

Petriho síť

©Zdeněk Hanzálek
hanzalek@fel.cvut.cz

16. června 2008

Obsah

1	Autonomní Petriho sítě	5
1.1	Základní koncept obecné Petriho sítě	6
1.1.1	Neoznačená Petriho síť	6
1.1.2	Označená Petriho síť	6
1.1.3	Autonomní a neautonomní Petriho sítě	6
1.1.4	Grafický popis	7
1.2	Vývoj stavu Petriho sítě	8
1.2.1	Uvolnění a přeskok přechodu	8
1.2.2	Konflikt	9
1.3	Vlastnosti Petriho sítí a jejich analýza	10
1.3.1	Vlastnosti závislé na značení	10
1.3.1.1	Ohraničená Petriho Síť	10
1.3.1.2	Živá Petriho síť	10
1.3.1.3	Reverzibilní Petriho síť	12
1.3.2	Vlastnosti nezávislé na značení	12
1.3.2.1	P-invarianty	12
1.3.2.2	T-invarianty	17
1.4	Podtřídy, zkrácené a rozšířené Petriho sítě	17
1.4.1	Podtřídy Petriho sítí	17
1.4.1.1	Stavový graf	17
1.4.1.2	Značený graf	18
1.4.2	Zkrácené Petriho sítě	20
1.4.2.1	Kapacitní Petriho sítě	20
1.4.2.2	Barvené Petriho sítě	20
1.4.3	Rozšířené Petriho sítě	25
1.4.3.1	Petriho síť s inhibitovanou hranou	25
1.5	Poznámky a reference	27
1.6	Závěr	27

Kapitola 1

Autonomní Petriho sítě

Klasická teorie systémů se tradičně zabývá zejména systémy dobře popsányými mechanikou, teorií obvodů, fyzikální chemií a dalšími vědami jež pracují s veličinami jako je rychlost, zrychlení, napětí, proud, tlak, teplota, průtok atd. Každá z těchto veličin je zpravidla reprezentována *spojitým stavem*, jenž je definován na oboru reálných čísel. Na tomto základě byla vyvinuta řada nástrojů a technik pro modelování, analýzu a řízení systémů kolem nás. Základní rámec tohoto oboru, zpravidla nazývaného *spojité systémy*, se opírá o obyčejné a parciální diferenciální rovnice. Pro systémy, kde je vývoj stavů vzorkován v ekvidistantních časových úsecích, potom zpravidla využíváme diferenční rovnice a nazýváme je trochu nepřesně *diskrétní systémy*. Nepřesnost spočívá ve slově *diskrétní*, jelikož stavy těchto systémů jsou spojité a události se dějí v ekvidistantních okamžicích (neboli v diskretním čase).

Nicméně ve světě reálných systémů kolem nás, které jsou stále více závislé na počítačích, si povšimneme dvou podstatných skutečností. Zaprvé, řada stavů má diskretní charakter (kolik je výrobků ve skladu, mám/nemám potvrdit zprávu, kolikrát přišla náběžná hrana, stav v dotazníku svobodná/vdaná/mrtvá, stav těhotenský - jiný stav ...) vedoucí na reprezentaci celými nezápornými čísly. Zadruhé řada těchto systémů je řízena diskretními událostmi ve spojitém čase (příchod zprávy před timeoutem, stisk tlačítka myši při dvojkliku, příchod přerušeni,...). Proto se pro tyto systémy vžil název *systémy diskretních událostí* (DES, discrete event systems).

Systémy diskretních událostí jsou typicky popisovány pomocí stavových automatů. Stavové automaty jsou dány množinou stavů, přechodovou funkcí a jedním aktuálním stavem. V případě distribuovaných systémů, kde je potřeba popisovat aktivity jednotlivých podsystémů a jejich vzájemné vztahy, může být modelování stavovými automaty těžkopádné. Je to proto, že každá možná kombinace stavů jednotlivých podsystémů (product of automata) potřebuje vlastní stav ve výsledném stavovém automatu, jehož rozměr může být exponenciálně závislý na počtu podsystémů. Analýza vlastností se poté odehrává na úrovni tohoto výsledného stavového automatu.

Zmíněný nedostatek částečně odstraňuje jiný nástroj pro modelování a analýzu systémů diskretních událostí, nazvaný Petriho sítě podle německého matematika C.A.Petri, jenž položil v roce 1962 základ tomuto oboru svou disertační prací [7]. Analýzu vlastností Petriho

sítě lze v některých případech provést bez vyčíslení grafu dosažitelných značení, který odpovídá výslednému stavovému automatu.

1.1 Základní koncept obecné Petriho sítě

1.1.1 Neoznačená Petriho síť

Neoznačená Petriho síť je uspořádaná čtveřice $(\mathcal{P}, \mathcal{T}, Pre, Post)$ kde \mathcal{P} je konečná množina míst (place), neboli $\mathcal{P} = \{p_1, \dots, p_i, \dots, p_m\}$, \mathcal{T} je konečná množina přechodů (transitions), neboli $\mathcal{T} = \{t_1, \dots, t_j, \dots, t_n\}$, Pre je matice zobrazení $\mathcal{P} \times \mathcal{T} \rightarrow \mathcal{Z}_0^+$ reprezentující spojení přechodu s předchozím místem (precondition) a $Post$ je matice zobrazení $\mathcal{P} \times \mathcal{T} \rightarrow \mathcal{Z}_0^+$ reprezentující spojení přechodu s následujícím místem (postcondition). Matice Pre a $Post$, reprezentující váhy hran, často souhrnně zapisujeme pomocí incidenční matice $W = Post - Pre$.

Pozor na skutečnost, že incidenční matice neumožňuje popis sítě, ve které existuje dvojice hran typu "self-loop" (neboli existuje jedna hrana z místa p_i do přechodu t_j a zároveň existuje druhá hrana z t_j do p_i). Jako příklady takové struktury Petriho sítě uveďme spojení přechodu t_6 s místem p_5 na obrázku 1.5b) nebo spojení přechodu t_5 s místem p_1 na obrázku 1.12.

Pro zjednodušení zápisu uveďme další značení. Množina \mathcal{t}_j zahrnuje místa vstupující do přechodu t_j . Neboli $p_i \in \mathcal{t}_j \Leftrightarrow Pre(p_i, t_j) \neq 0$. Množina \mathcal{t}_j° zahrnuje místa vystupující z přechodu t_j . Neboli $p_i \in \mathcal{t}_j^\circ \Leftrightarrow Post(p_i, t_j) \neq 0$. Množina \mathcal{p}_i zahrnuje přechody vstupující do místa p_i . Neboli $t_j \in \mathcal{p}_i \Leftrightarrow Post(p_i, t_j) \neq 0$. Množina \mathcal{p}_i° zahrnuje přechody vystupující z místa p_i . Neboli $t_j \in \mathcal{p}_i^\circ \Leftrightarrow Pre(p_i, t_j) \neq 0$. Vektory $Pre(:, t_j)$ resp. $Post(:, t_j)$ reprezentují j -tý sloupec matice Pre resp. $Post$. Řádkové vektory $Pre(p_i, :)$ resp. $Post(p_i, :)$ reprezentují i -tý řádek matice Pre resp. $Post$.

1.1.2 Označená Petriho síť

Označená Petriho síť je uspořádaná pětice $(\mathcal{P}, \mathcal{T}, Pre, Post, m_0)$ kde m_0 je vektor zobrazení $\mathcal{P} \rightarrow \mathcal{Z}_0^+$ reprezentující počáteční značení (initial marking). Prvek $m_0(p_i)$ reprezentuje počet tokenů (značek, pešků) obsažených v místě p_i na počátku vývoje Petriho sítě. Vektory m, m', m_1, m_2, \dots reprezentují možná značení Petriho sítě v průběhu vývoje systému.

1.1.3 Autonomní a neautonomní Petriho síť

Pokud Petriho síť popisuje systém, jež se chová autonomně (neboli nezávisle na okolním prostředí včetně času), potom hovoříme o autonomní Petriho síti. *Autonomní Petriho síť* popisuje události v systému z kvalitativního hlediska, ale nekvantifikuje, kdy k nim dojde v návaznosti na okolním prostředí. *Neautonomní Petriho síť* popisuje funkci systému, jehož vývoj je podmíněn externími vstupy (synchronizovaná Petriho síť) nebo časem (časovaná nebo stochastická nebo časová Petriho síť).

V kapitole 1 se zabýváme pouze autonomní Petriho sítí, jež umožňuje studovat všechna možná chování systému daného uspořádanou pětici $(\mathcal{P}, \mathcal{T}, Pre, Post, m_0)$ (respektive mírného rozšíření struktury o inhibovanou hranu).

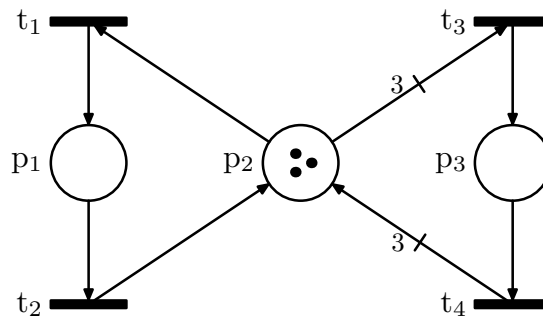
Příklad: Bit alternating protocol - Barthomieu

1.1.4 Grafický popis

Grafická reprezentace je asi nejrozšířenější způsob předávání informací mezi inženýry. Zvláště ve fázi tvorby koncepce realizovaného systému má svou nezastupitelnou úlohu. Jelikož většina námi realizovaných aplikací má distribuovaný charakter a interaguje s okolím, je potřeba reprezentovat jednotlivé podsystémy, mít možnost ověřit jejich funkci (například simulací nebo strukturální analýzou) a možnost spojovat tyto podsystémy. Pro takový postup jsou Petriho sítě velmi vhodné, jelikož mají jednoznačnou grafickou reprezentaci, umožňují propojovat jednotlivé podsystémy a mají oporu v matematickém formalismu, který usnadňuje počítačové zpracování včetně analýzy vlastností.

Neoznačená Petriho síť je orientovaný ohodnocený bipartitní graf. Jako každý graf se i Petriho síť skládá z uzlů, jež jsou propojeny hranami. Přívlastek orientovaný značí skutečnost, že hrany grafu jsou orientované. Přívlastek ohodnocený značí skutečnost, že hranám mohou být přiřazeny váhy. Přívlastek bipartitní značí skutečnost, že množina uzlů grafu se skládá ze dvou disjunktních podmnožin - množiny míst \mathcal{P} a množiny přechodů \mathcal{T} , přičemž místa a přechody se v průběhu cesty střídají.

Mějme tři zdroje (například tři komunikační linky, které máme k dispozici), jež jsou používány buď běžnými uživateli jednotlivě (jeden proces obsadí jednu komunikační linku) nebo náročnými uživateli jako trojice (jeden proces obsadí tři komunikační linky). Zároveň platí, že pokud je zdroj používán uživatelem, pak nemůže být používán jiným uživatelem.



Obrázek 1.1: Sdílení zdrojů

Tento mechanismus lze jednoduše modelovat pomocí Petriho sítě na obrázku 1.1 obsahující tři místa a čtyři přechody, neboli $\mathcal{P} = \{p_1, p_2, p_3\}$ a $\mathcal{T} = \{t_1, t_2, t_3, t_4\}$. Místo p_1 reprezentuje zdroje používané běžnými uživateli, místo p_2 reprezentuje volné zdroje a místo p_3 reprezentuje zdroje používané náročnými uživateli. Odebírání zdrojů reprezentují přechody t_1 a t_3 , vrácení zdrojů reprezentují přechody t_2 a t_4 . Jelikož náročný uživatel odebírá

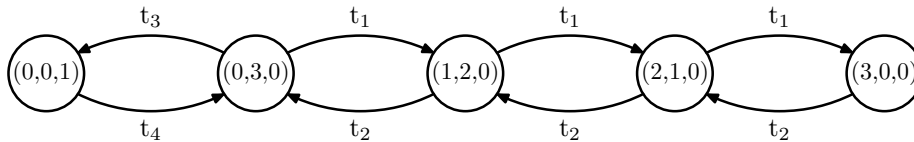
respektive vrací tři zdroje naráz, je váha hrany $Pre(p_2, t_3)$ respektive váha hrany $Post(p_2, t_4)$ rovna 3. Takto namodelovaný rezervační mechanismus lze snadno napojit na Petriho síť modelující chování uživatelů před odebráním zdroje (jako další vstupní hrany do přechodu t_1 resp. t_3) nebo po odebrání zdroje (jako další výstupní hrany z přechodu t_2 resp. t_4).

Petriho síť na obrázku 1.1 lze jednoznačně zapsat maticemi Pre a $Post$, které udávají její strukturu a vektorem m_0 , který udává její značení:

$$\left. \begin{array}{l} Pre = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ Post = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \right\} \Rightarrow W = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & -3 & 3 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad m_0 = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$$

1.2 Vývoj stavu Petriho sítě

Vývoj Petriho sítě je reprezentován přesunem značek v síti na základě aktivace přechodů. Každé nové značení tohoto systému diskrétních událostí reprezentuje nový stav systému (neboli dané místo v Petriho síti nepředstavuje jeden stav systému, ale hodnota jeho značení je součástí identifikátoru každého ze stavů). Obrázek 1.2 ukazuje graf dosažitelných značení Petriho sítě na obrázku 1.1. Ilustruje skutečnost, že graf dosažitelných značení autonomní Petriho sítě a stavový automat jsou dva ekvivalentní popisy téhož systému diskrétních událostí.



Obrázek 1.2: Graf dosažitelných značení Petriho sítě na obrázku 1.1

1.2.1 Uvolnění a přeskok přechodu

Přechod t_j je *uvolněn* (enabled, uschopněn) při značení m právě tehdy, když pro všechna místa p_i vstupující do přechodu t_j (dále jen vstupní místa přechodu t_j) platí, že počet tokenů je větší nebo roven váze hrany z p_i do t_j . Neboli t_j je uvolněn při značení $m \Leftrightarrow \forall p_i \in \mathcal{I}_{t_j}; m(p_i) \geq Pre(p_i, t_j)$. S odkazem na strukturu matice Pre lze tuto podmínku zapsat jako $\forall p_i \in \mathcal{P}; m(p_i) \geq Pre(p_i, t_j)$.

Pokud je přechod uvolněn, potom může být přeskočen. *Přeskok* (firing, zapálení, provedení) přechodu t_j odejme tokeny z míst vstupujících do přechodu a vloží tokeny do míst vystupujících z přechodu. Neboli $\forall p_i \in \mathcal{P}; m'(p_i) = m(p_i) + Post(p_i, t_j) - Pre(p_i, t_j)$, kde $m(p_i)$ je počet tokenů v místě p_i před přeskokem a $m'(p_i)$ je počet tokenů po přeskoku

přechodu t_j . Přeskok je nerozdělitelná akce (pro jednoduchost můžeme předpokládat, že trvá nulovou dobu).

Například na obrázku 1.1 po přeskoky přechodu t_1 je výsledné značení

$$m' = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$$

Jelikož chování autonomních Petriho sítí není vztaheno k okolnímu prostředí (tedy ani k času), není v nich determinováno, kdy je přechod přeskočen. Toto je důležité zjištění zejména v případě, že v daném stavu je v autonomní Petriho síti uvolněno více přechodů a graf dosažitelných značení pak obsahuje alternativní cesty odpovídající různým sekvencím přeskoků uvolněných přechodů. V autonomních Petriho sítích jakoby přechod byl uvolněn po libovolnou dobu z intervalu $[0, \infty]$ do okamžiku, kdy je přeskočen nebo přestane být uvolněn díky přeskoky jiného přechodu.

Přeskoková sekvence S (uschopněná přechodová sekvence) je sekvence přechodů, jež byly přeskočeny v průběhu vývoje systému. Například pro systém na obrázku 1.1 uvažme vývoj značení $m_0 \xrightarrow{t_1} m_1 \xrightarrow{t_1} m_2 \xrightarrow{t_1} m_3 \xrightarrow{t_2} m_4$. Odpovídající přeskoková sekvence je $S = t_1 t_1 t_1 t_2$. Neboli říkáme, že přeskoková sekvence S vede na změnu značení z m_0 na m_4 a píšeme $m_0 \xrightarrow{S} m_4$.

Značení m je *dosažitelným značením* právě tehdy, když existuje přeskoková sekvence S , která vede na změnu značení z m_0 na m .

Charakteristický vektor s udává kolikrát je daný přechod zastoupen v průběhu přeskokové sekvence S . Neboli výše zmíněné přeskokové sekvenci S odpovídá charakteristický vektor $s = [3, 1, 0, 0]^T$.

Vývoj Petriho sítě z počátečního značení m_0 do značení m je popsán *stavovou rovnicí*:

$$m = m_0 + (Post - Pre) \cdot s = m_0 + W \cdot s \quad (1.1)$$

1.2.2 Konflikt

Konflikt vyjadřuje nedeterminismus chování systému, neboli možnost volby (například volba zda sdílený zdroj na obrázku 1.1 bude využit jedním nebo druhým uživatelem). V časované Petriho síti dokonce konflikt může způsobit to, že doba vykonání cyklu je nepřímou úměrnou době přeskoky jednoho přechodu (souvisí s pojmem *rozvrhovací anomálie*).

Petriho síť má *strukturální konflikt* v místě p_i právě tehdy, když existují alespoň dva přechody vystupující z tohoto místa. Neboli struktura $\langle p_i, \{t_j, t_k\} \rangle$ je strukturální konflikt $\Leftrightarrow Pre(p_i, t_j) \cdot Pre(p_i, t_k) \neq 0$.

Označená Petriho síť má *efektivní konflikt* v místě p_i pro značení m právě tehdy, když má strukturální konflikt v místě p_i a $m(p_i)$ nepostačuje pro přeskok všech uvolněných přechodů vystupujících z tohoto místa. Neboli struktura $\langle p_i, \{t_j, t_k\} \rangle$ a značení m popisují efektivní konflikt $\Leftrightarrow \langle p_i, \{t_j, t_k\} \rangle$ je strukturální konflikt a t_j je uvolněn a t_k je uvolněn a $m(p_i) < Pre(p_i, t_j) + Pre(p_i, t_k)$. Příklad efektivního konfliktu je na obrázku 1.1.

1.3 Vlastnosti Petriho sítí a jejich analýza

Poté, co namodelujeme systém diskretních událostí pomocí Petriho sítě, se začneme zajímat o to, k čemu je takový model vhodný. Vedle základního využití pro simulaci se naskytá možnost použít model pro *analýzu vlastností Petriho sítě*. Formální analýza je důležitá pro návrh kritických aplikací, kde se nelze spokojit s pouhým testováním, ale je potřeba podchytit všechny dosažitelné stavy. Připomeňme, že dosažitelné značení m (respektive odpovídající dosažitelný stav) je značení, pro které existuje přeskoková sekvence S taková, že $m_0 \xrightarrow{S} m$.

Na úrovni abstraktního modelu máme možnost prokázat funkčnost a bezchybnost systému ještě před jeho praktickou realizací. Na tomto místě je dlužno podotknout, že tvorba modelu tak, aby podchytil podstatné rysy aplikace, je úkol neméně náročný.

Petriho sítě jsou vhodným nástrojem pro analýzu systémů diskretních událostí. Neboli poté, co systém namodelujeme (vytvoříme jeho grafickou reprezentaci a z ní odvodíme maticový popis struktury systému a vektor počátečního značení), máme možnost prokázat, jaké má systém vlastnosti – například, zda se nedostane do deadlocku (stavu, ve kterém zůstane zablokovaný). V následujícím textu uvedeme některé vlastnosti Petriho sítí závislé na značení a některé vlastnosti dané pouze strukturou Petriho sítě.

1.3.1 Vlastnosti závislé na značení

1.3.1.1 Ohraničená Petriho Sít'

Značená Petriho síť je ohraničená (omezená) právě tehdy, když pro libovolné dosažitelné značení je počet tokenů v každém místě shora ohraničen konečnou konstantou (viz Petriho síť na obrázku 1.3 vlevo, kde $m(p_1) \leq 3$, $m(p_2) \leq 3$ a $m(p_3) \leq 1$). Pro každou ohraničenou Petriho síť je množina dosažitelných stavů konečná a lze ji realizovat konečným automatem. Neboli Petriho síť $(\mathcal{P}, \mathcal{T}, Pre, Post, m_0)$ je ohraničená $\Leftrightarrow \forall m ; m_0 \rightarrow m$ a $\forall p_i \in \mathcal{P} \exists k \neq \infty ; m(p_i) \leq k$. Taková síť se často nazývá k -ohraničená.

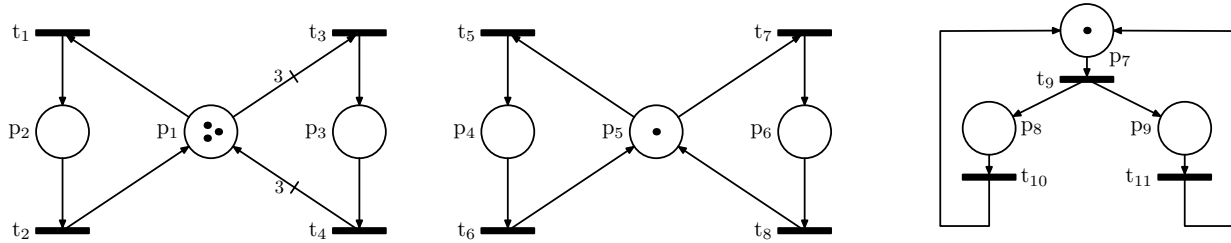
Pokud síť není ohraničená, viz příklad na obrázku 1.3 vpravo, pak značení některého místa může vzrůst nade všechny meze. Následkem toho roste množina dosažitelných značení nade všechny meze a takový systém nelze realizovat konečným automatem.

Říkáme, že Petriho síť je *binární* (binary, safe) právě tehdy, když je 1-ohraničená, neboli značení každého místa lze reprezentovat jedním bitem. Příklad binární Petriho sítě je na obrázku 1.3 uprostřed.

1.3.1.2 Živá Petriho síť

Značená Petriho síť je *pseudoživá* právě tehdy, když každý její přechod může být alespoň jednou přeskočen. Neboli Petriho síť $(\mathcal{P}, \mathcal{T}, Pre, Post, m_0)$ je pseudoživá $\Leftrightarrow \forall t_j \in \mathcal{T} ; t_j$ je pseudoživý. Přechod t_j je pseudoživý $\Leftrightarrow \exists S ; m_0 \xrightarrow{S} m$ a $\forall p_i \in \mathcal{P} ; m(p_i) \geq Pre(p_i, t_j)$.

Značená Petriho síť je *živá* (live) právě tehdy, když pro každé dosažitelné značení a pro každý její přechod existuje přeskoková sekvence, která tento přechod uvolní. Neboli Petriho

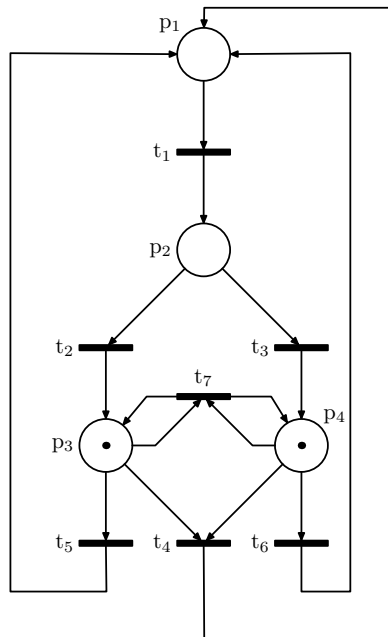


Obrázek 1.3: Ohraničená, binární a neohraničená Petriho síť

síť $(\mathcal{P}, \mathcal{T}, Pre, Post, m_0)$ je živá $\Leftrightarrow \forall t_j \in \mathcal{T}$; t_j je živý. Přejchod t_j je živý $\Leftrightarrow \forall m$; $m_0 \xrightarrow{S} m \exists S'$; $m \xrightarrow{S'} m'$ a $\forall p_i \in \mathcal{P}$; $m'(p_i) \geq Pre(p_i, t_j)$.

Uvážnutím (deadlockem) zpravidla rozumíme stav systému, kdy žádný z uvažovaných přechodů nemůže být přeskočen. Pokud je Petriho síť živá, pak se nikdy nedostane do deadlocku a ani žádná její část se nikdy nedostane do deadlocku.

Obrázek 1.4 ukazuje Petriho síť ve které jsou přechody t_1, t_2, t_3, t_5, t_6 živé a přechody t_4, t_7 jsou pseudoživé. Přejchod t_4 je přeskočitelný právě jednou, jelikož po jeho přeskočení je v celé síti obsažen jeden token, který nestačí k jeho dalšímu přeskočení. Přejchod t_7 je přeskočitelný do té doby, než je přeskočen přechod t_4 .



Obrázek 1.4: Živost Petriho sítě

1.3.1.3 Reverzibilní Petriho síť

Značená Petriho síť je reverzibilní (reversible) právě tehdy, když pro každé dosažitelné značení existuje přeskoková sekvence, která uvede síť do původního značení. Neboli Petriho síť $(\mathcal{P}, \mathcal{T}, Pre, Post, m_0)$ je reverzibilní $\Leftrightarrow \forall m; m_0 \xrightarrow{S} m \exists S'; m \xrightarrow{S'} m_0$.

Žádná z výše zmiňovaných vlastností (ohraničenost, živost a reverzibilita) neimplikuje vlastnost druhou. Jednoduchý důkaz tohoto tvrzení ukázal T.Murata v osmi Petriho sítích uvedených na obrázku 1.5. Každá z nich ukazuje odlišnou kombinaci tří vlastností značené Petriho sítě (živost, ohraničenost, reverzibilitu).

1.3.2 Vlastnosti nezávislé na značení

Pokud model sdílení zdrojů na obrázku 1.1 rozšíříme o jednoho běžného a o jednoho náročného uživatele, kteří opakovaně odebírají a vracejí sdílené zdroje, pak obdržíme model na obrázku 1.6.

Všimněme si, že pro libovolné dosažitelné značení m platí rovnice $m(p_1) + m(p_2) = m_0(p_1) + m_0(p_2) = 1$ reprezentující chování běžného uživatele (stejně platí i rovnice $m(p_4) + m(p_5) = m_0(p_4) + m_0(p_5) = 1$ pro náročného uživatele). Tato rovnice má jednoznačný význam: za každé situace, uživatel buď zdroj má ($m(p_2) = 1$) nebo nemá ($m(p_1) = 1$). Dále si povšimněme, že pro sdílené zdroje platí rovnice $m(p_2) + m(p_3) + 3 \cdot m(p_4) = m_0(p_2) + m_0(p_3) + 3 \cdot m_0(p_4) = 3$. Takový, na značení nezávislý, orientovaný sled míst (v případě běžného uživatele jde o uzavřený orientovaný sled ve kterém se neopakuje žádné místo, v případě sdílených zdrojů jde o uzavřený orientovaný sled s opakováním místa p_3) představuje matematický základ pro strukturální analýzu Petriho sítě.

1.3.2.1 P-invarianty

Množina míst se nazývá *konzervativní komponenta* právě tehdy, když celočíselná pozitivní lineární kombinace počtu tokenů obsažených v těchto místech je konstantní pro libovolné dosažitelné značení.

P-invarianta f^T je vektor celých nezáporných čísel charakterizující výše zmíněnou konzervativní komponentu. Neboli Petriho síť $(\mathcal{P}, \mathcal{T}, Pre, Post)$ má P-invariantu $f^T; f_i \in \mathbb{Z}_0^+ \forall i \in \{1 \dots \|\mathcal{P}\|\}$ právě tehdy, když $\forall m; m_0 \xrightarrow{S} m; f^T \cdot m = f^T \cdot m_0$.

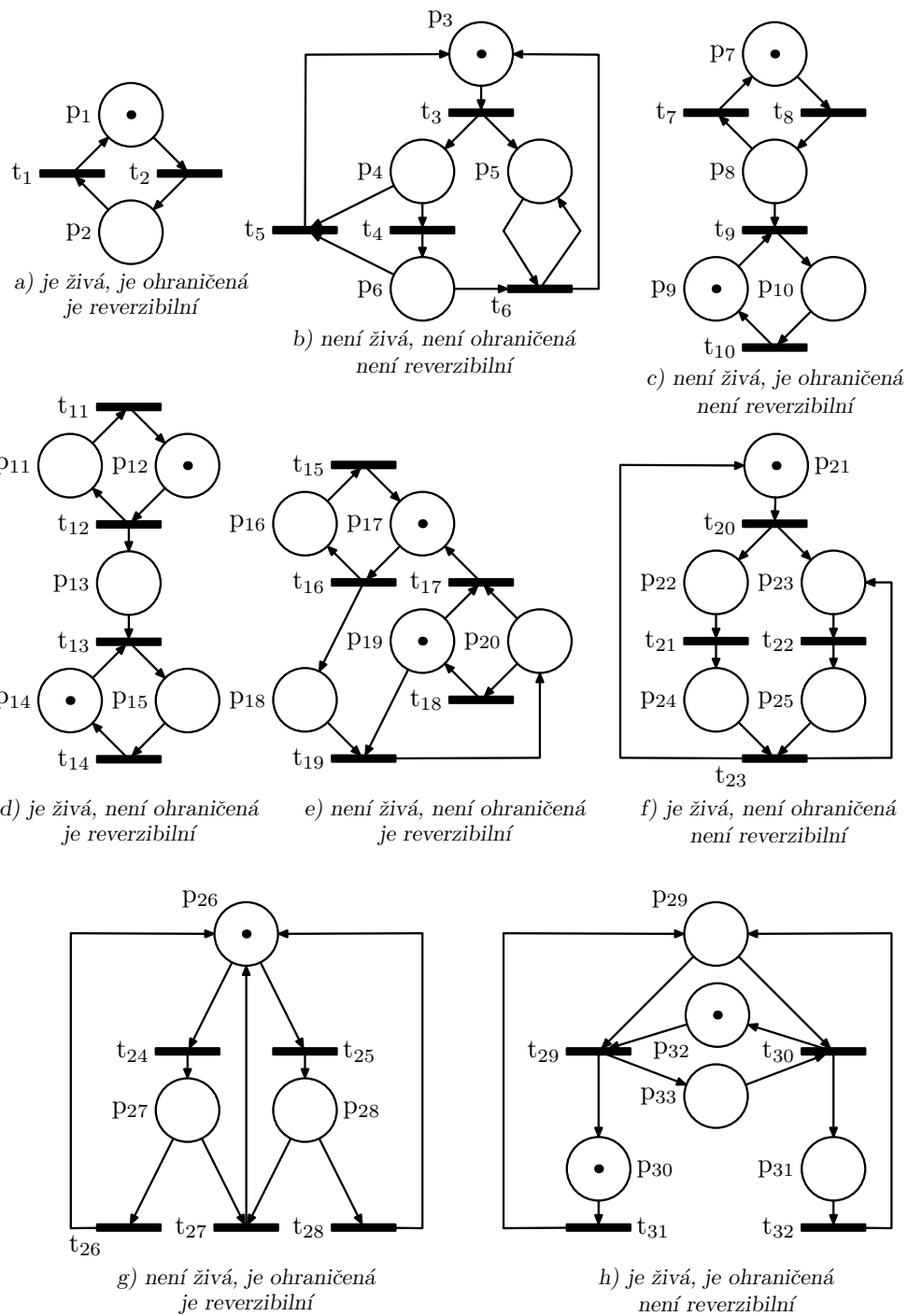
V následujícím textu jednoduše odvodíme, že P-invarianta je dána strukturou Petriho sítě. Vektorem f^T vynásobíme zleva stavovou rovnici:

$$f^T \cdot m = f^T \cdot m_0 + f^T \cdot W \cdot s \quad (1.2)$$

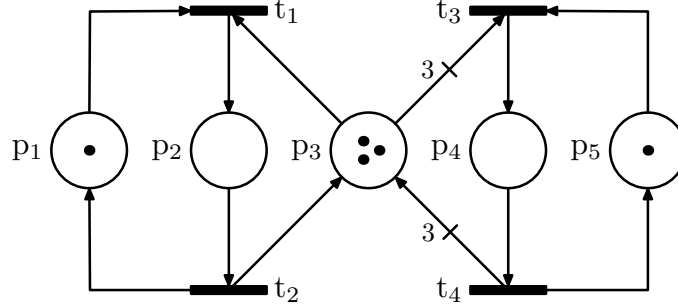
Má-li být vektor f^T P-invariantou, potom musí splňovat podmínku $f^T \cdot m = f^T \cdot m_0$ pro všechny přeskokové sekvence, neboli $\forall s$. Proto je vektor f^T P-invariantou právě tehdy, když:

$$f^T \cdot W = 0 \quad \text{kde } f_i \in \mathbb{Z}_0^+ \forall i \in \{1 \dots \|\mathcal{P}\|\} \quad (1.3)$$

jelikož tím se výše upravená stavová rovnice (1.2) změní na podmínku P-invarianty.



Obrázek 1.5: Příklady demonstrující různé kombinace živosti, ohraničenosti a reverzibility



Obrázek 1.6: Petriho síť se třemi minimálními P-invariantami

Neboli pro daný vektor f^T jsme schopni ověřit, zda jde o P-invariantu prostým dosazením do vztahu 1.3. Pokud ovšem chceme k dané Petriho síti nalézt nějakou jednoduchou charakteristiku všech jejích P-invariant, potom je postup mnohem náročnější. V tomto ohledu připomeňme, že každá celočíselná nezáporná lineární kombinace několika P-invariant je opět P-invariantou. S ohledem na definiční obor vektoru f^T je vztah 1.3 soustavou nerovnic (zahrnuje $f_i \geq 0 \forall i \in \{1 \dots \|\mathcal{P}\|\}$) a její řešení není triviální záležitostí.

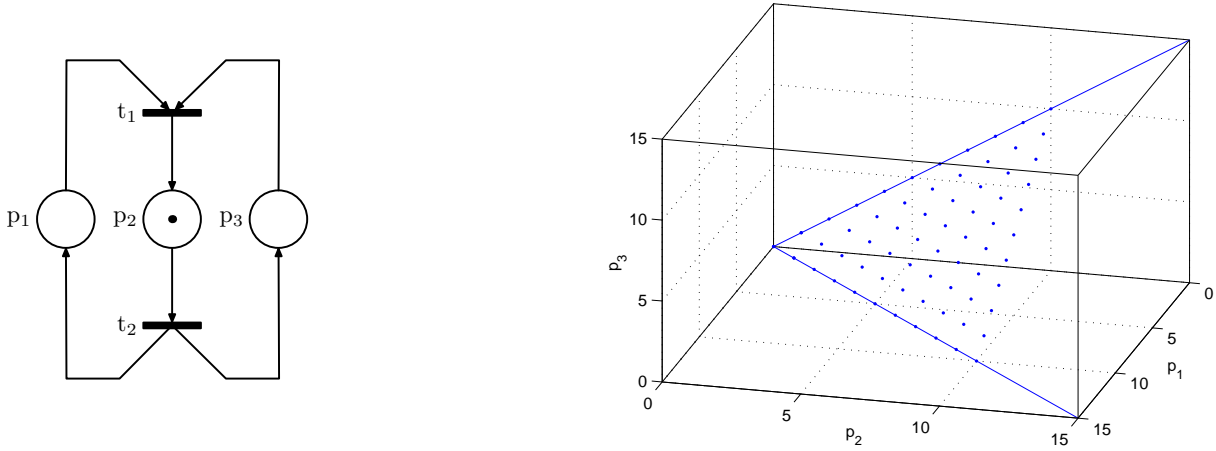
V následujících čtyřech bodech upustíme od požadavku na celočíselnost proměnných f_i . Potom lze říci, že řešení výše zmíněné soustavy nerovnic jsou konvexním prostorem. Pro úplnost připomeňme následující fakta:

- Pokud bychom připustili záporné hodnoty proměnných f_i , potom by se jednalo o soustavu rovnic a její řešení by tvořila afinní prostor procházející počátkem. Takový afinní prostor je dán vektory báze, nalezenými například Gausovou eliminační metodou v polynomiálním čase. Takovou charakteristiku Petriho sítě je sice snadné spočítat, ale je málo užitečná, jelikož pro záporné hodnoty v P-invariantě bychom ztratili její fyzikální význam (nešlo by o orientovaný sled míst).
- Připomeňme, že za *afinní* považujeme množinu \mathcal{X} právě tehdy, když v ní leží celá přímka spojující libovolné dva body této množiny; neboli množina $\mathcal{X} \subseteq \mathcal{R}^n$ je afinní pokud splňuje podmínku $x, y \in \mathcal{X}, \lambda \in \mathcal{R} \Rightarrow (\lambda x + (1 - \lambda)y) \in \mathcal{X}$.
- Připomeňme, že za *konvexní* považujeme množinu \mathcal{X} právě tehdy, když v ní leží celá úsečka spojující libovolné dva body této množiny; neboli množina $\mathcal{X} \subseteq \mathcal{R}^n$ je konvexní pokud splňuje podmínku $x, y \in \mathcal{X}, \lambda \in \mathcal{R}, 0 \leq \lambda \leq 1 \Rightarrow (\lambda x + (1 - \lambda)y) \in \mathcal{X}$.
- Připomeňme, že množina $\mathcal{X} \subseteq \mathcal{R}^n$ je *kužel* právě tehdy, když pro libovolný vektor $x \in \mathcal{X}$ a pro libovolný nezáporný skalár λ platí $\lambda x \in \mathcal{X}$.
- V případě množiny P-invariant jde o prostor, který se nazývá konvexní kužel (množina $\mathcal{X} \subseteq \mathcal{R}^n$ je *konvexní kužel* právě tehdy, když je konvexní a je kužel). Lze dokázat, že klasifikace "množina \mathcal{X} je konvexní kužel" je ekvivalentní klasifikaci "množina \mathcal{X} je uzavřená na nezáporné lineární kombinace" (neboli pro všechny vektory $x^1, \dots, x^k \in \mathcal{X}$

a pro všechny nezáporné skaláry $\lambda^1, \dots, \lambda^k$ je lineární kombinace $x' := \sum_{i=1}^k \lambda^i x^i$ také v \mathcal{X}).

Skutečnost, že P-invarianty dané vztahem 1.3 jsou celočíselné hodnoty výše zmíněného konvexního kuželu, ilustruje obrázek 1.7, kde jednotlivé body představují jednotlivé P-invarianty Petriho sítě (počátek představuje jakousi degenerovanou P-invariantu, kterou se nebudeme dále zabývat). Jelikož množina P-invariant může být pro danou Petriho síť nekonečná (pokud existuje jedna P-invarianta, pak jich existuje nekonečně mnoho), zajímá nás efektivní reprezentace konvexního kuželu, jež je dána jeho hranami (rays). P-invarianty, které jsou na dané hraně nejbližší počátku, se nazývají *minimální P-invarianty* nebo někdy též *generátory* nebo hrany konvexního kuželu.

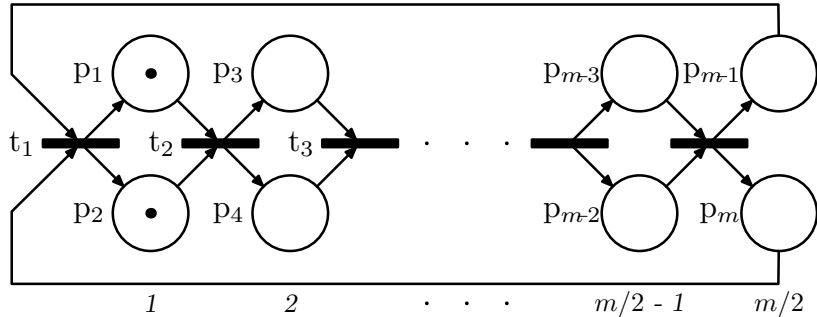
Pro Petriho síť na obrázku 1.7 je $\{(1, 1, 0)^T, (0, 1, 1)^T\}$ množinou minimálních P-invariant, ze které lze libovolnou P-invariantu vygenerovat jako nezápornou lineární kombinaci těchto dvou vektorů.



Obrázek 1.7: Petriho síť s odpovídajícími P-invarianty tvořícími konvexní kužel

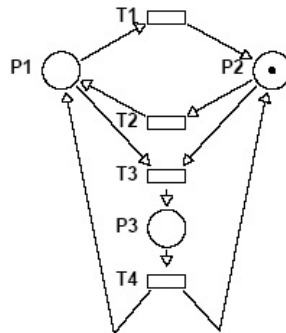
Množina minimálních P-invariant je velmi zajímavou charakteristikou struktury Petriho sítě a její interpretace v rámci modelovaného fyzikálního systému přináší rozbor struktury tohoto systému. Například pro Petriho síť na obrázku 1.6 je množina minimálních P-invariant $\{(1, 1, 0, 0, 0)^T, (0, 1, 1, 3, 0)^T, (0, 0, 0, 1, 1)^T\}$, kde P-invarianta $(1, 1, 0, 0, 0)^T$ reprezentuje předpis pro chování běžného uživatele, P-invarianta $(0, 1, 1, 3, 0)^T$ reprezentuje předpis pro chování sdíleného zdroje a P-invarianta $(0, 0, 0, 1, 1)^T$ reprezentuje předpis pro chování náročného uživatele. Naneštěstí počet minimálních P-invariant není shora omezen nějakou polynomiální funkcí velikosti Petriho sítě. Důkazem tohoto tvrzení je Petriho síť na obrázku 1.8 kde nalezneme $2^{m/2}$ různých minimálních P-invariant (každá dvojice míst může představovat jeden bit, kdy hodnota 1 znamená, že v P-invariantě se účastní horní místo z dvojice a 0 znamená, že v P-invariantě se účastní dolní místo z dvojice; množina minimálních P-invariant odpovídá množině všech možných kombinací bitů ve slově o délce $m/2$, kde m

je počet míst Petriho sítě). Výše zmíněné tvrzení lze také formulovat takto: *neexistuje polynomiální algoritmus pro nalezení množiny minimálních P-invariant, jelikož tato množina má pro některé Petriho sítě (například obrázek 1.8) exponenciální velikost*. Algoritmy na hledání množiny minimálních P-invariant lze nalézt v [2] a [5].



Obrázek 1.8: Petriho síť s exponenciálním počtem minimálních P-invariant

Užitečnost P-invariant pro analýzu vlastností ilustruje skutečnost, že existence alespoň jednoho tokenu v každé P-invariantě je nutnou podmínkou pro živost sítě bez izolovaných míst (izolované místo je samo o sobě konzervativní komponentou, ale jeho značení nemá vliv na žádný přechod). Neboli pokud je Petriho síť bez izolovaných míst živá, potom každá její P-invarianta obsahuje alespoň jeden token. Pro obecnou Petriho síť jde o podmínku nutnou, nikoli však postačující. Obrázek 1.10 ukazuje Petriho síť, jež je slabě souvislá, konzervativní komponenta $\{p_1, p_2, p_3, p_4\}$ obsahuje tokeny, ale síť není živá. Obrázek 1.9 ukazuje Petriho síť, jež je silně souvislá, konzervativní komponenta $\{p_1, p_2, p_3\}$ obsahuje token, ale síť není živá.



Obrázek 1.9: Silně souvislá PS jež není živá, přestože konzervativní komponenta odpovídající jediné minimální P-invariantě obsahuje token

Skutečnost, že studujeme strukturální vlastnosti, lze ilustrovat na obrázku 1.5g kde množina míst $\{p_{26}, p_{27}, p_{28}\}$ není konzervativní komponentou, jelikož ta musí platit pro libo-

volné počáteční značení. Pro počáteční značení na obrázku je sice součet tokenů zachován. Pokud však počáteční značení zdvojnásobíme, pak může být přechod t_{27} přeskočen a součet tokenů $m(p_{26}) + m(p_{27}) + m(p_{28})$ se zmenší ze dvou na jeden.

1.3.2.2 T-invarianty

Množina přechodů se nazývá *repetitivní komponenta* právě tehdy, když existuje přeskoková sekvence z těchto přechodů, která vede na původní značení Petriho sítě.

T-invarianta s je vektor celých nezáporných čísel charakterizující výše zmíněnou přeskokovou sekvenci. Neboli Petriho síť $(\mathcal{P}, \mathcal{T}, Pre, Post)$ má T-invariantu $s \Leftrightarrow s$ je charakteristickým vektorem přeskokové sekvence S takové, že $\forall m_0; m_0 \xrightarrow{S} m_0$. Jelikož charakteristický vektor udává počty přeskoků přechodů, tak platí: $s_j \in \mathcal{Z}_0^+ \forall j = 1 \dots \|\mathcal{T}\|$.

Při pohledu na stavovou rovnici $m = m_0 + W \cdot s$ snadno nahlédneme, že pokud má pro T-invariantu s platit $m = m_0$ pro libovolné m_0 , potom musí platit:

$$W \cdot s = 0 \quad s_i \in \mathcal{Z}_0^+ \quad (1.4)$$

Algoritmus pro nalezení množiny minimálních T-invariant je obdobný algoritmu pracujícím s P-invariantami, jelikož transpozicí incidenční matice W lze soustavu nerovnic 1.4 převést na soustavu nerovnic 1.3.

1.4 Podtřídy, zkrácené a rozšířené Petriho sítě

1.4.1 Podtřídy Petriho sítí

Podtřídy mají v porovnání s obecnými Petriho sítěmi, uvedenými v kapitole 1.1, redukovanou vyjadřovací schopnost. Neboli neumožňují modelovat některé struktury (například strukturální konflikt, spojení, paralelismus, synchronizaci), které existují v obecných Petriho sítích. Na druhou stranu, vlastnosti podtříd lze analyzovat jednoduššími postupy, než je tomu v případě obecných Petriho sítí.

1.4.1.1 Stavový graf

Stavový graf (state graph, state machine) je Petriho síť, ve které každý přechod má právě jednu vstupní a jednu výstupní hranu a váhy všech hran jsou rovny 1. Neboli Petriho síť je stavový graf $\Leftrightarrow |t_j| = |t_j^o| = 1 \forall t_j \in \mathcal{T}$ a $Pre(p_i, t_j) \in \{0, 1\}$ a $Post(p_i, t_j) \in \{0, 1\} \forall p_i \in \mathcal{P} \forall t_j \in \mathcal{T}$.

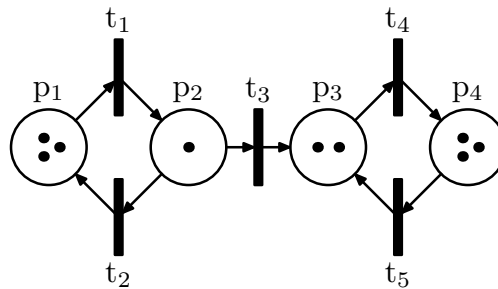
Následkem toho stavový graf může obsahovat strukturální konflikt, ale neobsahuje strukturální paralelismus, synchronizaci, zdrojový přechod a cílový přechod.

Z hlediska dalšího zpracování (například studia vlastností) stavového grafu si lze tuto Petriho síť představit jako orientovaný graf, ve kterém každý přechod s jeho vstupní a výstupní hranou nahradíme jednou orientovanou hranou a každé místo nahradíme vrcholem

grafu s uvedeným značením. Potom analýzu takového systému lze opřít o algoritmy pracující s orientovanými grafy.

Uveďme některé vlastnosti stavového grafu:

- Každý stavový graf je ohraničený.
- Každá uzavřená orientovaná cesta (neboli neopakuje se v ní vrchol a následkem toho ani hrana) odpovídá minimální T-invariantě.
- Stavový graf je živý \Leftrightarrow je silně souvislý (neboli existuje orientovaná cesta z libovolného vrcholu do libovolného jiného vrcholu) a v počátečním značení je alespoň jeden token. Například stavový graf na obrázku 1.10 není živý, protože není silně souvislý.



Obrázek 1.10: Příklad stavového grafu

Stavový automat (nebo graf dosažitelných značení Petriho sítě) se liší od stavového grafu tím, že ve stavovém automatu se jakoby nachází právě jeden token ve stavu, který je aktuální, zatímco ve stavovém grafu se může nacházet více tokenů najednou. Neboli stavový graf je podtřídou obecné Petriho sítě a stavový automat je jakoby podtřídou stavového grafu.

1.4.1.2 Značený graf

Značený graf (marked graph, event graph, graf událostí) je Petriho síť, kde každé místo má právě jednu vstupní a jednu výstupní hranu a váhy všech hran jsou rovny 1. Neboli Petriho síť je značený graf $\Leftrightarrow |p_i| = |p_i^o| = 1 \forall p_i \in \mathcal{P}$ a $Pre(p_i, t_j) \in \{0, 1\}$ a $Post(p_i, t_j) \in \{0, 1\} \forall p_i \in \mathcal{P} \forall t_j \in \mathcal{T}$.

Následkem toho značený graf může obsahovat strukturální paralelismus, ale neobsahuje strukturální konflikt, spojení, zdrojové místo a cílové místo.

Z hlediska dalšího zpracování (například studia vlastností) značeného grafu si lze tuto Petriho síť představit jako orientovaný graf, ve kterém každé místo s jeho vstupní a výstupní hranou nahradíme jednou orientovanou hranou s uvedeným značením a každý přechod nahradíme vrcholem grafu.

Uveďme některé vlastnosti značeného grafu:

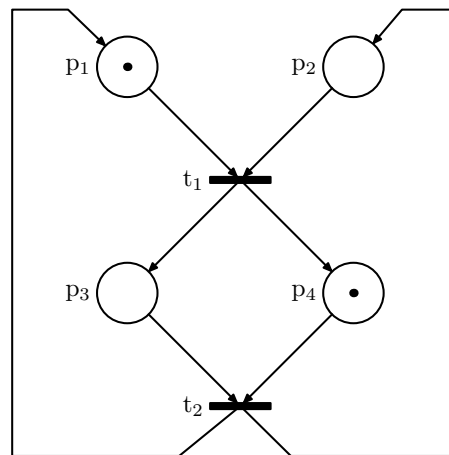
- Každá uzavřená orientovaná cesta (neboli neopakuje se v ní vrchol a následkem toho ani hrana) odpovídá minimální P-invariantě.

- Značený graf je živý \Leftrightarrow každá P-invarianta obsahuje alespoň jeden token. Povšimněme si, že narozdíl od obecné Petriho sítě jde v případě značeného grafu o podmínku nutnou i postačující.
- Z předchozí vlastnosti a skutečnosti, že každá P-invarianta je nezápornou lineární kombinací minimálních P-invariant vyplývá: Značený graf je živý \Leftrightarrow každá minimální P-invarianta obsahuje alespoň jeden token.

Ukážeme dva postupy jak dokázat živost značeného grafu (druhý zmiňovaný je evidentně lepší):

1. Nalezneme množinu minimálních P-invariant. Značený graf je živý právě tehdy, když každá z nich obsahuje alespoň jeden token. Tento postup má nepolynomiální složitost, jelikož existují příklady značeného grafu (obrázek 1.8), ve kterém je počet minimálních P-invariant exponenciálně závislý na počtu míst.
2. Ve značeném grafu odstraníme všechna místa, která obsahují alespoň jeden token. V takto redukováném grafu existuje uzavřená orientovaná cesta (neboli cyklus) právě tehdy, když značený graf není živý. Tento postup má polynomiální složitost, jelikož uzavřenou orientovanou cestu v grafu lze nalézt v čase $O(n^3)$ například Floydovým algoritmem (vstupem je adjugovaná matice s ∞ na digonále; vrchol účastnící se cyklu má na diagonále výsledné matice nejkratších cest hodnotu různou od ∞ právě tehdy, když existuje orientovaná cesta s konečným ohodnocením, která vede z tohoto vrcholu do něho samého, neboli existuje uzavřená orientovaná cesta procházející tímto vrcholem).

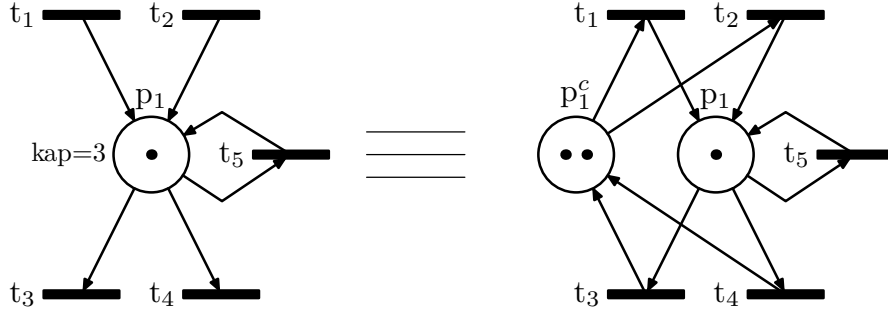
Obrázek 1.11 ilustruje příklad značeného grafu, který není živý. Důvodem je skutečnost, že P-invarianta $(0, 1, 1, 0)^T$ neobsahuje žádný token.



Obrázek 1.11: Deadlock ve značeném grafu

1.4.2 Zkrácené Petriho sítě

Cílem zkrácených Petriho sítí (někdy též zestručněných Petriho sítí) je zestručnit zápis modelu, čehož je dosaženo přidáním nějakého dalšího atributu. Tím je výsledný model stručnější, ale porozumění tomuto zápisu může být náročnější. Podstatné je, že každou zkrácenou Petriho síť lze rozepsat na obecnou Petriho síť.



Obrázek 1.12: Náhrada kapacitní Petriho sítě pomocí komplementárního místa

1.4.2.1 Kapacitní Petriho síť

U kapacitní Petriho sítě je místu p_i přiřazena kapacita $k(p_i)$ a přechod před ním není uvolněn v případě, že by se kapacita $k(p_i)$ měla přeskokem tohoto přechodu překročit. Neboli t_j je uvolněn pro značení m právě tehdy, když jsou splněny následující tři podmínky:

$$\forall p_i \in \mathcal{T}_j; m(p_i) \geq \text{Pre}(p_i, t_j) \text{ a}$$

$$\forall p_i \in \mathcal{T}_j^o \cap \mathcal{T}_j; m(p_i) + \text{Post}(p_i, t_j) - \text{Pre}(p_i, t_j) \leq k(p_i) \text{ a}$$

$$\forall p_i \in \mathcal{T}_j^o \setminus \mathcal{T}_j; m(p_i) + \text{Post}(p_i, t_j) \leq k(p_i).$$

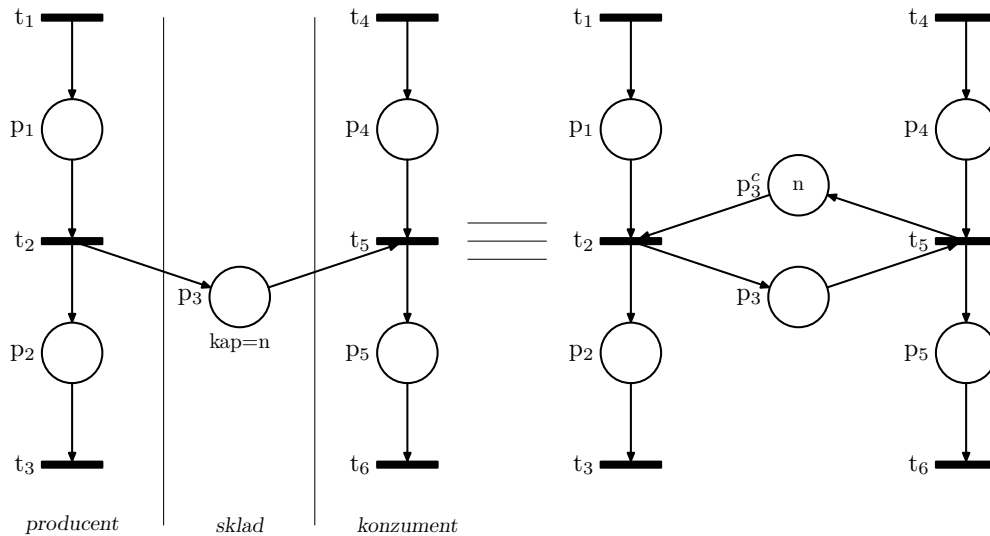
Obrázek 1.12 ukazuje v levé části kapacitní Petriho síť a v pravé části ekvivalentní obecnou Petriho síť s komplementárním místem p_1^c .

Komplementární místo p_i^c je spojeno s týmiž přechody jako p_i , ale hrany mají obrácenou orientaci, neboli $\text{Pre}(p_i, t_k) = \text{Post}(p_i^c, t_k)$ a $\text{Post}(p_i, t_l) = \text{Pre}(p_i^c, t_l)$. Místo p_i a jeho komplementární místo p_i^c dohromady tvoří P-invariantu. Počáteční značení komplementárního místa p_i^c je dáno počátečním značením místa p_i a jeho kapacitou neboli $m_0(p_i^c) = k(p_i) - m_0(p_i)$.

Orázek 1.13 uvádí příklad producenta a konzumenta, kteří jsou spojeni skladem o kapacitě n . Kapacitní Petriho síť na levé straně je ekvivalentní obecné Petriho síti uvedené na pravé straně, jelikož maximální počet tokenů v místě p_3 je dán počátečním značením vloženého komplementárního místa p_3^c . Tento a předchozí příklad ilustrují, jak lze každou kapacitní Petriho síť převést s použitím komplementárního místa na obecnou Petriho síť.

1.4.2.2 Barvené Petriho sítě

Základním účelem barvených Petriho sítí je podchytit a zestručnit části modelu, které se opakují (obdobně jako když zápisem ve *for* smyčce předejdeme opakovanému zápisu



Obrázek 1.13: Producent a konzument

podobných příkazů). Proto barvená Petriho síť přenáší informaci ze struktury sítě do očíslovaných tokenů a do funkcí přiřazených k hranám. Tím lze reprezentovat systémy značné (respektive neurčené) velikosti.

Všichni už jsme se setkali s podobnou abstrakcí, když jsme například pod nějakým operačním systémem spustili tři úlohy se stejným kódem. Model těchto tří úloh pomocí obecné Petriho sítě lze rozstříhat na tři části, ukazateli instrukce v každé z nich přiřadit jinou barvu a přeložit tyto tři strukturálně totožné části přes sebe. Pokud každá z úloh pracuje nezávisle na ostatních, potom funkce přiřazené hranám barvené Petriho sítě jsou vždy identita (nemění se barva tokenů). Pokud mezi sebou úlohy například komunikují, potom funkce přiřazené hranám mění barvy tokenů.

V barvené Petriho síti je přechod t_j uvolněn právě tehdy, když v jeho vstupních místech existují takové tokeny, že jejich transformace funkcemi přiřazenými vstupním hranám do t_j je změna na tokeny téže barvy. Potom může být přechod t_j touto barvou přeskočen.

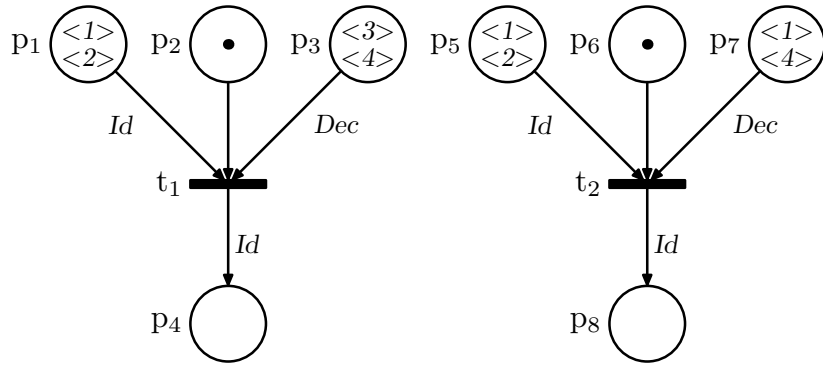
Obrázek 1.14 uvádí dva příklady značené Petriho sítě. Příklad nalevo ukazuje situaci, kdy přechod t_1 je uvolněn barvou 2 a po jeho přeskočení se objeví token o barvě 2 v místě p_4 . Druhý příklad ukazuje situaci, kdy přechod není uvolněn žádnou barvou.

Obrázek 1.15 ukazuje frontu (sklad) FIFO s kapacitou N , do které dva producenti dodávají data (výrobky) a ze které dva konzumenti data (výrobky) odebírají. Každý konzument odebírá pouze od svého producenta.

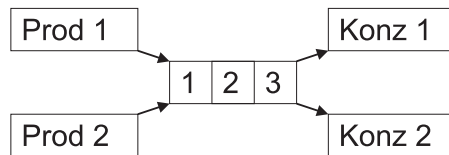
Obecná Petriho síť na obrázku 1.16 modeluje tento systém diskrétních událostí.

Zápis obecnou Petriho sítí lze zestručnit tak, že dané pozici ve frontě přidáme určitou barvu. Barvená Petriho síť na obrázku 1.17 využívá následující funkce přiřazené hranám:

- Id (identity) - nemění barvu tokenů
- Ex1 (exclusively 1) - akceptuje pouze token o barvě 1 a nemění jeho barvu



Obrázek 1.14: Uvolnění a přeskok přechodu v barvené Petriho síti



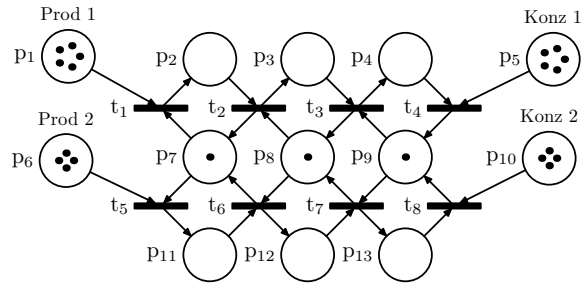
Obrázek 1.15: Producent-konzument s frontou FIFO (případ $N=3$)

- ExN (exclusively N) - akceptuje pouze token o barvě N a nemění jeho barvu
- Inc (increment) - zvýší barvu tokenu o 1
- Dec (decrement) - sníží barvu tokenu o 1

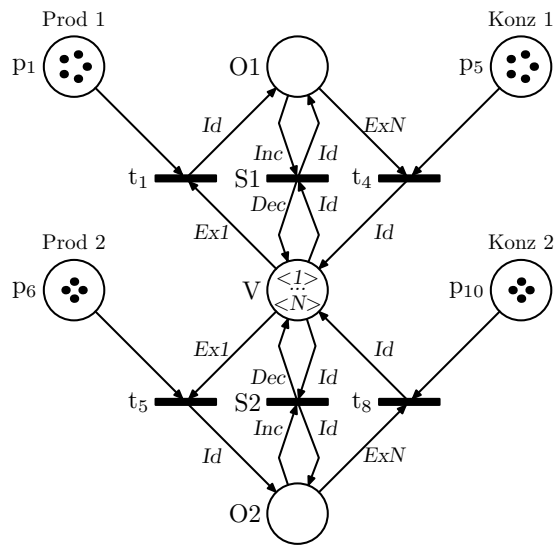
Místa reprezentující vstupy do fronty ze strany producentů (p_1, p_6) a požadavky na výstup ze strany konzumentů (p_5, p_{10}) obsahují nebarvené tokeny. Nebarvený token je přijímán přechodem bez ohledu na to jakou barvou je přeskokován. Místo V reprezentuje volné pozice ve frontě, místo $O1$ (respektive $O2$) reprezentuje pozice obsazené produktem 1 (respektive produktem 2). Vstupně-výstupní přechody jsou uvolňovány pouze vybranými barvenými tokeny (volná první pozice nebo obsazená N -tá pozice). Přechod $S1$ (respektive $S2$) reprezentuje posun produktu 1 (respektive produktu 2) na následující pozici ve frontě.

Klasický příklad z operačních systémů, večerící filosofové (*dinning philosophers*), ukazuje na obrázku 1.18 skupinu pěti filosofů, kde každý z nich potřebuje obě sousední vidličky V k tomu aby mohl jíst. Filosof se nachází ve dvou možných stavech: buď jí J nebo filosofuje F . Za tím účelem vidličky bere B a ukládá U .

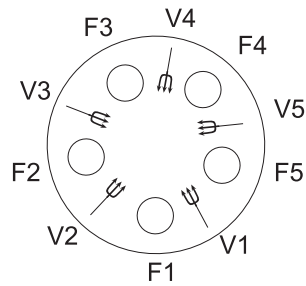
Obecné Petriho síti na obrázku 1.19 odpovídá incidenční matice:



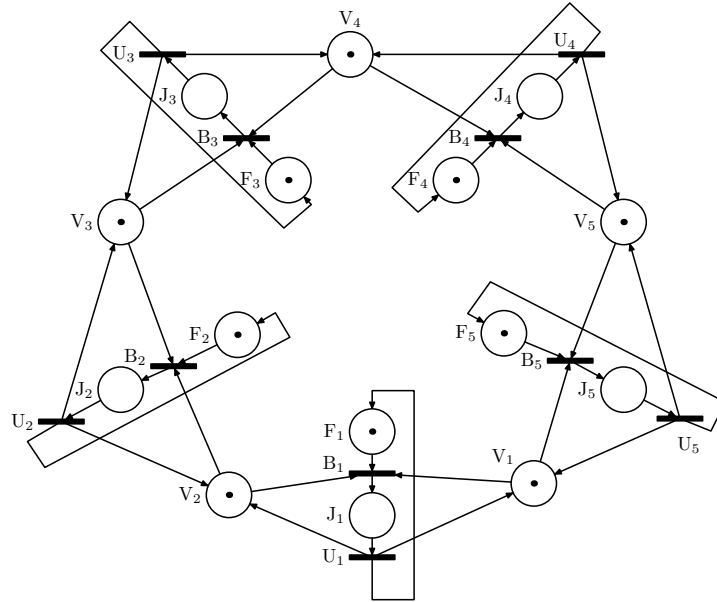
Obrázek 1.16: Fronta FIFO se třemi pozicemi modelovaná obecnou Petriho sítí



Obrázek 1.17: Fronta FIFO s neurčeným počtem pozic modelovaná barvenou Petriho sítí



Obrázek 1.18: Pět večerících filosofů

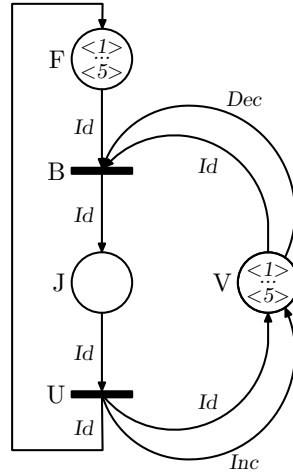


Obrázek 1.19: Obecná Petriho síť pro pět večerících filosofů

$$W = \begin{matrix} & B_1 & B_2 & B_3 & B_4 & B_5 & U_1 & U_2 & U_3 & U_4 & U_5 \\ \begin{matrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{matrix} & \left[\begin{array}{cccccc} -1 & & & & & 1 \\ & -1 & & \emptyset & & & 1 & & \emptyset \\ & & -1 & & & & & & 1 \\ & \emptyset & & -1 & & & \emptyset & & 1 \\ & & & & -1 & & & & & 1 \\ 1 & & & & & -1 & & & & & \\ & 1 & & \emptyset & & & -1 & & \emptyset \\ & & 1 & & & & & & -1 \\ & \emptyset & & 1 & & & \emptyset & & -1 \\ & & & & 1 & & & & & & -1 \\ -1 & & & & -1 & 1 & & & & & 1 \\ -1 & -1 & & \emptyset & & 1 & 1 & & \emptyset \\ & -1 & -1 & & & & 1 & 1 & & & \\ & \emptyset & -1 & -1 & & & \emptyset & 1 & 1 & & \\ & & & -1 & -1 & & & & 1 & 1 \end{array} \right] \end{matrix}$$

Pomocí barvené Petriho sítě lze redukovat počet míst-přechodů a tokenům přiřadit barvy. Incidenční matici lze rozdělit do submatic. Pravidelné submatice je potřeba popsat vhodnými funkcemi přiřazenými k hranám barvené Petriho sítě. V obrázku 1.20 se vedle funkce *id* vyskytují funkce *inc* a *dec*, jež zvýší-sníží barvu tokenů o 1 modulo 5. Tímto zavedením vzniká incidenční matice jako matice funkcí:

$$\begin{array}{c}
 F \\
 J \\
 V
 \end{array}
 \begin{bmatrix}
 & B & U \\
 -id & & id \\
 id & & -id \\
 -(id \wedge dec) & & -(id \wedge inc)
 \end{bmatrix}$$



Obrázek 1.20: Barvená Petriho síť pro pět večeřících filosofů

1.4.3 Rozšířené Petriho síť

V případě rozšířených Petriho sítí jsou k obecné Petriho síti, definované v kapitole 1.1, přidána nějaká pravidla rozšiřující její modelovací schopnosti. Všeobecně vzato nelze rozšířenou Petriho síť rozpdat na obecnou Petriho síť.

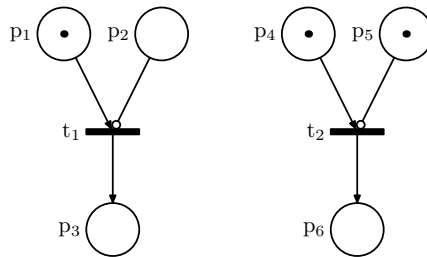
Je zřejmé, že pro mnoho systémů diskretních událostí není obecná Petriho síť postačující pro zachycení řady podstatných rysů systému. Proto rozšiřujeme model obecné Petriho sítě o další atributy umožňující například řešit konflikty pomocí priorit (prioritní Petriho síť) nebo testovat prázdnotu místa (Petriho síť s inhibitovanou hranou). Mezi rozšířené Petriho síť lze zařadit i skupinu neautonomních Petriho sítí (viz kapitola ??) rozšiřující obecnou Petriho síť o čas (časované, stochastické a časové Petriho síť) nebo o synchronizaci s vnějšími událostmi (synchronizované Petriho síť). V této kapitole se však budeme zabývat pouze takovými rozšířeními, která zachovávají autonomnost Petriho sítě.

1.4.3.1 Petriho síť s inhibitovanou hranou

Petriho síť s inhibitovanou hranou umožňuje jednoduše modelovat *test na nulu*. Inhibitovaná hrana je speciální typ orientované hrany, která vychází z místa p_i a vstupuje do přechodu t_j . Konec inhibitované hrany je označen malým kroužkem, jak je vidět na obrázku 1.21. Prázdnotu místa p_i je nezbytnou podmínkou pro uvolnění přechodu t_j . Neboli přechod t_1 na

obrázku 1.21 uvolněn je a jeho přeskokem se odebere jeden token z místa p_1 , vloží se jeden token do místa p_3 a značení místa p_2 se nezmění. Naproti tomu přechod t_2 na obrázku 1.21 uvolněn není, jelikož místo p_5 není prázdné.

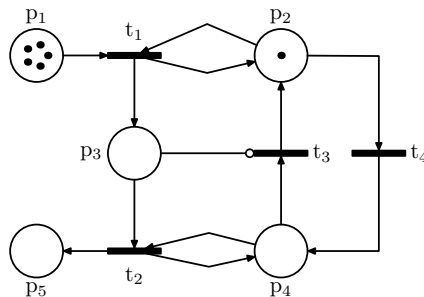
Z formálního hlediska je potřeba rozšířit definiční obor matice Pre o speciální prvek indikující skutečnost, že odpovídající hrana je inhibitovaná, neboli Pre je matice zobrazení $\mathcal{P} \times \mathcal{T} \rightarrow \mathbb{Z}_0^+ \cup \{\emptyset\}$. Přechod t_j je uvolněn $\Leftrightarrow \forall p_i \in \mathcal{P}; M(p_i) \geq Pre(i, j)$ pro $Pre(i, j) \in \mathbb{Z}_0^+$ a $M(p_i) = 0$ pro $Pre(i, j) = \emptyset$



Obrázek 1.21: Inhibitovaná hrana

Příklad na obrázku 1.22 ilustruje systém hromadné obsluhy. Představme si například, že jde o úřadovnu se řízenými vstupními dveřmi. Úředník otevře vstupní dveře a vpustí do úřadovny všechny zákazníky, kteří touží po odbavení. Poté vstupní dveře zavře a postupně odbavuje jednotlivé zákazníky. Zákazníci po odbavení odcházejí výstupními dveřmi, jež jsou stále otevřeny. Úředník vstupní dveře otevře až po odbavení všech zákazníků.

Na obrázku 1.22 místo p_1 reprezentuje zákazníky před příchodem do úřadovny, místo p_3 zákazníky v úřadovně a místo p_5 zákazníky po odchodu z úřadovny. Místo p_2 obsahuje token, když jsou vstupní dveře otevřeny a místo p_4 obsahuje token, když jsou vstupní dveře zavřeny. Přeskok přechodu t_1 reprezentuje vstup zákazníka, t_2 reprezentuje odchod zákazníka, t_3 reprezentuje otevření vstupních dveří a t_4 reprezentuje zavření vstupních dveří.

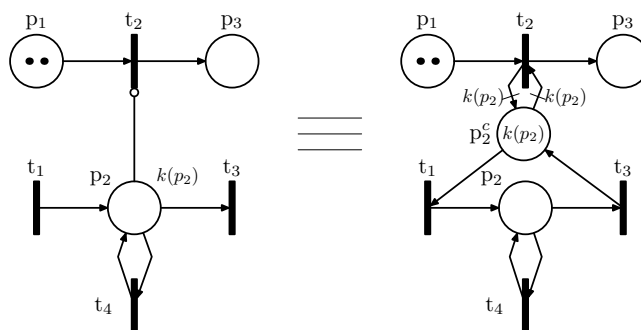


Obrázek 1.22: Model systému hromadné obsluhy pomocí Petriho sítě s inhibitovanou hranou

Z definice Petriho sítě s inhibitovanou hranou je zřejmé, že ji nelze tak snadno formalizovat za pomoci lineární algebry, jak to bylo možné u obecné Petriho sítě. Naneštěstí nelze

každou Petriho síť s inhibitovanou hranou převést na obecnou Petriho síť. Na druhou stranu platí následující tvrzení: pokud je Petriho síť s inhibitovanou hranou ohraničená, potom ji lze převést na obecnou Petriho síť.

Převod ohraničené Petriho sítě s inhibitovanou hranou vycházející z místa p_i a vstupující do přechodu t_j lze snadno realizovat pomocí nově založeného komplementárního místa p_i^c (viz. jeho definice v odstavci 1.4.2.1). Počáteční značení komplementárního místa p_i^c je dáno počátečním značením místa p_i a jeho ohraničeností neboli $m_0(p_i^c) = k(p_i) - m_0(p_i)$ kde $k(p_i)$ je maximální počet tokenů v místě p_i pro libovolné dosažitelné značení Petriho sítě. Jelikož předpokládáme, že je místo p_i ohraničené, lze inhibitovanou hranu z místa p_i do přechodu t_j nahradit dvojicí hran typu "self-loop" mezi komplementárním místem p_i^c a t_j . Neboli $Post(p_i^c, t_j) = Pre(p_i^c, t_j) = k(p_i)$. Tuto transformaci ilustruje obrázek 1.23 s komplementárním místem p_2^c .



Obrázek 1.23: Náhrada inhibitované hrany v ohraničené Petriho síti

1.5 Poznámky a reference

Řadu užitečných informací o Petriho sítích lze nalézt na stránce "Petri Nets World", jež se momentálně nachází na <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>. Jako úvodní text je velmi oblíbený článek, který napsal Tadao Murata [6]. Pro začátečníka srozumitelný materiál lze nalézt v úvodních kapitolách knihy, kterou napsali René David a Hasane Alla [3]. Širší pohled na problematiku systémů diskrétních událostí zahrnující automaty, teorii front, ale i pohled z úhlu klasické teorie systémů je zpracován v [1]. Kvalitní je výukový materiál používaný při ICATPN konferenci [4].

1.6 Závěr

V této kapitole jsme představili grafický a matematický nástroj zvaný Petriho síť, umožňující modelovat systémy, jejichž stavový prostor je diskrétní a rovněž události, vedoucí na změnu stavu, mají diskrétní povahu. Popis pomocí Petriho sítě je ekvivalentní popisu pomocí stavového automatu, jelikož graf dosažitelných značení Petriho sítě je totožný

se stavovým automatem. Je-li Petriho síť ohraničená, pak ji lze realizovat konečným automatem. Díky tokenům, paralelismu, synchronizaci a konfliktům umožňuje Petriho síť popsat jednoduchým způsobem systém s rozsáhlým stavovým prostorem. Modulární struktura popisu (navzájem propojené či synchronizované moduly, kde každý popisuje nějakou část systému) umožňuje kopírovat topologii modelovaného systému a tím zachovat srozumitelnost modelu. V některých případech (například analýza živosti v grafu událostí) je možné provádět analýzu Petriho sítě bez vyčíslení stavového prostoru, což je velmi výhodné z hlediska časové náročnosti odpovídajících algoritmů.

Poděkování.

K modelování systémů diskrétních událostí pomocí Petriho sítí mne přivedl Robert Valette z LAAS CNRS Toulouse. K důslednosti v používání tohoto aparátu mne posléze přiměl René David, s nímž jsem měl tu čest sdílet kancelář v LAG INPG Grenoble. Rigoróznost a laskavost obou těchto učitelů lze jen stěží napodobit.

Děkuji panu Antonínu Karolíkovi za to, že v rámci své semestrální práce vytvořil užitečný nástroj na editaci obrázků v tomto textu. Dále děkuji panu Petru Jančarovi z VŠB Ostrava a řadě doktorandů a studentů z našeho oddělení za velmi užitečné komentáře k tomuto textu.

Seznam obrázků

1.1	Sdílení zdrojů	7
1.2	Graf dosažitelných značení Petriho sítě na obrázku 1.1	8
1.3	Ohraničená, binární a neohraničená Petriho síť	11
1.4	Živost Petriho sítě	11
1.5	Příklady demonstrující různé kombinace živosti, ohraničenosti a reverzibility	13
1.6	Petriho síť se třemi minimálními P-invariantami	14
1.7	Petriho síť s odpovídajícími P-invariantami tvořícími konvexní kužel	15
1.8	Petriho síť s exponenciálním počtem minimálních P-invariant	16
1.9	Silně souvislá PS jež není živá, přestože konzervativní komponenta odpovídající jediné minimální P-invariantě obsahuje token	16
1.10	Příklad stavového grafu	18
1.11	Deadlock ve značeném grafu	19
1.12	Náhrada kapacitní Petriho sítě pomocí komplementárního místa	20
1.13	Producent a konzument	21
1.14	Uvolnění a přeskok přechodu v barvené Petriho síti	22
1.15	Producent-konzument s frontou FIFO (případ $N=3$)	22
1.16	Fronta FIFO se třemi pozicemi modelovaná obecnou Petriho sítí	23
1.17	Fronta FIFO s neurčeným počtem pozic modelovaná barvenou Petriho sítí	23
1.18	Pět večeřících filosofů	23
1.19	Obecná Petriho síť pro pět večeřících filosofů	24
1.20	Barvená Petriho síť pro pět večeřících filosofů	25
1.21	Inhibitovaná hrana	26
1.22	Model systému hromadné obsluhy pomocí Petriho sítě s inhibitovanou hranou	26
1.23	Náhrada inhibitované hrany v ohraničené Petriho síti	27

Literatura

- [1] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, September 1999.
- [2] J.M. Colom and M.Silva. Convex geometry and semiflows in p/t nets - a comparative study of algorithms for computation of minimal p-semiflows. In *Advances in Petri Nets 1990, LNCS 483, Springer Verlag*, pages 79–112, 1990.
- [3] R. David and H. Alla. *Discrete, Continuous, and Hybrid Petri Nets*. Springer, 2004.
- [4] J. Desel, G. Balbo, K. Jensen, W. Reisig, G. Rozenberg, and M. Silva. Introductory tutorial on petri nets. In *21st International Conference on Application and Theory of Petri Nets (Aarhus, Denmark)*, 2000.
- [5] J. Martinez and M. Silva. A simple and fast algorithm to obtain all invariants of a generalised petri net. In *in: C. Girault, W. Reisig (eds): Application and Theory of Petri Nets, Informatik Fachberichte No.52, Springer*, pages 301–310, 1982.
- [6] T. Murata. Petri nets: Properties, analysis and applications. *Proc. IEEE, Vol. 77, No. 4*, pages 541–580, 1989.
- [7] C.A. Petri. *Kommunikation mit Automaten*. Bonn:Institut fur Instrumentelle Mathematik, Schiften des IIM Nr.3, 1962.