

Základy složitosti a vyčíslitelnosti

NTIN090

Petr Kučera

2024/25 (8. přednáška)

Dnešní plán

- Problém **SPLNITELNOSTI** a znění Cookovy-Levinovy věty
- Polynomiální převoditelnost
- Převod **3-SAT** na **VRCHOLOVÉ POKRYTÍ**
- **NP**-úplnost
- Cookova-Levinova věta (**NP**-úplnost **SPLNITELNOSTI**)

Splnitelnost a Cookova-Levinova věta

Výroková formule

- Spočetná množina výrokových proměnných $\{x_0, x_1, \dots\}$
- Logické spojky $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$

Definice (Výroková formule)

- Proměnná je formule
- Jsou-li φ a ψ formule, pak $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$ a $\varphi \leftrightarrow \psi$ jsou též formule
- Nic jiného nejsou formule

Ohodnocení a splnitelnost

- **Ohodnocení** $\mathbf{a}$ přiřazuje výrokovým proměnným hodnoty **true/false** ($1/0, \top/\perp$)
- Ohodnocení $\mathbf{a}$ **splňuje formuli** φ pokud se $\varphi(\mathbf{a})$ vyhodnotí na **true**
- Říkáme též, že jde o **model** formule φ
- **Splnitelná** formule má alespoň jeden model
- **Nesplnitelná** formule nemá žádný model

Příklad

Formule

$$(x \vee y) \wedge (\neg x \vee z) \wedge (\neg y \vee \neg z)$$

je splněná ohodnocením $\{x \mapsto 1, y \mapsto 0, z \mapsto 1\}$

Konjunktivní normální forma

Literál je proměnná x nebo její negace $\neg x$

Klauzule je disjunkce literálů

- Například $x \vee \neg y \vee \neg z$
- Prázdná klauzule je sporná (false, $\perp$)

KNF formule je v **konjunktivní normální formě** pokud jde o konjunkci klauzulí

Příklad

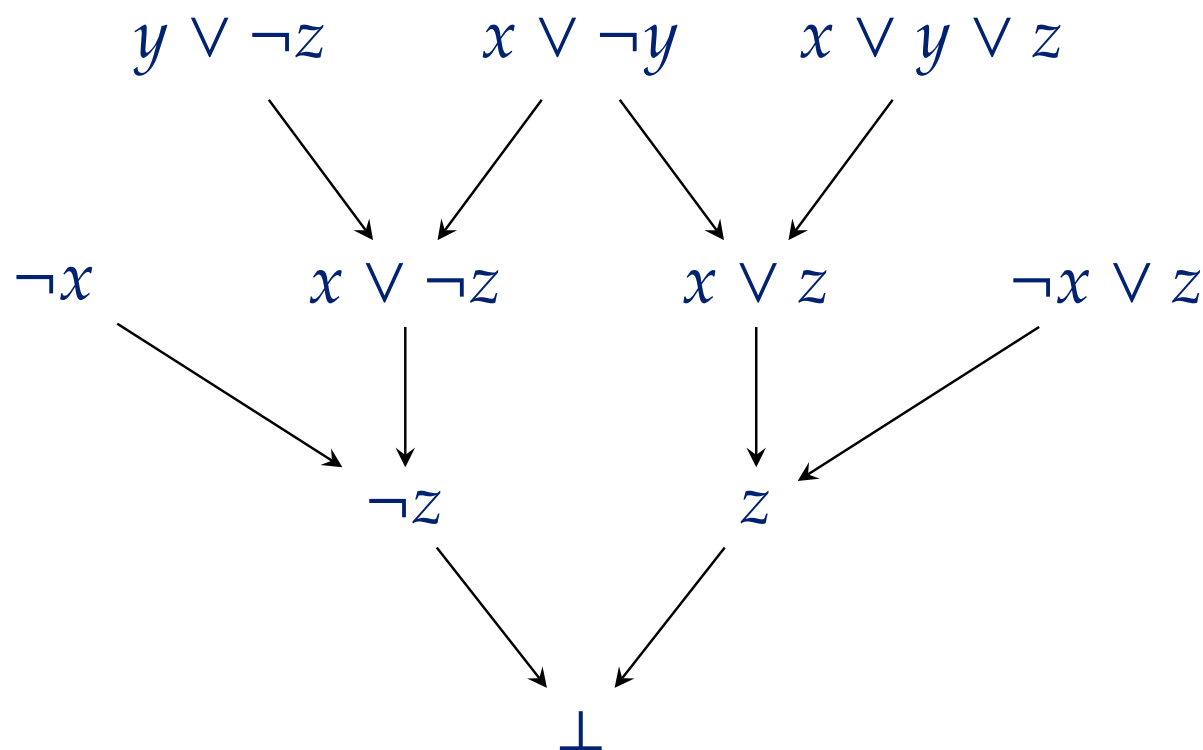
$$\varphi = \neg x \wedge (y \vee \neg z) \wedge (x \vee \neg y) \wedge (x \vee y \vee z) \wedge (\neg x \vee z)$$

Nesplnitelná KNF

Následující formule v KNF je nesplnitelná

$$\varphi = \neg x \wedge (y \vee \neg z) \wedge (x \vee \neg y) \wedge (x \vee y \vee z) \wedge (\neg x \vee z)$$

Nesplnitelnost lze ukázat **rezolucí**



Splnitelnost

Nesplnitelnost je jednodušší, protože stačí zjistit, jestli tam neexistuje spor.

SPLNITELNOST (SAT)

Instance: Formule φ v KNF.

Otázka: Je φ splnitelná?

SAT patří do NP

*↳ Tady bych musel
potřebovat vykoušet
všechny možné ohodnocení.*

- V polynomiálním čase lze ověřit, zda dané ohodnocení splňuje danou formuli

SAT je obecně jeden z nýtčestých problémů.



Patří **SAT** do P?

Cookova-Levinova věta

Věta (Cookova-Levinova)

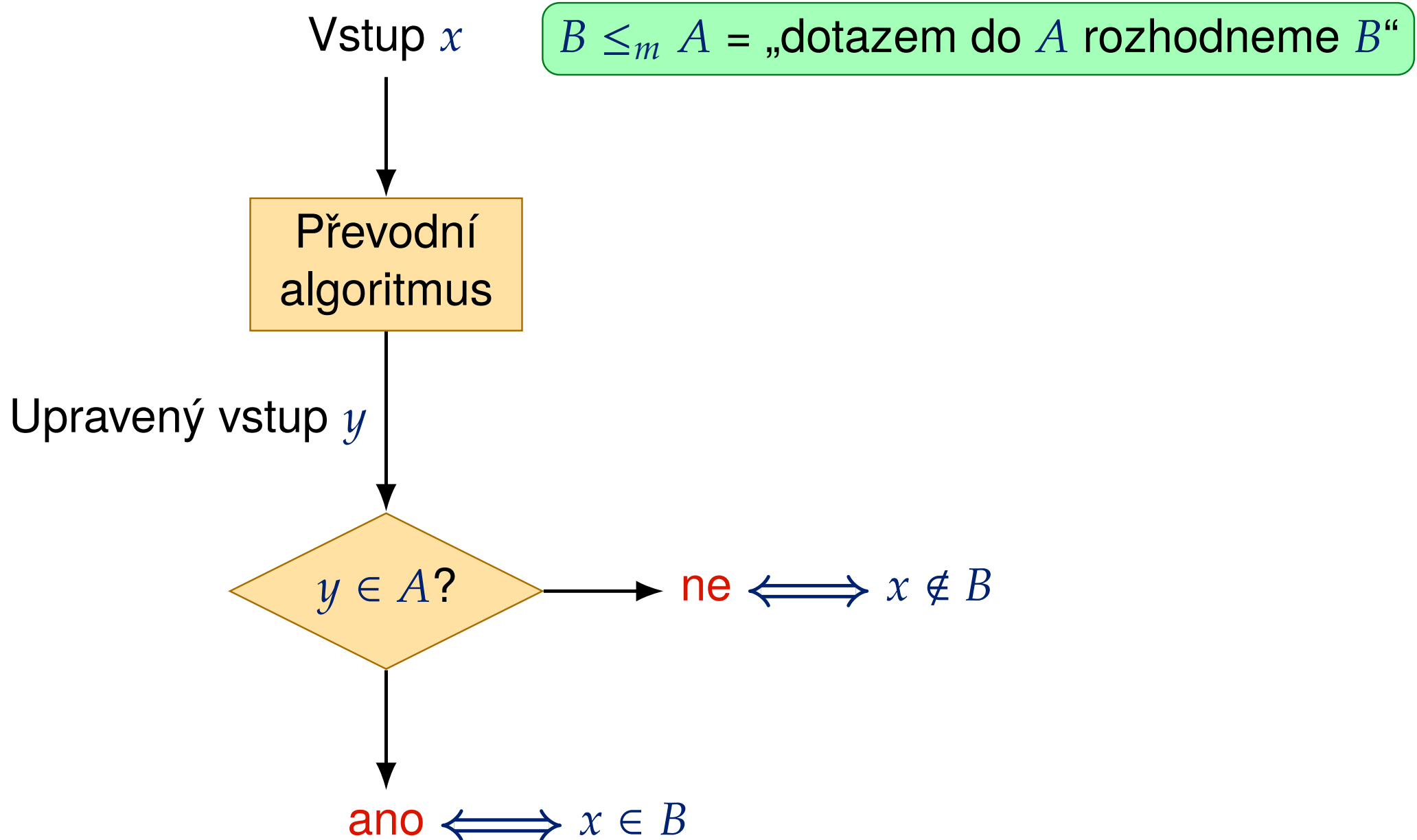
SAT patří do P právě když $P = NP$.

Idea důkazu:

- 1 Zavedeme pojem polynomiální převoditelnosti
 - m -převoditelnost, kde převodní algoritmus pracuje v polynomiálním čase
- 2 Zavedeme pojem NP -úplného problému
 - Nejtěžší problémy v NP vzhledem k polynomiální převoditelnosti
 - Je-li nějaký NP -úplný problém v P , pak $P = NP$
- 3 Ukážeme, že **SAT** je NP -úplný problém

Polynomiální převoditelnost

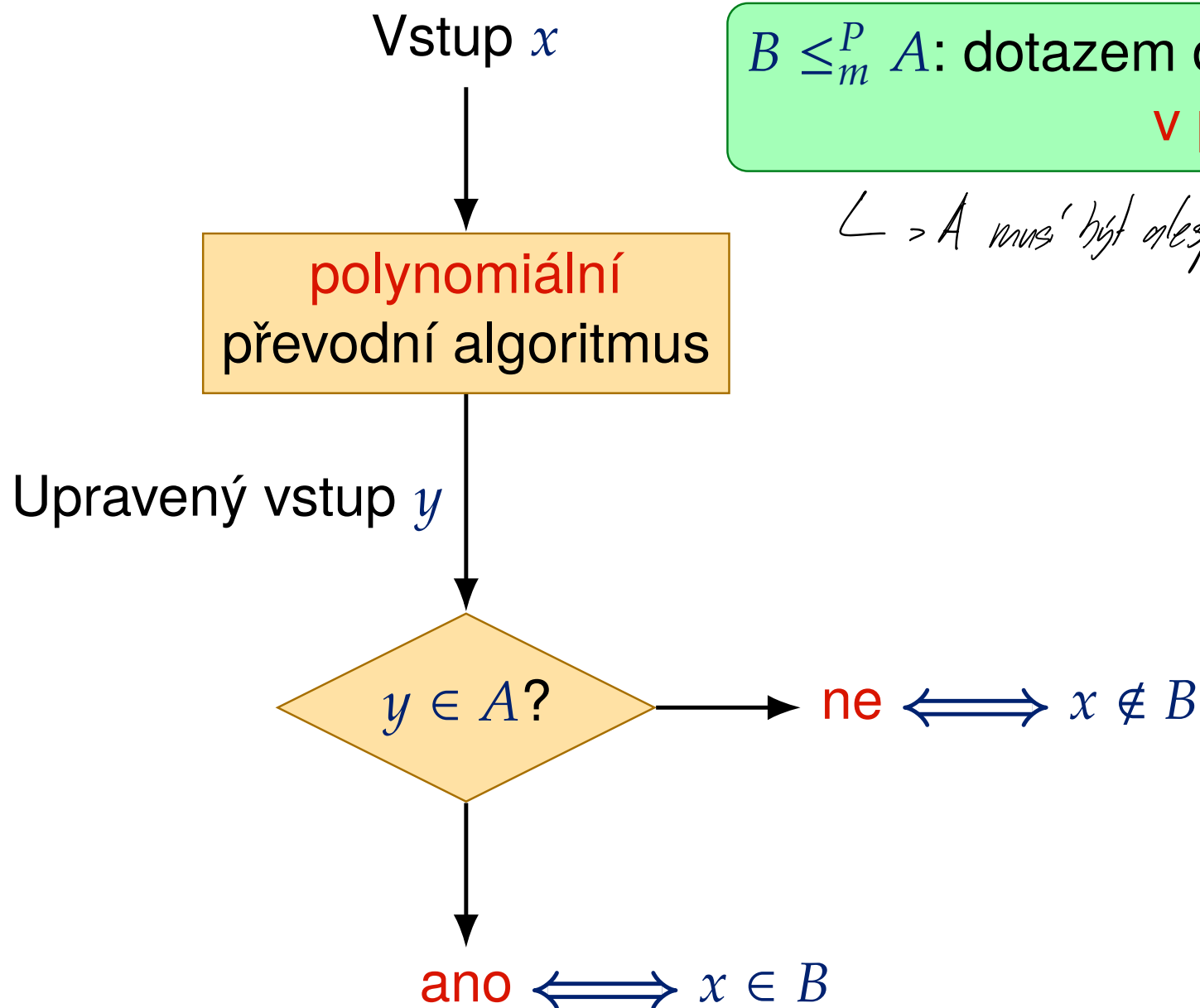
m -převoditelnost (princip)



Polynomiální převoditelnost (princip)

$B \leq_m^P A$: dotazem do A rozhodneme B
v polynomiálním čase

$\hookrightarrow A$ musí být alespoň tak těžký jako B .

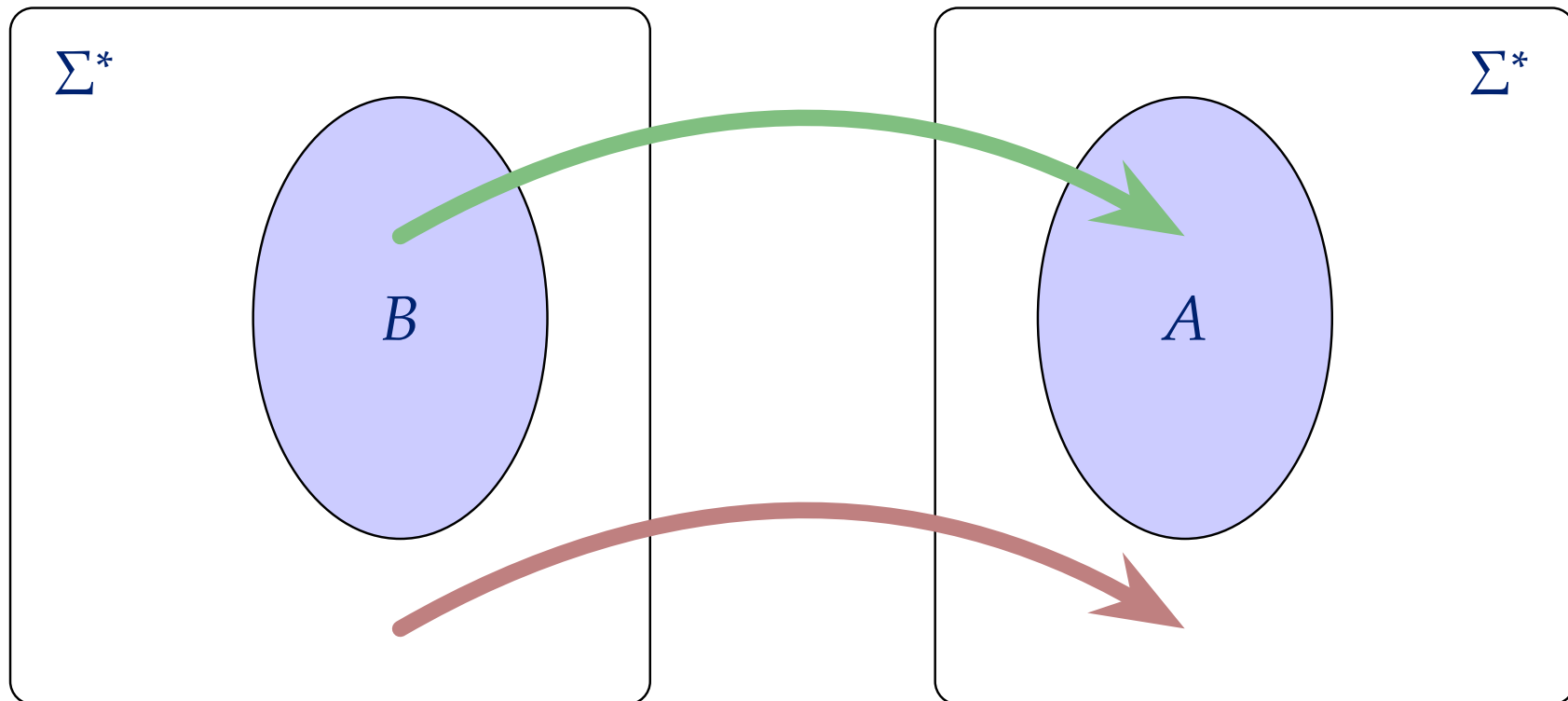


Polynomiální převoditelnost (definice)

Definice

Jazyk B je **polynomiálně převoditelný** na jazyk A ($B \leq_m^P A$), pokud existuje funkce $f : \Sigma^* \rightarrow \Sigma^*$ vyčíslitelná v polynomiálním čase, pro kterou platí

$$(\forall w \in \Sigma^*) [w \in B \iff f(w) \in A]$$



Polynomiální převoditelnost (vlastnosti)

Definice

Jazyk B je **polynomiálně převoditelný** na jazyk A ($B \leq_m^P A$), pokud existuje funkce $f : \Sigma^* \rightarrow \Sigma^*$ vyčíslitelná v polynomiálním čase, pro kterou platí

$$(\forall w \in \Sigma^*) [w \in B \iff f(w) \in A]$$

- $\leq_m^P$ je reflexivní a tranzitivní (**kvaziuspořádání**)
 - Je-li $B \leq_m^P A$ a $A \in P$, pak $B \in P$.
 - Je-li $B \leq_m^P A$ a $A \in NP$, pak $B \in NP$.
- } *A určuje obtížnost úlohy.*

Neznáme-li žádný polynomiální algoritmus pro B , pak neumíme zkonstruovat ani polynomiální algoritmus pro A .

Polynomiální převoditelnost (reflexivita)

Lemma (Reflexivita polynomiální převoditelnosti)

Pro každý jazyk A platí, že $A \leq_m^P A$.

Důkaz.

- Funkce identity $\text{id}(x) = x$ je vyčíslitelná v polynomiálním čase
- Pro každý řetězec $x \in \Sigma^*$ platí

$$x \in A \Leftrightarrow \text{id}(x) \in A$$



Polynomiální převoditelnost (tranzitivita)

Lemma (Tranzitivita polynomiální převoditelnosti)

Pro každé tři jazyky A , B a C : $A \leq_m^P B \wedge B \leq_m^P C \implies A \leq_m^P C$

- $A \leq_m^P B$ funkcí g
- $B \leq_m^P C$ funkcí h
- Definujme funkci $f(x) = h(g(x))$
 - f je vyčíslitelná v polynomiálním čase
- Pro každý řetězec $x \in \Sigma^*$ platí

$$x \in A \underbrace{\iff g(x) \in B}_{A \leq_m^P B} \underbrace{\iff h(g(x)) \in C}_{B \leq_m^P C} \underbrace{\iff f(x) \in C}_{f(x)=h(g(x))}$$

- $A \leq_m^P C$ funkcí f

Polynomiální převoditelnost srovnává dle obtížnosti

Lemma

Pokud $A \in P$ a $B \leq_m^P A$, pak $B \in P$.

- Uvažme TS M_A , který rozhoduje A v polynomiálním čase
- Popíšeme TS M_B , který rozhoduje B v polynomiálním čase

Výpočet M_B se vstupem x

```
1  $y \leftarrow f(x)$            //  $f$  ukazuje, že  $B \leq_m^P A$ 
2 Pust'  $M_A(y)$ 
3 if  $M_A$  přijal then
4   |   přijmi
5 else
6   |   odmítni
```

Polynomiální převoditelnost srovnává dle obtížnosti

Lemma

Pokud $A \in \text{NP}$ a $B \leq_m^P A$, pak $B \in \text{NP}$.

- Uvažme polynomiální verifikátor $V_A(x, y)$ pro jazyk A
- Popíšeme polynomiální verifikátor $V_B(x, y)$ pro jazyk B

Výpočet verifikátoru V_B se vstupem x a certifikátem y

```
1  $z \leftarrow f(x)$                                 //  $f$  ukazuje, že  $B \leq_m^P A$ 
2 Pust'  $V_A(z, y)$ 
3 if  $V_A$  přijal then
4   | přijmi
5 else
6   | odmítni
```

3-KNF formule φ je v **3-KNF**, pokud je v KNF a každá klauzule obsahuje právě 3 literály

3-SAT

Instance: Formule φ v 3-KNF.

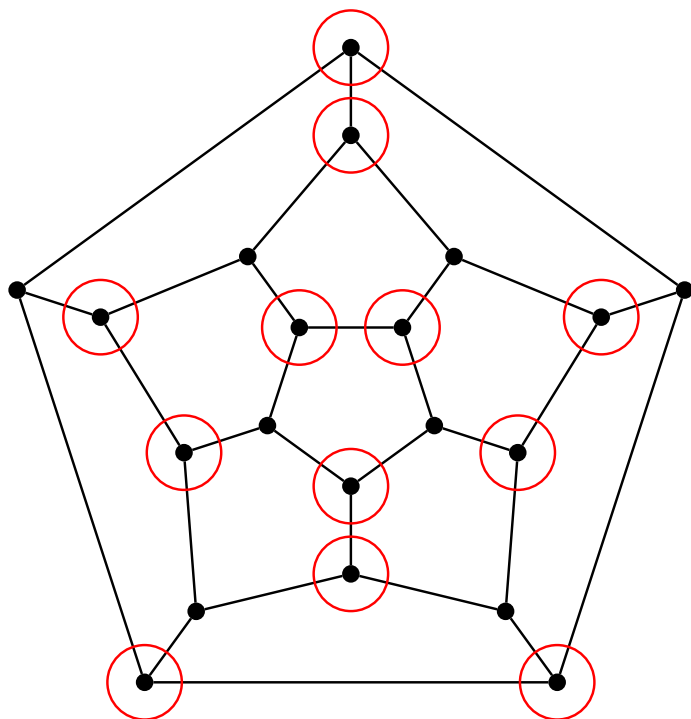
Otázka: Je φ splnitelná?

Vrcholové pokrytí

VRCHOLOVÉ POKRYTÍ

Instance: Neorientovaný graf $G = (V, E)$ a celé číslo $k \geq 0$.

Otázka: Existuje množina vrcholů $S \subseteq V$ velikosti nejvýš k , která obsahuje alespoň jeden vrchol z každé hrany $\{u, v\} \in E$ (tedy $\{u, v\} \cap S \neq \emptyset$)?

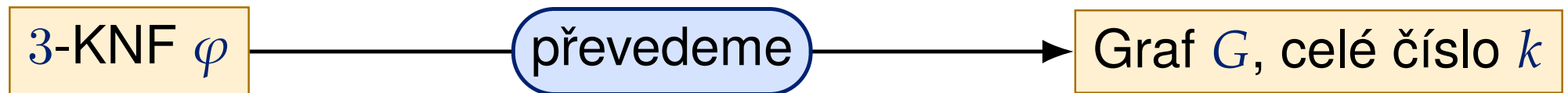


Desetistěn má vrcholové
pokrytí velikosti $k = 12$

3-SAT a VRCHOLOVÉ POKRYTÍ

Věta

3-SAT je polynomiálně převoditelný na VRCHOLOVÉ POKRYTÍ.



φ je splnitelná $\longleftrightarrow$ G má vrcholové pokrytí velikosti k

Převod 3-SAT na VRCHOLOVÉ POKRYTÍ

- Nechť φ je 3-KNF, která má
 - n proměnných $x_1, \dots, x_n$
 - m klauzulí $C_1, \dots, C_m$
 - Každá klauzule má právě 3 literály
- Popíšeme, jak sestavit graf $G = (V, E)$ a číslo k

Příklad

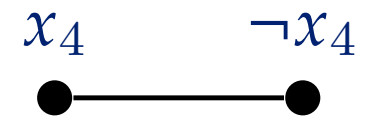
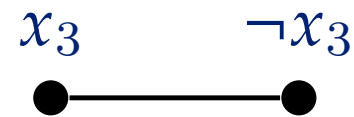
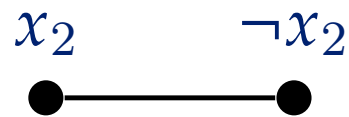
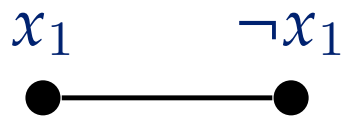
Uvažme například 3-KNF

$$\begin{aligned} \varphi = & \overbrace{(x_1 \vee \neg x_2 \vee x_3)}^{C_1} \wedge \overbrace{(x_2 \vee \neg x_3 \vee x_4)}^{C_2} \\ & \wedge \underbrace{(\neg x_1 \vee x_2 \vee x_4)}_{C_3} \wedge \underbrace{(\neg x_2 \vee x_3 \vee \neg x_4)}_{C_4} \end{aligned}$$

Konstrukce grafu (krok 1)

Pro každou proměnnou x_i , $i = 1, \dots, n$

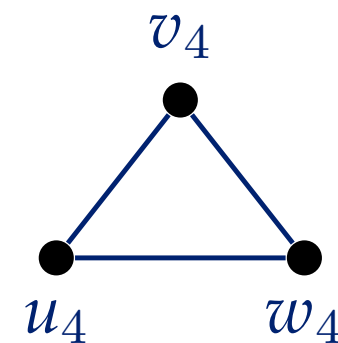
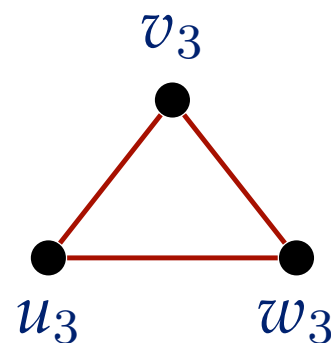
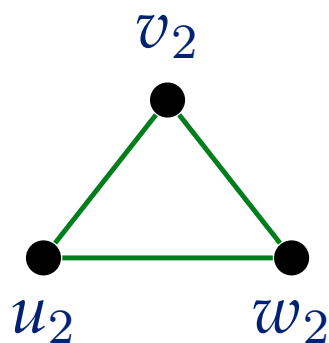
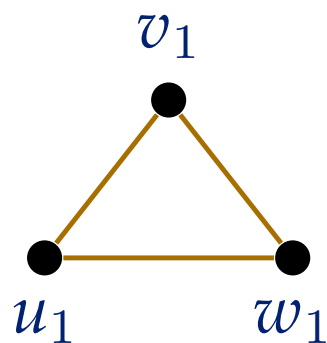
- přidáme dva vrcholy pro literály x_i , $\neg x_i$ a hranu $\{x_i, \neg x_i\}$



Konstrukce grafu (krok 2)

Pro každou klauzuli C_j , $j = 1, \dots, m$

- Přidáme trojúhelník na nově přidané trojici vrcholů u_j, v_j, w_j



$$x_1 \vee \neg x_2 \vee x_3$$

$$x_2 \vee \neg x_3 \vee x_4$$

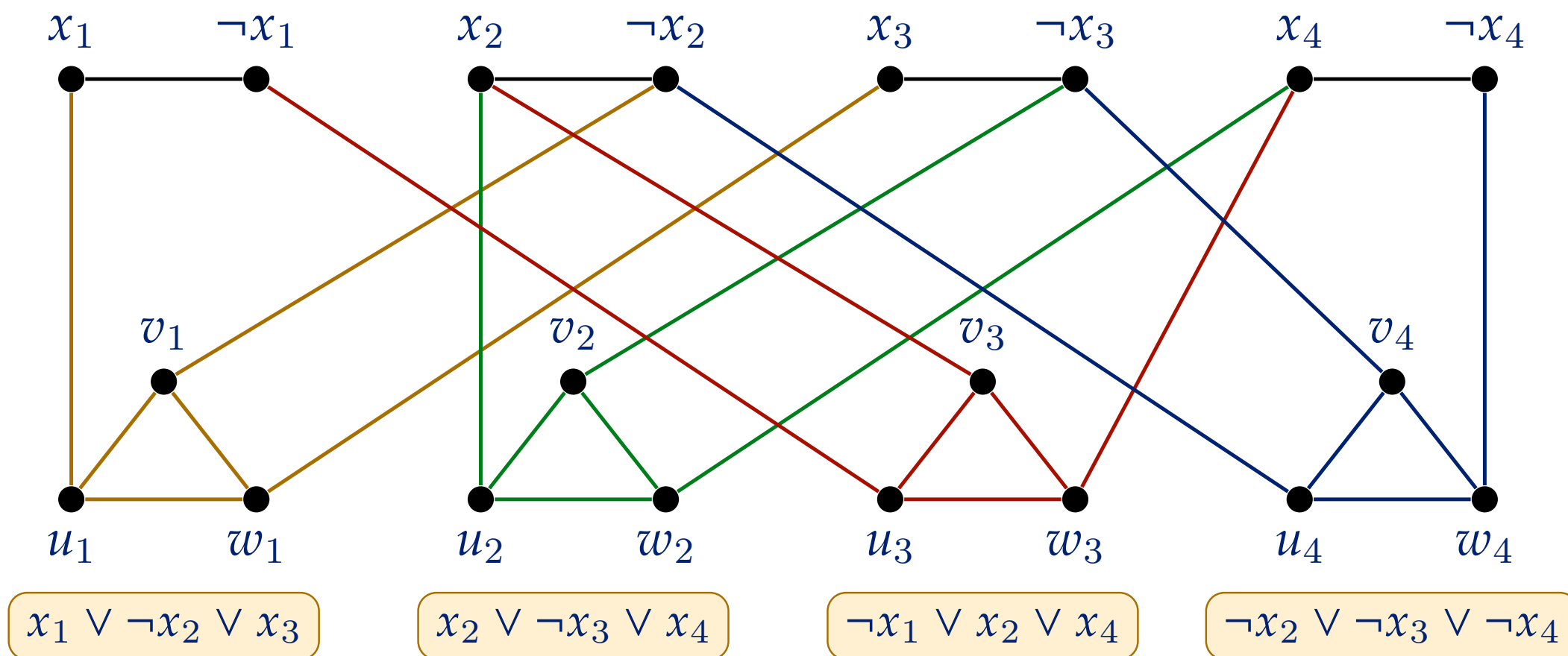
$$\neg x_1 \vee x_2 \vee x_4$$

$$\neg x_2 \vee \neg x_3 \vee \neg x_4$$

Konstrukce grafu (krok 3)

Pro každou klauzuli $C_j = (l_1 \vee l_2 \vee l_3)$ s literály l_1, l_2, l_3

- Přidáme hrany $\{u_j, l_1\}, \{v_j, l_2\}, \{w_j, l_3\}$



$$k = 2m + n$$

- 1 vrchol za každou hranu $\{x_i, \neg x_i\}$, $i = 1, \dots, n$
- 2 vrcholy za každý trojúhelník $\{u_j, v_j, w_j\}$, $j = 1, \dots, m$
- Vrcholové pokrytí musí obsahovat alespoň $2m + n$ vrcholů

Graf G má vrcholové pokrytí velikosti $k = 2m + n$, právě když formule φ je splnitelná.

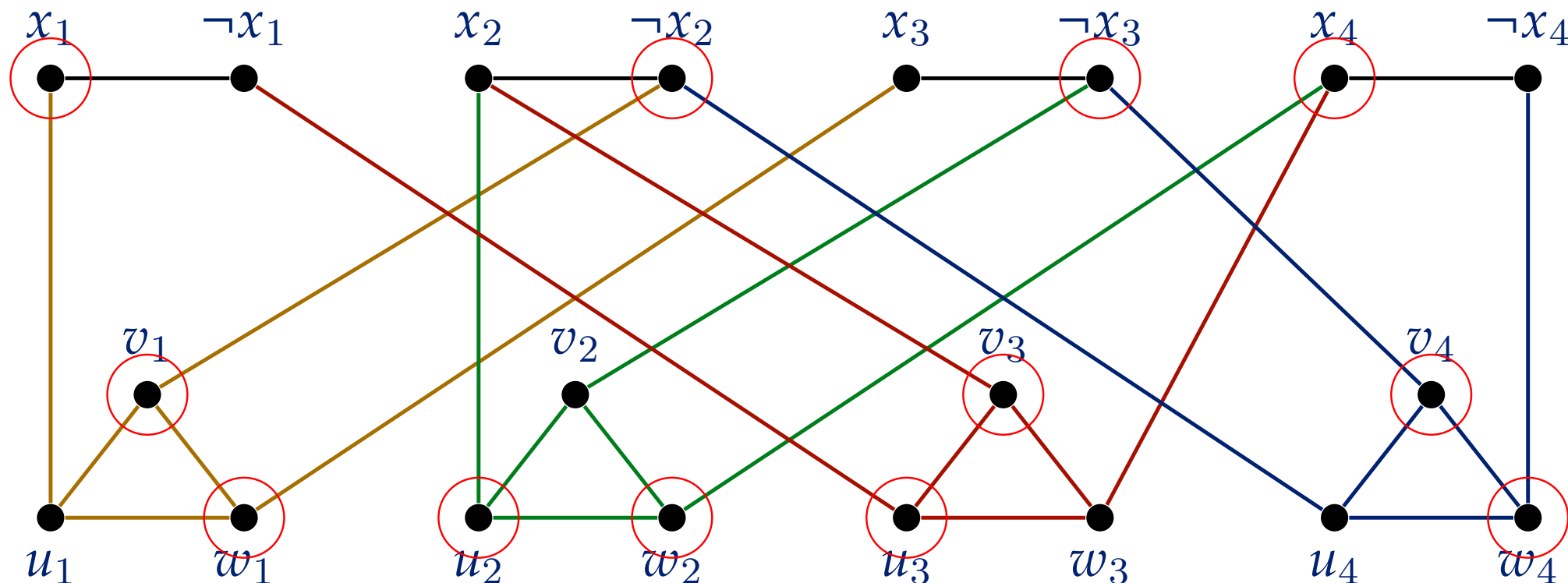
Důkaz „ $\implies$ “

- Nechť $S \subseteq V$ je vrcholové pokrytí velikosti $k = 2m + n$
- Každá hrana $\{x_i, \neg x_i\}$, $i = 1, \dots, n$ je pokryta právě jedním vrcholem
- Definujeme ohodnocení $\mathbf{a}$ tak, že pro každý index $i = 1, \dots, n$

$$\mathbf{a}(x_i) = \begin{cases} 1 & x_i \in S \\ 0 & \neg x_i \in S \end{cases}$$

Model daný vrcholovým pokrytím

Model daný pokrytím: $\{x_1 \mapsto 1, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 1\}$



$x_1 \vee \neg x_2 \vee x_3$

$x_2 \vee \neg x_3 \vee x_4$

$\neg x_1 \vee x_2 \vee x_4$

$\neg x_2 \vee \neg x_3 \vee \neg x_4$

Každá klauzule obsahuje splněný literál

$\mathbf{a}$ splňuje φ

- Uvažme klauzuli $C_j, j \in \{1, \dots, m\}$
- Předpokládejme, že $C_j = (l_1 \vee l_2 \vee l_3)$
- S obsahuje dva vrcholy trojúhelníku $\{u_j, v_j, w_j\}$
- Jeden vrchol trojúhelníku $\{u_j, v_j, w_j\}$ není v S
- Nechť $u_j \notin S$
 - Případy se zbylými dvěma vrcholy jsou symetrické
- Uvažme hranu $\{u_j, l_1\}$
- Protože $u_j \notin S$, l_1 musí být v S

$\Rightarrow \mathbf{a}(l_1) = 1$

$\Rightarrow C_j$ je splněna ohodnocením $\mathbf{a}$

Ohodnocení $\mathbf{a}$ splňuje každou klauzuli $C_j \in \varphi$, jde o model.

Důkaz „ $\Leftarrow$ “

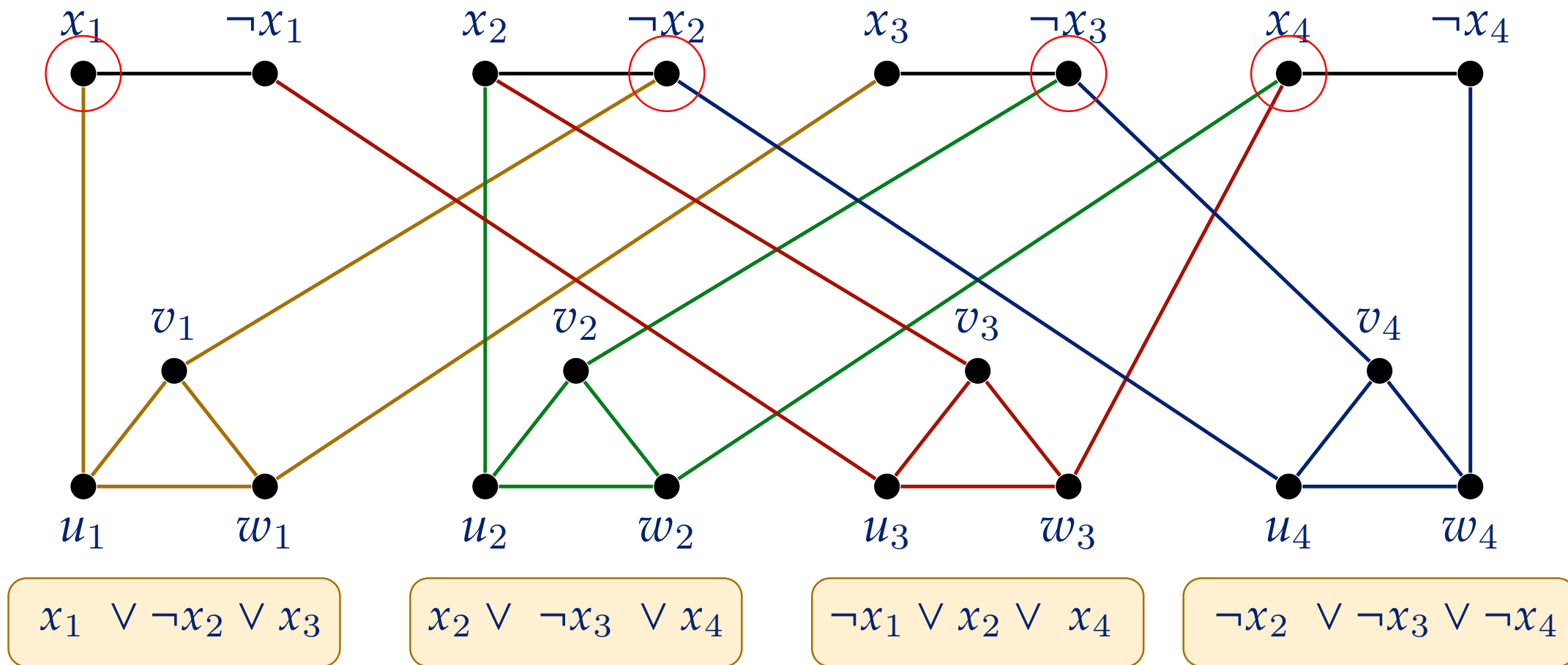
K danému splňující ohodnocení $\mathbf{a}$ definujeme množinu vrcholů S takto

- Pro každé $i = 1, \dots, n$
 - Přidáme do S vrchol x_i pokud $\mathbf{a}(x_i) = 1$
 - Přidáme do S vrchol $\neg x_i$ pokud $\mathbf{a}(x_i) = 0$
- Pro každý trojúhelník $\{u_j, v_j, w_j\}$, $j = 1, \dots, m$
 - Uvažme klauzuli $C_j = (l_1 \vee l_2 \vee l_3)$
 - Jeden z literálů je splněný ohodnocením $\mathbf{a}$
 - Předpokládejme, že literál l_1 je splněný
 - Případy s ostatními literály jsou symetrické
 - Hrana $\{l_1, u_j\}$ je již pokrytá vrcholem $l_1 \in S$
 - Zbylé vrcholy v_j, w_j přidáme do S

S je vrcholové pokrytí G velikosti $k = 2m + n$.

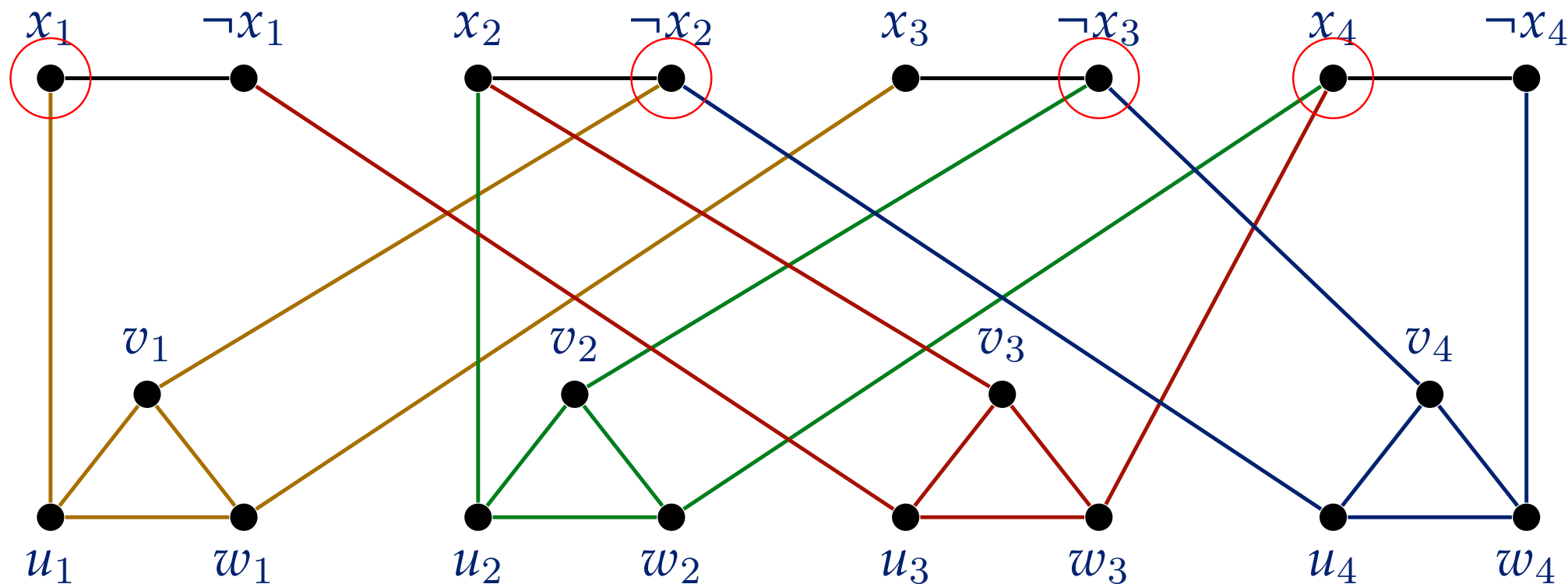
Vrcholové pokrytí dané modelem

Předpokládejme model: $\{x_1 \mapsto 1, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 1\}$



Vrcholové pokrytí dané modelem

Předpokládejme model: $\{x_1 \mapsto 1, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 1\}$



$x_1 \vee \neg x_2 \vee x_3$

$x_2 \vee \neg x_3 \vee x_4$

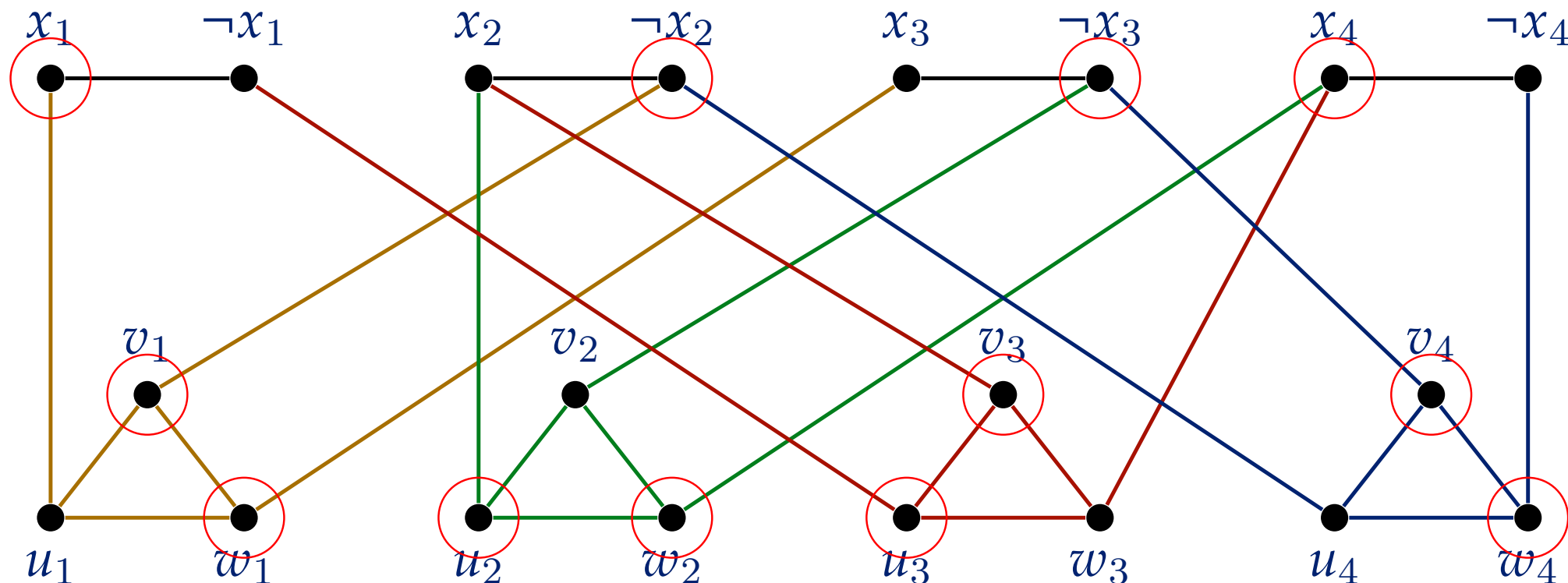
$\neg x_1 \vee x_2 \vee x_4$

$\neg x_2 \vee \neg x_3 \vee \neg x_4$

V každé klauzuli vybereme splněný literál

Vrcholové pokrytí dané modelem

Předpokládejme model: $\{x_1 \mapsto 1, x_2 \mapsto 0, x_3 \mapsto 0, x_4 \mapsto 1\}$



$x_1 \vee \neg x_2 \vee x_3$

$x_2 \vee \neg x_3 \vee x_4$

$\neg x_1 \vee x_2 \vee x_4$

$\neg x_2 \vee \neg x_3 \vee \neg x_4$

Trojúhelníky pokryjeme zbylými vrcholy

NP-úplnost a důkaz Cookovy-Levinovy věty

Definice

Jazyk B je

NP-těžký pokud $A \leq_m^P B$ pro každý jazyk A v NP

NP-úplný pokud splňuje následující dvě podmínky

- 1 $B \in \text{NP}$
- 2 B je NP-těžký

Jak ukazovat NP-úplnost

Věta

Je-li jazyk B NP-úplný a $B \leq_m^P C$ pro nějaký jazyk C v NP, pak jazyk C je také NP-úplný.

Důkaz.

- Z NP-úplnosti B platí pro každý jazyk $A \in \text{NP}$, že $A \leq_m^P B \leq_m^P C$.
- Z tranzitivity $\leq_m^P$ plyne NP-těžkost C
- C je v NP dle předpokladu, je tedy NP-úplný



Pro použití této věty potřebujeme již mít nějaký NP-úplný problém.

První NP-úplný problém

SPLNITELNOST (SAT)

Instance: Formule φ v KNF.

Otázka: Je formule φ splnitelná?

Věta

SAT je NP-úplný problém.

- Jako důsledek dostáváme Cookovu-Levinovu větu

Věta (Cookova-Levinova věta)

SAT patří do P, právě když $P = NP$.

SAT patří do NP

Lemma

SAT patří do NP

Důkaz.

Polynomiální verifikátor $V(\varphi, \mathbf{a})$ pro SAT:

- Pro danou KNF φ a ohodnocení $\mathbf{a}$
- ověří, zda $\mathbf{a}$ splňuje φ



Zbývá ukázat, že SAT je NP-těžký.

NP-těžkost SAT

- Necht' $A \subseteq \Sigma^*$ je jazyk v NP



$w \in A \iff \varphi$ je splnitelná

Idea

- A je přijímán nějakým NTS M v polynomiálním čase
- Modely φ popisují přijímající výpočty M nad w

Přijímající výpočet

- Nechť $M = (Q, \Sigma, \delta, q_0, F)$ je nedeterministický TS, který
 - přijímá A (tedy $A = L(M)$)
 - pracuje v čase n^k pro nějaké $k \in \mathbb{N}$
- Vstup w je přijat M , pokud existuje posloupnost konfigurací

$$C_0^w \xrightarrow{\delta} C_1 \xrightarrow{\delta} C_2 \xrightarrow{\delta} \dots \xrightarrow{\delta} C_{n^k}$$

- C_0^w je počáteční konfigurace výpočtu $M(w)$
- Jedna z konfigurací v posloupnosti je přijímající

Technické předpoklady

- M pracuje v čase $n^k - 3$
- Je-li C přijímající konfigurace, pak $C \xrightarrow{\delta} C$
 - Výpočet fakticky skončil, umožňuje doplnit posloupnost konfigurací na potřebnou délku

Tableau

tableau T pro $M(w)$ je matice typu $n^k \times n^k$

- Řádky popisují konfigurace
- Konfigurace začínají a končí znakem $\#$
- Stav je zapsán před políčkem, které je pod hlavou M
- Konfigurace na řádku $i > 1$ následuje z konfigurace na řádku $i - 1$ pomocí přechodové funkce δ

přijímající tableau na nějakém řádku je přijímající konfigurace

buňka jedno políčko tableau

$T[i, j]$ buňka na indexech $i, j \in \{1, \dots, n^k\}$

M přijímá w , právě když existuje přijímající tableau pro $M(w)$.

Tableau

n^k	#	q_0	w_1	w_2	$\dots$	w_n	λ	$\dots$	λ	#	C_0^w		
	#										#	C_1	
	#										#	C_2	
	$\vdots$										$\vdots$		
	#	b	i	g	q_3	λ	e	r	λ	λ	λ	#	C_i
	#	b	i	g	g	q_4	e	r	λ	λ	λ	#	C_{i+1}
	$\vdots$										$\vdots$		
	#										#	C_{n^k}	
n^k													

Proměnné

- $S = Q \cup \Sigma \cup \{\#\}$
 - Množina symbolů použitých v buňkách tableau
- φ má proměnné $x_{i,j,s}$ pro $i, j = 1, \dots, n^k$ a $s \in S$

V modelu φ má platit

$$x_{i,j,s} = 1 \iff T[i, j] = s$$

Struktura φ

φ je konjunkcí čtyř podformulí

Každé buňce je přiřazen právě jeden symbol

Každá řádka následuje z předchozí podle přechodové funkce δ

$$\varphi = \varphi_{\text{cell}} \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}} \wedge \varphi_{\text{accept}}$$

První řádek obsahuje počáteční konfiguraci C_0^w

Jeden z řádků obsahuje přijímající konfiguraci

Sémantika

Každé buňce je přiřazen právě jeden symbol z S

Alespoň jeden symbol v buňce $T[i, j]$

$$\varphi_{\text{cell}} = \bigwedge_{1 \leq i, j \leq n^k} \left[\left(\bigvee_{s \in S} x_{i,j,s} \right) \wedge \left(\bigwedge_{\substack{s, t \in S \\ s \neq t}} (\neg x_{i,j,s} \vee \neg x_{i,j,t}) \right) \right]$$

Nejvýš jeden symbol v buňce $T[i, j]$

Sémantika

První řádek obsahuje počáteční konfiguraci se vstupem w

#	q_0	w_1	w_2	...	w_n	λ	λ	...	λ	#
---	-------	-------	-------	-----	-------	-----------	-----------	-----	-----------	---

 C_0^w

$$\begin{aligned}
 \varphi_{\text{start}} = & x_{1,1,\#} \wedge x_{1,2,q_0} \\
 & \wedge x_{1,3,w_1} \wedge x_{1,4,w_2} \wedge \cdots \wedge x_{1,n+2,w_n} \\
 & \wedge x_{1,n+3,\lambda} \wedge \cdots \wedge x_{1,n^k-1,\lambda} \\
 & \wedge x_{1,n^k,\#}
 \end{aligned}$$

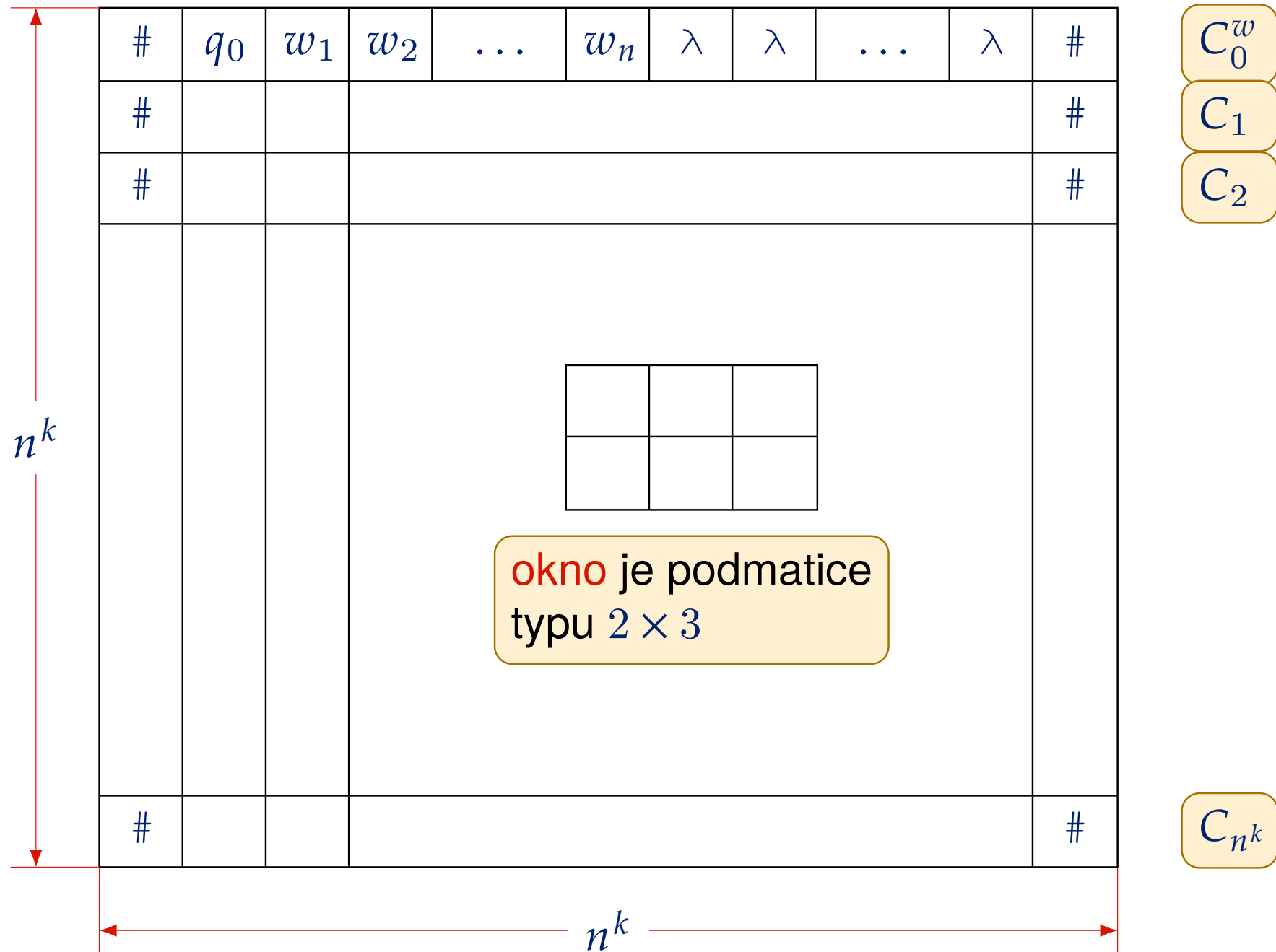
Sémantika

Jeden z řádků obsahuje přijímající konfiguraci

- Předpokládejme, že M má jediný přijímající stav q_1
- Požadujeme, aby nějaká buňka v T obsahovala q_1

$$\varphi_{\text{accept}} = \bigvee_{1 \leq i, j \leq n^k} x_{i,j,q_1}$$

Okno



Přípustné okno

okno podmatice T typu 2×3

(i, j) -**okno** má v levém horním rohu buňku $T[i, j]$

přípustné okno je součástí nějakého přechodu dle δ

Dle přechodové funkce

a $\rightarrow$ c, hlava se pohla doprava

$$\delta(q_2, a) = \{(q_4, c, L), (q_3, b, R)\}$$

$$\delta(q_4, b) = \{(q_2, a, L)\}$$

je následující přechod přípustný

#	λ	λ	b	q_2	a	c	λ	λ	λ	#
#	λ	λ	q_4	b	c	c	λ	λ	λ	#

Přípustná okna

#	λ	λ	b	q_2	a	c	λ	λ	λ	#
#	λ	λ	q_4	b	c	c	λ	λ	λ	#

Uvedený přechod ukazuje přípustnost následujících oken

#	λ	λ
#	λ	λ

Vždycky můžou nedělat nic

Přípustná okna

#	λ	λ	b	q_2	a	c	λ	λ	λ	#
#	λ	λ	q_4	b	c	c	λ	λ	λ	#

Uvedený přechod ukazuje přípustnost následujících oken

#	λ	λ
#	λ	λ

λ	λ	b
λ	λ	q_4

Avšak můžou nejdít víc

Přípustná okna

#	λ	λ	b	q_2	a	c	λ	λ	λ	#
#	λ	λ	q_4	b	c	c	λ	λ	λ	#

Uvedený přechod ukazuje přípustnost následujících oken

#	λ	λ
#	λ	λ

λ	λ	b
λ	λ	q_4

λ	b	q_2
λ	q_4	b

Přípustná okna

#	λ	λ	b	q_2	a	c	λ	λ	λ	#
#	λ	λ	q_4	b	c	c	λ	λ	λ	#

Uvedený přechod ukazuje přípustnost následujících oken

#	λ	λ
#	λ	λ

λ	λ	b
λ	λ	q_4

λ	b	q_2
λ	q_4	b

b	q_2	a
q_4	b	c

Přípustná okna

#	λ	λ	b	q_2	a	c	λ	λ	λ	#
#	λ	λ	q_4	b	c	c	λ	λ	λ	#

Uvedený přechod ukazuje přípustnost následujících oken

#	λ	λ
#	λ	λ

λ	λ	b
λ	λ	q_4

λ	b	q_2
λ	q_4	b

b	q_2	a
q_4	b	c

q_2	a	c
b	c	c

Přípustná okna

#	λ	λ	b	q_2	a	c	λ	λ	λ	#
#	λ	λ	q_4	b	c	c	λ	λ	λ	#

Uvedený přechod ukazuje přípustnost následujících oken

#	λ	λ
#	λ	λ

λ	λ	b
λ	λ	q_4

λ	b	q_2
λ	q_4	b

b	q_2	a
q_4	b	c

q_2	a	c
b	c	c

a	c	λ
c	c	λ

Přípustná okna

#	λ	λ	b	q_2	a	c	λ	λ	λ	#
#	λ	λ	q_4	b	c	c	λ	λ	λ	#

Uvedený přechod ukazuje přípustnost následujících oken

#	λ	λ
#	λ	λ

λ	λ	b
λ	λ	q_4

λ	b	q_2
λ	q_4	b

b	q_2	a
q_4	b	c

q_2	a	c
b	c	c

a	c	λ
c	c	λ

c	λ	λ
c	λ	λ

Přípustná okna

#	λ	λ	b	q_2	a	c	λ	λ	λ	#
#	λ	λ	q_4	b	c	c	λ	λ	λ	#

Uvedený přechod ukazuje přípustnost následujících oken

#	λ	λ
#	λ	λ

λ	λ	b
λ	λ	q_4

λ	b	q_2
λ	q_4	b

b	q_2	a
q_4	b	c

q_2	a	c
b	c	c

a	c	λ
c	c	λ

c	λ	λ
c	λ	λ

λ	λ	λ
λ	λ	λ

Přípustná okna

#	λ	λ	b	q_2	a	c	λ	λ	λ	#
#	λ	λ	q_4	b	c	c	λ	λ	λ	#

Uvedený přechod ukazuje přípustnost následujících oken

#	λ	λ
#	λ	λ

λ	λ	b
λ	λ	q_4

λ	b	q_2
λ	q_4	b

b	q_2	a
q_4	b	c

q_2	a	c
b	c	c

a	c	λ
c	c	λ

c	λ	λ
c	λ	λ

λ	λ	λ
λ	λ	λ

λ	λ	#
λ	λ	#

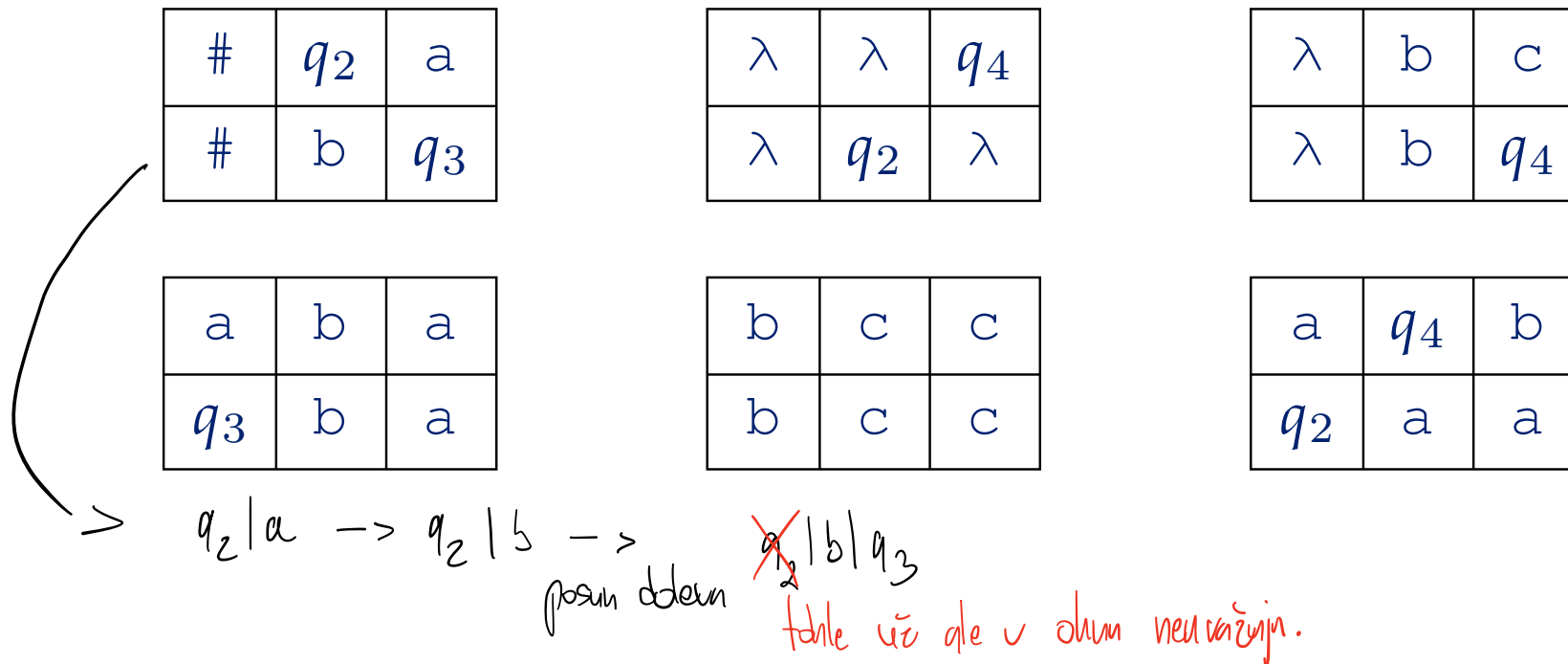
Další přípustná okna

- Uvažme následující přechodovou funkci

$$\delta(q_2, a) = \{(q_4, c, L), (q_3, b, R)\}$$

$$\delta(q_4, b) = \{(q_2, a, L)\}$$

- Podle ní jsou též následující okna přípustná



Nepřípustná okna

- Uvažme následující přechodovou funkci

$$\delta(q_2, a) = \{(q_4, c, L), (q_3, b, R)\}$$

$$\delta(q_4, b) = \{(q_2, a, L)\}$$

- Podle ní následující okna přípustná nejsou

#	q_2	a
#	q_2	a

q_3	λ	q_4
λ	q_2	λ

λ	b	c
λ	c	b

b	a	a
q_2	b	a

q_3	c	c
b	c	c

a	a	q_4
q_2	a	a

vždy můžeme mít jen jeden stav

*mění se dle
práce pod
hlavou, ne na 2
místech*

Hlavní vlastnost přípustných oken

Máme-li přípustnou každou řádku (resp. každé okno), máme přípustnou i celé tableau

Lemma

Předpokládejme, že

- *první řada tableau obsahuje počáteční konfiguraci se vstupem w a*
- *všechna okna v tableau jsou přípustná.*

Pak každá řádka tableau je konfigurací, jež následuje předchozí konfiguraci dle přechodové funkce.

Důkaz

- Indukcí dle pořadí řádku
- Řádka 1 je konfigurací z předpokladu
- Předpokládejme dvě následující řady i a $i + 1$
- Předpokládejme, že řádek i je konfigurací
- Ukážeme, že i řádek $i + 1$ je konfigurací

Přípustná okna tableau (důkaz)

Uvážíme každý symbol a konfigurace na řádku i

- 1 ■ $a = \#$
 - Přípustné okno okopíruje $\#$ z horní řady do dolní
 - Symboly $\#$ se vyskytují na okrajích každého řádku
- 2 ■ $a \in \Sigma$ v buňce, která nesousedí se symbolem stavu
 - a je uprostřed horní řady okna, jež neobsahuje symbol stavu
 - Z přípustnosti okna plyne, že a je též uprostřed spodní řady okna
 - a je v témž sloupci i v řadě $i + 1$
- 3 ■ $a = q \in Q$
 - q je uprostřed horní řady nějakého okna
 - Z přípustnosti okna plyne, že stav i okolní symboly jsou upraveny dle přechodové funkce δ

Je-li na řádce i konfigurace, pak na řádce $i + 1$ je konfigurace, která následuje i -tou konfigurací dle přechodové funkce.

2

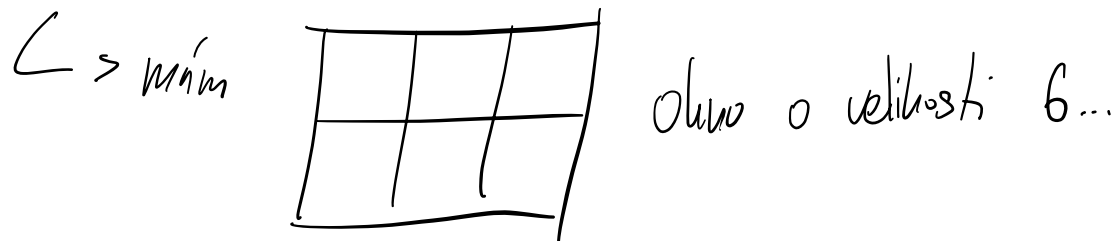
#		q		#
#	//	q	//	#

→ jenom tabulka se přepíše podle přechodové fce.

pak to zas bude odpovídat nějaké konfiguraci

Množina přípustných oken

- Úprava dle přechodové funkce je lokální
 - Změna se týká jen 5 oken
- Zbylá okna jen kopírují horní řadu do spodní
 - má tedy spodní řadu shodnou s horní
- Množinu W přípustných oken lze zkonstruovat se znalostí přechodové funkce δ
 - Pro dané okno je možné zkontrolovat, je-li přípustné
 - $|W| \leq |S|^6$ což je konstanta, je-li M pevně daný



Zakódování přípustných oken

- Následující formule reprezentuje fakt, že (i, j) -okno je přípustné

$$\text{legal}_{i,j} = \bigvee_{\substack{s_1, \dots, s_6 \\ \text{je přípustné okno}}} (x_{i,j,s_1} \wedge x_{i,j+1,s_2} \wedge x_{i,j+2,s_3} \\ \wedge x_{i+1,j,s_4} \wedge x_{i+1,j+1,s_5} \wedge x_{i+1,j+2,s_6})$$

- $\text{legal}_{i,j}$ má konstantní velikost
- $\text{legal}_{i,j}$ má ekvivalentní KNF konstantní velikosti
 - Lze zkonstruovat s použitím distributivity $\vee$ a $\wedge$

< > každý DNF můžeme převést na CNF.

Sémantika

Každý řádek následuje předchozí dle přechodové funkce

$$\varphi_{\text{move}} = \bigwedge_{1 \leq i, j \leq n^k} \text{legal}_{i,j}$$

- Použijeme-li KNF ekvivalentní formuli $\text{legal}_{i,j}$, pak φ_{move} je KNF

Velikost φ a korektnost převodu

- Počet proměnných je $(n^k)^2 \cdot |S|$ $\xrightarrow{\text{rozměr tabulky}} \xrightarrow{\text{počet symbolů}}$
- $|S|$ je konstantní, tedy počet proměnných je $O(n^{2k})$
- Konstrukci φ lze provést v polynomiálním čase

Velikost φ je polynomiální v n .

- Z konstrukce plyne, že

φ je splnitelná, právě když $w \in A$.