NP-úplnost problému SAT

Problém SAT je definován následovně:

SAT (splnitelnost booleovských formulí)

Vstup: Booleovská formule φ.

Otázka: Je φ splnitelná?

Příklad:
Formule $\varphi_1 = x_1 \wedge (\neg x_2 \vee x_3)$ je splnitelná:
např. při ohodnocení ν, kde $[x_1]_\nu = 1$, $[x_2]_\nu = 0$, $[x_3]_\nu = 1$, je $[\varphi_1]_\nu = 1$.

Formule $\varphi_2 = (x_1 \wedge \neg x_1) \vee (\neg x_2 \wedge x_3 \wedge x_2)$ není splnitelná:
pro libovolné ohodnocení ν platí $[\varphi_2]_\nu = 0$.

NP-úplnost problému SAT

Ukázat, že SAT patří do třídy NPTIME je snadné.

Nedeterministický algoritmus řešící SAT v polynomiálním čase pracuje následovně:

- Nedeterministicky zvolí ohodnocení ν, které přiřazuje booleovskou hodnotu každé proměnné vyskytující se ve formuli φ.
- Vyhodnotí φ při ohodnocení ν, tj. spočítá hodnotu $[\varphi]_\nu$.
- Pokud $[\varphi]_\nu = 1$, vrátí algoritmus odpověď ANO. Jinak vrátí odpověď NE.

NP-úplnost problému SAT

Ukázat, že problém SAT je NP-těžký je složitější.

Je třeba ukázat, že pro libovolný problém $P \in \text{NPTIME}$ existuje polynomiální redukce z problému P na problém SAT, tj. ukázat, že existuje algoritmus, který:

- dostane na svůj vstup (libovolnou) instanci problému P,
- k této instanci vyrobí booleovskou formuli φ takovou, že φ bude splnitelná právě tehdy, když pro danou instanci problému P bude odpověď Ano,
- bude mít polynomiální časovou složitost.

NP-úplnost problému SAT

Jestliže $P \in \mathrm{NPTIME}$, musí existovat **nedeterministický** Turingův stroj M a polynom $p(n)$ takový, že:

- Pro libovolnou instanci w problému P (reprezentovanou slovem v nějaké abecedě Σ) platí, že:
  - Pokud je odpověď pro w $\textsc{Ano}$, pak existuje alespoň jeden výpočet stroje M nad slovem w, při kterém stroj M vydá odpověď $\textsc{Ano}$.
  - Pokud je odpověď pro w $\textsc{Ne}$, pak všechny výpočty stroje M nad slovem w skončí s odpovědí $\textsc{Ne}$.
- Stroj M provede při libovolném výpočtu nad slovem w nejvýše $p(|w|)$ kroků.

NP-úplnost problému SAT

Ukážeme, jak pro daný nedeterministický Turingův stroj $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, polynom $p(n)$ a slovo $w \in \Sigma^*$ vyrobit booleovskou formuli φ takovou, že:

- φ bude splnitelná právě tehdy, když existuje výpočet stroje M nad slovem w, při kterém M udělá nejvýše $p(|w|)$ kroků a vydá odpověď Ano.
- Formuli φ bude možné vyrobit algoritmem v čase polynomiálním vzhledem k délce slova w.

NP-úplnost problému SAT

Připomeňme, že v definici Turingova stroje $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$:

- Q – množina stavů řídící jednotky
- Σ – vstupní abeceda $(\Sigma \subseteq \Gamma)$
- Γ – pásková abeceda
- $\delta : (Q - F) \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{-1, 0, +1\})$ – přechodová funkce
- $q_0 \in Q$ – počáteční stav řídící jednotky
- $F \subseteq Q$ – množina koncových stavů

Dále předpokládáme, že Γ obsahuje speciální prázdný symbol $\sqcup$ (blank), přičemž $\sqcup \notin \Sigma$.

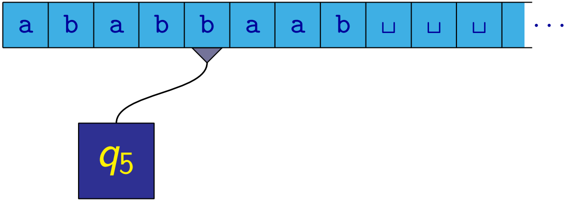
NP-úplnost problému SAT

Vzhledem k tomu, že se v následující konstrukci budeme zabývat pouze stroji, které řeší rozhodovací problémy, budeme předpokládat, že

$$F = \{q_{\text{ANO}}, q_{\text{NE}}\}$$

- Pokud stroj skončí ve stavu q_{ANO}, znamená to, že vydal odpověď Ano.
- Pokud stroj skončí ve stavu q_{NE}, znamená to, že vydal odpověď Ne.

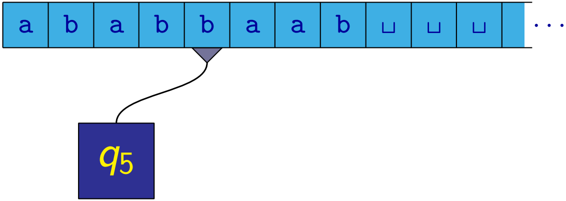
NP-úplnost problému SAT



Konfigurace Turingova stroje je dána:

- stavem řídící jednotky
- obsahem pásky
- pozicí hlavy

NP-úplnost problému SAT

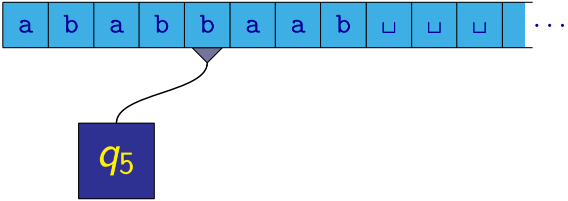


Konfiguraci můžeme zapsat jako slovo v abecedě $\Gamma \cup (Q \times \Gamma)$:

a	b	a	b	q_5 b	a	a	b

Toto slovo, vždy obsahuje právě jeden znak z $(Q \times \Gamma)$, který vyznačuje stav řídící jednotky i pozici hlavy.

NP-úplnost problému SAT

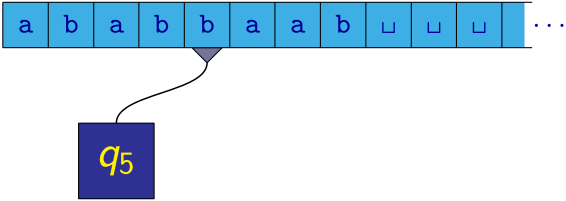


Konfiguraci můžeme zapsat jako slovo v abecedě $\Gamma \cup (Q \times \Gamma)$:

$$\boxed{a}\,\boxed{b}\,\boxed{a}\,\boxed{b}\,\boxed{\begin{matrix} q_5 \\ b \end{matrix}}\,\boxed{a}\,\boxed{a}\,\boxed{b}$$

Poznámka: Znaky z $(Q \times \Gamma)$ píšeme jako $\boxed{\begin{matrix} q \\ a \end{matrix}}$ místo $\boxed{(q, a)}$.

NP-úplnost problému SAT

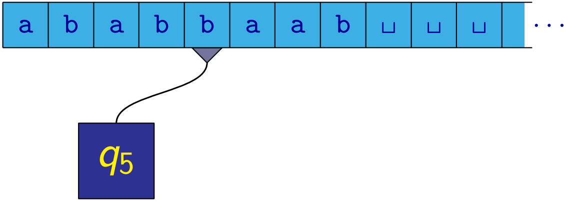


Konfiguraci můžeme zapsat jako slovo v abecedě $\Gamma \cup (Q \times \Gamma)$:

$$\boxed{a}\ \boxed{b}\ \boxed{a}\ \boxed{b}\ \boxed{\begin{matrix} q_5 \\ b \end{matrix}}\ \boxed{a}\ \boxed{a}\ \boxed{b}$$

Ostatní symboly (z Γ) reprezentují obsah pásky.

NP-úplnost problému SAT

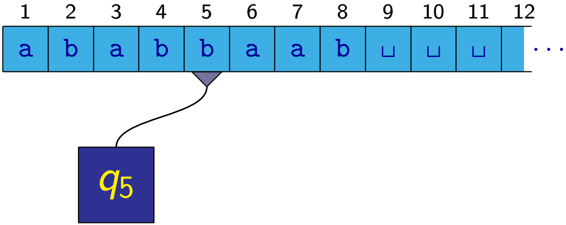


Konfiguraci můžeme zapsat jako slovo v abecedě $\Gamma \cup (Q \times \Gamma)$:

$$\boxed{a}\,\boxed{b}\,\boxed{a}\,\boxed{b}\,\boxed{\begin{matrix} q_5 \\ b \end{matrix}}\,\boxed{a}\,\boxed{a}\,\boxed{b}$$

Políčka pásky, která nejsou ve slově vyznačena, obsahují symbol $\boxed{\sqcup}$.

NP-úplnost problému SAT



Budeme předpokládat, že Turingův stroj používá jednostranně omezenou pásku.

Políčka pásky si můžeme očíslovat čísly $1, 2, 3, \ldots$.

NP-úplnost problému SAT

Výpočet Turingova stroje je posloupnost konfigurací, kde:

- První konfigurace je **počáteční konfigurace**

q_0 w_1	w_2	w_3	w_4	w_5	w_6	$\cdots\cdots$	w_{n-1}	w_n

  kde $w_1 w_2 \ldots w_n$ jsou jednotlivé symboly slova w, které je vstupem Turingova stroje M.

- Každá další konfigurace v posloupnosti je konfigurace, do které se může stroj dostat z předchozí konfigurace provedením jednoho kroku podle přechodové funkce δ.

- Pokud je výpočet konečný, v poslední konfiguraci musí být řídící jednotka v některém koncovém stavu z množiny F.

NP-úplnost problému SAT

Předpokládejme nyní, že časová složitost stroje M je shora omezena nějakou funkcí $p(n)$.

(Bez ztráty na obecnosti budeme předpokládat, že pro všechna n je $p(n) \geq n$.)

Pokud stroj M dostane jako vstup slovo w délky n, provede během výpočtu nanejvýš $p(n)$ kroků.

Protože stroj začíná s hlavou na políčku číslo 1, může se během toho výpočtu dostat hlava nanejvýš na políčko číslo $p(n) + 1$ (v každém kroku se posune nanejvýš o jedno políčko).

Pokud je tedy časová složitost stroje M shora omezena funkcí $p(n)$, můžeme všechny konfigurace ve výpočtu nad vstupem velikosti n zapsat jako slova délky $p(n) + 1$

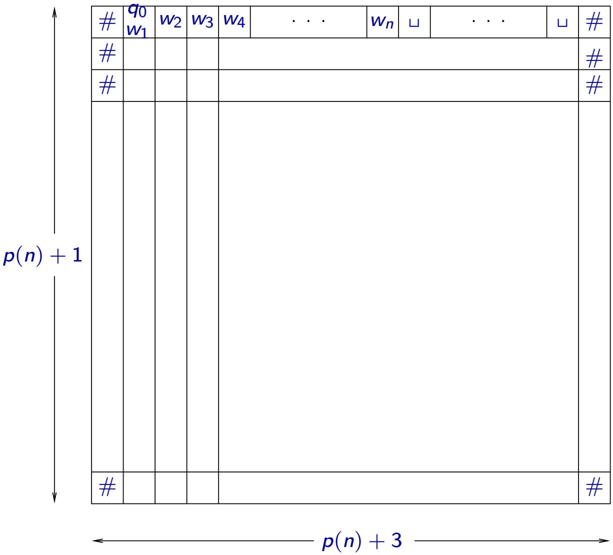
Na políčka s čísly většími než $p(n) + 1$ se stroj během výpočtu hlavou nedostane a tato políčka budou obsahovat symbol $\llcorner\!\lrcorner$ (připomeňme, že předpokládáme $p(n) \geq n$).

NP-úplnost problému SAT

Slova popisující jednotlivé konfigurace ve výpočtu stroje M nad slovem $w = w_1 w_2 \cdots w_n$ tedy můžeme zapsat pod sebe do tabulky, kde:

- Řádky odpovídají jednotlivým konfiguracím (zapsaným jako slova v abecedě $\Gamma \cup (Q \times \Gamma)$).
- Sloupce odpovídají políčkům pásky s čísly $1, 2, \ldots, p(n) + 1$.
- Z technických důvodů přidáme ještě zleva a zprava sloupce obsahující jen speciální oddělovací znaky $\#$ (přičemž $\# \notin Q \cup \Gamma$).

NP-úplnost problému SAT



NP-úplnost problému SAT

- Jednotlivá políčka tabulky tedy budou obsahovat symboly z abecedy
$$\Delta = \Gamma \cup (Q \times \Gamma) \cup \{\#\}$$
- Řádky tabulky budou mít čísla 0 až $p(n) + 1$.
(Řádek 0 bude obsahovat počáteční konfiguraci).
- Sloupce budou mít čísla 0 až $p(n) + 2$.
(Sloupce 0 a $p(n) + 2$ budou obsahovat znaky $\#$.)

Poznámka: Výpočet může být kratší než $p(n)$ kroků a v takovém případě by nebyla vyplněna celá tabulka.

Abychom i v tomto případě vyplnili celou tabulku, můžeme udělat to, že poslední konfiguraci, ve které se výpočet zastavil, zopakujeme ve všech zbylých řádcích tabulky.

NP-úplnost problému SAT

Pokud tedy máme dán (nedeterministický) Turingův stroj M s časovou složitostí omezenou shora polynomem $p(n)$ řešící problém P a instanci tohoto problému zapsanou jako slovo w, je pro tuto instanci odpověď Ano právě tehdy, když existuje nějaký přijímající výpočet stroje M nad slovem w.

Takový přijímající výpočet můžeme výše popsaným způsobem zapsat do tabulky o $p(n) + 1$ řádcích a $p(n) + 3$ sloupcích.

Poznámka: Všimněte si, že velikost tabulky je polynomiální vzhledem k n.

NP-úplnost problému SAT

Pro daný stroj M, polynom $p(n)$ a slovo w vytvoříme formuli φ takovou, že:

- Různá ohodnocení ν booleovských proměnných ve formuli φ budou reprezentovat všechny možné (i nesmyslné) obsahy dané tabulky.
- $[\varphi]_\nu = 1$ bude platit právě pro ta ohodnocení ν, která reprezentují takový obsah tabulky, který je zápisem přijímajícího výpočtu stroje M nad vstupem w.

Formule φ tedy bude splnitelná právě tehdy, pokud bude existovat přijímající výpočet stroje M nad vstupem w.

NP-úplnost problému SAT

Formule φ bude složena pomocí booleovských spojek z atomických tvrzení typu:

„*Políčko* (i, j) *obsahuje symbol* a*.*“

kde i, j, a budou konkrétní konstanty, například:

„*Políčko* $(9, 4)$ *obsahuje symbol* $\boxed{\begin{matrix} q_5 \\ b \end{matrix}}$*.*“

Poznámka: Když mluvíme o políčku (i, j), máme tím na mysli políčko v i-tém řádku a j-tém sloupci tabulky.

NP-úplnost problému SAT

Formule φ tedy bude obsahovat booleovské proměnné $x^a_{i,j}$, kde:

- $0 \leq i \leq p(n) + 1$
- $0 \leq j \leq p(n) + 2$
- $a \in \Delta$ (kde $\Delta = \Gamma \cup (Q \times \Gamma) \cup \{\#\}$)

s tím, že $[x^a_{i,j}]_\nu = 1$ znamená, že ν reprezentuje obsah tabulky, při kterém políčko (i, j) obsahuje symbol a,

a $[x^a_{i,j}]_\nu = 0$ znamená, že ν reprezentuje obsah tabulky, při kterém políčko (i, j) neobsahuje symbol a.

Příklad: Proměnná $x_{9,4}^{q_5,\mathrm{b}}$ reprezentuje tvrzení:

„*Políčko* $(9,4)$ *obsahuje symbol* $\boxed{\begin{matrix} q_5 \\ \mathrm{b} \end{matrix}}$.“

Poznámka: U hodnot z $(Q \times \Gamma)$ budeme v indexech raději používat zápis q, a místo $\boxed{\begin{matrix} q \\ a \end{matrix}}$.

Pokud tedy $[x_{9,4}^{q_5,\mathrm{b}}]_\nu = 1$, znamená to, že při obsahu tabulky reprezentovaném ν políčko $(9,4)$ obsahuje symbol $\boxed{\begin{matrix} q_5 \\ \mathrm{b} \end{matrix}}$,

a pokud $[x_{9,4}^{q_5,\mathrm{b}}]_\nu = 0$, tak při ν políčko $(9,4)$ neobsahuje $\boxed{\begin{matrix} q_5 \\ \mathrm{b} \end{matrix}}$.

NP-úplnost problému SAT

Celá formule φ, která jako celek bude říkat:

> *Tabulka obsahuje přijímající výpočet stroje M nad slovem w,*

bude složena z mnoha podformulí, kde každá z těchto podformulí bude formulovat nějakou jednoduchou podmínku, která musí být splněna, aby obsah tabulky byl přijímajícím výpočtem stroje M nad slovem w.

Tyto podformule budou spojeny pomocí konjunkce.

Pokud tedy bude při daném ohodnocení ν některá z těchto podmínek porušena, bude celá formule φ při ν nabývat hodnoty FALSE, tj. $[\varphi]_\nu = 0$.

V následujícím výkladu si popíšeme tyto jednotlivé podformule.

Pokud v dalším výkladu o nějaké formuli ψ řekneme, že

„*ψ přidáme do φ*,“

máme tím na mysli, že ψ spojíme pomocí konjunkce ($\wedge$) s dosud vytvořenou částí formule φ.

NP-úplnost problému SAT

Aby tabulka obsahovala přijímající výpočet stroje M nad vstupem w, musí být splněny následující podmínky:

1. Každé políčko tabulky obsahuje právě jeden symbol z Δ.
2. Řádek číslo 0 obsahuje počáteční konfiguraci se slovem w.
3. Každý řádek tabulky (kromě řádku 0) obsahuje buď:
   - konfiguraci, která je jedním krokem (podle přechodové funkce δ) dosažitelná z konfigurace zapsané v předchozím řádku, nebo
   - koncovou konfiguraci shodnou s konfigurací v předchozím řádku.
4. Poslední řádek tabulky obsahuje konfiguraci se stavem q_{ANO}.

Je zřejmé, že pokud tabulka obsahuje přijímající výpočet, tak jsou tyto čtyři podmínky splněny.

Na druhou stranu je rovněž zřejmé, že pokud budou tyto čtyři podmínky splněny, tak tabulka opravdu obsahuje přijímající výpočet stroje M nad slovem w.

Podívejme se nejprve na první podmínku:

Každé políčko tabulky obsahuje právě jeden symbol z Δ.

Tu zajistíme tak, že pro každé políčko (i,j) přidáme do φ podformuli, která bude říkat:

Políčko (i,j) *obsahuje právě jeden symbol z* Δ,

což můžeme formulovat též takto:

Právě jedna z proměnných $x_{i,j}^{a_1}, x_{i,j}^{a_2}, \dots, x_{i,j}^{a_k}$ *má hodnotu* TRUE,

kde $\{a_1, a_2, \dots, a_k\}$ je množina všech symbolů z Δ.

NP-úplnost problému SAT

Vyjádřit tvrzení, že právě jedna z proměnných $x_1, x_2, \ldots, x_k$ má hodnotu TRUE (kde $x_1, x_2, \ldots, x_k$ jsou nějaké libovolné booleovské proměnné) není složité.

Ukážeme si to na příkladě, kde pro jednoduchost budeme mít jen čtyři booleovské proměnné A, B, C, D:

$$
\begin{array}{l}
(\ \ A \wedge \neg B \wedge \neg C \wedge \neg D) \ \vee \\
(\neg A \wedge \ \ B \wedge \neg C \wedge \neg D) \ \vee \\
(\neg A \wedge \neg B \wedge \ \ C \wedge \neg D) \ \vee \\
(\neg A \wedge \neg B \wedge \neg C \wedge \ \ D) \ \vee
\end{array}
$$

Není těžké ověřit, že tato formule nabývá hodnoty TRUE právě při těch ohodnoceních, při kterých má právě jedna z proměnných A, B, C, D hodnotu TRUE.

NP-úplnost problému SAT

Obecně pro množinu proměnných $X = \{x_1, x_2, \ldots, x_k\}$ můžeme tuto podmínku zapsat následovně:

$$\bigvee_{x_i \in X} \left(x_i \wedge \bigwedge_{x_j \in X - \{x_i\}} \neg x_j \right)$$

Všimněme si, že pro k proměnných má tato formule velikost $O(k^2)$.

V našem případě je $k = |\Delta|$, takže velikost podformule, kterou přidáváme pro každé políčko, je $O(|\Delta|^2)$ a je to tedy konstanta nezávislá na velikosti vstupu w.

Poznámka: Existuje způsob, jak zapsat výše uvedenou podmínku tak, aby velikost vytvořené formule byla $O(k \log k)$, ale my to zde nepotřebujeme.

Další podmínkou, která musí být splněna, je:

Řádek 0 *obsahuje počáteční konfiguraci se slovem* w.

Pokud tedy $w_1 w_2 \dots w_n$ jsou jednotlivé symboly slova w, přičemž $n \geq 1$, musí platit následující:

- Políčko $(0, 1)$ obsahuje symbol (q_0, w_1), kde q_0 je počáteční stav.
- Políčka $(0, 2)$, $(0, 3)$, ..., $(0, n)$ obsahují symboly $w_2, w_3, \dots, w_n$.
- Políčka $(0, n + 1)$, $(0, n + 2)$, ..., $(0, p(n + 1))$ obsahují symboly $\sqcup$.
- Políčka $(0, 0)$ a $(0, p(n) + 2)$ obsahují symboly $\#$.

NP-úplnost problému SAT

Tuto podmínku tedy můžeme zapsat následující formulí, kterou přidáme do φ:

$$x_{0,1}^{q_0,w_1} \;\wedge\; \left(\bigwedge_{i=2}^{n} x_{0,i}^{w_i}\right) \;\wedge\; \left(\bigwedge_{i=n+1}^{p(n)+1} x_{0,i}^{\sqcup}\right) \;\wedge\; x_{0,0}^{\#} \;\wedge\; x_{0,p(n)+2}^{\#}$$

Velikost této formule je $O(p(n))$.

Poznámka: Případ, kdy $n = 0$, by se lišil jen v tom, že místo $x_{0,1}^{q_0,w_1}$ by formule obsahovala $x_{0,1}^{q_0,\sqcup}$.

NP-úplnost problému SAT

Asi nejsložitější je zajištění třetí podmínky:

> *Každý řádek (kromě řádku 0) obsahuje konfiguraci, která vznikne z předchozí provedením jednoho kroku (a nebo je kopií předchozí koncové konfigurace).*

Vezměme si nějaké dvě po sobě jdoucí konfigurace.

Tyto dvě konfigurace se vždy liší nanejvýš na dvou pozicích:

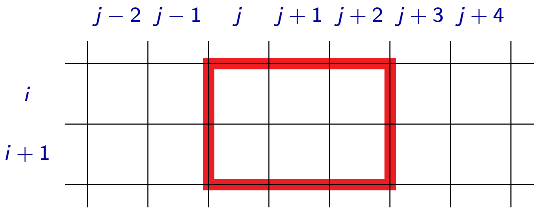
- na pozici, kde se v první z nich nachází hlava,
- a na jedné z pozic, které s ní bezprostředně sousedí.

Obsahy dvojic řádků i a $i+1$ v tabulce jsou tedy velice úzce svázány.

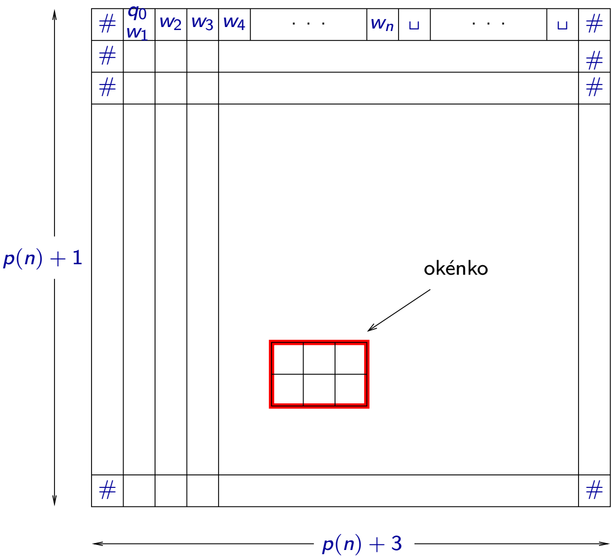
NP-úplnost problému SAT

Pokud obsah řádku $i+1$ neodpovídá konfiguraci dosažitelné jedním krokem z konfigurace na řádku i, pak se o tom můžeme přesvědčit tak, že najdeme konkrétní pozici, kde konfigurace do sebe „nepasují“.

Můžete si rozmyslet, že v takovém případě můžeme vždycky najít v dané dvojici řádků „okénko“ velikosti 2×3 takové, že o tom, že řádky **neobsahují** dvě po sobě jdoucí konfigurace se můžeme přesvědčit jen prozkoumáním obsahu tohoto okénka (tj. aniž bychom se museli dívat na obsah ostatních políček).



NP-úplnost problému SAT



NP-úplnost problému SAT

Obsahům okének, které se mohou vyskytnout ve dvou po sobě jdoucích konfiguracích budeme říkat **korektní** a těm, které se nemohou vyskytnout ve dvou po sobě jdoucích konfiguracích (a které tedy dosvědčují, že dané dva řádky neobsahují dvě po sobě jdoucí konfigurace) budeme říkat **nekorektní**.

Přesné konkrétní podmínky, které musí korektní obsahy okének splňovat, zde nebudeme uvádět.

Místo toho si ukážeme konkrétní příklady korektních a nekorektních okének.

Zkuste si však (po prohlédnutí příkladů) tyto podmínky sami zformulovat.

NP-úplnost problému SAT

Příklady **nekorektních** okének, přičemž předpokládáme, že $\delta(q_5, \mathtt{a}) = \{(q_8, \mathtt{b}, -1), (q_3, \mathtt{a}, +1)\}$:

a	b	a
b	#	a

a	q_5 a	a
b	a	a

a	q_5 a	b
q_4 ⊔	b	q_7 a

q_5 a	b	a
b	q_6 b	a

q_{ANO} a	a	b
a	q_8 a	b

a	b	b
a	b	a

Příklady **korektních** okének, přičemž předpokládáme, že $\delta(q_5, \mathtt{a}) = \{(q_8, \mathtt{b}, -1), (q_3, \mathtt{a}, +1)\}$:

a	b	a
a	b	a

a	q_5 a	b
a	a	q_3 b

b	a	⊔
q_3 b	a	⊔

q_5 a	b	a
b	b	a

a	a	q_{ANO} b
a	a	q_{ANO} b

⊔	⊔	#
⊔	⊔	#

NP-úplnost problému SAT

Množinu všech šestic znaků, které tvoří korektní obsah okének (pro daný konkrétní stroj M) si označíme $Corr$.

Tj. $(a, b, c, d, e, f) \in Corr$ právě když

a	b	c
d	e	f

je korektní obsah okénka.

Celkový počet všech možných obsahů okének je $|\Delta|^6$, což je konstanta nezávislá na velikosti vstupu w, takže i počet prvků množiny $Corr$ je konstanta nezávislá na velikosti vstupu.

NP-úplnost problému SAT

Pro každé okénko v tabulce přidáme do φ podformuli, která tvrdí, že obsah daného okénka je korektní (neboli jinak řečeno, že obsahuje některou z korektních šestic).

Tj. pro každé i takové, že $0 \le i < p(n)$, a každé j takové, že $0 \le j \le p(n)$, přidáme do φ formuli

$$\bigvee_{\substack{(a,b,c,d,e,f)\\ \in Corr}} \left(x^{a}_{i,j} \wedge x^{b}_{i,j+1} \wedge x^{c}_{i,j+2} \wedge x^{d}_{i+1,j} \wedge x^{e}_{i+1,j+1} \wedge x^{f}_{i+1,j+2} \right)$$

Každá z těchto podformulí má velikost maximálně $O(|\Delta|^6)$, tj. omezenou nějakou konstantou nezávislou na velikosti vstupu w. Celkový počet těchto podformulí je $O(p(n)^2)$

Nyní zbývá zaručit poslední podmínku:

Poslední řádek obsahuje koncovou konfiguraci, kde stav řídící jednotky je q_{ANO}.

To je opět jednoduché – stačí přidat do φ podformuli, která tvrdí, že na některé pozici v posledním řádku (tj. řádku $p(n)$) se nachází dvojice (q_{ANO}, a), kde a je nějaký symbol z Γ.

Tato podformule vypadá takto:

$$\bigvee_{j=1}^{p(n)+1} \bigvee_{a\in\Gamma} x^{a}_{p(n),j}$$

Velikost této formule je $O(p(n))$.

NP-úplnost problému SAT

Z předchozího výkladu vidíme, že velikost formule φ vytvořené k danému vstupu w velikosti n je $O(p(n)^2)$.

Jestliže je $p(n)$ polynom, tak i $p(n)^2$ je polynom, takže velikost φ je polynomiální vzhledem k n.

Vzhledem k jednoduché a pravidelné struktuře formule φ je zřejmé, že časová složitost algoritmu, který ke slovu w formuli φ vyrobí, je v podstatě úměrná velikosti formule φ, tedy také $O(p(n)^2)$.

NP-úplnost problému SAT

Ukázali jsme tedy, že konstrukce je polynomiální, a nyní stručně shrneme, proč je korektní:

- Předpokládejme, že pro w je v problému P odpověď ANO, což znamená, že existuje výpočet nedeterministického stroje M (který řeší problém P) nad slovem w vedoucí k odpovědi ANO.

  Tento výpočet můžeme zapsat do odpovídající tabulky a proměnným ve formuli φ můžeme přiřadit booleovské hodnoty podle obsahu této tabulky.

  Je zřejmé, že při tomto ohodnocení bude mít φ hodnotu TRUE, protože všechny podmínky testované ve formuli φ budou splněny.

NP-úplnost problému SAT

- Předpokládejme nyní, že formule φ je splnitelná, tj. pro nějaké ohodnocení ν platí $[\varphi]_\nu = 1$.

  Podle ohodnocení ν nyní můžeme vyplnit tabulku.

  Z toho, že při ohodnocení ν musí být splněny všechny podmínky popsané ve formuli φ, vyplývá, že takto vyplněná tabulka obsahuje popis výpočtu stroje M nad slovem w, při kterém stroj vydá odpověď $\textsc{Ano}$, a že tedy takovýto výpočet existuje.

Vidíme tedy, že formule φ je splnitelná právě tehdy, když existuje výpočet stroje M vedoucí k přijetí slova w, tj. právě tehdy, když odpověď pro w je $\textsc{Ano}$.