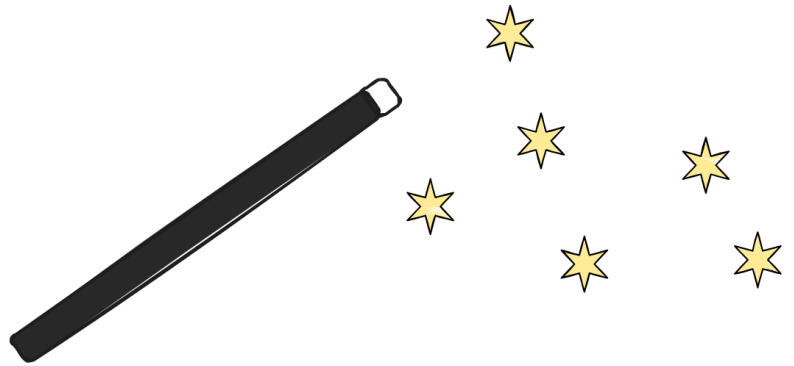
PH, preroki, vezja

Uroš Čibej

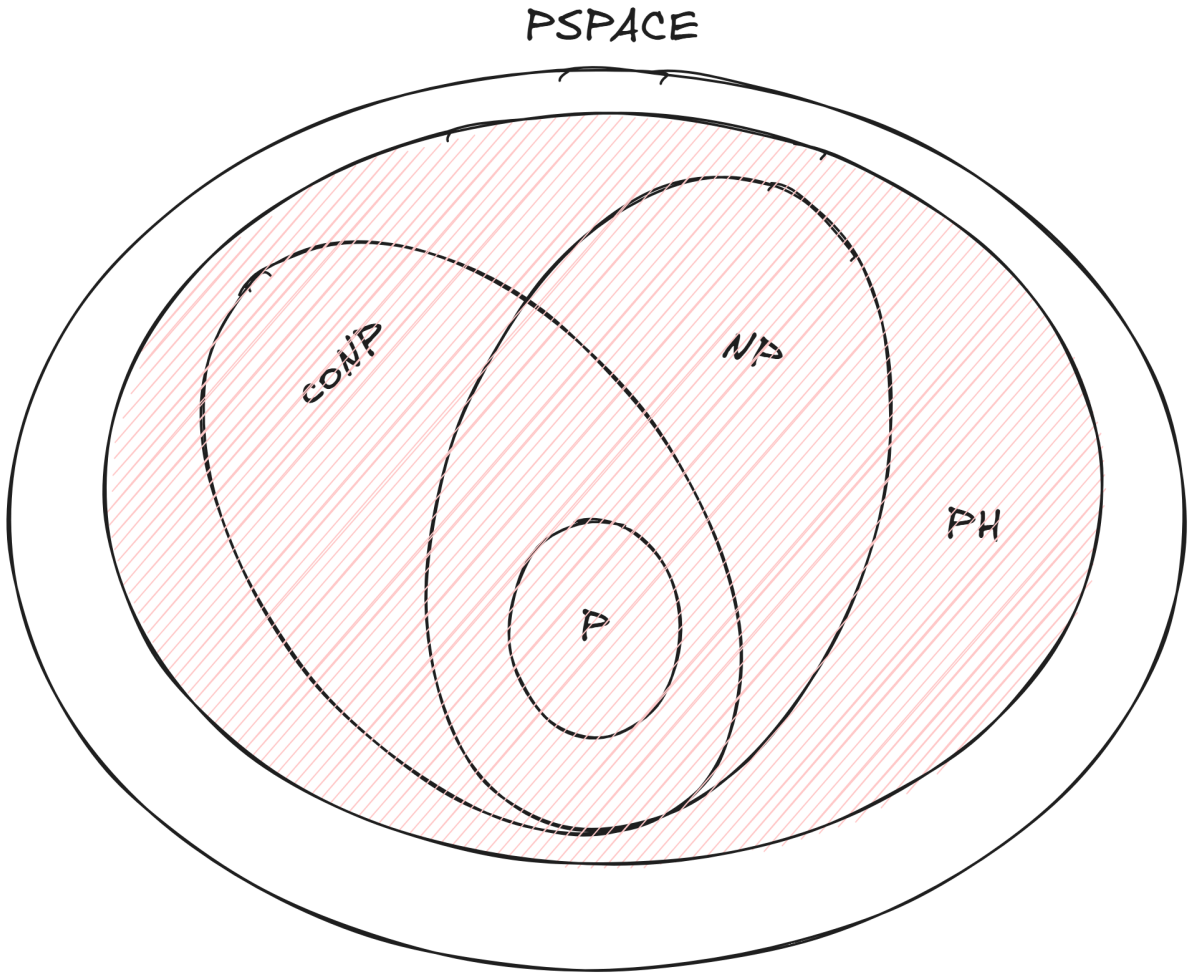


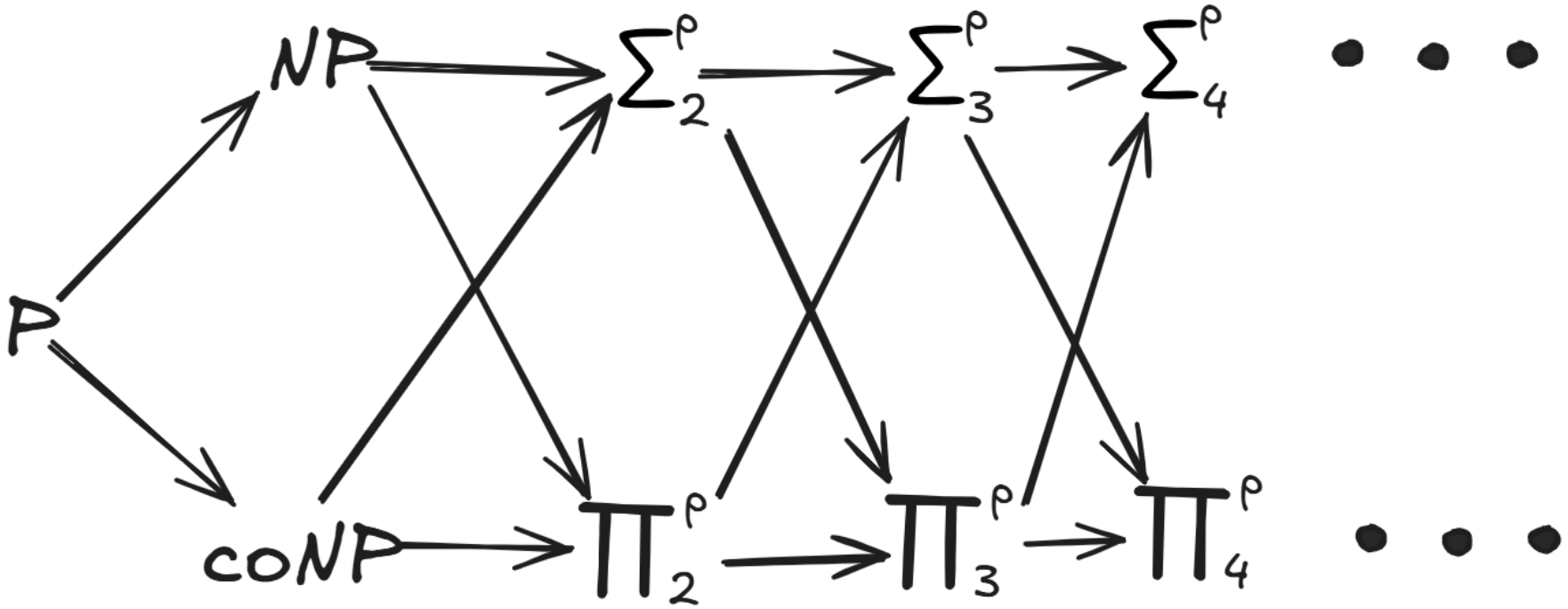
Pregled snovi

1. Še nekaj rezultatov iz PH
2. Alternirajoči Turingovi stroji
3. Turingovi stroji s preroki
4. Vezja
5. $P/poly$

Ponovimo

- Σ_i^p, Π_i^p
- PH




NP
P
coNP
Σ_2^P
Σ_3^P
Σ_4^P
Π_2^P
Π_3^P
Π_4^P

Kolaps polinomske hierarhije

Izrek:

$$P = NP \implies P = PH$$

Intuicija dokaza:

Če $P = NP$, vemo:

- $P = NP = \exists P$
- $P = coNP = \forall P$

$$\Sigma_2^p = \exists \underbrace{\forall P}_{P} = \exists P = P$$

$$\Pi_2^p = \forall \underbrace{\exists P}_{P} = \forall P = P$$

Prikaz na primeru

$$SCIRC = \{\langle C \rangle \mid C \text{ je najmanjše vezje, ki računa } f_c\}$$

Opis rešitve:

$$\forall C' : |C'| < |C|, \exists x : C'(x) \neq C(x)$$

$$DIFFC = \{\langle C_1, C_2\rangle \mid \text{ vezji računata različni funkciji } \}$$

- $DIFFC \in NP$
- $P = NP \implies$ imamo polinomski algoritem za $DIFFC$ (recimo mu A)
- $SCIRC$ potem lahko opišemo ($SCIRC \in coNP, SCIRC \in P$):

$$\forall C' : A(C, C')$$

Kolaps do nivoja i

Izrek:

$$\Sigma_i^p = \Pi_i^p \implies PH = \Sigma_i^p$$

Intuicija dokaza:

Npr., če $NP = coNP$, vemo:

- $\exists P = \forall P$

$$\Sigma_2^p = \exists \underbrace{\forall P}_{\exists P} = \exists\exists P = \exists P = NP$$

$$\Pi_2^p = \forall \underbrace{\exists P}_{\forall P} = \forall\forall P = \forall P = \exists P = NP$$

Σ_k^p-polnost

Problem A je Σ_k^p-poln:

1. $A \in \Sigma_k^p$
2. $\forall B \in \Sigma_k^p : B \leq_p A$

Problem $\Sigma_k^p - SAT$

vhod: množica n spremenljivk $x_1 \dots x_n$ in particija te množice na k particij $y_1, \dots, y_k$, ter kvantificirana formula:

$$\exists y_1 \forall y_2 \exists y_3 \dots : \phi(x_1, \dots, x_n)$$

Primer problema $k = 2$

x_1, x_2, x_3

$y_1 = \{x_1, x_2\}$

$y_2 = \{x_3\}$

$$\exists x_1, x_2 \quad \forall x_3 : \quad ((x_1 \vee x_3) \wedge (x_2 \vee \neg x_3))$$

$\Sigma_k^p - SAT$ je Σ_k^p-poln problem
$\Pi_k^p - SAT$ je Π_k^p-poln problem

Dokaz: Zopet ponovimo Cook-Levinov dokaz.

PH-polnost?

Trditev: Če obstaja A, ki je PH-poln, potem polinomska hierarhija kolapsira ($\exists i : PH = \Sigma_i^p$)

dokaz: $A \in PH$, $A \in \Sigma_i^p$

Obstajajo prevedbe iz vseh problemov v PH:

$$\forall B \in PH : B \leq_p A \implies B \in \Sigma_i^p \implies PH \subseteq \Sigma_i^p$$

$PSPACE$ in PH

Trditev: Če $PH = PSPACE$, potem polinomska hierarhija kolapsira.

dokaz $PSPACE$ ima poln problem, če $PH = PSPACE$ ima tudi PH poln problem, po prejšnji trditvi potem PH kolapsira.

Alternativni orisi PH

- Alternirajoči Turingovi stroji
- Turingovi stroji s preroki

Alternirajoči Turingovi stroji

- posplošitev nedeterminizma
- NP
  - kriterij za sprejetje $\exists$ pot $q_0 \rightarrow q_F$
- $coNP$
  - kriterij za sprejetje $\forall$ pot, kjer se stroj ustavi $q_0 \rightarrow q_F$
- To posplošimo z oznako vseh vozlišč $\exists, \forall$

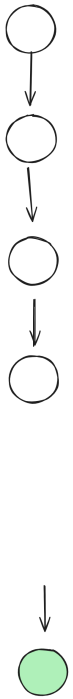
Definicija ATS

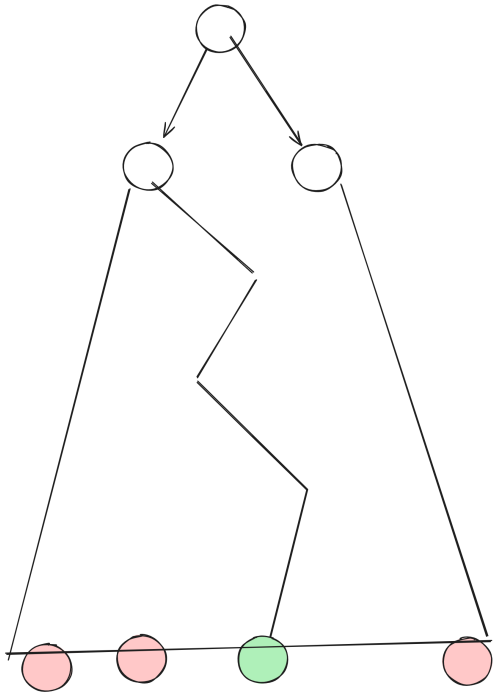
$$M = (Q, \Sigma, \Gamma, \delta, q_0, g)$$

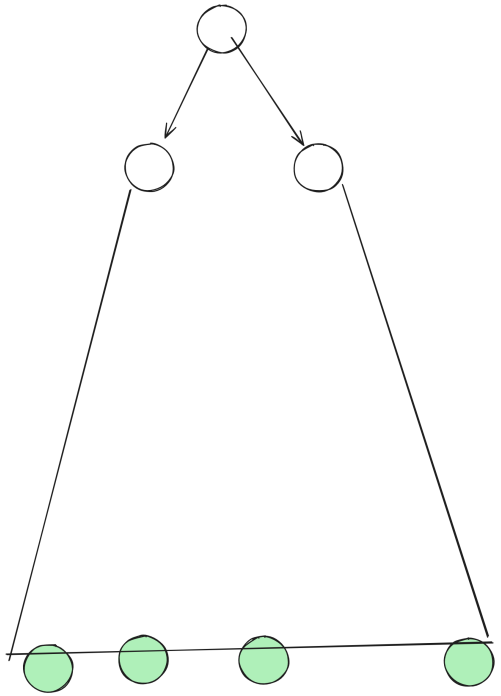
$g : Q \to \{\forall, \exists, accept, reject\}$

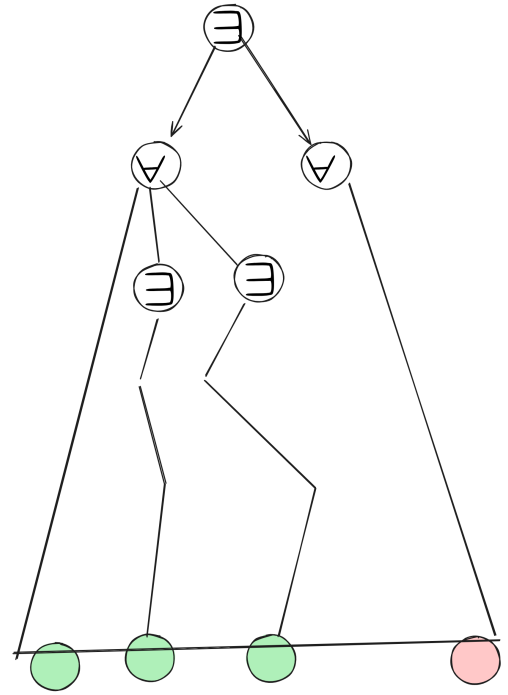
rekurzivna definicija sprejemanja konfiguracije

1. (α, q, a, β), $g(q) = accept : M$ besedo sprejme
2. (α, q, a, β), $g(q) = reject : M$ besedo zavrne
3. (α, q, a, β), $g(q) = \exists : M$ **vsaj ena** sosednja konfiguracija sprejme
4. (α, q, a, β), $g(q) = \forall : M$ **vse** sosednje konfiguracije sprejmejo









Alternirajoči čas

$$ATIME(f(n)) = \{L \mid L \text{ ima alternirajoči TS, ki porabi } f(n) \text{ časa}\}$$

Polinomski alternirajoči problemi so:

$$AP = \bigcup_i ATIME(n^i)$$

Σ_i^p alternirajoči TS

Če omejimo število uporabljenih labeliranih vozlišč $\forall, \exists$ na i, dobimo Σ_i^p alternirajoče stroje.

$$\Sigma_i^p = \bigcup_k \Sigma_i^p TIME(n^k)$$

Preroki

Motivacija

- recimo, da imamo dostop do **zelo učinkovitega** preroka za SAT (ORACLE_SAT)
- Očitno znamo rešiti vse probleme iz NP v polinomskem času
- Enako velja za probleme iz $coNP$
- A znamo rešiti kar vse probleme v PH?

Koda za HC

```
def HC(G):
    phi = prevedbaHC(G)
    return ORACLE_SAT(phi)
```

Koda za $\overline{HC}$

```
def coHC(G):
    phi = prevedbaHC(G)
    return not ORACLE_SAT(phi)
```

Koda za CHROM4

Problem: Ali je kromatsko število grafa 4?

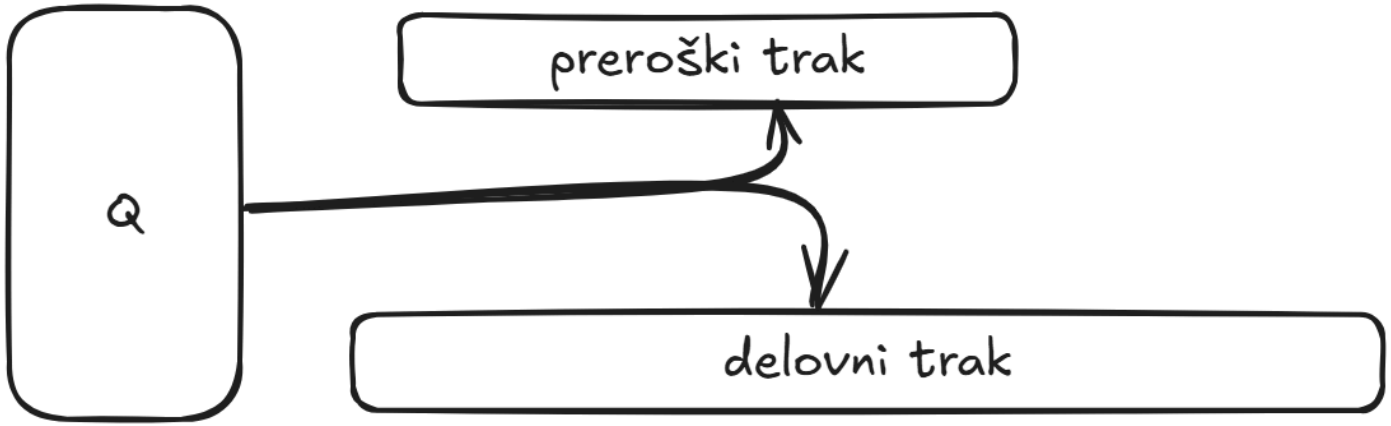
```
def CHROM4(G):
    phi1 = prevedba4COL(G)
    phi2 = prevedba3COL(G)
    return ORACLE_SAT(phi1) and not ORACLE_SAT(phi2)
```

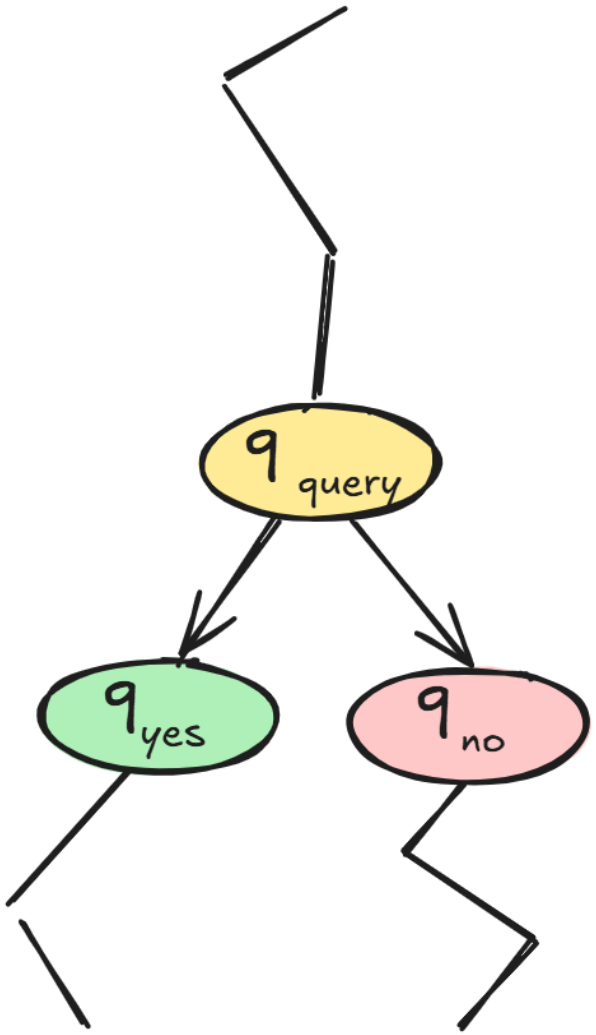
Koda za $SCIRC$?

Prerok $\neq$ algoritem

Prerok ne povroči sesutja polinomske hierarhije.

Turingovi stroji s preroki





Definicija TS z B-prerokom

$$M^B = (Q, \Sigma, \Gamma, \delta, q_{query}, q_{yes}, q_{no})$$

Standarden Turingov stroj, s tremi posebnimi stanji

Funkcija prehodov

$$\delta : (Q \setminus \{q_{query}\}) \times \Gamma \to Q \times \Gamma^2 \times \{L, R, S\}^2$$

Ko stroj skoči v stanje q_{query}, prerok za problem B poskrbi, da je naslednje stanje q_{yes} ali q_{no}

Razred P^{SAT}

$$P^{SAT} = \{L \mid L \text{ rešljiv v poli. času na } SAT - \text{preroškem stroju } M^{SAT}\}$$

$NP \subseteq P^{SAT}$

$coNP \subseteq P^{SAT}$

$CHROM4 \in P^{SAT}$

Posplošitev

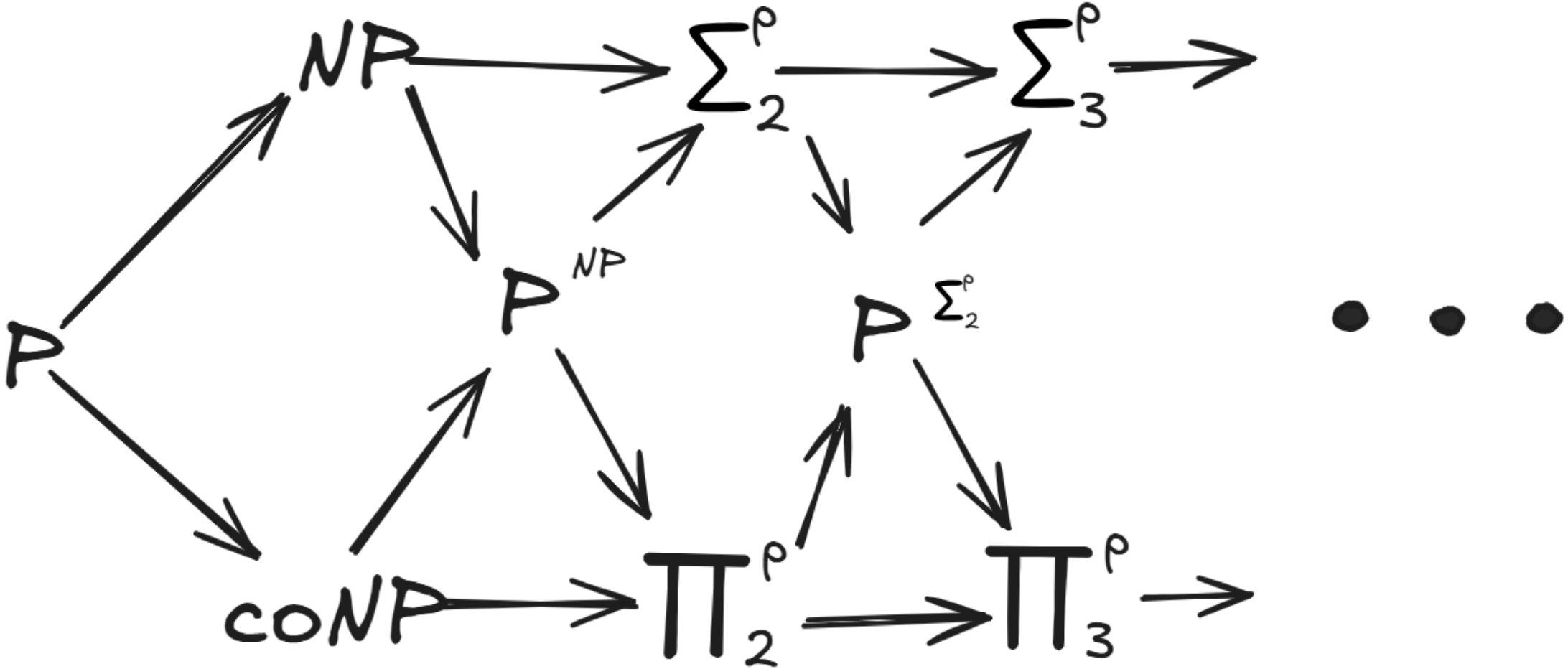
$$P^B = \{L \mid L \text{ rešljiv v poli. času na } B - \text{preroškem stroju } M^B\}$$

posebna notacija

$P^B \subseteq P^{SAT} : \forall B \in NP$ (zato uporabljamo notacijo P^{NP})

Polinomska hierarhija in preroki

$$P^{NP} \subseteq \Sigma_2^p$$
$$P^{NP} \subseteq \Pi_2^p$$

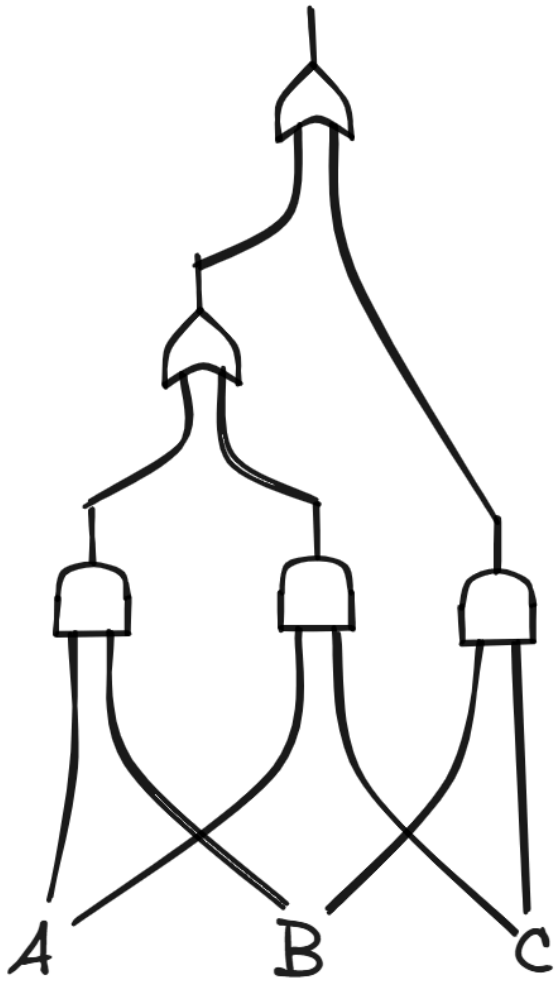

NP
Σ_2^P
Σ_3^P
P
P^{NP}
$P^{\Sigma_2^P}$
coNP
Π_2^P
Π_3^P

Vezja

Motivacija

- standarden model je Turingov stroj
- moderno računanje pa je implementirano z vezji
- kako formalno modelirati računanje z vezji
- kakšni rezultati sledijo iz teh modelov

Primer



Primer (AND)

$$AND = \{\langle x_1, x_2, \ldots, x_n \rangle \mid x_1 \wedge x_2 \wedge \ldots \wedge x_n\}$$

Primer (pariteta)

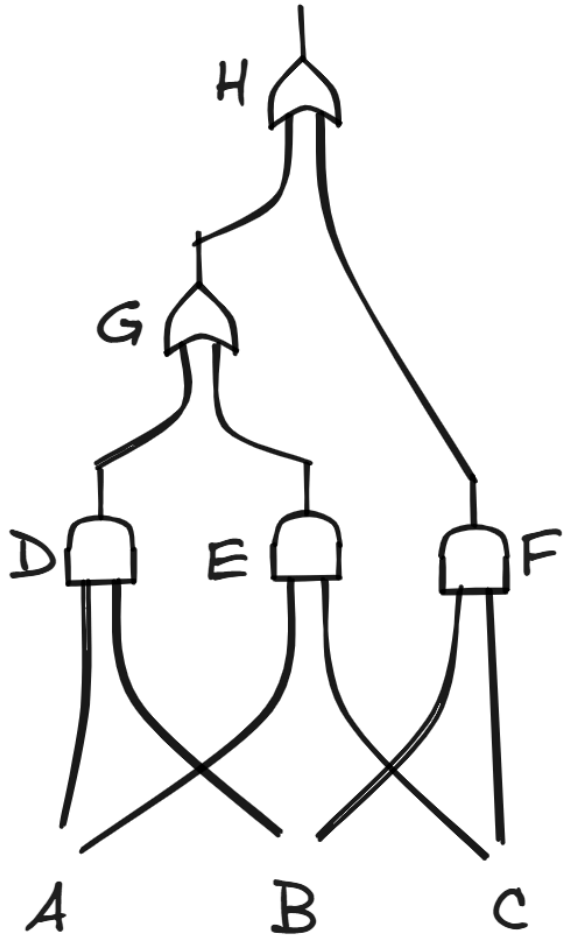
$$PARITY = \{\langle x_1, x_2, \ldots, x_n \rangle \mid \text{ na vhodu je liho stevilo enic}\}$$

Primer (večina)

$$MAJORITY = \{\langle x_1, x_2, \ldots, x_n \rangle \mid \text{ na vhodu je vsaj polovica enic}\}\$\$$$

Vezje ≡ program brez zank

```
A
B
C
D and A B
E and A C
F and B C
G or D E
H or G F
```



Model Booleovih vezij

Vezje $C = \langle V, E \rangle$ je usmerjen acikličen graf:

1. imamo n vhodnih vozlišč (vozlišč z vhodno stopnjo 0)
2. eno vozlišče z izhodno stopnjo 0, ostala vsaj 1
3. ne-vhodna vozlišča so označena z $\wedge$, $\vee$, $\neg$.
4. $\wedge$ in $\vee$ vozlišča imajo vhodno stopnjo 2, $\neg$ pa

Metrike vezja

1. število vrat
2. najdaljša pot od vhodnega do izhodnega vozlišča (**globina**)

Splošnost vezja

- vezje je definirano za fiksno dolžino vhoda
- da bi reševali nek splošen problem, definiramo družino vezij

$$C = \{C_0, C_1, C_2, \ldots\}$$

kjer ima vezje C_n n vhodov

Jezik družine vezij

Družina vezij $C = \{C_0, C_1, C_2, \ldots\}$ razpoznava jezik $L \subseteq \{0,1\}^*$, če $\forall w \in \{0,1\}^*$ velja:

$$w \in L \iff C_{|w|}(w) = 1$$

Razred kompleksnosti

$$SIZE(f(n)) = \{L \mid \text{ izračunljiv z družino vezij velikosti } O(f(n))\}$$

Polinomska vezja

$$P/poly = \bigcup_i SIZE(n^i)$$