(Motwani, Raghavan)

Markovským řetězcem se dá výpočet reprezentovat, když:

a) algoritmus je deterministický, vstupní data jsou vybrána náhodně (náš případ)

b) algoritmus je randomizovaný

Randomizovaný 2-SAT problém

problém SAT, kde každý faktor obsahuje právě 2 literály

$$(f_{11} \vee f_{12}) \wedge (f_{21} \vee f_{22}) \wedge \dots \wedge (f_{k1} \vee f_{k2})$$

kde f_{ij} je buď atomická proměnná nebo její negace

deterministický algoritmus, který prochází všechna možná přiřazení, má složitost 2^m, kde m je počet proměnných

randomizovaný algoritmus:

1) zvolíme náhodně počáteční ohodnocení proměnných
2) není-li formule splněna, vybereme náhodně 1 faktor, který není splněn; v opačném případě konec
3) v tomto faktoru vybereme náhodně 1 literál a změníme jeho ohodnocení na opačné
4) pokračujeme krokem 2)

Algoritmus je typu Monte Carlo: pro nesplnitelné formule dává vždy odpověď "nesplněno", pro splnitelné najde správné ohodnocení jen s jistou (vysokou) pravděpodobností

opačně: najde-li vyhovující ohodnocení, je odpověď "splněno" správná, nenajde-li, formule ještě nemusí být nesplnitelná

Reprezentace Markovovým řetězcem:

stavy: $0, 1, \dots, n$ budou počty správně ohodnocených proměnných

při každém průchodu cyklem se může počet správně ohodnocených proměnných o 1 zvýšit nebo snížit s pstí $1/2$

Matice pstí přechodu:

$$
\begin{array}{c}
 \\ 0 \\ 1 \\ 2 \\ \vdots \\ \\ n-1 \\ n
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} 0 & 1 & 2 & \cdots & \cdots & n-1 & n \end{array} \\
\begin{pmatrix}
0 & 1 & 0 & \cdots & \cdots & 0 & 0 \\
1/2 & 0 & 1/2 & 0 & \cdots & 0 & 0 \\
0 & 1/2 & 0 & 1/2 & \cdots & 0 & 0 \\
 & & & \vdots & & & \\
 & & & & & & \\
0 & 0 & \cdots & \cdots & 1/2 & 0 & 1/2 \\
0 & 0 & \cdots & & & 0 & 1
\end{pmatrix}
\end{array}
= P
$$

Je to vlastně symetrická náhodná procházka s jednou odrážející a jednou pohlcující stěnou.

doba výpočtu = počet kroků z počátečního stavu i do stavu m poprvé

$$P(T_m = k) = f_{i,m}^{(k)}$$

očekávaná doba výpočtu $ET_m = \sum_k k f_{i,m}^{(k)} = h_{im}$

tenhle výpočet by byl hodně složitý, jinak:

platí $h_{i,m} = \frac{1}{2}(1 + h_{i-1,m}) + \frac{1}{2}(1 + h_{i+1,m})$ pro $i \geq 1$

$$h_{0,m} = 1 + h_{1,m}$$

$$h_{m,m} = 0$$

řešíme tuhle soustavu rovnic postupným dosazováním:

$$h_{1,m} = h_{0,m} - 1$$

$$h_{2,m} = 2h_{1,m} - h_{0,m} - 2 = h_{0,m} - 4$$

$$h_{3,m} = 2h_{2,m} - h_{1,m} - 2 = 2h_{0,m} - 8 - h_{0,m} + 1 - 2 = h_{0,m} -$$

$$h_{4,m} = 2h_{3,m} - h_{2,m} - 2 = 2h_{0,m} - 18 - h_{0,m} + 4 - 2 = h_{0,m} - 1$$

$$\vdots$$

$$h_{i,m} = h_{0,m} - i^2$$

$$\vdots$$

$$h_{m,m} = h_{0,m} - m^2$$

protože $h_{m,m} = 0$, je $h_{0,m} = m^2$

$$h_{i,m} = m^2 - i^2$$

Pro první počáteční stav i je očekávaná doba výpočtu $O(m^2)$.

Pravděpodobnost chyby = P(formule je splnitelná, ale algoritmus v čase n nenajde řešení)

pro splnitelnou formuli $\sum_{t=1}^{\infty} f_{im}^{(t)} = 1$

(stav m je trvalý, v nekonečné posloupnosti pokusů dříve či později nastane s pstí 1)

P(algoritmus najde řešení v čase n) = $\sum_{t=1}^{n} f_{im}^{(t)} \to 1$ pro $n \to \infty$

P(–"– nenajde –"–) = $1 - \sum_{t=1}^{n} f_{im}^{(t)} \to 0$ pro $n \to \infty$

Je-li n dostatečně velké, je pst chyby malá %.

Poznámka. Ve skutečnosti je $p_{i,i+1} \geq \frac{1}{2}$ a $p_{i,i-1} \leq \frac{1}{2}$, protože vždycky vybíráme pouze z nesplněných fabloví.

%. Otázka je, jestli čas $O(m^2)$ bude dostatečně velký na to, aby pst chyby byla dostatečně malá!

Náhodná důkaz (Papadimitriou 1991)

Hammingova distance = rozdíl v počtu správně ohodnocených proměnných mezi správnými a aktuálním ohodnocením

rozdíl = 0 ... máme správné řešení, algoritmus končí

rozdíl = m .. všechno je špatně, algoritmus pokračuje a v následujícím kroku sníží rozdíl na m-1

matice psti přechodu bude zase náhodná procházka s jednou pohlcující stěnou (0) a jednou odrážející stěnou (m)

Interpretace v terminologii teorie her.

hrají proti sobě hráč a bankéř, v každém kroku o 1 dolar

bankéř nemůže nikdy prohrát – jakmile vyplatí všech m dolarů co má, dostane v příští hře určitě 1 dolar zpátky

naopak hráč může o svůj poslední dolar přijít definitivně

Má platit. Hráč musí být zruinován po $O(m^2)$ krocích s pstí libovolně blízkou 1

Randomizovaný k-SAT (Schöning 1999)

(tj. v každém faktoru je $\leq k$ literálů)

předchozí algoritmus se upraví následovně: opakuje se procedura A

A:
- vybere se náhodné počáteční ohodnocení
- 3m-krát se opakuje:
  - jestliže je formule splněna, tak konec
  - jinak se náhodně vybere 1 nesplněný faktor, v něm 1 literál a změní se jeho ohodnocení na opačné

Složitost procedury A je zřejmě polynomiální vzhledem k m

Označme p pravděpodobnost, že A najde správné řešení

kolikrát se bude muset A opakovat (nezávisle)? Zřejmě $\frac{1}{p}$-krát v průměrném případě

(je to geometrické rozdělení – čekáme na 1. výskyt úspěchu, pst úspěchu je p)

Pst, že po t opakováních nenajdeme řešení, je $(1-p)^t$

z analýzy: $(1-p)^t \leq e^{-pt}$

Chceme-li tedy pst chyby algoritmu např. nejvýše e^{-20}, musí být počet opakování $t = \frac{20}{p}$

autor odhadl, že $p \geq \left(\frac{1}{2}\left(1+\frac{1}{k-1}\right)\right)^m$

tedy $Et \leq \left(2\left(1-\frac{1}{k}\right)\right)^m$

speciálně pro 3-SAT je to $\left(\frac{4}{3}\right)^m$

Testování prvočíselnosti

~~patří do P, t.j. není znám deterministický algoritmus, který by jej vyřešil v polynomiálním~~ čase. Pravděpodobnostní algoritmus, který popíšeme, je založen na dvou tvrzeních, která uvedeme bez důkazu.

Označme T množinu všech dvojic přirozených čísel (k, n) takových, že $k < n$ a platí jedna z následujících podmínek

a) k^{n-1} není kongruentní s 1 mod n

b) existuje i takové, že $m = \frac{n-1}{2^i}$ je celé číslo a největší společný dělitel čísel $k^{m-1} - 1$ a n je větší než 1, ale menší než n.

Tvrzení 9.3.2.1. *Číslo n je složené právě tehdy, když existuje $k < n$ takové, že $(k, n) \in T$.*

Tvrzení 9.3.2.~~1~~2. *Nechť n je číslo složené. Pak existuje alespoň $\frac{n-1}{2}$ čísel $k < n$ takových, že $(k, n) \in T$.*

Algoritmus (Rabin-Millerův test):

VSTUP: n testované číslo, m libovolné přirozené číslo

begin
  for $i := 1$ **to** m **do**
    $k_i := Random(1, n-1)$
    if $(k_i, n) \in T$ **then** "n je složené"; *konec* **fi**
  od
  "n je prvočíslo"
end.

Důležité je, aby všechna volání funkce $Random(1, n-1)$, která vrací náhodné číslo rovnoměrně vybrané z množiny $\{1, \ldots, n-1\}$, byla navzájem nezávislá (za okamžik uvidíme, proč).

Časová složitost tohoto algoritmu je zřejmě polynomiální, skládá se z m volání *Random* a stejného počtu testů, zda daná dvojice čísel je prvkem T.

Všimněme si jedné věci, která u deterministických algoritmů nemohla nastat, a to, že algoritmus se může dopustit chyby. Chyba nastává v případě, kdy n je složené a pro všechna náhodně vygenerovaná k_i bude platit, že (k_i, n) nepatří do T, takže algoritmus nakonec rozhodne, že n je prvočíslo. Tato chyba se vyskytuje s pravděpodobností, kterou je možno předem shora ohraničit. Platí totiž

$$\begin{aligned}
P(\text{chyby}) &= P(\text{číslo složené je prohlášeno za prvočíslo}) \\
&= P((k_i, n) \text{ neleží v } T \text{ pro všechna } i = 1, \ldots, m) \\
&= \prod_{i=1}^{m} P((k_i, n) \text{ neleží v } T) \\
&\leq \left(\frac{\frac{n-1}{2}}{n-1}\right)^m = \left(\frac{1}{2}\right)^m.
\end{aligned}$$

Vhodnou volbou čísla m si tedy můžeme zajistit dostatečně malou pravděpodobnost chyby. Například pro $m = 10$ nastává chyba s pravděpodobností menší než 0.00097, pro $m = 50$ je to méně než $8 \cdot 10^{-16}$.

Třídy složitosti randomizovaných algoritmů

K formální definici se používá pravděpodobnostní Turingův stroj, který je oproti deterministickému rozšířen o možnost "hodu mincí".

Třída PP (Probabilistic Polynomial Time) je třída jazyků přijímaných pravděpodobnostním Turingovým strojem v polynomiálním čase.

Platí $NP \subseteq PP \subseteq PSPACE$.

Třída BPP (Bounded-error Probabilistic Polynomial time) je třída jazyků rozpoznávaných pravděpodobnostním Turingovým strojem v polynomiálním čase s pravděpodobností chyby ohraničenou shora nějakou konstantou $0 < \varepsilon < \frac{1}{2}$.
(Algoritmy typu Monte Carlo)
Pravděpodobnost chyby je podíl počtu výpočtů na slově x dávajících špatnou odpověď a celkového počtu výpočtů na x.

Platí $BPP \subseteq PP$.

Třída R (nebo taky RP) je "jednostranná" verze BPP. Je to třída jazyků přijímaných pravděpodobnostním Turingovým strojem v polynomiálním čase s nulovou pravděpodobností chyby pro slova, které nejsou v jazyce, a pravděpodobností chyby ohraničenou nějakým $\varepsilon < \frac{1}{2}$ pro slova v jazyce.

(Algoritmy typu Monte Carlo s jednostrannou chybou - příklad: Rabin - Millerův algoritmus na testování prvočíselnosti.)

Platí $R \subseteq BPP$, $R \subseteq NP$, $P \subseteq B$

<u>Třída ZPP</u> (Zero-error Probabilistic Polynomial Time) je třída jazyků rozpoznávaných .... s nulovou pravděpodobností chyby.

(Algoritmy typu Las Vegas – příklad: randomizovaný Quicksort.)

Platí $ZPP \subseteq R$, $ZPP = R \cap co\text{-}R$, $ZPP \subseteq NP$, $P \subseteq ZPP$

Jiný příklad - <u>náhodné procházky na grafech</u>

Nechť $G = (V, E)$ je souvislý neorientovaný graf s n vrcholy a m hranami, pro každé $v \in V$ nechť $\Gamma(v)$ je seznam sousedů vrcholu v.

souvislý graf: pro každé dva různé vrcholy ex. cesta mezi nimi

náhodná procházka je posloupnost kroků: z vrcholu v do náhodně vybraného vrcholu z $\Gamma(v)$ (tj: všechny vrcholy z $\Gamma(v)$ mají stejnou pravděpodobnost)

Nechť G je navíc úplný (tj: hrany jsou všechny dvojice navzájem různých vrcholů).

Jaký je očekávaný počet kroků z vrcholu u do vrcholu v?

Matice pravděpodobností přechodu:

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ \vdots \\ n \end{array}
\begin{array}{c}
\begin{array}{ccccc} \quad 1 \quad & \quad 2 \quad & \quad 3 \quad & \cdots & \quad n \quad \end{array} \\
\left(\begin{array}{ccccc}
0 & \frac{1}{n-1} & \frac{1}{n-1} & \cdots & \frac{1}{n-1} \\
\frac{1}{n-1} & 0 & \frac{1}{n-1} & \cdots & \frac{1}{n-1} \\
\frac{1}{n-1} & \frac{1}{n-1} & 0 & \cdots & \frac{1}{n-1} \\
 & & & & \\
- & - & - & - & - \\
 & & & & \\
\frac{1}{n-1} & \frac{1}{n-1} & \frac{1}{n-1} & \cdots & 0
\end{array}\right)
\end{array}
= P
$$

pro všechna $u \in V - \{v\}$ platí

$$h_{uv} = \sum_{\substack{w \in \Gamma(u) \\ w \neq v}} \frac{1}{n-1}(1 + h_{wv}) + \frac{1}{n-1} \cdot 1$$

soustava je splněna pro $h_{uv} = n-1$

$$\left(= (n-2)\frac{1}{n-1}(1 + n - 1) + \frac{1}{n-1} = \frac{(n-2)n}{n-1} + \frac{1}{n-1} = \right.$$

$$\left. = \frac{n(n-1) - n + 1}{n-1} = n - 1 \right)$$

Jaký je očekávaný počet kroků, abychom cestou z u navštívili všechny vrcholy grafu?

$$h_u = \frac{1}{n-1}(n-1) + \frac{1}{n-2}(n-1) + \frac{1}{n-3}(n-1) + \dots + 1 \cdot (n-1) =$$

$$= (n-1)\sum_{k=1}^{n-1}\frac{1}{n-k} = (n-1)\sum_{k=1}^{n-1}\frac{1}{k} = (n-1)H_{n-1} \sim (n-1)\log(n-1)$$

Dokážeme, že $H_n \sim \log n$:

$\forall k > 0$ a $x \in \langle k, k+1 \rangle$ platí $\frac{1}{k+1} \leq \frac{1}{x} \leq \frac{1}{k}$, protože $\frac{1}{x}$ je klesající

$$\text{z toho } \frac{1}{k+1} \leq \int_k^{k+1} \frac{1}{x}\,dx \leq \frac{1}{k}$$

$$H_n = \sum_{k=1}^{n}\frac{1}{k} \geq \sum_{k=1}^{n}\int_k^{k+1}\frac{dx}{x} = \int_1^{n+1}\frac{dx}{x} = [\log x]_1^{n+1} = \log(n+1) \geq \log n$$

$$H_n = \sum_{k=1}^{n} \frac{1}{k} = 1 + \sum_{k=1}^{n-1} \frac{1}{k+1} \le 1 + \sum_{k=1}^{n-1} \int_k^{k+1} \frac{dx}{x} = 1 + \int_1^n \frac{dx}{x} = 1 + \log n$$

tedy $\log n \le H_n \le 1 + \log n$

$$H_n \sim \log n$$

Náhodné procházky se používají např. k randomizovanému testování souvislosti grafu. Všechny vrcholy, které navštívíme cestou z u, patří do téže souvislé komponenty jako u.

Jiná úloha z Markovských řetězců: Jaký je očekávaný počet přechodů daným stavem? (Hofri)

ozn. $X_{ij}^{(k)}$ náh. veličinu, která nabývá hodnot:

$$X_{ij}^{(k)} = \begin{cases} 1 & \text{když systém půjde ze stavu } i \text{ do stavu } j \text{ po } k \text{ krocích} \\ 0 & \text{jinak} \end{cases}$$

zřejmě $EX_{ij}^{(k)} = 1 . p_{ij}^{(k)} = (P^k)_{ij}$

R_{ij} = počet přechodů stavem j, když počáteční stav je i (předp. $j \ne n$ koncový stav)

pak $R_{ij} = \sum_{k \ge 0} X_{ij}^{(k)}$ $\quad ER_{ij} = \sum_{k \ge 0} EX_{ij}^{(k)} = \sum_{k \ge 0} p_{ij}^{(k)} = \sum_{k \ge 0} (P^k)_{ij}$

Matice po této přechodu algoritmu má tvar

$$\begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & & & \\ p_{n-1,1} & p_{n-1,2} & \cdots & p_{n-1,n} \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

protože konec výpočtu je absorbční stav

nahradíme tuto matici maticí π

$$\begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & & & \\ p_{n-1,1} & p_{n-1,2} & \cdots & p_{n-1,n} \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

to je tzv. fundamentální matice algoritmu

zřejmě platí $(P^k)_{ij} = (\pi^k)_{ij}$ pro $1 \le i,j < n$

dále $\pi^m \to 0$ pro $m \to \infty$ geometricky rychle

tedy existuje součet $\sum_{i \ge 0} \pi^i = (\underline{I} - \pi)^{-1}$

nakonec máme

$$E R_{ij} = \sum_{k \ge 0} (P^k)_{ij} = \sum_{k \ge 0} (\pi^k)_{ij} = (\underline{I} - \pi)^{-1}_{ij}$$

Další použití Markovských procesů: odhad velikosti dynamických datových struktur

Mějme nějakou datovou strukturu (lineární seznam, strom, hašovací tabulku), která připouští operace Insert a Delete, tj. její velikost se s časem mění. Zajímá nás velikost (očekávaná nebo maximální) těla struktury v dlouhém časovém intervalu.

Možnosti reprezentace:

a) diskrétním Mark. řetězcem se stavy (n, x), kde n je čas a x velikost struktury (tj. počet prvků) a přechody

$$(n, x) \to (n+1, x+1)$$
$$(n, x) \to (n+1, x)$$
$$(n, x) \to (n+1, x-1)$$

Pravděpodobnosti přechodu závisí na typu datové struktury (např. v lineárním seznamu se Delete prvku dá udělat okamžitě, zatímco třeba v zásobníku ne) a poměru četnosti operací Insert a Delete.

b) systémem hromadné obsluhy $G/G/\infty$ (Lachout, Prášková) (jinak se tomu říká model teorie front)

kde G je typ rozdělení – první G je rozdělení příchodů (Inser
– druhé G je rozdělení doby obsluhy (po skončení "obsluhy" následuje Delete, doba obsluhy tedy znamená dobu setrvání prvku ve struktuře)

symbol ∞ znamená, že mám k dispozici nekonečný počet obslužných linek, takže žádný zákazník nemusí čekat (v našem případě to znamená, že Delete se neodkládá, provede se hned, jak přijde požadavek na tuto operaci)

velikost datové struktury je rovna počtu "zákazníků" v systému

V systémech hromadné obsluhy se čas bere jako spojitý, nikoli diskrétní. Pracuje se s tzv. Markovskými procesy se spojitým časem.

Def. Markovský proces se spojitým časem je systém celočíselných náh. veličin $\{X_t\}_{t\geq 0}$ (t probíhá všechna reálná nezáporná čísla), pro který platí

$$P(X_{t+h}=j \mid X_t=i,\ X_{t_r}=i_r,\ \dots,\ X_{t_1}=i_1) = P(X_{t+h}=j \mid X_t=i)$$

pro všechna $0\leq t_1<t_2<\dots<t_r<t<t+h$ a všechna přirozená $i_1, i_2, \dots, i_r, i, j$.

Podmíněné pravděpodobnosti $P(X_{t+h}=j \mid X_t=i)=p_{ij}(t,t+h)$ se nazývají pravděpodobnosti přechodu a tvoří matici pravd. přechodu $\mathbb{P}=\{p_{ij}\}_{i,j}$.

V případě, že $p_{ij}(t,t+h)=p_{ij}(h)$, tj. nezávisí na t, nazývá se proces homogenní. V opačném případě je nehomogenní.

Kromě pravd. přechodu máme ještě absolutní pravděpodobnosti $P(X_t=j)=p_j(t)$, speciálně $p_j(0)=P(X_0=j)$ je počáteční rozdělení.

Počáteční rozdělení a matice pstí přechodu plně charakterizují Mark. proces.

Další pojmy

Limitní rozdělení $\pi_j = \lim_{t\to\infty} p_{ij}(t)$, pokud tato limita existuje a nezávisí na i; navíc musí být $\sum_j \pi_j = 1$. Je to zároveň limitní rozdělení i pro absolutní psti, tj: $\pi_j = \lim_{t\to\infty} p_j(t)$.

Intenzity přechodu:

dílčí $$q_{ij} = \lim_{h\to 0_+} \frac{p_{ij}(h)}{h} \quad \text{pro } i \neq j$$

celková $$q_i = \lim_{h\to 0_+} \frac{1-p_{ii}(h)}{h}$$

platí $$q_i = \sum_{j\neq i} q_{ij}$$

(jsou to vlastně derivace p_{ij} v bodě 0, neboť $\lim_{h\to 0_+} p_{ij}(h) = \delta_{ij}$)

Matice intenzit přechodu je matice s prvky q_{ij} pro $i\neq j$, diagonální prvky se dodefinují jako $q_{ii} = -q_i$.

Mezi maticí pstí přechodu a maticí intenzit platí vztahy:

$$\mathbb{P}'(t) = Q\,\mathbb{P}(t)$$

$$\mathbb{P}'(t) = \mathbb{P}(t)\,Q$$

(tzv. Kolmogorovy diferenciální rovnice)

Dk: zřejmě platí $p_{ij}(s+t) = \sum_k p_{ik}(s) p_{kj}(t)$

(tzv. Chapman-Kolmogorova rovnost)

budeme tuto rovnost derivovat podle s:

$$\frac{\partial p_{ij}(s+t)}{\partial s} = \sum_{k \neq i} \frac{\partial p_{ik}(s)}{\partial s} p_{kj}(t) + \frac{\partial p_{ii}(s)}{\partial s} p_{ij}(t)$$

pro $s=0$ máme

$$p_{ij}'(t) = \sum_{k \neq i} q_{ik} p_{kj}(t) - q_i p_{ij}(t)$$

maticově $\mathbb{P}'(t) = Q\,\mathbb{P}(t)$, protože $q_{ii} = -q_i$

(druhou rovnici dostaneme derivováním podle t a dosazením $t=0$)

Řešení Kolmogorovových dif. rovnic je zřejmě $\mathbb{P}(t) = e^{Qt}$, takhle se ale obvykle nepočítá, musel by se použít Perronův vzorec a to je i pro malé matice komplikované.

Řešením Kolm. dif. rovnic se dají získat i absolutní psti $p_j(t)$ – to je speciální případ, kdy matice $\mathbb{P}$ má všechny řádky stejné, i limitní psti π_j – to je ještě speciálnější, kdy má matice $\mathbb{P}$ nejen všechny řádky stejné, ale navíc ještě nezávisí na t, takže $\mathbb{P}'(t) = \mathbb{O}$.

Takže $\mathbb{O} = Q\pi = \pi Q = Q^T \pi$.

Nechť pro datovou strukturu platí:

Inserty nastávají náhodně v čase tak, že v krátkém časovém intervalu $(t, t+h)$ nastane Insert s pstí $\lambda h + o(h)$, nezávisle na t a na počtu insertů v intervalu $(0, t)$ (jednotlivých prvků jsou navzájem nezávislé a)

Pravděpodobnost operace Delete (tj. ukončení doby setrvání prvku ve struktuře) je $\mu h + o(h)$ – opět nezávisle na t a na počtu deletů v $(0, t)$, navíc nechť delety jednotlivých prvků jsou na sobě nezávislé.

Pst více než jedné operace v $(t, t+h)$ je $o(h)$ (operaci Member nepočítáme, ta nemění velikost datové struktury).

(Tyto předpoklady jsou ekvivalentní tomuto: systém $M/M/\infty$

Intervaly mezi Inserty jsou navzájem nezávislé náh. veličiny s exponenciálním rozdělením s parametrem λ, tj:

$F(I) = 1 - e^{-\lambda I}$, $f(I) = \lambda e^{-\lambda I}$, $EI = \frac{1}{\lambda}$ = očekávaná doba mezi dvěma následujícími Inserty.

Doba setrvání prvku ve struktuře má exponenciální rozdělení s parametrem μ, tj. $\frac{1}{\mu}$ je očekávaná doba setrvání prvku, pro jednotlivé prvky je to stejné a navzájem nezávislé.)

Je-li v čase t ve struktuře j prvků, pak v čase $t+h$ jich bude

$j+1$ s pstí $\lambda h + o(h)$

$j-1$ s pstí $j\mu h + o(h)$

j s pstí $1 - (\lambda h + j\mu h + o(h))$

jiný počet s pstí $o(h)$

Intenzity přechodu jsou tedy:

$$q_{j,j+1} = \lambda$$

$$q_{j,j-1} = j\mu$$

$$q_j = -q_{jj} = \lambda + j\mu$$

v ostatních případech 0

Matice intenzit přechodu:

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \dots & & & & & \\ \mu & -(\lambda+\mu) & \lambda & 0 & \dots & & & & & \\ 0 & 2\mu & -(\lambda+2\mu) & \lambda & 0 & \dots & & & & \\ \dots & \dots & \dots & \dots & \dots & & & & & \\ 0 & \dots & \dots & \dots & \dots & 0 & j\mu & -(\lambda+j\mu) & \lambda & 0 \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix}$$

Vypočteme limitní pstí $\pi_j = \lim_{t\to\infty} p_j(t) = \lim_{t\to\infty} p_{ij}(t)$:

z rovnice $Q^T \pi = 0$

$$\begin{pmatrix} -\lambda & \mu & 0 & \dots & 0 & \dots \\ \lambda & -(\lambda+\mu) & 2\mu & & \vdots & \\ 0 & \lambda & -(\lambda+2\mu) & & \vdots & \\ 0 & 0 & \lambda & & \vdots & \\ \vdots & \vdots & \vdots & & 0 & \\ \vdots & \vdots & 0 & & j\mu & \\ \vdots & \vdots & \vdots & & -(\lambda+j\mu) & \\ \vdots & \vdots & \vdots & & \lambda & \\ & & & & 0 & \end{pmatrix} \begin{pmatrix} \pi_0 \\ \pi_1 \\ \pi_2 \\ \vdots \\ \pi_j \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \end{pmatrix}$$

dostaneme: $-\lambda \pi_0 + \mu \pi_1 = 0$

$$\lambda \pi_0 - (\lambda + \mu)\pi_1 + 2\mu \pi_2 = 0$$

- - - -

$$\lambda \pi_{j-1} - (\lambda + j\mu)\pi_j + (j+1)\mu \pi_{j+1} = 0$$

- - - -

Řešení trikem:

zavedeme veličiny $k_j = j\mu \pi_j - \lambda \pi_{j-1}$

soustavu pak můžeme přepsat jako: $k_1 = 0$

$$k_{j+1} = k_j \quad \text{pro } j \geq 1$$

Tedy $k_j = k_{j-1} = \dots = k_1 = 0$

z toho $j\mu \pi_j - \lambda \pi_{j-1} = 0$

$$\pi_j = \frac{\lambda}{j\mu} \pi_{j-1}$$

postupně dosazujeme $\pi_j = \frac{\lambda}{j\mu}\pi_{j-1} = \frac{\lambda^2}{j(j-1)\mu^2}\pi_{j-1} = \dots = \left(\frac{\lambda}{\mu}\right)^j \frac{1}{j!}\pi_0$

π_0 se určí z podmínky $\sum_{j=0}^{\infty} \pi_j = 1 = \pi_0 \sum_{j=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^j \frac{1}{j!} = \pi_0 e^{\frac{\lambda}{\mu}}$

$$\Rightarrow \pi_0 = e^{-\frac{\lambda}{\mu}}$$

Limitní psti, že ve struktuře je j prvků, jsou $\pi_j = e^{-\frac{\lambda}{\mu}} \left(\frac{\lambda}{\mu}\right)^j \frac{1}{j!}$

To je Poissonovo rozdělení s parametrem $\frac{\lambda}{\mu}$, jehož střední hodnota i rozptyl jsou $\frac{\lambda}{\mu}$.

Očekávaný počet prvků ve struktuře je tedy $\frac{\lambda}{\mu}$.