полиномиален верификатор.

Ако L се разпознава от полиномиална недетрминистична МТ N. Построяваме верификатор V:

V: Input $\langle w, c \rangle$

1. Симулира изпълнението на N над вход w, третирайки всеки символ на $c \in \Sigma^*$ като описание на недетрминистичния избор за всяка стъпка.

2. Ако този клон на изчислението на N приема w, то accept, иначе reject.

Класът **NP**

Твърдение: $L \in \mathbf{NP} \Leftrightarrow L$ има полиномиален верификатор.

Ако L има верификатор V с време n^k. Построяваме МТ N:

N: Input w $|w| = n$

1. Недетрминистично избира $c \in \Sigma^*$.
2. Изпълнява V над $\langle w, c \rangle$.
3. Ако V я приема, то accept, иначе reject.

Класът **NP**

Твърдение: $CLIQUE \in \mathbf{NP}$

$CLIQUE$ има полиномиален верификатор. Всяка клика в графа е сертификат.

Построяваме верификатор V:

V: Input $\langle\langle G,k\rangle,c\rangle$

1. Тества дали c е множество от k върха.
2. Тества дали G съдържа всички ребра свързващи всеки два върха от c .
3. Ако да, accept, иначе reject.

Въпрос за 1 000 000 $

$$\mathrm{P} = \mathbf{NP}?$$

Един от 7-те математически проблема, за които Clay Mathematics Institute дава награда за 1 000 000 US$.

Наблюдение: $\mathrm{P} \subseteq \mathbf{NP}$

Анкета

100 изследователи са питани дали $\mathrm{P} = \mathbf{NP}$

61: Не

09: Да

22: Не знам

08: без отговор.

Полиномиална сводимост

Нека $A \subseteq \Sigma^*$ и $B \subseteq \Gamma^*$ са езици.

$A \leq_p B$ (A е полиномиално сводимо към B)

$\Leftrightarrow$

$\exists f : \Sigma^* \to \Gamma^* : \forall w \in \Sigma^* : w \in A \Leftrightarrow f(w) \in B$

където f е изчислима за полиномиално време.

Пример

Теорема: Хамилтонов цикъл$\leq_p$TSP

Д-во:

Нека $G = (V,E)$ е неориентиран граф с n върха и $\alpha > 0$.

Дефинираме $d(u,v) = \begin{cases} 1 & \text{if } (u,v) \in E \\ 1+\alpha & else \end{cases}$

Тогава, ако G има Хамилтонов цикъл, то

$\exists$ TSP тур с тегло n

(иначе оптималното тегло $\geq n+\alpha$)

Нека $G = (V,E)$ е произволен неориентиран граф.

$$d(u,v) = \begin{cases} 1 & \text{if } (u,v) \in E \\ 1+\alpha & else \end{cases}$$

$\exists$ тур C с тегло n

$\longrightarrow$ няма ребра в C с тегло > 1

$\longrightarrow$ всички ребра в C имат тегло 1

$\longrightarrow$ всички ребра в C са дъги в G

$\longrightarrow$ C е Хамилтонов цикъл в G

Нека $G = (V, E)$ е произволен неориентиран граф.

$$d(u,v) = \begin{cases} 1 & \text{if } (u,v) \in E \\ 1+\alpha & \mathit{else} \end{cases}$$

G има Хамилтонов цикъл C

$\longrightarrow C$ е тур с тегло n.

Лема: $A \leq_p B, B \in \mathrm{P} \rightarrow A \in \mathrm{P}$

Д-во:

Нека $A \leq_p B$ с функцията f.

Нека M_f е МТ, такава, че f е изчислима с полиномиална граница p.

Нека освен това $B \in \mathrm{P}$ с полинимиална граница q с МТ M_B.

Да разгледаме композицията $M_A := (M_f; M_B)$.

M_A разрешава A.

Времето за изпълнение над вход w:

$p(|w|) + q(|f(w)|) \leq p(|w|) + q(|w| + p(|w|))$

Полиномиално в $|w| = n$

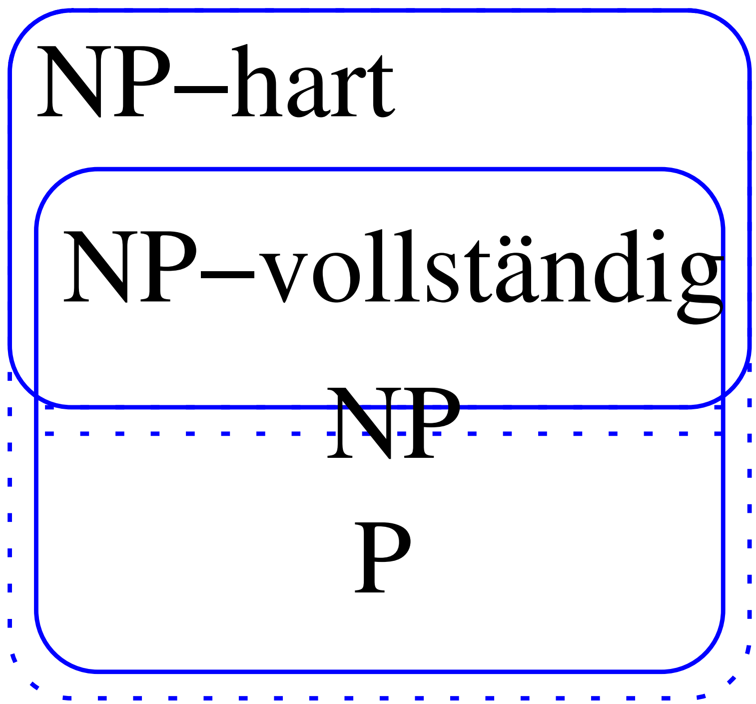
Лема: $A \leq_p B, B \in \mathbf{NP} \rightarrow A \in \mathbf{NP}$

Д-во: Аналогично.

NP-трудни и **NP**-пълни задачи

A е **NP**-трудна:$\Leftrightarrow \forall L \in \mathbf{NP} : L \leq_p A$

A е **NP**-пълна:$\Leftrightarrow A$ е **NP**-трудна и $A \in \mathbf{NP}$.



Прост начин към Известност и Богатство?

Твърдение: Нека A е **NP**-пълна. Тогава: $A \in \mathrm{P} \Leftrightarrow \mathrm{P} = \mathbf{NP}$

Д-во:

Случай $\mathrm{P} = \mathbf{NP}$:

$A \in \mathbf{NP} = \mathrm{P}$ и следователно $A \in \mathrm{P}$

Случай $A \in \mathrm{P}$:

Нека $L \in \mathbf{NP}$ е произволен.

Тъй като A е **NP**-трудна, то $L \leq_p A$

и тъй като $A \in P$ следва, че $L \in \mathrm{P}$

Също $\mathrm{P} = \mathbf{NP}$

SAT: Проблемът за изпълнимост

Дадено: Формула F в съждителната логика

$(\wedge \vee \neg \rightarrow ()$, Variables)

Въпрос: Дали F изпълнима?, т.е.

$\exists$ остойностяване на променливите (оценка)в $\{0,1\}$

така, че истинната стойност на $(F) = 1$?

Формално:

SAT := $\{\mathrm{code}(F) \in \Sigma^* : F$ е изпълнима формула$\}$,

code е подходящо кодиране на формулите като изрази.

Теорема: [Cook 1971] и [Levin 1971] SAT е **NP**-пълна задача.

Д-во на SAT$\in$ **NP**

Procedure satisfiability(F)

guess стойности $\in \{0,1\}$ за всяка променлива на F

замести променливите с техните стойности

пресметни резултатната истинна стойност на v от F

if $v = 1$ then accept

Д-во на: SAT е **NP**-трудна.

Нека

$L \in \mathbf{NP}$ произволен,

$M = (\{1,\ldots,k\}, \Sigma, \{1,\ldots,\ell\}, \delta, 1, H)$ НМТ, където

$L(M) = L$, $(\sqcup = \ell)$

$p(n)$ полиномиална горна граница за разрешаването на L

с M за, $w = w_1 w_2 \cdots w_n \in \Sigma^*$ произволен вход.

Основен метод: Показваме, че $L \leq_p$SAT,

като конструираме формула F с полиномиална

дължина, такава че

$$w \in L \Leftrightarrow F \text{ е изпълнима}$$

Променливи на F

- $\mathrm{state}_{tz} = 1 \Leftrightarrow$ след t стъпки M е в състояние z
- $\mathrm{pos}_{ti} = 1 \Leftrightarrow$ след t стъпки главата на M е в позиция на лентата i
- $\mathrm{band}_{tia} = 1 \Leftrightarrow$ след t стъпки позицията не лентата i е със съдържание a

Забележете, че

$$0 \leq t \leq p(n) \text{ и}$$
$$-p(n) \leq i \leq p(n) + 1$$

Има само $\mathcal{O}\left(p(n)^2\right)$ променливи.

Структурата на $F =$

R "‘Формула описваща: МТ винаги е в точно една конфигурация"’$\wedge$

A "‘Формула описваща: в началото МТ е в началната конфигурация$(1)w$"’$\wedge$

U_1 "‘В позицията на главата съответните модификации са както посочва δ"’$\wedge$

U_2 "‘В останалите позиции нищо не се променя"’$\wedge$

E "‘M приема w за $\leq p(n)$ стъпки"’

Д-во $w \in L \longrightarrow F$ е изпълнима

$w \in L$

$\longrightarrow \exists$ редица от конфигурации $P = (1)w \vdash^* u(h)v$ с $h \in H$ за M, $|P| = p(n)$

P дефинира остойностяване на променливите на F, така че F да е 1.

$\longrightarrow F$ е изпълнима.

Д-во F е изпълнима $\longrightarrow w \in L$

F изпълнима с вярностни стойности на променливите $\longrightarrow$

- R ще е вярна $\longrightarrow$
  точно една конфигурация ще е в сила
- A ще е вярна $\longrightarrow$
  Формулата описваща началната конфигурация $(1)w$
- U_1, U_2 ще са верни $\longrightarrow$
  от t до $t+1$ има преход съобразно δ
- E е вярна $\longrightarrow$
  изчислението завършва успешно в едно заключително състояние.

$\longrightarrow w \in L$

Може да е в точно една конфигурация

$G(v_1,\ldots,v_m)=1 \Leftrightarrow$ Точно една $v_i=1$.

Реализация:

$G(v_1,\ldots,v_m)=$

$v_1 \wedge \neg v_2 \wedge \neg v_3 \wedge \cdots \wedge \neg v_m \vee$

$\neg v_1 \wedge v_2 \wedge \neg v_3 \wedge \cdots \wedge \neg v_m \vee$

$\neg v_1 \wedge \neg v_2 \wedge v_3 \wedge \cdots \wedge \neg v_m \vee$

$\ldots$

$\neg v_1 \wedge \neg v_2 \wedge \neg v_3 \wedge \cdots \wedge v_m$

Брой променливи: $\mathcal{O}(m^2)$

Ограничители условия

"‘Променливите винаги описват конфигурация"’

$$R = \bigwedge_t G(\mathrm{state}_{t1},\ldots,\mathrm{state}_{tk}) \wedge$$
$$G(\mathrm{pos}_{t,-p(n)},\ldots,\mathrm{pos}_{tp(n)}) \wedge$$
$$\bigwedge_i G(\mathrm{band}_{ti1},\ldots,\mathrm{band}_{ti\ell})$$

Брой променливи:

$$\mathscr{O}\left(p(n)(k^2+p(n)^2+p(n)\ell^2)\right) = \mathscr{O}\left(p(n)^3\right)$$

Началното условие

"‘Формулата описва началната конфигурация $(1)w$"’

$$A = \mathrm{state}_{01} \wedge \mathrm{pos}_{01} \wedge$$
$$\bigwedge_{j=1}^{n} \mathrm{band}_{0jw_j} \wedge$$
$$\bigwedge_{j=-p(n)}^{0} \mathrm{band}_{0j\sqcup} \wedge$$
$$\bigwedge_{j=n+1}^{p(n)} \mathrm{band}_{0j\sqcup}$$

Общо: $\mathcal{O}(1+1+n+p(n)+p(n)-n) = \mathcal{O}(p(n))$

Условията за преход U_1, $t \to t+1$

"‘В позицията на главата съответният преход е съобразно δ"’

$$\mathrm{U}_1 = \bigwedge_{t,z,i,a} \Big((\mathrm{state}_{tz} \wedge \mathrm{pos}_{ti} \wedge \mathrm{band}_{tia}) \\ \to \bigvee_{\{(z',a',y) \in \delta(z,a)\}} (\mathrm{state}_{t+1,z'} \wedge \mathrm{pos}_{t+1,i+y} \wedge \mathrm{band}_{t+1,ia'}) \Big)$$

навсякъде посоките на движение на главата се интерпетират като числа,
$L = -1$, $N = 0$, $R = +1$.

Общо $\mathcal{O}((p(n) \cdot k \cdot p(n) \cdot \ell) \cdot (k \cdot \ell \cdot 3)) = \mathcal{O}\left(p(n)^2\right)$

Условията за преход U_2, $t \to t+1$

"‘В останалите позиции нищо не се променя"’

$$\mathrm{U}_2 = \bigwedge_{t,i,a} ((\neg \mathrm{pos}_{ti} \wedge \mathrm{band}_{tia}) \longrightarrow (\mathrm{band}_{t+1,i,a}))$$

Общо $\mathcal{O}((p(n) \cdot p(n) \cdot \ell)) = \mathcal{O}\left(p(n)^2\right) = \mathcal{O}\left(p(n)^2\right)$

Условия за завършване E

"‘M приема w за $\leq p(n)$ стъпки"’

$$E = \bigvee_{t \leq p(n)} \bigvee_{z \in H} \mathrm{state}_{t,z}$$

Общо $\mathcal{O}(p(n))$

Общ брой променливи на F

R Ограничителното условие $\mathcal{O}\left(p(n)^3\right)$

A Начално условие $\mathcal{O}\left(p(n)\right)$

U_1 Условие за преход 1 $\mathcal{O}\left(p(n)^2\right)$

U_2 Условие за преход 2 $\mathcal{O}\left(p(n)^2\right)$

E Заключително условие "‘M приема w за $\leq p(n)$ стъпки"’ $\mathcal{O}\left(p(n)\right)$

Общо $\mathcal{O}\left(p(n)^3\right)$.

Това е отново полиномиално

и може да се напише автоматично.

Твърдение: $\mathbf{NP} \subseteq \bigcup_{p \text{ polynomial}} \mathrm{TIME}(2^{\mathcal{O}(p(n))})$

Д-во:

Нека $T = (Q, \Sigma, \Gamma, \delta, s, F)$ е НМТ, която разпознава L с времева функция $p(n)$.

Може да се конструира детерминистичен алгоритъм, такъв че всички комбинации $\leq p(n)$ на недетерминистичен избор систематично да се пробват да се удовлетворят.

Има най-много

$$(3 \cdot |Q| \cdot |\Gamma|)^{p(n)} = 2^{\log_2(3 \cdot |Q| \cdot |\Gamma|) p(n)} = 2^{\mathcal{O}(p(n))}$$

такива комбинации.

Други **NP**-пълни проблеми

Наблюдение: $\leq_p$ е транзитивна, т.е.,
$\forall L, L', L'' : L \leq_p L' \wedge L' \leq_p L'' \longrightarrow L \leq_p L''$.

Лема A: $L \in \mathbf{NP} \wedge SAT \leq_p L \longrightarrow L$ е **NP**-пълна.
Д-во: Трябва да докажем, че $\forall L' \in \mathbf{NP} : L' \leq_p L$.
SAT е **NP**-пълна също $L' \leq_p$SAT.
От асоциативността на $\leq_p$ следва, че
$L' \leq_p \mathrm{SAT} \leq_p L$.
qed

Други **NP**-пълни проблеми

Лема А: $L \in \mathbf{NP} \wedge SAT \leq_p L \longrightarrow L$ е **NP**-пълен.

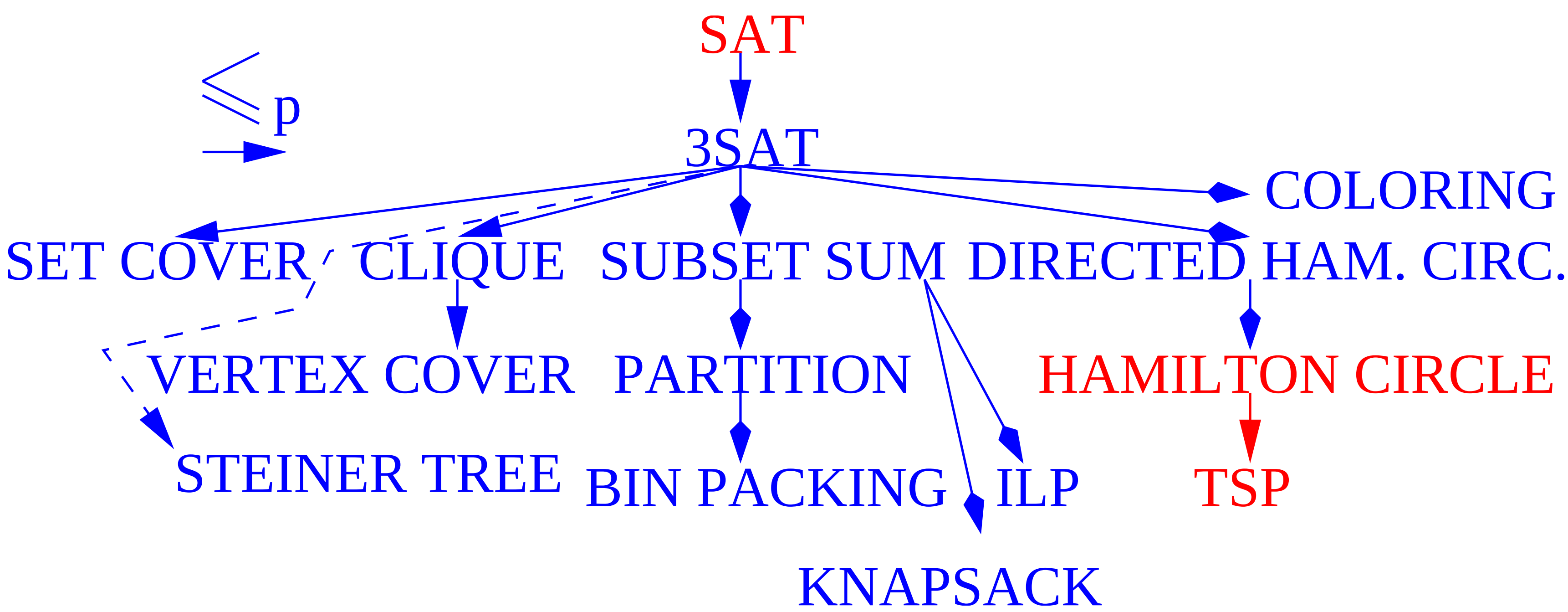
Лема А’:

$L \in \mathbf{NP} \wedge L'$ е NP пълен $\wedge L' \leq_p L$

$\longrightarrow L$ е **NP**-пълен.

Д-во: Аналогично на док. на Лема А.

Други **NP**-пълни проблеми



3SAT (CNF)

Дадено: Формула в конюнктивна нормална форма, max. 3 литерали в клауза

Въпрос: Дали F е изпълнима?

Пример: $(x \vee \neg y \vee z)(\neg u \vee x \vee z)(\neg x)$

Променливи: $x, y, z, \ldots$

Литерал: Variable, $\neg$Variable

Клаза: Literal$\vee \cdots \vee$Literal

CNF формула: (clause)$\cdots$(clause)

Твърдение: 3SAT е **NP**-пълна

Д-во:

Тъй като SAT е в **NP**, достатъчно е да покажем, че 3SAT е **NP**-трудна и

SAT$\leq_p$3SAT

Внимание: Не помага изобщо привеждането в CNF.

- CNF може да е с експоненциален размер
- Не е задължително да получим 3CNF

Д-во "‘3SAT е **NP**-трудна"’

Ще опишем полиномиален алгоритъм,
който трансформира всяка формула F в еквивалентна по изпълнимост 3CNF-формула F', т.е.,

$$F \text{ изпълнима} \Leftrightarrow F' \text{ изпълнима}$$

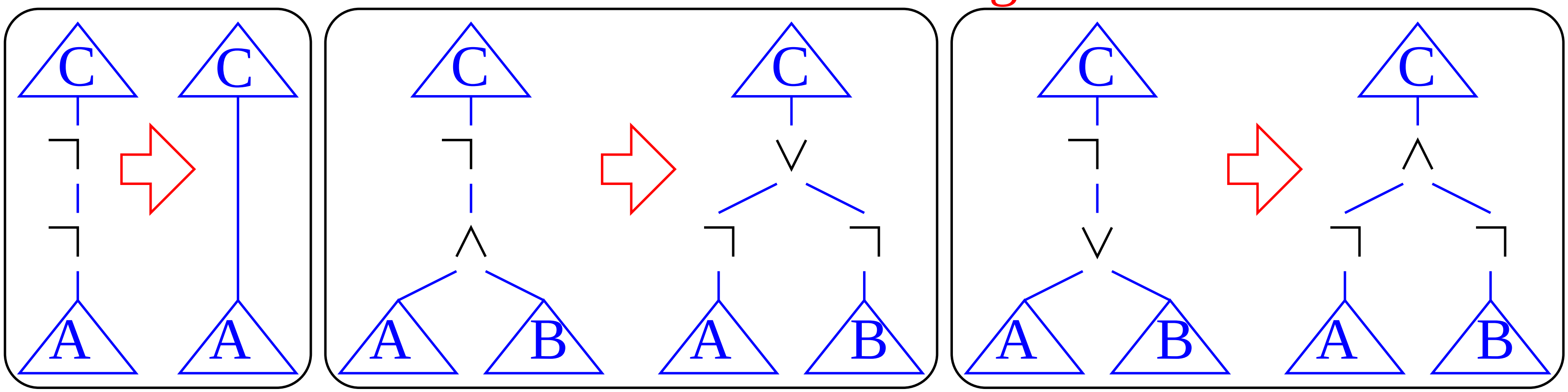
Основен метод: една редица от еквивалентни по изпълнимост трансформации.

Разглеждаме формулата като едно дърво с max. grad две.

Листа: променливи x (или литерали $\bar{x} = \neg x$) Други възли: $\wedge$, $\vee$, $\neg$

Преместване на отрицанието пред листата

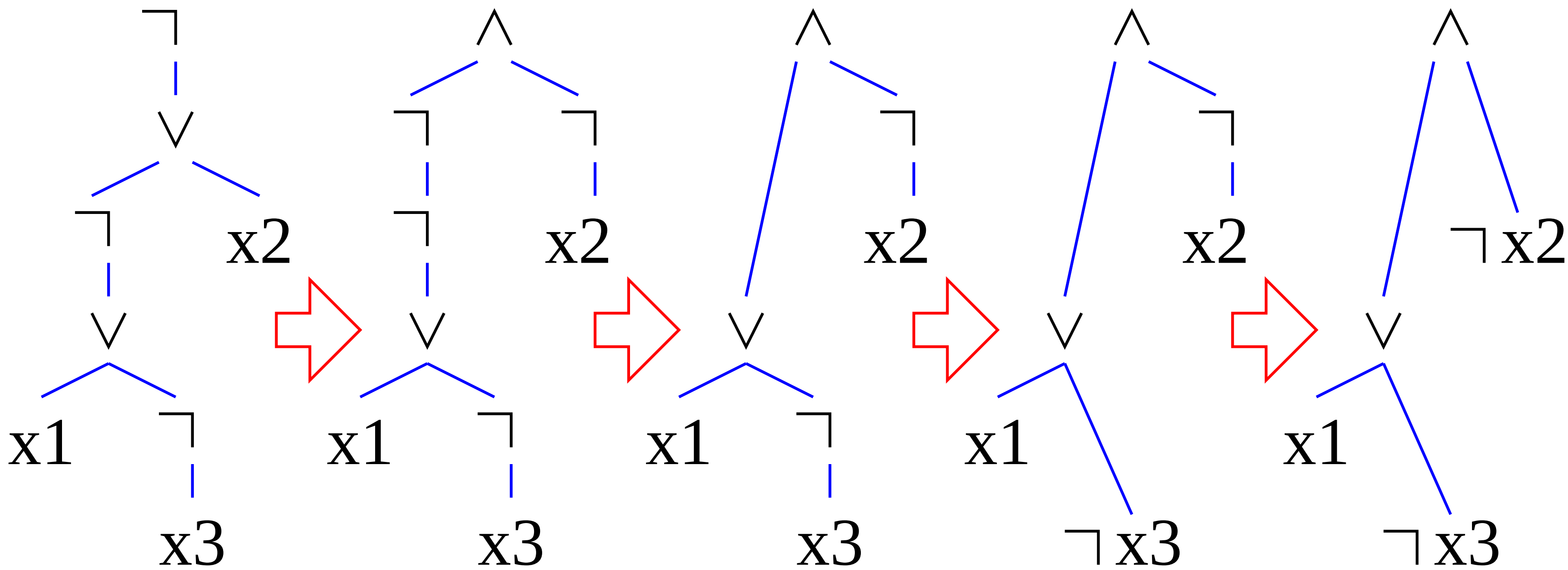
De Morgan



Обхождайки отгоре надолу се изпълнява за линейно време.

postcondition: Интерпретираме формулата като двоично дърво с вътрешни възли $\vee$, $\wedge$ и литерали като листа.

Пример



Преместване на отрицанието до листата

normalize($\neg x$) return negNormalize(x)

normalize($x \wedge y$) return normalize(x) $\wedge$ normalize(y)

normalize($x \vee y$) return normalize(x) $\vee$ normalize(y)

normalize(v) return v

negNormalize($\neg x$) return normalize(x)

negNormalize($x \wedge y$) return negNormalize(x) $\vee$ negNormalize(y)

negNormalize($x \vee y$) return negNormalize(x) $\wedge$ negNormalize(y)

negNormalize(v) return $\neg v$

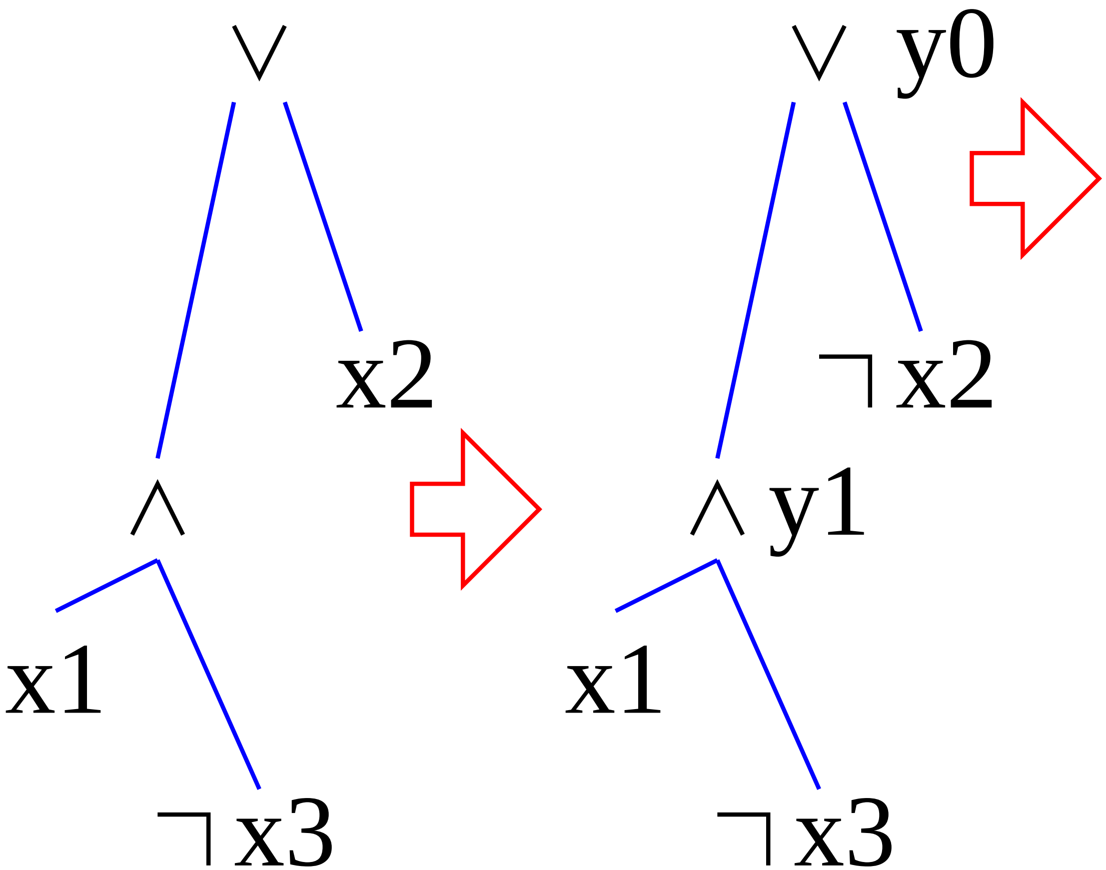
За вътрешните възли → нови променливи

Добавяме нови променливи за $\wedge$, $\vee$.

Нека y_0 да е литерал на корена.

$$F_1 := (y_0) \wedge \Big(\bigwedge_{\overbrace{y \;\wedge\; z}^{v}} v \leftrightarrow (y \wedge z) \Big) \wedge \Big(\bigwedge_{\overbrace{y \;\vee\; z}^{v}} v \leftrightarrow (y \vee z) \Big)$$

Пример



$$(y0)$$
$$(y0 \leftrightarrow (y1 \vee \neg x2))$$
$$(y1 \leftrightarrow (x1 \wedge \neg x3))$$

Еквивалентност по изпълнимост на F_1

Д-во F е изпълнима $\longrightarrow F_1$ е изпълнима:

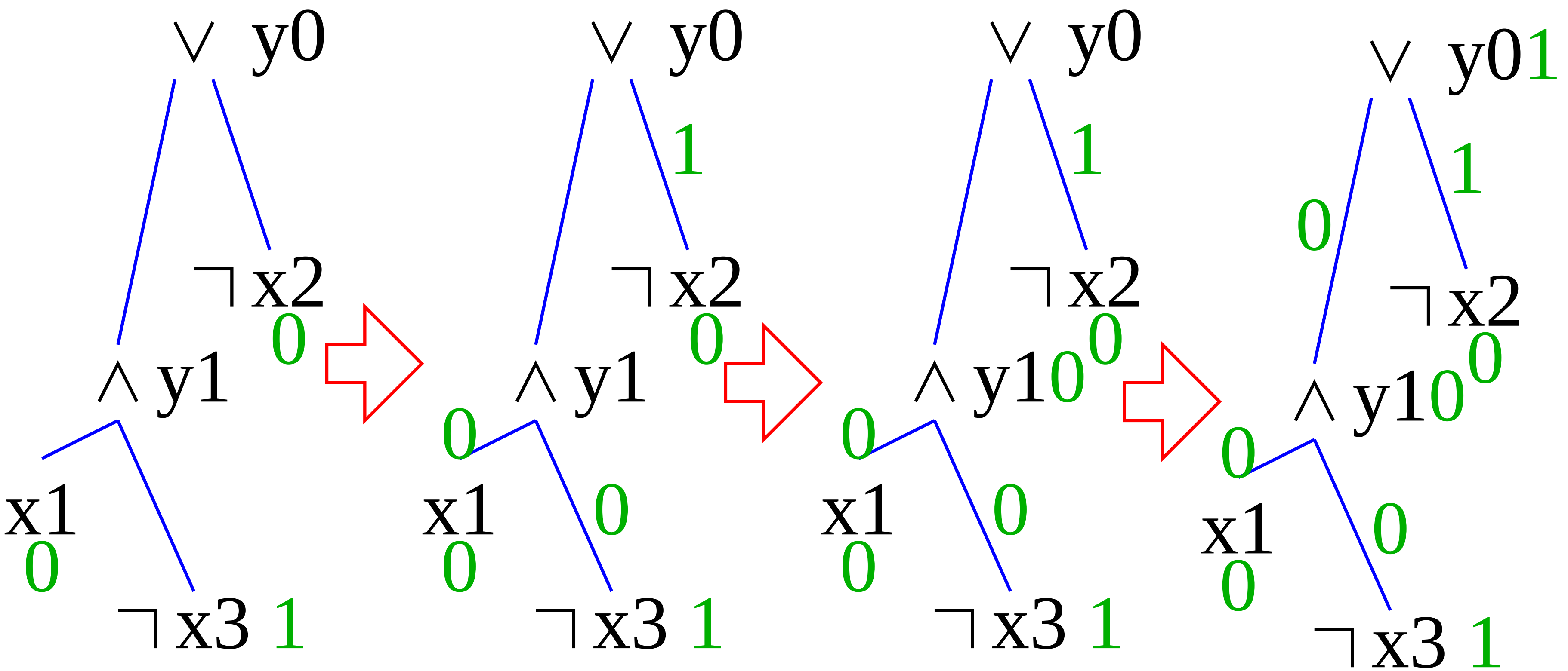
Нека има оценка на променливите на F, такова че F е вярна.

Разглеждаме тази оценка като частично остойностяване на F_1.

Остойностяваме формулата отдолу нагоре.

Взимаме получените истинни стойности на поддървото с корен променливата v за оценка на v. Така F_1 е вярна при тази оценка.

Пример


∨ y0
¬ x2 0
∧ y1
x1 0
¬ x3 1
∨ y0
1
¬ x2 0
∧ y1
0
x1 0
0
¬ x3 1
∨ y0
1
¬ x2 0
∧ y1 0
0
x1 0
0
¬ x3 1
∨ y0 1
1
0
¬ x2 0
∧ y1 0
0
x1 0
0
¬ x3 1

Еквивалентност по изпълнимост на F_1

Д-во F_1 е изпълнима $\longrightarrow F$ е изпълнима:

Да разгледаме оценка на F_1, за която F_1 е истина.

Взимаме истинните стойности на листата за оценка на F.

Възлите, които не са листа в F имат същата истинна стойност като съответните променливи на F_1.

y_0 е има стойност 1.

Следователно F е истина.

$F_1 \rightsquigarrow$ 3CNF

Построяваме CNF за всяка подформула:

$a \leftrightarrow (b \vee c) \rightsquigarrow (a \vee \neg b)(\neg a \vee b \vee c)(a \vee \neg c)$

$a \leftrightarrow (b \wedge c) \rightsquigarrow (\neg a \vee b)(\neg a \vee c)(a \vee \neg b \vee \neg c)$

a	b	c	$b \vee c$	$b \wedge c$	$a \leftrightarrow (b \vee c)$	$a \leftrightarrow (b \wedge c)$
0	0	0	0	0	1	1
0	0	1	1	0	0	1
0	1	0	1	0	0	1
0	1	1	1	1	0	0
1	0	0	0	0	0	0
1	0	1	1	0	1	0
1	1	0	1	0	1	0
1	1	1	1	1	1	1

excursus: 2SAT$\in$ P

Дадено : Съждителна формула в конюнктивна нормална форма, с max. 2 литарала в клауза.
Въпрос: Дали F е изпълнима?

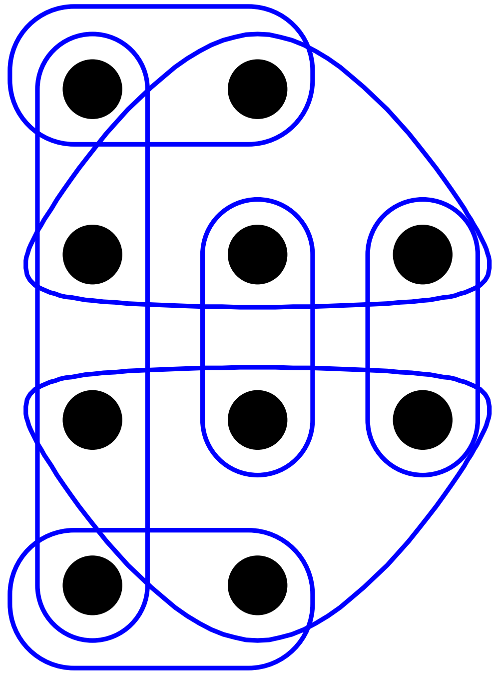
Дефинираме $\overline{x}:= \neg x$, $\overline{\neg x}:= x$.
Граф $G = (V,E)$, $V:= \{x,\neg x : x \in F\}$,
$E:= \left\{(\overline{A},B),(\overline{B},A) : (A\vee B) \in F\right\}$

Наблюдение: F е изпълнима
$\neg\exists$ цикъл C в G и променлива $x : x \in C \wedge \neg x \in C$.
Тези условия можем да проверим за линейно време.
(Силно свързана компонента.)

SET COVER

Дадено:	Пример:
Основни множества M	$\{1,2,3,4,5\}$
Система от множества $\mathscr{T} \subseteq 2^M$	$\{\{1,3\},\{1,2\},\{2,4,5\}\}$
Параметър n	2
Въпрос:	
Има ли елементи $T_1 \in \mathscr{T}, \ldots, T_n \in \mathscr{T}$?	Yes:
$T_1 \cup \cdots \cup T_n = M$?	$\{1,3\},\{2,4,5\}$

Set Cover Пример



Минимално n: 3

Твърдение: SET COVER е **NP**-пълна задача

Д-во на SET COVER $\in$ **NP**:

Избираме недетерминистично n множества $T_1 \in \mathscr{T}, \ldots, T_n \in \mathscr{T}$

Образуваме обединението $M' = T_1 \cup \cdots \cup T_n$

Проверяваме дали $M = M'$

Д-во "‘SET COVER е **NP**-трудна"’ задача

Ние ще покажем, че 3SAT$\leq_p$SET COVER.

Нека $F = K_1 \wedge \cdots \wedge K_m$ е CNF-формула с променливи $x_1, \ldots, x_n$.

Избираме $M = \{1, \ldots, m+n\}$.

$T_i := \{j : x_i$ участва в клаузата $K_j\} \cup \{m+i\}$

$T_i' := \{j : \neg x_i$ участва в клаузата $K_j\} \cup \{m+i\}$

Полагаме $\mathscr{T} := \{T_1, \ldots, T_n, T_1', \ldots, T_n'\}$.

Параметър n.

Да се докаже : Примерът за покритие $(M, \mathscr{T}, n)$ има решение, ако F е изпълнима.

Пример

$F = (x_1 \vee x_2 \vee x_3)(x_2 \vee \neg x_3 \vee x_4)(\neg x_3 \vee \neg x_4 \vee \neg x_1)$

$n = 4,\ m = 3.$

$T_1 = \{1,4\},$ $\{3,4\} = T_1'$

$T_2 = \{1,2,5\},$ $\{5\} = T_2'$

$T_3 = \{1,6\},$ $\{2,3,6\} = T_3'$

$T_4 = \{2,7\},$ $\{3,7\} = T_4'$

$T_1 \cup T_2 \cup T_3' \cup T_4 = \{1,2,3,4,5,6,7\}$

Така $x_1 = x_2 = x_4 = 1, x_3 = 0$

Д-во F изпълнима $\longrightarrow (M, \mathscr{T}, n)$ има решение.

Избираме $M = \{1, \ldots, m+n\}$.

$T_i := \{j : x_i$ участва в клаузата $K_j\} \cup \{m+i\}$

$T_i' := \{j : \neg x_i$ участва в клаузата $K_j\} \cup \{m+i\}$

for $i = 1..n$

Избери T_i, ако $x_i = 1$.

Избери T_i', ако $x_i = 0$.

$\{m+1, \ldots, m+n\}$ ще се покрие, тъй като за всяка променлива x_i или T_i, или T_i' е избрано.

$j \in \{1, \ldots, m\}$ ще се покрие:

Нека L да е литералът в K_j, който има стойност "true"

Случай $L = x_i \longrightarrow x_i = 1 \longrightarrow j \in T_i$ и T_i ще бъде избрано.

Случай $L = \neg x_i \longrightarrow x_i = 0 \longrightarrow j \in T_i'$ и T_i' ще бъде избрано.

Д-во $(M, \mathscr{T}, n)$ има решение$\longrightarrow F$ изпълнима

$M = \{1, \ldots, m+n\}$.

$T_i := \{j : x_i$ участва в клаузата $K_j\} \cup \{m+i\}$

$T_i' := \{j : \neg x_i$ участва в клаузата $K_j\} \cup \{m+i\}$

$\{m+1, \ldots, m+n\}$ ще се покрие $\longrightarrow$ за всяка променлива x_i или T_i, или T_i' ще бъде избрано, иначе някое от $m+i$ няма да се покрие.

$T_i \rightsquigarrow x_i = 1$.

$T_i' \rightsquigarrow x_i = 0$.

$\forall j \in 1..m :$

Случай 1, $j \in T_i$, T_i е избрано: $x_i \in K_j$ прави K_j истина $(x_i = 1)$

Случай 2, $j \in T_i'$, T_i' е избрано: $\neg x_i \in K_j$ прави K_j истина $(x_i = 0)$

CLIQUE

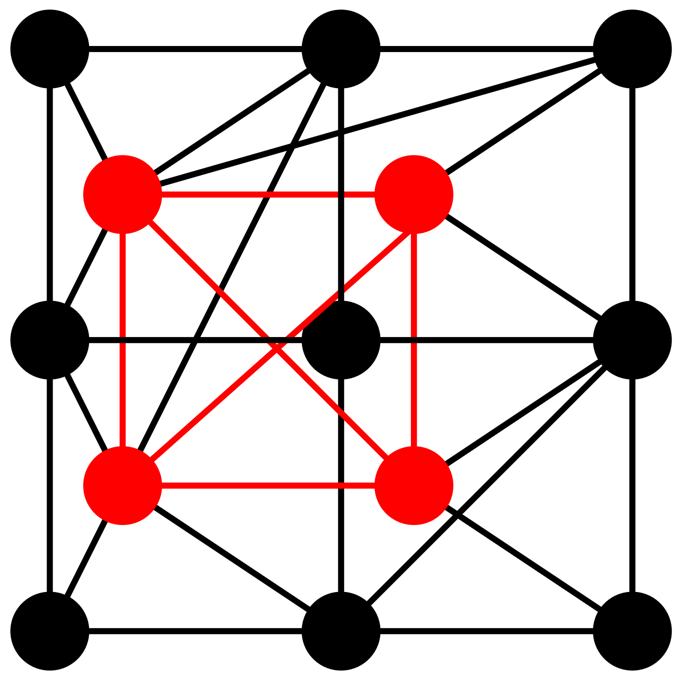
Дадено: Неориентиран граф $G = (V, E)$,

Параметър $k \in \mathbb{N}$.

Въпрос:

Дали G съдържа клика са размер k?

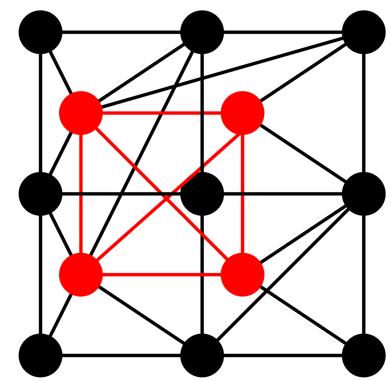
т.е. $\exists V' \subseteq V : |V'| = k \wedge \forall u \neq v \in V' : \{u, v\} \in E$



Д-во Clique$\in$ **NP**

Избери k възела от V';
провери дали те оформят клика.

Дадено: Неориентиран граф $G = (V, E)$,
параметър $k \in \mathbb{N}$.
Въпрос:
Дали G съдържа клика с размер k?
т.е. $\exists V' \subseteq V : |V'| = k \wedge \forall u \neq v \in V' : \{u, v\} \in E$



Д-во за "‘CLIQUE е **NP**-трудна"’

Ние ще покажем, че 3SAT$\leq_p$CLIQUE.

Нека $F = (z_{11} \vee z_{12} \vee z_{13}) \wedge \cdots \wedge (z_{m1} \vee z_{m2} \vee z_{m3})$ с

литерали $z_{ij} \in \{x_1, \ldots, x_n\} \cup \{\neg x_1, \ldots \neg x_n\}$.

Построяваме граф $G = (V, E)$:

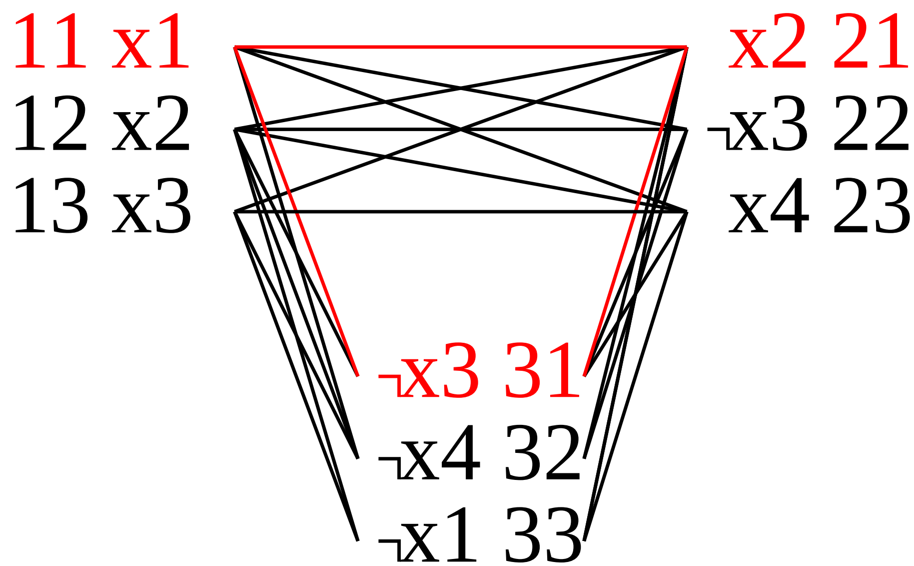
$V = \{(1,1), (1,2), (1,3), \ldots, (m,1), (m,2), (m,3)\}$,

$E = \{\{(i,j),(p,q)\} : i \neq p, z_{ij} \neq \overline{z_{pq}}\}$.

Параметър $k = m$.

Пример:

$$F = (x_1 \vee x_2 \vee x_3)(x_2 \vee \neg x_3 \vee x_4)(\neg x_3 \vee \neg x_4 \vee \neg x_1)$$



Идея: Във всяка клауза има удовлетворим литерал.

Дъгите свързват "‘съвместими "’ литерали

$F = (z_{11} \vee z_{12} \vee z_{13}) \wedge \cdots \wedge (z_{m1} \vee z_{m2} \vee z_{m3})$.
$V = \{(1,1),(1,2),(1,3),\ldots,(m,1),(m,2),(m,3)\}$,
$E = \{\{(i,j),(p,q)\} : i \neq p, z_{ij} \neq \overline{z_{pq}}\}$.

F е изпълнима $\longrightarrow$ $G = (V,E)$ съдържа m-клика

F е изпълнима с оценка $B : \{x_1,\ldots,x_n\} \to \{0,1\}$
$\longrightarrow \forall j \in 1..m : \exists z_{j,k_j} : B$ прави z_{jk_j} истина
Тези литерали са два по два не контрарни
и идват от различни клаузи.
$\longrightarrow (1,k_1),\ldots,(m,k_m)$ два по два свързани.
Построяваме m-клика
qed

$$F = (z_{11} \vee z_{12} \vee z_{13}) \wedge \cdots \wedge (z_{m1} \vee z_{m2} \vee z_{m3}).$$
$$V = \{(1,1),(1,2),(1,3),\ldots,(m,1),(m,2),(m,3)\},$$
$$E = \{\{(i,j),(p,q)\} : i \neq p, z_{ij} \neq \overline{z_{pq}}\}.$$

$G = (V,E)$ има m-клика $\longrightarrow$ F е изпълнима

$(j_1,k_1),\ldots(j_m,k_m)$ е m-клика с литерали z_{j_i,k_i}.

дъги само за различни j_i.

$\longrightarrow$тези j_i са два по два различни.

$\longrightarrow j_1,\ldots,j_m = 1,\ldots,m$

Дъги само между не контрарни литерали.

$\longrightarrow$ тези литерали са два по два неконтрарни.

$\longrightarrow$ $\exists$ оценка $B : \forall j \in 1..m : z_{jk_j}[B] = 1$

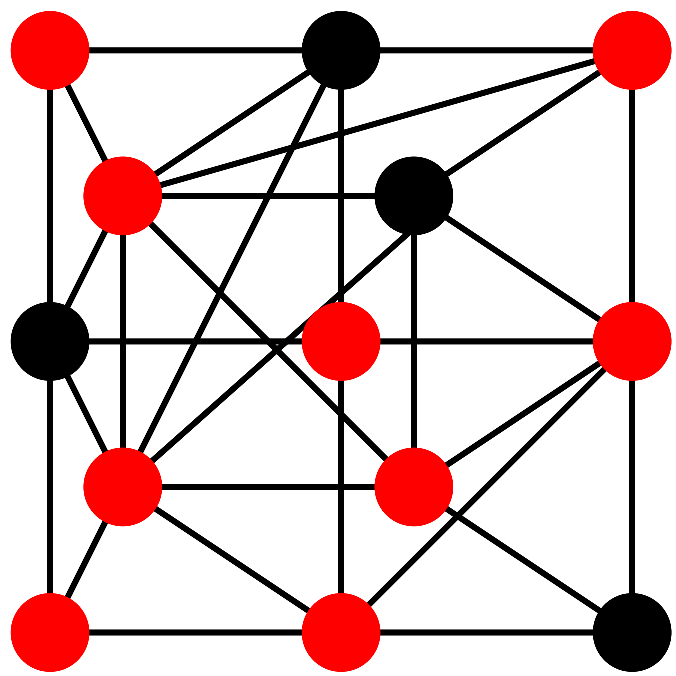
$\longrightarrow$ F е изпълнима с B qed

VERTEX COVER (покритие на възлите)

Дадено: неориентиран граф $G = (V, E)$,

Параметър $k \in \mathbb{N}$.

Въпрос: $\exists V' \subseteq V : |V'| = k \wedge \forall \{u, v\} \in E : u \in V' \vee v \in V'$



Д-во VERTEX COVER $\in$ **NP**

…като предишното…

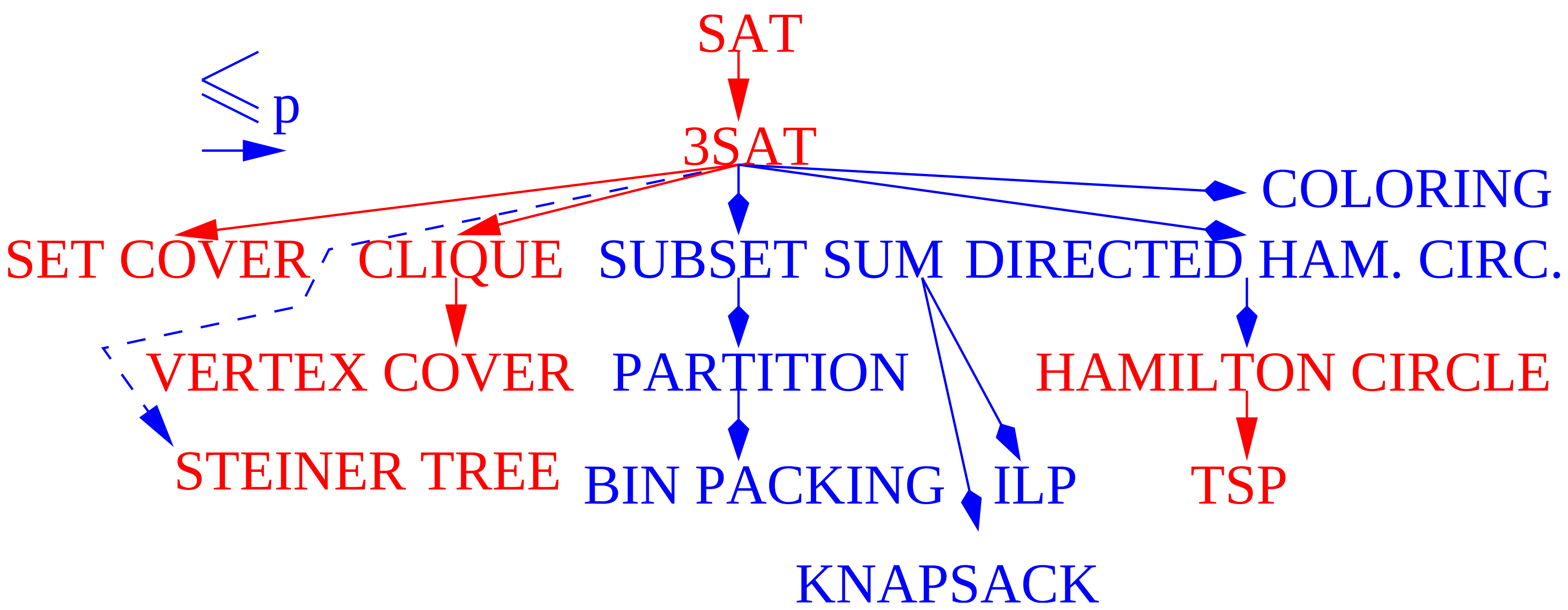
Д-во за "‘VERTEX COVER е **NP**-трудна"’

Показваме: CLIQUE$\leq_p$VERTEX COVER.

Разглеждаме допълнителния граф
$\bar{G} = (V, \bar{E})$ с $\bar{E} = \{u \neq v \in V : \{u,v\} \notin E\}$.

$$
\begin{aligned}
&G \text{ има Vertex Cover } V' \text{ с размер } k && \Leftrightarrow\\
&\exists V' \subseteq V : |V'| = k \wedge \forall u \neq v : \{u,v\} \in E \rightarrow (u \in V' \vee v \in V') && \Leftrightarrow\\
&\exists V' \subseteq V : |V'| = k \wedge \forall u \neq v : \{u,v\} \notin E \vee (u \in V' \vee v \in V') && \Leftrightarrow\\
&\exists V' \subseteq V : |V'| = k \wedge \forall u \neq v : \{u,v\} \notin E \vee \neg(u \notin V' \wedge v \notin V') && \Leftrightarrow\\
&\exists V' \subseteq V : |V'| = k \wedge \forall u \neq v : (u \notin V' \wedge v \notin V') \rightarrow \{u,v\} \notin E && \Leftrightarrow\\
&\exists V' \subseteq V : |V'| = k \wedge \forall u \neq v : \{u,v\} \in V \setminus V' \rightarrow \{u,v\} \in \bar{E} && \Leftrightarrow\\
&\exists \bar{V}' \text{ – клика с размер } |V| - k \text{ за } \bar{G}.
\end{aligned}
$$

Разгледани сводимости



Техники за $A \leq_p B$:

Специален пример

Показваме, че A е специален пример (в широк смисъл) за B.

Всеки елементарен обект във формулирания проблем за A отговаря на един елементарен обект във формулирания проблем за B.

Пример: HAMILTON CYCLE$\leq_p$TSP.

Техники за $A \leq_p B$:

Друга интерпретация

Показваме, че A е специален случай (в широк смисъл) за B.

Всеки елементарен обект във формулирания проблем за A отговаря на един елементарен обект във формулирания проблем за B.

Пример: VERTEX COVER$\leq_p$CLIQUE.

Пример: CLIQUE$\leq_p$VERTEX COVER.

Д-во - техники за $A \leq_p B$:

Gadgets

Всеки елементарен обект във формулирания проблем за A ще конструира няколко свързани обекта в B .

Пример: 3SAT$\leq_p$SET COVER; $x_i \rightsquigarrow T_i, T_i'$.

Пример: SAT$\leq_p$3SAT; възли за $F \rightsquigarrow$ три клаузи.

- Състояния, позиция на главата, символи на лентата $\rightsquigarrow$ много променливи.
- Началната конфигурация $\rightsquigarrow$ формула A
- $\delta \rightsquigarrow$ преход, отговарящ на U_1
- Заключителни състояния$\rightsquigarrow$ условие за завършване E

Д-во - техники за $A \leq_p B$:

Ограничителни условия

Налагаме определени характеристики с допълнителни конструкции.

Пример: SAT$\leq_p$3SAT; (y_0).

Пример: 3SAT$\leq_p$COLORING

COLORING

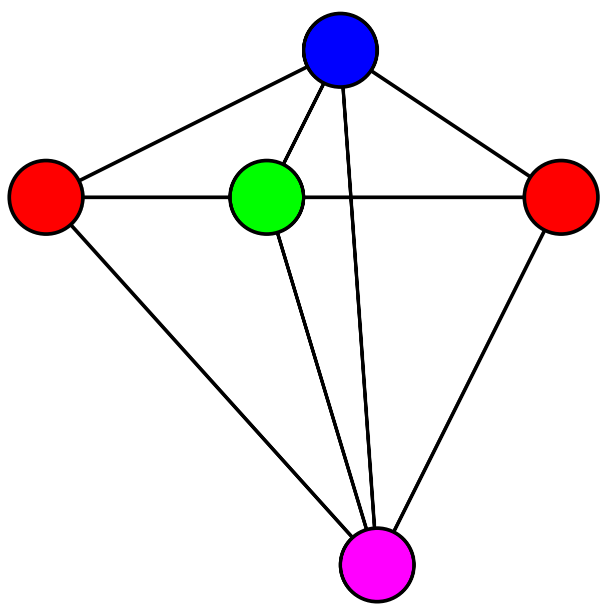
Дадено: неориентиран граф $G = (V, E)$,

Параметър $k \in \mathbb{N}$.

Въпрос:

Има ли оцветяване на възлите $c : V \to \{1, \ldots, k\}$

така, че $\forall \{u, v\} \in E : c(u) \neq c(v)$.



Д-во COLORING $\in$ **NP**

. . . същото. . .

Д-во за "‘COLORING е **NP**-трудна"’

Показваме, че 3SAT$\leq_p$3-COLORING (т.е. $k = 3$).

Нека $F = K_1 \wedge \cdots \wedge K_m$ с литерали от

$\{x_1, \ldots x_n\} \cup \{\neg x_1, \ldots, \neg x_n\}$

точно 3 литерала в клауза.

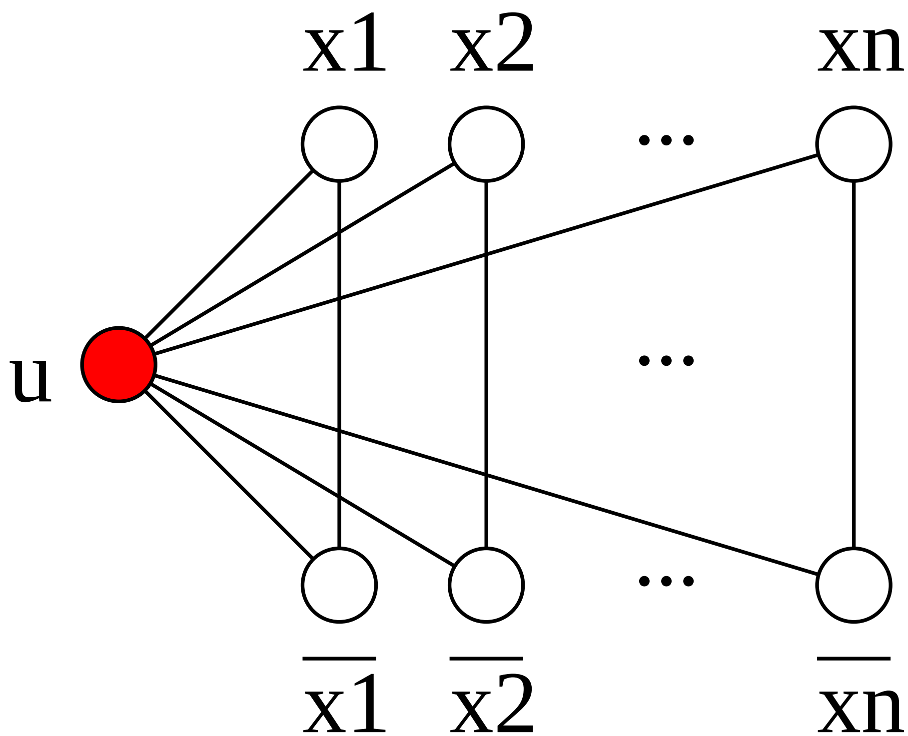
Идеа:

Конструираме граф $G = (V, E)$ с $2n + 5m + 2$ възли, така че

G 3-coloring $\Leftrightarrow$ F е изпълнима.

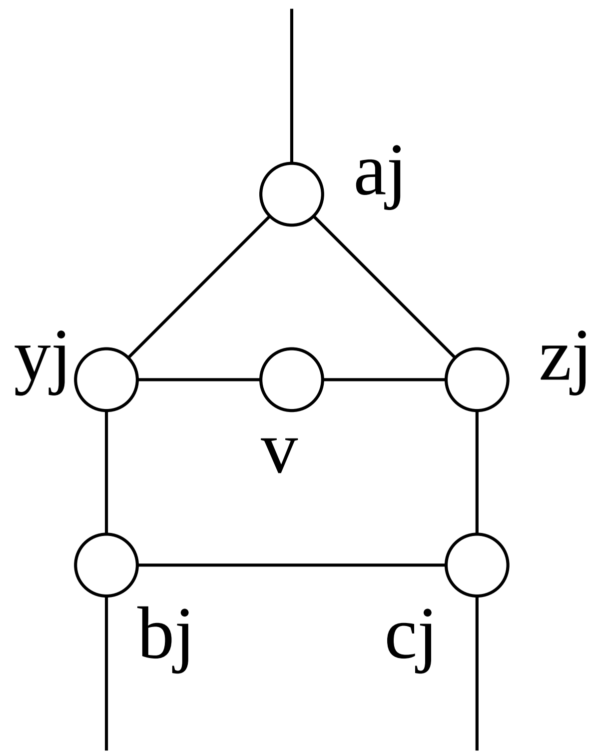
Ограничително условие налага непротиворечиви вярностни стойности за литерали

Една част от G:



Colors $0 = \text{true}, 1 = \text{false}, 2 = \text{rot} = c(u)$.

Gadget за клауза $K_j = (a_j \vee b_j \vee c_j)$

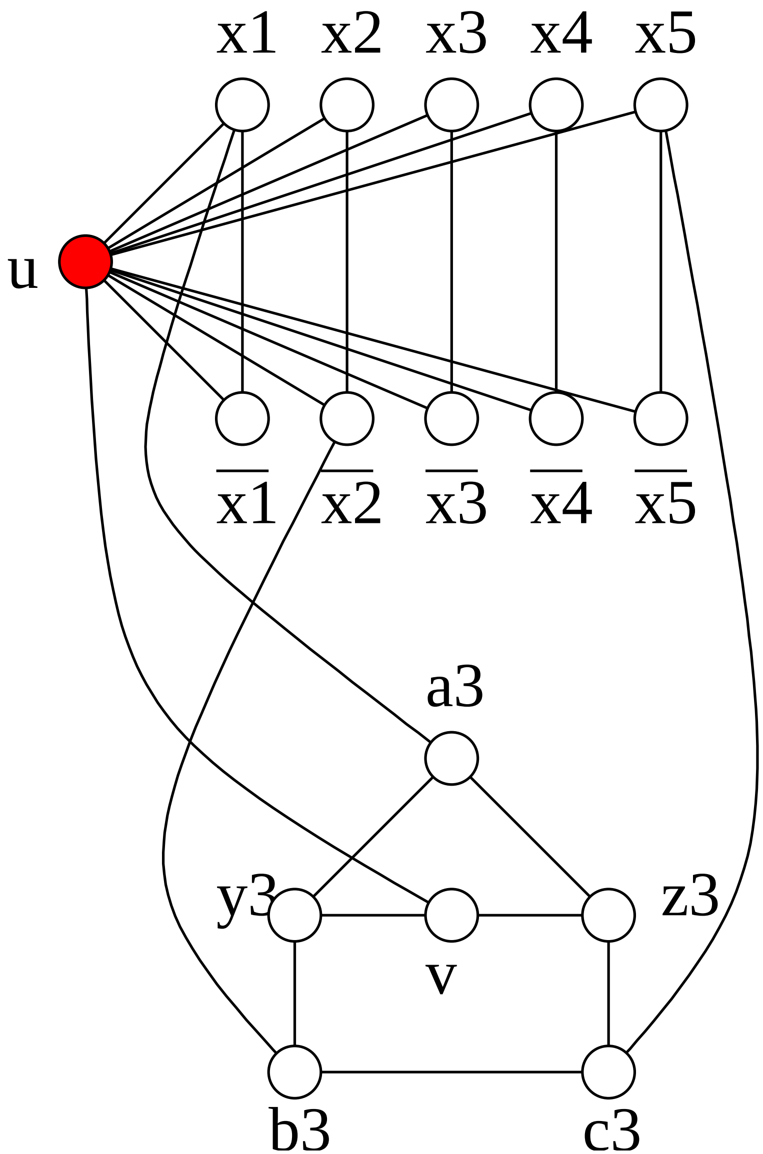


v: Навсякъде $(u, v) \in E$

a_j, b_j, c_j: Нови възли за всяка клауза.

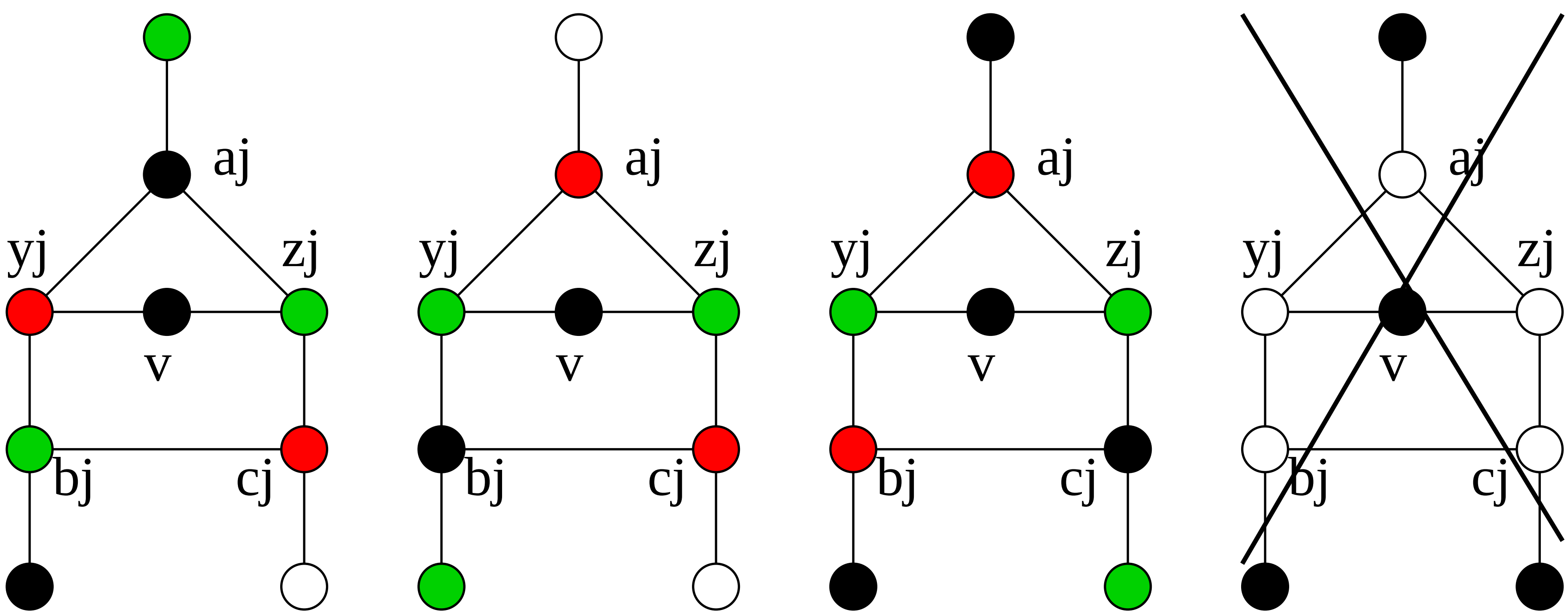
Внимание: Цветът на $a_j \neq$ стойността на $a_j, \ldots$!

Пример: Gadget за клаузата $K_3 = (x_1 \vee \overline{x_2} \vee x_5)$



Д-во: F е изпълнима $\longrightarrow$ G 3-coloring

$c(u)$:= червено, $c(v)$:= false,
избираме $c(x_i) = \neg c(\overline{x_i})$ съответстваща на изпълнима оценка.
За клаузата j -Gadget j:



Д-во: G 3-coloring$\longrightarrow F$ е изпълнима

Нека фиксираме цвета на u например червено.

Тогава $c(v) \neq c(u)$ false.

Тогава третият цвят е true.

Графът налага

непротиворечиви истинни стойности за

литерарите-възли $x_i, \overline{x_i} \in \{\text{true}, \text{false}\}$.

Ще покажем:

Оценката B прави F true— $F[B] = \text{true}$

Д-во С допускане на противното.

Да допуснем $F[B] = \text{false}$?

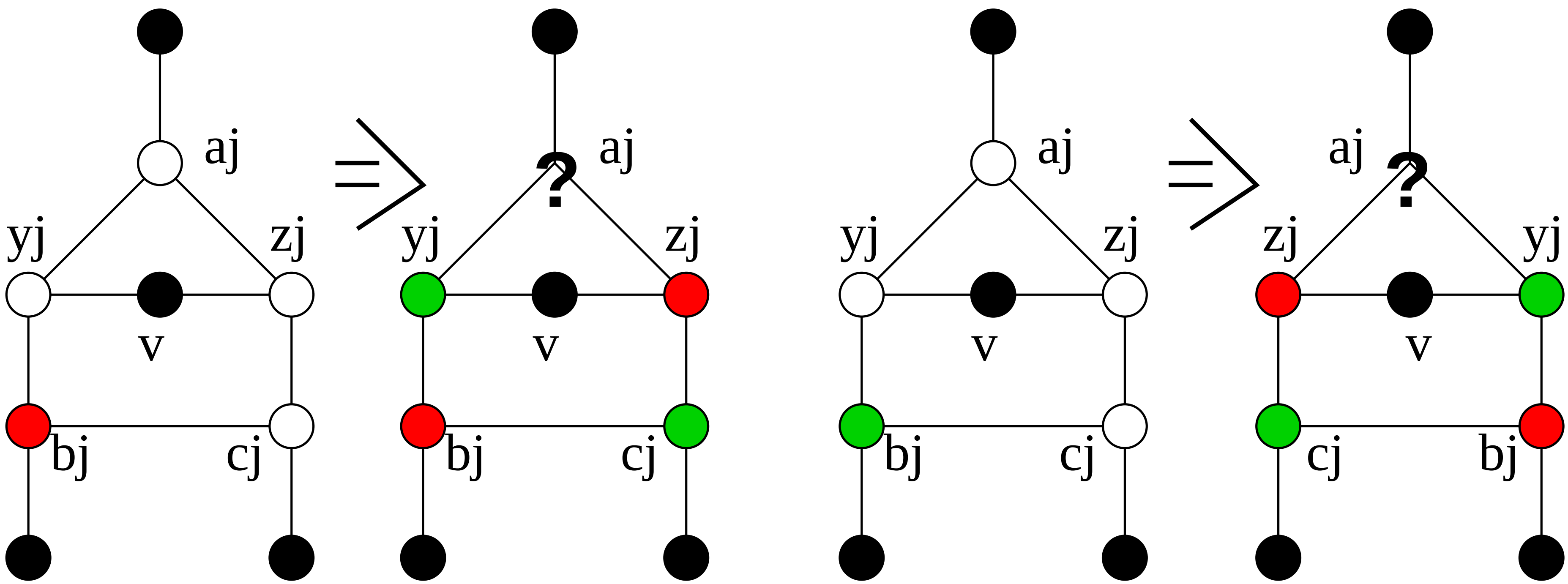
$\longrightarrow \exists j :$ $K_j[B] = \text{false}$.

$\longrightarrow$ Всички литерали на K_j са false.

Д-во: G 3-coloring$\longrightarrow F$ е изпълнима

$[\cdots]$ Всички литерали на K_j са false

$\longrightarrow c(b_j) =$червено или $c(b_j) =$true



Очевидно $F[B]$ е също true.

qed

SUBSET SUM

Дадено: n обекта с тегло $w_i \in \mathbb{N}$

Параметър $W \in \mathbb{N}$.

Въпрос:

Има ли подмножество M на $\{1,\ldots,n\}$,

така, че $\sum_{i \in M} w_i = W$

$\approx$Специален случай на задачата за раниците с равна печалба.

Д-во SUBSET SUM $\in$ **NP**

...знаем го...

Д-во "‘SUBSET SUM е **NP**-трудна"’

Ще покажем, че 3SAT$\leq_p$SUBSET SUM.

Нека

$F = K_1 \wedge \cdots \wedge K_m = (z_{11} \vee z_{12} \vee z_{13}) \wedge \cdots \wedge (z_{m1} \vee z_{m2} \vee z_{m3})$ с литерали $z_{ij} \in \{x_1, \ldots, x_n\} \cup \{\neg x_1, \ldots, \neg x_n\}$.

Дефинираме следния пример за SUBSET SUM с тегла

$(v_1, \ldots, v_n, v'_1, \ldots v'_n, c_1, \ldots, c_m, d_1, \ldots, d_m)$

и сума W

SUBSET SUM пример

$(v_1,\ldots,v_n,v'_1,\ldots v'_n,c_1,\ldots,c_m,d_1,\ldots,d_m),W$

$m+n$ десетични числа (думи в $\{0,\ldots,9\}$)

Нека v_{ij}:= е броят на участията на x_i в клаузата j.

(Десетични цифри)

Нека v'_{ij}:= броят на участията $\neg x_i$ в клаузата j.

(Десетични цифри)

$v_i = v_{i1}v_{i2}\cdots v_{im}0^{i-1}10^{n-i}$,

$v'_i = v'_{i1}v'_{i2}\cdots v'_{im}0^{i-1}10^{n-i}$

$c_j := 0^{j-1}10^{m-j}0^n$

$d_j := 0^{j-1}20^{m-j}0^n$

$W := \underbrace{4\cdots4}_{m}\underbrace{1\cdots1}_{n}$

Пример

$$F = (x_1 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee x_5 \vee x_4) \wedge (\neg x_2 \vee \neg x_2 \vee \neg x_5)$$

$m = 3$, $n = 5$

$$\begin{array}{ll||ll}
v_1 & = 100\ 10000 & v'_1 & = 010\ 10000 \\
v_2 & = 000\ 01000 & v'_2 & = 002\ 01000 \\
v_3 & = 000\ 00100 & v'_3 & = 100\ 00100 \\
v_4 & = 010\ 00010 & v'_4 & = 000\ 00010 \\
v_5 & = 110\ 00001 & v'_5 & = 001\ 00001 \\
\hline
c_1 & = 100\ 00000 & d_1 & = 200\ 00000 \\
c_2 & = 010\ 00000 & d_2 & = 020\ 00000 \\
c_3 & = 001\ 00000 & d_3 & = 002\ 00000
\end{array}$$

$$W = 444\ 11111$$

Д-во идея

Наблюдение:

с десетично събиране на теглата (няма overflow).

$$W = 444\ 11111$$

Десният блок гарантира за всяка променлива x_i или $x_i \equiv v_i$ или $\neg x_i \equiv v_i'$ е избрана.

Позицията j на левия блок е за клауза j.

Избраните v_i, v_i' = броят на удовлетворените литерали: 1,2 или 3 за удовлетворена клауза.

c_j, d_j са допълнителни-променливи, позицията j от удовлетворените клаузи на стойностите могат да се удовлетворят и сумата да е 4 .

F изпълнима$\longrightarrow$ Решение на примера

Да разгледаме оценката B с $F[B] = 1$.

Избираме за M, v_i когато $x_i[B] = 1$ или v_i' ако $x_i[B] = 0$.

До сега е в сила $\sum_{i \in M} w_i = t_1 t_2 \cdots t_m 1^n$

с $t_j \in \{1,2,3\}$ (всяка клауза има 1–3 изпълними литерали).

for $j := 1$ to m do

if $t_j = 1$ then select c_j and d_j // $1+1+2=4$

if $t_j = 2$ then select d_j // $2+2=4$

if $t_j = 3$ then select c_j // $3+1=4$

Така $\sum_{i \in M} w_i = 4^m\ 1^n = W$.

qed.

Решение на примера $\longrightarrow F$ изпълнима

Да разгледаме M с $\sum_{i \in M} w_i = W = 4^m \; 1^n$.

За всяка променлива i или v_i или v'_i ще бъде избрана,

иначе i в десния блок ще бъде $\neq 1$.

Избираме оценка B така че $x_i = 1$ ако v_i е избрана, $x_i = 0$ иначе.

Твърдим : $F[B] = 1$ (С допускане на противното)

Да допуснем: $F[B] = 0$ и следователно $\exists j : K_j[B] = 0$.

$\longrightarrow \sum$ на избраните v_i, v'_i е 0 на позиция j

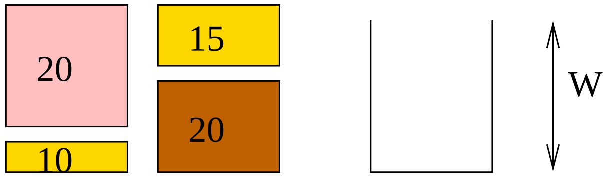
$\longrightarrow \sum$ на избраните c_i, d_i е $\leq 3 \neq 4$ на позиция j

Противоречие.

Така $F[B] = 1$

qed.

Пример Knapsack problem



- n обекта с тегла $w_i \in \mathbb{N}$ и печалба p_i
- Да се избере подмножество $\mathbf{x}$ от обекти
- така, че $\sum_{i \in \mathbf{x}} w_i \leq W$ и
- максимална печалба $\sum_{i \in \mathbf{x}} p_i$

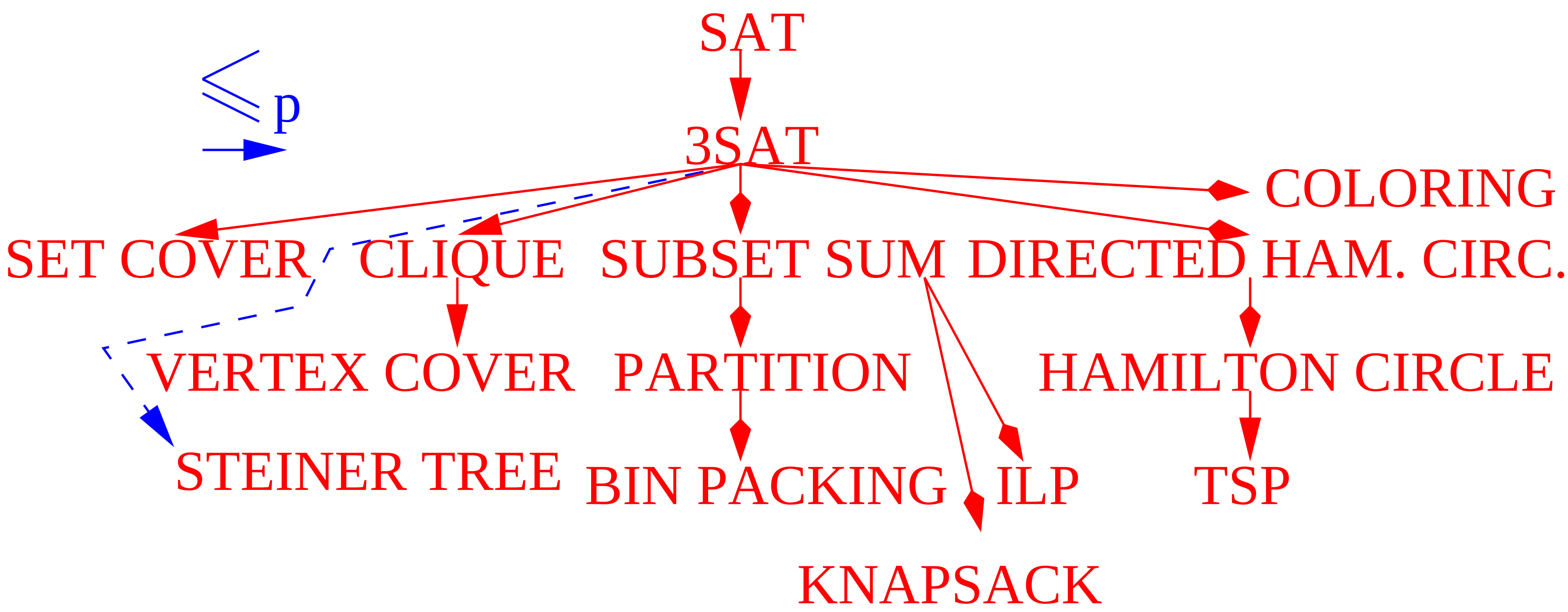
Knapsack problem

$\in$ **NP**: знаем

NP-трудна: Ще покажем SUBSET SUM$\leq_p$KNAPSACK

Очевидно SUBSET SUM е един пример
$(w_1, \ldots, w_n)$ със сума $= W$
на KNAPSACK
$\underbrace{(w_1, \ldots, w_n)}_{\text{тегла}}, \underbrace{(w_1, \ldots, w_n)}_{\text{печалба}}$ с
огараничение на теглата $\leq W$
ограничение на печалбата $\geq W$.

NP-пълни задачи



PARTITION

Дадено: n обекта с тегла $w_i \in \mathbb{N}$

Въпрос:

Има ли подмножество M на $\{1,\ldots,n\}$,

такова, че $\sum_{i \in M} w_i = \sum_{i \notin M} w_i$?

in **NP**, тъй като ...

Д-во SUBSET SUM$\leq_p$PARTITION

Дадено: една задача за SUBSET SUM

$S = (w_1, \ldots, w_k), W$.

Нека $M := \sum_{i=1}^{k} w_i$.

Да разгледаме следната задача PARTITION

$P = (w_1, \ldots, w_k, M - W + 1, W + 1)$.

Общо тегло: $M + M - W + 1 + W + 1 = 2(M + 1)$.

Ще покажем: P има решение $\Leftrightarrow S$ има решение.

SUBSET SUM $S = (w_1, \ldots, w_k), W.$ $M := \Sigma_{i=1}^{k} w_i.$

PARTITION $P = (w_1, \ldots, w_k, M - W + 1, W + 1).$ $(\Sigma = 2(M+1)).$

P има решение $\longrightarrow S$ има решение:

Нека J е решение за P, т.е., $\sum_{j \in J} P[j] = M + 1.$

Или $M - W + 1$ или $W + 1$ е избрано:

$M - W + 1 + W + 1 = M + 2 > M + 1$, $\Sigma_{i=1}^{k} w_i = M < M + 1$

1. Случай Ако $M - W + 1$ е избрано:

Тогава $\sum_{j \in J \cap \{1,\ldots,k\}} P[j] = M + 1 - (M - W + 1) = W.$

Така $J \cap \{1, \ldots, k\}$ е решение за S.

2. Случай Ако $W + 1$ е избрано:

Тогава $\sum_{j \in J \cap \{1,\ldots,k\}} P[j] = M + 1 - (W + 1) = M - W.$

Следователно $\{1, \ldots, k\} \setminus J$ е решение за S.

SUBSET SUM $S = (w_1, \ldots, w_k), W$.

$M := \sum_{i=1}^{k} w_i$.

PARTITION $P = (w_1, \ldots, w_k, M - W + 1, W + 1)$.

Общо тегло: $2(M+1)$.

S има решение $\longrightarrow P$ има решение:

Нека I е решение S, т.е., $\sum_{i \in I} w_i = W$.

Тогава $J := I \cup \{k+1\}$ е решение за P

$$\sum_{j \in J} P[j] = \sum_{i \in I} w_i + M - W + 1 = W + M - W + 1 = M + 1$$

Пакетиране на сладки-BIN PACKING

Дадено:

Кутии от сладки с размер $b \in \mathbb{N}$.

Сладки с размери $w_1, \ldots, w_n \in \mathbb{N}$

Брой кутии k.

Въпрос:

Може ли да сложим сладките в кутии?, т.е.,

$$\exists f : 1..n \longrightarrow 1..k : \forall j \in 1..k : \sum \{w_i : f(i) = j\} \leq b$$

BIN PACKING е **NP**-пълна

Очевидно BIN PACKING е в **NP**.

От друга старна PARTITION$\leq_p$BIN PACKING (Специален пример):

$$(w_1, \ldots, w_k) \mapsto \begin{cases} \text{размер на кутиите:} & b = \frac{1}{2}\sum_{i=1}^{k} w_i \\ \text{брой:} & k = 2 \\ \text{тегла:} & w_1, \ldots, w_k \end{cases}$$

Целочислено линейно програмиране (ILP)

Дадено: вектор на променливите $\mathbf{x} = (x_1, \ldots, x_n)$,
Множество от условията(constraints) от вида $\mathbf{a} \cdot \mathbf{x} R b$ с $R \in \{\leq, \geq, =\}$, $b \in \mathbb{Z}$, $a \in \mathbb{Z}^n$.

Въпрос:
Има ли остойностяване за $\mathbf{x}$ от $\mathbb{Z}^n$, така че всички условия да се удовлетворят?

Д-во за "‘ILP е **NP**-трудна"’

Ще покажем, че SUBSET SUM$\leq_p$ILP

Да разгледаме една задача SUBSET SUM $(w_1,\ldots,w_n),W$.

Тя е очевидно еквивалентна на ILP задача

$\{\mathbf{x}\cdot(w_1,\ldots,w_n)=W\}\cup$

$\{x_1\geq 0,\ldots,x_n\geq 0\}\cup$

$\{x_1\leq 1,\ldots,x_n\leq 1\}$

Д-во ILP$\in$ **NP**?

Избираме n стойности на променливите.
Проверяваме дали са удовлетворени условията.

Проблем: дали са полиномиално много бита за остойностяване на променливите?

[Papadimitriou 1981] Да.

Основата на доказателството: Правило на Крамер колко много са битовете.

DIRECTED HAMILTON CYCLE (DHC)

$\in$ **NP**

$M :=$
$\{G = (V, E \subseteq V \times V) : \exists C \subseteq E : |C| = |V|, C$ е прост цикъл$\}$

Избираме C

Проверяваме дали всеки възел е посетен веднъж.

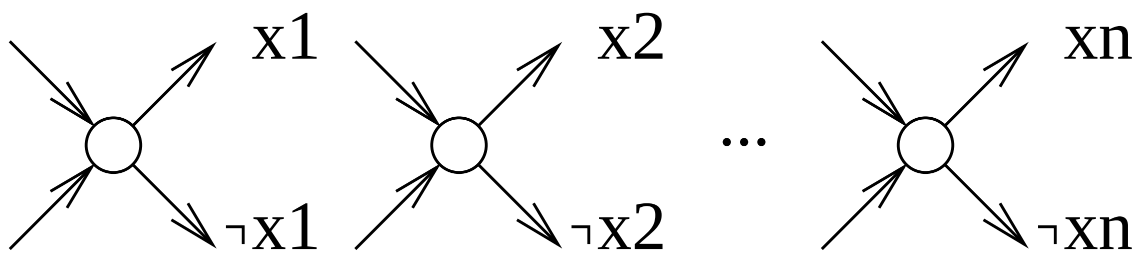
Д-во на "‘DHC е **NP**-трудна"’

Ще покажем 3SAT$\leq_p$HAMILTON-CYCLE.

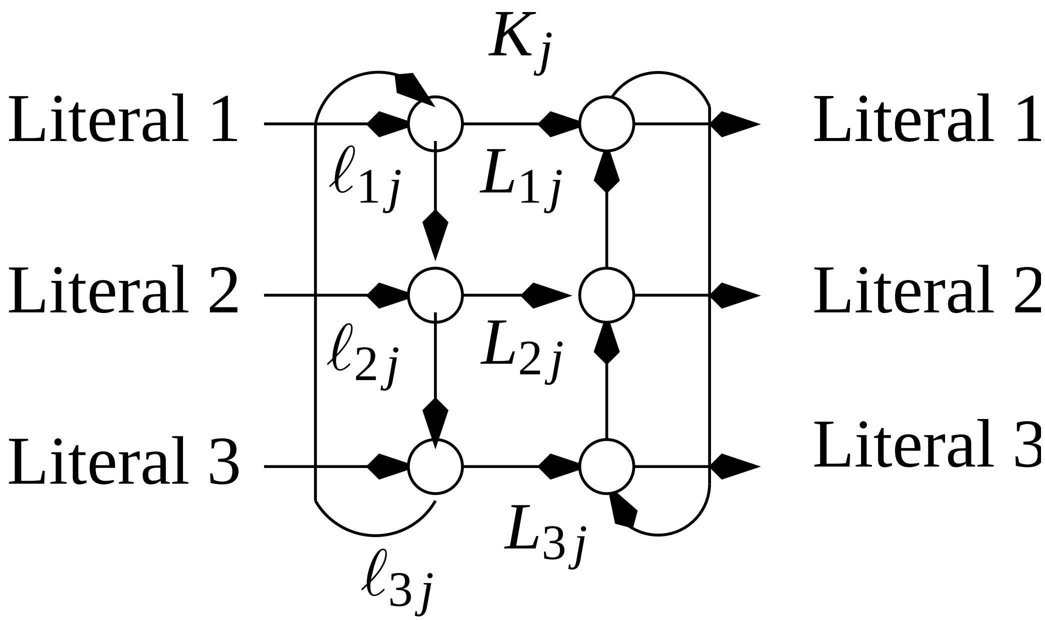
Нека $F = (z_{11} \vee z_{12} \vee z_{13}) \wedge \cdots \wedge (z_{m1} \vee z_{m2} \vee z_{m3})$ с литерали $z_{ij} \in \{x_1, \ldots, x_n\} \cup \{\neg x_1, \ldots, \neg x_n\}$.
Построяваме графа $G = (V, E)$ с $n + 6m$ възела.

- Един възел за променлива
- Gadget с 6 възела за клауза.

Един възел за променлива



Gadget K_j с 6 възела за всяка клауза

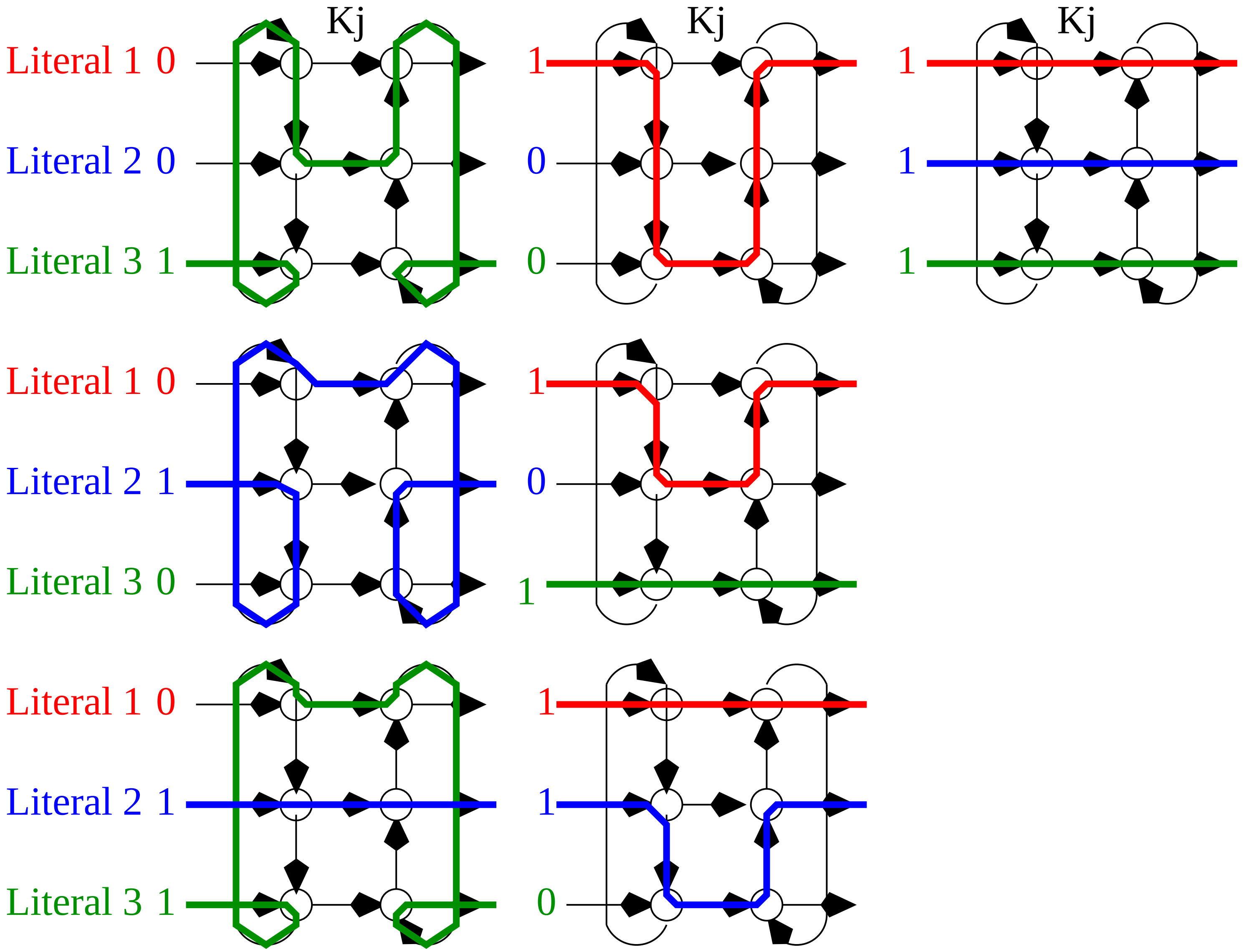


Хамилтоновият цикъл обикаля един Gadget 1–3×.

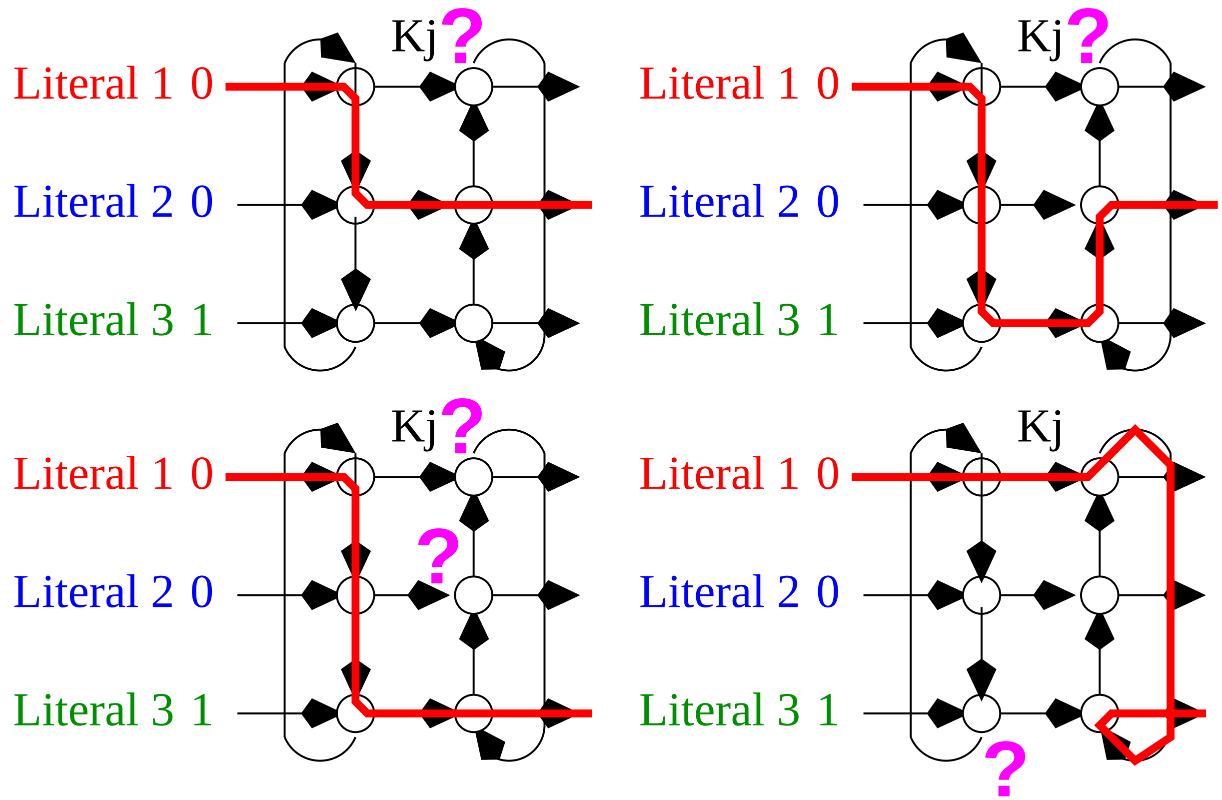
1× за всеки удовлетворен литерал.

Влиза в даден ред и напуска в даден момент на същия ред.

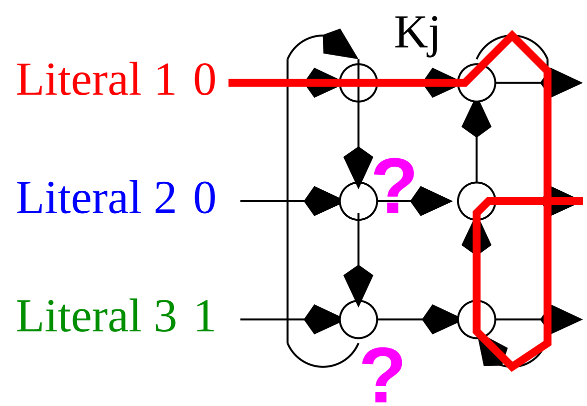
"‘Алокирани"’ Хамилтонови цикли



Невъзможни Хамилтонови цикли



Gadgets ще се напускат винаги
на същия ред
в който влизат.



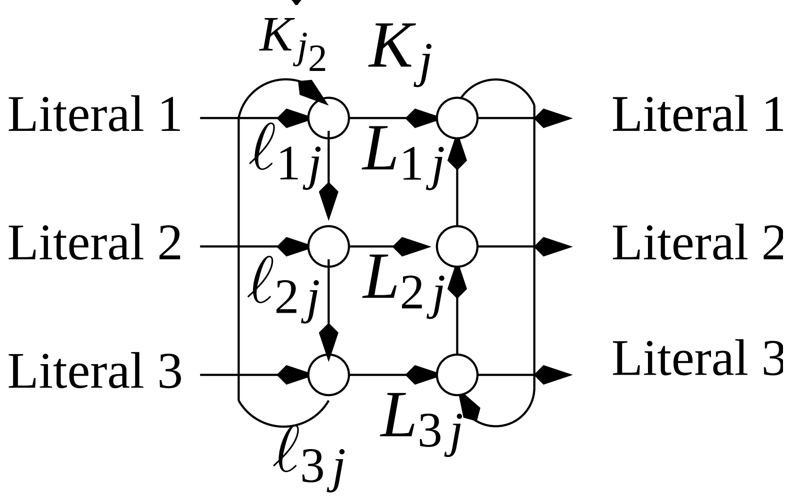
Свързване на Gadgets

Нека $\{j_1, \ldots, j_k\}$ е множеството индексите на клаузите, в които литерарът z_i ($z_i = x_i$ или $z_i = \neg x_i$) участва.

Нека $\{p_1, \ldots, p_k\}$ са съответните позиции, в които z_i участва.

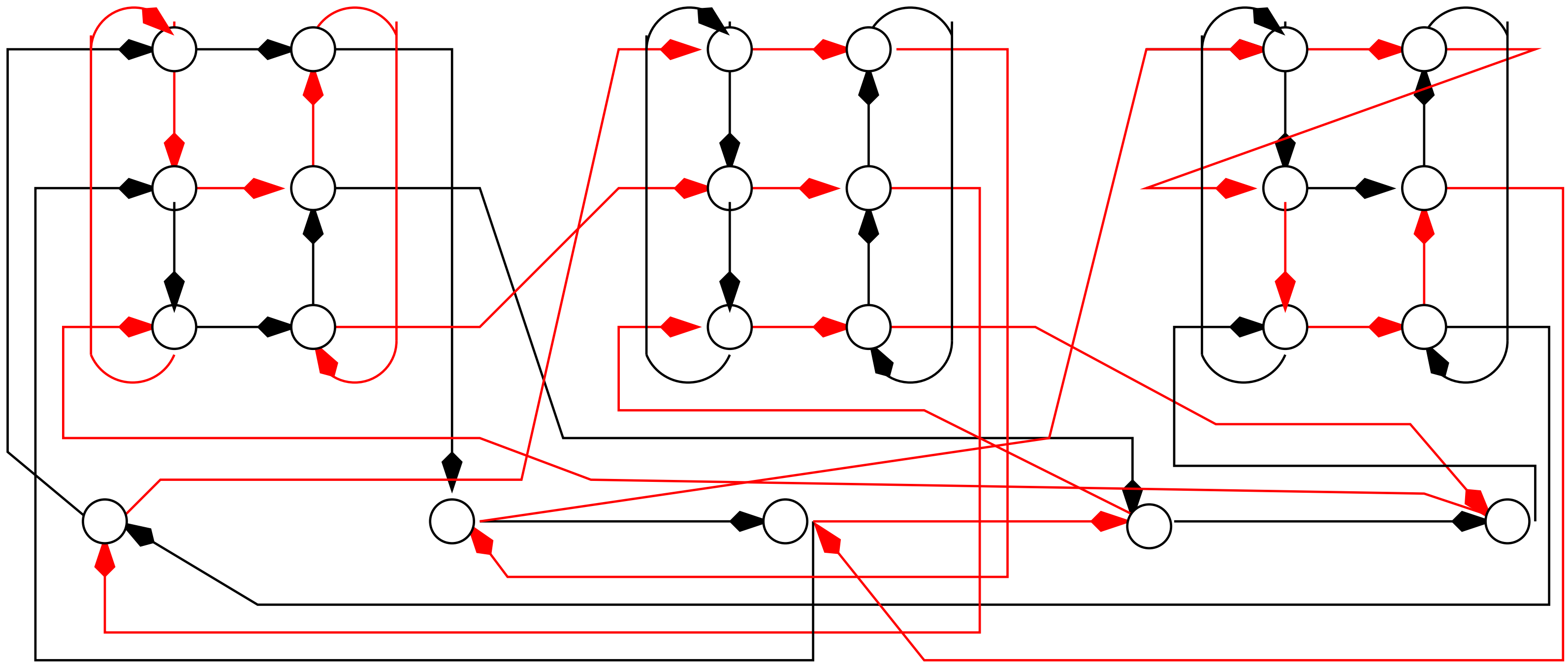
Тогава има път

$$x_i \to \underbrace{\ell_{p_1 j_1} \to L_{p_1 j_1}}_{K_{j_1}} \to \underbrace{\ell_{p_2 j_2} \to L_{p_2 j_2}}_{K_{j_2}} \to \cdots \to \underbrace{\ell_{p_k j_k} \to L_{p_k j_k}}_{K_{j_k}} \to x_{i+1 \bmod n}$$



Пример

$$F = (x_1 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee x_5 \vee x_4) \wedge (\neg x_2 \vee \neg x_2 \vee \neg x_5)$$



Стойности: $x_1 = x_2 = 0,\ x_3 = x_4 = x_5 = 1$

F е изпълнима$\longrightarrow$ Задачата за Хам. цикъл има решение

Да разгледаме оценка B, при която $F[B]=1$.

$x_i = 1$: Между възлите x_i и $x_i+1 \bmod n$ минаващите Gadgets $\{K_j : x_i \in$ клаузата $j\}$.

$x_i = 0$: Между възлите x_i и $x_i+1 \bmod n$ минаващите Gadgets $\{K_j : \neg x_i \in$ клаузата $j\}$.

- Всеки Gadget ще е поне веднъж посетен.
- Ние знаем, че всеки възел от Gadget ще се посети точно веднъж.
- Всеки възел за променлива ще се посети точно веднъж.

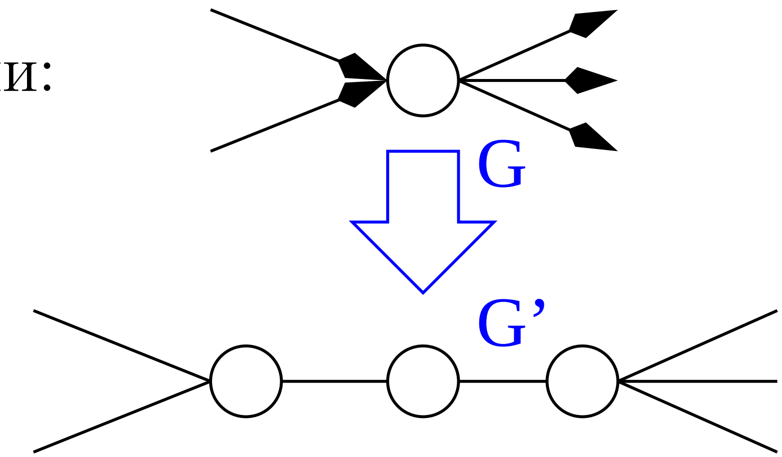
Задачата за Хам. цикъл има решение $\longrightarrow F$ е изпълнима

Да разгледаме Хам. цикъл C.

- Между x_i и x_{i+1} по C има само Gadgets такива, че в съответните клаузи или само x_i или само $\neg x_i$ се появяват.
- Ако са само x_i така, че всички тези съответващи клаузи ще се удовлетворят.
- Всеки Gadget ще се посети поне веднъж.
- Всички клаузи също ще се удовлетворят.

DHC$\leq_p$HAMILTON CYCLE

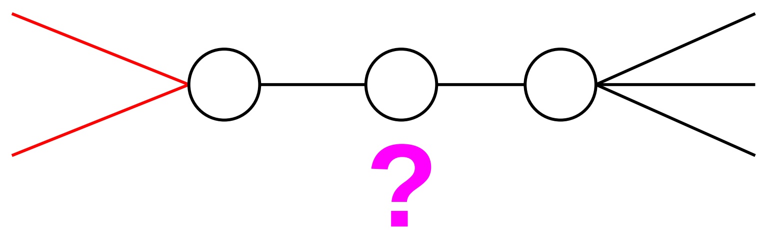
Gadget за възли:



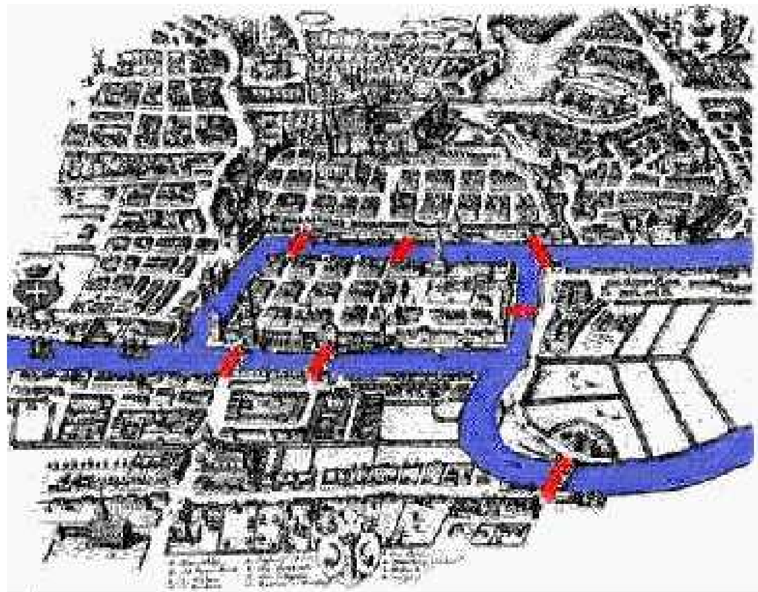
G има HC $\longrightarrow$ G' има HC (ясно).

G' има HC $\longrightarrow$ G има HC:

HC, който прескача Gadget не е възможен:



Допълнение: Ойлеров цикъл$\in$ P



$M := \{G = (V, E) \mid \exists C \subseteq E : C$ е цикъл при който всяка дъга е посетена точно веднъж$\}$.

Твърдение:[Euler 1736] G има Ойлеров цикъл $\Leftrightarrow$ G е свързан и от всеки възел излизат четен брой дъги.

Разгледани свеждания