

Petriho sítě I

Obsah

1. Úvod - neformální výklad

- 1.1. Co jsou Petriho sítě?
- 1.2. C/E Petriho sítě (Condition/Event PN)
- 1.3. P/T Petriho sítě (Place/Transitions PN)
- 1.4. Petriho sítě s inhibičními hranami (PN with inhibitors)
- 1.5. Petriho sítě s prioritami (PN with priorities)
- 1.6. Časované Petriho sítě (Timed PN)
- 1.7. Barvené Petriho sítě (Coloured PN)
- 1.8. Hierarchické Petriho sítě (Hierarchical PN)
- 1.9. Objektové Petriho sítě (Object-Oriented PN)

2. P/T sítě a jejich vlastnosti

- 2.1. Struktura a dynamika P/T sítí
- 2.2. Vlastnosti a stavová analýza P/T sítí
- 2.3. Strukturní analýza P/T sítí
- 2.4. Podtřídy P/T Petriho sítí
- 2.5. Hierarchická syntéza Petriho sítí

3. Petriho sítě a jazyky

- 3.1. Jazyky Petriho sítí
- 3.2. Jazyky PN a Chomského hierarchie
- 3.3. Jazyky řízené Petriho sítěmi

4. Časované Petriho sítě

- 4.1. Petriho sítě s časovanými přechody
- 4.2. Stochastické Petriho sítě (SPN)
- 4.3. Petriho sítě s prioritami
- 4.4. Zobecněné stochastické Petriho sítě (GSPN)

Přílohy:

- A1. Multimnožiny
- A2. Markovovy procesy
- A3. Problém hromadné obsluhy řešený pomocí GSPN
- A4. Algoritmy analýzy PN

Literatura

- [1] J.Peterson: Petri Net Theory and the Modelling of Systems. Prentice Hall, 1981
 - [2] M.Češka: Petriho sítě /Úvod do teorie a nástrojů aplikací Petriho sítí/. CERM, Brno, 1994.
 - [3] V.E.Kotov: Seti Petri. Nauka, Moskva, 1984.
 - [4] R.David, H.Alla: Petri Nets and Grafcet /Tools for modelling discrete event systems/. Prentice Hall Ltd, 1992
 - [5] M.A.Marsan, G.Balbo,G.Conte, S.Donatelli, G.Franceschinis: Modelling with Generalized Stochastic Petri Nets. John Wiley & Sons, 1995.
 - [6] W.Reisig: A Primer in Petri Net Design. Springer-Verlag, 1992.
 - [7] Karásek P.,Skácel M.: Stochastic Petri Nets and their Modifications - Perspective Tool for Modelling. Proceedings of Conference MOSIS '94, Ostrava, 1994.
 - [8] Janoušek V.: OOPN:A High Level Language for Modeling and Simulation. Proceedings of Conference MOSIS '94, Ostrava, 1994.
 - [9] Brabec M.: Language Representation of Petri Net. Proceedings of Conference MOSIS '94, Ostrava, 1994.
 - [10]H.J.Genrich: Predicate/Transition Nets. In: K.Jensen, G.Rozenberg (Eds.): High-level Petri Nets /Theory and Application/, Springer- Verlag, 1991.
 - [11]K.Jensen: Coloured Petri Nets: A High Level Language for System Design and Analysis. In: K.Jensen, G.Rozenberg (Eds.): High-level Petri Nets /Theory and Application/, Springer-Verlag, 1991.
 - [12]J.Billington: Many-sorted High-level Nets. In: K.Jensen, G.Rozenberg (Eds.): High-level Petri Nets /Theory and Application/, Springer-Verlag, 1991.
 - [13]W.Reisig: Petri Nets and Algebraic Specifications. In: K.Jensen, G.Rozenberg (Eds.): High-level Petri Nets /Theory and Application/, Springer-Verlag, 1991.
 - [14]B.Krämer, H.W.Schmidt: Types and Modules for Net Specifications. In: K.Jensen, G.Rozenberg (Eds.): High-level Petri Nets /Theory and Application/, Springer-Verlag, 1991.
 - [15]P.Huber, K.Jensen, R.M.Shapiro: Hierarchies in Coloured Petri Nets. In: K.Jensen, G.Rozenberg (Eds.): High-level Petri Nets /Theory and Application/, Springer-Verlag, 1991.
 - [16] Markl J.: Modelování procesů hierarchickými Petriho sítěmi. Proceedings of Conference MOSIS '94, Ostrava, 1994.
 - [17] Christensen S., Petrucci L.: Towards a Modular Analysis of Coloured Petri Nets. In: K.Jensen (Ed.): Application and Theory of Petri Nets 1992. Springer-Verlag, 1992.
 - [18] R.David, H.Alla: Petri Nets and Grafcet /Tools for modelling discrete event systems/. Prentice Hall Ltd., 1992
 - [19] W.Resig-G.Rozenberg (Eds.): Lectures on Petri Nets I: Basic Models, LNCS 149, Springer, 1998.
 - [20] W.Resig-G.Rozenberg (Eds.): Lectures on Petri Nets II: Applications, LNCS 1492, Springer, 1998.
 - [21] M.A.Marsan, G.Balbo, G.Conte, S.Donatelli, G.Franceschinis: Modelling with Generalised Stochastic Petri Nets. John Wiley & Sons, 1995.
 - [22] E.Brinksma, H.Hermanns, J.-P. Katoen (Eds.): Lectures on Formal Methods and Performance Analysis, LNCS 2090, Springer 2001
 - [23] C.Girault, R.Valk: Petri Nets for Systems Engineering (A Guide to Modeling, Verification, and Applications), Springer-Verlag Berlin Heidelberg 2003
 - [24] <http://www.daimi.au.dk/PetriNets/>
(výchozí webová stránka k Petriho sítím)
 - [25] H.Ehrig, G.Juhás, J.Padberg, G.Rozenberg (Eds.): Unifying Petri Nets (Advances in Petri Nets), LNCS 2128, Springer-Verlag Berlin Heidelberg 2001
 - [26] L.Priese, H.Wimmel: Theoretische Informatik Petri-Netze, Springer-Verlag Berlin Heidelberg 2003
- [] ...

1. Úvod - neformální výklad

1.1 Co jsou Petriho sítě?

Petri Nets is a formal and graphical appealing language which is appropriate for modelling systems with concurrency. Petri Nets has been under development since the beginning of the 60'ies, where C.A.Petri defined the language. It was the first time a general theory for discrete parallel systems was formulated. The language is a generalization of automata theory such that the concept of concurrently occurring events can be expressed.

Petri Nets has nothing to do with network communication. However, Petri Nets have proven useful for describing protocols typically used in networks.

A. Petri was the first to formally define the Petri Nets language. He did so with his PhD thesis (Kommunikation mit Automaten) in the beginning of the 60'ies.

History of Petri Nets

Petri nets were originally developed in the 60'ies and the 70'ies, and they were soon recognised as being one of the most adequate and sound languages for description and analysis of synchronisation, communication and resource sharing between concurrent processes. However, attempts to use Petri nets in practice revealed two serious drawbacks. First of all, there were no data concepts and hence the models often became excessively large, because all data manipulation had to be represented directly into the net structure (i.e., by means of places and transitions). Secondly, there were no hierarchy concepts, and thus it was not possible to build a large model via a set of separate submodels with well-defined interfaces.

The development of high-level Petri nets in the late 70'ies and hierarchical Petri nets in the late 80'ies removed these two serious problems. Coloured Petri Nets (also called CP-nets or CPN) is one of the two most well-known dialects of high-level Petri nets. CP-nets incorporate both data structuring and hierarchical decomposition - without compromising the qualities of the original Petri nets.

Pojem Petriho sítě byl postupně obohacován a zobecnován tak, aby jeho modelovací schopnost vyhověla praktickým potřebám. V této úvodní kapitole neformálním způsobem probereme několik typů Petriho sítí a na příkladech vysvětlíme jak fungují a k čemu mohou sloužit. Postupovat budeme historicky od dřívějšího k pozdějšímu, od jednoduchého k složitějšímu, od speciálního k obecnějšímu. Postupně projdeme tyto typy Petriho sítí:

- C/E (Condition/Event) Petriho sítě,
- P/T (Place/Transitions) Petriho sítě,
- P/T Petriho sítě s inhibičními hranami,
- P/T Petriho sítě s prioritami,
- TPN Časované (Timed)Petriho sítě,
- CPN Barevné (Coloured, barvené) Petriho sítě,
- HPN Hierarchické (Hierarchical) Petriho sítě,
- OOPN Objektové (Object Oriented) Petriho sítě.

1.2. C/E Petriho síť (Condition/Event Petri Nets)

C/E Petriho síť je zadána následujícími údaji (a na základě těchto údajů může být také graficky zobrazena):

- *podmínkami* (conditions) zobrazovanými kroužky,
- *událostmi* (events) zobrazovanými obdélníky (případně úsečkami),
- šípkami vedoucími od podmínek k událostem,
- šípkami vedoucími od událostí k podmínkám,
- *tokens* (zobrazeny tečkami v kroužcích podmínek) indikujícími logický stav (pravdivost) podmínek, jejich počáteční rozložení v síti nazýváme *počátečním stavem* nebo *počátečním značením* (initial marking) síť.

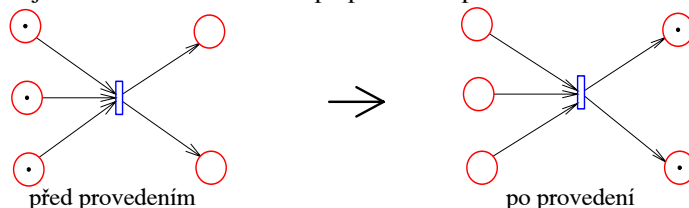
V C/E Petriho síti:

- podmínka *c* je *vstupní podmínkou* (precondition) události *e*, jestliže od podmínky *c* vede šípka k události *e*,
- podmínka *c* je *výstupní podmínkou* (postcondition) události *e*, jestliže od události *e* vede šípka podmínce *c*,
- každá podmínka je vždy buď *splněna* nebo *nesplněna*,
- každá splněná podmínka je indikována tokenem, tj. tečkou uvnitř kroužku zobrazujícího podmínku,
- celkový (globální) *stav síť* je zadán množinou podmínek, které jsou v daném okamžiku splněny,

Změny stavů C/E Petriho síti probíhají podle následujících pravidel:

- ke změnám stavu síti dochází uskutečňováním událostí,
- **enabling rule**: událost může nastat, jsou-li všechny její vstupní podmínky splněny a současně všechny její výstupní podmínky nesplněny - takovou událost nazýváme *proveditelnou*,
- provedena může být pouze proveditelná událost, proveditelná událost může také zůstat neprovedena,
- **firing rule**: po *provedení* proveditelné události jsou všechny její výstupní podmínky splněny a všechny její vstupní podmínky nesplněny.

Na obr.1.2.1 je zobrazena změna stavu po provedení proveditelné události



Obr.1.2.1

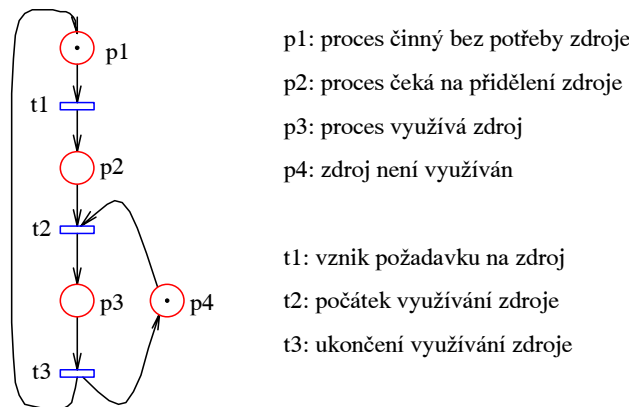
Poznámky 1.2.1:

1. Speciálním případem C/E Petriho síti je částečný *konečný automat*. Je to síť vyznačující se tím, že každá událost má právě jednu vstupní podmínku a právě jednu výstupní podmínku. V síti se pak pohybuje jediný token, který označuje podmínku definující aktuální stav automatu.
2. Speciálním případem C/E Petriho síti jsou tzv. *čisté síť* (pure nets), tj. C/E Petriho síť, ve kterých nejsou přípustné elementární smyčky tvořené podmínkou *c*, událostí *e* a hranami (c,e) , (e,c) . Podmínka *c* je současně vstupní i výstupní podmínkou události *e* a provedení události *e* nemění stav podmínky *c*. Hran (c,e) , (e,c) elementární smyčky nazýváme také *testovacími hranami*. Pomocí těchto hran událost *e* pouze testuje podmínku *c*, aniž by měnila její platnost. Při grafickém zobrazení dvojici těchto hran často zobrazujeme jedinou úsečkou s šípkami na obou koncích.
3. Speciálním případem C/E Petriho síti jsou tzv. *elementární síť* (elementary nets), což jsou C/E Petriho síť, pro které platí současně:
 - každá událost má alespoň jednu vstupní podmínku (událost, která by tuto vlastnost neměla, by umožňovala vstup libovolného počtu tokenů do síti),

- každá událost má alespoň jednu výstupní podmínku (událost, která by tuto vlastnost neměla, by umožňovala výstup libovolného počtu tokenů ze sítě),
 - žádná událost nemá žádnou podmínku, která by pro ni byla současně vstupní i výstupní (tj. síť je čistá).
4. Je možná dvojí duální interpretace podmínek a událostí - statická a dynamická. Zpravidla interpretujeme podmínku jako statický stav a událost jako dynamický děj. Je ale možná i opačná interpretace: podmínka reprezentuje děj, který probíhá, pokud podmínka platí a události představují bodové okamžiky, ve kterých se mění platnost podmínek.

Příklad 1.2.1:

Na obr.1.2.2 je pomocí C/E Petriho sítě zobrazen cyklický proces, který čas od času potřebuje využívat nějaký zdroj. Může se jednat např. o výpočetní proces v počítači, který čas od času tiskne výsledky na tiskárně.

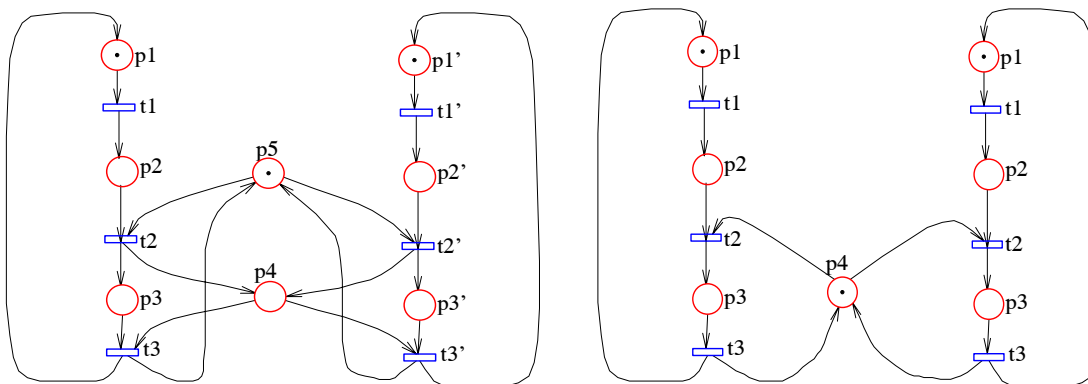


Obr.1.2.2 Využívání zdroje jedním procesem

Příklad 1.2.2:

Na obr.1.2.3 jsou pomocí C/E Petriho sítě reprezentovány dva nekonečné cyklické procesy, které čas od času potřebují využívat nějaký společný zdroj. Může se jednat např. o dva nezávislé výpočetní procesy ve dvou počítačích, které sdílejí společnou tiskárnu.

Obr.1.2.4 zobrazuje tutéž situaci, ale poněkud úsporněji: místo dvou míst (podmínek) p4,p5 pro zobrazení stavu využívání zdroje, je použito pouze místo jediné p4.



Obr.1.2.3 - 4 Využívání sdíleného zdroje dvěma procesy

Příklad 1.2.3:

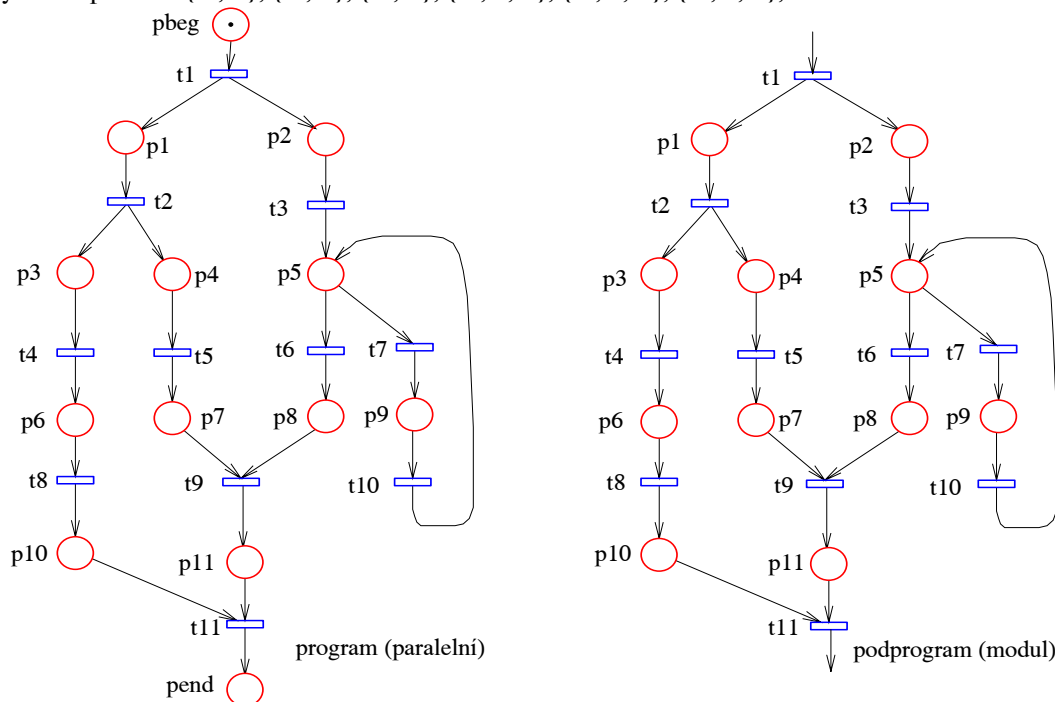
Na obr.1.2.5 je pomocí C/E Petriho sítě reprezentován paralelní program (řídící toky paralelního programu). Zobrazené síť lze interpretovat dvojím způsobem:

1. Obdélníky (události) zobrazují programové příkazy a kroužky (podmínky) statické stavy programu po skončení jednoho příkazu a před zahájením dalšího příkazu. Tokeny v kroužcích indikují stavy kterými prochází výpočetní proces v jednotlivých programových vláknech paralelního programu.
2. Obdélníky zobrazují události zahájování a ukončování jednotlivých příkazů a kroužky zobrazují dynamické stavy provádění jednotlivých programových příkazů. Tokeny v kroužcích indikují, které příkazy jsou právě prováděny.

Pokud nebude řečeno jinak, budeme pracovat s prvou interpretací. Paralelní program obsahuje tři programová vlákna tvořená následujícími posloupnostmi příkazů:

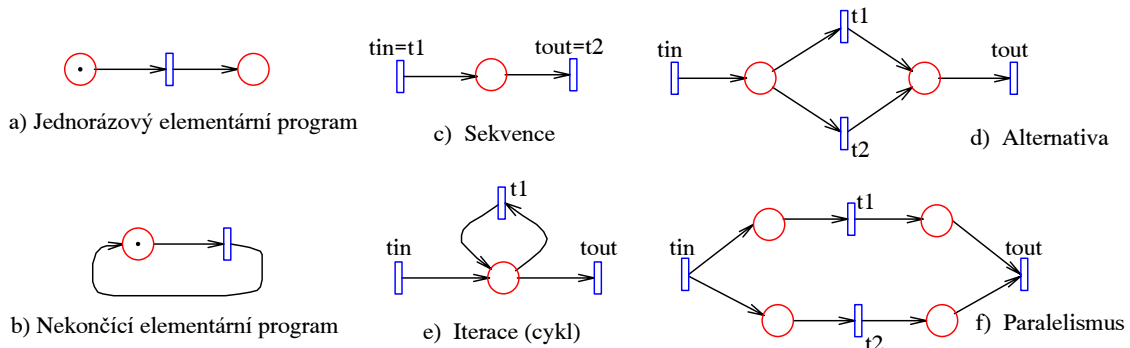
- t1, t2, t4, t8, t11
- t1, t2, t5, t9, t11
- t1, t3, (t7, t10)*, t6, t2, t9, t11

Některé příkazy programu mohou být prováděny současně. jedná se např. o následující podmnožiny množiny všech příkazů: {t2,t3}, {t2,t6}, {t2,t7}, {t3,t4,t5}, {t4,t5,t6}, {t4,t5,t7},...



Obr.1.2.5 Paralelní program

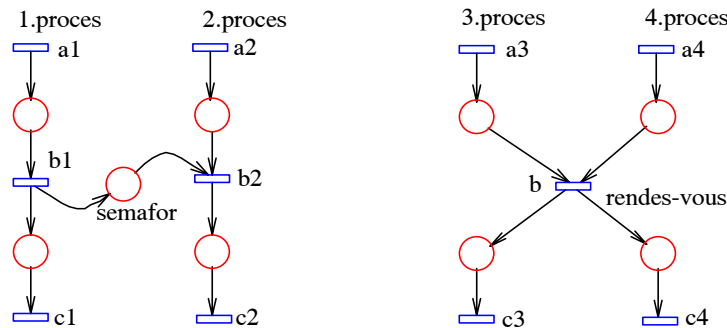
Na obr. 1.2.6 jsou pomocí Petriho sítě zobrazeny dva základní typy programů - obr a), b) a čtyři základní programové konstrukty řízení výpočtu - obr. c), d), e), f). Jediná událost sítě na obr. a) nebo b) představuje program na nejvyšší abstrakční úrovni. Postupným rozvíjením (zpodrobňováním) této události konstrukcemi podle obr. c), d), e), f) lze vytvořit (a Petriho síť zobrazit) libovolný **dobře strukturovaný program** (podle Duijkstry) a to **sekvenční** (nebyla-li použita konstrukce f)) nebo **paralelní** (byla-li použita konstrukce f)). Poznamenejme, že program z obr. 1. 2.5 není dobře strukturován ve smyslu Duijkstrových zásad strukturovaného programování.



Obr.1.2.6 Elementární programy a základní programové konstrukty

Příklad 1.2.4:

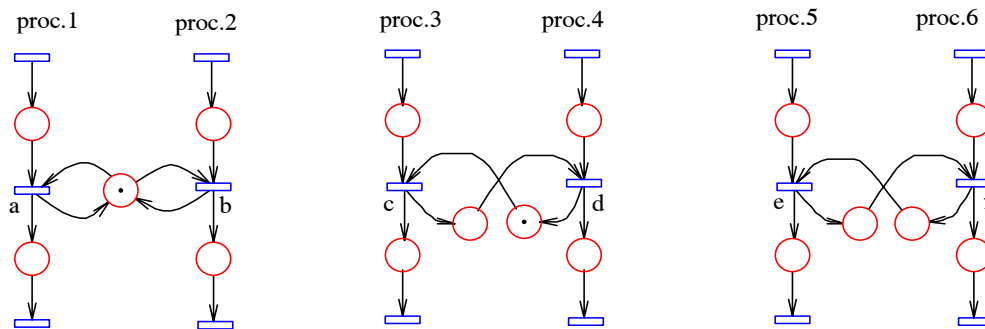
Na obr.1.2.7 je zobrazen fragment Petriho sítě, ilustrující synchronizaci paralelních procesů a) pomocí *semaforu*, b) pomocí tzv. *rendes-vous*. Proces 1. nastavuje semafor na "volno" pro proces 2. (událost b2 nemůže nastat dříve než událost b1), který jej za sebou shazuje (nastavuje na "stát"). Událost b představuje schůzku (rendes-vous) procesů 3.a 4., kterákoliv z událostí c3,c4 může nastat až po uskutečnění obou dvou událostí a3,a4.



Obr.1.2.7 Synchronizace procesů

Příklad 1.2.5:

Na obr.1.2.8 jsou zobrazeny fragmenty Petriho sítí, ilustrující některé některé další typické situace se kterými se setkáváme při modelování paralelních procesů a to: a) vyloučení souběhu činnosti v tzv. *kritické sekci* (obr.1.2.8.a), b) zabezpečení pravidelného střídání dvou činností (obr.1.2.8.b) a c) triviální případ tzv. *deadlocku* neboli uzamčení (obr.1.2.8.c). Události a, b procesů 1 a 2 nemohou nastat současně (činnosti a, b nemohou současně probíhat). Události c, d procesů 3 a 4 se musí pravidelně střídát. Žádná z událostí e, f nemůže nikdy nastat.



Obr.1.2.8.a

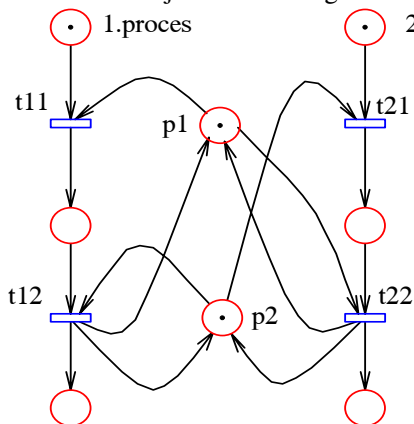
Obr.1.2.8.b

Obr.1.2.8.c

Obr.1.2.8 Kritická sekce, střídání událostí, triviální deadlock

Příklad 1.2.6:

Na obr.1.2.7 je zobrazen fragment Petriho síť, ilustrující možnost vzniku netriviálního deadlocku.



Posloupnosti provádění přechodů:

t11, t12, t21, t22 ... oba procesy projdou

t21, t22, t11, t12 ... oba procesy projdou

t11, t21 žádný proces neprojde

t21, t11 žádný proces neprojde

p1, p2 zdroje

Obr.1.2.7 Netriviální deadlock

1.3. P/T Petriho síť (Place/Transitions PN)

P/T Petriho síť je tvořena následujícími objekty:

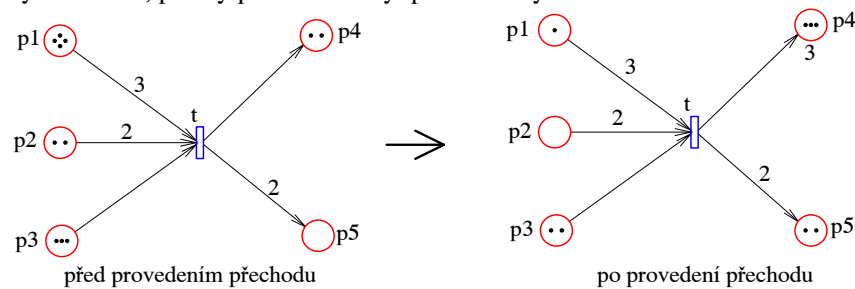
- *místa* (places), graficky reprezentovanými kružnicemi,
- *přechody* (transitions), graficky reprezentovanými obdélníky,
- orientovanými *hranami* (arcs), graficky reprezentovanými šipkami směřujícími od míst k přechodům nebo od přechodů k místům,
- udáním *kapacity* (capacity indications) pro každé místo sítě, tj. přirozeného čísla udávajícího maximální počet tokenů, který se může v místě nacházet,
- udáním *váhy* (weights) pro každou hranu sítě, tj. přirozeného čísla udávajícího násobnost hrany,
- udáním *počátečního značení* (initial marking), udávajícího počet tokenů pro každé místo sítě.

Kapacity míst a násobnosti hran se na grafovém diagramu objevují jako ohodnocení míst a hran. Hranu bez ohodnocení považujeme za jednoduchou (s násobností 1) a místo bez ohodnocení považujeme za místo s neomezenou (nekonečnou) kapacitou.

Změny stavů (značení) P/T Petriho sítě jsou charakterizovány následujícími pravidly:

- *stav sítě* je určen značením, tj. počtem tokenů v každém místě,
- místo p patří do *vstupní množiny* (pre-set) přechodu t, jestliže z místa p vede hrana do přechodu t a místo p patří do *výstupní množiny* (post-set) přechodu t, jestliže z přechodu t vede hrana do místa p,
- přechod t je *proveditelný* (enabled, activated), jestliže:
 - pro každé místo p vstupní množiny přechodu t platí, že obsahuje alespoň tolik tokenů, kolik činí násobnost hrany vedoucí z místa p do přechodu t,
 - pro každé místo p výstupní množiny přechodu t platí, že počet tokenů obsažených v místě p zvětšený o násobnost hrany, mířící z přechodu t do místa p, nepřevyšuje kapacitu místa p,
- při *provedení* (firing) proveditelného přechodu t se změní stav (značení, marking) sítě takto:
 - počet tokenů v každém vstupním místě p přechodu t se zmenší o násobnost hrany spojující toto místo s tímto přechodem,
 - počet tokenů v každém výstupním místě p přechodu t se zvětší o násobnost hrany spojující toto místo s tímto přechodem.

Na obr.1.3.1 je zobrazen změna stavu sítě po provedení proveditelného přechodu. Kdyby násobnost hrany (t, p4) byla větší nebo rovna 2, nebo kdyby kapacita místa p4 byla menší nebo rovna 2 a nebo kdyby výchozí značení místa p4 bylo rovno 3, pak by přechod t nebyl proveditelný.



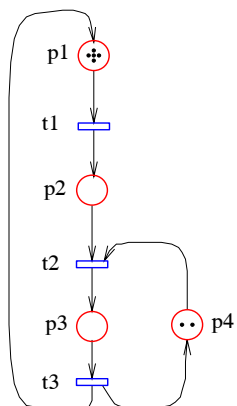
Obr.1.3.1

Poznámky 1.3.1:

1. V P/T Petriho sítích místa zpravidla označují stavy modelovaného systému a přechody změny stavu. Stav je charakterizován celým nezáporným číslem daným značením daného místa (počtem tokenů v daném místě). Při modelování počítačových dějů jedná se např. o počty jednotek volné nebo obsazené paměti (např. bufferu), o počty jednotek disponibilních nebo využívaných zdrojů různého typu a pod.
2. Implicitně (defaultně, tj. když není výslovně uvedeno jinak) předpokládáme násobnost hrany 1 a kapacitu místa nekonečnou. Násobnost jednoduchých hran ($w=1$) a kapacitu kapacitně neomezených míst ($K=\text{nekonečno}$) na grafech Petriho sítí nemusíme uvádět a kvůli větší přehlednosti obrázku také neuvádíme.
3. Zavedení vícenásobných hran a omezených kapacit míst nikterak nezvyšuje modelovací možnosti (expresivitu) P/T Petriho sítí, ale pouze umožňuje jejich jednodušší zápis. Každou P/T Petriho síť lze převést na ekvivalentní síť bez násobných hran a bez kapacitně omezených míst a přitom modelující stejný reálný problém. V příkladě 1.3.2 je ilustrován postup, jak lze ze sítě vyloučit násobnost hran a kapacitní omezenost míst.
4. Speciálním případem P/T Petriho sítí jsou:
 - **C/E Petriho síť**, což jsou P/T Petriho síť ve kterých je kapacita každého místa a násobnost každé hrany rovna 1.
 - **Obyčejné Petriho síť** (ordinary Petri nets) jsou P/T Petriho síť ve kterých je kapacita každého místa nekonečná (tj. žádné místo není kapacitně omezené) a násobnost každé hrany je rovna 1. V souladu s výše uvedenou konvencí - viz poznámka 2. - se v obyčejných sítích nevyskytují inskripce kapacit a násobností.
 - **Elementární obyčejné Petriho síť** jsou obyčejné Petriho síť pro které platí následující podmínky (viz poznámky 1.2.1):
 - každý přechod má alespoň jedno vstupní místo (přechod bez vstupních míst by umožňoval nekontrolovaný vstup libovolného počtu tokenů do sítě),
 - každý přechod má alespoň jedno výstupní místo (přechod bez výstupních míst by umožňoval nekontrolovaný výstup libovolného počtu tokenů ze sítě),
 - žádný přechod nemá žádné místo, která by pro něj bylo současně vstupním i výstupním (tj. síť je čistá).

Příklad 1.3.1:

Na obr.1.3.2 je P/T Petriho síť zobrazující pět cyklických procesů (pět exemplářů téhož procesu) využívajících dva zdroje (dva exempláře téhož zdroje). Např. se může jednat o pět terminálů počítačové sítě společně využívající dvě tiskárny. Příklad je zobecněním příkladu 1.2.1.



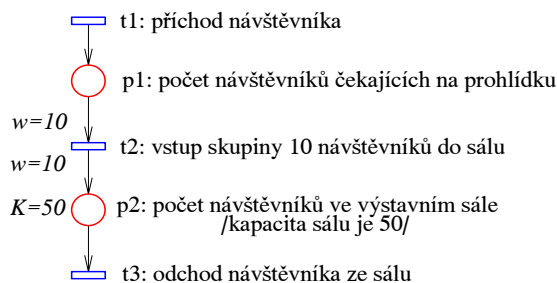
- p1: počet procesů nevyužívajících zdroj
- p2: počet procesů čekajících na zdroj
- p3: počet procesů využívajících zdroj
- p4: počet nevyužitých zdrojů
- t1: vznik požadavku na přidělení zdroje
- t2: přidělení zdroje procesu
- t3: uvolnění zdroje procesem

Obr.1.3.2

Příklad 1.3.2:

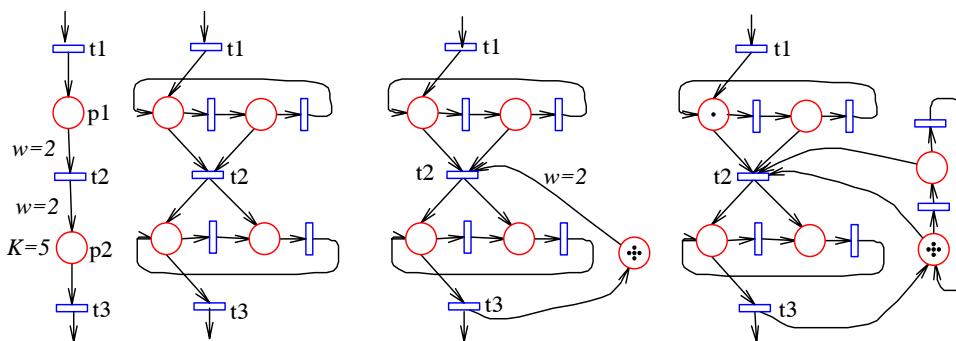
Na obr.1.3.3 je P/T síť zobrazujících pohyb návštěvníků ve výstavní síni. Platí následující pravidla:

- (1) počet návštěvníků ve výstavní síni nesmí v žádném okamžiku překročit hodnotu $K = 50$,
- (2) do výstavní síně vstupují návštěvníci po skupinách $w = 10$ osob (např. vždy spolu s průvodcem),
- (3) návštěvníci mohou odcházet jednotlivě.



Obr.1.3.3

Na obrázcích 1.3.3.a-d je ilustrován postup eliminace násobných hran a kapacitně omezených míst. Na obr. 1.3.3.a je výše popsána situace popsána sítí s násobnými hranami a s kapacitním omezením místa (až na to, že kvůli jednoduchosti jsme místo $w=10$ a $K=50$ volili $w=2$ a $K=5$). Na obr. 1.3.3.b jsou eliminovány násobné hrany (místa p1, p2 jsou nahrazeny podsítěmi tvořenými dvěma místy a dvěma přechody), není však uvažováno kapacitní omezení. To je napraveno na obr. c, vzniká však nová násobná hrana. Tato násobná hrana je eliminována v síti na obr. d, a to stejným způsobem jakým byla eliminována násobná hrana (p1, t2) při přechodu od sítě z obr.a k síti na obr.b. Síť z obr.d je ekvivalentní se sítí z obr.a. (obě sítě zobrazují stejné chování návštěvníků ve výstavní síni).

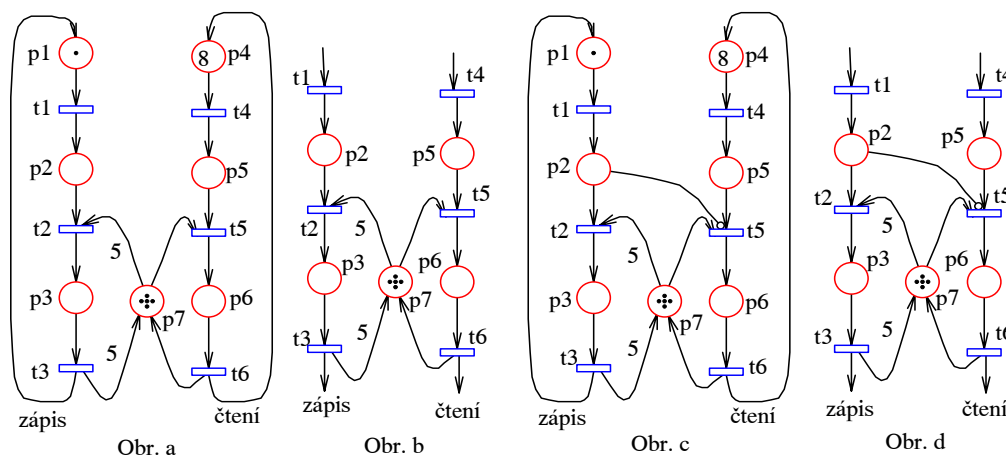


Obr.1.3.3.a-d

Příklad 1.3.3:

Na obr.1.3.4 je pomocí P/T sítě zobrazen systém zahrnující osm procesů, které čas od času čtou údaje z dané databáze a jeden proces, který čas od času do databáze údaje zapisuje (aktualizuje databázi). Přitom platí, že současně může z databáze číst nejvýše pět procesů a pokud je do databáze zapisováno, pak z ní nemůže číst ani jeden proces. Systém je implementován pomocí tzv. klíčů, kterých je v systému celkem 5. K tomu, aby proces mohl číst z databáze musí získat jeden klíč a k tomu, aby mohl zapisovat do databáze musí získat všech pět klíčů. Tato situace je zobrazena uzavřenou sítí na obr. 1.3.4.a. Na obr. 1.3.4.b. je zobrazena odpovídající otevřená síť - zapisující iprocesy vstupují do sítě přechodem t1 a čtoucí procesy přechodem t4 (počty zapisujících a čtoucích procesů nejsou předem dány).

Při uspořádání podle obrázků a) a b) může dojít k tomu, že zapisující proces se nikdy nedostane ke slovu (pět klíčů nebude nikdy k dispozici). Abychom zabránili této možnosti lze požadovat přednost procesu zapisování před procesem čtením: jakmile vznikne požadavek na zápis (objeví se token v místě p2) zakáže se vstup dalšího "čtenáře" do databáze (znemožní se provedení přechodu t5). To lze zařídit pomocí tzv. inhibiční hrany vedené z místa p2 do přechodu t5 (zobrazeno na obr. 1.3.4 c, d) - viz dále kap. 1.4 o Petriho sítích s inhibičními hranami, anebo přiřazení vyšší priority přechodu t2 ve srovnání s přechodem t5 - viz dále kap. 1.5 o Petriho sítích s prioritami.



Obr.1.3.4 Několik (osm) čtecích a jeden zapisovací proces

Počty tokenů v jednotlivých místech mají následující význam:

- p1: token označuje neexistenci potřeby zápisu
- p2: token označuje čekání na povolení zápisu (čekání na osm klíčů)
- p3: token označuje, že probíhá zápis do databáze
- p4: počet procesů nemajících potřebu čtení
- p5: počet procesů čekajících na povolení čtení (čekání na jeden klíč)
- p6: počet procesů čtoucích v databázi
- p7: počet disponibilních klíčů (poukázek ke vstupu do databáze)

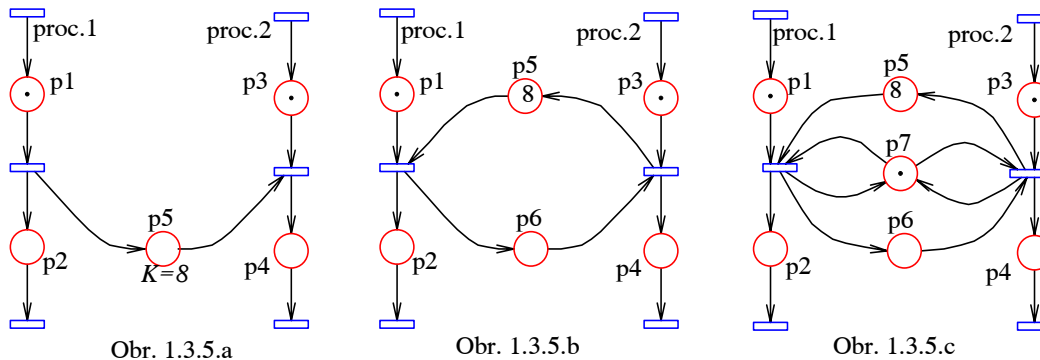
Přechody v sítích mají tento význam:

- t1: vznik potřeby zápisu
- t2: získání povolení zápisu
- t3: ukončení zápisu (vrácení pěti klíčů)
- t4: vznik potřeby čtení
- t5: získání povolení čtení
- t6: ukončení čtení (vrácení klíče)

Příklad 1.3.4:

Na obr.1.3.5 je zobrazen fragment P/T Petriho sítě modelující vyrovnávací paměť (buffer) pro předávání dat produkovaných procesem 1 procesem 2. Ve variantě a) je buffer modelován místem p5 s kapacitou $K(p5)=8$. Ve variantě b) je buffer modelován dvěma místy p5, p6 a to oběma s neomezenou kapacitou. Počet tokenů

v místě p5 signalizuje počet volných pozic v bufferu a počet tokenů v místě p6 počet obsazených pozic. Celkový počet pozic (kapacita vyrovnávací paměti) je opět 8. Obě varianty a), b) jsou rovnocenné a ilustrují skutečnost, že pojem kapacity místa umožňuje sice zjednodušit síť, ale v principu je možné se bez něj obejít. Síť na obr. c) vylučuje současný přístup k bufferu pro oba procesy: nelze současně do bufferu ukládat i vybírat.



Obr.1.3.5 Buffer pro předávání dat mezi procesy

1.4. Petriho síť s inhibičními hranami (P/T PN with inhibitors)

Petriho síť s inhibičními hranami je P/T síť (viz popis P/T sítí v kapitole 1.3) ve které se mohou vyskytovat tzv. **inhibiční hrany** (inhibitor arcs) směřující výhradně od míst k přechodům a zobrazované orientovanými úsečkami zakončenými kroužkem namísto šípky.

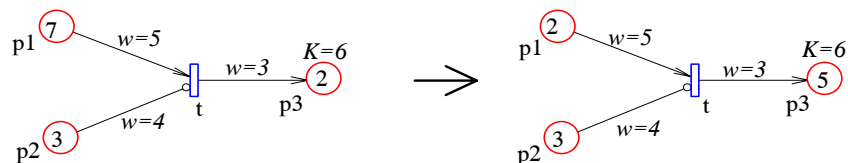
V Petriho síti s inhibičními hranami:

- místo p patří do **vstupní inhibiční množiny** přechodu t, jestliže z místa p vede inhibiční hrana do přechodu t
- **enabling rule**: přechod t je **proveditelný** (enabled), jestliže současně:
 - přechod t je proveditelný v příslušné P/T síti (tj. síti, která vznikne z dané sítě odstraněním inhibičních hran),
 - pro každé místo p vstupní inhibiční množiny přechodu t platí, že obsahuje méně (<) tokenů než kolik činí násobnost hrany vedoucí z místa p do přechodu t.
- **firing rule**: při **provedení** (firing) proveditelného přechodu t se změní stav (marking) sítě přesně stejným způsobem jako u příslušné P/T sítě (sítě, která vznikne z dané sítě odstraněním inhibičních hran).

Poznámky 1.4.1:

1. Na rozdíl od "obyčejných" hran se po inhibičních hranách "nepřesouvají" tokeny. Inhibiční hrany jsou vlastně jen negativně pojatými testovacími hranami.
2. P/T Petriho síť je speciálním případem Petriho sítě s inhibitory, ve které je množina inhibičních hran prázdná.
3. Modelovací síla (expresivita) Petriho sítí s inhibičními hranami je stejná jako modelovací síla Turingových strojů a tedy podstatně větší než obyčejných P/T Petriho sítí.

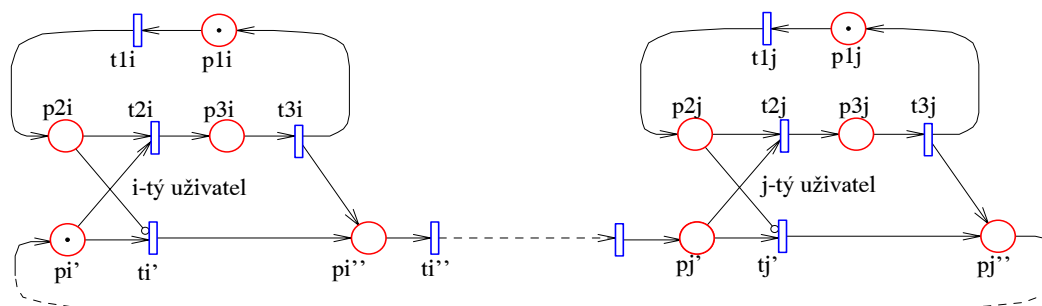
Na obr.1.4.1 je zobrazena změna stavu sítě před a po provedení proveditelného přechodu. Kdyby hrana (p2,t) měla násobnost menší nebo rovnou 3, nebo kdyby značení místa p2 bylo větší nebo rovno 4, pak by přechod t nebyl proveditelný.



Obr.1.4.1 Změna značení po provedení proveditelného přechodu

Příklad 1.4.1:

Na obr.1.4.2 je pomocí Petriho sítě (s inhibitory) zobrazen jednoduchý systém komunikace v kruhově organizované počítačové síti (Token Ring LAN). Právo ovládat komunikaci v síti je cyklicky nabízeno každému terminálu sítě. Má-li terminál nějakou zprávu k odeslání, pak svého práva využije a zprávu odešle. Nemá-li žádnou zprávu k odeslání, pak právo ovládnutí komunikace v síti ihned předá dalšímu (sousednímu) terminálu.



Obr.1.4.2 Komunikace v kruhové počítačové síti

Význam míst a přechodů ($i=1,2,\dots,N$):

- $p1i$: stav i -tého terminálu: bez zprávy k odeslání
- $t1i$: vznik zprávy v i -tém terminálu
- $p2i$: zpráva je vyhotovena a i -tý terminál čeká na příchod tokenu v síti (až se terminál dostane na řadu)
- $t2i$: i -tý terminál přebírá kontrolu nad komunikací v síti
- $p3i$: zpráva od i -tého terminálu je předávána v síti
- $t3i$: zpráva byla předána, i -tý terminál vrací kontrolu
- pi' : i -tý terminál je na řadě, má-li zprávu, může ji předat
- ti' : i -tý terminál rezignuje na kontrolu sítě
- pi'' : i -tý terminál skončil kontrolu sítě
- ti'' : kontrola je předávána následujícímu ($i+1$)-ému terminálu

Příklad 1.4.2:

Jiný příklad použití inhibičních hran byl již zmíněn v příkladu 1.3.3 - viz text a obrázky 1.3.4.c,d.

1.5. Petriho síť s prioritami (P/T PN with priorities)

Petriho síť s prioritami je P/T síť (viz popis P/T sítě v kapitole 1.3) ve které je navíc ke každému přechodu síť přiřazeno celé nezáporné číslo udávající tzv. *prioritu přechodu*.

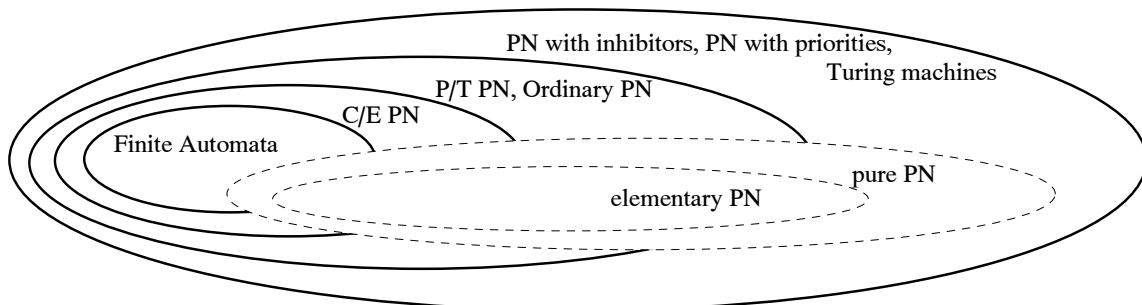
V Petriho síti s prioritami je:

- přechod t je *povolen* (has concession), je-li proveditelný v odpovídající P/T síti bez priorit.

- přechod t je **proveditelný** (enabled), jestliže současně:
 - je povolen,
 - žádný jiný povolený přechod nemá vyšší prioritu.
- při **provedení** (firing) proveditelného přechodu t se změní stav (marking) sítě stejným způsobem jako u P/T sítě bez priorit. Současně mohou být provedeny pouze přechody se stejnou prioritou. Jsou-li přechody se stejnou prioritou v konfliktu, pak výběr přechodu k provedení je nedeterministický.

Poznámky 1.5.1:

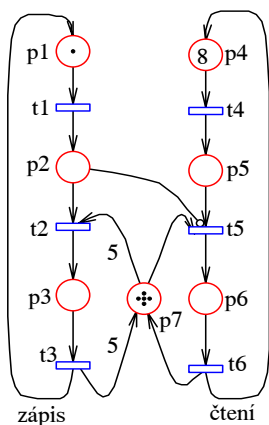
1. P/T Petriho síť je speciálním případem Petriho sítě s prioritami, ve které je všem přechodům přiřazena jedna a táž priorita (což je totéž jakoby priorita přechodů nebyla vůbec definována).
2. Petriho síť s prioritami mají stejnou modelovací sílu jako Petriho síť s inhibičními hranami, tj. modelovací sílu Turingových strojů lze jimi modelovat libovolný algoritmus.
3. Porovnání modelovací síly dosud zmiňovaných typů Petriho sítí ukazuje následující množinový obrázek 1.5.1 (Vennův diagram).



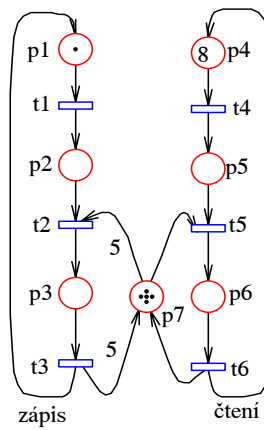
Obr.1.5.1

Příklad 1.5.1:

Petriho síť s inhibičními hranami (a bez priorit) mohou být převedeny na ekvivalentní síť s prioritami (a bez inhibičních hran). Na obr. 1.5.1 je znovu pojednán příklad 1.3.3 ve variantě, kdy požadavek zápisu do databáze má přednost před požadavkem čtení z databáze. Na obr. 1.5.1.a je zobrazeno řešení s inhibičními hranami a na obr.1.5.1.b je zobrazeno řešení s prioritami.



Obr. a



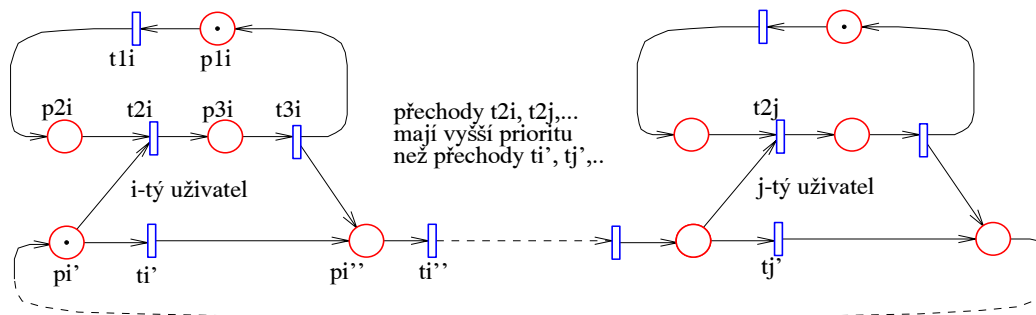
Obr. b

přechod t_2 má vyšší prioritu než přechod t_5

Obr. 1.5.1

Příklad 1.5.2:

Na obr.1.5.2 je Petriho síť s prioritami zobrazující systém komunikace v kruhově organizované počítačové síti (viz příklad 1.4.1). Petriho síť z obr.1.5.2 je ekvivalentní (popisuje tentýž komunikační systém) s Petriho sítí s inhibičními hranami z obr.1.4.2.



Obr.1.5.2 Komunikace v kruhové počítačové síti

1.6. Časované Petriho síť (Timed PN)

Žádný z dosud uvedených typů Petriho sítí nepracuje s pojmem času a trvání. Předpokládá se, že provedení libovolného přechodu je okamžité. Posloupnost stavů (značení) Petriho sítě můžeme sice zobrazit posloupností bodů na časové ose, podstatné je však pouze pořadí těchto bodů a nikoliv jejich konkrétní umístění (na časové ose není zavedena metrika).

Existuje několik způsobů jak obohatit aparát Petriho sítí pojmem času a uvažovat změny stavů (provádění přechodů) jako děje trvajících, tj. spotřebovávajících více či méně času. Je zřejmé, že tímto obohacením významně rozšíříme modelovací možnosti Petriho sítí. Trvání dějů může být charakterizováno:

- deterministicky (přiřazené časy jsou konstanty),
- stochasticky (přiřazené časy jsou náhodné, zpravidla s exponenciálním rozdělením).
- kombinovaným způsobem (přiřazené časy jsou pro některé přechody konstanty a pro jiné realizacemi náhodných veličin)

V prvním případě hovoříme jednoduše o **časovaných Petriho sítích** (bez přívlastku), v druhém případě o **stochastických Petriho sítích** (SPN - Stochastic Petri Nets) a v třetím případě o **zobecněných stochast. Petriho sítích** (GSPN - Generalized Stochastic Petri Nets).

Časové charakteristiky mohou být spojeny s různými stavebními prvky Petriho sítí a to s:

- přechody (t-timed PN),
- místy (p-timed PN),
- hranami (a-timed PN),
- tokeny (token timed PN).

V případě časování přechodů si představujeme, že tokeny pobývají po jistý čas (po dobu trvání přechodu) uvnitř přechodu.

V případě časování míst si představujeme, že tokeny pobývají po určitou dobu ve vstupních místech přechodu, který má být proveden. Tato doba se začíná počítat od okamžiku uschopnění (enabling) přechodu a vyjadřuje dobu trvání přechodu.

V případě časování hran si představujeme, že tokeny se pohybují po hranách konečnou rychlostí, např. tak, že doba cesty ze vstupních míst do přechodu, který má být proveden, je rovna trvání přechodu.

V případě časování tokenů předpokládáme, že provádění přechodů je sice okamžité, ale tokeny opouštějící provedený přechod jsou opatřeny tzv. časovým razítkem (time stamp), které zaznamenává okamžik počínaje kterým mohou být tokeny znovu použity. Hodnota časového razítka je rovna aktuální hodnotě globálního času zvětšeného v okamžiku provedení přechodu o stanovenou dobu trvání přechodu.

Lze ukázat, že všechny výše uvedené modely jsou navzájem ekvivalentní. Prvý model se zdá být nejpřirozenější, je však nejméně výhodný z pohledu použití formálních metod analýzy Petriho sítí. Vede totiž k zavedení přechodných značení, kdy některé tokeny jsou "schovány" v probíhajících přechodech. Tato skutečnost

komplikuje použití dobře propracovaných metod analýzy nečasovaných sítí k analýze časovaných sítí, které vzniknou z nečasovaných časovým ohodnocením přechodů.

1.7. Barevné Petriho síť (Coloured PN)

Barevná (nebo také barvená) **Petriho síť** (CPN - Coloured Petri Net) je obyčejná P/T Petriho síť ve které navíc:

- namísto jednoho druhu tokenů ("černých" tokenů "•" z P/T sítí) je možná pracovat s mnoha různými tokeny ("barevnými" tokeny, tokeny lišícími se barvou); množina všech různobarevných tokenů je rozložena do tříd (tříd barev); tokeny z téže třídy považujeme za tokeny téhož typu;
- typy (obrazně: třídy barev) můžeme chápat jako datové typy, které mohou být konečné (např. výčtové typy, ...) nebo nekonečné (např. typ přirozené číslo, ...), elementární (např. den, měsíc, rok, reálné číslo, ...) nebo složené (např. datum, vektor reálných čísel, ...); datový typ je dán množinou hodnot a s datovými typy jsou spojeny rozmanité operace (např. logické operace s booleovským datovým typem, číselné operace s číselnými datovými typy, ...); systém datových typů spolu se systémem operací (jednodruhovými i vícedruhovými), pak představuje vícedruhovou algebru (sigma algebru); každá konkrétní CPN je tak založena na konkrétní vícedruhové algebře
- každému místu je přiřazen je **typ tokenů** (třída barev - colour set), které se mohou v daném místě nacházet; aktuální značení (stav) každého místa je dán konkrétní multimnožinou tokenů toho typu, který místu náleží
- každému přechodu je přiřazena **podmínka přechodu** (guard) reprezentovaná výrazem utvořeným z konstant a proměnných, který po vyhodnocení (tj. po dosazení konkrétních hodnot za proměnné a provedení výpočtu) dává booleovskou (pravdivostní) hodnotu
- každé hraně je přiřazen tzv. **hranový výraz** (arc expression) utvořený z konstant a proměnných, který po vyhodnocení představuje multimnožinu tokenů toho typu, který náleží místu jež je s hranou incidentní
- **počáteční značení** (initial marking) sítě je dáno počátečním značením všech míst sítě: ke každému místu je přiřazena konkrétní multimnožina tokenů z té třídy, která je místu přiřazena.

Změny stavů (značení) barevné Petriho sítě jsou charakterizovány následujícími pravidly:

- **enabling rule**: přechod t je **proveditelný** (enabled), jestliže :
 - multimnožina tokenů obsažená v každém vstupním místě p přechodu t je větší nebo rovná multimnožině, která byla vypočtena při vyhodnocení hranového výrazu, který přísluší hraně vedoucí z místa p do přechodu t ,
 - je splněná podmínka přechodu t , tj. vyhodnocení booleovského výrazu přiřazeného přechodu t dává hodnotu true

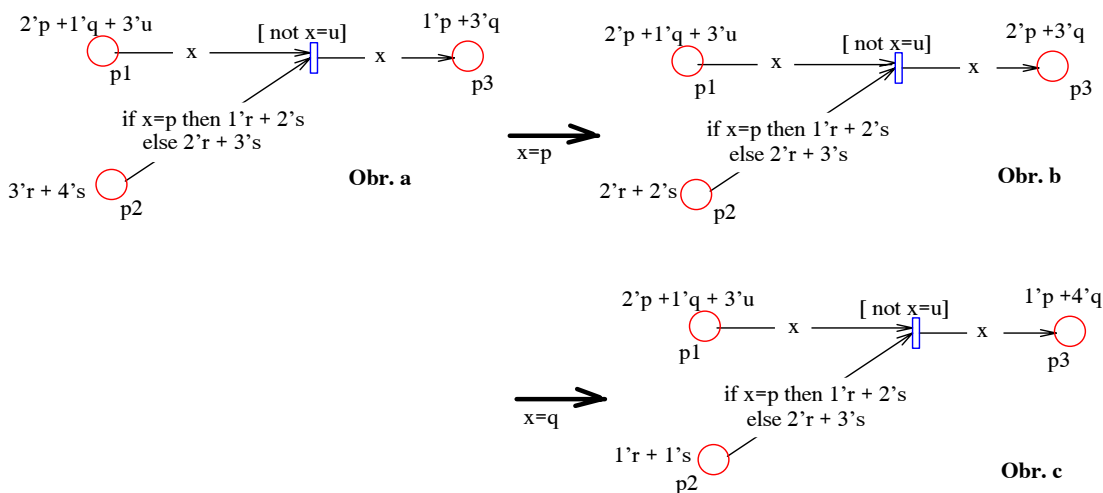
Jestliže v hranových výrazech hran incidentních s přechodem t nebo v podmínce přechodu t se nachází nějaké proměnné, pak je třeba těmto proměnným přiřadit konkrétní hodnoty. Různým přiřazením pak odpovídají různé způsoby (módy) provedení přechodu. Přechod může být proveditelný při všech, některých nebo žádném přiřazení.

- (firing rule) při **provedení** (firing) proveditelného přechodu t se změní stav (značení) sítě takto :
 - multimnožina tokenů v každém místě vstupní množiny přechodu t se zmenší o multimnožinu, která vznikne vyhodnocením hrany spojující toto místo s přechodem t ,
 - multimnožina tokenů v každém výstupním místě přechodu t se zvětší o multimnožinu, která vznikne vyhodnocením výrazu přiřazeného hraně spojující přechod t s tímto místem.

Vyhodnocení hranových výrazů všech vstupních i výstupních hran prováděného přechodu musí být uskutečněno ve stejném módu, tj. při stejných přiřazeních konkrétních hodnot stejným proměnným.

Na obr.1.7.1 je ilustrován pojem proveditelnosti přechodu a změna značení po jeho provedení. Místa p_1, p_3 mají přidělen výčtový datový typ $\{p, q, u, v\}$ (prvky p, q, u, v mohou představovat např. různé druhy procesů), tj. v těchto místech se mohou vyskytovat libovolné multimnožiny tokenů nad množinou $\{p, q, u, v\}$. Proměnná x je

téhož datového typu, tj. může nabývat libovolné hodnoty z množiny $\{p,q,u,v\}$. Místo p2 má přidělen datový typ $\{r,s,w\}$ (prvky r,s,w mohou označovat např. různé druhy zdrojů), tj. v místě p2 se mohou nacházet rozmanité multimnožiny tokenů nad množinou $\{r,s,w\}$. Na obr. 1.7.1.a-c jsou vyznačeny aktuální značení (stavy, tokenové obsazení) všech tří míst, hranové výrazy všech tří hran a podmínka spojená s přechodem (uvedená v hranatých závorkách, zápis "not $x=u$ " znamená $\neg x=u$, neboli $x \neq u$). Na obrázku 1.7.1.a je zobrazen stav sítě před provedením přechodu t , na obr. 1.7.1.b stav sítě po provedení přechodu v módu $x=p$ a na obr. 1.7.1.c stav sítě po provedení přechodu v módu $x=q$. V dalších dvou možných módech $x=u$ a $x=v$ je přechod neproveditelný: v prvním případě proto, že není splněna podmínka přechodu a v druhém případě proto, že při výchozím značení se v místě p1 žádný exemplář tokenu v nevyskytuje.



Obr.1.7.1 Proveditelnost a provedení přechodu

Poznámky 1.7.1:

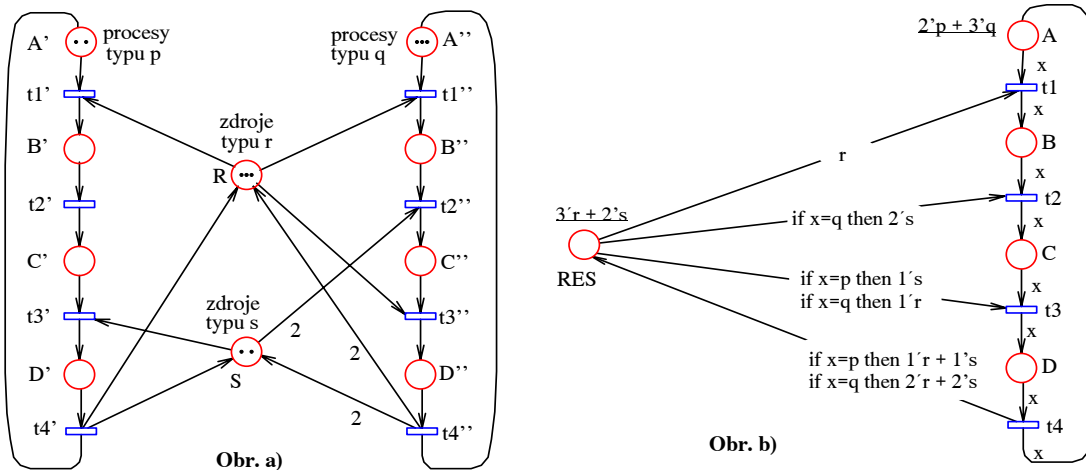
1. P/T PN jsou speciálním případem CPN, která pracuje s jediným datovým typem a to výtčovým datovým typem s jedinou hodnotou $\{\bullet\}$. Značení míst se pak vyjadřuje multimnožinami nad $\{\bullet\}$, např. $3 \bullet$.
2. CPN nepotřebuje násobné hrany, omezené kapacity míst, inhibiční hrany, priority. Všechny tyto konstrukty z P/T PN mohou být plně nahrazeny obecnějšími konstrukcemi CPN: hranové výrazy, podmínky přechodů,...
3. Podobně jako místo může mít různé značení (reprezentovat různý lokální stav), tak i přechod - v případě CPN - může být prováděn různými způsoby. V jednom a téže místě mohou být různé multimnožiny tokenů (téhož typu) a podobně jeden a tentýž přechod může provádět různé transformace vstupních multimnožin tokenů (na vstupních hranách přechodu) na výstupní multimnožiny (na výstupních hranách přechodu).

Příklad 1.7.1:

Na obrázcích 1.7.2 je zobrazen systém procesů, které využívají v různých fázích své cyklické činnosti různé zdroje. Jedná se o dva procesy typu p a tři procesy typu q a dva zdroje typu r a tři zdroje typu s . Procesy různých typů se liší požadavky na přidělování zdrojů (co do typu i množství) během cyklu, procesy téhož typu mají požadavky identické.

Na obr.1.7.2.a je systém zobrazen pomocí obyčejné P/T Petriho sítě (P/T PN) a na obr.1.7.2.b je tentýž systém reprezentován kompaktněji barevnou Petriho sítí (CPN). Zatímco P/T PN pracuje pouze s tokeny jediné "černé" barvy $\{\bullet\}$, CPN pracuje se čtyřmi barvami tokenů $\{p,q,r,s\}$. Tyto čtyři barvy jsou rozděleny do dvou datových (barevných) tříd: $\text{proces} = \{p,q\}$ a $\text{zdroj} = \{r,s\}$. V každém místě v CPN se může vyskytovat libovolný počet tokenů, ale vždy jen téhož typu, tj. patřících do téže datové třídy (třídy barev). V místech A,B,C,D se mohou vyskytovat tokeny typu proces a v místě RES tokeny typu zdroj . Aktuální značení míst A,B,C,D je dáno multimnožinami nad $\{p,q\}$ a aktuální značení místa RES multimnožinou nad $\{r,s\}$. Počáteční značení je

zakresleno na obrázcích. Počáteční značení CPN na obr.1.7.2.b jest: v místech B,C,D jsou prázdné multimnožiny tokenů (tj. tato místa neobsahují žádné procesy), v místě A se nachází multimnožina $2^{\cdot}p+3^{\cdot}q$, (tj. v místě A se nachází dva exempláře procesu p a tři exempláře procesu q) a v místě RES se nachází multimnožina $3^{\cdot}r+2^{\cdot}s$ (tj. tři exempláře zdroje r a dva exempláře zdroje s). Hranám v CPN jsou přiřazeny hranové výrazy obsahující konstanty p,q,r,s a proměnnou x , která je typu *proces*, tj. probíhá množinu $\{p,q\}$. Po volbě hodnoty proměnné x lze hranové výrazy vyhodnotit, tj. hranové výrazy pak představují multimnožinu tokenů nad datovým typem příslušným místu se kterým je hrana incidentní.



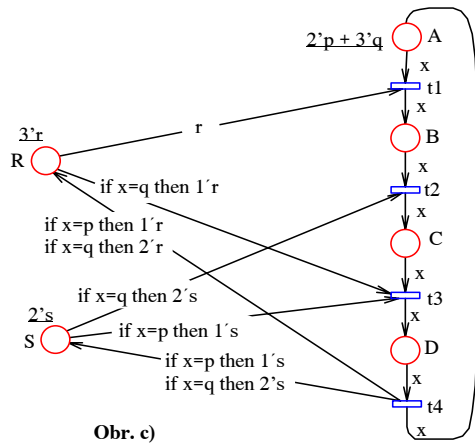
Obr. 1.7.2.a-b

Obrázky 1.7.2.b-d představují různé varianty CPN pro P/T PN z obrázku 1.7.2.a. Dosud diskutovaná varianta (b) pracuje s elementárními výčtovými datovými typy $\{p,q\}$ a $\{r,s\}$, varianta (c) pracuje s elementárními výčtovými datovými typy $\{p,q\}$, $\{r\}$ a $\{s\}$.

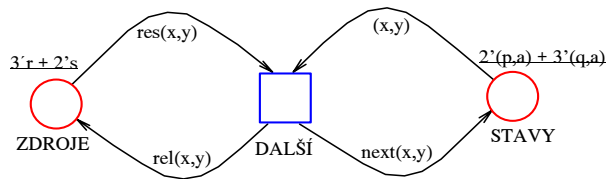
Varianta (d) vychází z elementárních výčtových datových typů $\{p,q\}$, $\{r,s\}$ a $\{a,b,c,d\}$, kde a,b,c,d jsou hodnoty dalšího výčtového datového typu (typu *fáze* = $\{a,b,c,d\}$), vyjadřující v jaké fázi se ten či onen proces nachází (tj. v kterém z míst A,B,C,D se nachází token zobrazující příslušný proces). Na základě elementárních datových typů $\{p,q\}$ a $\{a,b,c,d\}$ lze konstruovat složený datový typ *stav* s osmi hodnotami $\{p,q\} \times \{a,b,c,d\} = \{(p,a),(p,b),(p,c),(p,d),(q,a),(q,b),(q,c),(q,d)\}$. CPN pracuje se dvěma elementárními proměnnými x typu *proces* a y typu *fáze*, hodnota složené proměnné (x,y) udává, že proces typu x je ve fázi y . CPN ve variantě (d) je tvořena dvěma místy: ZDROJE (obsahuje multimnožinu aktuálně disponibilních zdrojů) a STAVY (obsahuje multimnožinu stavů všech procesů) a jedním přechodem DALŠÍ (jeho provedením se mění stav jednoho procesu /a tím i multimnožiny stavů všech procesů/ a multimnožiny disponibilních zdrojů). Hranové výrazy mají následující význam:

- (x,y) elementární stav (proces typu x ve fázi y), který je předmětem změny
- $res(x,y)$ multimnožina zdrojů potřebná k uskutečnění změny (res ... reserve)
- $rel(x,y)$ multimnožina zdrojů uvolněná po uskutečnění změny (rel ... release)
- $next(x,y)$... nový elementární stav nahrazující stav (x,y)

Podrobné definice funkcí $res(x,y)$, $rel(x,y)$ a $next(x,y)$ jsou uvedeny v tabulce 1.7.1 (přímé uvedení v obrázku 1.7.2.d by tento obrázek příliš zatížilo).



Obr. c)



Obr. d)

Obr. 1.7.2.c-d

| (x,y) | $res(x,y)$ | $rel(x,y)$ | $next(x,y)$ |
|---------|-------------|-------------|-------------|
| (p,a) | $1'r$ | \emptyset | $1'(p,b)$ |
| (p,b) | \emptyset | \emptyset | $1'(p,c)$ |
| (p,c) | $1's$ | \emptyset | $1'(p,d)$ |
| (p,d) | \emptyset | $1'r + 1's$ | $1'(p,a)$ |
| (q,a) | $1'r$ | \emptyset | $1'(q,b)$ |
| (q,b) | $2's$ | \emptyset | $1'(q,c)$ |
| (q,c) | $1'r$ | \emptyset | $1'(q,d)$ |
| (q,d) | \emptyset | $2'r + 2's$ | $1'(q,a)$ |

Tab. 1.7.1

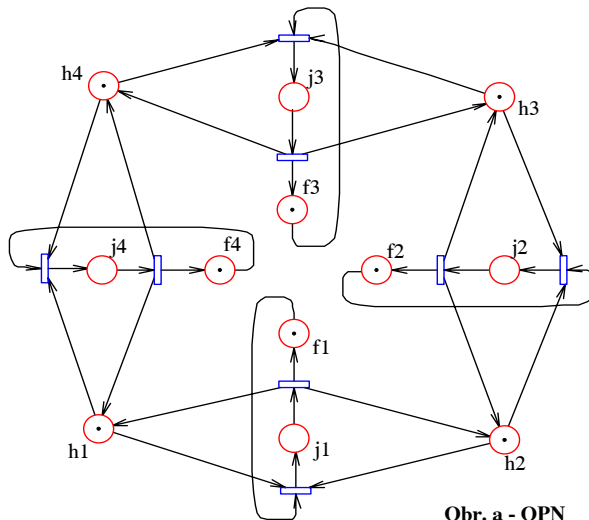
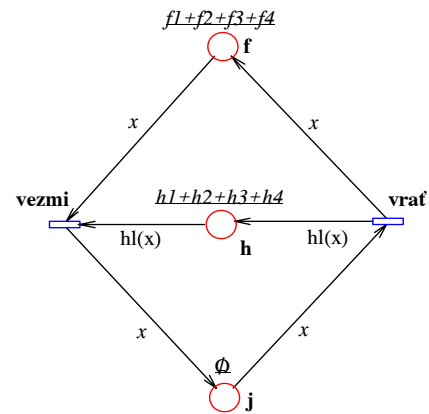
Poznamenejme, že síť zobrazená na obr. 1.7.2.a (a tedy také síť na obr.1.7.2.b-d) obsahují deadlocky. Např. posloupnost přechodů $t1'', t1'', t1'', t2''$ vede k uzamčenému (deadlockovému) stavu sítě a také některé další posloupnosti přechodů. Deadlockům je možné se vyhnout změnou počátečního značení sítě (zmenšením počtu procesů, zvětšením počtu zdrojů a někdy překvapivě i zmenšením počtu zdrojů) a u CPN verzí také využitím podmínek přechodů .

Příklad 1.7.2:

Klasický problém multiprocesové synchronizace, známý pod anglickým názvem *Dining Philosophers* (večeřící nebo jedící filozofové), budeme reprezentovat pomocí OPN (obyčejné P/T Petriho sítě) a CPN (barevné Petriho sítě).

Problém spočívá v následujícím: Několik filozofů (v našem případě 4) sedí okolo kulatého stolu a nemají nic jiného na práci než přemýšlet a jíst. Mezi každými dvěma sousedními filozofy je na stole položena jedna jídelní hůlka. K tomu, aby filozof byl schopen jíst musí mít dvě jídelní hůlky: hůlku z levé i pravé strany. Problém nastane tehdy, když každý filozof chopí hůlku po levici a čeká až se uvolní hůlka po pravici. V tomto případě nastane deadlock a všichni filozofové zahynou hladem. Tomu se lze vyhnout např. tím, že žádný filozof nesmí zdvihat hůlku jednotlivě, ale vždy jen současně (to dále také předpokládáme). Často se požaduje také spravedlnost (fairness): každý filozof má mít možnost se najíst tolikrát jako ostatní (tento požadavek v sítích na obr. 1.7.3 zabezpečen není).

Na obr. 1.7.3.a je problém zobrazen pomocí OPN. Token v místě h_i ($i=1,2,3,4$) znamená, že i -tá hůlka je neuchopena na stole, token v místě f_i značí, že i -tý filozof přemýšlí (filozofuje) a token v místě j_i indikuje, že i -tý filozof se věnuje jídlu ($i=1,2,3,4$). Na obrázku je zobrazeno počáteční značení sítě, kdy všichni filozofové přemýšlí, žádný nejí a všechny hůlky jsou položeny na svém místě na stole, tj. 1.filozof mezi 1. a 2. hůlkou, 2.filozof mezi 2. a 3. hůlkou. 3. filozof mezi 3. a 4. hůlkou a konečně 4.filozof mezi 4. a 1. hůlkou.


 Obr. a - OPN
 Obr.1.7.3.


Obr. b - CPN

Na obr. 1.7.3.b je uvažovaná varianta "Dining Philosophers" problému vyjádřena ekvivalentní barevnou Petriho sítí. CPN pracuje s:

- datovým typem $hůlka = \{h1, h2, h3, h4\}$,
- datovým typem $filosof = \{f1, f2, f3, f4\}$,
- proměnnou x typu filosof, tj proměnnou nabývající hodnot z množiny $\{f1, f2, f3, f4\}$,
- funkci $hl(x)$ přiřazující ke každému filozofovi dvojici (multimnožinu, množinu) hůlek jež jsou mu sousední, tj.
 $hl(f1) = 1 \cdot h1 + 1 \cdot h2 = h1 + h2$, $hl(f2) = h2 + h3$, $hl(f3) = h3 + h4$, $hl(f4) = h4 + h1$.

Místo h sítě CPN je vlastně sjednocením míst $h1, h2, h3, h4$ sítě OPN, místo f představuje sjednocení míst $f1, f2, f3, f4$ a místo j je sjednocením míst $j1, j2, j3, j4$. Počátečním značením míst h, f, j jsou postupně (multi)množin $h1+h2+h3+h4, f1+f2+f3+f4$ a \emptyset .

V počátečním značení je proveditelný přechod **vezmi** a to ve všech čtyřech módech: $x=f1, f2, f3, f4$. Přechod **vrať** je proveditelný vždy, obsahuje-li místo j aspoň jeden token.

1.8. Hierarchické Petriho síť (Hierarchical PN)

Jednoúrovňový způsob návrhování a modelování systémů má řadu známých nevýhod:

- ztráta přehledu, záběr příliš mnoha detailů v jednom okamžiku,
- žádné nebo nedostatečné zobrazení vnitřní struktury systému,
- pracný návrh, malá spolehlivost navrženého systému, obtížná údržba systému.

Hierarchický způsob návrhu a modelování tyto nedostatky překonává a vyznačuje se následujícími přednostmi:

- rozdělení systému do dobře definovaných komponent,
- zakrytí vnitřní struktury komponent při práci s komponentami,
- možnost vícenásobného užití komponent při návrhu systému,
- možnost návrhu systému metodou "shora dolů" i "zdola nahoru",
- možnost paralelní práce při návrhu systému,
- snadná údržba systému.

Hierarchická Petriho síť je částečně uspořádaná množina nehierarchických Petriho sítí (které všechny mohou být libovolného, ale téhož typu: C/E, P/T, barevné,...), tzv. stránek (pages). V tomto uspořádání je stránka A pod stránkou stránky B jestliže síť na stránce A rozvíjí (definuje pomocí Petriho sítě) některý prvek (přechod nebo místo) ze sítě na stránce B.

Při tomto rozvíjení lze používat různé **hierarchizační konstrukty** (hierarchy constructs):

- substituce přechodů (substitutions of transitions),
- substituce míst (substitution of places),
- volání přechodů (invocation of transitions),
- slučování přechodů (fusion of transitions),
- slučování míst (fusion of places).

Při **substituci přechodu** je daný přechod v dané síti nahrazen substituující sítí, která poskytuje podrobnější popis aktivity, kterou reprezentuje substituovaný přechod. Podobně při **substituci místa** je dané místo v dané síti nahrazeno substituující sítí, která poskytuje podrobnější popis stavu, který reprezentuje substituované místo. Poznamenejme, že stav může mít dynamického charakter, tj. může vyjadřovat skutečnost, že probíhá jistý proces.

Substituce je hierarchizační konstrukce, která se uplatňuje pouze při návrhu Petriho sítě, ale nikoliv při jejím provozování. Interpretace (execuce) sítě se uskutečňuje v základní (hierarchicky nejnižší úrovni) ve které již neexistují žádné substituční přechody nebo místa (tj. všechna dosazení byla již provedena). Z tohoto důvodu není možné, aby v substituující síti byl přímo nebo zprostředkovaně použit substituovaný přechod (nebo místo), který je touto sítí definován (zákaz rekurze).

Při **volání přechodu** je tomu jinak. Zatímco substituci přechodu můžeme přirovnat k dosazení makra, volání přechodu představuje volání podprogramu. Substituující síť je v prvním případě kódem makra (kód musí být dosazen všude tam, kde je makro použito), zatímco v druhém případě je kódem podprogramu (který je z různých míst programu volán, ale v programu se vyskytuje pouze jednou). Konstrukce volání přechodu umožňuje rekurzi, tj. v definiční síti volaného přechodu může být definovaný přechod použit (přímo nebo zprostředkovaně). Při interpretaci se definiční síť volaného přechodu vytvoří teprve tehdy, až je přechod volán a po návratu do hierarchicky vyšší úrovně se tato síť opět zruší. Počet exemplářů definiční sítě se v průběhu interpretace mění (zejména v případě rekurzivního volání). Rozsah kódu a potřebná paměť se dynamicky mění. Tím se liší použití konstruktu volání od konstruktu substituce, kdy jsou rozsah kódu a potřebná paměť konstantní, ale zpravidla podstatně větší.

Konstrukce **slučování (fúzování) míst nebo přechodů** nejsou hierarchizačními konstrukcemi v pravém slova smyslu; jsou to spíše technické prostředky, které ulehčují a zpřehledňují vazby mezi jednotlivými stránkami hierarchické Petriho sítě. Jedná se o to, aby jeden a tentýž reálný prvek (místo nebo přechod) mohl být podle potřeby a pro pohodlí projektanta zobrazen současně na různých stránkách hierarchické sítě (a případně i také několikrát na téže stránce). Představujeme si, že všechny takto zobrazené fiktivní prvky fúzují v jediný reálný prvek.

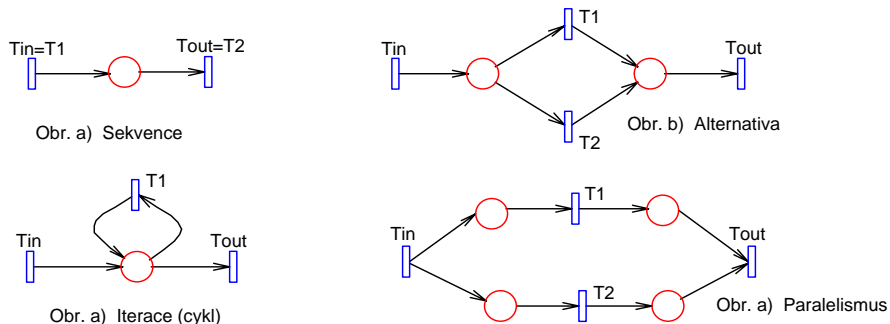
V případě slučování míst musí mít všechna místa, patřící do téže množiny fúzovaných míst, vždy totéž značení, tj. přibude-li token do jednoho z fúzovaných míst, přibude i do všech ostatních míst a je-li odebrán token z jednoho místa je současně odebrán i ze všech ostatních míst fúzované množiny. Fakticky se totiž jedná o místo jediné.

V případě slučování přechodů musí všechny fúzované přechody být vždy současně proveditelné nebo neproveditelné a je-li proveden kterýkoliv z fúzovaných přechodů znamená to, že jsou provedeny všechny přechody fúzované množiny. Z obsahového hlediska se totiž jedná o přechod jediné.

Příklad 1.8.1:

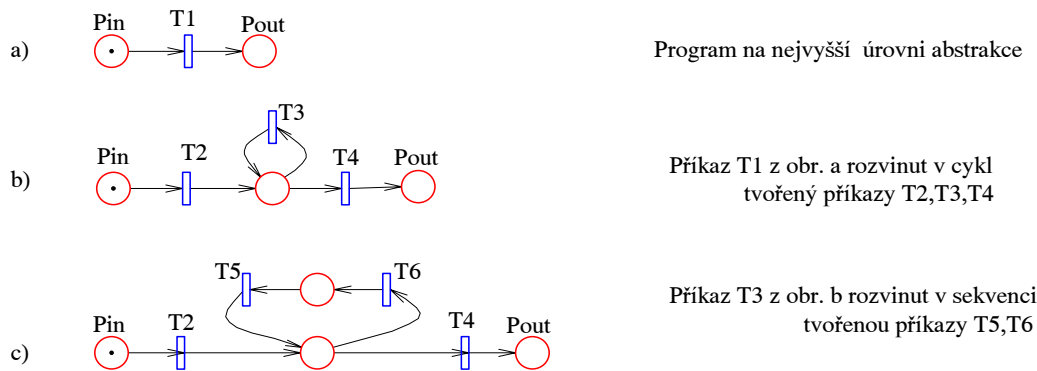
Na obrázcích 1.8.1-1.8.2 je ukázáno jak lze pomocí hierarchických Petriho sítí zobrazovat strukturované paralelní programy.

Na obr.1.8.1 jsou zobrazeny základní strukturní vzory paralelního programování: sekvenční řazení příkazů T1 a T2, alternativní provedení příkazů T1 a T2, iterace (cyklické provádění) příkazu T1 a paralelní provádění provádění příkazů T1 a T2.

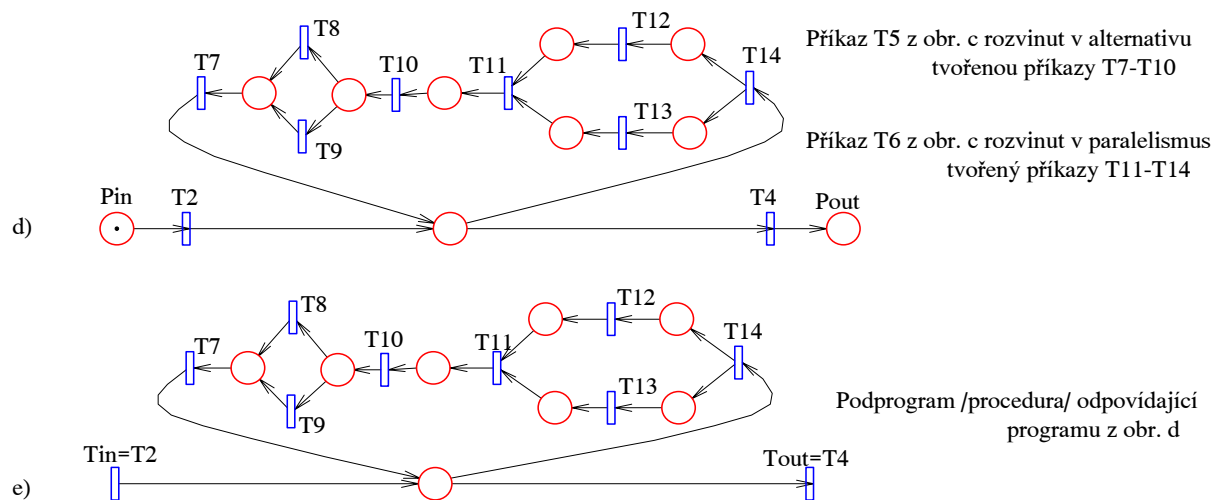


Obr.1.8.1 Základní programové konstrukty

Na obr.1.8.2 je předvedeno postupné rozvíjení paralelního programu počínaje obrázkem a) a konče obrázkem d), a to pouze s využitím konstrukcí zobrazených na obr.1.8.1. Použitým hierarchizačním konstruktem je substituce přechodů.



Obr.1.8.2.a-c Návrh paralelních programů metodou shora dolů



br.1.8.2.d-e Návrh paralelních programů metodou shora dolů

1.9. Objektové Petriho síť (Object-Oriented PN)

Petriho síť vysoké úrovně (HLPN - High-level Petri Nets), tj. Petriho síť, které jsou současně barevné i hierarchické, mají řadu cenných vlastností:

- jsou intuitivně velmi dobře srozumitelné,
- mají názornou grafickou reprezentaci,
- mohou sloužit jako specifikační jazyk (programovací jazyk vysoké úrovně),
- jsou exaktně definovány a mohou být traktovány jako formální matematická teorie.

Objektově-orientované přístupy představují zavedenou metodologii tvorby programů, která:

- výrazně ulehčuje a urychluje vývoj kvalitního softwaru,
- podstatně zlepšuje udržovatelnost, modifikovatelnost a opětné užití již vytvořeného softwaru.

Objektově-orientovaná metodologie byla vytvořena z pragmatických důvodů a její hlavní nevýhodou je neexistence exaktního formálního základu.

V posledních letech sílí snaha o integraci obou dvou metodologií (Petriho sítí a objektově-orientované přístupů). Nejjednodušší a nejvíce se nabízející cestu této integrace (tzv. "*Objects inside Petri net*" paradigma) lze zhruba charakterizovat takto:

- Tokeny kolující v Petriho síti jsou považovány za instance objektových tříd popsaných v nějakém objektově orientovaném programovacím jazyku.
- Přechody Petriho sítě představují metody aplikované na objekty reprezentované vstupujícími tokeny.
- Provedení přechodu nemusí nutně znamenat zničení vstupních a stvoření výstupních tokenů. Většinou tokeny skrze přechody pouze procházejí s tím, že metoda představovaná přechodem mění hodnoty některých jejich atributů. Provedení přechodu může také znamenat vytvoření nového objektu nebo zničení starého.
- Hrany Petriho sítě popisují možné toky objektů v systému.
- Struktura Petriho sítě vyjadřuje řídicí strukturu modelovaného systému, zatímco hierarchie tokenových typů vyjadřuje datovou strukturu modelovaného systému.
- Systém může být modelován jedinou rozsáhlou Petriho sítí a nebo může být hierarchicky strukturován.

Jiný způsob spojení objekt.-orientované metodologie s formalismem Petriho sítí je založen na modelování vnitřního chování objektů Petriho sítěmi (tzv. "*Petri net inside objects*" paradigma). Tento způsob vychází z následujících myšlenek:

- Každý objekt systému má svou vlastní Petriho síť. Značení sítě reprezentuje vnitřní stav objektu.
- Přechody sítě reprezentují provádění metod objektem. Některé přechody reprezentují vnitřní spontánní chování objektů, zatímco jiné nabízejí služby jiným objektům. Petriho síť objektu tak modeluje: dostupnost metod pro objekt, možné posloupnosti provádění metod, možná paralelní provádění metod.
- Komunikace mezi objekty (komunikační protokol) je rovněž modelována pomocí Petriho sítě. Aparát Petriho sítě je tak použit jak k modelování vnitřního chování objektů, tak i k modelování meziobjektové komunikace.

Oba výše charakterizované přístupy "*objects inside Petri net*" and "*Petri net inside objects*" jsou spíše objektově založené ("object-based") než objektově orientované ("object-oriented"). Vlastnosti dědičnosti, polymorfismu a dynamického vytváření objektů nejsou zde zpravidla uvažovány. Tyto vlastnosti však mohou být zakomponovány do formalismu "*object-based*" Petriho sítí, dovolíme-li, aby tokeny mohly představovat nejenom datové hodnoty rozmanitých typů, ale také odkazy (směrníky, ukazatele) na jiné objekty, tj. na jiné síť.

2. P/T síť a jejich vlastnosti

V této kapitole jsou formálně definovány P/T Petriho síť a exaktním způsobem studovány jejich vlastnosti. Oproti charakteristice z kap.1.3. je pojem P/T síť modifikován ve dvou bodech:

1. dovoluje se používání inhibičních hran,
2. neuvažují se kapacity míst (tj. v místech se mohou nacházet libovolné počty tokenů).

Pokud dále v této kapitole hovoříme prostě o Petriho sítích, máme tu vždy na mysli P/T síť s uvedenými modifikacemi.

Prvá modifikace (zobecnění) je motivována tím, že zavedení inhibičních hran podstatně rozšiřuje modelovací možnosti Petriho sítí.

Druhá modifikace (specializace) je motivována tím, že užívání kapacitních omezení míst poněkud komplikují analýzu Petriho sítí a přitom nikterak nerozšiřují jejich modelovací schopnost.

2.1. Struktura a dynamika P/T sítí

Statickou strukturu Petriho sítě popíšeme na třech úrovních abstrakce jako tzv.:

- strukturu Petriho sítě,
- systém Petriho sítě,
- parametrizovaný systém Petriho sítí.

Bude-li však z kontextu zřejmé na jaké abstrakční úrovni se pohybujeme, pak budeme jednoduše hovořit o Petriho sítí.

Definice 2.1.1:

Struktura Petriho sítě (PN-struktura) je pětice $\langle P, T, I, O, H \rangle$, kde

- P je konečná *množina míst*,
- T je konečná *množina přechodů*, $T \cap P = \emptyset$,
- I, O, H jsou zobrazení typu $T \rightarrow P_{MS}$, po řadě tzv. *vstupní funkce*, *výstupní funkce* a *vstupní inhibiční funkce*.
Připomeňme, že P_{MS} je množina všech multimnožin nad množinou P .

Poznámky 2.1.1:

1. PN- strukturu zobrazujeme diagramem, ve kterém místa znázorňujeme kružnicemi, přechody úsečkami nebo obdélníky, vstupní funkce hranami mířícími do přechodů (přitom orientované hrany zobrazující vstupní inhibiční funkci jsou opatřeny malým kroužkem namísto šipkou) a výstupní funkce hranami mířícími z přechodů.
2. Často je uvažován speciální případ, kdy inhibiční funkce přiřazuje všem přechodům $t \in T$ prázdnou multimnožinu z P_{MS} . V tomto případě není třeba inhibiční funkci H v definici struktury PN vůbec uvádět.
3. Dále se budeme používat následující označení:
 - $I(t)$ multimnožina vstupních míst přechodu t
 - $O(t)$ multimnožina výstupních míst přechodu t
 - $H(t)$ multimnožina inhibičních míst přechodu t
 - $I(t,p)$ násobnost prvku p v multimnožině $I(t)$, násobnost hrany z místa p do přechodu t
 - $O(t,p)$... násobnost prvku p v multimnožině $O(t)$, násobnost hrany z přechodu t do místa p
 - $H(t,p)$... násobnost prvku p v multimnožině $H(t)$, násobnost inhibiční hrany z místa p do přechodu t
 - ${}^*t = \{p \in P: I(t,p) > 0\}$ množina vstupních míst přechodu t
 - $t^* = \{p \in P: O(t,p) > 0\}$... množina výstupních míst přechodu t
 - ${}^{\circ}t = \{p \in P: H(t,p) > 0\}$... množ. vstupn. inhibičních míst přechodu t

- $\bullet_p = \{t \in T: O(t,p) > 0\}$... množina vstupních přechodů místa p
- $\bullet^p = \{t \in T: I(t,p) > 0\}$ množina výstupních přechodů místa p

Poznamenejme, že množiny \bullet_t , \bullet^t , \bullet_t stejně tak jako množiny \bullet_p , \bullet^p nemusí být nutně disjunktní.

4. Vstupní, výstupní a inhibiční funkce I,O,H můžeme též pojímat jako matice **I,O,H** typu $|T| \times |P|$ s prvky $I(t,p)$, $O(t,p)$, $H(t,p)$. Často se používají také matice **Pre=I^T**, **Post=O^T**, **C=Post-Pre=O^T-I^T** typu $|P| \times |T|$ - tyto matice nazývané po řadě **zpětnou incidenční maticí** (backward incidence matrix), **dopřednou incidenční maticí** (forward incidence matrix) a **incidenční maticí** (incidence matrix) Petriho sítě.

Definice 2.1.2:

Systém Petriho sítě (PN-systém) je šestice $\langle P,T,I,O,H,M_0 \rangle$, kde

- $\langle P,T,I,O,H \rangle$ je struktura Petriho sítě (viz definice 2.1.1),
- M_0 je zobrazení typu $P \rightarrow N$, tzv. **počáteční značení**, kde $N = \{0,1,2,\dots\}$ je množina přirozených čísel.

Poznámky 2.1.2:

1. Zápisem $M_0(p)$ rozumíme přirozené číslo přiřazené místu p a říkáme, že místo p obsahuje $M_0(p)$ **tokenů**. Je-li počet tokenů přiřazený danému místu malý, zobrazujeme je tučnými tečkami uvnitř kružnice zobrazující dané místo. Je-li počet tokenů větší, vpisujeme do kružnice přímo číslo $M_0(p)$.
2. Je-li speciálně $M_0(p)=0$ pro všechna $p \in P$, pak PN systém představuje pouze PN strukturu.
3. Obecně symbolem M označujeme jakékoliv zobrazení $P \rightarrow N$ a symbolem $M(p)$ přirozené číslo přiřazené tímto zobrazením místu $p \in P$. Značení sítě lze také pojímat jako následující multimnožinu nad množinou míst:

$$\sum_{p \in P} M(p) \cdot p = M(p_1) \cdot p_1 + M(p_2) \cdot p_2 + \dots + M(p_{|P|}) \cdot p_{|P|}$$

Definice 2.1.3:

Parametrizovaný systém Petriho sítě (parametrizovaný PN-systém) je osmice

$\langle P,T,I,O,H,PAR,COND,MP \rangle$, kde

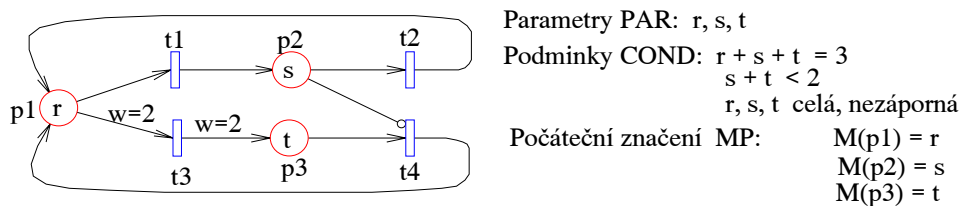
- $\langle P,T,I,O,H \rangle$ je struktura Petriho sítě (viz definice 2.1.1),
- PAR je množina parametrů,
- COND je množina podmínek omezujících obory, které mohou probíhat hodnoty parametrů,
- MP je zobrazení typu $P \rightarrow N \cup PAR$, tzv. **počáteční parametrické značení**, přiřazující každému místu buď přirozené číslo nebo parametr, jehož hodnota probíhá nějakou podmnožinu množiny přirozených čísel (podmnožinu vymezenou predikáty z PRED).

Poznámky 2.1.3:

1. Je-li speciálně množina parametrů prázdná ($PAR = \emptyset$), pak definování COND nemá smyslu a MP se redukuje na M_0 . Parametrizovaný systém pak představuje pouhý systém Petriho sítě.
2. Parametrizovaný systém je vlastně třída systémů jejichž počáteční značení vyhovuje daným podmínkám. Strukturu Petriho sítě můžeme pak pojímat jako nejobecnější parametrizovaný systém s prázdnou množinou podmínek (všechna počáteční značení jsou možná).

Příklad 2.1.1:

Na obr.2.1.1 je zobrazen jednoduchý parametrizovaný PN systém. Představuje třídu PN systémů s počátečními značeními: (3,0,0), (2,1,0), (2,0,1). Odmyslíme-li si počáteční značení, dostáváme PN strukturu. V tab.2.1.1 jsou zobrazeny vstupní, výstupní a vstupní inhibiční funkce I,O,H ve tvaru matic $I(t,p)$, $O(t,p)$, $H(t,p)$. Z tabulek např. vyplývá: $I(t_3) = 2 \cdot p_1$, $O(t_3) = 1 \cdot p_2 + 1 \cdot p_3$, $H(t_4) = 1 \cdot p_2$



Obr.2.1.1 Parametrizovaný PN systém

| I | p1 | p2 | p3 |
|----|----|----|----|
| t1 | 1 | 0 | 0 |
| t2 | 0 | 1 | 0 |
| t3 | 2 | 0 | 0 |
| t4 | 0 | 0 | 1 |

| O | p1 | p2 | p3 |
|----|----|----|----|
| t1 | 0 | 1 | 0 |
| t2 | 1 | 0 | 0 |
| t3 | 0 | 0 | 2 |
| t4 | 1 | 0 | 0 |

| H | p1 | p2 | p3 |
|----|----|----|----|
| t1 | 0 | 0 | 0 |
| t2 | 0 | 0 | 0 |
| t3 | 0 | 0 | 0 |
| t4 | 0 | 1 | 0 |

Tab.2.1.1 Vstupní, výstupní a vstupní inhibiční funkce

Až dosud jsme se zabývali popisem statiky Petriho sítí. Nyní budeme definovat dynamiku Petriho sítí, tj. stanovíme pravidla změny stavu sítě. Stav sítě je dán značením, tj. počty tokenů v jednotlivých místech. Změny stavu nastávají v důsledku provádění přechodů. V daném stavu sítě jsou jen některé přechody proveditelné. Dynamika Petriho sítí je tedy posána dvěma pravidly:

- pravidlo stanovící podmínky proveditelnosti přechodu (enabling rule),
- pravidlo popisující změnu stavu sítě po provedení přechodu (firing rule) .

Definice 2.1.4:

Přechod t je proveditelný (enabled) při značení M (M-enabled) jestliže platí

$$(\forall p \in {}^*t) [M(p) \geq I(t,p)] \wedge (\forall p \in {}^\circ t) [M(p) < H(t,p)].$$

Množinu přechodů proveditelných při značení M označíme E(M). Značení M, při kterém žádný přechod není proveditelný, tj. pro které platí $E(M) = \emptyset$, nazýváme *uzamčením* (deadlock).

Provedení přechodu t (transition firing), proveditelného při značení M, vede od značení M k značení M' takovému, že platí

$$M' = M + O(t) - I(t),$$

(M', M, O(t), I(t) jsou multimnožiny) neboli rozepsáno

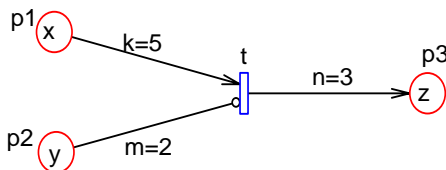
$$(\forall p \in P) [M'(p) = M(p) + O(t,p) - I(t,p)],$$

neboli ještě podrobněji rozepsáno

$$\begin{aligned} M'(p) &= M(p) && \text{pro } p \notin {}^*t \wedge p \notin {}^\circ t, \\ M'(p) &= M(p) - I(t,p) && \text{pro } p \in {}^*t \wedge p \notin {}^\circ t, \\ M'(p) &= M(p) + O(t,p) && \text{pro } p \notin {}^*t \wedge p \in {}^\circ t, \\ M'(p) &= M(p) - I(t,p) + O(t,p) && \text{pro } p \in {}^*t \wedge p \in {}^\circ t. \end{aligned}$$

Příklad 2.1.2:

Na obr.2.1.2 a v tab..2.1.2 je ilustrováno pravidlo proveditelnosti (enabling rule) a provedení přechodu (firing rule) pro čtyři různá výchozí značení.



Obr.2.1.2 Proveditelnost a provedení přechodu

| M | | | M' | | |
|---------|---------|---------|-----------|-----------|-----------|
| x=M(p1) | y=M(p2) | z=M(p3) | x'=M'(p1) | y'=M'(p2) | z'=M'(p3) |
| 6 | 1 | 3 | 1 | 1 | 6 |
| 5 | 2 | 2 | 0 | 2 | 5 |
| 2 | 3 | 1 | neodef. | | |
| 7 | 1 | 0 | | neodef. | |

Tab.2.1.2 Značení před a po provedení přechodu

Poznámky 2.1.4:

- Skutečnost, že provedení (proveditelného) přechodu t vede od značení M ke značení M' zapisujeme stručně

$$M[t > M' \text{ nebo také } M \xrightarrow{t} M'.$$

a říkáme, že značení M' je *bezprostředně dosažitelné* (directly reachable) ze značení M .

- Přirozeným zobecněním pojmu provedení přechodu je pojem provedení posloupnosti přechodů. Posloupnost přechodů

může startovat ze značení M právě tehdy, jestliže existuje posloupnost značení

$$M = M_{(1)}, M_{(2)}, \dots, M_{(k+1)} = M'$$

taková, že $i=1,2,\dots,k$ platí

$$M_{(i)}[t_{(i)} > M_{(i+1)}.$$

Tuto skutečnost označujeme $M[\sigma > M'$ (nebo také $M \xrightarrow{\sigma} M'$) a říkáme, že značení M' je *dosažitelné* (reachable) ze značení M .

- Většinou předpokládáme (není to však nutné), že PN-systém je navržen tak, že provedení dvou různých přechodů vede z daného stavu sítě k různým výsledným stavům, tj.

$$M[t_1 > M_1 \wedge M[t_2 > M_2 \Rightarrow M_1 \neq M_2.$$

Definice 2.1.5:

Množina dosažitelnosti (reachability set) PN-systému $\langle P, T, I, O, H, M_0 \rangle$ je množina $RS(M_0)$ definovaná induktivně takto:

- $M_0 \in RS(M_0)$
- $M \in RS(M_0) \wedge (\exists t \in T) \{M[t > M'\} \Rightarrow M' \in RS(M_0)$

Pokud je M_0 pevně dáno, budeme psát místo $RS(M_0)$ jednoduše RS . Vedle zápisu $RS(M_0)$ bývá také často používán zápis $[M_0 >$.

Graf dosažitelnosti (reachability graph) PN systému s množinou dosažitelnosti $RS(M_0)$ je hranově ohodnocený orientovaný graf $RG(M_0)$ s následujícími vlastnostmi:

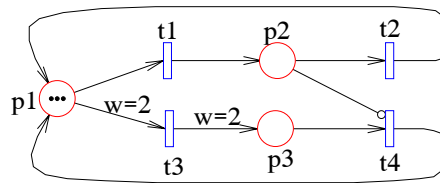
- $RS(M_0)$ je množina uzlů grafu,
- množina hran A grafu je definována takto:
 - $A \subseteq RS \times RS \times T$,
 - $\langle M_i, M_j, t \rangle \in A \Leftrightarrow M_i[t > M_j$,

M_0 je počáteční uzel grafu.

Pokud je M_0 pevně dáno, budeme psát místo $RG(M_0)$ jednoduše RG .

Příklad 2.1.3:

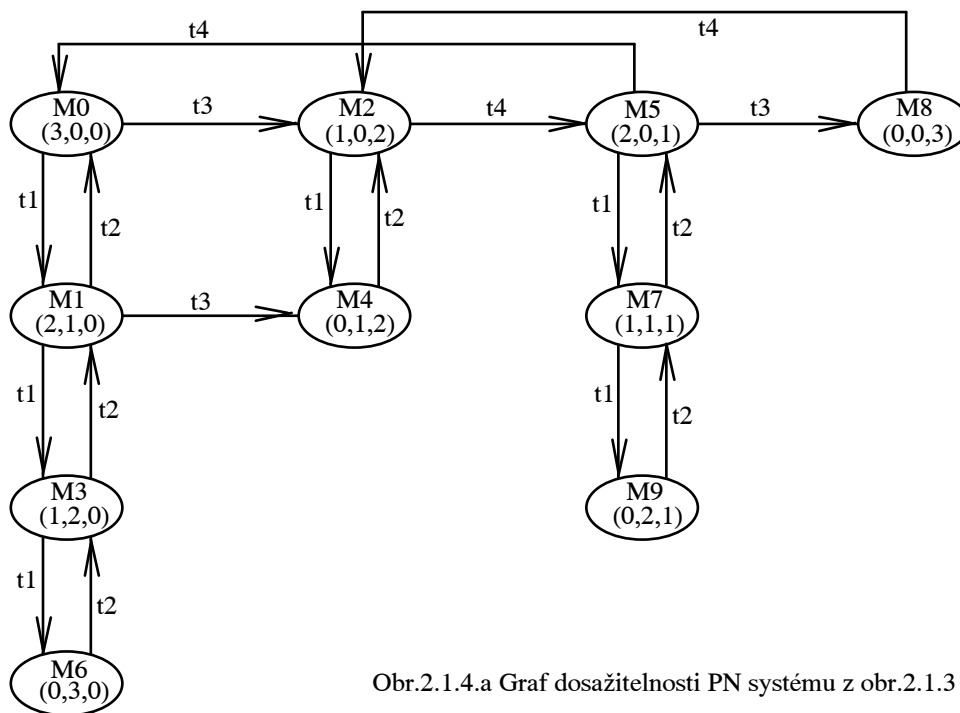
Na obr.2.1.3 je zobrazen jednoduchý PN-systém. V následující tabulce 2.1.3 je nalezena množina dosažitelných značení. Tabulku vyplňujeme po sloupcích zleva doprava. V posledním řádku této tabulky jsou uvedeny přechody, které jsou v tom či onom značení proveditelné a také následná značení ke kterým provedení přechodů vede. Na základě této tabulky je potom na obr.2.1.4 a zkonstruován graf dosažitelnosti Petriho sítě (PN-systému).



Obr.2.1.3 Jednoduchý PN systém

| | M0 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 |
|-------|------------------|----------------------------|------------------|------------------|--------|----------------------------|--------|------------------|--------|--------|
| p1 | 3 | 2 | 1 | 1 | 0 | 2 | 0 | 1 | 0 | 0 |
| p2 | 0 | 1 | 0 | 2 | 1 | 0 | 3 | 1 | 0 | 2 |
| p3 | 0 | 0 | 2 | 0 | 2 | 1 | 0 | 1 | 3 | 1 |
| t-> M | t1->M1 t3->M2 | t1->M3 t2->M0 t3->M4 | t1->M4 t4->M5 | t1->M6 t2->M1 | t2->M2 | t1->M7 t3->M8 t4->M0 | t2->M3 | t1->M9 t2->M5 | t4->M2 | t2->M7 |

Tab.2.1.3 Množina dosažitelných značení



Obr.2.1.4.a Graf dosažitelnosti PN systému z obr.2.1.3

Poznámky 2.1.5:

- Na množině všech možných značení dané Petriho sítě zavedeme následující binární relace:
 - $\rightarrow : M \rightarrow M' \Leftrightarrow_{df} (\exists t \in T) \{M[t>M'\}$
 - $\rightarrow^n : n$ -tá mocnina relace \rightarrow
 - $\rightarrow^+ :$ tranzitivní uzávěr relace \rightarrow
 - $\rightarrow^* :$ reflexivně tranzitivní uzávěr relace \rightarrow
 - $\leq : M \leq M' \Leftrightarrow_{df} (\forall p \in P) [M(p) \leq M'(p)]$... značení M' **pokrývá** (cover) značení M
 - $< : M < M' \Leftrightarrow_{df} M \leq M' \wedge (\exists p \in P) [M(p) < M'(p)]$... značení M' **ostře pokrývá** (strictly cover) značení M

2. Zápísem $RS(M)$ nebo také $[M>]$ označujeme množinu značení dosažitelných z libovolného (předem zadaného) značení M :

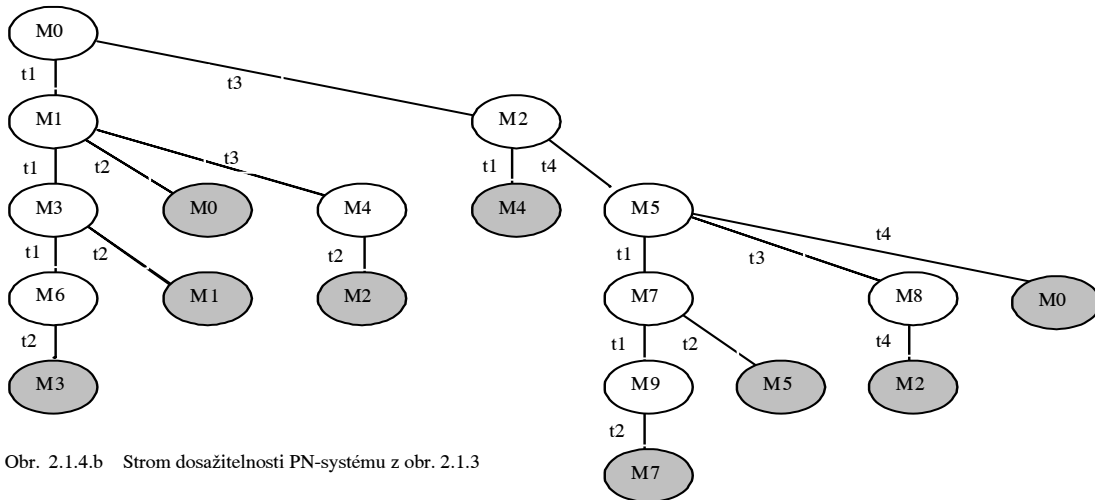
$$RS(M) = \{M' : (\exists \sigma \in T^*)[(M[\sigma > M'])]\}$$

Místo $RS(M)$ budeme často psát pouze RS , nehrozí-li nebezpečí nedorozumění.

3. Induktivní definice množiny dosažitelnosti $RS(M_0)$ představuje vlastně **algoritmus konstrukce** této množiny. Postupujeme v těchto krocích:

- (1) Položíme $X = \{M_0\}$.
- (2) Přidáme k množině X všechna značení, která jsou bezprostředně dosažitelná ze značení, která jsou již v X obsažena.
- (3) Zvětšila-li se množina X v kroku (2), pak se krok (2) opakuje. V opačném případě je $RS(M_0) = X$.

Grafickým zobrazením popsané konstrukce množiny dosažitelnosti je tzv. **strom dosažitelnosti** (reachability tree). Strom dosažitelnosti PN-systému z obr.2.1.3 je zobrazen na následujícím obrázku 2.1.4.b. Porovnejte grafy z obrázků 2.1.4.a-b zobrazující dosažitelné stavy $RS(M_0)$ (graf dosažitelnosti $RG(M_0)$) a strom dosažitelnosti $RT(M_0)$ a nalezněte jak lze jeden převést v druhý.



Obr. 2.1.4.b Strom dosažitelnosti PN-systému z obr. 2.1.3

4. Množina dosažitelných značení může být nekonečná (množina X se v každém kroku zvětšuje). To odpovídá situaci, kdy některá místa v síti mohou neomezeně zvětšovat počet tokenů v nich obsažených. V tomto případě je nekonečný i graf dosažitelnosti. Tento nekonečný graf může být reprezentován s jistou ztrátou informace, ale konečným způsobem, pomocí tzv. **stromu pokrytí** (coverability tree), který vznikne ze stromu dosažitelnosti vypuštěním těch značení (tj. odpovídajících vrcholů stromu), která pokrývají některé jiné značení.
6. Graf dosažitelnosti PN systému je reprezentací částečného konečného (nebo i nekonečného) automatu, kde
- o M_0 je počáteční stav,
 - o $RS = RS(M_0)$ je množina stavů,
 - o T je vstupní abeceda,
 - o **přechodová funkce** δ je částečné zobrazení $RS \times T \rightarrow RS$ definované takto:

$$\delta(M,t) = M' \Leftrightarrow M[t > M']$$

Zobecněná přechodová funkce δ^* je částečné zobrazení $RS \times T^* \rightarrow RS$ definované induktivně takto:

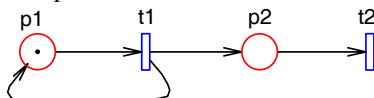
- o $\delta(M,e) = M$, kde e je prázdné slovo nad T ,
- o $\delta^*(M,t\sigma) = \delta^*(\delta(M,t),\sigma)$, kde σ je libovolné slovo nad T .

Slovo σ pro které je definována přechodová funkce $\delta^*(M_0,\sigma)$, nazýváme **výpočetní posloupností PN-systému** (firing sequence, occurrence sequence). Množina všech výpočetních posloupností charakterizuje chování PN systému, podobně jako množina všech dosažitelných značení PN systému charakterizuje množinu všech jeho stavů).

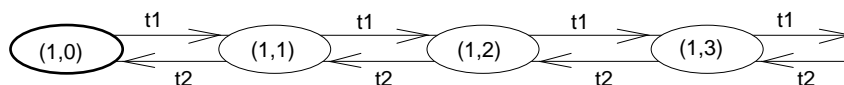
Poznamenejme, že je-li posloupnost přechodů σ výpočetní posloupností daného PN-systému, pak také každý počáteční úsek posloupnosti σ je jeho výpočetní posloupností. Každá výpočetní posloupnost PN-systému je v grafu dosažitelnosti jednoznačně zobrazena orientovanou cestou vycházející z počátečního vrcholu.

Příklad 2.1.4:

Na obr.2.1.5 je zobrazen PN systém s nekonečnou množinou dosažitelných značení. Na obr.2.1.6 je zobrazen nekonečný graf dosažitelnosti tohoto systému. Strom pokrytí sestává v tomto případě z jediného vrcholu (1, ω), kde ω představuje libovolné nezáporné celé číslo.



Obr.2.1.5 PN systém s nekonečnou množinou dosažitelnosti



Obr.2.1.6 Graf dosažitelnosti pro PN systém z obr.2.1.5

Příklad 2.1.5:

Na obr.2.1.7 jsou ukázány způsoby testování počtu tokenů v místě.



Obr.2.1.7 Modelování testů

V dalším zobecníme pojem proveditelnosti přechodu zavedením pojmu stupně proveditelnosti přechodu. To o nám umožní definovat na množině všech přechodů PN systému, proveditelných při daném značení M, tyto relace:

- přechody jsou v konfliktu
- přechody jsou souběžné
- přechody jsou v kauzální závislosti
- přechody se vzájemně vylučují

Definice 2.1.6:

Stupeň proveditelnosti (enabling degree) $ED(t, M)$ přechodu t při značení M, je přirozené číslo k udávající kolikrát je přechod t proveditelný při značení M.

Je-li množina vstupních míst přechodu je disjunkttní s množinou výstupních míst (tj. $t \cap t^* = \emptyset$) lze skutečnost $ED(t, M) = k$ vyjádřit formulí predikátové logiky:

$$\forall (p \in \bullet t) [M(p) \geq k \cdot I(t, p)] \wedge (\exists p \in \bullet t) [M(p) < (k+1) \cdot I(t, p)] \wedge (\forall p \in \circ t) [M(p) < H(t, p)].$$

Uvažujme dva různé přechody t_j, t_k PN-systému. Přechod t_j je v **efektivním konfliktu** (effective conflict) s přechodem t_k při značení M, jestliže platí

$$M[t_j > M'] \wedge (ED(t_k, M') < ED(t_k, M)).$$

Skutečnost, že přechod t_j je v efektivním konfliktu s přechodem t_k při značení M označujeme relačním zápisem $t_j EC(M) t_k$.

Přechody t_j, t_k jsou ve **strukturním konfliktu** (structural conflict), jestliže mají společné vstupní místo.

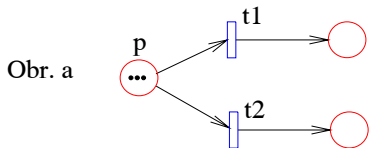
Poznámky 2.1.6:

1. Proveditelný přechod je přechod se stupněm proveditelnosti $n \geq 1$. Nproveditelný přechod má stupeň proveditelnosti rovný 0.
2. Konflikt je situace, při které provedení jednoho přechodu snižuje stupeň proveditelnosti druhého přechodu. Konflikt (efektivní) je *symetrický*, jestliže provedení kteréhokoliv ze dvou přechodů snižuje stupeň proveditelnosti druhého přechodu. Konflikt je *asymetrický*, jestliže provedení jednoho přechodu snižuje stupeň proveditelnosti druhého přechodu, ale nikoliv naopak.

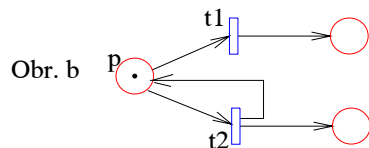
Příklad 2.1.6:

Na obr.2.1.8 jsou zobrazeny typické příklady (efektivních) konfliktů mezi dvěma přechody:

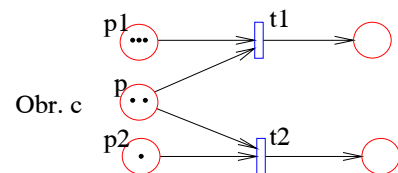
- a) symetrický konflikt
- b) asymetrický konflikt podmíněný strukturou sítě
- c) asymetrický konflikt podmíněný značením sítě (např. pro značení $M: 2 \cdot p_1 + 3 \cdot p + 1 \cdot p_2$ se stává konflikt symetrickým)



$ED(t_1, M) = 3, ED(t_2) = 3$
 $t_1 \in EC(M) \ t_2$ a současně $t_2 \in EC(M) \ t_1$
 symetrický konflikt
 provedení t_2 snižuje stupeň proveditelnosti t_1
 provedení t_1 snižuje stupeň proveditelnosti t_2



$ED(t_1, M) = 1, ED(t_2) = \text{inf}$
 $t_1 \in EC(M) \ t_2$, ale nikoliv $t_2 \in EC(M) \ t_1$
 asymetrický konflikt: asymetrie v důsledku struktury
 provedení t_2 nesnižuje stupeň proveditelnosti t_1
 provedení t_1 znemožňuje provedení t_2



$ED(t_1, M) = 2, ED(t_2) = 1$
 $t_2 \in EC(M) \ t_1$, ale nikoliv $t_1 \in EC(M) \ t_2$
 asymetrický konflikt: asymetrie v důsledku značení
 provedení t_2 snižuje stupeň proveditelnosti t_1
 provedení t_1 nesnižuje stupeň proveditelnosti t_2

Definice 2.1.7:

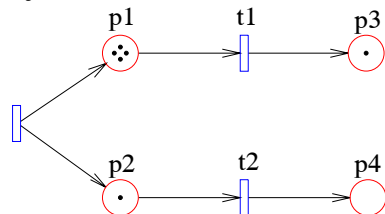
Přechody t_i, t_k nazýváme *souběžné* (concurrent) při značení M , jestliže
 $t_i, t_k \in E(M) \wedge \neg (t_i \in EC(M) t_k) \wedge \neg (t_k \in EC(M) t_i)$.

Poznámka 2.1.7:

Přechody t_i, t_k jsou při značení M souběžné, jestliže jsou oba proveditelné při značení M a přitom nejsou v konfliktu.

Příklad 2.1.7:

Na obr.2.1.9 je zobrazena Petriho síť se dvěma souběžnými aktivitami (přechody) t_1, t_2 .

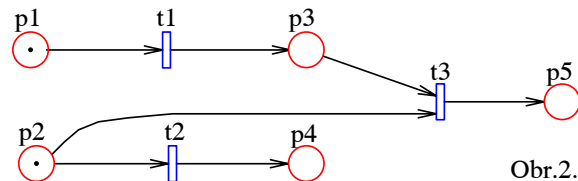


přechody t_1, t_2 jsou souběžné:
 - oba jsou proveditelné
 - nejsou v konfliktu

Obr.2.1.9 Souběžné činnosti

Příklad 2.1.8:

Na obr. 2.1.10 je zobrazena situace ve které je souběžnost přechodů t_1, t_2 kombinována s konfliktem přechodů t_2, t_3 . Při značení zobrazeném na obrázku jsou přechody t_1, t_2 souběžné, přesto však záleží na tom, který z nich bude proveden jako první. Takovou situaci nazýváme *konfúzí* (confusion). Bude-li totiž proveden jako první přechod t_1 , potom t_2 je v konfliktu s t_3 , kdežto bude-li jako první proveden přechod t_2 , potom žádný konflikt nevzniká.



Obr.2.1.10. Konfúze

Definice 2.1.8:

O přechodech t_i, t_k říkáme, že jsou v *kauzálním vztahu* (causal connection) při značení M a značíme $t_i CC(M)t_k$, jestliže

$$M[t_i > M' \Rightarrow ED(t_k, M') > ED(t_k, M).$$

Poznámka 2.1.8:

Přechod t_i je v kauzálním vztahu s přechodem t_k , jestliže provedení přechodu t_i zvyšuje stupeň proveditelnosti přechodu t_k .

V Petriho síti zobrazené na obr.2.1.10 provedení přechodu t_1 umožňuje provedení přechodu t_3 , tj. $t_1 CC(M)t_3$. Změníme-li značení na $M = p_1 + 2 \cdot p_2 + p_3$, pak přechod t_1 je rovněž v příčinném vztahu s přechodem t_3 , neboť zvyšuje stupeň proveditelnosti přechodu t_3 z 1 na 2. Naproti tomu při značení $M = p_1 + p_2 + p_3$ provedení přechodu t_1 nezvyšuje stupeň proveditelnosti t_3 , tj. $\neg t_1 CC(M)t_3$.

Definice 2.1.9:

O přechodech t_i, t_k říkáme, že se *vzájemně vylučují* (transitions are mutually exclusive), jestliže

$$\neg(\exists M \in RS(M_0))[t_i \in E(M) \wedge t_k \in E(M)].$$

O místech p_i, p_k říkáme, že se *vzájemně vylučují* (places are mutually exclusive), jestliže

$$\neg(\exists (M \in RS(M_0))[M(p_i) > 0 \wedge M(p_k) > 0].$$

Poznámka 2.1.8:

Dva přechody se vzájemně vylučují, jestliže neexistuje značení (dosažitelné z počátečního značení) při kterém by oba přechody byly současně proveditelné. Dvě místa se vzájemně vylučují, jestliže neexistuje značení (dosažitelné z počátečního značení) při kterém by obě místa měla současně nenulová značení. Relace vzájemné výlučnosti, na rozdíl od dříve diskutovaných relací (konfliktu, souběžnosti, konfúze a kauzálního vztahu), není relací vztahující se k jednotlivému konkrétnímu značení PN systému, ale relací vztahující se ke všem možným dosažitelným značením PN systému.

2.2. Vlastnosti a stavová analýza P/T sítí

V této kapitole jsou vyjmenovány nejdůležitější vlastnosti Petriho sítí (PN-struktur, PN-systémů, parametrizovaných PN-systémů), které jsou významné při studiu diskrétních systémů modelovaných Petriho sítěmi. Dále je pojednána základní metoda ověřování těchto vlastností - metoda stavové analýzy.

Definice 2.2.1:

Značení M' je dosažitelné (reachable) ze značení M , jestliže existuje posloupnost přechodů σ_M , která je proveditelná ve značení M a která převádí Petriho síť ze značení M do značení M' , tj.

$$(\exists \sigma_M \in T^*) [M \xrightarrow{\sigma_M} M'].$$

Značení M Petriho sítě $\langle P, T, I, O, H, M_0 \rangle$ se nazývá *vždy dosažitelným (home state marking)*, jestliže je dosažitelné z každého dosažitelného značení, tj. platí-li

$$(\forall M' \in RS(M_0)) [M \in RS(M')].$$

Připomeňme, že $RS(M)$ je množina všech značení dosažitelných ze značení M , tj.

$$RS(M) = \{M' : (\exists \sigma_M \in T^*) [M \xrightarrow{\sigma_M} M']\}.$$

PN-systém $\langle P, T, I, O, H, M_0 \rangle$ se nazývá *reversibilní (reversible)*, je-li počáteční značení vždy dosažitelné, tj. platí-li

$$(\forall M \in RS(M_0)) [M_0 \in RS(M)].$$

Poznámky 2.2.1:

1. V reversibilním systému je libovolné dosažitelné značení dosažitelné z libovolného dosažitelného značení:

$$(\forall M' \in RS(M)) [M \in RS(M')]$$
2. Typickým problémem analýzy Petriho sítí je *problém dosažitelnosti (reachability problem)*: je dán PN-systém a značení M a ptáme se, zda značení M je dosažitelné, tj. zda platí $M \in RS(M_0)$.
3. Zobecněním problému dosažitelnosti je *problém pokrytí (coverability problem)*: je dán PN-systém a značení M a ptáme se, zda existuje dosažitelné značení, které značení M pokrývá, tj. zda existuje značení $M' \in RS(M_0)$ takové, že $M \leq M'$.

Definice 2.2.2:

PN-systém $\langle P, T, I, O, H, M_0 \rangle$ se nazývá *systémem bez uzamčení (deadlock-free)*, jestliže z počátečního značení M_0 není dosažitelné žádné značení ve kterém není žádný přechod proveditelný, tj. platí-li

$$\neg (\exists M \in RS(M_0)) [E(M) = \emptyset].$$

Připomeňme, že $E(M)$ je množina všech přechodů proveditelných při značení M .

Poznámky 2.2.2:

1. Existence uzamčení je někdy žádoucí a jindy nežádoucí. Jestliže např. PN-model popisuje program, o kterém se předpokládá, že skončí, pak je nejen žádoucí, aby uzamčení v systému existovalo, ale také aby bylo vždy (tj. ze všech značení) dosažitelné. Jestliže PN-model popisuje program nepřetržitě běžící v reálném čase (např. operační systém), pak je nepřijatelné, aby obsahoval uzamčení.
2. Jestliže PN-systém obsahuje uzamčení, pak:

$$\text{PN-systém je reverzibilní} \Leftrightarrow \text{počáteční značení je uzamčením.}$$

Definice 2.2.3:

Přechod t je mrtvý při značení M , jestliže přechod není proveditelný v žádném značení dosažitelném ze značení M , tj. platí-li

$$(\forall M' \in [M >) [\neg M' [t >], \text{ neboli: } (\forall M' \in RS(M) [t \notin E(M)].$$

Přechod t je živý při značení M , jestliže není mrtvý v žádném značení dosažitelném ze značení M , tj. platí-li

$$(\forall M' \in [M >) (\exists M'' \in [M' >) [M'' [t >], \text{ neboli: } (\forall M' \in RS(M) (\exists M'' \in RS(M') [t \in E(M'')].$$

Přechod t je v daném PN systému mrtvý, je-li mrtvý při počátečním značení, tj. platí-li

$$(\forall M \in [M_0 >) [\neg M [t >], \text{ neboli: } (\forall M \in RS(M_0) [t \notin E(M)].$$

Přechod t je v daném PN systému živý, je-li živý při počátečním značení, tj. platí-li

$$(\forall M \in [M_0 >) (\exists M' \in [M >) [M' [t >], \text{ neboli: } (\forall M \in RS(M_0)) (\exists M' \in RS(M)) [t \in E(M')].$$

PN systém je mrtvý, jsou-li všechny jeho přechody mrtvé,

$$(\forall t \in T) (\forall M \in [M_0 >) [\neg M [t >], \text{ neboli: } (\forall t \in T) (\forall M \in RS(M_0) [t \notin E(M)].$$

PN systém je živý, jsou-li všechny jeho přechody živé, tj. platí-li

$$(\forall t \in T) (\forall M \in [M_0 >) (\exists M' \in [M >) [M' [t >], \text{ neboli: } (\forall t \in T) (\forall M \in RS(M_0)) (\exists M' \in RS(M)) [t \in E(M')].$$

Poznámky 2.2.3:

1. PN-systém je živý, jestliže při vývoji systému žádný přechod nikdy neztrácí možnost, že bude někdy v budoucnu znovu proveden.
2. Pro každou PN-strukturu lze definovat odpovídající PN-systém, který není živý. Např. volbou nulového počátečního značení, které každému místu přiřazuje nulový počet tokenů.
3. Obsahuje-li PN-systém aspoň jeden živý přechod, pak neobsahuje uzamčení. Na druhé straně neexistence uzamčení nezaručuje živost PN-systému.
4. Pojem živosti přechodu v PN-systému lze zobecnit: přechod t PN-systému **je k -živý** (k -live), jestliže z každého dosažitelného značení je dosažitelné značení ve kterém je přechod t alespoň k -proveditelný, tj. platí-li

$$(\forall M \in RS(M_0)) (\exists M' \in RS(M)) [ED(t, M') \geq k].$$
 Přechod je živý, je-li 1-živý.
5. Definici živosti PN-systému je možné formulovat také takto: z každého dosažitelného značení je možné spustit výpočetní posloupnost obsahující všechny přechody.

Definice 2.2.4:

Místo p PN-systému se nazývá k -omezené (k -bounded), jestliže pro každé dosažitelné značení je počet tokenů v tomto místě nanejvýše rovný k , tj.

$$(\forall M \in RS(M_0)) [M(p) \leq k].$$

PN-systém se nazývá k -omezený, jestliže všechna jeho místa jsou k -omezená, tj. platí-li

$$(\forall p \in P) (\forall M \in RS(M_0)) [M(p) \leq k].$$

PN-systémy se nazývají bezpečnými (safe), jestliže jsou 1-omezené.

Poznámky 2.2.4:

1. Důležitým důsledkem omezenosti PN-systému je konečnost jeho stavového prostoru (množiny všech dosažitelných značení). Je-li $|P|$ počet míst k -omezeného PN-systému, pak počet stavů (značení) nemůže přesáhnout číslo $(k+1)^{|P|}$.
2. Omezenost, živost a reversibilita jsou navzájem nezávislé (ortogonální) vlastnosti. Všech 8 kombinací těchto vlastností je možných - viz příklad 2.2.1.
3. V naprosté většině případů požadujeme omezenost PN-modelů. Bezpečnost požadujeme tehdy, kdy všechna místa modelují logické podmínky.

Definice 2.2.5:

Dvě místa p, p' se navzájem vylučují (are mutually exclusive) v PN-systému, jestliže nemohou současně obsahovat kladný (nenulový) počet tokenů, tj. jestliže platí:

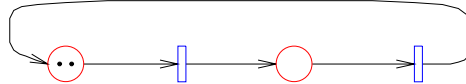
$$(\forall M \in RS(M_0)) [M(p).M(p')=0].$$

Dva přechody t, t' se navzájem vylučují (are mutually exclusive) v PN-systému, jestliže nikdy nemohou být současně proveditelné, tj. jestliže platí

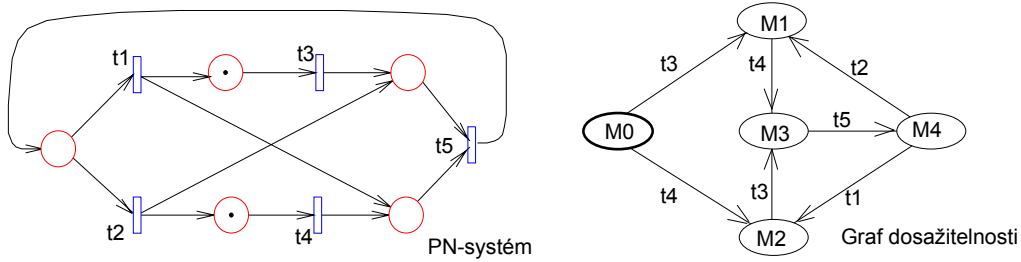
$$(\forall M \in RS(M_0)) [-(t \in E(M) \wedge t' \in E(M))].$$

Příklad 2.2.1:

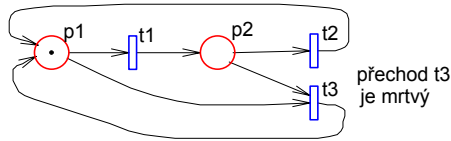
Na následujících obrázcích 2.2.1-8 jsou zobrazeny PN-systémy se všemi možnými kombinacemi vlastností omezenosti, živosti a reverzibility.



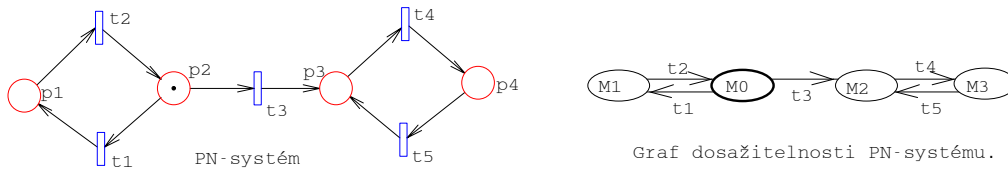
Obr.2.2.1. Omezený, živý a reverzibilní PN-systém



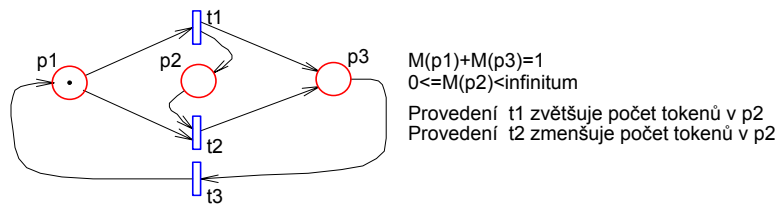
Obr.2.2.2. Omezený, živý a nereverzibilní PN-systém



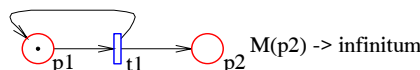
Obr.2.2.3. Omezený, neživý a reverzibilní PN-systém



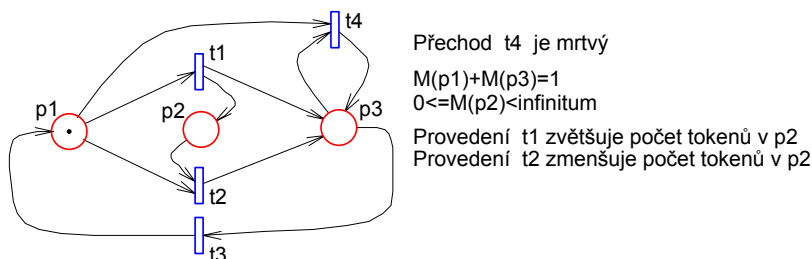
Obr.2.2.4. Omezený, neživý a nereverzibilní PN-systém



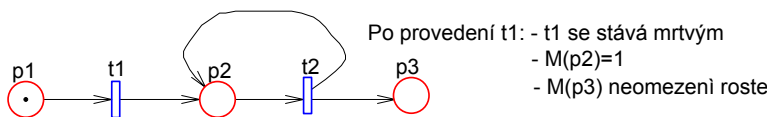
Obr.2.2.5. Neomezený, živý a reverzibilní PN-systém



Obr.2.2.6. Neomezený, živý a nereverzibilní PN-systém



Obr.2.2.7. Neomezený, neživý a reverzibilní PN-systém



Obr.2.2.8. Neomezený, neživý a nereverzibilní PN-systém

Metoda stavové analýzy PN-systému $\langle P, T, I, O, H, M_0 \rangle$ se zakládá na konstrukci množiny dosažitelných značení $RS(M_0)$ a grafu dosažitelnosti RG (viz definice 2.1.5 a poznámky 2.1.5). Algoritmus konstrukce grafu dosažitelnosti je jednoduchý a byl popsán v poznámce 2.1.5.3. V dalším budeme předpokládat, že algoritmus konstrukce grafu RG ukončí svou práci po konečném počtu kroků a že tedy množina $RS(M_0)$ i graf RG jsou konečné. V opačném případě (existuje nekonečná množina značení - stavů sítě) nemá většinou analyzovaný PN-systém praktický význam.

Máme-li popsán PN-systém grafem dosažitelnosti, redukuje se problém analýzy PN-systému na problém analýzy orientovaného grafu - jak ukazuje následující věta:

Věta 2.2.1:

- (1) Značení M' je v PN-systému dosažitelné ze značení M . \Leftrightarrow V grafu RG vede orientovaná cesta z uzlu M do uzlu M' .
- (2) Značení M je v PN-systému vždy dosažitelným stavem. \Leftrightarrow Z každého uzlu grafu RG vede orientovaná cesta do uzlu M .
- (3) PN-systém $\langle P, T, I, O, H, M_0 \rangle$ je reversibilní. \Leftrightarrow V grafu RG vede z každého uzlu orientovaná cesta do uzlu M_0
 \Leftrightarrow V RG grafu vede orientovaná cesta z každého uzlu do každého (tj. graf RG je silně souvislý).
- (4) PN-systém neobsahuje uzamčení. \Leftrightarrow V RG grafu neexistuje uzel, ze kterého by nevedla žádná hrana.
- (5) PN-systém je živý. \Leftrightarrow Pro všechny koncové (finální) silně souvislé komponenty RG grafu platí: každý přechod ohodnocuje aspoň jednu hranu komponenty.
- (6) PN-systém je omezený. \Leftrightarrow Graf RG je konečný.

Důkaz:

Všechna tvrzení vyplývají ihned z porovnání definic jednotlivých vlastností PN-systémů a definice RG -grafu.

Poznámky 2.2.5:

- 1. Metoda stavové analýzy, založená na konstrukci grafu dosažitelnosti, je v principu jednoduchá metoda, která umožňuje snadno zjistit všechny důležité vlastnosti PN-systémů. V mnoha praktických případech však bývá

množina $RS(M_0)$ natolik početná, že paměťová a časová náročnost konstrukce a následného studia RG-grafu přesahuje přijatelné meze.

- Podrobnější prohlídka grafu dosažitelnosti umožňuje také identifikovat vzájemnou vylučnost míst, vzájemnou vylučnost přechodů, značení při kterých vznikají efektivní konflikty a některá další fakta o situacích při evoluci PN-systému.

Příklad 2.2.2:

PN-systém z příkladu 2.1.2 zadaný diagramem z obr.2.1.2 a s grafem dosažitelnosti RG zobrazeným na obr.2.1.3 je podle věty 2.2.1:

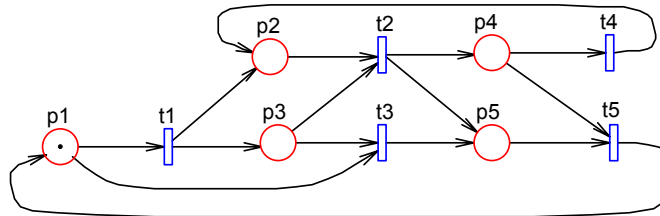
- reversibilní, neboť z každého uzlu grafu RG vede orientovaná cesta do uzlu M_0 a tedy také do každého jiného uzlu (graf RG je silně souvislý),
- bez uzamčení, neboť v RG neexistuje uzel, ze kterého by nevedla žádná hrana,
- živý, neboť graf RG je silně souvislý a každý přechod PN-systému je ohodnocením aspoň jedné hrany grafu RG,
- omezený, protože graf RG je konečný.

Z hodnot souřadnic dosažitelných značení (viz obr.2.1.3 nebo tab.2.1.2) vyplývá, že systém je:

- 3-omezený,
- konzervativní s invariantem značení: $m_1 + m_2 + m_3 = 3$.

Příklad 2.2.3:

Na obr.2.2.9 je diagramem zadán PN-systém, jehož vlastnosti jsou dále zjišťovány.



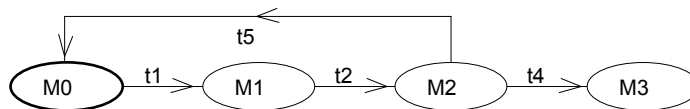
Obr.2.2.9. PN-systém s uzamčením

Nejdříve je v tab.2.2.1 nalezena množina dosažitelných značení a pak je na základě této tabulky konstruován graf dosažitelnosti - obr.2.2.9.

z

| | M0 | M1 | M2 | M3 |
|-------|---------|---------|--------------------|-----|
| M(p1) | 1 | 0 | 0 | 0 |
| M(p2) | 0 | 1 | 0 | 1 |
| M(p3) | 0 | 1 | 0 | 0 |
| M(p4) | 0 | 0 | 1 | 0 |
| M(p5) | 0 | 0 | 1 | 1 |
| t → M | t1 → M1 | t2 → M2 | t4 → M3 t5 → M0 | --- |

Tab.2.2.1



Obr.2.2.10.

Z grafu dosažitelnosti na obr. 2.2.10 vyplývají na základě věty 2.2.1 následující vlastnosti PN-systému:

- PN-systém není reversibilní, neboť ze značení M3 není dosažitelné počáteční značení M0 (RG není silně souvislý),
- PN-systém obsahuje uzamčení M3, neboť z uzlu M3 nevychází žádná hrana,
- PN-systém není živý, neboť obsahuje uzamčení.
- PN-systém je omezený, protože graf dosažitelnosti je konečný.
- Z tabulky 2.2.1 vyplývá, že PN-systém je 1-omezený, tj. bezpečný.

Definice 2.2.5:

Vlastnosti, které jsme zavedli pro PN-systémy můžeme vztahovat také na parametrizované PN-systémy: Parametrizovaný PN-systém $\langle P, T, I, O, H, PAR, COND, MP \rangle$ má danou vlastnost (je reverzibilní, živý, omezený, bezpečný,...), jestliže tuto vlastnost má každý PN-systém, který vznikne z parametrizovaného PN-systém dosazením konstant za parametry z PAR splňujícím podmínky COND.

Někdy zavedené vlastnosti PN-systémů vztahujeme také na PN-struktury a to ve dvojnásobném smyslu:

1) PN-struktura $\langle P, T, I, O, H \rangle$ má danou vlastnost, jestliže buď existuje počáteční značení M_0 takové, že PN-systém $\langle P, T, I, O, H, M_0 \rangle$ tuto vlastnost má, a nebo 2) PN-systém tuto vlastnost má pro všechna možná počáteční značení PN-systému.

Definice 2.2.6:

Strom dosažitelnosti (reachability tree) daného PN-systému (**RT graf**) je grafová reprezentace množiny $RS(M_0)$ daného PN-systému spolu s relací bezprostřední dosažitelnosti mezi prvky této množiny. Tato grafová reprezentace je typu strom a její konstrukce probíhá podle indukční definice množiny $RS(M_0)$ - viz definice 2.1.5. Konstrukce každé větve stromu je ukončena, jakmile koncový uzel větve má značení shodné se značením některého předchozího (dříve ustanoveného) uzlu (včetně kořenu M_0).

Graf dosažitelnosti (reachability graph) daného PN-systému (**RG graf**) je definován v definici 2.1.5.

RG graf vznikne z RT grafu odstraněním koncových zdvojených uzlů a jejich náhradou zpětnými hranami dvojnícům.

Strom pokrytí (coverability tree) daného PN-systému (**CT graf**) je zobecnění pojmu RT grafu, které umožňuje konečnou grafovou reprezentaci PN-systému a to i v případě, kdy PN-systém je neomezený a množina dosažitelných značení $RS(M_0)$ nekonečná. Toho se docílí rozkladem nekonečné množiny $RS(M_0)$ na konečný systém tříd takový, že v každé třídě se nacházejí všechna značení, která jsou pokryta (dominována) nějakým shora neomezeným značením (tj. značením vyznačujícím se tím, že v jednom nebo více místech se může vyskytovat neomezený počet tokenů). Přesná definice CT grafu je dána algoritmem jeho konstrukce - viz poznámky 2.2.6, bod 4.

Graf pokrytí (coverability graph) daného PN-systému (**CG graf**) je alternativní zobrazení stromu pokrytí, které získáme ze stromu pokrytí stejným způsobem jako graf dosažitelnosti ze stromu dosažitelnosti, tj. odstraněním dublujících koncových uzlů a jejich náhradou zpětnými hranami k uzlům dublovaným.

Poznámky 2.2.6:

1. RT graf a RG graf jsou dvě navzájem ekvivalentní reprezentace množiny dosažitelných značení $RS(M_0)$ daného PN-systému a relace dosažitelnosti mezi prvky této množiny. Tyto reprezentace jsou nekonečné v případě neomezených PN-systémů.
2. CT graf a CG graf jsou dvě navzájem ekvivalentní reprezentace množiny dosažitelných značení $RS(M_0)$ daného PN-systému a relace dosažitelnosti mezi prvky této množiny. Tyto reprezentace jsou vždy konečné (tj. i v případě neomezených PN-systémů).
3. CT a CG grafy jsou zobecněním RT a RG grafů. V případě omezených PN-systémů přechází CT a CG grafy automaticky v RT a RG grafy. Třídy rozkladu množiny $RS(M_0)$ jsou pak jednoprvkové.
4. Uvedeme konstruktivní definici stromu pokrytí (CT grafu). Nejprve připomeňme několik pojmů.

Značení PN-systému je dáno vektorem $M = (m_1, m_2, \dots, m_{|P|})$, kde $|P|$ je početnost množiny míst P.

Značení M dominuje (pokrývá) značení $N = (n_1, n_2, \dots, n_{|P|})$, jestliže

$$(\forall i) [m_i \geq n_i] \wedge (\exists i) [m_i > n_i].$$

Jestliže počet tokenů v místě p_i může nabývat libovolně velkých hodnot, pak klademe $m_i = \omega$.

Pro počítání se symbolem ω , symbolizující nekonečno, platí:

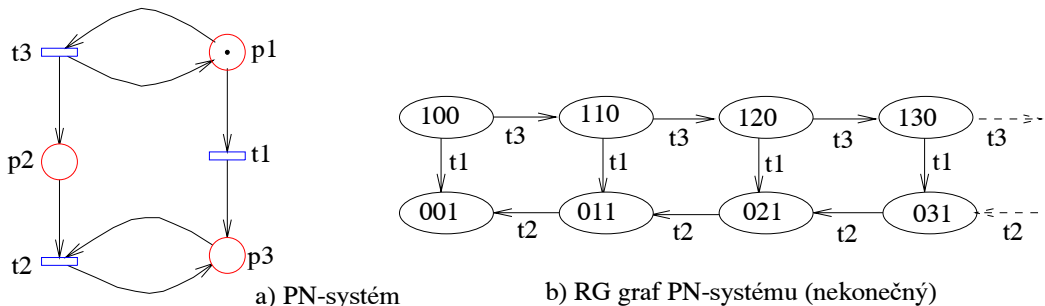
$$(\forall k)[\omega + k = \omega = \omega - k], (\forall k)[\omega > k], \text{ kde } k \text{ je celé nezáporné číslo.}$$

Konstrukce stromu pokrytí (CT grafu) daného PN-systému.

- (1) Kořenem stromu ustanovíme počáteční značení M_0 .
- (2) Pro každé nově ustanovené značení M provedeme
 - (a) Je-li $E(M) = \emptyset$ (tj. není-li ve značení M proveditelný žádný přechod, M je uzamčením), pak M je terminálním uzlem (listem) stromu.
 - (b) Je-li $E(M) \neq \emptyset$, pak pro všechna $t \in E(M)$ nalezni bezprostředně následující uzly uzlu M (se značeními M') podle těchto pravidel:
 - (i) Jestliže $m_i = \omega$ pro nějaké $i \in \{1, 2, \dots, |P|\}$, pak také $m'_i = \omega$.
 - (ii) Jestliže v dosud vytvořeném stromě na cestě od kořene (tj. uzlu M_0 a včetně uzlu M_0) do uzlu M' existuje uzel N , který je dominován (pokryt) uzlem M' , pak polož $m'_i = \omega$ pro všechna $i \in \{1, 2, \dots, |P|\}$ taková, že $m'_i > n_i$.
 - (iii) Jinak značení nového uzlu M' vypočti obvyklým způsobem podle pravidla provedení přechodu (firing rule), $M[t > M'$.
- (3) Rozšiřování stromu se ukončí, jakmile všechny koncové uzly jsou buď uzamčením (dead) a nebo se na své větvi stromu vyskytují již po druhé (dupl. - duplicated).

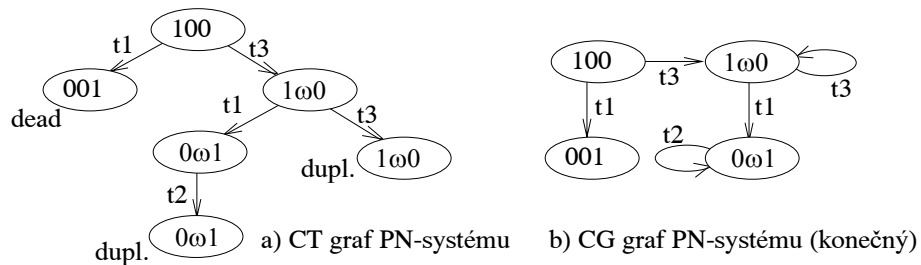
Příklad 2.2.4:

Budeme analyzovat vlastnosti PN-systému zobrazeného na obr. 2.2.11.a. Je zřejmé, že PN-systém je neomezený: opakovaným prováděním přechodu t_3 roste neomezeně počet tokenů v místě p_2 . Provedením přechodu t_1 se nevratně mění režim inkrementace počtu tokenů v místě p_2 v režim dekrementace. Opakované provádění přechodu t_2 dekrementuje počet tokenů v místě p_2 až na nulu, kdy dochází k uzamčení systému.



Obr. 2.2.11

Na obr. 2.2.11.b je nakreslen graf dosažitelnosti pro daný PN-systém. Uzly grafu jsou označeny vektorem (m_1, m_2, m_3) , zkráceně $m_1 m_2 m_3$, označující značení PN-systému (počty tokenů v místech p_1, p_2, p_3). Množina uzlů RG grafu (množina dosažitelných značení) je nekonečná. Na obr. 2.2.12 je zobrazen strom pokrytí (CT) a graf pokrytí (CG) téhož PN-systému. Počet uzlů těchto grafů je konečný. Symbol ω na místě souřadnice ve vektoru značení $M = (m_1, m_2, m_3)$ označuje, že v příslušném místě počet tokenů může nabývat libovolně velkých hodnot. Značení $(1, \omega, 0)$ představuje nekonečnou třídu značení $\{(1, 1, 0), (1, 2, 0), (1, 3, 0), \dots\}$ a značení $(0, \omega, 1)$ nekonečnou třídu značení $\{(0, 1, 1), (0, 2, 1), (0, 3, 1), \dots\}$ - porovnej obr. 2.2.11.b a 2.2.12.b. Dokonce i počáteční značení $(1, 0, 0)$ a koncové (deadlockové) značení $(0, 0, 1)$ lze zahrnout do první resp. druhé třídy.



Obr. 2.2.12

Analýzou CG grafu PN-systému lze stanovit vlastnosti PN systému:

1. CG graf obsahuje značení se symbolem $\omega \Rightarrow$ PN-systém je neomezený.
2. CG graf obsahuje uzel z něhož nevychází žádná hrana (se značením $(0,0,1)$) \Rightarrow PN-systém obsahuje uzamčení (deadlock).
3. PN-systém obsahuje uzamčení \Rightarrow PN-systém není živý.
4. Do počátečního uzlu nevede orientovaná cesta z žádného jiného uzlu CG grafu \Rightarrow PN-systém není reverzibilní.
5. Do uzlu zobrazujícího deadlock vede orientovaná cesta ze všech uzlů CG grafu \Rightarrow deadlockový stav (značení) PN-systému je vždy dosažitelný.
6. Všechna dosažitelná značení PN-systému jsou tvaru $(1,k,0)$ nebo $(0,k,1)$ $k=0,1,2, \dots$
7. V počátečním značení lze spustit sekvenci přechodů obsahující všechny přechody z množiny přechodů $T \Rightarrow$ PN-systém neobsahuje žádný mrtvý přechod.

2.3. Strukturní analýza P/T sítí

Odhlédneme-li u PN-systémů od počátečního značení, získáme bipartitní orientovaný multigraf, který popisuje statickou strukturu Petriho sítě (PN-strukturu). Studium PN-struktur je zajímavé z toho důvodu, že všechny vlastnosti Petriho sítí dokázané pouze na základě struktury jsou platné pro všechny PN-systémy, které lze získat z PN-struktury přidáním libovolného počátečního značení.

K strukturní analýze se uchylujeme zejména v případech, kdy selhává stavová analýza Petriho sítí založená na pojmu grafu dosažitelnosti. K takovému selhání dochází ve dvou případech:

- PN-systém je neomezený a množina dosažitelných značení a tedy i graf dosažitelnosti jsou nekonečné.
- PN-systém je omezený, ale množina dosažitelných značení je tak početná, že analýza grafu dosažitelnosti je mimo možnosti současných počítačů (state explosion problem).

Metody strukturní analýzy lze rozdělit do dvou skupin:

- metody lineární algebry (pracují s maticovou reprezentací PN-struktur),
- grafové metody (pracují přímo s grafovým popisem Petriho sítí).
 - metody redukce PN-systémů
 - metody využívající vlastností zámků a pastí
 - studium vlastností speciálních typů PN-struktur (kap. 2.4)

.....

Algebraické metody strukturní analýzy

Definice 2.3.1:

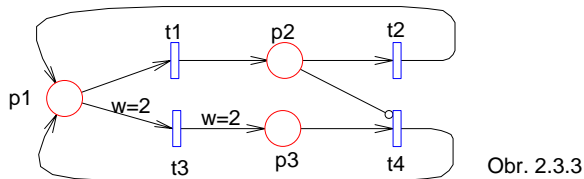
Incidenční matice (incidence matrix, change matrix) PN-struktury $\langle P, T, I, O, H \rangle$ je matice $C = \{C(p, t)\}$ typu $|P| \times |T|$

$$C = O^T - I^T,$$

kde $O = \{O(t, p)\}$, $I = \{I(t, p)\}$ jsou matice typu $|T| \times |P|$ reprezentující vstupní a výstupní funkci O, I .

Příklad 2.3.1:

Uvažujme PN-strukturu zobrazenou na obr.2.3.3.



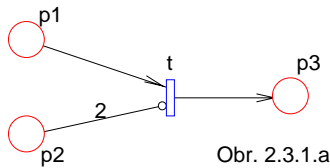
Incidenční matice Petriho sítě je

$$C = O^T - I^T = \begin{matrix} & & & t_1 & t_2 & t_3 & t_4 & & \\ & & & -1 & 1 & -2 & 1 & & p_1 \\ & & & 1 & -1 & 0 & 0 & & p_2 \\ & & & 0 & 0 & 2 & -1 & & p_3 \end{matrix}$$

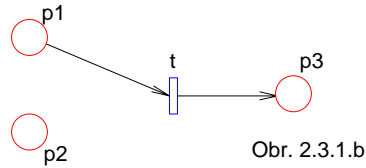
Řádky v maticích O resp. I , tj. sloupce v maticích O^T resp. I^T jsou tvořeny koeficienty multimnožin $O(t)$ resp. $I(t)$.

Poznámky 2.3.1:

- Prvek $C(p,t)$ matice C udává změnu počtu tokenů (kladnou, zápornou nebo nulovou) v místě p při provedení přechodu t .
 Vektor $C(.,t)$ (t -sloupec matice C) udává změnu počtu tokenů v jednotlivých místech sítě při provedení přechodu t .
 Vektor $C(p,.)$ (p -řádek matice C) udává změnu značení místa p při provedení jednotlivých přechodů sítě.
- Při zobrazení struktury Petriho sítě pomocí incidenční matice může docházet ke ztrátě informace a to v případě použití inhibičních nebo testovacích hran:
 - Existence inhibičních hran se nijak neprojeví na incidenční matici. Strukturně různé PN-systémy na obr.2.3.1.a,b mají stejnou incidenční matici.

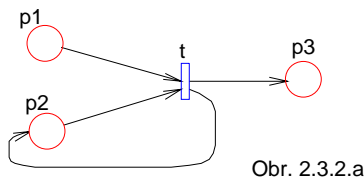


Obr. 2.3.1.a

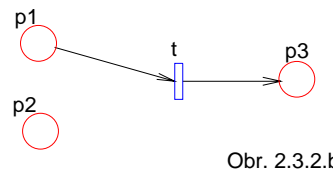


Obr. 2.3.1.b

- Na obr.2.3.2.a hrany (p_2,t) , (t,p_2) jsou tzv. **testovací hrany** - pomocí nich přechod t testuje přítomnost tokenů v místě p_2 aniž by měnil počet tokenů v tomto místě. Strukturně různé PN-systémy na obr.2.3.2.a,b mají stejnou incidenční matici - existence "testovacích" hran se nijak neprojeví na incidenční matici.



Obr. 2.3.2.a



Obr. 2.3.2.b

Petriho sítě ve kterých se nevyskytují jednoduché smyčky typu p_2-t zobrazené na obr.2.3.2.a, tj sítě pro něž $t \cap t^* = \emptyset$, se nazývají **čisté** (pure).

- Struktura čistých sítí bez inhibičních hran je incidenční maticí popsána úplně a jednoznačně.

V dalším textu této kapitoly předpokládáme, aniž bychom to vždy výslovně uváděli, že všechny uvažované Petriho sítě jsou jednoznačně popsány incidenční maticí !!!

Definice 2.3.2:

Uvažujme posloupnost (sekvenci) přechodů

$$\sigma = t_{(1)}t_{(2)} \dots t_{(k)}$$

chápanou jako slovo nad abecedou $T = \{t_1, t_2, \dots, t_{|T|}\}$, tj. $\sigma \in T^*$. **Charakteristickým vektorem** (transition count vector, Parikh vector) sekvence σ nazveme vektor

$$V_\sigma = (v_1, v_2, \dots, v_{|T|})^T,$$

kde v_i označuje počet výskytů přechodu t_i v sekvenci σ .

Charakteristický vektor V je pro značení M **realizovatelný** (possible), existuje-li sekvence σ spustitelná ze značení M taková, že $V = V_\sigma$.

Poznámky 2.3.2:

- Speciálně charakteristickým vektorem přechodu t_i je vektor $V_{t_i} = (v_1, v_2, \dots, v_{|T|})^T$, s jedinou nenulovou souřadnicí $v_i = 1$ a s $v_k = 0$ pro všechna $k \neq i$.
- Stejně sekvence mají přiřazeně též charakteristický vektor, ale témuž charakteristickému vektoru může odpovídat mnoho různých sekvencí z nichž některé mohou být realizovatelné a jiné nikoliv.

Věta 2.3.1:

Nechť σ je sekvence přechodů převádějící PN-systém ze značení M do značení M' , tj. $M[\sigma > M']$. Potom platí

$$M' = M + C \cdot V_{\sigma}, \tag{*}$$

kde značení M, M' chápeme jako matice typu $|P| \times 1$ (sloupcové vektory), C je incidenční matice typu $|P| \times |T|$ a charakteristický vektor V_{σ} jako matici typu $|T| \times 1$ (sloupcový vektor). Vztah (*) bývá někdy nazýván **fundamentální rovnicí** (fundamental equation, state equation, marking equation).

Důkaz:

Důkaz provedeme matematickou indukcí podle délky k posloupnosti σ .

- (1) Nechť $k=1$, tj. $\sigma=t$. V tomto případě vztah (*) platí, neboť:

$$M' = M + C(.,t) = M + C \cdot V_t = M + C \cdot V_{\sigma}.$$

- (2) Dokážeme, že z platnosti tvrzení pro sekvence délky k vyplývá platnost tvrzení pro sekvence délky $k+1$.

Uvažujme sekvenci $\tau = \sigma t$ délky $k+1$ převádějící PN-systém ze značení M do značení M' . Přitom sekvence σ délky k převádí systém ze značení M do značení M'' a sekvence t délky 1 ze značení M'' do značení M' . Platí

$$M' = M'' + C \cdot V_{\sigma} = M + C \cdot V_{\sigma} + C \cdot V_t = M + C \cdot (V_{\sigma} + V_t) = M + C \cdot V_{\tau}.$$

Poznámky 2.3.3:

1. Platnost fundamentální rovnice (*) je pouze nutnou podmínkou pro dosažitelnost $M[\sigma > M']$, nikoliv však podmínkou postačující. Viz příklad v posledním odstavci textu následujícího příkladu 2.3.2.
2. (Důsledek věty 2.3.1). Jsou-li dvě různé sekvence σ, τ se společným charakteristickým vektorem $V_{\sigma} = V_{\tau}$ spustitelné ze značení M , pak provedení obou sekvencí vede vždy ke stejnému konečnému značení M' .
3. (Difference lemma). Pro značení M, N a sekvenci přechodu σ platí:

$$M[\sigma > M'] \wedge N[\sigma > N'] \Rightarrow M' - M = N' - N$$

Důkaz:

4. (Monotonicity lemma). Sekvence přechodů σ je spustitelná při značení $M \Rightarrow$ Sekvence σ je spustitelná při všech značeních pokrývajících značení M (tj. je-li $N \geq M$, pak je σ je spustitelná také při značení N).

Důkaz:

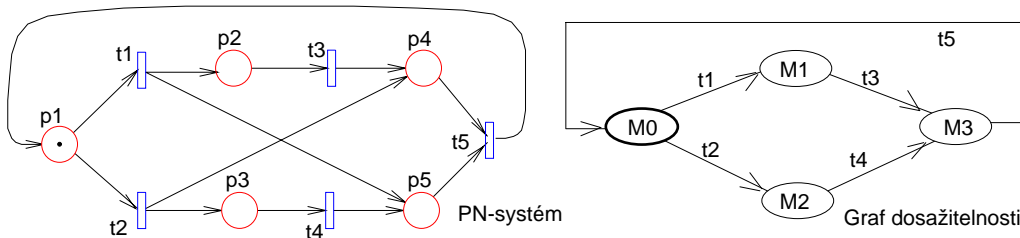
5. (Exchange lemma). Pro značení M, N, L a přechody u, v platí

$$M[u > N] \wedge N[v > L] \wedge u \cdot v = \emptyset \Rightarrow M[vu > L].$$

Důkaz:

Příklad 2.3.2:

Na obrázku 2.3.9 je zobrazen jednoduchý PN-systém a jeho graf dosažitelnosti (porovnej s PN-systémem a jeho grafem dosažitelnosti na obr.2.2.2). Algebraický výpočet grafu dosažitelnosti (na základě incidenční matice, tj. rovnice $M' = M + C(.,t)$) je proveden v tabulce 2.3.2.



Obr.2.3.9

| | t1 | t2 | t3 | t4 | t5 | M0 | M1 | M2 | M3 |
|----|---------------------|----|----|----|----|----------------|-------|-------|-------|
| p1 | -1 | -1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| p2 | 1 | 0 | -1 | 0 | 0 | 0 | 1 | 0 | 0 |
| p3 | 0 | 1 | 0 | -1 | 0 | 0 | 0 | 1 | 0 |
| p4 | 0 | 1 | 1 | 0 | -1 | 0 | 0 | 1 | 1 |
| p5 | 1 | 0 | 0 | 1 | -1 | 0 | 1 | 0 | 1 |
| | Incidenční matice C | | | | | t1→M1 t2→M2 | t3→M3 | t4→M3 | t5→M0 |

Tab.2.3.2

PN systém je bezpečný, živý a reversibilní. Pro všechna dosažitelná značení $\{M_0, M_1, M_2, M_3\}$ má rovnice $M - M_0 = C.X$ řešení (řešením jsou charakteristické vektory sekvencí, které převádějí PN-systém z počátečního stavu M_0 do stavu M).

Rovnice $M - M_0 = C.X$ má však často řešení i pro nedosažitelná značení. Tak např. pro značení $M=(0,1,1,0,0)^T$, které není dosažitelné, má fundamentální rovnice řešení $X=(1,1,0,0,1)^T$.

Definice 2.3.3:

P-invariantem struktury $\langle P, T, I, O \rangle$ (p-semiflow, p-invariant) nazýváme vektor $Y=(y_1, y_2, \dots, y_{|P|})^T$, $y_i \in N = \{0, 1, 2, \dots\}$, který anuluje zleva incidenční matici C, tj. vektor pro který platí $Y^T.C = 0^T$.

Množina $P_Y = \{p_i \in P: y_i > 0\}$ se nazývá *nosičem p-invariantu* Y. Je-li aspoň jedno $y_i > 0$, pak p-invariant nazýváme *netriviálním*. Triviální nulový invariant $Y=0$ nemá žádný praktický význam. Jsou-li všechna $y_i \in \{0, 1\}$, pak p-invariant nazýváme *binárním*.

Věta 2.3.2:

- Nechť Y_1, Y_2 jsou p-invarianty a k_1, k_2 celá nezáporná čísla. Potom platí:
- (1) Lineární kombinace $k_1 Y_1 + k_2 Y_2$ je rovněž p-invariantem a jeho nosičem je sjednocení nosičů p-invariantů Y_1, Y_2 .
 - (2) Je-li $Y_1 - Y_2 \geq 0$, pak $Y_1 - Y_2$ je rovněž p-invariantem.

Důkaz:

1. $(k_1 Y_1 + k_2 Y_2)^T.C = (k_1 Y_1^T + k_2 Y_2^T).C = k_1 Y_1^T C + k_2 Y_2^T C = k_1 0^T + k_2 0^T = 0^T$
2. $(Y_1 - Y_2)^T.C = (Y_1^T - Y_2^T).C = Y_1^T C - Y_2^T C = 0^T - 0^T = 0^T$

Poznámky 2.3.4:

1. *Petriho síť je pokryta p-invarianty*, jestliže každé místo patří do nosiče nějakého p-invariantu PN-struktury. Je-li Petriho síť pokryta p-invarianty, pak existuje p-invariant, který pokrývá celou síť (jeho nosičem je množina všech míst).
2. Množina všech p-invariantů dané PN-struktury představuje speciální druh lineárního (vektorového) prostoru ve kterém koeficienty lineárních kombinací mohou být pouze celá nezáporná čísla.
3. P-invariant $Y=(y_1, y_2, \dots, y_{|P|})^T$ je v *kanonickém tvaru* jestliže jeho souřadnice y_i jsou nesoudělné (jejich největší společný dělitel je 1). P-invariant $Y=(y_1, y_2, \dots, y_{|P|})^T$ je *minimální*, jestliže neexistuje žádný jiný p-invariant Y' (též dané struktury) pro který by platilo $Y' \leq Y \wedge Y' \neq Y$. Systém p-invariantů (dané struktury) je *úplný*, jestliže každý p-invariant (též dané struktury) je vyjadřitelný jako jejich lineární kombinace. Následuje popis *algoritmu výpočtu úplného systému minimálních p-invariantů*:

Vstup: Incidenční matice C typu $(m,n)=(|P|,|T|)$.

Výstup: Množina všech minimálních p-invariantů.

Stavová veličina: složená matice $[A|B]$

Vlastní algoritmus:

1. Inicializace: $A = C$, $B = I_m$, (I_m je jednotková matice řádu $m = |P|$).
2. Pro $j=1,2,\dots,n$ ($n=|T|$) prováděj:
 - 2.1. Přidej ke složené matici $[B|A]$ všechny řádky, které jsou lineárními kombinacemi s přirozenými koeficienty dvojic řádků z matice $[B|A]$ a které současně anulují j -tý sloupec matice A .
 - 2.2. Vyškrtej z matice $[B|A]$ všechny řádky s nenulovým prvkem v j -tém sloupci matice A .
3. Všechny řádky matice B převed' do kanonického tvaru (tj. vyděl každý řádek největším společným dělitelem všech jeho prvků).
4. Odstraň z matice B všechny neminimální p-invarianty (tj. vyškrtej z matice B všechny ty řádky, které pokrývají nějaký jiný řádek matice B).
5. Řádky matice B představují úplný soubor minimálních p-invariantů PN struktury zadané incidenční maticí C .

Příklad 2.3.7 ilustruje použití tohoto algoritmu při výpočtu p-invariantů konkrétní PN-struktury a také při výpočtu t-invariantů (s využitím věty 2.3.6).

$$4. Y \text{ je p-invariant} \Leftrightarrow (\forall t \in T) [\sum_{p \in t} Y(p) = \sum_{p \in t} Y(p)] .$$

Důkaz: ...

Věta 2.3.3:

Nechť Y je p-invariantem PN-struktury $\langle P, T, I, O \rangle$. Potom pro všechny PN-systémy $\langle P, T, I, O, M_0 \rangle$ platí:

$$(\forall M \in RS(M_0) [Y^T M = Y^T M_0].$$

Důkaz:

Podle věty 2.3.1 platí pro libovolná značení M, M' implikace

$$M[\sigma] M' \Rightarrow M' = M + CV_\sigma,$$

neboli speciálně také

$$M_0[\sigma] M \Rightarrow M = M_0 + CV_\sigma.$$

Vynásobíme-li poslední maticovou rovnici vektorem (řádkovou maticí) Y^T dostaneme vztah

$$Y^T M = Y^T M_0 + Y^T C V_\sigma.$$

Podle předpokladu věty je Y p-invariantem příslušné PN-struktury. Platí tedy $Y^T C = \mathbf{0}^T$ a poslední rovnice se redukuje na vztah $Y^T M = Y^T M_0$, který měl být dokázán.

Definice 2.3.4:

Vztah

$$Y^T M = Y^T M_0$$

platný podle věty 2.3.3 pro všechna značení M dosažitelná z počátečního značení M_0 , nazýváme **p-invariantem PN-systému** $\langle P, T, I, O, M_0 \rangle$ (conservation law). **P-invariant Y příslušné struktury** $\langle P, T, I, O \rangle$ má zde význam **váhového vektoru**: jeho souřadnice y_i jsou konstanty udávající váhu značení $m_i = M(p_i)$ jednotlivých míst p_i . Rozepsáním maticového zápisu do skalární podoby, dostáváme vztah

$$\sum y_i m_i = k, \text{ (sumace probíhá pro } i=1,2,\dots,|P| \text{)},$$

kde k je konstanta určená vztahem

$$k = Y^T M_0 = \sum y_i m_{0i}.$$

Podsystém systému $\langle P, T, I, O, M_0 \rangle$ indukovaný nosičem p-invariantu P_Y se nazývá **konzervativní komponentou** tohoto systému (spolu se všemi místy z nosiče P_Y patří do komponenty také všechny přechody, které jsou bezprostředními sousedy těchto míst a také všechny hrany spojující tato místa a přechody).

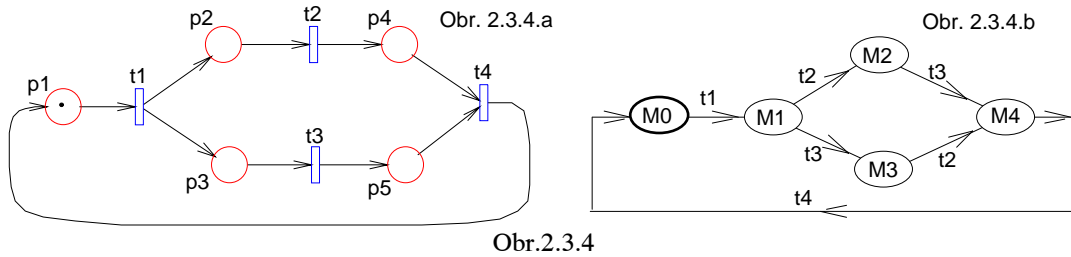
Systém $\langle P, T, I, O, H, M_0 \rangle$ je **konzervativní**, jestliže existuje P-invariant takový že $P_Y = P$. Systém je **striktně konzervativní**, jestliže navíc $y_i = 1$ pro všechna $i = 1, 2, \dots, |P|$.

Poznámka 2.3.5:

Je-li PN-systém konzervativní a je-li počáteční značení konečné, pak je PN-systém omezený.

Příklad 2.3.3:

Uvažujme PN-systém zobrazený na obr.2.3.4.a. V tabulce 2.3.1 je vypočtena množina dosažitelných značení a na obr. 2.3.4.b je zobrazen graf dosažitelnosti.



| | M0 | M1 | M2 | M3 | M4 |
|----------------|-----------------------|--|-----------------------|-----------------------|-----------------------|
| $m_1 = M(p_1)$ | 1 | 0 | 0 | 0 | 0 |
| $m_2 = M(p_2)$ | 0 | 1 | 0 | 1 | 0 |
| $m_3 = M(p_3)$ | 0 | 1 | 1 | 0 | 0 |
| $m_4 = M(p_4)$ | 0 | 0 | 1 | 0 | 1 |
| $m_5 = M(p_5)$ | 0 | 0 | 0 | 1 | 1 |
| | $t_1 \rightarrow M_1$ | $t_1 \rightarrow M_1$ $t_3 \rightarrow M_3$ | $t_3 \rightarrow M_4$ | $t_2 \rightarrow M_4$ | $t_4 \rightarrow M_0$ |

Tab.2.3.1. Množina dosažitelných značení

Nejdříve nalezneme strukturní p-invarianty PN-systému zobrazenému na obr.2.3.4.a. a to jako nezáporná řešení systému rovnic $Y^T \cdot C = 0$, tj. systému

$$\begin{pmatrix} y_1 & y_2 & y_3 & y_4 & y_5 \end{pmatrix} \cdot \begin{pmatrix} -1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 \end{pmatrix} = 0,$$

tj. systému

$$\begin{aligned}
 -y_1 + y_2 + y_3 &= 0, \\
 -y_2 + y_4 &= 0, \\
 -y_3 + y_5 &= 0, \\
 y_1 - y_4 - y_5 &= 0.
 \end{aligned}$$

Poslední rovnici netřeba uvažovat, protože je lineární kombinací prvních tří. Ze zbylých rovnic je ihned patrné, že jejich řešeními jsou vektory

$$\begin{aligned}
 Y_1 &= (1, 1, 0, 1, 0)^T, \\
 Y_2 &= (1, 0, 1, 0, 1)^T
 \end{aligned}$$

a všechna ostatní řešení jsou tvaru $k_1 Y_1 + k_2 Y_2$, $k_1, k_2 \in \{0, 1, 2, \dots\}$. Ze složených p-invariantů je nejvýznamnější $Y_3 = Y_1 + Y_2 = (2, 1, 1, 1, 1)^T$.

Všechny uvedené invarianty jsou v kanonickém tvaru. Invarianty Y_1, Y_2 jsou minimální a tvoří úplný systém minimálních invariantů (tj. každý jiný invariant lze vyjádřit jako jejich lineární kombinaci s celočíselnými nezápornými koeficienty). Invariant Y_3 minimální není, protože např. $Y_3 \geq Y_1$.

Vzhledem k tomu, že existuje p-invariant pokrývající celou množinu P ($P_{Y_3} = P = \{p_1, p_2, p_3, p_4, p_5\}$), je PN-systém konzervativní, ale nikoliv striktně.

Strukturním p-invariantům Y_1, Y_2, Y_3 odpovídají systémové p-invarianty

$$Y_1^T M = Y_1^T M_0, Y_2^T M = Y_2^T M_0, Y_3^T M = Y_3^T M_0,$$

které lze rozepsat do následujících rovnic

$$\begin{aligned} m_1 + m_2 + m_4 &= 1, \\ m_1 + m_3 + m_5 &= 1, \\ 2m_1 + m_2 + m_3 + m_4 + m_5 &= 2. \end{aligned}$$

Platnost uvedených rovnic pro všechna značení dosažitelná z počátečního značení M_0 lze ověřit z tabulky dosažitelných značení 2.3.1 a nebo - díky jednoduchosti zvoleného příkladu - nahlédnout přímo z obrázku 2.3.4.

Definice 2.3.5:

T-invariantem struktury $\langle P, T, I, O \rangle$ (t-semiflow, t-invariant) nazýváme vektor $X = (x_1, x_2, \dots, x_{|T|})^T$, $x_i \in N = \{0, 1, 2, \dots\}$, který anuluje zprava incidenční matici, tj. vektor pro který platí

$$C \cdot X = 0.$$

Množina $T_X = \{t_i \in T: x_i > 0\}$ se nazývá *nosičem t-invariantu X*. Je-li aspoň jedno $x_i > 0$, pak invariant nazýváme *netriviálním*, jsou-li všechna $x_i \in \{0, 1\}$, pak t-invariant nazýváme *binárním*.

Věta 2.3.4:

Nechť X_1, X_2 jsou t-invarianty a k_1, k_2 celá nezáporná čísla. Potom platí:

- (1) Lineární kombinace $k_1 X_1 + k_2 X_2$ je rovněž t-invariantem a jeho nosičem je sjednocení nosičů invariantů X_1, X_2 .
- (2) Je-li $X_1 - X_2 \geq 0$, pak $X_1 - X_2$ je rovněž t-invariantem.

Důkaz:

Obdobný k důkazu věty 2.3.2.

Poznámky 2.3.7:

1. *Petriho síť je pokryta t-invarianty*, jestliže každé přechod patří do nosiče nějakého t-invariantu PN-struktury. Je-li Petriho síť pokryta t-invarianty, pak existuje t-invariant, který pokrývá celou síť (jeho nosičem je množina všech přechodů).
2. Množina všech t-invariantů dané PN-struktury představuje druh lineárního (vektorového) prostoru ve kterém jsou přípustné pouze celočíselné a nezáporné lineární kombinace vektorů..
3. Pojmy kanonický tvar invariantu, minimální invariant, úplný systém invariantů jsou pro t-invarianty definovány stejným způsobem jako pro p-invarianty - viz Poznámky 2.3.....
4. Výpočet úplného systému minimálních t-invariantů lze převést na výpočet úplného systému minimálních p-invariantů duální PN-struktury - viz dále definice 2.3.7 a věta 2.3.6. Z této definice a věty vyplývá, že pro výpočet t-invariantů lze použít tehož algoritmu jako pro výpočet p-invariantů, (tj. algoritmu popsáno v poznámce 2.3.4) s jediným rozdílem: místo incidenční matice C použijeme její transpozici C^T (přesněji: v inicializačním kroku algoritmu klademe $A = C^T, B = I_n$, kde I_n je jednotková matice řádu $n = |T|$).
5. X je t-invariant $\Leftrightarrow (\forall p \in P) [\sum_{t \in \bullet p} X(t) = \sum_{t \in p \bullet} X(t)]$
Důkaz: ...

Věta 2.3.5:

Nechť X je t-invariantem PN-struktury $\langle P, T, I, O \rangle$. Potom existuje PN-systém $\langle P, T, I, O, M_0 \rangle$ takový že:

$$(\exists \sigma \in T^*) [M_0[\sigma] M_0 \wedge V_{\sigma} = X],$$

tj. vždy platí

$$(\exists M_0 \in \underline{M}) (\exists \sigma \in T^*) [M_0[\sigma] M_0 \wedge V_{\sigma} = X],$$

kde \underline{M} je množina všech možných značení dané struktury.

Důkaz:

Podle věty 2.3.1 platí pro libovolná značení M, M' implikace

$$M[\sigma \triangleright M' \Rightarrow M' = M + CV_\sigma,$$

neboli speciálně také

$$M_0[\sigma \triangleright M \Rightarrow M = M_0 + CV_\sigma. \quad (*)$$

Položíme-li $V_\sigma = X$, pak $CV_\sigma = CX = 0$ (neboť X je t-invariant) a tedy $M = M_0$.

Implikaci v (*) nelze obrátit (viz poznámky k větě 2.3.1) a tedy nikoliv jakákoliv posloupnost přechodů σ s daným charakteristickým vektorem $V_\sigma = X$ musí být proveditelná při jakémkoliv značení M_0 . S tím souvisí použití existenčních kvantifikátorů v tvrzení věty.

Definice 2.3.6:

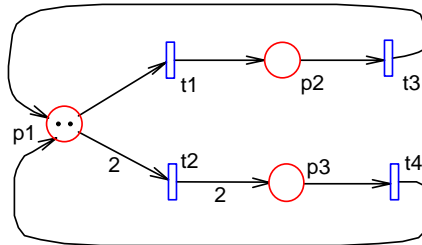
Posloupnost přechodů $\sigma \in T^*$ pro kterou platí $M_0[\sigma \triangleright M_0$ nazýváme *t-invariantem PN-systému* $\langle P, T, I, O, M_0 \rangle$ (reproduction law).

Podsystem systému $\langle P, T, I, O, M_0 \rangle$ indukovaný nosičem t-invariantu T_X se nazývá *repetiční komponentou* tohoto systému (spolu se všemi přechody z nosiče T_X patří do komponenty také všechna místa, které jsou bezprostředními sousedy těchto přechodů a také všechny hrany spojující tyto přechody a místa).

Systém $\langle P, T, I, O, H, M_0 \rangle$ je *repetiční (konzistentní)*, jestliže existuje t-invariant takový že $T_X = T$. Systém je *striktně repetiční*, jestliže navíc $x_i = 1$ pro všechna $i = 1, 2, \dots, |T|$.

Příklad 2.3.4:

Uvažujme PN-systém $\langle P, T, I, O, M_0 \rangle$ zobrazený na obr.2.3.5.



Obr. 2.3.5

Nejdříve nalezneme strukturní t-invarianty PN-systému zobrazenému na obr.2.3.5. Incidenční matice je

$$C = \begin{pmatrix} -1 & -2 & 1 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 2 & 0 & -1 \end{pmatrix}$$

a t-invarianty nalezneme jako nezáporné celočíselné řešení rovnice

$$C.X = 0,$$

tj. systému rovnic

$$\begin{aligned} -x_1 - 2x_2 + x_3 + x_4 &= 0 \\ x_1 - x_3 &= 0 \\ 2x_2 - x_4 &= 0 \end{aligned}$$

První rovnici netřeba uvažovat, protože je lineární kombinací dalších dvou. Ze zbylých rovnic je ihned patrné, že řešenými jsou vektory

$$\begin{aligned} X_1 &= (1, 0, 1, 0)^T, \\ X_2 &= (0, 1, 0, 2)^T \end{aligned}$$

a všechny jejich lineární kombinace $k_1 X_1 + k_2 X_2$, např.

$$\begin{aligned} X_3 &= 2X_1 = (2, 0, 2, 0)^T, \\ X_4 &= X_1 + X_2 = (1, 1, 1, 2)^T. \end{aligned}$$

Strukturní t-invarianty X_1, X_2 jsou minimální a množina $\{X_1, X_2\}$ tvoří úplný systém. Invarianty X_3, X_4 nejsou minimální. Invarianty X_1, X_2, X_4 jsou v kanonickém tvaru, invariant X_3 v kanonickém tvaru není.

Systémovými t-invarianty jsou např. následující posloupnosti přechodů:

- $t_1 t_3$ s charakteristickým vektorem X_1 (realizace strukturního invariantu X_1)

- $t_2t_4t_4$ s charakteristickým vektorem X_2 (realizace strukturního invariantu X_2)
- $t_1t_1t_3t_3, t_1t_3t_1t_3$ s charakteristickým vektorem X_3 (realizace strukt. invariantu X_3)
- $t_1t_3t_2t_4t_4, t_1t_2t_3t_4t_4, \dots$ s charakteristickým vektorem X_4 (realizace strukt. invariantu X_3)
-

Ze skutečnosti, že existuje t-invariant pokrývající celou množinu T (např. systémový invariant $t_1t_3t_2t_4t_4$, realizující strukturní invariant $X_4=(1,1,1,2)^T$, vyplývá, že PN-systém je repetiční (konzistentní), nikoliv však striktně repetiční.

Vzhledem k jednoduchosti uvažovaného PN-systému lze systémové t-invarianty odečítat přímo z obrázku 2.3.5.

Definice 2.3.7:

Předpokládejme PN-struktury bez inhibičních hran. Potom PN-struktura $\langle P', T', I', O' \rangle$ je *duální* k struktuře $\langle P, T, I, O \rangle$, jestliže platí:

$$P' = T \wedge T' = P \wedge I' = O \wedge O' = I$$

Poznámky 2.3.8:

1. Duální struktura vznikne z původní vzájemnou záměnou míst a přechodů a změnou orientace všech hran při zachování jejich násobností.
2. Z definice vyplývá: je-li $\langle P', T', I', O' \rangle$ duální k $\langle P, T, I, O \rangle$, pak také $\langle P, T, I, O \rangle$ je duální k $\langle P', T', I', O' \rangle$. Uvedené struktury jsou tedy duální navzájem.

Věta 2.3.6:

Nechť $\langle P', T', I', O' \rangle, \langle P, T, I, O \rangle$ jsou vzájemně duální sítě s incidenčními maticemi C', C . Potom platí:

1. $C' = C^T$,
2. p-invariant struktury $\langle P, T, I, O \rangle$ je t-invariantem struktury $\langle P', T', I', O' \rangle$,
3. t-invariant struktury $\langle P, T, I, O \rangle$ je p-invariantem struktury $\langle P', T', I', O' \rangle$.

Důkaz:

1. Plyne ihned z definice.
2. Platí:

$$Y^T \cdot C = \mathbf{0}^T \Leftrightarrow (Y^T \cdot C)^T = (\mathbf{0}^T)^T \Leftrightarrow C^T \cdot Y = \mathbf{0} \Leftrightarrow C' \cdot Y = \mathbf{0}.$$

Je-li tedy Y p-invariantem nečárkované struktury, pak je také t-invariantem čárkované struktury.

3. Platí:

$$C \cdot X = \mathbf{0} \Leftrightarrow (C \cdot X)^T = \mathbf{0}^T \Leftrightarrow X^T \cdot C^T = \mathbf{0}^T \Leftrightarrow X^T \cdot C' = \mathbf{0}^T.$$

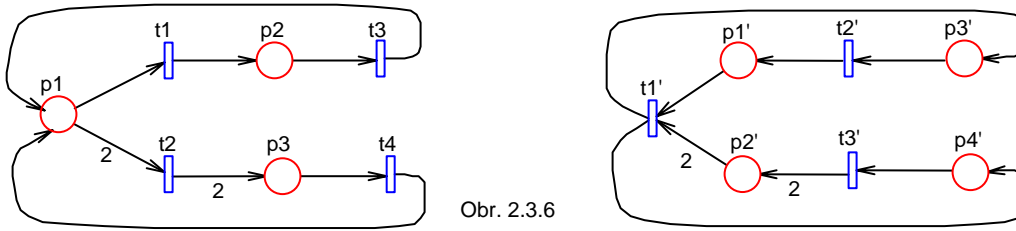
Je-li tedy X t-invariantem nečárkované struktury, pak je také t-invariantem čárkované struktury.

Důsledek:

Výpočet úplného systému minimálních t-invariantů dané PN-struktury může být proveden podle téhož algoritmu jako výpočet úplného systému minimálních p-invariantů (viz popis algoritmu - poznámka 2.3.4.3) s jediným rozdílem: místo s incidenční maticí C pracujeme s její transpozicí C^T (tj. popsany algoritmus startujeme nikoliv s $A=C$, ale s $A=C^T$). Výpočet je ilustrován v příkladě 2.3.7.

Příklad 2.3.5:

Na obr.2.3.6 je zobrazena dvojice navzájem duálních struktur.



Obr. 2.3.6

Dále jsou uvedeny incidenční matice obou struktur (je patrné, že $C' = C^T$) a p- a t-invarianty obou struktur (jak je patrné, p-invariant jedné struktury je t-invariantem druhé).

Nečárkovaná struktura:

$$C = \begin{pmatrix} -1 & -2 & 1 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 2 & 0 & -1 \end{pmatrix}$$

t-invarianty:

$$\begin{aligned} X_1 &= (1, 0, 1, 0)^T, \\ X_2 &= (0, 1, 0, 2)^T, \\ X_3 &= X_1 + X_2 = (1, 1, 1, 2)^T, \dots \end{aligned}$$

p-invarianty:

$$Y_1 = (1, 1, 1)^T, \dots$$

Čárkovaná struktura:

$$C' = \begin{pmatrix} -1 & 1 & 0 \\ -2 & 0 & 2 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{pmatrix}$$

p-invarianty:

$$\begin{aligned} Y_1' &= (1, 0, 1, 0)^T, \\ Y_2' &= (0, 1, 0, 2)^T, \\ Y_3' &= Y_1' + Y_2' = (1, 1, 1, 2)^T, \dots \end{aligned}$$

t-invarianty:

$$X_1' = (1, 1, 1)^T, \dots$$

Užití invariantů k analýze P/T sítí

Následující věta ukazuje některé možnosti jak využít invarianty k analýze P/T sítí.

Věta 2.3.7:

- (Postačující podmínka pro omezenost PN-systému).
Existuje p-invariant pokrývající celou síť \Rightarrow PN-systém je omezený
- (Nutná podmínka pro živost a omezenost PN-systému).
PN-systém je živý a omezený \Rightarrow Existuje t-invariant pokrývající celou síť
- (Nutná podmínka pro živost PN-systému). Pro živý PN-systém platí:
Y je kladný (netriviální) p-invariant $\Rightarrow Y^T \cdot M_0 > 0$
- (Důkaz nedosažitelnosti v daném PN-systému).
Existuje p-invariant Y s vlastností $Y^T \cdot M \neq Y^T \cdot M' \Rightarrow$ Značení M' není dosažitelné ze značení M

Důkaz:

Ad 1) Nechť Y je p-invariant jehož nosič je množina všech míst. Uvažujme libovolné dosažitelné značení $M \in RS(M_0)$. Pro p-invariant Y platí

$$Y^T \cdot M_0 = Y^T \cdot M \geq Y(p) \cdot M(p),$$

kde p je libovolné místo z P. Protože $Y(p) > 0$, platí pro všechna p

$$M(p) \leq Y^T \cdot M_0 / Y(p)$$

a PN-systém je tedy omezený.

Ad 2) Z předpokladu živosti vyplývá, že existují posloupnosti přechodů $\sigma_1, \sigma_2, \sigma_3, \dots$ takové, že všechna σ_i obsahují všechny přechody a že současně platí $M_0[\sigma_1 > M_1[\sigma_2 > M_2[\sigma_3 > M_3 \dots$. Z předpokladu omezenosti pak vyplývá, že musí existovat indexy $i < j$ takové, že $M_i = M_j$. Charakteristický vektor posloupnosti přechodů $\sigma_{i+1} \sigma_{i+2} \dots \sigma_j$ je t-invariant, jehož nosičem je celá množina T.

Ad 3) Nechť p je místo patřící do nosiče p-invariantu Y, tj. $p \in P_Y$, a uvažujme přechod t, který je sousední k místu p, tj. $t \in {}^*p \cup p^*$. Protože PN-systém je živý může být přechod t vždy znovu

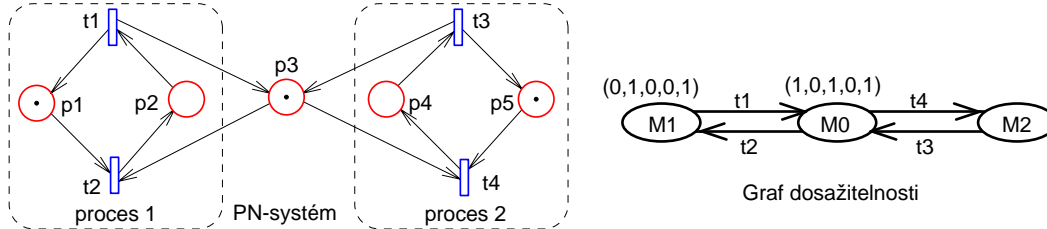
uschopněn, tj. z každého dosažitelného značení je dosažitelné značení M ve kterém je přechod t proveditelný. Jeho provedením se PN-systém dostane do značení M' , tj. $M[t > M'$. Jelikož Y je p-invariant, musí platit $Y^T M' = Y^T M = Y^T M_0$.

Je-li nyní $t \in p^*$, pak $M'(p) > 0$ a protože $p \in P_Y$ je $Y^T M' > 0$ a tedy také $Y^T M_0 > 0$. Je-li $t \in p^*$, pak $M(p) > 0$ a protože $p \in P_Y$ je $Y^T M > 0$ a tedy také $Y^T M_0 > 0$.

Ad 4) Důkaz sporem: necht' značení M' je dosažitelné ze značení M . Potom existuje posloupnost přechodů σ taková, že $M[\sigma > M'$. Podle věty 2.3.1 pak platí fundamentální rovnice $M' = M + CV_\sigma$, odkud plyne $Y^T M' = Y^T M + Y^T CV_\sigma$. Y je p-invariant a tedy platí $Y^T C = \mathbf{0}^T$. Odtud plyne $Y^T M' = Y^T M$, což je ve sporu s předpokladem.

Příklad 2.3.6:

Na obr.2.3.10 je zobrazen jednoduchý PN-systém a jeho graf dosažitelnosti. Z grafu dosažitelnosti vyplývá, že PN-systém je bezpečný, živý a reverzibilní.



Obr.2.3.10 PN-systém a jeho graf dosažitelnosti

Nyní budeme studovat vlastnosti daného PN-systému alternativními algebraickými metodami strukturní analýzy, tj. pomocí p- a t-invariantů příslušné PN-struktury. Z grafického zobrazení(obr.2.3.1) odečítáme incidenční matici C -viz následující tabulka .

| $C=C(p,t)$ | t1: | t2: | t3: | t4: |
|------------|-----|-----|-----|-----|
| p1: | 1 | -1 | 0 | 0 |
| p2: | -1 | 1 | 0 | 0 |
| p3: | 1 | -1 | 1 | -1 |
| p4: | 0 | 0 | -1 | 1 |
| p5: | 0 | 0 | 1 | -1 |

Tab.2.3.10 Incidenční matice PN-struktury

P-invarianty PN-struktury nalezneme jako nezáporná celočíselná řešení maticové rovnice $Y^T.C = \mathbf{0}^T$. Jejím rozepsáním dostáváme soustavu rovnic:

$$\begin{aligned} y_1 - y_2 + y_3 &= 0 \\ -y_1 + y_2 - y_3 &= 0 \\ y_3 - y_4 + y_5 &= 0 \\ -y_3 + y_4 - y_5 &= 0 \end{aligned}$$

2.rovnice je násobkem 1. a 4.rovnice je násobkem 3. Stačí tedy pracovat pouze s 1.a 3.rovnicí. Snadno nalezneme následující bázecká řešení:

$$\begin{aligned} Y_1 &= (1, 1, 0, 0, 0)^T \\ Y_2 &= (0, 0, 0, 1, 1)^T \\ Y_3 &= (0, 1, 1, 1, 0)^T \end{aligned}$$

P-invariantem je i každá lineární kombinace $Y = c_1 Y_1 + c_2 Y_2 + c_3 Y_3$. Těmto strukturním p-invariantům odpovídají následující systémové p-invarianty ($m_i = M(p_i)$):

$$\begin{aligned} m_1 + m_2 &= 1 \\ m_4 + m_5 &= 1 \\ m_2 + m_3 + m_4 &= 1 \end{aligned} \quad (*)$$

Složenému strukturnímu p-invariantu $Y = Y_1 + Y_2 + Y_3$ odpovídá systémový p-invariant

$$m_1 + 2m_2 + m_3 + 2m_4 + m_5 = 3 \quad (**)$$

pokřývající všechna místa, tj. PN-sytém je tedy konzervativní (ale nikoliv striktně). Z (**) vyplývá, že PN-sytém je omezený (tvrzení 1. věty 2.3.7) a z (*) navíc, že je bezpečný (žádné m_i nemůže být větší než 1).

T-invarianty PN-struktury nalezneme jako nezáporná celočíselná řešení maticové rovnice $C \cdot X = 0$. Jejím rozepsáním dostáváme soustavu rovnic:

$$\begin{aligned} x_1 - x_2 &= 0 \\ -x_1 + x_2 &= 0 \\ x_1 - x_2 + x_3 - x_4 &= 0 \\ x_3 - x_4 &= 0 \\ -x_3 + x_4 &= 0 \end{aligned}$$

Vyloučením závislých rovnic dostáváme soustavu:

$$\begin{aligned} x_1 - x_2 &= 0 \\ -x_3 + x_4 &= 0 \end{aligned}$$

ze které vyplývá, že následující dvojice vektorů

$$\begin{aligned} X_1 &= (1, 1, 0, 0)^T, \\ X_2 &= (0, 0, 1, 1)^T, \end{aligned}$$

tvoří úplný systém minimálních t-invariantů. Strukturním t-invariantem je také každá jejich lineární kombinace $X = c_1 X_1 + c_2 X_2$, např. $X_3 = X_1 + X_2 = (1, 1, 1, 1)^T$.

Strukturním t-invariantům X_1, X_2, X_3 odpovídají systémové t-invarianty (posloupnosti přechodů převádějící počáteční značení opět v počáteční značení):

X_1 : $t_2 t_1$ ale nikoliv $t_1 t_2$

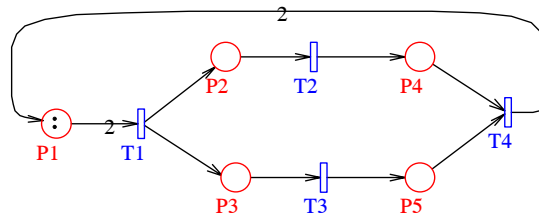
X_2 : $t_4 t_3$ ale nikoliv $t_3 t_4$

X_3 : $t_2 t_1 t_4 t_3, t_4 t_3 t_2 t_1$ ale nikoliv jiné permutace přechodů t_1, t_2, t_3, t_4

Protože existuje t-invariant pokrývající celou síť (např. $t_2 t_1 t_4 t_3$) je PN-sytém repetiční (konzistentní) a to striktně (neboť $X_3 = (1, 1, 1, 1)^T$).

Příklad 2.3.7:

V tomto příkladě budeme počítat invarianty PN-sytému (zobrazeného na obr. 2.3.11) pomocí algoritmu popsánoho v poznámce 2.3.4 (p-invarianty a s využitím věty 2.3.6 také t-invarianty).



Obr. 2.3.11

Výpočet úplného systému minimálních p-invariantů je zachycen v tabulce 2.3.11. Silně zarámovaná horní část (řádky (1) - (5)) představuje počáteční stav tabulky (tj. složené matice $(A|B) = (C|I)$), silně zarámovaná dolní část (řádky (8),(9)) představují koncový stav tabulky po ukončení práce algoritmu. Pravá část této spodní části obsahuje úplný systém minimálních p-invariantů dané struktury. Vlevo od tabulky je popsán postup výpočtů (vytváření nových řádků) a vpravo od tabulky je vyznačeno postupně škrtnání dále nepotřebných řádků.

| výpočet p-invariantů | $A = C$ | | | | $B = I$ | | | | | |
|----------------------|---------|----|----|----|---------|---|---|---|---|---|
| (1) | -2 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | x |
| (2) | 1 | -1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | x |
| (3) | 1 | 0 | -1 | 0 | 0 | 0 | 1 | 0 | 0 | x |
| (4) | 0 | 1 | 0 | -1 | 0 | 0 | 0 | 1 | 0 | x |

| | | | | | | | | | | |
|---------------|---|----|----|----|---|---|---|---|---|------------------|
| (5) | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | 1 | x |
| (6)=(1)+2.(2) | 0 | -2 | 0 | 2 | 1 | 2 | 0 | 0 | 0 | x |
| (7)=(1)+2.(3) | 0 | 0 | -2 | 2 | 1 | 0 | 2 | 0 | 0 | x |
| (8)=2.(4)+(6) | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 2 | 0 | = Y ₁ |
| (9)=2.(5)+(7) | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 2 | = Y ₂ |

Tab. 2.3.11

Z tabulky vyplývá, že úplný systém minimálních p-invariantů je tvořen vektory $Y_1=(1,2,0,2,0)^T$, $Y_2=(1,0,2,0,2)^T$ a tedy libovolný p-invariant PN-systému má tvar $Y=k_1Y_1+k_2Y_2$, kde $k_i \in \{0,1,2,\dots\}$. PN-struktura je p-invarianty Y_1, Y_2 pokryta, např. invariant $Y_1+Y_2=(2,2,2,2,2)^T$ a tedy také invariant $Y_3=(1,1,1,1,1)^T$ v kanonickém tvaru pokrývá celou PN-strukturu.

Každému p-invariantu struktury Y odpovídá p-invariant systému $Y^T M = Y^T M_0$. V našem případě invariantům struktury Y_1, Y_2, Y_3 odpovídají následující invarianty systému:

$$\begin{aligned} m_1+2m_2 + 2m_4 &= 2, \\ m_1 + 2m_3 + 2m_5 &= 2, \\ m_1 + m_2 + m_3 + m_4 + m_5 &= 2. \end{aligned}$$

PN-systém má tedy dvě netriviální konzervativní komponenty indukované množinami míst $\{p_1, p_2, p_4\}$ a $\{p_1, p_3, p_5\}$ a také celý PN-systém (indukovaný množinou všech míst $P=\{p_1, p_2, \dots, p_5\}$) tvoří konzervativní komponentu. PN-systém je tedy konzervativní a to striktně. Z konzervativnosti systému vyplývá jeho omezenost. Z výše uvedeného systému rovnic lze navíc odvodit $m_1 \leq 2$ a $m_i \leq 1$ pro $i=2,3,4,5$ a pro $\forall M \in RS(M_0)$.

V tabulce 2.3.12 jsou t-invarianty dané struktury počítány stejným způsobem jako p-invarianty v tabulce 2.3.11. Jediný rozdíl je ve výchozím stavu tabulky, místo tvaru (C, I_5) je tvaru (C^T, I_4) .

| výpočet t-invariantů | A = C ^T | | | | | B = I | | | | |
|----------------------|--------------------|----|----|----|----|-------|---|---|---|------------------|
| (1) | -2 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | x |
| (2) | 0 | -1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | x |
| (3) | 0 | 0 | -1 | 0 | 1 | 0 | 0 | 1 | 0 | x |
| (4) | 2 | 0 | 0 | -1 | -1 | 0 | 0 | 0 | 1 | x |
| (5)=(1)+(4) | 0 | 1 | 1 | -1 | -1 | 1 | 0 | 0 | 1 | x |
| (6)=(2)+(5) | 0 | 0 | 1 | 0 | -1 | 1 | 1 | 0 | 1 | x |
| (7)=(3)+(6) | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | = X ₁ |

Tab. 2.3.12

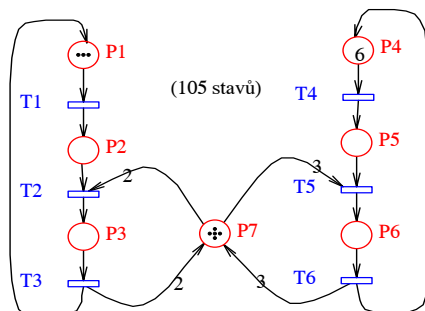
Z tabulky 2.3.12 vyplývá, že úplný systém minimálních t-invariantů je tvořen jediným vektorem $X_1=(1,1,1,1,1)^T$ a každý t-invariant dané PN-struktury má tvar $X=k_1X_1$, kde $k_1 \in \{0,1,2,\dots\}$. Existují-li t-invarianty daného PN-systému, tj. posloupnosti přechodů spustitelné v počátečním značení M_0 a převádějící PN-systém z M_0 opět do M_0 , pak charakteristický vektor této posloupnosti (t-invariantu PN-systému) musí být roven t-invariantu struktury. V našem případě této podmínce vyhovují posloupnosti $t_1t_2t_3t_4$ a $t_1t_3t_2t_4$. PN-systém je tedy konzistentní a to striktně. Z konzistence PN-systému vyplývá jeho živost.

Příklad 2.3.8:

V tomto příkladě ukážeme, že invarianty PN-systémů můžeme mnohdy stanovit přímo ze zadání PN-systému bez toho, že bychom byli nuceni formálně počítat invarianty přískušných PN-struktur. To platí pro jednoduché PN-systémy (např. pro PN-systém z předchozího příkladu), ale často i pro složitější systémy, jsou-li dobře navrženy (dobrý návrh systému počítá s následnou analýzou a je strukturován tak, aby analýza byla snadno proveditelná, což speciálně znamená, aby invarianty byly na prvý pohled viditelné). U složitějších systémů hrozí však riziko, že některé invarianty mohou být přehlédnuty.

Uvažujme PN-systém zobrazený grafovým diagramem na obr. 2.3.2 (konkrétní interpretace PN-systému může být např. podobná interpretacím podobných systémů v příkladech 1.3.3 nebo 1.5.1). Lze dokázat, že množina dosažitelných značení (stavů) tohoto systému má 105 prvků a s rostoucím počtem tokenů v místech p_1, p_4 a p_7 dále prudce stoupá. Analýza metodou stavového prostoru (pomocí grafu dosažitelnosti) je pracná a postupně se stává

prakticky nemožnou. Naproti tomu strukturální analýza pomocí p- a t-invariantů je podstatně jednodušší a její pracnost je stále stejná a nezávislá na počtu tokenů v místech.



Obr. 2.3.12

U našeho PN-systému můžeme invarianty snadno odečítat přímo z obr. 2.3.12. Úplný systém minimálních p-invariantů je dán následujícími třemi rovnicemi:

$$\begin{aligned} m_1 + m_2 + m_3 &= 3, \\ m_4 + m_5 + m_6 &= 6, \\ 2m_3 + 3m_6 + m_7 &= 5. \end{aligned}$$

PN-systém obsahuje tedy tři konzervativní komponenty s nosiči $\{p_1, p_2, p_3\}$, $\{p_4, p_5, p_6\}$, $\{p_3, p_6, p_7\}$. Tyto komponenty pokrývají celý PN-systém. Sečtením uvedených třech rovnic dostáváme rovnici

$$m_1 + m_2 + 3m_3 + m_4 + m_5 + 4m_6 + m_7 = 14,$$

která představuje konzervativní zákon pro celý systém. PN-systém je tedy konzervativní, ale nikoliv striktně. Z konzervativnosti PN-systému vyplývá jeho omezenost. Analýzou výše uvedených rovnic lze odvodit další poznatky, např.

$$m_1, m_2 \in \{0, 1, 2, 3\}, \quad m_3 \in \{0, 1, 2\}, \quad m_4, m_5 \in \{0, 1, 2, \dots, 6\}, \quad m_6 \in \{0, 1\}$$

(tj. místo p_6 je bezpečné).

Také t-invarianty lze odečítat bezprostředně z grafového obrázku PN-systému. Úplný systém minimálních t-invariantů je tvořen následujícími dvěma posloupnostmi přechodů:

$$t_1 t_2 t_3, \quad t_4 t_5 t_6,$$

které představují dvě repetiční komponenty s nosiči $\{t_1, t_2, t_3\}$, $\{t_4, t_5, t_6\}$. Tyto dvě repetiční komponenty se kryjí s výše určenými konzervativními komponentami a pokrývají celý PN-systém. Repetičními posloupnostmi daného PN-systému (tj. PN-systému s daným počátečním značením) jsou např. posloupnosti přechodů

$$t_1 t_2 t_3 t_4 t_5 t_6, \quad t_4 t_5 t_6 t_1 t_2 t_3, \quad t_1 t_4 t_2 t_3 t_5 t_6, \dots$$

(ale nikoliv např. $t_2 t_1 t_3 t_4 t_5 t_6$, $t_1 t_3 t_2 t_4 t_5 t_6$, ...). PN-systém je tedy repetiční (a to striktně - ve striktně repetiční posloupnosti se každý přechod vyskytuje právě jednou). PN-systém je tedy také živý.

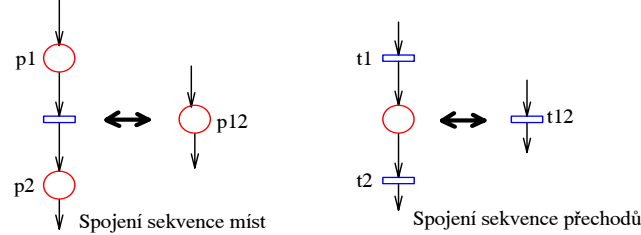
Grafové techniky strukturální analýzy

Metoda redukce PN-systémů

Podstatou této metody je postupná transformace daného PN-systému na PN-systémy stále jednodušší až nakonec získáme PN-systém jehož stavová analýza je buď triviální nebo alespoň snadno zvládnutelná. Každý krok transformace spočívá v použití nějakého redukčního pravidla (reduction rule) podle kterého nahrazujeme nějaký podsystém daného systému jiným jednodušším podsystémem přičemž jsou zachovány některé vybrané vlastnosti PN-systémů (jako jsou omezenost, živost, reverzibilita,...). Jestliže jsme při transformaci používali pravidla, která všechna zachovávají daný soubor vlastností, pak výsledný PN-systém tyto vlastnosti má také, pokud je má výchozí systém.

Ukázka dvou jednoduchých redukčních pravidel, které evidentně zachovávají omezenost a živost PN-systémů je na obr. 2.3.13. Při analýze PN-systémů nahrazujeme (zjednodušujeme) levou stranu pravou stranou,

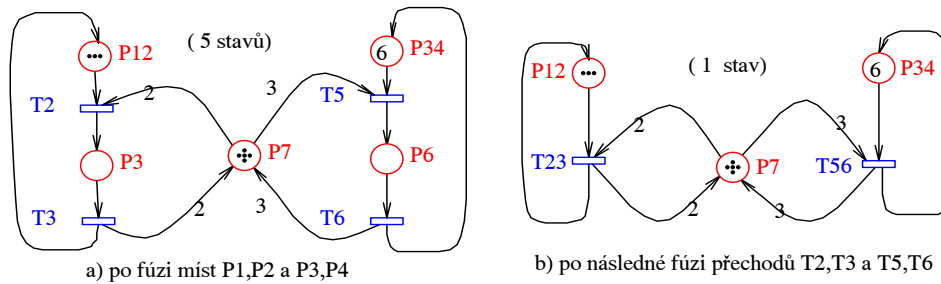
při syntéze nahrazujeme (rozvíjíme) pravou stranu stranou levou. K pravidlu spojení sekvence míst (fusion of series places) dodejme, že každá případná další vstupní hrana místa p1 je také vstupní hranou místa p12 a podobně také každá další výstupní hrana místa p2 je také výstupní hranou místa p12. Navíc počet tokenů v místě p12 musí být roven součtu počtu tokenů v místech p1 a p2. Pro pravidlo spojení sekvence přechodů (fusion of series transitions) platí tataž poznámka o případných dalších vstupních hranách přechodu t1 a výstupních hranách přechodu t2. V místě mezi oběma přechody se nesmí nacházet žádné tokeny.



Obr. 2.3.13

Příklad 2.3.9:

Uvažujme PN-systém definovaný v příkladě 2.3.8 obrázkem 2.3.12. Pomocí pravidla o spojení sekvence míst transformujeme PN-systém z tohoto obrázku obrázkem na PN-systém zobrazený na obr.2.3.14.a a následně pomocí pravidla o spojení sekvence přechodů na PN-systém zobrazený na obr. 2.3.14.b. Zatímco původní systém má 105 stavů (dosažitelných značení), druhý systém má již jen 5 stavů a konečný systém má již jen 1 stav. Jeho analýza je tedy triviální: systém je živý a omezený (6-omezený). Protože transformační (redukční) pravidla zachovávají omezenost i živost, musí být i výchozí systém z obr. 2.3.12 omezený a živý.



Obr. 2.3.14

Zámky a pasti

V této části předpokládáme Petriho sítě jen s jednoduchými hranami (tj. jen s násobností 1) a bez inhibičních hran. Předpokládáme tedy *obyčejné Petriho sítě* (ordinary Petri nets).

Definice 2.3.10:

Podmnožinu Q množiny míst P ordinární struktury <P,T,I,O> nazveme *zámkem* (lock, siphon), jestliže platí

$${}^{\bullet}Q \subseteq Q^{\bullet}$$

Podmnožinu Q množiny míst P ordinární struktury <P,T,I,O> nazveme *pastí* (trap), jestliže platí

$$Q^{\bullet} \subseteq {}^{\bullet}Q$$

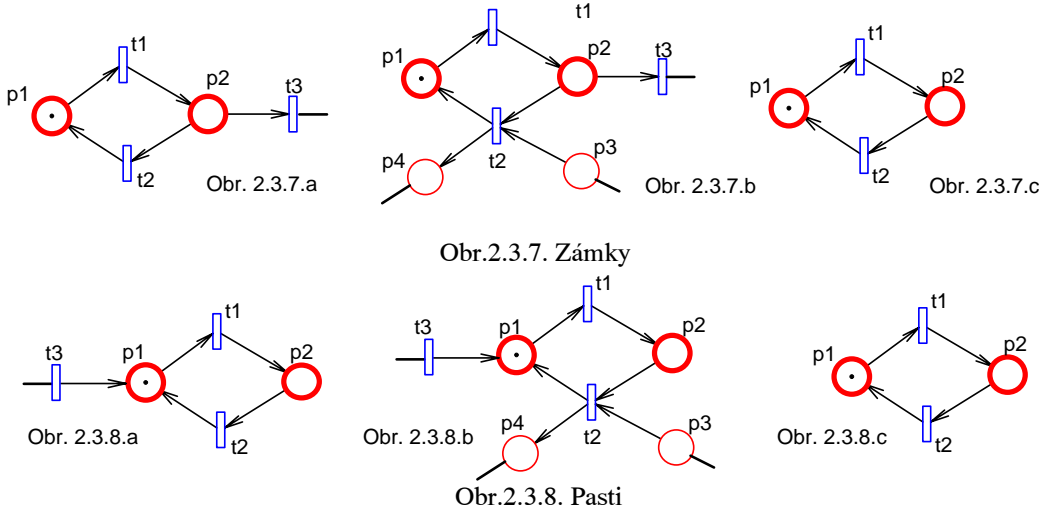
Kvůli úplnosti dodejme, že

$${}^{\bullet}Q = \bigcup_{p \in Q} \{ {}^{\bullet}p \}, \quad Q^{\bullet} = \bigcup_{p \in Q} \{ p^{\bullet} \}, \quad {}^{\bullet}p = \{ t \in T : O(t,p) > 0 \}, \quad p^{\bullet} = \{ t \in T : I(t,p) > 0 \}.$$

Příklad 2.3.8:

Na obr. 2.3.8 jsou zobrazeny jednoduché zámky a na obrázku 2.3.8 jednoduché pasti. Zámky, resp. pasti, jsou vždy tvořeny vyznačenou množinou míst $Q = \{p_1, p_2\}$. Pro zobrazené PN-struktury platí:

- Obr.2.3.7.a,b: $\{t_1, t_2\} = {}^*Q \subseteq Q^* = \{t_1, t_2, t_3\}$
- Obr.2.3.7.c: $\{t_1, t_2\} = {}^*Q \subseteq Q^* = \{t_1, t_2\}$
- Obr.2.3.8.a,b: $\{t_1, t_2, t_3\} = Q^* \subseteq {}^*Q = \{t_1, t_2\}$
- Obr.2.3.8.c: $\{t_1, t_2\} = Q^* \subseteq {}^*Q = \{t_1, t_2\}$



Poznámky 2.3.9:

1. Zámek (sifon) je podmnožina množiny míst tvořená místy, která když ztratí tokeny, nemohou je již nikdy získat. Naproti tomu past je podmnožina množiny míst tvořená místy, která když získají tokeny, nemohou je již nikdy ztratit.
2. Z definice 2.3.8 vyplývá, že sjednocení dvou zámků je zámek a sjednocení dvou pastí je past. Pro Petriho sítí mohou být tedy definovány minimální zámky a pasti.
3. Zámky a pasti lze využít k analýze PN-systémů. Tak např.:
 - Obsahuje-li PN-struktura zámek z obrázku 2.3.7, který při počátečním značení neobsahuje žádný token, pak PN-systém není živý.
 - Obsahuje-li PN-struktura past z obrázku 2.3.8, která při počátečním značení obsahuje aspoň jeden token, pak PN-systém je bez uzamčení.

2.4. Podtřídy P/T Petriho sítí

V této kapitole se budeme zabývat následujícími speciálními typy P/T Petriho sítí:

- automatové Petriho sítě,
- synchronizační Petriho sítě,
- sítě s volným výběrem.

Všechny uvedené typy jsou speciálním případem tzv. obyčejných Petriho sítí a ty opět speciálním případem tzv. zobecněných Petriho sítí. Zobecněné Petriho sítě jsou pak speciálním případem PN-systému z definice 2.1.2.

Definice 2.4.1:

Zobecněná Petriho síť (generalized Petri net - GPN) je PN-systém $\langle P, T, I, O, H, M_0 \rangle$ s prázdnou množinou inhibičních hran $H = \emptyset$, tj. PN-systém $\langle P, T, I, O, M_0 \rangle$.

Obyčejná Petriho síť (ordinary Petri net - OPN) je zobecněná Petriho síť $\langle P, T, I, O, M_0 \rangle$, kde hodnotami vstupní a výstupní funkce I, O nejsou multimnožiny, ale množiny (tj. všechny hrany mají stejnou násobnost 1).

Automatová síť (state-machine Petri net, P-system) je obyčejná Petriho síť ve které každý přechod má právě jedno vstupní a právě jedno výstupní místo, tj.

$$(\forall t \in T)[|\bullet t| = |t \bullet| = 1].$$

Synchronizační síť (marked graph, synchronization graph, T-system) je obyčejná Petriho síť ve které každé místo má právě jeden vstupní a právě jeden výstupní přechod, tj.

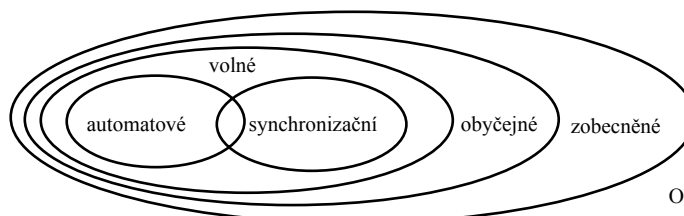
$$(\forall p \in P)[|\bullet p| = |p \bullet| = 1].$$

Volná síť (síť s volným výběrem, free-choice Petri net) je obyčejná Petriho síť ve které všichni předchůdci každého přechodu mají stejnou množinu následníků, tj.

$$(\forall t \in T)(\forall p, p' \in \bullet t)[p \bullet = p' \bullet].$$

Poznámky 2.4.1:

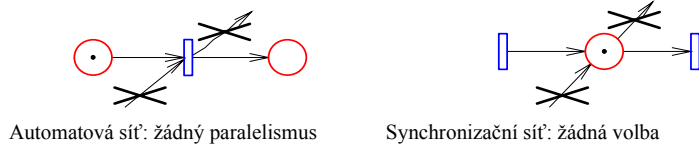
1. Vztahy mezi třídami sítí zobecněných, obyčejných, volných, automatových a synchronizačních jsou zobrazeny následujícím množinovým obrázkem. Platnost zobrazených vztahů vyplývá ihned z definice 2.4.1.



Obr. 2.4.1

2. Z definice 2.4.1 rovněž ihned vyplývá:
 - Duální síť k automatové síti je síť synchronizační a obráceně.
 - Duální síť k volné síti je opět volná síť.

3. Automatové sítě nejsou schopny vyjádřit paralelismus, synchronizační sítě nejsou schopny vyjádřit volbu (alternativu) - viz obr.2.4.0.a. Volné sítě jsou schopny zobrazit obé, i když v omezeném rozsahu. V linii od P/T sítí (přes zobecněné a obyčejné sítě) k volným sítím a dále k synchronizačním (nebo automatovým) sítím postupně klesá modelovací síla sítí, ale současně se zlepšují možnosti a snadnost jejich analýzy. Zdá se, že volné sítě představují ideální kompromis mezi oběma protichůdnými tendencemi.

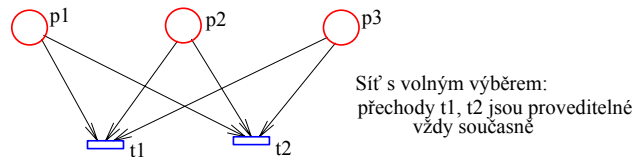


Automatová síť: žádný paralelismus

Synchronizační síť: žádná volba

Obr.2.4.0.a

4. Definici volné sítě můžeme ekvivalentním způsobem formulovat také takto: Pro každou hranu (p,t) vedoucí z místa p do přechodu t platí: z každého místa $p' \in \bullet t$ míří hrana do každého přechodu $t' \in p^*$. Tento tvar definice ilustruje obr.2.4.0.b.

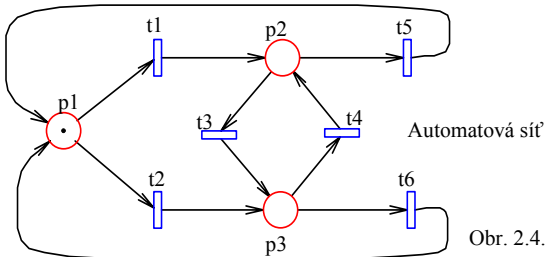


Síť s volným výběrem:
přechody t_1, t_2 jsou proveditelné
vždy současně

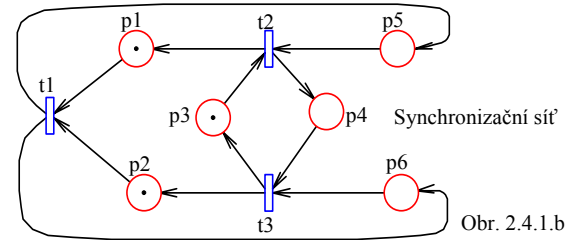
Obr.2.4.0.b

Příklad 2.5.1:

Na obr.2.4.1.a je zobrazena automatová a na obr.2.4.1.b synchronizační síť. Všimněme si, že se jedná o navzájem duální Petriho sítě. Všimněme si rovněž, že zobrazení přechodů automatové sítě je nadbytečné - pomocí symbolů t_1, t_2, \dots, t_6 můžeme ohodnotit přímo hrany spojující místa. Získáme tak diagram konečného automatu (částečného).

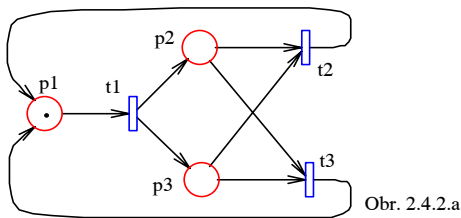


Obr. 2.4.1.a

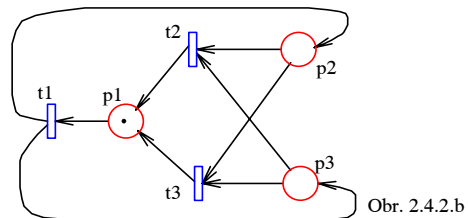


Obr. 2.4.1.b

Na obr.2.4.2.a, b jsou zobrazeny sítě s volným výběrem. Všimněme si, že tyto sítě jsou navzájem duální.



Obr. 2.4.2.a



Obr. 2.4.2.b

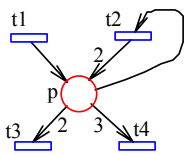
Obyčejné Petriho sítě

Věta 2.4.1:

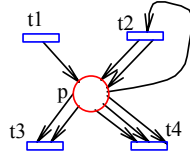
Obyčejné Petriho sítě (OPN) mají stejnou modelovací schopnost jako zobecněné Petriho sítě (GPN). Ke každé GPN existuje ekvivalentní OPN, která má totéž chování a tytéž vlastnosti jako původní GPN.

Důkaz:

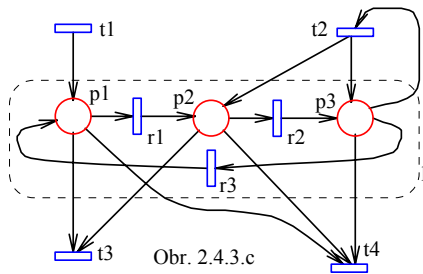
Vysvětlíme pouze způsob konstrukce ekvivalentní OPN $\langle P', T', I', O', M_0' \rangle$ k dané GPN $\langle P, T, I, O, M_0 \rangle$. Při konstrukci postupujeme tak, že každé místo $p \in P$ sítě GPN rozvineme v podsítí OPN. Nejdříve určíme maximum m_p z násobností všech hran incidentních z daným místem p . Např. pro místo p zobrazené na obr.2.4.3.a (resp. na ekvivalentním obrázku 2.4.3.b) je to číslo 3. Místo p pak nahradíme jednoduchou cyklickou podsítí obsahující m_p míst a m_p přechodů. Např. pro místo p z obr.2.4.3.a je to podsítí tvořena místy p_1, p_2, p_3 a přechody r_1, r_2, r_3 vyznačená na obr.2.4.3.c. Nakonec spojíme vstupní a výstupní přechody uvažovaného místa p s místy p_1, p_2, p_3 podsítě libovolně, ale tak, aby všechny hrany byly jednoduché (s násobností 1). Na obr.2.4.3.c je ukázáno jedno takové spojení pro situaci z obr. 2.4.3.a. Je-li některý z výstupních přechodů t_3, t_4 proveditelný v situaci na obr.2.4.3.a, pak ve vyznačené podsítí na obr.2.4.3.c se vždy mohou tokeny v místech podsítě rozmístit tak, aby byl příslušný přechod také proveditelný.



Obr. 2.4.3.a



Obr. 2.4.3.b



Obr. 2.4.3.c

Pro popis vlastností automatových a synchronizačních sítí se nám budou hodit následující pojmy a notace.

Definice 2.4.2:

Cyklem v PN-struktuře nazveme slovo

$$c = p_{(1)}t_{(1)}p_{(2)}t_{(2)}\dots p_{(n)}t_{(n)}p_{(n+1)}$$

nad abecedou $P \cup T$, kde $p_{(i)} \in P$, $t_{(i)} \in T$ pro $i=1,2,\dots,n$ a

$$p_{(n)} = p_{(n+1)} \wedge (\forall i)[p_{(i)} \in I(t_{(i)})] \wedge (\forall i)[p_{(i+1)} \in O(t_{(i)})].$$

Budeme předpokládat, že cykl je **jednoduchý**, tj. že všechny přechody cyklu jsou různé a stejně tak i místa, s jedinou výjimkou míst $p_{(n)}$ a $p_{(n+1)}$. Množinu všech cyklů dané PN-struktury označíme písmenem C .

P-nosičem cyklu c nazveme množinu

$$P_c = \{p \in P: p \text{ se vyskytuje v slově } c\}$$

a **T-nosičem cyklu** c nazveme množinu

$$T_c = \{t \in T: t \text{ se vyskytuje v slově } c\}.$$

Symbolem $n_M(c)$ označíme **počet tokenů v cyklu** c při značení M , tj.

$$n_M(c) = \sum_{p \in P_c} M(p)$$

a symbolem C_p **množinu všech cyklů, jejichž nosiče obsahují místo** p , tj.

$$C_p = \{c \in C: p \in P_c\}.$$

Zápisem $x \rightarrow y$ označíme skutečnost, že v síti $\langle P, T, I, O \rangle$ vede z uzlu x (místa nebo přechodu) do uzlu y (přechodu nebo místa) orientovaná cesta.

Automatové síť

Věta 2.4.2:

Automatové síť se vyznačují následujícími vlastnostmi:

(1) Jsou konzervativní, tj. platí

$$(\forall M \in RS(M_0) [\sum_{p \in P} M(p) = \sum_{p \in P} M_0(p)]).$$

(2) Jsou omezené.

(3) Je-li $\sum_{p \in P} M_0(p) = 1$, pak je síť bezpečná.

(4) T-nosič každého cyklu potenciálně vytváří repetiční komponentu sítě.

(5) Je-li síť silně souvislá a počáteční značení je nenulové, pak je automatová síť živá.

Důkaz:

- (1) Plyne z faktu, že provedením libovolného přechodu se počet tokenů v síti nemění (token se přesune z jediného vstupního místa přechodu do jediného výstupního místa).
- (2) Je důsledkem konzervativnosti.
- (3) Je důsledkem konzervativnosti.
- (4) Provedením všech přechodů T-nosiče cyklu se obnovuje původní značení. Je-li proveditelný jeden přechod z T-nosiče cyklu, pak jsou proveditelné všechny přechody T-nosiče.
- (5) Je-li počáteční značení nenulové, pak je proveditelný aspoň jeden přechod sítě. V důsledku silné souvislosti sítě, může být pak kdykoliv (ne však vždy okamžitě) proveden kterýkoliv přechod sítě.

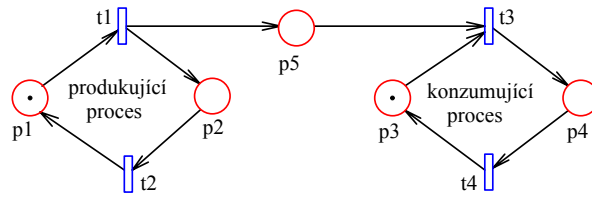
Synchronizační síť

Příklad 2.4.2:

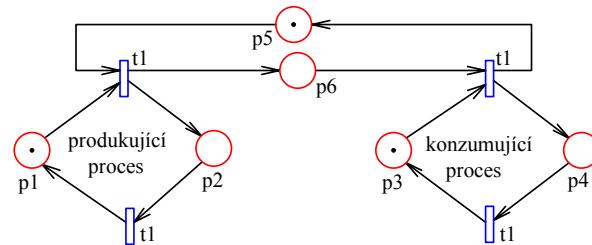
Na obr.2.4.4-6 jsou zobrazeny synchronizační síť, ukazující jak lze synchronizovat dva procesy, z nichž jeden lze interpretovat jako produkující a druhý jako konzumující produkty jistého druhu. Přechod t_1 reprezentuje expedici, t_2 produkci, t_3 přejímku a t_4 konzumci výrobku.

Na obr.2.4.4 je znázorněna nejméně těsná vazba mezi oběma procesy: spotřebovat nelze více než je vyprodukováno. Síť obsahuje cykly s následujícími P-nosiči: $\{p_1, p_2\}$, $\{p_3, p_4\}$.

Na obr.2.4.5 je zobrazena těsnější vazba: další výrobek lze expedovat teprve tehdy až předchozí byl zkonzumován. Zvětšením počtu tokenů v místě p_5 můžeme modelovat výrobu na sklad s omezenou kapacitou skladu. Síť obsahuje cykly s následujícími P-nosiči: $\{p_1, p_2\}$, $\{p_3, p_4\}$, $\{p_5, p_6\}$.

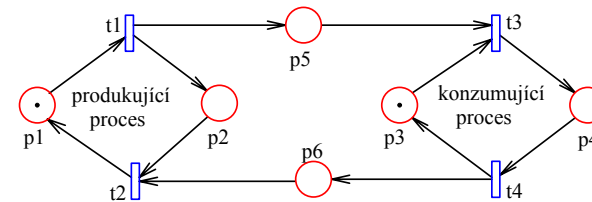


Obr. 2.4.4



Obr. 2.4.5

Konečně na obr.2.4.6 je zobrazena ještě těsnější vazba: další exemplář výrobku může být produkován až po konzumaci předchozího exempláře. Síť obsahuje cykly s následujícími P-nosiči: $\{p_1, p_2\}$, $\{p_3, p_4\}$, $\{p_1, p_5, p_4, p_6\}$.



Obr. 2.4.6

Věta 2.4.3:

Synchronizační síť se vyznačují následujícími vlastnostmi:

- (1) Počet tokenů v každém cyklu je konstantní, tj. P-nosič každého cyklu potenciálně vytváří konzervativní komponentu, tj pro každý cyklus platí:

$$(\forall M \in RS(M_0)[n_{M(c)} = n_{M_0(c)} = n(c)].$$

- (2) Jestliže počáteční značení umísťuje do každého cyklu aspoň jeden token, tj. jestliže

$$(\forall c \in C)[n_{M_0(c)} \geq 1],$$

pak je síť živá.

- (3) Maximální počet tokenů, který se může nacházet v daném místě je roven minimu z konstantních počtů tokenů kolujících v cyklech, které se protínají v daném místě, tj.

$$(\forall p \in P)[\max_{M \in RS(M_0)} \{M(p)\} = \min_{c \in C_p} \{n(c)\}].$$

- (4) Jestliže každé místo sítě patří k nějakému cyklu s počtem tokenů 1, tj. jestliže

$$(\forall p \in P)(\exists c \in C)[p \in P_c \wedge n(c)=1],$$

pak je síť bezpečná.

Důkaz:

- (1) Každé místo v síti, a tedy také v každém cyklu sítě, má jediný vstupní a jediný výstupní přechod. Provedením libovolného přechodu cyklu se tedy nemůže změnit celkový počet tokenů v cyklu.
- (2) Je-li na počátku v každém cyklu aspoň jeden token, pak každý přechod, který leží v nějakém cyklu může být vždy znovu proveden. Uvažujme přechod, který nepatří do žádného cyklu. Takový přechod leží na nějaké cestě,

kteřá buď začíná přechodem patřícím do nějakého cyklu a nebo přechodem, který nemá vstupní místo. V obou případech může být uvažovaný přechod opakovaně prováděn.

- (3) Uvažujme místo, které patří do několika cyklů sítě. Podle (1) zůstává při evoluci sítě počet tokenů v každém cyklu konstantní a tedy může být maximálně rovno této konstantě. To platí pro každý cyklus procházející daným místem a tedy maximální počet tokenů, které se v daném místě mohou nacházet musí být roven minimu z těchto konstant.
- (4) Tvrzení je jednoduchým důsledkem vlastnosti (3).

Platnost všech tvrzení věty lze názorně demonstrovat na příkladech synchronizačních sítí uvedených v příkladu 2.4.2.

Sítě s volným výběrem

Věta 2.4.4:

(Commoner's Theorem) Pro volné sítě platí:

Každý zámek sítě obsahuje past s aspoň jedním tokenem. \Leftrightarrow Síť je živá.

Důkaz:

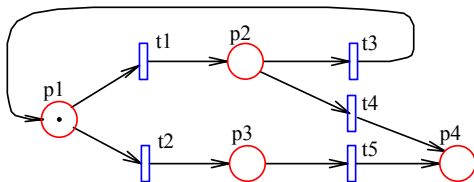
Úplný a přesný důkaz je trochu komplikovaný. Omezíme se proto jenom na hlavní ideu důkazu (a to jenom ve směru \Leftarrow). Přechod mající vstupní místo v zámku nemůže být proveden, jestliže je zámek prázdný (tj. neobsahuje-li žádný token). Z toho vyplývá, že síť obsahující zámek, který může při evoluci sítě ztratit všechny tokeny, není živá. K tomu, aby se tokeny nemohly ze zámku nikdy vytrátit, postačí bude-li každý zámek obsahovat past s aspoň jedním tokenem.

Příklad 2.4.3:

Platnost věty budeme demonstrovat na třech jednoduchých příkladech: 1) automatové sítě, 2) synchronizační sítě a 3) volné sítě, která není ani automatová ani synchronizační.

1) Na obr.2.4.7 je zobrazena automatová síť, která evidentně není živá (provedení sekvence t_1t_4 , nebo t_2t_5 vede dokonce k uzamčení systému). Z pohledu věty 2.4.4 dojdeme k témuž závěru takto: podmnožina $P' = \{p_1, p_2\}$ je zámek, který neobsahuje žádnou past (natož pak past s aspoň jedním tokenem). Pro úplnost vypíšeme všechny zámky a pasti zobrazené sítě:

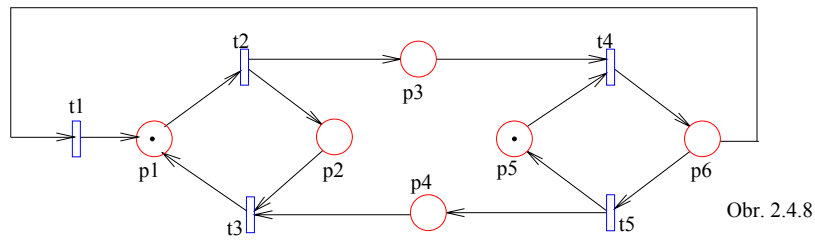
- Netriviální zámky: $\{p_1, p_2\}$, $\{p_3, p_4\}$, $\{p_1, p_2, p_3\}$.
Triviální zámek: $\{p_1, p_2, p_3, p_4\}$.
- Netriviální pasti: $\{p_4\}$, $\{p_3, p_4\}$, $\{p_1, p_2, p_4\}$.



Obr. 2.4.7

2) Na obr.2.4.4 je zobrazena synchronizační síť, která je evidentně živá. Zjistíme to buď stavovou analýzou, nebo na základě tvrzení (2) věty 2.4.3. Z pohledu věty 2.4.4 dojdeme k témuž závěru takto: každý ze zámků $\{p_1, p_2\}$, $\{p_3, p_4\}$, $\{p_1, p_2, p_5\}$ obsahuje jednu ze dvou pastí $\{p_1, p_2\}$, $\{p_3, p_4\}$ z nichž každá obsahuje aspoň jeden token.

3) Na obr.2.4.8 je zobrazena volná síť, která obsahuje netriviální zámky $\{p_5, p_6\}$, $\{p_4, p_5, p_6\}$, neobsahující žádné pasti (natož pasti s aspoň jedním tokenem). Síť tedy není, podle věty 2.4.4, živá.



K stejnému závěru dojdeme metodou stavové analýzy (viz tab.2.4.1 a obr.2.4.9) :

| | t1 | t2 | t3 | t4 | t5 | M0 | M1 | M2 | M3 | M4 | M5 |
|----|----|----|----|----|----|-------|-------|----------------|-------|-------|-----|
| p1 | 1 | -1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| p2 | 0 | 1 | -1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 |
| p3 | 0 | 1 | 0 | -1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| p4 | 0 | 0 | -1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| p5 | 0 | 0 | 0 | -1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| p6 | -1 | 0 | 0 | 1 | -1 | 0 | 0 | 1 | 0 | 0 | 0 |
| | | | | | | t2→M1 | t4→M2 | t1→M3 t5→M4 | t2→M5 | t3→M0 | --- |

Tab. 2.4.1.

Provedením sekvence přechodů $t_2 t_4 t_1 t_2$ dospějeme z počátečního značení (1,0,0,0,1,0) do značení (0,2,1,0,1,0), ve kterém není žádný přechod proveditelný. Síť obsahuje deadlock a tedy není živá.

2.5. Hierarchická syntéza Petriho sítí

Dosud nebyla předpokládána žádná vnitřní struktura míst nebo přechodů Petriho sítě. V této kapitole připustíme, že místa a přechody dané sítě mohou představovat opět sítě (hierarchicky nižší úroveň vzhledem k dané síti) a na druhé straně celá daná síť může být pojímána jako jediné místo nebo jediný přechod sítě hierarchicky vyšší vzhledem k dané síti.

S touto představou souvisí představa, že každou Petriho síť lze získat z jisté výchozí triviální sítě jejím postupným rozvíjením, tj. nahrazováním jejích míst a přechodů zpodrobnujícími sítěmi. Dodržují-li se při tomto rozvíjení jistá pravidla, potom rozvíjené Petriho sítě mají v každé fázi svého rozvoje zachovány jisté žádoucí vlastnosti (např. jsou "správné" - viz následující definice 2.5.1)

V této kapitole se budeme zabývat pouze vybranými hierarchizačními konstrukty (substituce míst a přechodů speciálního typu) a to pouze pro speciální typ Petriho sítí (obyčejné Petriho sítě). Později budeme používat širší sadu hierarchizačních konstruktů a to pro podstatně obecnější typy Petriho sítí (barevné sítě).

Definice 2.5.1:

Obyčejnou Petriho síť nazveme *správnou* (korektní) *sítí* jestliže je současně bezpečná, živá a reverzibilní, tj. platí-li současně:

- | | |
|---|------------------|
| (1) $(\forall p \in P)(\forall M \in RS(M_0))[M(p) \leq 1]$ | (bezpečnost), |
| (2) $(\forall t \in T)(\exists M, M' \in RS(M_0))[M[t] > M']$ | (živost), |
| (3) $(\forall M, M' \in RS(M_0))(\exists \sigma \in T^*)[M[\sigma] > M']$ | (reverzibilita). |

Poznámky 2.5.1:

- O správnosti předložené Petriho sítě se lze přesvědčit nalezením množiny $RS(M_0)$, setrojením grafu, setrojením grafu dosažitelnosti RG a provedením stavové analýzy podle věty 2.2.1. Protože síť o n místech má nejvýše 2^n bezpečných značení, je algoritmus zjišťování správnosti sítě konečný. S rostoucím počtem míst může počet jeho kroků exponenciálně vzrůstat, což podstatně omezuje možnosti stavové analýzy při prověřování správnosti Petriho sítě.
- Poznamenejme, že podmínka živosti (2) uvedená v definici 2.5.1 je slabší než podmínka živosti uvedená v definici 2.2.3:
(4) $(\forall t \in T)(\forall M \in RS(M_0))(\exists M', M'' \in RS(M))[M'[t] > M'']$.
Silnější podmínka (4) však vyplývá z konjunkce podmínek (2) a (3).
- Dále bude vyložena metoda konstrukce Petriho sítě, která již svou podstatou zaručuje správnost výsledné (libovolně složité) sítě. Obtížný test správnosti výsledné sítě je pak zbytečný.

Definice 2.5.2:

Elementární síť je síť sestávající z jediného místa obsahující jeden token, z jediného přechodu, z hrany od místa k přechodu a od přechodu k místu, tj. síť $\langle P, T, I, O, M_0 \rangle$, kde $P = \{p\}$, $T = \{t\}$, $I(t) = O(t) = \{p\}$, $M_0(p) = 1$ (viz obr.2.6.5).

p-modul (p-blok) je síť $\langle P, T, I, O, M_0 \rangle$, který má vyčleněny dvě neprázdné podmnožiny množiny míst P : množinu vstupních míst P_{in} a množinu výstupních míst P_{out} . Předpokládá se $(\forall p \in P_{in})\{M(p) = 0\}$.

t-modul (t-blok) je síť $\langle P, T, I, O, M_0 \rangle$, který má vyčleněny dvě neprázdné podmnožiny množiny přechodů T : množinu vstupních přechodů T_{in} a množinu výstupních přechodů T_{out} .

Substituce p-modulu $N = \langle P, T, I, O, M_0 \rangle$ za místo $p_s \in P'$ sítě

$$N' = \langle P', T', I', O', M_0' \rangle$$

je operace, která vytváří síť

$$N'' = \langle P'', T'', I'', O'', M_0'' \rangle,$$

kde

$$\begin{aligned} P'' &= (P' - \{p_s\}) \cup P, \\ T'' &= T' \cup T, \\ I''(t) &= I(t) && \text{pro } t \in T, \\ &= I'(t) && t \in T' \wedge t \notin p_s^\bullet, \\ &= P_{\text{out}} && t \in T' \wedge t \in p_s^\bullet, \\ O''(t) &= O(t) && \text{pro } t \in T, \\ &= O'(t) && t \in T' \wedge t \notin p_s, \\ &= P_{\text{in}} && t \in T' \wedge t \in p_s, \\ M''(p) &= M(p) && \text{pro } p \in (P - P_{\text{in}}), \\ &= M'(p) && p \in (P' - \{p_s\}), \\ &= M'(p_s) && p \in P_{\text{in}}. \end{aligned}$$

Substituce t-modulu $N = \langle P, T, I, O, M_0 \rangle$ za přechod $t_s \in T$ sítě

$$N' = \langle P', T', I', O', M_0' \rangle$$

je operace, která vytváří síť

$$N'' = \langle P'', T'', I'', O'', M_0'' \rangle,$$

kde

$$\begin{aligned} P'' &= P' \cup P, \\ T'' &= (T' - \{t_s\}) \cup T, \\ I''(t) &= I'(t) && \text{pro } t \in T', \\ &= I(t) && t \in T \wedge t \notin T_{\text{in}}, \\ &= t_s^\bullet && t \in T \wedge t \notin T_{\text{in}}, \\ O''(t) &= O'(t) && \text{pro } t \in T', \\ &= O(t) && t \in T \wedge t \notin T_{\text{out}}, \\ &= t_s^\bullet && t \in T \wedge t \in T_{\text{out}}, \\ M''(p) &= M(p) && \text{pro } p \in P, \\ &= M'(p) && p \in P'. \end{aligned}$$

p-modul (t-modul) je **správný**, jestliže jeho substituce do elementární sítě vytváří správnou síť.

Poznámky 2.5.2:

- Formálně můžeme p-modul definovat jako trojici

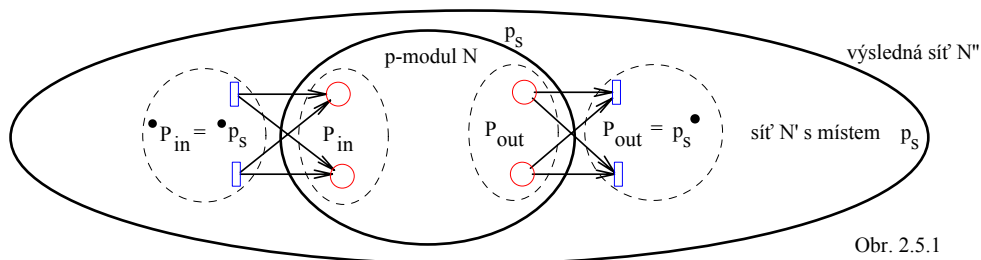
$$\langle \langle P, T, I, O, M_0 \rangle, P_{\text{in}}, P_{\text{out}} \rangle$$

a t-modul jako trojici

$$\langle \langle P, T, I, O, M_0 \rangle, T_{\text{in}}, T_{\text{out}} \rangle,$$

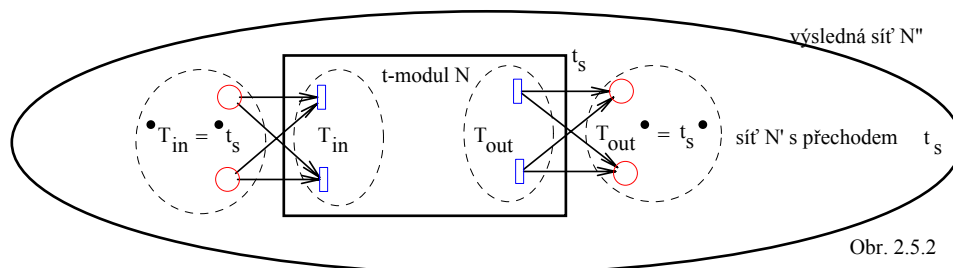
kde $\langle P, T, I, O, M_0 \rangle$ je obyčejná Petriho síť a $P_{\text{in}}, P_{\text{out}} \in P$, $T_{\text{in}}, T_{\text{out}} \in T$.

- Neformálně vyjádřeno, operace substituce p-modulu odstraňuje ze sítě N' místo p_s a nahrazuje jej p-modulem N tak, že všichni bezprostřední předchůdci místa p_s se stanou vstupními přechody pro libovolné vstupní místo p-bloku N a všichni bezprostřední následníci místa p_s se stanou výstupními přechody pro libovolné koncové místo p-bloku N . Přitom značení všech míst zůstane beze změny s výjimkou vstupních míst substituujícího p-bloku N , která všechna převezmou značení substituovaného místa p_s . Situace je zobrazena na obr.2.5.1.



Obr. 2.5.1

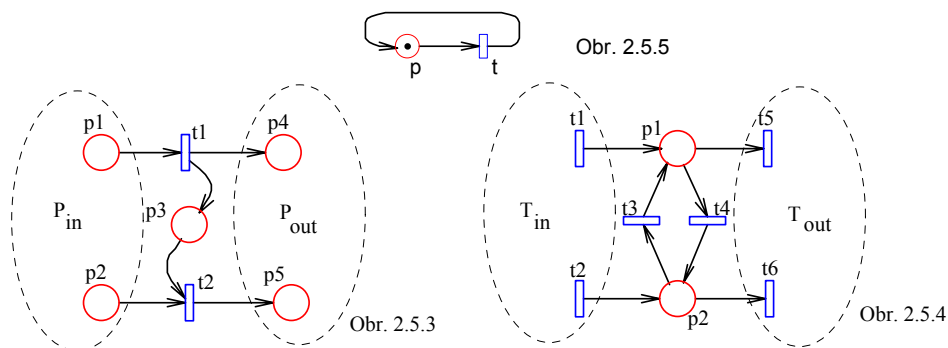
3. Podobně, operace substituce t-modulu odstraňuje ze sítě N' přechod t_s a nahrazuje jej t-modulem N tak, že všichni bezprostřední předchůdci přechodu t_s se stanou vstupními místy pro libovolný vstupní přechod t-modulu N a bezprostřední následníci přechodu t_s se stanou výstupními místy pro libovolný výstupní přechod t-modulu N . Přitom značení všech míst sítě N' , jakož i bloku N , zůstane beze změny. Situace je zobrazena na obr.2.5.2.



Obr. 2.5.2

Příklad 2.5.1:

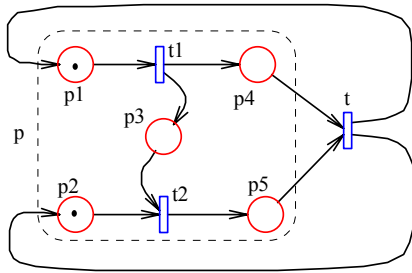
Na obr.2.5.3 je zobrazen p-modul (s vyznačenými množinami P_{in}, P_{out}) a na obr.2.5.4 t-modul (s vyznačenými množinami T_{in}, T_{out}). Zjistíme, zda zobrazené moduly jsou správné. Po dosazení p-modulu do elementárního cyklu (viz obr.2.5.5) za místo p , získáme síť zobrazenou na obr.2.5.6 a podobně po dosazení t-modulu za přechod t elementárního cyklu získáme síť zobrazenou na obr.2.5.7. Analýza těchto sítí je provedena v tabulkách tab.2.5.1-2. Příslušné grafy dosažitelnosti jsou zobrazeny na obr.2.5.8 a 2.5.9. Z těchto grafů vyplývá, že obě sítě z obr.2.5.6-7 jsou správné a tedy také oba moduly z obr.2.5.3-4 jsou rovněž správné.



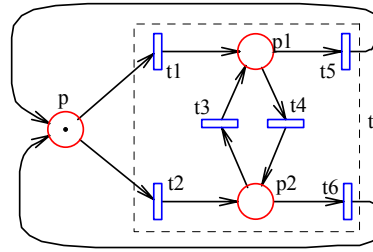
Obr. 2.5.3

Obr. 2.5.4

Obr. 2.5.5



Obr. 2.5.6



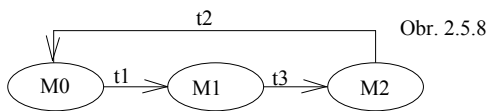
Obr. 2.5.7

| | t1 | t2 | t | M0 | M1 | M2 |
|----|----|----|----|-------|-------|------|
| p1 | -1 | 0 | 1 | 1 | 0 | 0 |
| p2 | 0 | -1 | 1 | 1 | 1 | 0 |
| p3 | 1 | -1 | 0 | 0 | 1 | 0 |
| p4 | 1 | 0 | -1 | 0 | 1 | 1 |
| p5 | 0 | 1 | -1 | 0 | 0 | 1 |
| | | | | t1→M1 | t2→M2 | t→M0 |

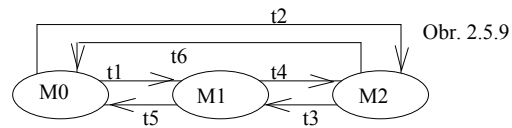
Tab. 2.5.1

| | t1 | t2 | t3 | t4 | t5 | t6 | M0 | M1 | M2 |
|----|----|----|----|----|----|----|----------------|----------------|----------------|
| p | -1 | -1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| p1 | 1 | 0 | 1 | -1 | -1 | 0 | 0 | 1 | 0 |
| p2 | 0 | 1 | -1 | 1 | 0 | -1 | 0 | 0 | 1 |
| | | | | | | | t1→M1 t2→M2 | t4→M2 t5→M0 | t3→M1 t6→M0 |

Tab. 2.5.2



Obr. 2.5.8



Obr. 2.5.9

Definice 2.5.3:

pt-sít' je induktivně definována takto:

- (1) Elementární síť je pt-sít'.
- (2) Síť vzniklá z pt-sítě substitucí správného p-modulu je rovněž pt-sítí.
- (3) Síť vzniklá z pt-sítě substitucí správného t-modulu je rovněž pt-sítí.
- (4) Jiné pt-sítě, než podle bodů (1),(2),(3), nejsou.

Věta 2.5.1:

Petriho síť je pt-sítí. ⇔ Petriho síť je správná.

Důkaz:

1. Nejdříve dokážeme, že každá Petriho síť, která je pt-sítí, je správná. K tomu je zapotřebí dokázat:
 - (1) Elementární síť (viz obr.2.5.5) je správná.
 - (2) Substitucí správného p-modulu za místo správné sítě vznikne správná síť.
 - (3) Substitucí správného t-modulu za přechod správné sítě vznikne správná síť.

Ad (1):

Bezpečnost, živost a reverzibilita elementární sítě je bezprostředně zřejmá.

Ad (2):

Dokážeme, že substitucí správného p-modulu N za místo p_s správné sítě N' vznikne správná síť N'' - viz situace zobrazená na obr.2.6.1.

1. N' je správná síť. (předpoklad)
2. N je správný p-modul. (předpoklad)
3. N' je bezpečná. (plyne z 2.)
4. $M'(p_s)=1 \Rightarrow$ žádný přechod z $\bullet p_s$ nemůže být proveden dříve než některý z přechodů z $p_s \bullet$ (plyne z 3.)
5. Síť, která vznikne po dosazení p-modulu N do elementární sítě, je bezpečná, reverzibilní a živá. (plyne z 2.)
6. Při provádění p-modulu N se tokeny přesunou z P_{in} do P_{out} tak, že počet tokenů v žádném místě modulu není nikdy větší než 1.
7. Síť N'' je bezpečná (plyne z 3.,4.,6.)
8. N' je reverzibilní (plyne z 1.)
9. $(\forall M' \in RS(M'_0))[M' \rightarrow M'']$, tj. každé dosažitelné značení může být v síti N' vždy znovu dosaženo. (plyne z 8.)
10. $M'(p_s)=1 \Rightarrow (M''_{out} \rightarrow M''_{in})$, kde M''_{out} (resp. M''_{in}) je značení sítě N'' shodné se značením M' sítě N' s tím, že všechna místa z P_{out} (resp. P_{in}) obsahují tokeny (plyne z 9.).
11. $(\forall M \in RS(M_{in}))[M \rightarrow M_{out}]$, tj. z každého dosažitelného značení M bloku N lze dosáhnout k oncového značení M_{out} (všechny tokeny jsou v P_{out}) (plyne z 5.).
12. Síť N'' je reverzibilní (plyne z 10.,11.)
13. Síť N' je živá. (plyne z 1.)
14. Síť N'' je živá. (plyne z 12.,13.,5.)
15. Síť N'' je správná. (plyne z 7.,12.,14. - mělo být dokázáno)

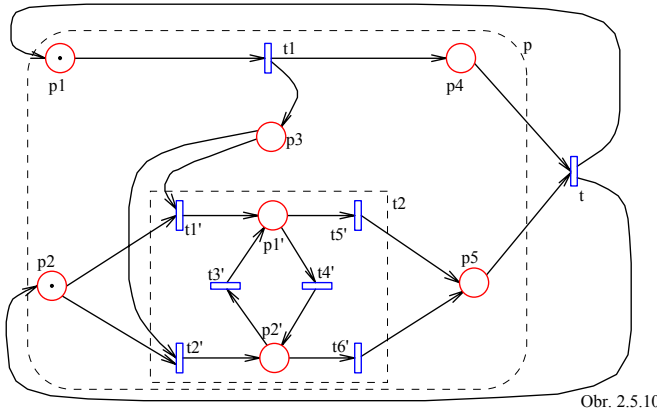
Ad (3):

O obdobně jako v případě (2).

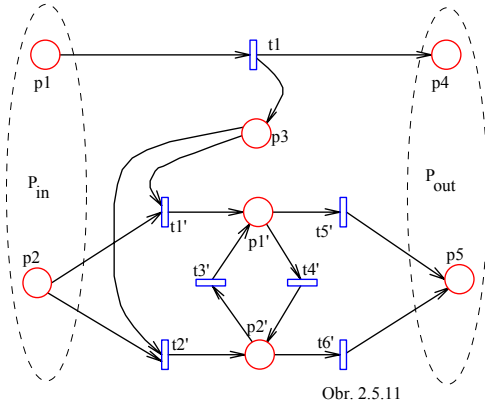
2. Dokážeme, že každá správná síť N je pt-sítí. Nalezneme v síti N takový přechod t , jehož všechna výstupní místa obsahují token a jehož všechna vstupní místa token neobsahují. Takový přechod vždy existuje aspoň jeden, protože každý přechod správné sítě je vždy proveditelný a popsaná situace je situací po provedení přechodu t . Zbylou část sítě (vzniklou ze sítě N odebráním přechodu t a hran s ním incidentních) vytváří p-modul, který je správný, protože jeho dosazení do elementární sítě vytváří správnou síť N . Síť N tedy vznikl adosazením správného modulu do pt-sítě (elementární sítě) a je tedy pt-sítí.

Příklad 2.5.2:

Na obr.2.5.10 je ilustrován způsob konstrukce správných sítí pomocí pt-sítí. Vycházíme z elementární sítě zobrazené na obr.2.5.5. Substitucí p-modulu, definovaného obrázkem 2.5.3, za místo p elementární sítě získáme síť zobrazenou na obr.2.5.6. Substitucí t -bloku, definovaného obrázkem 2.5.4, za místo t_2 sítě z obr.2.5.6 získáme síť zobrazenou na obr.2.5.10 (tímto způsobem můžeme pokračovat dále, tj. v síti z obr.2.5.10 substituovat za zvolené místo nebo přechod p- nebo t-modul, atd.). Jestliže jsme vždy do správné sítě substituovali správný modul je výsledkem opět správná síť. Anž bychom síť na obr.2.5.10 analyzovali, můžeme prohlásit, že je správná a rovněž také p-modul na obr.2.5.11.



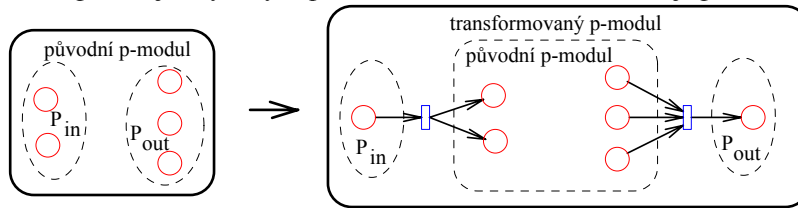
Obr. 2.5.10



Obr. 2.5.11

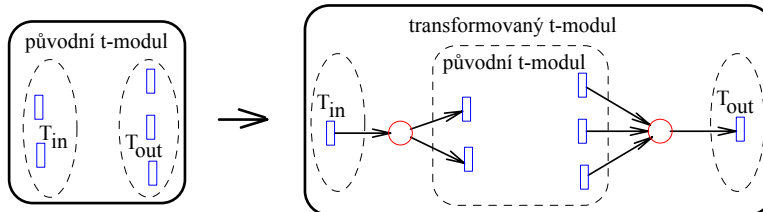
Poznámky 2.5.3:

1. Každý p-modul lze převést na ekvivalentní **bipolární p-modul**, tj. na p-modul vlastností $|P_{in}|=1 \wedge |P_{out}|=1$ (s jediným vstupním a jediným výstupním místem). Tato transformace je předvedena na obr.2.5.12.



Obr. 2.5.12

2. Každý t-modul lze převést na ekvivalentní **bipolární t-modul**, tj. na t-modul vlastností $|T_{in}|=1 \wedge |T_{out}|=1$ (s jediným vstupním a jediným výstupním přechodem). Tato transformace je předvedena na obr.2.5.13.



Obr. 2.5.13

Věta 2.5.2:

P-modul $N = \langle \langle P, T, I, O, M_0 \rangle, P_{in}, P_{out} \rangle$ je správný, jestliže platí současně:

- (1) p-modul N je acyklickou (tj. neobsahující cykl) synchronizační sítí,
- (2) počáteční značení všech míst je nulové, tj. $(\forall p \in P)[M_0(p)=0]$,
- (3) vstupní místa modulu nemají v síti N žádné vstupní přechody, tj. $(\forall p \in P_{in})[p^\bullet = \emptyset]$,
- (4) výstupní místa modulu nemají v síti N žádné výstupní přechody, tj. $(\forall p \in P_{out})[p^\bullet = \emptyset]$,
- (5) každé místo a každý přechod leží na orientované cestě, která vede z některého počátečního do některého koncového místa, tj.

$$(\forall v \in (P \cup T)(\exists p_r \in P_{in})(\exists p_s \in P_{out})[p_r \rightarrow v \rightarrow p_s])$$

kde zápis $x \rightarrow y$ značí, že z uzlu x vede do uzlu y v síti $\langle P, T, I, O \rangle$ orientovaná cesta.

Důkaz:

Nechť p-modul $\langle\langle P, T, I, O, M_0 \rangle, P_{in}, P_{out}\rangle$ splňuje podmínky (1)-(5). Dosadíme tento modul do elementární sítě a dokážeme, že tak vznikne správná síť. Po dosazení obsahují všechna vstupní místa p-modulu po jednom tokenu. Vzhledem k tomu, že p-modul je synchronizační síť, nemůže docházet ke konfliktům při provádění přechodů. Protože se jedná o acyklickou síť, nemohou se tokeny "vracet", ale mohou se pouze pohybovat "zleva doprava", dokud neskončí ve výstupních místech p-modulu. Přitom vzhledem k vlastnosti (5) musí každým místem a každým přechodem p-bloku "projít" nějaký token. Po provedení přechodu výchozí elementární síť se vrací celá síť do původního značení. Každá cesta z počátečního do koncového místa se po provedení tohoto přechodu uzavírá v cykl a v každém cyklu je počet tokenů konstantní a roven jedné. Celá síť, vzniklá dosazením p-bloku do elementární sítě je tedy bezpečná, reverzibilní a živá, tj. správná.

Věta 2.5.3:

T-modul $N = \langle\langle P, T, I, O, M_0 \rangle, T_{in}, T_{out}\rangle$ je správný, jestliže platí současně:

- (1) t-modul N je automatovou sítí,
- (2) počáteční značení všech míst je nulové, tj.

$$(\forall p \in P)[M_0(p)=0],$$
- (3) vstupní přechody modulu nemají v síti N žádná vstupní místa, tj.

$$(\forall t \in T_{in})[t^\bullet = \emptyset],$$
- (4) výstupní přechody modulu nemají v síti N žádná výstupní místa, tj.

$$(\forall t \in T_{out})[t^\bullet = \emptyset],$$
- (5) každý přechod a každé místo leží na orientované cestě, která vede z některého počátečního do některého koncového přechodu, tj.

$$(\forall v \in (P \cup T)(\exists t_r \in T_{in})(\exists t_s \in T_{out})[t_r \rightarrow v \rightarrow t_s],$$

kde zápis $x \rightarrow y$ značí, že z uzlu x vede do uzlu y orientovaná cesta.

Důkaz:

Nechť t-modul $\langle\langle P, T, I, O, M_0 \rangle, T_{in}, T_{out}\rangle$ splňuje podmínky (1)-(5). Dosadíme tento modul do elementární sítě a ukážeme, že tak vznikne správná síť. Dosazením vznikne silně souvislá automatová síť, ve které pouze jediné místo - místo výchozí elementární sítě - obsahuje token. Do substituovaného t-modulu vnikne prostřednictvím některého vstupního přechodu modulu a vzhledem ke konzervativnosti automatové sítě, bude se v síti nacházet vždy jen jeden token. Tento jediný token může t-modul opustit pouze prostřednictvím nějakého výstupního přechodu t-bloku. Síť je tedy bezpečná a reverzibilní. Vzhledem k vlastnosti (5) každý přechod může být opakovaně proveden. Síť je tedy i živá. Podle definice je tedy t-blok bezpečný, živý a reverzibilní, tj. správný.

Poznámky 2.5.4:

1. Provéřit podmínky (1)-(5) věty 2.5.2 nebo věty 2.5.3 je zpravidla podstatně snazší a rychlejší než prověřit správnost bloku na základě definice a pomocí konstrukce grafu dosažitelnosti (viz příklad 2.5.1 a 2.5.3).
2. Obrácená tvrzení k tvrzením vět 2.5.2 a 2.5.3 neplatí. Podmínky (1)-(5) jsou pro správnost bloku pouze postačující, nikoliv nutné.

Příklad 2.5.3:

P-modul z obr.2.5.3 je správný modul podle věty 2.5.2 a t-modul z obr.2.5.4 je správný modul podle věty 2.5.3.

Věta 2.5.4:

Bipolární p-blok $N = \langle\langle P, T, I, O, M_0 \rangle, p_{in}, p_{out}\rangle$ je správný, jsou-li splněny následující podmínky:

- (1) síť N je automatovou sítí,
- (2) počáteční značení všech míst je nulové, tj.

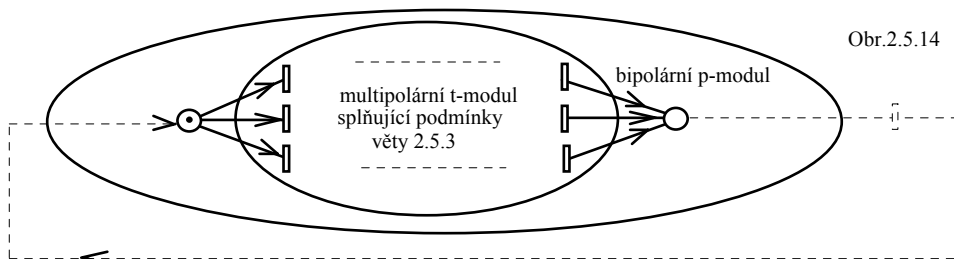
$$(\forall p \in P)[M_0(p)=0],$$

- (3) každý uzel sítě (přechod nebo místo) leží na nějaké cestě z počátečního do koncového místa, tj.

$$(\forall v \in (P \cup T))[p_{in} \rightarrow v \rightarrow p_{out}].$$

Důkaz:

Vydělíme-li z bipolárního p-modulu vstupní a výstupní místo, získáme multipolární t-modul - viz obr.2.5.1
 4. Splňuje-li p-modul podmínky



(1)-(3) věty 2.5.4, pak vnitřní multipolární t-modul splňuje podmínky (1)-(5) věty 2.5.3. Vnější bipolární p-modul je potom správný, neboť po dosazení do elementární sítě vytváří správnou síť.

Věta 2.5.5:

Bipolární t-modul $N = \langle \langle P, T, I, O, M_0 \rangle, t_{in}, t_{out} \rangle$ je správný, jsou-li splněny následující podmínky:

- (1) síť N je acyklickou synchronizační sítí,
 (2) počáteční značení všech míst je nulové, tj.

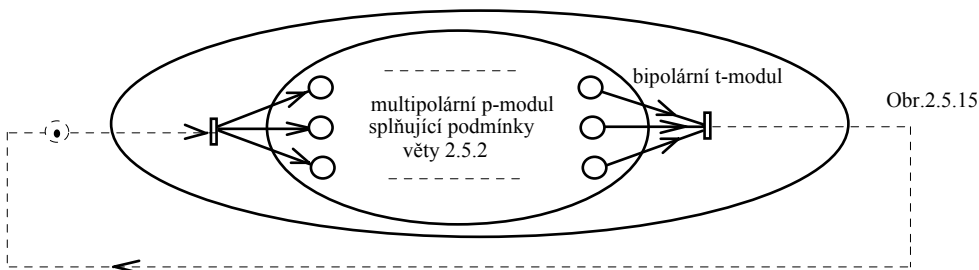
$$(\forall p \in P)[M_0(p)=0],$$

- (3) každý uzel (přechod nebo místo) sítě leží na nějaké cestě z počátečního do koncového přechodu, tj.

$$(\forall v \in (P \cup T))[t_{in} \rightarrow v \rightarrow t_{out}].$$

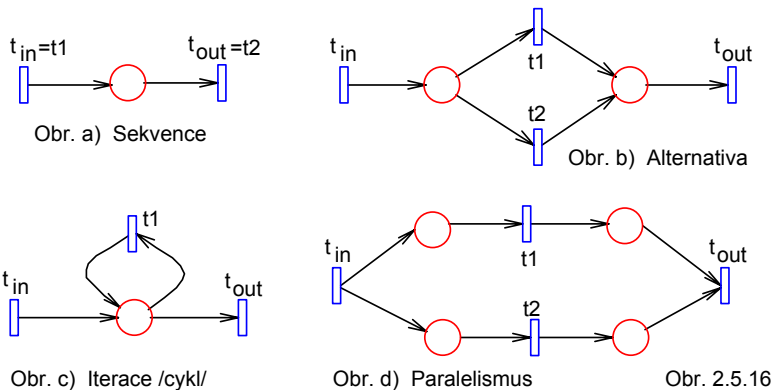
Důkaz:

Vydělíme-li z bipolárního t-modulu vstupní a výstupní přechod, získáme multipolární p-modul - viz obr.2.5.15. Splňuje-li t-modul podmínky (1)-(3) věty 2.5.5, pak vnitřní multipolární p-modul splňuje podmínky (1)-(5) věty 2.5.2. Vnější bipolární t-modul je správný, neboť po dosazení do elementární sítě vytváří korektní síť.



Příklad 2.5.4:

Na obr.2.5.16 jsou zobrazeny čtyři bipolární t-moduly představující základní řídicí programové konstrukty: sekvenci, alternativu, iteraci a paralelismus. Všechny tyto t-moduly jsou správnými moduly, což zjistíme buď podle definice správného modulu a nebo rychleji podle věty 2.5.5 (v případě sekvence a paralelismu) a nebo podle věty 2.5.3 (v případě alternativy a iterace).



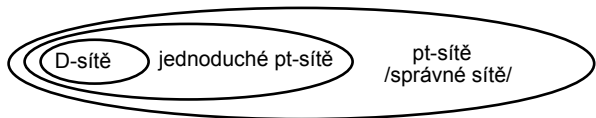
Definice 2.5.4:

Pt-sítě, které vzniknou z elementární sítě opakovanými substitucemi čtyř t-modulů uvedených v příkladu 2.5.4 nazýváme **D-sítěmi**. P-bloky, které splňují podmínky věty 2.5.2 nebo 2.5.4 nazýváme **jednoduchými p-moduly** a t-bloky, které splňují podmínky věty 2.5.3 nebo 2.5.5 nazýváme **jednoduchými t-moduly**.

pt-sítě, které vzniknou z elementární sítě pouze substitucemi jednoduchých p-modulů a t-modulů nazýváme **jednoduchými pt-sítěmi**.

Poznámky 2.5.5:

1. Strukturované paralelní programy (zobecnění strukturovaných sekvenčních programů ve smyslu Duikstrově) mohou být reprezentovány D-sítěmi.
2. Na obr.2.5.17 je zobrazen vztah mezi třídami sítí z definice 2.5.4.



Obr. 2.5.17

3. Petriho sítě a jazyky

3.1. Jazyky Petriho sítí

Nejdříve připomeňme několik pojmů z teorie jazyků a automatů.

Definice 3.1:

Jazyk L nad abecedou A je libovolně zvolená podmnožina množiny všech slov nad touto abecedou, tj. $L \subseteq A^*$, kde $A^* = A^0 \cup A^1 \cup A^2 \cup A^3 \cup \dots$, $A^0 = \{e\}$.

Konečný automat FA (finite automaton) je pětice $\langle Q, A, \delta, q_0, F \rangle$, kde Q je neprázdná konečná množina stavů, A neprázdná konečná množina vnějších symbolů (vnější abeceda), δ je zobrazení $Q \times A \rightarrow Q$ (přechodová funkce), $q_0 \in Q$ (počáteční stav) a $F \subseteq Q$ (množina koncových stavů).

Jazyk rozpoznávaný konečným automatem FA = $\langle Q, A, \delta, q_0, F \rangle$ je podmnožina $L(\text{FA})$ množiny všech slov nad abecedou A definovaná takto:

$$L(\text{FA}) = \{w \in A^* : \delta^*(q_0, w) \in F\},$$

kde δ^* je zobrazení $Q \times A^* \rightarrow Q$ (zobecněná přechodová funkce odpovídající přechodové funkci δ).

Definice 3.2:

Ohodnocená Petriho síť (labeled PN - LPN) je trojice

$$\langle \langle P, T, I, O, M_0 \rangle, A, \lambda, M_f \rangle,$$

kde $\langle P, T, I, O, M_0 \rangle$ je OPN nebo GPN (obyčejná nebo zobecněná Petriho síť - viz definice 2.4.1 a věta 2.5.1), A je abeceda, tj. neprázdná konečná množina symbolů, λ je zobrazení $T \rightarrow A \cup \{e\}$ zvané ohodnocení (labeling) přechodu a $M_f \subseteq \text{RS}(M_0)$ je podmnožina množiny dosažitelných značení, tzv. množina koncových značení.

Jazyk rozpoznávaný ohodnocenou Petriho sítí LPN = $\langle \langle P, T, I, O, M_0 \rangle, A, \lambda, M_f \rangle$ je podmnožina množiny všech slov nad abecedou A definovaná takto:

$$L(\text{LPN}) = \{w \in A^* : (\exists \sigma \in T^*) (\exists M \in M_f) [w = \lambda^*(\sigma) \wedge M_0[\sigma \triangleright M]]\},$$

kde λ^* je zobecněné zobrazení λ , tj. zobrazení $T^* \rightarrow A^*$ definované takto:

$$\text{je-li } \sigma = t_{(1)}t_{(2)}\dots t_{(k)}, \text{ pak } \lambda^*(\sigma) = \lambda(t_{(1)})\lambda(t_{(2)})\dots\lambda(t_{(k)}).$$

Poznámky 3.2.1:

1. Některé speciální případy definice ohodnocení přechodů:

- $A = T$ a λ je identické zobrazení.
- λ je obecné zobrazení $T \rightarrow A$ (různé přechody mohou mít i stejná ohodnocení).
 - λ je injektivní zobrazení $T \rightarrow A$, tj. platí:

$$t \neq t' \Rightarrow \lambda(t) \neq \lambda(t')$$
 (různé přechody mají vždy různá ohodnocení).
- λ je obecné zobrazení $T \rightarrow A \cup \{e\}$ (přechody mohou být ohodnoceny zcela libovolně, některé přechody mohou být dokonce bez ohodnocení).

Pokud ohodnocovací funkce $\lambda(t)$ není blíže specifikována, předpokládáme poslední nejobecnější případ.

2. Některé speciální případy definice množiny koncových značení. Množina koncových značení M_f není libovolně zvolenou podmnožinou množiny $\text{RS}(M_0)$ všech dosažitelných značení, ale splňuje ještě jistou dodatečnou podmínku.

- **Jazyk typu G.** Spolu s každým značením patří do koncové množiny M_f i všechna větší značení, tj.

$$M' \in M_f \wedge M'' \triangleright M' \Rightarrow M'' \in M_f.$$

- **Jazyk typu T.** Množina koncových značení je rovna množině uzamčení (deadlocků) Petriho sítě, tj.

$$M \in M_f \Leftrightarrow E(M) = \emptyset.$$

- **Jazyk typu P.** Množina koncových značení splývá s množinou všech dosažitelných značení, tj.

$$M_f = RS(M_0).$$

Tento jazyk se vyznačuje vlastností

$$uv \in L \Rightarrow u \in L$$

(kde u, v jsou slova v abecedě A) a nazývá se proto **prefixovým jazykem**.

Pokud typ jazyka není blíže specifikován, předpokládáme obecný případ, kdy na množinu koncových stavů M_f neklademe žádné podmínky. V tomto případě hovoříme o **jazyku typu L**.

3. Různé alternativy ohodnocení přechodů (viz pozn.1) se mohou libovolně kombinovat s různými alternativami definice množiny koncových značení (viz pozn.2).
4. Množina všech sekvencí (posloupností přechodů) spustitelných z počátečního značení představuje prefixový jazyk nad abecedou T rozpoznávaný Petriho sítí $\langle P, T, I, O, M_0 \rangle$. Tento jazyk charakterizuje všechna možná chování uvažované Petriho sítě.
5. V souvislosti s rozpoznáváním jazyků Petriho sítěmi vznikají dvě úlohy:
 - K dané (značené) Petriho sítí nalézt jazyk, který je touto sítí rozpoznáván, tj. charakterizovat tento jazyk generativní gramatikou, regulárním výrazem či nějakým jiným způsobem.
 - K danému jazyku (charakterizovanému gramatikou, regulárním výrazem, výčtem vlastností) nalézt značenou Petriho síť, která jej rozpoznává.

Definice 3.3:

Standardní počáteční značení ohodnocené Petriho sítě

$$\langle \langle P, T, I, O, M_0 \rangle, A, \lambda, M_f \rangle$$

je značení M_0 s následující vlastností

$$(\exists p_s \in P)[M_0(p_s) = 1 \wedge (\forall p \in (P - \{p_s\}))[M_0(p) = 0]],$$

tj. při počátečním značení M_0 existuje místo p_s , které jako jediné obsahuje token (jediný), zatímco všechna ostatní místa jsou bez tokenů.

Standardní množina koncových značení ohodnocené Petriho sítě je pak množina M_f s následujícími vlastnostmi

$$M_f = \{ M_f \} \wedge (\exists p_f \in P)[M_f(p_f) = 1 \wedge (\forall p \in (P - \{p_f\}))[M_f(p) = 0]],$$

tj. množina M_f koncových značení obsahuje pouze jediné značení M_f a to takové, že při něm jen jediné místo p_f obsahuje token (jediný) a všechna ostatní místa jsou bez tokenů.

Ohodnocená Petriho síť je ve **standardním tvaru**, jestliže jak její počáteční značení, tak i množina koncových značení, jsou standardní.

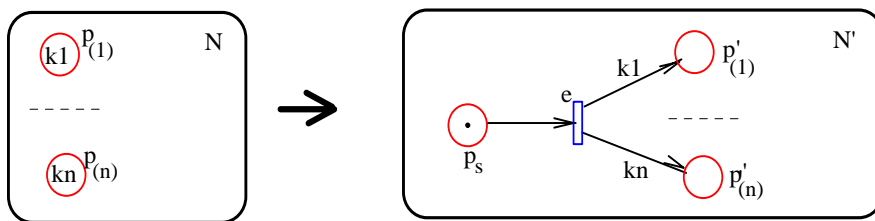
Věta 3.1.1:

Ke každé ohodnocené Petriho sítí N existuje ekvivalentní ohodnocená Petriho síť N' ve standardním tvaru. Sítě jsou ekvivalentní v tom smyslu, že rozpoznávají stejný jazyk, tj. platí

$$L(N) = L(N').$$

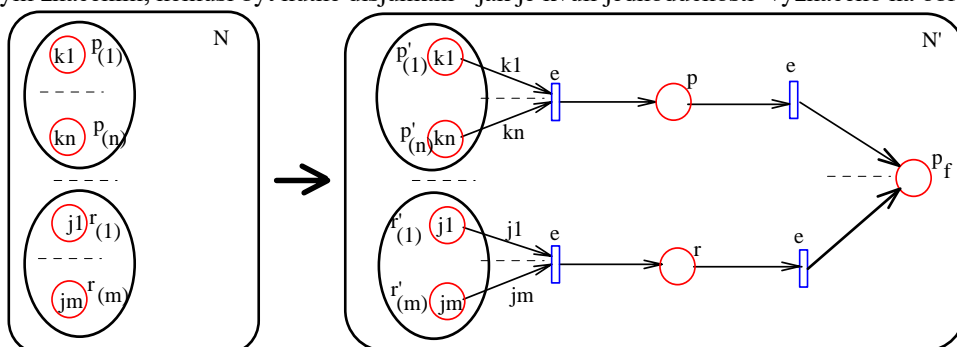
Důkaz:

Snadno se nahlédne, že síť N' získáme ze sítě N následujícími dvěma úpravami. Na obr.3.1.1 je ilustrována prvá konstrukce převádějící obecně nestandardní počáteční značení na standardní. Předpokládá se, že při počátečním značení sítě N jsou tokeny pouze v místech $p_{(1)}, \dots, p_{(n)}$ a to v počtu k_1, \dots, k_n . V ekvivalentní síti N' představují čísla k_1, \dots, k_n násobnosti hran. Provedením přechodu za místem p_s vznikne v síti N' situace odpovídající počáteční situaci v síti N .



Obr. 3.1.1

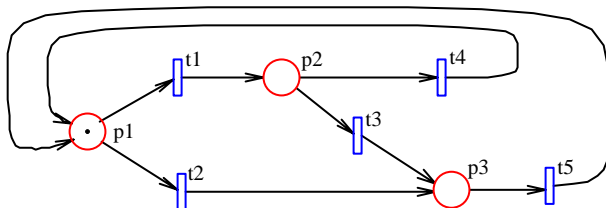
Podobně na obr.3.1.2 je ilustrována druhá konstrukce převádějící obecně nestandardní množinu koncových značení na standardní. Na obr. jsou v síti N schematicky zobrazena koncová značení vyjádřená skupinami míst, která obsahují potřebné nenulové počty tokenů (např. $p_{(1)}, \dots, p_{(n)}$ s k_1, \dots, k_n tokeny). Síť N' představuje rozšíření sítě N , které převádí různá nestandardní koncová značení sítě N na jediné standardní koncové značení sítě N' (reprezentované koncovým místem p_f). Poznamenejme ještě, že množiny míst $p_{(1)}, \dots, p_{(n)}$ a $r_{(1)}, \dots, r_{(m)}$, odpovídající různým koncovým značením, nemusí být nutně disjunktní - jak je kvůli jednoduchosti vyznačeno na obrázku.



Obr. 3.1.2

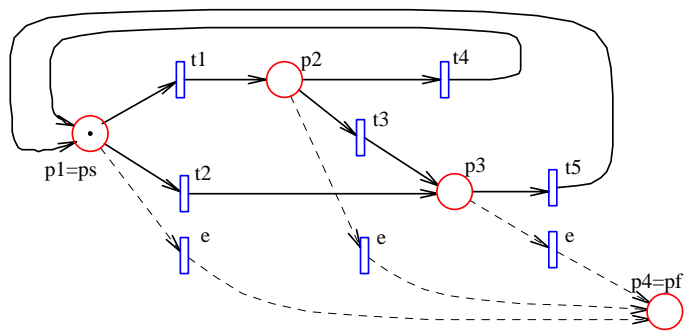
Příklad 3.1.1:

Popíšeme jazyk charakterizovaný Petriho sítí zadanou obrázkem 3.1.3. Síť je neznačená, ale můžeme ji chápat jako identicky značenou ($A=T$ a λ je identické zobrazení). Na obrázku je zobrazeno počáteční značení, které je standardní. Za koncové značení můžeme definovat opět toto značení. Máme tak ohodnocenou Petriho síť ve standardním tvaru. Jazyk (nad abecedou $\{t_1, t_2, t_3, t_4, t_5\}$) rozpoznávaný touto sítí je dán regulárním výrazem $(t_1(t_4+t_3t_5)+t_2t_5)^*$.



Obr. 3.1.3

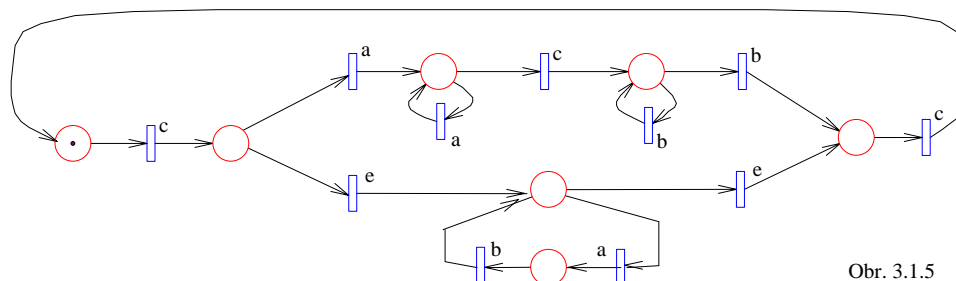
Zvolíme-li za množinu koncových značení množinu všech dosažitelných značení $RS(M_0)$ (prefixový jazyk, jazyk typu P), pak do jazyka rozpoznávaného Petriho sítí patří všechna slova popsaná regulárním výrazem $(t_1(t_4+t_3t_5)+t_2t_5)^*$ spolu se všemi předponami (prefixy) těchto slov. Rozpoznávající Petriho síť můžeme transformovat do standardního tvaru pomocí konstrukce popsané v důkazu věty 3.1.1 - viz obr.3.1.4.



Obr. 3.1.4

Příklad 3.1.2:

Regulární jazyk nad abecedou $A=\{a,b,c\}$ charakterizovaný regulárním výrazem $c(a^+cb^+(ab)^*)c$ vyjádříme pomocí Petriho sítě, která tento jazyk rozpoznává. Rozpoznávající síť ve standardním tvaru je na obr.3.1.5. Způsob konstrukce vyplývá okamžitě ze sémantiky pojmů regulární výraz a značená Petriho síť.



Obr. 3.1.5

3.2. Jazyky PN a Chomského hierarchie

Nejprve připomeneme několik základních pojmů z teorie jazyků.

Definice 3.2.1:

(**Generativní**) **gramatika** je čtveřice $G = \langle N, T, S, P \rangle$, kde N a T jsou dvě disjunktní konečné abecedy, $S \in N$ a P je konečná podmnožina kartézského součinu $(N \cup T)^* N (N \cup T)^* \times (N \cup T)^*$. Množina N se nazývá **abecedou neterminálních symbolů** (neterminálů, proměnných), množina T **abecedou terminálních symbolů** (terminálů, konstant) a S je **počáteční symbol**. Prvky množiny P jsou **přepisovací pravidla** gramatiky. Přepisovací pravidla mají tvar $\alpha \rightarrow \beta$ kde $\alpha, \beta \in (N \cup T)^*$ přičemž α obsahuje aspoň jeden neterminál.

Jazyk $L(G)$ **generovaný gramatikou** G je množina všech slov v terminální abecedě T , které lze odvodit (**derivovat**) z počátečního symbolu, tj.

$$L(G) = \{w \in T^* : S \Rightarrow^* w\}.$$

Podle Chomského klasifikujeme gramatiky (a jimi generované jazyky) v závislosti na tvaru přepisovacích pravidel takto: gramatiku $G = \langle N, T, S, P \rangle$ nazýváme

- **neomezenou (typu 0)**, jestliže na množinu přepisovacích pravidel neklademe žádná omezení, tj. mají-li přepisovací pravidla tvar

$$\alpha X \beta \rightarrow \gamma,$$

- **kontextovou (typu 1)**, jestliže všechna přepisovací pravidla mají tvar

$$\alpha X \beta \rightarrow \alpha \gamma \beta,$$

kde $|\gamma| > 0$ (jedinou výjimkou může být pravidlo $S \rightarrow e$, jehož výskyt v seznamu pravidel však znamená, že počáteční symbol S se nesmí nacházet na pravé straně žádného přepisovacího pravidla),

- **bezkontextovou (typu 2)**, jestliže všechna přepisovací pravidla mají tvar

$$X \rightarrow \gamma,$$

- **regulární (typu 3)**, jestliže přepisovací pravidla mají tvar

$$X \rightarrow wY \text{ nebo } X \rightarrow w.$$

Připomeňme, že $X, Y \in N$, $\alpha, \beta, \gamma \in (N \cup T)^*$, $w \in T^*$.

Příklady 3.2.1:

- $L_1 = \{a^n b^n : n > 0\}$, $L_2 = \{ww^R : w \in A^*\}$, ... bezkontextové, ale nikoliv regulární jazyky
- $L_3 = \{a^n b^n c^n : n > 0\}$, $L_4 = \{ww : w \in A^*\}$, $L_5 = \{a^n b^m c^n d^m : n > 0, m > 0\}$, ... kontextové, ale nikoliv bezkontextové jazyky

Definice 3.2.2:

PN-jazyk je jazyk, ke kterému existuje LPN (značená obyčejná nebo zobecněná Petriho síť), která jej charakterizuje (rozpoznává či generuje).

Věta 3.2.1:

Každý regulární jazyk je PN-jazyk.

Důkaz:

Důkaz můžeme vést trojím způsobem podle způsobu definice regulárního jazyka:

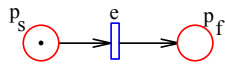
1. Regulární jazyk je jazyk, který lze získat z elementárních jazyků $(\emptyset, \{e\}, \{a\})$ užitím konečného počtu regulárních operací (sjednocení, zřetězení a iterací).
2. Regulární jazyk je jazyk, který je rozpoznatelný konečným automatem.
3. Regulární jazyk je jazyk, který je generovatelný regulární gramatikou.

1. důkaz (na základě 1. definice):

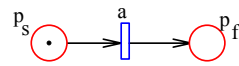
Na obr.3.2.1 jsou zobrazeny LPN ve standardním tvaru vyjadřující elementární jazyky $\{\}$, $\{e\}$, $\{a\}$ (prázdný jazyk, neprázdný jazyk tvořený prázdným slovem, jazyk tvořený jediným jednopísmenným slovem).



Obr. 3.2.1.a $\{\}$



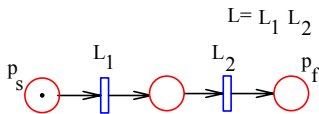
Obr. 3.2.1.b $\{e\}$



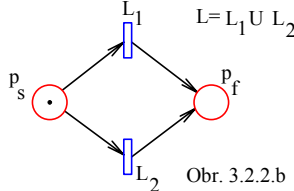
Obr. 3.2.1.c $\{a\}$

Na obr.3.2.2 jsou zobrazeny tři základní operace pomocí nichž lze konstruovat regulární jazyky:

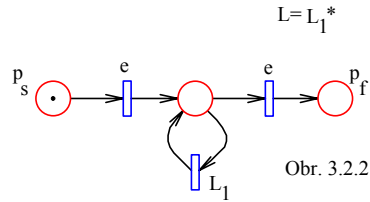
- zřetězení: $L=L_1L_2=\{w: w=uv, u \in L_1, v \in L_2\}$
- sjednocení: $L=L_1 \cup L_2=\{w: w \in L_1 \vee w \in L_2\}$
- iterace: $L=L_1^* = L_1^0 \cup L_1^1 \cup L_1^2 \cup L_1^3 \cup \dots = \{e\} \cup L_1 \cup L_1^2 \cup L_1^3 \cup \dots$



Obr. 3.2.2.a



Obr. 3.2.2.b



Obr. 3.2.2.c

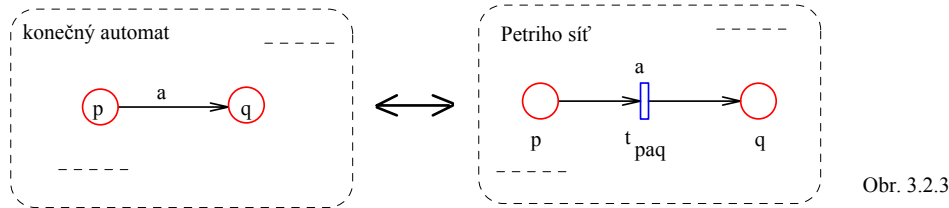
2.důkaz (na základě 2. definice):

K jazyku rozpoznávanému automatem $FM = \langle Q, A, \delta, q_0, F \rangle$ nalezneme značenou Petriho síť

LPN $= \langle \langle P, T, I, O, M_0 \rangle, A, \lambda, M_f \rangle$, která tento jazyk reprezentuje. Jednotlivé složky hledané Petriho sítě jsou definovány takto:

- $P=Q$,
- $T = \{t_{paq} : p, q \in Q, a \in A, q = \delta(p, a)\} \dots$ viz obr.3.2.3,
- $I(t_{paq}) = \{p\}$, $O(t_{paq}) = \{q\}$, $\lambda(t_{paq}) = a$,
- $M_0(q_0) = 1 \dots$ standardní počáteční značení,
- $M_f = \{M_f \in RS(M_0) : (\exists q \in F)[M_f(q) > 0]\}$.

Z definice T, I, O vyplývá, že Petriho síť je automatová a tedy konzervativní. Počáteční značení je standardní a tedy v síti je vždy jen jediný token.



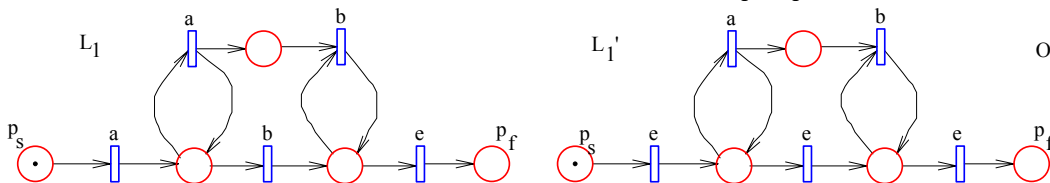
Obr. 3.2.3

Věta 3.2.2:

Ne každý PN-jazyk je regulární, tj. existuje PN-jazyk, který není regulární.

Důkaz:

Takovým jazykem je např. jazyk $L_1 = \{a^n b^n : n > 0\}$ nebo $L_1' = \{a^n b^n : n \geq 0\}$, kde n je libovolné celé nezáporné číslo. Tyto jazyky nejsou regulární protože nejsou rozpoznatelné žádným konečným automatem. Na obr.3.2.4 jsou zobrazeny značené Petriho sítě ve standardním tvaru generující jazyky L_1 a L_1' .



Obr. 3.2.4

Věta 3.2.3:

Ne každý bezkontextový jazyk je PN-jazykem, tj. existuje bezkontextový jazyk, který PN-jazykem není

Důkaz:

Dokážeme, že takovým jazykem je jazyk $L_2 = \{ww^R : w \in A^*\}$. Označme $|T|=m, |A|=k, |w|=r$. Počet různých slov délky r je k^r . Označme symbolem $RS_r(M_0)$ množinu všech stavů (značení) sítě, které jsou dosažitelné z počátečního stavu M_0 provedením sekvence přechodů o délce r . Má-li Petriho síť rozpoznávat uvedený jazyk musí platit

$$|RS_r(M_0)| \geq k^r, \tag{*}$$

neboť jinak by nebylo možné, aby si síť zapamatovala pomocí různého značení různá slova w délky r .

Zpracování slova w délky r znamená provedení sekvence přechodů $\sigma = t_{(1)} \dots t_{(r)}$, která převádí Petriho síť ze stavu M_0 do stavu M_r . Tedy platí (fundamentální rovnice)

$$M_r = M_0 + C \cdot V_\sigma$$

kde $V_\sigma = (v_1, v_2, \dots, v_m)^T$ je charakteristický vektor sekvence s vlastností $v_1 + v_2 + \dots + v_m = r$. Platí tedy:

$$\begin{aligned} |RS_r(M_0)| &\leq (\text{počet charakter.vektorů splňující vztah } v_1 + v_2 + \dots + v_m = r) = \\ &= (\text{počet rozkladů přirozeného čísla } r \text{ na } m \text{ přiroz. sčítanců}) = \\ &= (r+m-1) \cdot (r+m-2) \dots (r+1) / (m-1) \cdot (m-2) \dots 1 \leq \\ &\leq (r+m-1) \cdot (r+m-2) \dots (r+1) \leq \\ &\leq (r+m)^k \leq k^r. \end{aligned}$$

Poslední nerovnost interpretujeme ve smyslu $(\forall k, m)(\exists r)[(r+m)^k < k^r]$, což lze ověřit pomocí L'Hospitalova pravidla (exponenciální funkce roste rychleji než mocninná funkce). Platí tedy

$$|RS_r(M_0)| < k^r,$$

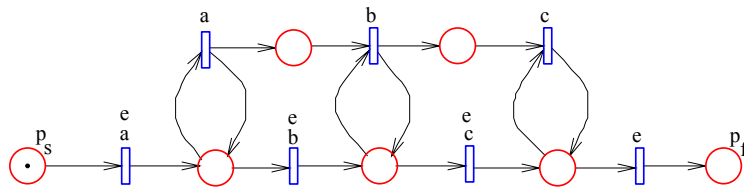
což je ve sporu s (*). Při dostatečně velké délce slova w nestačí si žádná PN si toto slovo zapamatovat. Žádná PN (OPN resp. GPN) tedy nedokáže rozpoznávat jazyk $\{ww^R : w \in A^*\}$.

Věta 3.2.4:

Existuje PN-jazyk, který není bezkontextový.

Důkaz:

Takovým jazykem je např. jazyk $L_3 = \{a^n b^n c^n : n > 0\}$ nebo $L_3' = \{a^n b^n c^n : n \geq 0\}$. Kdyby jazyky L_3 resp. L_3' byly bezkontextové, pak by byly rozpoznávatelné zásobníkovými automaty, to však být nemohou. Na obr.3.2.5 je zobrazena značená Petriho síť ve standardním tvaru generující jazyk L_3 (značení a, b, c tři dolních přechodů) resp. L_3' (značení e, e, e tři dolních přechodů).



Obr. 3.2.5

Věta 3.2.5:

Každý PN-jazyk je kontextový.

Důkaz:

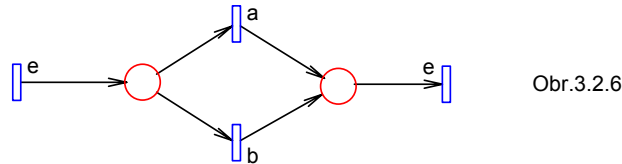
.....

Věta 3.2.6:

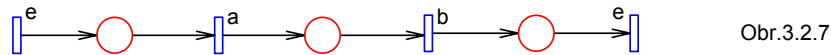
Každý bezkontextový jazyk lze reprezentovat pomocí obyčejných Petriho sítí s hierarchizačním konstruktem invokace přechodu.

Důkaz:

Bezkontextový jazyk pracuje s přepisovacími pravidly tvaru $X \rightarrow \alpha$, kde $X \in N$, $\alpha, \beta \in (N \cup T)^*$. Konstrukci $X \rightarrow \alpha\beta$ (alternativní pravidla, pravidla se stejnou levou stranou) lze popsat fragmentem Petriho sítě zobrazeném na obr.3.2.6 a konstrukci $X \rightarrow \alpha\beta$ (zřetězení slov) fragmentem Petriho sítě na obr.3.2.7. Jiné konstrukce bezkontextové gramatiky neznají. Slova α, β mohou obsahovat neterminály, včetně neterminálu X . Terminálními symboly jsou označeny elementární nehierarchické přechody a neterminálními symboly hierarchické invokované přechody. Jestliže v pravidle $X \rightarrow \alpha$ slovo α obsahuje opět neterminál X , pak volání příslušného přechodu je rekurzivní. Konstrukce Petriho sítě reprezentující bezkontextovou gramatiku je ilustrována v příkladu 3.2.2.



Obr.3.2.6



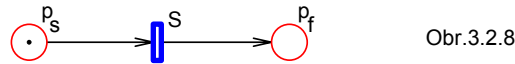
Obr.3.2.7

Příklad 3.2.2:

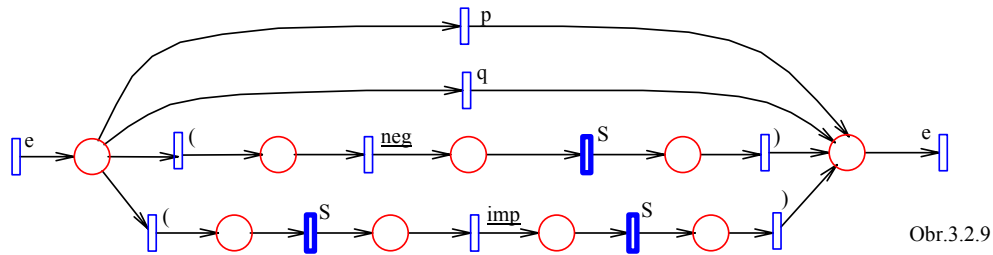
Jazyk výrokové logiky popsán bezkontextovou gramatikou

$$S \rightarrow p \mid q \mid (\neg S) \mid (S \Rightarrow S)$$

lze charakterizovat sítěmi na obr.3.2.8 (výchozí elementární síť) a 3.2.9 (síť popisující přepisovací pravidlo s levou stranou S).



Obr.3.2.8



Obr.3.2.9

Věta 3.2.7:

Každý neomezený jazyk (jazyk typu 0, viz definice 3.2.1), tj. jazyk rozpoznatelný Turingovým strojem, lze reprezentovat buď P/T sítěmi s inhibičními hranami nebo P/T sítěmi bez inhibičních hran, ale s prioritami.

Důkaz:

Idea důkazu je naznačena v následujících bodech:

1. Věta: Libovolný rekurzivně spočetný jazyk (= jazyk generovaný gramatikou typu 0 = jazyk rozpoznávaný Turingovým strojem) může být generován čítačovým automatem.

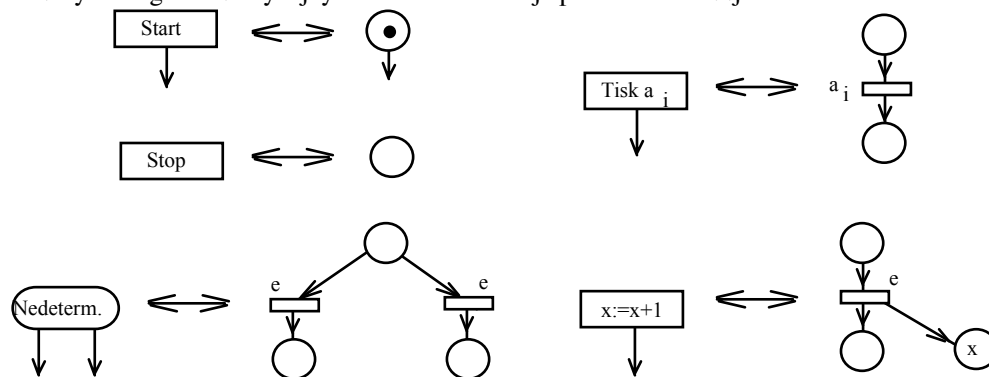
Důkaz: Viz lit.[3].

2. Definice: **Čítačový automat** je automat sestávající z konečného počtu **čítačů** $\{x_i: i=1,2,\dots,n\}$ a **programu** automatu.

Program automatu je souvislý orientovaný graf s jedním vrcholem bez vstupních hran (operátor **startu**) a s jedním vrcholem bez výstupních hran (operátor **ukončení**). Ostatní vrcholy grafu odpovídají operátorům **inkrementace čítačů** ($x_i = x_i + 1$) a operátorům **dekrementace čítačů** ($x_i = x_i - 1$). Oba tyto operátory mohou mít libovolný počet vstupních hran. Operace inkrementace je vždy proveditelná a operátor má jedinou výstupní hranu. Operace dekrementace je proveditelná a provádí se jen tehdy, je-li $x_i > 0$. Případům $x_i > 0$ a $x_i = 0$ odpovídají dvě výstupní hrany operátoru. Navíc jsou zde operátory **tisku** pro jednotlivé symboly abecedy (příslušný uzel má libovolný počet vstupních hran a jedinou výstupní hranu) a operátor **nedeterminovaného**

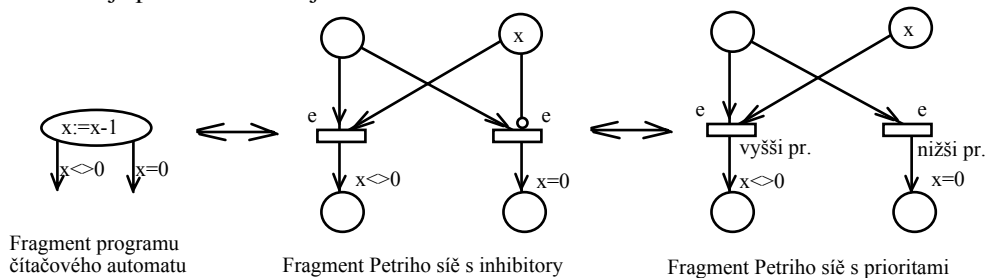
přechodu (s libovolným počtem vstupních a dvěma výstupními hranami odpovídajícími dvěma alternativním nedeterminovaným pokračováním výpočetního procesu).

3. Všechny výše uvedené operace čítačových automatů (s výjimkou operace dekrementace čítače) lze realizovat po moci příslušných fragmentů obyčejných Petriho sítí. To je patrné z následujícího obr.3.2.10.



Obr.3.2.10

4. Operaci dekrementace čítače lze realizovat pomocí Petriho sítí s inhibičními hranami nebo pomocí Petriho sítí s prioritami. To je patrné z následujícího obr.3.2.11.



Obr.3.2.11

Věta 3.2.8:

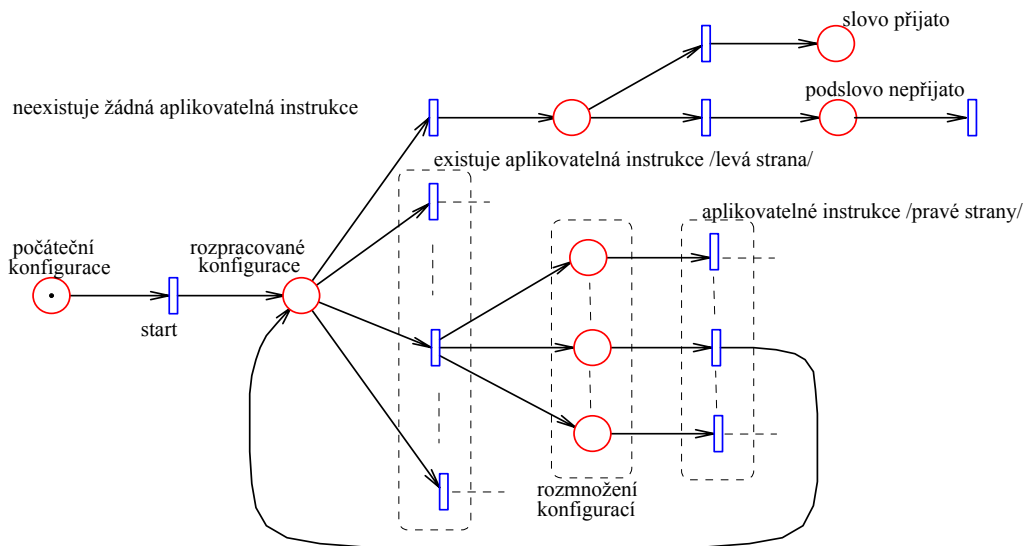
Každý neomezený jazyk (jazyk typu 0), tj. jazyk rozpoznatelný Turingovým strojem, lze reprezentovat barevnými Petriho sítěmi pracujícími s nekonečnou množinou barev (s datovými typy představujícími nekonečné množiny hodnot).

Důkaz:

Kterýkoliv matematický stroj (konečný automat, zásobníkový automat) a tedy také Turingův stroj lze popsat definicemi následujících objektů (pojmu) :

- pojmu konfigurace - stavu výpočtu prováděného strojem,
- pojmu počáteční konfigurace,
- pojmu koncové konfigurace (žádná instrukce stroje není k této konfiguraci aplikovatelná, podle stavu procesoru může být ukončení výpočtu různě interpretováno (např. vst. slovo přijato - nepřijato)),
- pojmu instrukce stroje (aplikovatelnost instrukce k dané konfiguraci, aplikace instrukce vede k nové konfiguraci),
- struktury instrukce jako relace typu zobrazení, levá část instrukce obsahuje: stav procesoru stroje (kód instrukce) případně identifikaci relevantní části paměti (operand, operandy) , pravá část obsahuje: následující stav procesoru stroje (kód instrukce) a případně identifikaci následující relevantní části paměti.

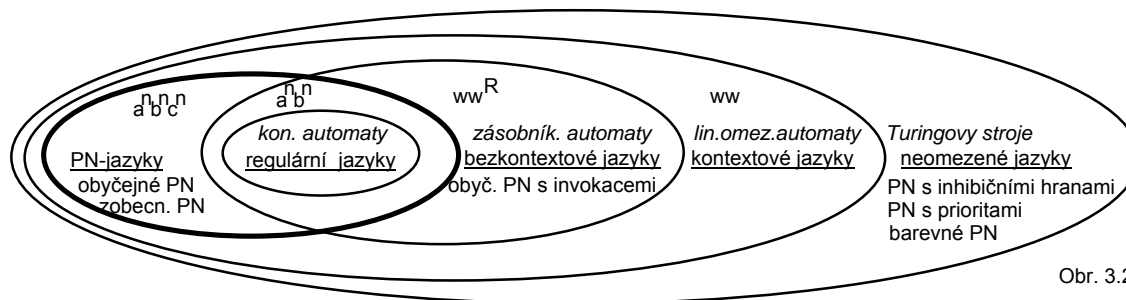
Všechny stroje mohou být reprezentovány barevnými Petriho sítěmi o společné struktuře. Liší se pouze typy tokenů kolujícími v síti, jejichž struktura a hodnota reprezentuje strukturu a hodnotu konfigurací příslušných strojů. Přechody představují instrukce měnící hodnotu konfigurací. Schéma barevné Petriho rozpoznávacího daný jazyk je zobrazeno na obr.3.2.12.



Obr. 3.2.12

Poznámka 3.2.1:

Z výše dokázaných vět vyplývá, že vzájemný vztah PN-jazyků a jazyků podle Chomského hierarchie je zobrazen množinovým diagramem na obr.3.2.10.



Obr. 3.2.10

4. Časované Petriho sítě

Several approaches are possible for the introduction of temporal specifications in Petri Nets. Time can be associated with

- tokens (token timed PN),
- transitions (t-timed PN),
- places (p-timed PN),
- arcs (a-timed PN).

Furthermore timing can be specified in a *deterministic* or in a *stochastic* manner, indicating the associated time is a constant or a random variable.

Token timed Petri Nets. It is assumed, the tokens carry a time stamp that records the time instant from which the tokens are able to fire a transition. The value of time stamp can be incremented at each transition firing.

Transition timed Petri Nets. There are two possibilities associating time to transitions:

- **atomic firing:**
 - tokens remain in input places for the delay associated with the transition to be fired,
 - tokens are consumed from input places and immediately generated in output places when the transition fires.
- **three-phase firing:**
 - tokens are consumed from input places when the transition is enabled,
 - tokens are inside the transition, until the delay elapses,
 - after delay elapsing the tokens are generated in output places.

Place timed Petri Nets. Time is associated with places, so that a token arriving into place can enable a transition only after this waiting time is elapsed. For each token it is necessary to specify if it is in a *waiting state* or a *ready state*.

Arc timed Petri Nets: a travelling delay is associated with each arc; tokens are available for firing only when they reach a transition.

It can be proved that all above models are equivalent, since every model can be translated into the another. The token timed PN and the transition timed PN (with atomic firing) are preferred because they do not introduce "intermediate" markings which do not correspond to markings in the underlying untimed CP-net.

4.1. Petriho sítě s časovanými přechody

V této kapitole podrobněji objasníme sémantiku a způsob interpretace Petriho sítě s časovanými přechody (t-timed PN with atomic firing). Zvolený model časování je charakterizován následujícími rysy:

1. Každému přechodu sítě $t \in T$ je přiřazeno nezáporné číslo θ_t udávající dobu trvání přechodu t . Tato doba může být:

- konstanta, tj. každé provedení daného přechodu má vždy stejné trvání,
- náhodná veličina s daným pravděpodobnostním rozdělením, tj. různá provedení téhož přechodu mají obecně různou dobu trvání.

Z jiného hlediska můžeme rozlišovat přechody:

- s nulovou dobou trvání (*immediate transition*),
- s nenulovou dobou trvání (*timed transition*).

2. Představujeme si, že každý přechod s nenulovou dobou trvání je vybaven časovačem (*timer*), který je v okamžiku uschopnění (enabling) přechodu $t \in T$ nastaven na hodnotu θ_t . Hodnota časovače se pak s přibývajícím časem rovnoměrně snižuje a teprve v okamžiku, kdy dosáhne nulové hodnoty, je přechod proveden (tj. dojde ke změně značení sítě). Představujeme si, že vlastní aktivita reprezentovaná přechodem

probíhá v době, kdy je přechod uschopněný a ukončení této aktivity je spojeno se změnou značení sítě. Hodnota časovače představuje paměť spojenou s přechodem.

3. Evoluce Petriho sítě s časovanými přechody je dána posloupností

$$\{(\tau_{(1)}, t_{(1)}), (\tau_{(2)}, t_{(2)}), \dots, (\tau_{(i)}, t_{(i)}), \dots\},$$

kde $\tau_{(i)}$ je okamžik provedení přechodu $t_{(i)}$.

4. Je-li uschopněno současně více přechodů, nastaví se současně časovače všech uschopněných přechodů a začne probíhat jejich rovnoměrná dekrementace (se stejnou rychlostí pro všechny uschopněné přechody).

5. Dojde-li k zneschopnění přechodu dříve než časovač přechodu dosáhl nulové hodnoty, pak se v okamžiku zneschopnění zastaví dekrementace časovače (činnost reprezentovaná přechodem je přerušena).

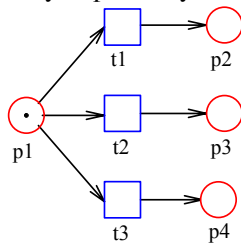
Při znovuschopnění přechodu můžeme volit mezi dvěma možnostmi:

- časovač startuje ze zapamatované snížené hodnoty časovače (**continue, resume mechanismus**), tj. práce reprezentovaná přechodem pokračuje z bodu ve kterém byla přerušena,
- časovač je inicializován počáteční nesníženou hodnotou doby trvání přechodu (**restart mechanismus**), tj. práce provedená před přerušením je ztracena a začíná se opět od počátku.

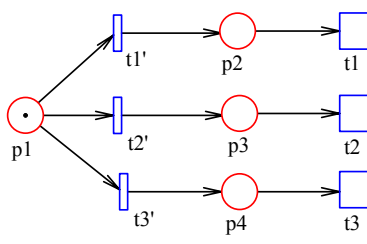
6. Konflikty mezi okamžitými přechody jsou svou povahou nedeterministické a mohou být řešeny:

- **deterministickým mechanismem priority** (přechody, které jsou v konfliktu mají přiřazenu různou prioritu),
- **stochastickým mechanismem priority** reprezentovaným diskrétní distribuční funkcí na množině přechodů, které jsou v konfliktu.

7. Konflikt mezi časovanými přechody vyhrává vždy přechod s nejkratší dobou trvání. Chceme-li zamezit tomuto výsledku, separujeme řešení konfliktu a časovou specifikaci přechodů postupem naznačeným na obr.4.1.1 (preselection policy). Důsledné uplatnění **principu separace** eliminuje konflikty mezi časovanými přechody.

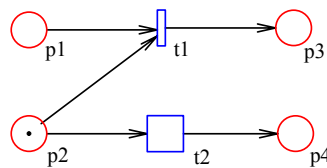


Obr. 4.1.1.a



Obr. 4.1.1.b

8. Konflikt mezi časovaným a okamžitým přechodem vyhrává vždy okamžitý přechod. To je ilustrováno na obr.4.1.2. Provedení okamžitého přechodu t_1 způsobí přerušování činnosti reprezentované časovaným přechodem t_2 .



Obr. 4.1.2

9. Nejčastěji užívané strategie práce s pamětmi přechodů jsou tyto:

- Při každé změně stavu (značení) sítě, tj. při provedení kteréhokoliv přechodu sítě, se zapomene hodnota časovačů všech přechodů. Práce vykonaná nedokončenými aktivitami je tak ztracena. Jedná se o strategii univerzálního restartu (**resampling**).
- Při změně stavu sítě, tj. při provedení libovolného přechodu, si časovače všech přechodů podrží svou aktuální hodnotu. Paměť každého přechodu měří kumulativní dobu kdy přechod je v uschopněném stavu (počítáno od posledního provedení přechodu). Práce vykonaná nedokončenými aktivitami není ztracena, při opětovném vyvolání aktivity se pokračuje z bodu přerušování. Jedná se o strategii univerzálního resumu (**age memory**).
- Při provedení kteréhokoliv přechodu, tj. při jakékoliv změně stavu sítě, nastane následující:
 - časovače přechodů, které byly změnou zneschopněny jsou restartovány,

- o časovače přechodů, které nebyly zneschopněny, nemění svou hodnotu, tj. aktivity odpovídající těmto přechodům nejsou změnou stavu sítě nijak dotčeny a pokračují v práci. Paměť každého přechodu měří dobu kdy přechod je v uschopněném stavu (počítáno od posledního uschopnění přechodu).

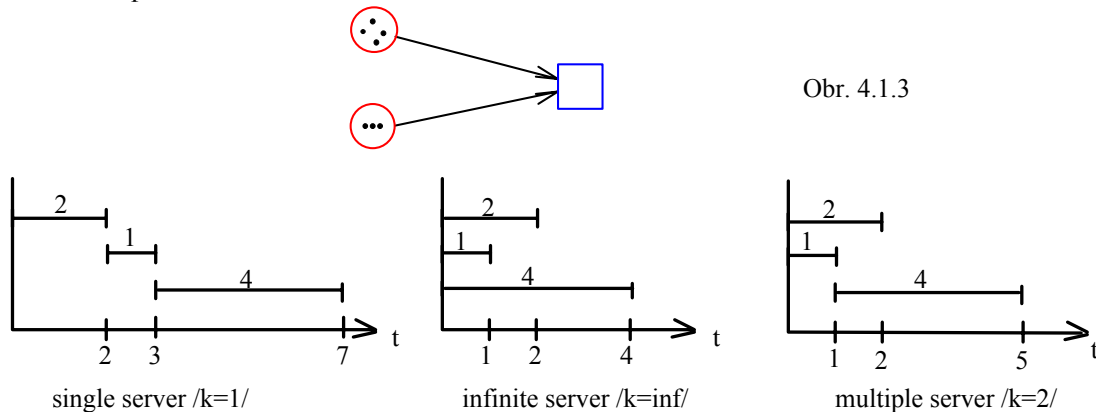
Tato strategie se označuje termínem **enabling memory**.

- o Smíšená strategie: pro každý časovaný přechod se individuálně stanoví, zda bude restartován (vždy při každé změně stavu sítě nebo jen při zneschopnění přechodu) nebo resumován (při opětném znovuschopnění).

10. V případě, kdy stupeň proveditelnosti časovaného přechodu je větší než 1 (multiple enabling), může proběhnout vícenásobné provedení přechodu následujícími způsoby:
- o Tokeny nebo jejich sestavy potřebné pro jednorázové provedení přechodu zpracovává přechod postupně, seriovým způsobem. Prováděný přechod je interpretován jako obslužný systém s jedním kanálem obsluhy (**single-server semantics**).
 - o Přechod zpracovává všechny sestavy tokenů potřebné pro jednorázové provedení přechodu současně, paralelním způsobem. Prováděný přechod je interpretován jako obslužný systém s neomezeným počtem kanálů obsluhy (**infinite-server semantics**).
 - o Přechod je schopen zpracovávat současně jen omezený počet uschopňujících tokenových sestav. Z pohledu teorie hromadné obsluhy představuje pak přechod obslužný systém s omezeným počtem obslužných kanálů (**multiple-server semantics**).

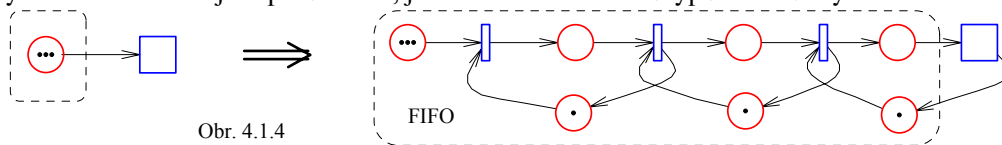
Pro každý časovaný přechod je třeba stanovit kapacitu zpracování k (počet obslužných kanálů, $k \geq 1$). Volba $k=1$ odpovídá prvé alternativě a volba $k=\infty$ druhé alternativě. Ostatní volby alternativě třetí.

Příklad: Časovaný přechod z obr.4.1.3 má stupeň proveditelnosti 3, tj. může být proveden celkem 3 krát. Předpokládejme, že doba trvání přechodu je náhodná veličina, která se realizuje hodnotami 2,1,4. Na obr. jsou zobrazeny časové průběhy trojnásobného provedení přechodu v režimu single-, infinite- nebo multiple-server.



Obr. 4.1.3

11. Vzhledem k tomu, že v černobílých Petriho sítích nelze rozlišovat individualitu tokenů, nelze ani explicitně stanovit disciplínu obsluhy tokenů (service policy, queueing discipline), jakými jsou např. disciplíny FIFO (First In First Out), LIFO (Last In First Out) a pod. Různé disciplíny obsluhy lze však modelovat pomocí podsíti p-bloků rozvádějících místo ve kterém chceme realizovat frontu tokenů se zvolenou disciplínou fronty. Tak na obr.4.1.4 je např. ukázáno, jak lze realizovat frontu typu FIFO délky 3.



Obr. 4.1.4

Příklad 4.1.1:

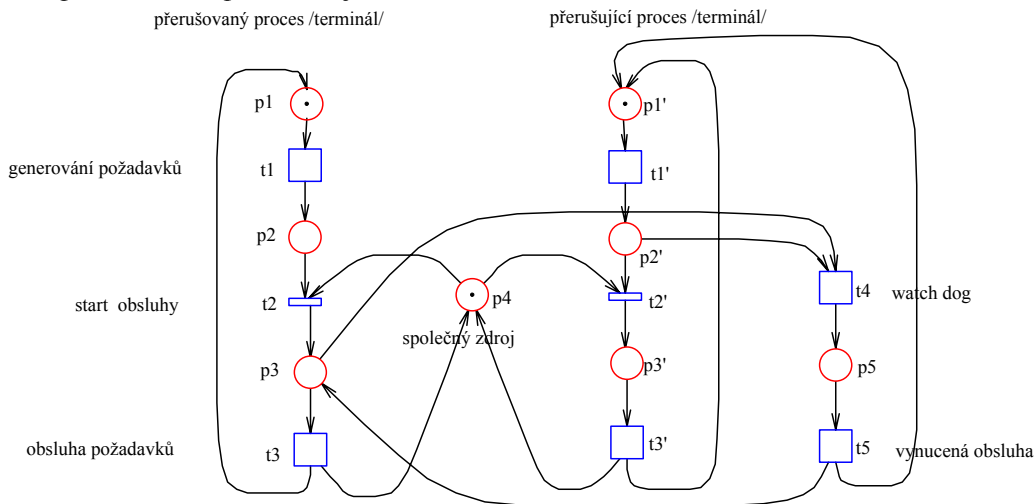
Na obr.4.1.5 je zobrazen PN-systém modelující systém dvou terminálů využívajících služeb společně sdíleného zdroje např. dva počítače na kterých se připravují dokumenty k tisku a společná tiskárna. Navíc jeden terminál má přednostní právo přístupu ke zdroji: pokud druhý (přerušovaný) terminál neskončí využívání zdroj e v krátkem čase (do vypršení předem daného tzv. **timeoutu**) od vzniku požadavku na zdroj u prvního

(přerušujícího) terminálu, převezme prvý terminál zdroj bez ohledu na to, zda druhý terminál ukončil práci se zdrojem či nikoliv. Po skončení práce vrací prvý terminál zdroj zpět druhému terminálu.

Na obr.4.1.5 je cyklická práce obou terminálů zobrazena podsítěmi $p_1, t_1, p_2, t_2, p_3, t_3$ (případně přerušovaný terminál) a $p_1', t_1', p_2', t_2', p_3', t_3'$ (případně přerušující terminál). Časované přechody t_1 , resp. t_1' představují autonomní práci terminálů, po dobu jejich činnosti se token nachází v místě p_1 resp. p_1' . Ukončení přechodu t_1 nebo t_1' , tj. objevení se tokenu v místě p_2 nebo p_2' , znamená vznik požadavku na přidělení společně sdíleného zdroje. Přechody t_2, t_2' jsou okamžité, protože zobrazují pouhý akt přidělení zdroje. Časované přechody t_3 , resp. t_3' představují proces používání zdroje tím či oním terminálem. Po dobu používání zdroje se token nachází v místě p_3 , resp. p_3' .

Uvažujme nyní situaci, kdy v době využívání zdroje potenciálně přerušitelným procesem, vznesl požadavek na poskytnutí zdroje potenciálně přerušující proces. V této situaci se nachází tokeny v místě p_3 a p_2' a tedy dva časované přechody t_3 a t_4 jsou v konfliktu. Doba trvání přechodu t_4 (*watch dog*) je pevně nastavena (timeout). Podaří-li se přerušitelnému procesu dokončit práci se zdrojem do vypršení timeoutu, pak přechod t_3 konflikt vyhraje, do místa p_4 se vrátí token, přechod t_4 je zneschopněn a je uschopněn okamžitý přechod t_2' , který je ihned proveden (k převzetí zdroje došlo bez přerušení). Nepodaří-li se přerušitelnému procesu u dokončit práci se zdrojem v rámci časového limitu timeoutu, pak konflikt vyhraje přechod t_4 a přerušující proces si vynucuje přidělení zdroje. Token přichází do místa p_5 , kde se nachází až do okamžiku ukončení práce se zdrojem. Potom se token vrací do místa p_3 a přerušovaný proces může svou práci s navráceným zdrojem dokončit.

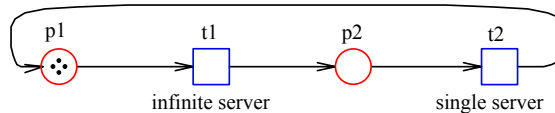
Předpokládáme, že přechod t_4 je po znechopnění restartován a přechod t_3 resumován. Můžeme si ovšem také představit, že i přechod t_3 je restartován.



Obr. 4.1.5

Příklad 4.1.2:

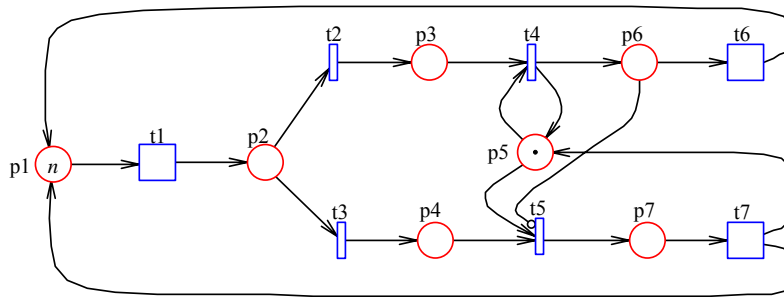
Na obr.4.1.5a je zobrazen PN-systém modelující počítačový systém se čtyřmi uživatelskými stanicemi a jedním sdíleným procesorem. Přechod t_1 reprezentuje vznikání požadavků na práci procesoru. Vzhledem k tomu, že tyto požadavky vznikají v náhodných okamžicích nezávisle na sobě, je nutno přechod t_1 interpretovat jako obslužný systém s neomezenou kapacitou obsluhy. Naproti tomu přechod t_2 představuje výpočty prováděné procesorem a to tak, že v každý okamžik je vždy prováděn výpočet jediný. Přechod t_2 je proto nutné interpretovat jako obslužný systém s jediným kanálem obsluhy.



Obr. 4.1.5a

Příklad 4.1.3:

Na obr. 4.1.6 je zobrazena Petriho síť modelující systém tvořený množinou n procesů, které mají přístup do společné databáze a to buď za účelem čtení nebo zápisu. Z databáze může současně číst libovolný počet procesů, zapisovat však může vždy jen jediný proces. Pokud probíhá zápis, žádný jiný proces, kromě právě zapisujícího, nemá do databáze přístup.



Obr. 4.1.6.

Místa a přechody zobrazená na obr. 4.1.6 mají následující význam:

- p1: značení místa udává počet procesů, které v daný okamžik nevyžadují přístup k databázi
- t1: časovaný přechod modelující proces vzniku potřeby přístupu do databáze u některého z procesů
- p2: značení místa udává počet procesů, které v daný okamžik vyžadují přístup k databázi
- t2: přístup do databáze se požaduje za účelem čtení
- t3: přístup do databáze se požaduje za účelem zápisu
- p3: značení místa udává počet procesů, které v daný okamžik, čekají na čtení z databáze
- p4: značení místa udává počet procesů, které v daný okamžik, čekají na zápis do databáze
- p5: token v místě značí, že neprobíhá zápis do databáze
- t4: začátek procesu čtení
- t5: začátek procesu zápisu
- p6: značení místa udává počet procesů čtoucích z databáze
- p7: token v místě značí, že probíhá zápis do databáze
- t6: časovaný přechod modelující proces čtení
- t7: časovaný přechod modelující proces zápisu

4.2. Stochastické Petriho síť

Definice 4.2.1:

Stochastická Petriho síť (SPN - Stochastic Petri Net) je speciální případ PN-systému s časovanými přechody, kdy trvání θ_i všech přechodů t_i jsou náhodné veličiny s exponenciálním rozdělením

$$F(x) = P(\theta_i \leq x) = 1 - e^{-\mu_i x}, \quad x \geq 0.$$

Symbol $\mu_i = \mu(t_i)$ označuje intenzitu přechodu t_i a symbol

$$\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_{|T|})^T$$

označuje vektor intenzit všech přechodů.

Formálně je SPN zadána dvojicí $\langle\langle P, T, I, O, H, M_0 \rangle, \boldsymbol{\mu}\rangle$, kde $\langle P, T, I, O, H, M_0 \rangle$ je PN-systém a $\boldsymbol{\mu}$ zobrazení množiny přechodů T do množiny kladných reálných čísel \mathbb{R}^+ ($\boldsymbol{\mu}: T \rightarrow \mathbb{R}^+$). Obecněji lze funkci $\boldsymbol{\mu}$ definovat jako zobrazení $T \times \mathbb{R}S(M_0) \rightarrow \mathbb{R}^+$.

Poznámky 4.2.1:

- O exponenciálním rozdělení, Poissonově rozdělení a markovských procesech viz přílohu B (Appendix B). Zde jenom připomeňme, že exponenciální rozdělení je "bez paměti" (memoryless), což znamená, že doba dokončení přechodu v intervalu $(t, t+dt)$ nezávisí na t , tj. na tom, jak dlouho již provádění přechodu trvá. Pro časované Petriho síť má tato skutečnost velmi příjemný důsledek spočívající v tom, že neexistuje rozdíl mezi různými strategiemi práce s paměťmi časovaných přechodů (resampling, age-memory, enabling-memory) - viz kap.4.1. Pro jednoduchost si můžeme např. představovat, že při každé změně stavu (značení) sítě, tj. po provedení jakéhokoliv přechodu, se hodnoty časovačů všech přechodů zapomínají (resampling).
- Jiným příjemným důsledkem exponenciálního rozdělení dob trvání přechodů je skutečnost, že *kvalitativní analýza* (tj. analýza neuvažující dobu trvání přechodů) stochast. sítě $\langle\langle P, T, I, O, H, M_0 \rangle, \boldsymbol{\mu}\rangle$ je identická s kvalitativní analýzou příslušné nečasované sítě $\langle P, T, I, O, H, M_0 \rangle$. To souvisí s tím, že hustota pravděpodobnosti exponenciálního rozdělení je nenulová na celém intervalu $\langle 0, \infty \rangle$. V případě konfliktu přechodů má potom kterýkoliv přechod nenulovou pravděpodobnost provedení, tj. kterýkoliv proveditelný přechod může být v principu proveden. Obě sítě SPN $\langle\langle P, T, I, O, H, M_0 \rangle, \boldsymbol{\mu}\rangle$ i PN-systém $\langle P, T, I, O, H, M_0 \rangle$ mají tedy společnou množinu proveditelných sekvencí přechodů a společnou množinu dosažitelných značení. Vyšetřování bezpečnosti, omezenosti, živosti, uzamčení, reverzibility a pod. pro SPN se tedy převádí na vyšetření těchto vlastností přidruženého PN-systému.
- Dalším významným důsledkem exponenciálního rozdělení dob trvání přechodů je skutečnost, že *kvantitativní analýza* stochastických Petriho sítí může být převedena - jak hned dále ukážeme - na kvantitativní analýzu přidružených markovských procesů. Při této analýze nás zajímají pravděpodobnosti jednotlivých stavů-značení sítě (a to přechodové i ustálené), střední hodnoty počtu tokenů v jednotlivých místech, střední hodnoty počtu provedení jednotlivých přechodů za jednotku času a pod. Analýza markovských procesů je teoreticky plně vyřešena a numericky snadno proveditelná - viz Appendix B.
- Při speciální definici funkce $\boldsymbol{\mu}$ ($\boldsymbol{\mu}: T \rightarrow \mathbb{R}^+$) intenzity provádění přechodů nezávisí na aktuálním značení sítě (marking independent rates), při obecné definici funkce $\boldsymbol{\mu}$ ($\boldsymbol{\mu}: T \times \mathbb{R}S(M_0) \rightarrow \mathbb{R}^+$) intenzity provádění přechodů mohou na aktuálním značení sítě záviset (marking dependent rates).

Definice 4.2.2:

Uvažujme stochastickou Petriho síť $\langle\langle P, T, I, O, H, M_0 \rangle, \boldsymbol{\mu}\rangle$. K této síti přiřadíme *přidružený markovský proces* s diskrétní množinou stavů a spojitým časem takto:

- Množinou I stavů markovského procesu je množina $\mathbb{R}S(M_0)$, tj. množina všech dosažitelných značení Petriho sítě.
- Matice intenzit $Q = (q_{ij})$ markovského procesu je definována takto:

$$q_{ij} = \sum \mu(t_k), \text{ kde sumace probíhá pro } t_k \in E_j(M_i), i \neq j,$$

$$q_{ii} = - \sum \mu(t_k), \text{ kde sumace probíhá pro } t_k \in E(M_i).$$

Připomeňme, že:

$E(M_i)$ označuje množinu všech přechodů proveditelných ve stavu M_i

$E_j(M_i)$ označuje množinu všech přechodů, které jsou ve stavu M_i proveditelné a jejichž provedení převádí síť do stavu M_j

- Počáteční pravděpodobnostní rozdělení markovského procesu je definováno takto:

$$p_{M_0}(0)=1, p_{M_i}(0)=0 \text{ pro } i \neq 0.$$

Poznámky 4.2.2:

1. Grafový diagram přidruženého markovského procesu (viz Appendix B) se od grafu dosažitelnosti $RG(M_0)$ Petriho sítě (viz definice 2.1.5) liší v těchto bodech:

- Zatímco graf dosažitelnosti je ohodnoceným orientovaným multigrafem, (uzly - stavy M_i, M_j mohou být spojeny více orientovanými hranami, pokud existuje více přechodů převádějících síť ze stavu M_i do stavu M_j), je graf markovského procesu ohodnoceným orientovaným grafem (uzly-stavy M_i, M_j jsou spojeny nejvýše jednou orientovanou hranou).
- Zatímco hrany grafu dosažitelnosti vedoucí z uzlu M_i do uzlu M_j jsou ohodnoceny identifikátory přechodů převádějících Petriho sítě ze stavu M_i do stavu M_j , je hrana grafu markovského procesu vedoucí z uzlu M_i do uzlu M_j ohodnocena součtem intenzit všech těchto přechodů.

Na obr.4.2.1 je ilustrována transformace fragmentu grafu dosažitelnosti na graf přidruženého markovského procesu.



Obr. 4.2.1

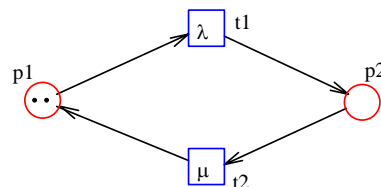
2. Pro prvky matice přechodu q_{ii} platí

$$q_{ii} = -\sum q_{ij} \text{ sčítáno pro } j \neq i$$

a tedy jejich hodnota neobsahuje žádnou novou informaci. Hrany odpovídající těmto intenzitám (smčky z uzlu M_i do uzlu M_j) na grafu markovského procesu proto nezobrazujeme.

Příklad 4.2.1:

Uvažujme stochastickou Petriho síť zobrazenou na obr.4.2.2, kde přechod t_1 má intenzitu λ a přechod t_2 intezitu μ .



Obr. 4.2.2

Provedeme nejdříve kvalitativní analýzu Petriho sítě.

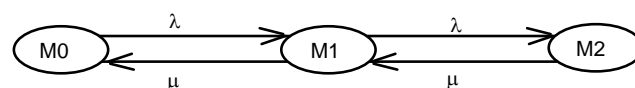
| | | | | | |
|----|----|----|-------|-------|-------|
| | t1 | t2 | M0 | M1 | M2 |
| p1 | -1 | 1 | 2 | 1 | 0 |
| p2 | 1 | -1 | 0 | 1 | 2 |
| | | | t1→M1 | t1→M2 | t2→M1 |
| | | | | t2→M0 | |

Tab. 4.2.1.

Na základě tabulky 4.2.1 sestojíme graf dosažitelnosti a graf přidruženého markovského procesu (obr.4.2.3).



Obr. 4.2.3.a Graf dosažitelnosti



Obr. 4.2.3.b Graf markovského procesu

Z tabulky a grafu dosažitelnosti vyplývá, že Petriho síť je 2-omezená, živá a reverzibilní.

Přejdeme ke kvantitativní analýze stochastické Petriho sítě. Matice intenzit přechodů má tvar:

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

Ze silné souvislosti grafu markovského procesu vyplývá existence ustálených pravděpodobností

$$\pi = (\pi_0, \pi_1, \pi_2)^T$$

stavů M_0, M_1, M_2 . Ty vypočteme řešením soustavy lineárních algebraických rovnic (viz Appendix B):

$$Q^T \cdot \pi = 0$$

$$\pi_0 + \pi_1 + \pi_2 = 1,$$

neboli

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

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

$$\lambda\pi_1 - \mu\pi_2 = 0$$

$$\pi_0 + \pi_1 + \pi_2 = 1$$

Prvé tři rovnice jsou lineárně závislé, můžeme tedy libovolnou z nich, např. druhou, vypustit. Jako řešení dostáváme ustálené pravděpodobnosti stavů M_0, M_1, M_2 .

$$\pi_0 = (1 + (\lambda/\mu) + (\lambda/\mu)^2)^{-1}, \quad \pi_1 = (\lambda/\mu)\pi_0, \quad \pi_2 = (\lambda/\mu)^2\pi_0.$$

Vypočteme ještě střední hodnoty počtů tokenů v jednotlivých místech sítě $E\{n(p1)\}, E\{n(p2)\}$ a střední hodnoty počtů provedení jednotlivých přechodů za jednotku času $E\{n(t1)\}, E\{n(t2)\}$.

$$E\{n(p1)\} = 2 \cdot \pi_0 + 1 \cdot \pi_1,$$

$$E\{n(p2)\} = 1 \cdot \pi_1 + 2 \cdot \pi_2,$$

$$E\{n(t1)\} = \lambda \cdot (\pi_0 + \pi_1),$$

$$E\{n(t2)\} = \mu \cdot (\pi_1 + \pi_2).$$

Definice 4.2.3:

Ustálené pravděpodobnostní rozdělení $\pi = (\pi_1, \pi_2, \dots, \pi_{|RS|})^T$ je základem pro kvantitativní vyhodnocení chování stochastických Petriho sítí (SPN). Nejčastěji užívanými kvantitativními ukazateli jsou:

- **Pravděpodobnost určitého jevu** v SPN:

$$R = \sum r(M_i) \cdot \pi_i,$$

kde $r(M_i) = 1$ (resp. 0), jestliže jev nastává (resp. nenastává) při značení M_i .

- **Očekávaná hodnota počtu tokenů v daném místě p_j :**

$$E\{n(p_j)\} = \sum r(M_i) \cdot \pi_i,$$

kde $r(M_i) = n$, je-li $M_i(p_j) = n$.

- **Očekávaná hodnota počtu provedení přechodu t_j za jednotku času:**

$$E\{n(t_j)\} = \sum r(M_i) \cdot p_i,$$

kde $r(M_i) = \mu_j$ (resp. $r(M_i) = 0$), je-li (resp. není-li) $t_j \in E(M_i)$.

Všimněme si, že všechny tři ukazatele jsou lineárními kombinacemi ustálených pravděpodobností.

Příklad 4.2.1 (pokračování):

Uvažujme několik modelů Petriho sítě z obr.4.2.2. Těmito modely budou spolehlivostní modely systému sestávajícího ze dvou identických prvků (z hlediska hodnot spolehlivostních parametrů). Markovské modely jsou zobrazeny grafovými diagramy na obr.4.2.4 - 4.2.9. Význam použitých symbolů je následující:

λ intenzita poruchy prvku

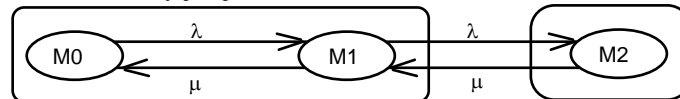
μ intenzita obnovy (opravy) prvku

M_0 stav systému, kdy oba prvky jsou provozuschopné (oba tokeny jsou v místě p1)

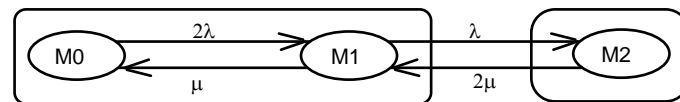
M_1 stav systému, kdy jeden prvek je provozuschopný a jeden porušený (jeden token v místě p1 a jeden v místě p2)

M_2 stav systému, kdy oba prvky jsou porušené (oba tokeny jsou v místě p2)

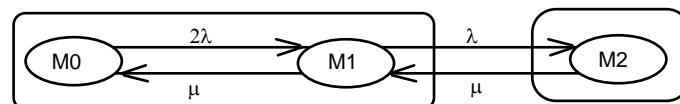
Na následujících obr.4.2.4 - 4.2.5 jsou zobrazeny tzv. **paralelní systémy**. Systém se považuje za provozuschopný, je-li provozuschopný aspoň jeden prvek (systém je ve stavu M_0 nebo M_1). Prvky se vzájemně zálohují. Na obr.4.2.4 je zobrazen model tzv. **studené (nezatížené) zálohy**: jsou-li provozuschopné oba prvky (systém je ve stavu M_0) je provozován pouze jeden prvek (intenzita přechodu ze stavu M_0 do stavu M_1 je λ). Na obrázcích 4.2.5 a 4.2.6 jsou zobrazeny modely tzv. **horké (zatížené) zálohy**: provozovány jsou všechny provozuschopné prvky (intenzita přechodu ze stavu M_0 do stavu M_1 je 2λ a ze stavu M_1 do stavu M_2 je λ). Modely zobrazené na obrázcích 4.2.5 a 4.2.6 se liší režimem obnovy porušených prvků: model na obr.4.2.5 pracuje s **neomezenou kapacitou obnovy** (všechny porušené prvky jsou obnovovány, intenzita přechodu ze stavu M_2 do stavu M_1 je 2μ), kdežto model z obr.4.2.6 pracuje s **omezenou kapacitou obnovy** (z porušených prvků je obnovován vždy jen jeden).



Obr. 4.2.4. Studená /nezatížená/ záloha s omezenou kapacitou obnovy

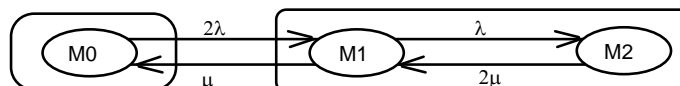


Obr. 4.2.5. Horká /zatížená/ záloha s neomezenou kapacitou obnovy

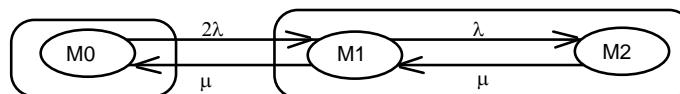


Obr. 4.2.6. Horká /zatížená/ záloha s omezenou kapacitou obnovy

Na následujících obr.4.2.7 - 4.2.8 jsou zobrazeny tzv. **seriové systémy**. Systém se považuje za provozuschopný, jsou-li provozuschopné oba dva prvky (systém je ve stavu M_0). Oba modely předpokládají, že všechny provozuschopné prvky jsou provozovány, model z obr.4.2.7 je s neomezenou kapacitou obnovy a model z obr.4.2.8 je s omezenou kapacitou obnovy.



Obr. 4.2.7. Seriový systém s neomezenou kapacitou obnovy



Obr. 4.2.8. Seriový systém s omezenou kapacitou obnovy

4.3. Petriho síť s prioritami

Zavedení prioritních úrovní v PN-systémech zvětšuje jejich popisnou sílu a poskytuje více možností při návrhu systému. Na druhé straně zavedení prioritní struktury vyžaduje redefinici některých pojmů a modifikaci metod analýzy.

Definice 4.3.1:

Petriho síť s prioritami je dvojice $\langle\langle P, T, I, O, H, M_0 \rangle, \rho\rangle$, kde $\langle P, T, I, O, H, M_0 \rangle$ je Petriho síť ve smyslu definice 2.1.2 a ρ je zobrazení množiny přechodů do množiny přirozených čísel $\rho: T \rightarrow \mathbb{N}$.

Symbolem ρ_j budeme označovat přirozené číslo přiřazené přechodu $t_j \in T$, tj. číslo $\rho(t_j)$. Toto číslo nazýváme *prioritou přechodu* t_j a má následující význam: je-li přechod t_j s prioritou k proveditelný, pak je vždy proveden dříve, než kterýkoliv jiný proveditelný přechod s prioritou $m < k$.

Poznámky 4.3.1:

- Pojem priority umožňuje:
 - rozklad množiny přechodů na třídy přechodů různých prioritních stupňů,
 - deterministické řešení konfliktů.
- Bez újmy na obecnosti můžeme předpokládat, že všechny nižší prioritní úrovně jsou neprázdné, tj.

$$(\forall t_j \in T)[\rho_j > 0 \Rightarrow (\exists t_k \in T)[\rho_k = \rho_{j-1}]]$$
- Nejnižší prioritou je priorita 0. V časovaných Petriho sítích je tato priorita rezervována pro časované přechody (tj. pro přechody s nenulovým trváním).

Definice 4.3.2:

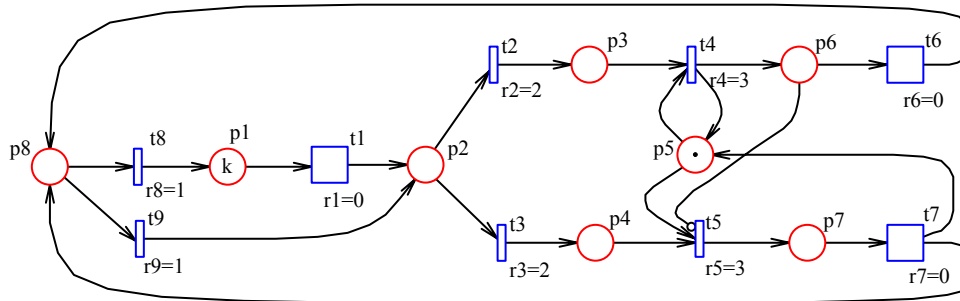
Přechod t_j sítě $\langle\langle P, T, I, O, H, M_0 \rangle, \rho\rangle$ je při značení M *povolen* (has concession), jestliže je v síti $\langle P, T, I, O, H, M_0 \rangle$ proveditelný.

Přechod t_j sítě $\langle\langle P, T, I, O, H, M_0 \rangle, \rho\rangle$ je při značení M *proveditelný* (is enabled), jestliže je při značení M povolen a jestliže žádný jiný povolený přechod nemá vyšší prioritu. Dva přechody mohou být současně proveditelné (uschopněné) při daném značení pouze tehdy, patří-li do stejné prioritní třídy.

Přechod t_j může být *proveden* (can fire) pouze tehdy, je-li proveditelný. Efekt provedení přechodu v síti $\langle\langle P, T, I, O, H, M_0 \rangle, \rho\rangle$ je stejný jako v síti $\langle P, T, I, O, H, M_0 \rangle$.

Příklad 4.3.1:

Na obr.4.3.1 je zobrazena Petriho síť modelující systém tvořený množinou procesů, které mají přístup do společné databáze a to buď za účelem čtení nebo zápisu. Z databáze může současně číst libovolný počet procesů, zapisovat však může vždy jen jediný proces. Pokud probíhá zápis, žádný jiný proces, kromě zapisujícího, nemá do databáze přístup.



Obr. 4.3.1.a Časovaná Petriho síť s prioritami - varianta (a)

- Příklad je rozšířením příkladu 4.1.3. Jednotlivá místa a přechody mají následující význam:
- p1: značení udává počet procesů, které v daný okamžik nevyžadují přístup k databázi
 - t1: vznik potřeby přístupu do databáze u některého z procesů
 - p2: značení udává počet procesů, které v daný okamžik vyžadují přístup k databázi a to buď kvůli čtení nebo kvůli zápisu
 - t2: požaduje se přístup do databáze za účelem čtení
 - t3: požaduje se přístup do databáze za účelem zápisu
 - p3: značení udává počet procesů, které v daný okamžik, čekají na čtení z databáze
 - p4: značení udává počet procesů, které v daný okamžik, čekají na zápis do databáze
 - p5: token značí, že neprobíhá zápis do databáze
 - t4: začátek procesu čtení
 - t5: začátek procesu zápisu
 - p6: značení udává počet procesů čtoucích z databáze
 - p7: token značí, že probíhá zápis do databáze
 - t6: ukončení procesu čtení
 - t7: ukončení procesu zápisu
 - p8: značení udává počet procesů, které ukončily přístup do databáze
 - t8: proces bude po určitou dobu pracovat pouze s lokální pamětí
 - t9: proces vyžaduje okamžitě další přístup do společné databáze

Všechny rozumné varianty prioritní struktury předpokládají, že časované přechody t1,t6,t7 mají přidělenou nejnižší prioritu, tj. $\rho_1=\rho_6=\rho_7=0$.

Nečasované okamžité přechody představují tři skupiny konfliktů {t2,t3} - volba mezi čtením a zápisem, {t4,t5} - start čtení nebo zápisu, {t8,t9} - volba mezi užíváním lokální paměti nebo společné databáze. Je zřejmé, že nemá smysl přiřazovat přechodům t2 a t3 různou prioritu, protože mají stejnou množinu vstupních míst (místo p2). Kdyby např. bylo $\rho_2 > \rho_3$, pak by přechod t3 nikdy nebyl proveden. Obdobný důvod platí pro přechody t8 a t9. Naproti tomu tím, že přiřadíme různé priority přechodům t4 a t5 dosáhneme toho, že čtoucí procesy budou mít přednost před procesy zapisujícími nebo naopak.

Uvažujme následující dvě varianty prioritní struktury:

- (a) $\rho_2=\rho_3=2, \rho_4=\rho_5=3, \rho_8=\rho_9=1$. Při tomto přiřazení priorit platí: jakmile je rozhodnuto o způsobu přístupu procesu do databáze (čtení nebo zápis) je tento přístup okamžitě zahájen (dříve než kterýkoliv jiný proces zvolí způsob přístupu). Volba způsobu přístupu a zahájení přístupu představují nedělitelnou akci.
- (b) $\rho_2=\rho_3=2, \rho_4=\rho_5=1, \rho_8=\rho_9=3$. Při tomto přiřazení priorit platí: když několik procesů má rozhodnout o způsobu přístupu k databázi, pak dříve než přístup kteréhokoliv z nich je zahájen, musí všechny učinit svou volbu.

Diskutujme nejprve alternativu (a) prioritní struktury - viz obr.4.3.1.a a předpokládejme počáteční značení s $k=3$, tj.

$$M_0 = (3,0,0,0,1,0,0,0) = 3 \cdot p_1 + 1 \cdot p_5.$$

Uvažujme značení

$$M_1 = (1,0,0,1,0,0,1,0) = 1 \cdot p_1 + 1 \cdot p_4 + 1 \cdot p_7,$$

kteřé je z počátečního značení dosažitelné. V tomto značení časované přechody t1 a t7 jsou uschopněny a probíhají. Stav M_1 je tedy trvajícím. Předpokládejme, že přechod t7 je proveden. Síť přejde do značení

$$M_2 = (1,0,0,1,1,0,0,1) = 1 \cdot p_1 + 1 \cdot p_4 + 1 \cdot p_5 + 1 \cdot p_8,$$

ve kterém povolení mají přechody t1, t5, t8, t9, avšak jenom přechod t5 je uschopněn (ze všech povolených přechodů má nejvyšší prioritu). Přechod t5 je okamžitý, tedy značení M_2 není trvalé, ale provedením přechodu t2 okamžitě přechází do značení

$$M_3 = (1,0,0,0,0,0,1,1) = 1 \cdot p_1 + 1 \cdot p_7 + 1 \cdot p_8,$$

ve kterém jsou uschopněny přechody t8 a t9. Oba tyto přechody jsou okamžité (nečasované) a tedy značení M_3 opět není trvalé. Provedením přechodu t9 přecházíme do značení

$$M_4 = (1,1,0,0,0,0,1,0) = 1 \cdot p_1 + 1 \cdot p_2 + 1 \cdot p_7,$$

ve kterém jsou uschopněny přechody t2 a t3. Oba tyto přechody jsou nečasované a tedy značení M_4 není trvalé. Provedením přechodu t2 přecházíme do značení

$$M_5 = (1,0,1,0,0,0,1,0) = 1 \cdot p_1 + 1 \cdot p_3 + 1 \cdot p_7,$$

ve kterém jsou uschopněny přechody t1 a t7. Oba tyto přechody jsou časované (trvající) a tedy i značení M_5 je trvajícím.

Provedení posloupnosti přechodů $\sigma = t5, t9, t2$ je okamžité a vnější pozorovatel, který je schopen vnímat pouze stavy, které po určitou dobu trvají, bude vývoj sítě

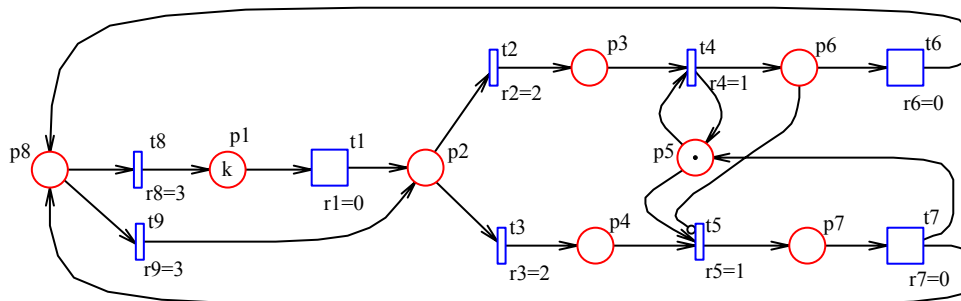
$$M_2[t5 > M_3[t9 > M_4[t2 > M_5 \tag{*}$$

chápat jako jednorázový akt $M_2[\sigma > M_5$.

Nyní uvažujme alternativu (b) přidělení priorit - viz obr.4.3.1.b a startujme interpretační proces (hru tokenů) vedoucí ze značení M_2 do značení M_5 . Posloupnost přechodů a značení bude následující:

$$M_2[t9 > M_3'[t2 > M_4'[t2 > M_5, \tag{**}$$

kde při značení M_2 jsou uschopněny přechody t8 a t9 s prioritou 3, při značení M_3' jsou uschopněny přechody t2 a t3 s prioritou 2 a při značení M_4' jsou uschopněny přechody t4 a t5 s prioritou 1.



Obr. 4.3.1.b Časovaná Petriho síť s prioritami - varianta (b)

Hlavní rozdíl mezi posloupnostmi (*) a (**) spočívá v tom, že provedení přechodu t5 v posloupnosti (*) je deterministické, zatímco v v posloupnosti (**) je výsledkem řešení konfliktu.

Konflikty.

- Konflikty jsou řešeny spíše na základě pravděpodobnostního rozložení než na základě čistého nedeterminismu. Jestliže např. 80% procent požadavků na vstup do databáze vyžaduje čtení a 20% zápis, pak přechody t2 (čtení) a t3 (zápis) je vhodné provádět s pravděpodobnostmi 0.8 a 0.2 - viz obr.4.3.1.
- Pojem efektivního konfliktu (viz definice 2.1.6) je třeba pro Petriho síť s prioritami redefinovat takto:

Definice 4.3.3:

Přechod t je v *efektivním konfliktu* s přechodem t' při značení M , jestliže:

- přechod t' je povolen při značení M ,
- přechod t je proveditelný (je uschopněn) při značení M ,
- po provedení přechodu t stupeň proveditelnosti přechodu t' se zmenší.

Definice pojmu *stupeň proveditelnosti přechodu*, formulovaná pro PN-systémy (viz def.2.1.6), zůstává beze změny i pro PN-systémy s prioritou.

Vlastnosti PN-systémů s prioritami.

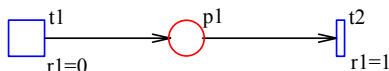
V tomto odstavci se budeme zabývat vlivem zavedení priority na vlastnosti PN-systémů. Vlastnosti PN-systémů mohou být rozděleny do dvou skupin:

- Vlastnosti, které musí platit pro všechny stavy systému (safety properties, invariant properties): omezenost, bezpečnost, absence uzamčení,...
- Vlastnosti, které platí jen pro některé stavy, které však mohou v budoucnosti vždy nastat (eventuality properties, progress properties): dosažitelnost (zda dané značení může být dosaženo), živost (zda daný přechod může být proveden),...

Zavedení priorit může pouze zmenšit počet stavů (značení) PN-systému. Proto všechny vlastnosti prvního druhu (safety properties) se přenáší z PN-systémů bez priorit na PN-systémy s prioritami. Vlastnosti druhého druhu (eventuality properties) zachovány být nemusí. Má-li však vlastnost druhého druhu PN-systém s prioritami, pak ji má i příslušný PN-systém bez priorit.

- *Omezenost.*

Omezený PN-systém bez priorit zůstává omezeným i po zavedení priorit. Naopak neomezený PN-model se po vhodném zavedení priorit může stát omezeným. To je ilustrováno jednoduchým PN-systémem z obr.4.3.2. Počet tokenů v místě p_1 může neomezeně růst díky opakovanému provádění přechodu t_1 . Definujeme-li priority $\rho(t_1)=0$ a $\rho(t_2)=1$, pak PN-systém s prioritami se stává bezpečným, tj. 1-omezeným (za předpokladu $M_0(p_1)<2$).



Obr. 4.3.2

- *Dosažitelnost.*

Množinu dosažitelných značení $RS(M_0)$ a graf dosažitelnosti $RG(M_0)$ systému s prioritami získáme snadno redukcí množiny $RS(M_0)$ a grafu $RG(M_0)$ systému bez priorit. Zavedení priorit může radikálně zmenšit množinu RS a graf RG . Je-li značení M' dosažitelné ze značení M v PN-systému s prioritou, pak je dosažitelné i v odpovídajícím PN-systému bez priority, nikoliv však obráceně.

- *Živost.*

Živý PN-systém může zavedením nevhodné prioritní struktury živost ztratit. Obráceně PN-model, který není živý, se živým může stát přidáním vhodné prioritní struktury.

Stavová analýza PN-systémů s prioritami.

Konstrukce grafu dosažitelnosti pro PN-systémy s prioritami je obdobná konstrukci pro PN-systémy bez priorit. Jediný rozdíl spočívá v tom, že pravidlo proveditelnosti (enabling rule) pro PN s prioritami vede k menšímu grafu dosažitelnosti co do počtu uzlů i hran. I tyto menší grafy dosažitelnosti dále zmenšujeme (redukujeme) tak, že "nevnímáme" značení, ve kterých jsou "uschopněny" přechody s prioritou vyšší než je daná priorita.

Definice 4.3.4:

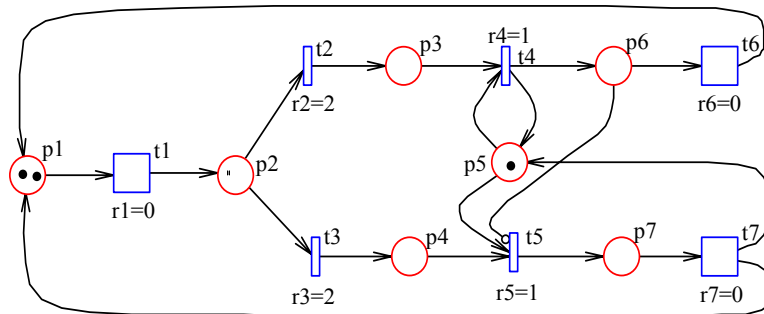
Redukovaný graf dosažitelnosti úrovně L je graf dosažitelnosti, který neobsahuje značení (stavy) ve kterých jsou proveditelné přechody s prioritou větší než L .

(Neredukovaný) **graf dosažitelnosti** je redukovaný graf dosažitelnosti úrovně L , kde L je nejvyšší priorita použitá v PN-systému.

Maximálně redukovaným grafem dosažitelnosti je redukovaný graf dosažitelnosti úrovně 0. Tento graf obsahuje pouze **trvající stavy** (tangible states), ve kterých PN-systém po určitou dobu setrvává a neobsahuje žádné **přechodné stavy** (vanishing states), kterými PN-systém pouze prochází aniž by v nich setrval sebekratší dobu.

Příklad 4.3.2:

Na obr. 4.3.3 je zobrazen poněkud zjednodušený PN-systém s prioritami popsány v příkladu 4.3.1 (význam míst a přechodů je nezměněný).

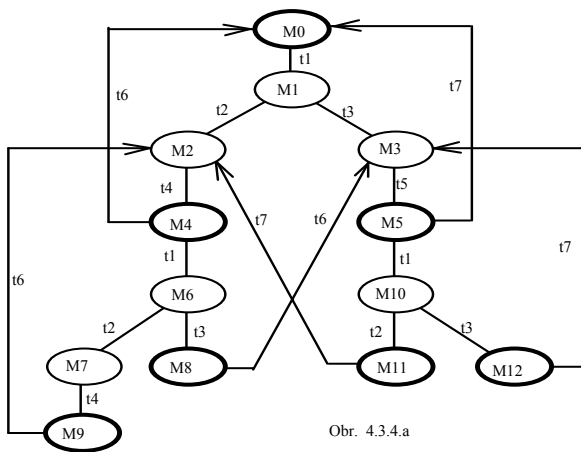


Obr. 4.3.3

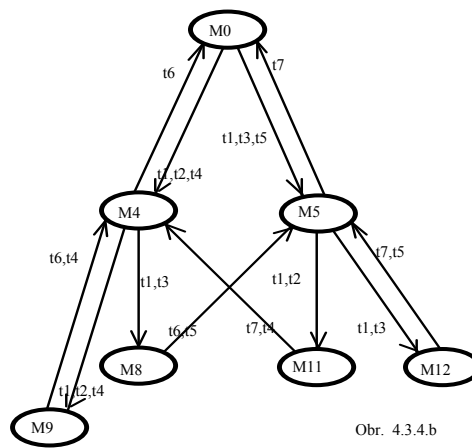
Dále je konstruována množina dosažitelných značení. Tučně jsou vyznačeny trvající stavy a časované (trvající) přechody.

| | | | |
|------------------------|----|--|-------------------------------|
| M0 = 2p1 | p5 | | t1 →M1 |
| M1 = p1+ p2+ | p5 | | t2→M2, t3→M3 |
| M2 = p1+ p3+ | p5 | | t4→M4 |
| M3 = p1+ p4+ p5 | | | t5→M5 |
| M4 = p1+ p5+ p6 | | | t1 →M6, t6 →M6 |
| M5 = p1+ p7 | | | t1 →M10, t7 →M0 |
| M6 = p2+ p5+ p6 | | | t2→M7, t3→M8 |
| M7 = p3+ p5+ p6 | | | t4→M9 |
| M8 = p4+ p5+ p6 | | | t6 →M3 |
| M9 = p5+ 2p6 | | | t6→M2 |
| M10 = p2+ p7 | | | t2→M11, t3→M12 |
| M11 = p3+ p7 | | | t7 →M2 |
| M12 = p4+ p7 | | | t7 →M3 |

Na obrázku 4.3.4.a je zobrazen úplný (neredukovaný) graf dosažitelnosti PN-systému z obr.4.3.3 a na obrázku 4.3.4.b redukovaný graf dosažitelnosti úrovně 0.



Obr. 4.3.4.a



Obr. 4.3.4.b

Z grafů dosažitelnosti (neredukovaného i redukovaného) vyplývá, že PN-systém je omezený, živý, reverzibilní a bez uzamčení.

Appendix A:

Multimnožiny

Pojem multimnožiny (multiset, bag) je užitečné zobecnění pojmu množiny. Multimnožina, na rozdíl od množiny, může obsahovat vícenásobné výskyty téhož prvku. Množina je speciální případ množiny, ve které se každý prvek vyskytuje nanejvýše jednou.

Definice A1:

Nechť S je libovolná neprázdná a konečná množina. **Multimnožina m nad množinou S** (multiset /bag/ over domain S) je zobrazení $m: S \rightarrow N$, kde $N = \{0, 1, 2, \dots\}$. Nezáporné celé číslo $m(s) \in N$ udává **počet výskytů prvku $s \in S$** v multimnožině m .

Poznámky A1:

1. Multimnožinu m zpravidla zapisujeme ve tvaru symbolického součtu:

$$\sum_{s \in S} m(s) \cdot s.$$

Nezáporná celá čísla $\{m(s) : s \in S\}$ nazýváme **koeficienty multimnožiny m** a číslo $m(s)$ **koeficientem prvku $s \in S$** . Symbolem

$$|m| = \sum_{s \in S} m(s)$$

označujeme **početnost multimnožiny m** (cardinality of a bag).

2. **Prvek $s \in S$ patří do multimnožiny m** právě tehdy, když $m(s) \neq 0$. Tuto skutečnost zapisujeme $s \in m$.
3. **Prázdná multimnožina** je multimnožina, jejíž všechny koeficienty jsou nulové. Tuto multimnožinu označujeme \emptyset .
4. **Množinu všech multimnožin nad množinou S** (multiset /bag/ space) budeme označovat S_{MS} . Na množině všech multimnožin nad danou množinou lze definovat různé operace (definice A2) a relace (definice A3).

Definice A2:

Pro multimnožiny $m_1, m_2 \in S_{MS}$ a pro $n \in N$ lze definovat následující binární operace:

- $m_1 + m_2 =_{\text{def}} \sum_{s \in S} (m_1(s) + m_2(s)) \cdot s$ **součet multimnožin**
- $n * m_1 =_{\text{def}} \sum_{s \in S} (n \cdot m_1(s)) \cdot s$ **násobení multimnožiny celým nezáporným číslem**

Poznámky A2:

1. Algebraický systém $\langle S_{MS}; + \rangle$ je komutativní monoid.
2. Algebraický systém $\langle S_{MS}; N; +, * \rangle$ je heterogenní algebraická struktura (typu vektorový prostor).

Definice A3:

Pro multimnožiny $m_1, m_2 \in S_{MS}$ lze definovat následující binární relace:

- $m_1 = m_2 \Leftrightarrow_{\text{def}} (\forall s \in S) [m_1(s) = m_2(s)]$ **rovnost multimnožin**
- $m_1 \leq m_2 \Leftrightarrow_{\text{def}} (\forall s \in S) [m_1(s) \leq m_2(s)]$ **uspořádání multimnožin**

Poznámky A3:

- Na základě relací $=, \leq$ lze definovat další relace $\neq, <$ a také $\geq, >$:
 - $m_1 \neq m_2 \Leftrightarrow_{\text{def}} \neg(m_1 = m_2)$
 - $m_1 < m_2 \Leftrightarrow_{\text{def}} (m_1 \leq m_2) \wedge (m_1 \neq m_2)$
 - $m_1 \geq m_2 \Leftrightarrow_{\text{def}} m_2 \leq m_1$
 - $m_1 > m_2 \Leftrightarrow_{\text{def}} m_2 < m_1$
- Je-li $m_1 \geq m_2$, lze na multimnožinách definovat další operaci:
 - $m_1 - m_2 =_{\text{def}} \sum_{s \in S} (m_1(s) - m_2(s)) \cdot s$... **rozdíl multimnožin**
- Relace \leq představuje relaci částečného (neúplného) a neostrého uspořádání na množině S_{MS} a relace $<$ relaci ostrého uspořádání. V relačním systému $\langle S_{MS}, \leq \rangle$ existuje jediný minimální prvek (prázdná multimnožina) a ani jeden maximální prvek.

Příklad A1:

Nechť $S = \{a, b, c, d\}$, $m_1 = 2'a + 3'b + d$, $m_2 = 5'a + 3'b + 2'c + 4'd$.

Potom:

- $m_1 + m_2 = 7'a + 6'b + 2'c + 5'd$,
- $2 * m_1 = 4'a + 6'b + 2'd$,
- $m_1 - m_2$... není definováno,
- $m_2 - m_1 = 3'a + 2'c + 3'd$,
- $m_1 \neq m_2$,
- $m_1 < m_2$,
- $m_1 \leq m_2$.

Při zápisu multimnožin jsme využívali následující konvence:

- Koeficienty zapisujeme s apostrofy (abychom mohli zapisovat i multimnožiny nad množinami přirozených čísel).
- Koeficient 1 nezapíšeme.
- Symbolický člen s koeficientem 0 nezapíšeme.

Appendix B:

Markovské procesy

Základní pojmy

Uvažujme soustavu libovolné povahy, která se v každém okamžiku t (z jisté množiny okamžiků) může nacházet vždy právě v jednom stavu i (z jisté množiny stavů I). Proces probíhající v soustavě může být:

- **deterministický**: stav soustavy v každém okamžiku $t \in T$ je vždy reprezentován jediným určitým prvkem množiny I ,
- **stochastický**: stav soustavy v každém okamžiku $t \in T$ je reprezentován pravděpodobnostním rozdělením na možině I .

Poznamenejme, že deterministický proces můžeme považovat za speciální případ stochastického procesu, kdy v každém okamžiku jsou pravděpodobnosti všech stavů rovny nule, s výjimkou jediného, jehož pravděpodobnost je rovna jedné.

Podle charakteru množiny časových okamžiků rozlišujeme procesy:

- s **diskrétním časem**: $T = \{0, 1, 2, \dots\}$, resp. $T = \{t_0, t_1, t_2, \dots\}$,
- se **spojitým časem**: $T = \langle 0, \infty \rangle$.

Stochastické procesy s diskretním časem se často nazývají stochastickými posloupnostmi nebo řetězci.

Podle charakteru stavové množiny I rozlišujeme procesy:

- s **diskrétní množinou stavů** (konečnou nebo spočetně nekonečnou),
- se **spojitou množinou stavů** (vždy nespočetně nekonečnou).

Stav soustavy x_t v okamžiku t může obecně záviset na všech předchozích stavech (tj. na množině stavů $\{x_\tau: \tau=0, 1, 2, \dots, t-1\}$ v případě procesu s diskretním časem, nebo na množině stavů $\{x_\tau: \tau \in \langle 0, t \rangle\}$ v případě procesu se spojitým časem τ). Stochastický proces nazýváme **markovským procesem**, má-li následující (markovskou) vlastnost: stav soustavy v bezprostředně následujícím okamžiku závisí pouze na přítomném stavu soustavy a nikoliv na stavech minulých. Markovský proces nazýváme **stacionárním (homogenním)**, jestliže způsob závislosti bezprostředně budoucího stavu na přítomném stavu se nemění s časem.

Markovské řetězce

Uvažujme stochastický proces s diskretním časem $T = \{0, 1, 2, \dots\}$ a s konečnou (a tedy diskretní) množinou stavů $I = \{1, 2, \dots, r\}$. Stav soustavy x_t v libovolném okamžiku $t \in T$ je popsán vektorem pravděpodobností jednotlivých stavů

$$p(t) = (p_1(t), p_2(t), \dots, p_r(t))^T,$$

kde

$$p_i(t) = P(x_t = i), \quad i \in I.$$

Pro všechna $t \in T$ platí:

$$\begin{aligned} 0 \leq p_i(t) \leq 1, \quad i \in I, \\ p_1(t) + p_2(t) + \dots + p_r(t) = 1. \end{aligned}$$

Stochastickou posloupnost $\{x_t: t=0, 1, 2, \dots\}$ nazýváme **markovským řetězcem**, jestliže platí

$$P(x_{t+1} = i_{t+1} / x_0 = i_0, x_1 = i_1, \dots, x_t = i_t) = P(x_{t+1} = i_{t+1} / x_t = i_t).$$

Nezávisí-li pravděpodobnost

$$P(x_{t+1} = i / x_t = j) = p_{ij}(t)$$

na čase, tj. můžeme-li psát

$$p_{ij}(t) = p_{ij},$$

pak se jedná o **homogenní markovský řetězec**. Matici

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1r} \\ p_{21} & p_{22} & \dots & p_{2r} \\ \dots & \dots & \dots & \dots \\ p_{r1} & p_{r2} & \dots & p_{rr} \end{pmatrix}$$

nazýváme **maticí přechodu** homogenního markovského řetězce. Veličina p_{ij} udává pravděpodobnost, že soustava, která je v daném okamžiku ve stavu $i \in I$ bude v následujícím okamžiku ve stavu $j \in I$. Platí:

$$\begin{aligned} 0 \leq p_{ij} \leq 1, \quad i, j \in I, \\ p_{i1} + p_{i2} + \dots + p_{ir} = 1, \quad i \in I. \end{aligned}$$

Z elementárních pouček teorie pravděpodobnosti vyplývá:

$$p_j(t+1) = p_1(t) \cdot p_{1j} + p_2(t) \cdot p_{2j} + \dots + p_r(t) \cdot p_{rj}, \quad j \in I,$$

neboli, zapíšeme-li tyto rovnice v maticovém tvaru

$$\mathbf{p}(t+1) = \mathbf{P}^T \cdot \mathbf{p}(t). \tag{A2.1}$$

Vyřešením této maticové diferenční rovnice s počáteční podmínkou $\mathbf{p}(0)$ dostáváme

$$\mathbf{p}(t) = (\mathbf{P}^T)^t \cdot \mathbf{p}(0). \tag{A2.2}$$

Homogenní markovský řetězec je tedy zadán, jsou-li dány:

- vektor počátečních pravděpodobností $\mathbf{p}(0)$,
- matice přechodu \mathbf{P} .

Předpokládejme nyní, že pro $t \rightarrow \infty$ existuje limita

$$\lim_{t \rightarrow \infty} \mathbf{p}(t) = \mathbf{p} = (p_1, p_2, \dots, p_r),$$

nezávislá na $\mathbf{p}(0)$. Markovský řetězec pak nazýváme **stochasticky stabilním** a vektor \mathbf{p} vektorem **stacionárních pravděpodobností**. Z (A2.1) pro stacionární pravděpodobnosti vyplývá

$$\mathbf{p} = \mathbf{P}^T \cdot \mathbf{p}, \tag{A2.3}$$

neboli

$$(\mathbf{P}^T - \mathbf{I}) \cdot \mathbf{p} = \mathbf{0}. \tag{A2.4}$$

Odtud a z podmínky

$$p_1 + p_2 + \dots + p_r = 1$$

můžeme stacionární pravděpodobnosti vypočítat.

Markovské procesy (se spojitým časem)

Uvažujme nyní stochastický proces se spojitým časem $t \in \langle 0, \infty \rangle$ a konečnou množinou stavů $I = \{1, 2, \dots, r\}$. Tento proces nazveme **homogenním markovským procesem**, jsou-li splněny následující dvě podmínky:

1. posloupnost stavů, tak jak za sebou následují bez ohledu na čas, tvoří homogenní markovský řetězec s **maticí přechodu**

$$\mathbf{P} = \begin{pmatrix} 0 & p_{12} & \dots & p_{1r} \\ p_{21} & 0 & \dots & p_{2r} \\ \dots & \dots & \dots & \dots \\ p_{r1} & p_{r2} & \dots & 0 \end{pmatrix}$$

2. pravděpodobnost toho, že soustava, která je v okamžiku t ve stavu i , změní v infinitesimálním intervalu $(t, t+dt)$ svůj stav je $\alpha_i dt$. Konstanta α_i se nazývá intenzitou výstupu i -tého stavu a vektor

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_r)$$

vektorem intenzit výstupu.

Z elementárních pouček teorie pravděpodobnosti vyplývá

$$\begin{aligned} P(x_{t+dt}=j/x_t=i) &= \alpha_i dt \cdot p_{ij} \\ P(x_{t+dt}=i/x_t=i) &= 1 - \alpha_i dt \end{aligned}$$

a odtud

$$p_j(t+dt) = p_1(t)\alpha_1 dt + p_{1j} + \dots + p_j(t)(1-\alpha_j dt) + \dots + p_r(t)\alpha_r dt + p_{rj}, \quad j \in I.$$

Maticový zápis tohoto systému rovnic je

$$\mathbf{p}(t+dt) = (\mathbf{I} + \mathbf{Q}dt)^T \mathbf{p}(t), \tag{A2.5}$$

kde

$$\mathbf{p}(t) = (p_1(t), p_2(t), \dots, p_r(t))^T,$$

$$\mathbf{Q} = (q_{ij}), \quad i, j \in I,$$

kde

$$q_{ii} = -\alpha_i, \quad i \in I,$$

$$q_{ij} = \alpha_i p_{ij}, \quad i, j \in I, \quad i \neq j.$$

Matrice \mathbf{Q} je tzv. **matice intenzit přechodu** homogenního markovského procesu. Rovnici (A2.5) lze upravit na tvar

$$(\mathbf{p}(t+dt) - \mathbf{p}(t))/dt = \mathbf{Q}^T \mathbf{p}(t),$$

neboli

$$\mathbf{p}'(t) = \mathbf{Q}^T \mathbf{p}(t). \tag{A2.6}$$

Tato vektorová diferenciální rovnice představuje soustavu r lineárních diferenciálních rovnic 1. řádu s konstantními koeficienty. Z této soustavy a z podmínky

$$1 = p_1(t) + p_2(t) + \dots + p_r(t), \quad t \in \langle 0, \infty \rangle,$$

lze nalézt všechny funkce $p_i(t)$, $i \in I$.

Homogenní markovský proces je tedy jednoznačně definován:

- vektorem počátečních pravděpodobností

$$\mathbf{p}(0) = (p_1(0), p_2(0), \dots, p_r(0))^T,$$

který udává počáteční podmínky řešení soustavy diferenciálních rovnic,

- maticí intenzit přechodů $\mathbf{Q} = (q_{ij})$, která udává koeficienty soustavy diferenciálních rovnic.

Předpokládejme nyní, že pro $t \rightarrow \infty$ existuje limita

$$\lim \mathbf{p}(t) = \mathbf{p} = (p_1, p_2, \dots, p_r)^T,$$

nezávislá na $\mathbf{p}(0)$. Markovský proces pak nazýváme **stochasticky stabilním** a vektor \mathbf{p} **vektorem stacionárních pravděpodobností**. Pro stacionární pravděpodobnosti platí $\mathbf{p}'(t) = \mathbf{0}$ a tedy podle (A2.6)

$$\mathbf{Q}^T \mathbf{p} = \mathbf{0}. \tag{A2.7}$$

Z této soustavy lineárních algebraických rovnic a z podmínky

$$p_1 + p_2 + \dots + p_r = 1$$

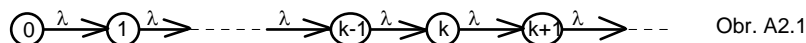
můžeme vypočítat stacionární (ustálené, limitní) pravděpodobnosti.

Je-li markovský proces stochasticky stabilní, pak pro dostatečně velká t platí přibližná rovnost $\mathbf{p}(t) = \mathbf{p}$, tj. pravděpodobnosti jednotlivých stavů jsou již ustálené a dále nezávisí na čase. Říkáme potom, že soustava se nachází ve **ustáleném** (stacionárním) režimu. Počáteční fázi stochastického procesu označujeme pak jako **přechodový** (transientní) režim.

Homogenní markovské procesy často zobrazujeme pomocí diagramů ohodnocených orientovaných grafů. Stavům procesu odpovídají uzly, přechodům orientované hrany mezi uzly a intenzitám přechodů ohodnocení hran. Stavové přechody s nulovými intenzitami přitom neznázorňujeme.

Poissonův proces

Uvažujme speciální případ markovského procesu zadaného následujícím diagramem na obr.A2.1.



Matrice intenzit přechodů má následující jednoduchý tvar

$$\mathbf{Q} = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \dots \\ 0 & -\lambda & \lambda & 0 & \dots \\ 0 & 0 & -\lambda & \lambda & \dots \end{pmatrix}$$

... ..

Je zřejmé, že Poissonův proces nemá stacionární režim (stacionární pravděpodobnosti neexistují - proces přechází do stále vyšších a vyšších stavů). Vyřešíme systém diferenciálních rovnic (A2.6) popisující přechodový děj. Pro výše uvedený speciální tvar matice intenzit přechodů má tento systém následující tvar:

$$\begin{aligned} p_0'(t) &= -\lambda \cdot p_0(t), \\ p_k'(t) &= \lambda \cdot (p_{k-1}(t) - p_k(t)), \quad k=1,2,\dots \end{aligned} \tag{A2.8}$$

Soustavu doplníme počátečními podmínkami:

$$p_0(0)=1, p_k(0)=0 \text{ pro } k=1,2,\dots$$

a budeme ji řešit pomocí Laplaceovy transformace. Obrazem soustavy diferenciálních rovnic (A2.8) je soustava algebraických rovnic

$$\begin{aligned} s \cdot p_0^*(s) - 1 &= -\lambda \cdot p_0^*(s), \\ s \cdot p_k^*(s) &= \lambda \cdot (p_{k-1}^*(s) - p_k^*(s)), \quad k=1,2,\dots \end{aligned} \tag{A2.9}$$

neboli soustava

$$\begin{aligned} p_0^*(s) &= 1/(s+\lambda), \\ p_k^*(s) &= \lambda \cdot p_{k-1}^*(s)/(s+\lambda), \quad k=1,2,\dots \end{aligned} \tag{A2.10}$$

neboli

$$p_k^*(s) = \lambda^k / (s+\lambda)^{k+1}, \quad k=0,1,2,\dots \tag{A2.10}$$

Provedeme-li nyní inverzní transformaci k Laplaceově, obdržíme

$$p_k(t) = (\lambda t)^k \cdot e^{-\lambda t} / k!, \quad k=0,1,2,\dots \tag{A2.11}$$

Poslední vzorec udává **Poissonovo pravděpodobnostní rozdělení** s parametrem $a = \lambda t$, tak jak je známe z teorie pravděpodobnosti.

Exponenciální rozdělení

Uvažujme stacionární proud bodových událostí vznikajících v náhodné okamžiky. Následující tři tvrzení jsou ekvivalentní:

- (1) Pravděpodobnost, že událost nastane v intervalu $(t, t+dt)$ je $\lambda \cdot dt$, $\lambda > 0$.
- (2) Náhodná veličina k , udávající počet událostí za dobu t , tj. během intervalu $(0, t)$, má Poissonovo rozdělení (A2.11).
- (3) Náhodná veličina θ udávající dobu mezi dvěma bezprostředně za sebou následujícími událostmi má **exponenciální rozdělení** s distribuční funkcí

$$F(t) = P(\theta \leq t) = 1 - e^{-\lambda t}, \quad t \geq 0. \tag{A2.12}$$

Ekvivalence uvedených tvrzení bude dokázána, dokážeme-li platnost řetězce implikací $(1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (1)$. Implikace $(1) \Rightarrow (2)$ byla již dokázána v předchozím odstavci. Důkazem implikace $(2) \Rightarrow (3)$ je následující řetězec rovností

$$F(t) = P(\theta \leq t) = 1 - P(\theta > t) = 1 - p_0(t) = 1 - (\lambda t)^0 \cdot e^{-\lambda t} / 0! = 1 - e^{-\lambda t}.$$

Důkazem implikace $(3) \Rightarrow (1)$ je následující řetězec rovností

$$P(\theta \in (t, t+dt) / \theta > t) = P(t < \theta \leq t+dt) / P(\theta > t) = F'(t) dt / (1 - F(t)) = \lambda \cdot dt.$$

Hustota pravděpodobnosti (frekvenční funkce) exponenciálního rozdělení je derivací distribuční funkce:

$$f(t) = F'(t) = \lambda \cdot e^{-\lambda t}, \quad t \geq 0. \tag{A2.13}$$

Pro $t < 0$ je $f(t) = F(t) = 0$. Připomeňme ještě názorný význam hustoty pravděpodobnosti:

$$f(t) dt = P(t < \theta \leq t+dt).$$

Pro střední hodnotu a disperzi exponenciálně rozdělené náhodné veličiny θ snadno odvodíme:

$$E\{\theta\} = 1/\lambda, \quad D\{\theta\} = 1/\lambda^2. \tag{A2.14}$$

Appendix C:

Problém hromadné obsluhy zobrazený a řešený pomocí GSPN

1. Popis problému

Uvažujme systém hromadné obsluhy s homogenním proudem vstupních požadavků na obsluhu (jobs) s homogenní jednofázovou obsluhou zabezpečenou několika identickými aparáty obsluhy (servers) a s frontou (queue) omezené délky do které se řadí požadavky na obsluhu v případě, že všechny aparáty obsluhy jsou obsazené.

Budeme předpokládat, že vstupní proud požadavků i doba obsluhy jsou náhodné veličiny: vstupní proud je poissonovský (počet požadavků příšlých za jednotku má vždy totéž Poissonovo rozdělení, neboli – což je totéž - doba mezi příchody dvou bezprostředně za sebou následujících požadavků má exponenciální rozdělení) a doba obsluhy každého požadavku kterýmkoliv aparátem obsluhy má vždy totéž exponenciální rozdělení. Tyto předpoklady umožňují formulovat jednoduchý markovský model náhodných procesů pobíhajících v systému.

Kvantitativně je systém charakterizován následujícími parametry:

- strukturní parametry:
 - S ... počet aparátů obsluhy v systému (maximální počet požadavků, které mohou být v systému obsluhovány současně); $S \geq 1$
 - Q ... maximálně možná délka fronty (maximální počet požadavků čekajících v systému na obsluhu); $Q \geq 1$
 - N ... $S+Q$... maximální celkový počet požadavků (obsluhovaných i čekajících), který se v systému může nacházet
- parametry náhodných procesů:
 - λ ... intenzita vstupního proudu požadavků ($1/\lambda$... střední hodnota doby mezi příchody požadavků); $\lambda > 0$
 - μ ... intenzita obsluhy požadavků ($1/\mu$... střední hodnota doby obsluhy jednotlivých požadavků); $\mu > 0$

Systém hromadné obsluhy o dané (výše popsané) struktuře je jednoznačně určen čtveřicí parametrů (S, Q, λ, μ).

Stavové veličiny systému:

- s ... aktuální počet obsluhovaných požadavků (= aktuální počet pracujících aparátů obsluhy)
- q ... aktuální počet požadavků čekajících ve frontě na obsluhu (= aktuální délka fronty)
- ($n=s+q$... aktuální počet všech požadavků nacházejících se v systému)

2. Model systému pomocí GSPN

Popsaný systém hromadné obsluhy budeme nyní modelovat pomocí Petriho sítí (PN), stochastických Petriho sítí (SPN) a zobecněných stochastických Petriho sítí (GSPN). Na obr. 1 je společně grafické zobrazení všech těchto sítí s konkrétními hodnotami parametrů $S=2$ a $Q=3$. Pro zobrazení obecné PN muselo by místo p_3 mít počáteční značení nikoliv $2 \cdot e$, ale $S \cdot e$ a hrany incidentní s přechodem t_4 násobnosti nikoliv $3 \cdot e$ a $4 \cdot e$, ale $Q \cdot e$ a $(Q+1) \cdot e$.

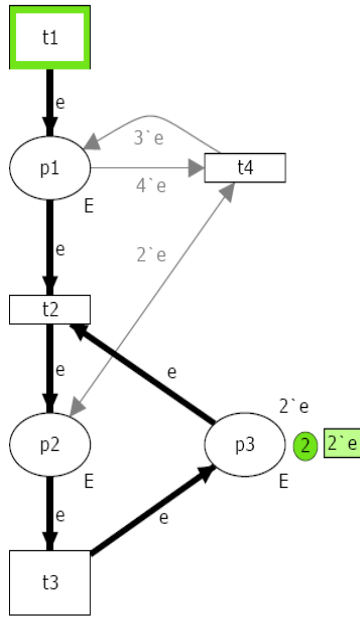
Přísně vzato, na tomto obrázku je zobrazena klasická PN, u které doba provádění přechodů není vůbec reflektována a záleží jen na pořadí provádění přechodů. Obrázek nám bude představovat SPN, jestliže předpokládáme, že všechny přechody mají přiřazenu dobu trvání a to náhodnou a exponenciálně rozloženou a ke každému přechodu je přiřazena hodnota parametru tohoto rozdělení (intenzita provádění přechodu). Obrázek bude představovat GSPN, jestliže předpokládáme, že přechody jsou dvojího druhu okamžité (immediate) a časované (timed) s exponenciální dobou trvání. Na obr. 1 jsou tyto typy přechodů rozlišeny různými velikostmi obdélníků zobrazujících přechody (tenčí okamžité, tlustší časované). Podrobnou a úplnou definici PN, SPN a GSPN lze nalézt v příslušných kapitolách učebních textů.

Petriho síť na obr. 1 byla vytvořena programovým nástrojem CPNTools, který je schopen pracovat s tokeny mnoha různých typů. V uvažovaném případě klasické PN, máme tokeny jediného typu $E = \{e\}$: v každém místě se může nacházet libovolná multimnožina tokenů nad E a násobnost hran může být též dána multimnožinou nad E . Např. popiska $2 \cdot e$ u místa značí, že v daném místě se nachází dva tokeny „e“ a podobně popiska $3 \cdot e$ u hrany znamená, že hrana má násobnost tři.

Petriho síť zobrazená na obrázku je uložena v souboru `obsluha_3.cpn` a je pomocí programového nástroje CPNTools interpretovatelná (simulovatelná), analyzovatelná, modifikovatelná a zobecnitelná.

Uvedeme význam jednotlivých míst a přechodů:

- p_1 ... místo zobrazuje frontu; počet tokenů udává počet požadavků čekajících na obsluhu
- p_2 ... místo zobrazuje obsluhu; počet tokenů udává počet právě obsluhovaných požadavků
- p_3 ... místo zobrazuje volné aparáty obsluhy; počet tokenů udává počet volných aparátů obsluhy
- t_1 ... časovaný vstupní přechod modelující příchod požadavků do systému; intenzita provádění přechodu λ nezávisí na značení sítě (intenzita je marking independent)
- t_2 ... okamžitý přechod modelující zahájení obsluhy
- t_3 ... časovaný výstupní přechod modelující proces obsluhy; intenzita provádění přechodu (a současně i intenzita výstupu ze systému) je závislá na značení místa p_2 : je rovna intenzitě μ vynásobené aktuálním počtem tokenů v místě p_2 , tj. je rovna $s \cdot \mu$ (intenzita je marking dependent)
- t_4 ... okamžitý přechod modelující odmítnutí příchozího požadavku jsou-li fronta a tedy i celý systém plné, tj. platí-li $q=Q$ resp. $n=N$; násobnosti hran incidentních s tímto přechodem zajišťují, že počet tokenů v p_1 nemůže překročit maximální hodnotu Q (při překročení hodnoty na $Q+1$ se okamžitě vrací na Q)



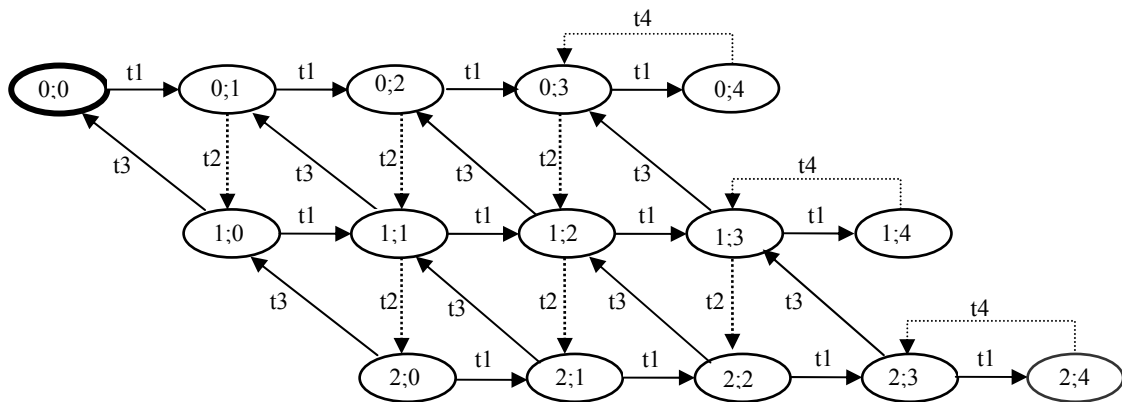
Tokeny generované přechodem t1 (tento přechod je vždy proveditelný neboť nemá žádná vstupní místa) představují požadavky na obsluhu. Postupují přes místo p1 (stav čekání ve frontě), přechod t2 (přidělení aparátu obsluhy z místa p3 a zahájení obsluhy), místo p2 (stav provádění obsluhy) a odcházejí ze systému prostřednictvím přechodu t3 (ukončení obsluhy a vrácení aparátu obsluhy do p3).

Okamžitý přechod t4 zabezpečuje, aby počet tokenů v místě p1 (tj. počet požadavků ve frontě) nepřekročil hodnotu 3. V okamžiku, kdy by do místa p1 vstoupil 4. token, přechod t4 neprodleně všechny 4 tokeny odebere a do místa p1 vrátí jen 3 tokeny. Testovací hrana spojující přechod t4 a místo p2 s násobností 2 zařizuje, aby přechod t4 proveditelný jen tehdy, není-li proveditelný přechod t2, tj. aby systém neodmítl požadavek v situaci, kdy jeden z aparátů obsluhy není využit.

Obr.1 PN - systém hrom. obsluhy

3. Graf dosažitelnosti PN – stavový model systému

PN systém zobrazený na obr. 1 má konečný a jen malý počet různých značení (stavů). Základní metodou analýzy takových PN systému je konstrukce stavového modelu, kterým je tzv. graf dosažitelnosti. Tento graf je pro náš PN systém zobrazen na obr.2.



Obr.2 Graf dosažitelnosti PN systému zobrazeného na obr.1

Ovály zobrazují stavy (značení, markings) a dvě celá čísla uvedená uvnitř oválu udávají hodnoty stavových veličin s a q , které jednoznačně určují stav celého PN systému (viz kap.1). Stav PN systému je určen rozmístěním tokenů v místech PN systému. Hodnota veličiny s je rovna počtu tokenů v místě p2 a hodnota veličiny q je rovna počtu tokenů v místě p1. Počet tokenů v místě p3 netřeba udávat, protože se vždy rovná hodnotě $2 - (\text{počet tokenů v místě p2})$.

„Řádky“ grafu odpovídají různým hodnotám stavové veličiny s :

- první řádek odpovídá hodnotě $s=0$; všem 5 stavům v tomto řádku je společné, že v nich není obsluhován žádný požadavek

- druhý řádek odpovídá hodnotě $s=1$; všem 5 stavům v tomto řádku je společné, že v nich je obsluhován právě jeden požadavek
- třetí řádek odpovídá hodnotě $s=2$; v každém z 5 stavů tohoto řádku jsou obsluhovány vždy dva požadavky

„Sloupce“ grafu odpovídají různým hodnotám stavové veličiny $n=s+q$ udávající celkový počet požadavků v systému: v prvním sloupci je to 0 požadavků, v druhém 1 požadavek, v třetím 2 požadavky, atd.

Na obr.2 je zobrazen graf dosažitelnosti PN systému s parametry $S=2$ a $Q=3$. Graf dosažitelnosti PN systému s obecnými parametry by měl $S+1$ řádků a v každém řádku $Q+2$ stavů a tedy by celkem měl $(S+1)(Q+2)$ stavů. Graf by byl větší, ale struktura obdobná. Všechny vlastnosti obecného systému můžeme proto studovat na jeho jednoduchém konkrétním příkladě.

Z grafu dosažitelnosti na obr.2 můžeme dedukovat:

- PN systém je omezený; místo p_1 je 4-omezené, místa p_2, p_3 jsou 2-omezená
- PN systém je živý; ať se nacházíme kterémkoliv stavu, vždy je možné přejít do stavu v němž je kterýkoliv (předem vybraný) přechod proveditelný, tj. žádný ze čtyř přechodů t_1, t_2, t_3, t_4 neztrácí vlastnost být znovu proveditelný
- PN systém je reverzibilní; z kteréhokoliv z 15 dosažitelných značení (stavů) je počáteční značení $(0;0)$ dosažitelné

Až dosud jsme náš PN systém traktovali jako nečasovanou (untimed) klasickou P/T (place/transition) Petriho síť. Přiblížíme se nyní blíže ke skutečnosti tím, že PN systém z obr.1 budeme považovat za časovanou **stochastickou Petriho síť** (SPN). Předpokládáme pak, že doba provedení každého přechodu je nenulová, náhodná, s exponenciálním rozdělením a charakterizovaná jediným číselným parametrem – intenzitou provádění přechodu (převrácená hodnota intenzity je střední hodnota doby provádění přechodu). Popis SPN získáme z popisu nečasované PN tak, že každému přechodu přiřadíme intenzitu jeho provádění. Intenzita daného přechodu může (ale nemusí) záviset i na stavu ve kterém se systém nachází (např. intenzita přechodu t_3 může být ve stavu $(2;2)$ jiná než ve stavu $(2;1)$). Přechody t_1, t_2, t_3, t_4 našeho příkladu můžeme rozdělit do dvou skupin: pomalé (t_1, t_3) a rychlé (t_2, t_4). Např. přechod t_3 představuje provádění obsluhy, která skutečně trvá; naproti tomu přechod t_2 představující zahájení obsluhy je proveden prakticky okamžitě (těmto přechodům přiřazujeme v SPN velmi velké intenzity, tj. prakticky nulové střední hodnoty). Na obr.2 jsou pomalé přechody vyznačeny plnou šipkou a rychlé přechody tečkovanou šipkou.

Zobrazení a řešení reálného systému - např. našeho systému hromadné obsluhy - ve tvaru SPN je z teoretického hlediska sice velmi dobře možné, v praxi však naráží na značné výpočetní problémy. Vede totiž na řešení velkého systému rovnic, a to diferenciálních zajímá-li nás obecné řešení nebo algebraických zajímá-li nás jen ustálené řešení. Navíc koeficienty těchto rovnic (utvořené z intenzit), jsou co do velikosti řádově extrémně odlišné. Získat řešení je pak náročné a přesnost výsledků omezená. Žádoucí je proto maximálně redukovat počet stavů a pracovat s koeficienty, které se řádově příliš neliší. Toho lze docílit, budeme-li považovat PN systém z obr.1 za **zobecněnou stochastickou Petriho síť** (GSPN). GSPN – jak již výše bylo zmíněno - uvažuje přechody dvojího druhu: okamžité a trvající s exponenciálně rozdělenou dobou trvání. Při konfliktu přechodů různých druhů vždy vyhrává přechod okamžitý a tato skutečnost vede ke zmenšení počtu dosažitelných stavů. Jestliže v nějakém stavu (značení) jsou proveditelné dva přechody, z toho jeden okamžitý a druhý trvající, pak druhý nebude nikdy proveden. Neboli: jestliže z nějakého uzlu grafu dosažitelnosti vychází dvě hrany, z nichž jedna zobrazuje okamžitý a druhá trvalý přechod, pak po druhé hraně se náhodný proces probíhající v síti nikdy ubírat nebude. Množina stavů (uzlů grafu dosažitelnosti) původního nečasovaného PN systému se tak rozpadne na tři části:

- stavy do nichž se náhodný proces nikdy nedostane

- stavy jimiž náhodný proces proběhne, ale nezastaví se v nich (doba setrvání v těchto stavech má nulovou délku) ... vanishing states
- stavy v nichž náhodný proces po určité době setrvává (doba setrvání v těchto stavech má nenulovou délku) ... tangible states

4. Redukovaný graf dosažitelnosti

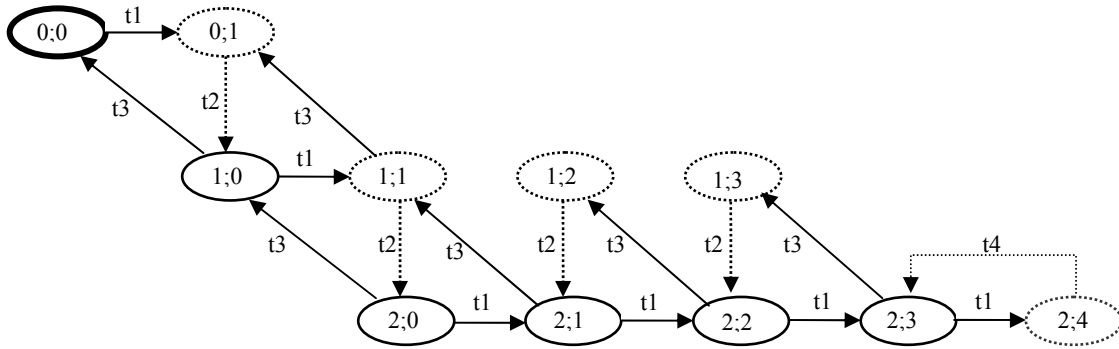
Počínaje touto kapitolou, považujeme uvažovaný PN systém za GSPN. Přejídy t_2, t_4 považujeme za okamžité a přejídy t_1, t_3 za trvající. Eliminací reálně nedosažitelných stavů z množiny všech teoreticky dosažitelných stavů získáme z grafu dosažitelnosti zobrazeného na obr.2 nový graf, tzv. **redukovaný graf dosažitelnosti** - viz obr.3. Původní množina teoreticky dosažitelných stavů se rozpadá na tři podmnožiny:

1. reálně nedosažitelné stavy: (0;2), (0;3), (0;4)
2. stavy jimiž systém jen prochází: (0;1), (1;1), (1;2), (1;3), (2;4)
3. stavy v nichž systém setrvává: (0;0), (1;0), (2;0), (2;1), (2;2), (2;3), (2;4)

Stavy 1.skupiny nejsou na redukovaném grafu vůbec zobrazeny, stavy 2.skupiny jsou zobrazeny tečkovaným oválem a stavy 3.skupiny jsou nezměněně nakresleny plným oválem. Podobně okamžité přejídy jsou zobrazeny tečkovanými šipkami a časované (trvající) přejídy plnými šipkami.

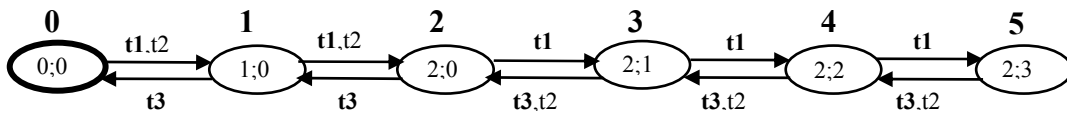
Správnost rozkladu lze ověřit analýzou grafu dosažitelnosti na obr.2 :

- Stav (0;2) je reálně nedosažitelný. Zdůvodnění:
 - Do tohoto stavu lze přejít jedině ze stavu (0;1) provedením přejídy t_1 .
 - Ve stavu (0;1) jsou proveditelné jen dva přejídy t_1 a t_2 .
 - Přejídy t_1 nebude nikdy proveden, protože vždy prohrává konflikt s okamžitým přejídy t_2 .
- Stav (0;3) je reálně nedosažitelný. Zdůvodnění:
 - Do tohoto stavu lze přejít jedině ze stavu (0;2).
 - O stavu (0;2) bylo již dokázáno, že je reálně nedosažitelný.
- Stav (0;4) je reálně nedosažitelný. Zdůvodnění:
 - Do tohoto stavu lze přejít jedině ze stavu (0;3).
 - O stavu (0;3) bylo již dokázáno, že je reálně nedosažitelný.
- Stavy (0;1), (1;1), (1;2), (1;3), (2;4) jsou stavy jimiž systém jen prochází, ale nesetrvává v nich. Zdůvodnění:
 - Každý z těchto stavů je reálně dosažitelný z počátečního stavu (značení).
 - V každém z těchto stavů je proveditelný okamžitý přejídy.
- Stavy (0;0), (1;0), (2;0), (2;1), (2;2), (2;3), (2;4) jsou stavy v nichž systém setrvává. Zdůvodnění:
 - Každý z těchto stavů je reálně dosažitelný z počátečního stavu (značení).
 - Ve všech těchto stavech jsou proveditelné jen časované přejídy.



Obr.3 Redukovaný graf dosažitelnosti GSPN systému

Stavy, jimiž systém jen prochází, nejsou zajímavé pro vnější pozorovatele a pro analýzu chování systému jsou bezvýznamné. Z vnějška pozorovatelné a pro návrh a analýzu systému důležité jsou pouze ty stavy, ve kterých systém setrvává (bez ohledu na dobu setrvání). Redukovaný graf dosažitelnosti je proto vhodné překreslit do přehlednějšího tvaru – viz obr.4. Na tomto obrázku nejsou zobrazeny stavy 2.skupiny (jimiž se jenom prochází, ale ve kterých se nesetrvává) a na druhé straně hrany mohou být popsány proveditelnými posloupnostmi přechodů, posloupnostmi, které jsou-li provedeny, převádějí systém z výchozího uzlu hrany (výchozího stavu) do koncového uzlu (následného stavu). V každé posloupnosti je právě jeden přechod časovaný (jeho identifikátor je vytištěn tučně). Délky posloupností v našem konkrétním případě jsou 1 nebo 2 – viz obr.4. Všimněme si také, že podstatné stavy, ve kterých systém setrvává, mohou být charakterizovány jediným parametrem $n=s+q$ namísto dvojice parametrů $(s;q)$.



Obr.4 Redukovaný graf dosažitelnosti GSPN systému – upravená verze

5. Graf přidruženého markovského procesu

V modelu GSPN jsou doby všech časovaných přechodů exponenciálně rozděleně, samozřejmě s obecně různými intenzitami. Základní vlastností exponenciálně rozdělené náhodné veličiny je neexistence paměti (memoryless property). Pravděpodobnost, že v infinitesimálním (nekonečně malém) intervalu $(t, t+dt)$ bude přechod dokončen nezávisí na tom jak dlouho je již přechod prováděn (probíhající proces si nepamatuje jak dlouho již probíhá) a má vždy stejnou pravděpodobnost αdt , kde α je jediný parametr exponenciálního rozdělení. Tento parametr má v GSPN (tedy i v SPN) význam intenzity provádění přechodu a jeho převrácená hodnota je střední hodnotou doby trvání.

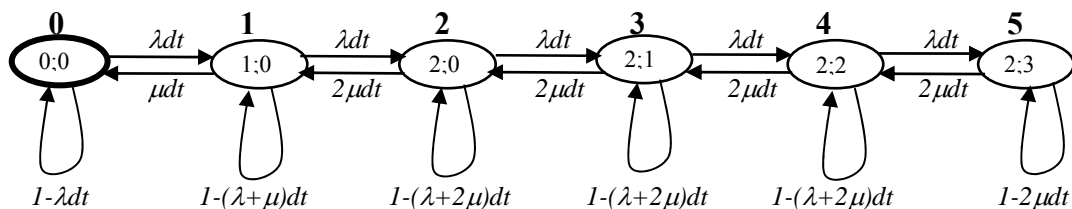
Memoryless vlastnost umožňuje definovat náhodný proces probíhající v GSPN (či SPN) jako markovský proces. Markovský proces může být zadán a zobrazen orientovaným grafem jehož uzly představují stavy procesu, hrany označují možné stavové změny a popisky hran udávají infinitesimální pravděpodobnosti příslušných stavových změn během infinitesimálního časového intervalu dt . Podrobné informace o náhodných a speciálně

markovských procesech (včetně přesných definic a odvození) lze nalézt v příloze B učebních textů).

Na obr.5 je zobrazen graf markovského procesu přidruženého k zobecněné stochastické Petriho síti (GSPN) zobrazené redukovaným grafem dosažitelnosti na obr.4. Množina stavů je u obou grafů totožná. Pravděpodobnosti přechodů během elementárního časového intervalu dt odvodíme snadno ze zadání zobrazovaného systému hromadné obsluhy:

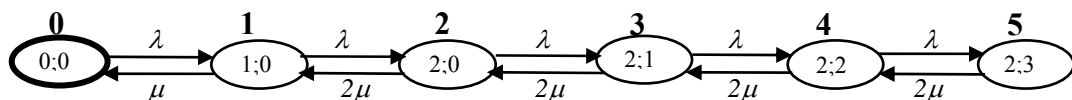
- všechny hrany mířící zleva doprava mají ohodnocení λdt – to souvisí s tím, že přechod t_1 generuje požadavky na obsluhu vždy s intenzitou λ nezávisle na stavu systému
- všechny hrany mířící zprava doleva mají ohodnocení $k\mu$, kde $k = 1, 2$ podle toho zda přechod t_3 značí obsluhu jen jednoho požadavku (v místě p_2 je jen jeden token) nebo značí obsluhu dvou požadavků (v místě p_2 jsou dva tokeny)
- součet ohodnocení všech hran vycházejících z každého uzlu (stavu) musí být pro každý uzel (stav) roven jedné, neboť během infinitesimální doby dt může systém přejít pouze do sousedního stavu a nebo se stav systému nezmění. Ohodnocení smyček (pravděpodobnosti, že systém během dt nezmění svůj stav) je tedy jednoznačně dáno ohodnocením „vodorovných“ hran a sice:

$$1 - (\text{součet ohodnocení vodorovných hran vycházejících z daného uzlu}) \cdot dt$$



Obr.5 Graf přidruženého markovského procesu

Bez ztráty informace můžeme přidružený markovský proces zobrazit úsporněji tak, že hrany grafu ohodnotíme přímo intenzitami přechodu – viz obr.6. Ze zjednodušeného grafu na obr.6 lze graf přidruženého markovského procesu snadno zrekonstruovat.



Obr.6 Zjednodušený graf přidruženého markovského procesu

6. Výpočet ustálených pravděpodobností stavů

Soustavu hromadné obsluhy, popsanou v kap.1, jsme reprezentovali pomocí GSPN (kap.2-4) a v předchozí kap.5 jsme tuto reprezentaci transformovali na rovnocennou reprezentaci pomocí markovského procesu se spojitým časem ($t \in \langle 0; \infty \rangle$) a konečnou množinou stavů ($i \in \{0,1,2,\dots,5\}$). Tento markovský proces je plně definován stavovým diagramem na obr.5. Zbývá vyřešit tento markovský proces - jeho řešení pak dává odpověď na všechny otázky týkající se výchozí soustavy hromadné obsluhy.

Obecným řešením našeho markovského procesu je vektor funkcí

$$\pi(t) = (\pi_0(t), \pi_1(t), \dots, \pi_5(t)), \quad t \in \langle 0; \infty \rangle,$$

kde $\pi_i(t)$ udává pravděpodobnost, že systém se v okamžiku $t \in \langle 0; \infty \rangle$ nachází ve stavu $i \in \{0, 1, 2, \dots, 5\}$. Máme-li obecné řešení, známe pravděpodobnost libovolného stavu v libovolném okamžiku. Z významu funkcí $\pi_i(t)$ vyplývá:

- (1) $\pi_i(t) \geq 0$ pro všechna $t \in \langle 0; \infty \rangle$ a všechna $i \in \{0, 1, 2, \dots, 5\}$
- (2) $\pi_0(t) + \pi_1(t) + \dots + \pi_5(t) = 1$ pro všechna $t \in \langle 0; \infty \rangle$... neboť systém je v každém okamžiku v právě jednom stavu

Dále předpokládáme, že v okamžiku $t=0$ je systém ve stavu **0** (neboli (0;0)); odtud plyne

- (3) $\pi(0) = (1, 0, 0, 0, 0, 0)$

Obecné řešení $\pi(t) = (\pi_0(t), \pi_1(t), \dots, \pi_5(t))$ je řešením systému diferenciálních rovnic, které sestavíme na základě grafu markovského procesu z obr.5. Postup ilustrujeme např. pro stav **1**. S využitím elementárních vět o pravděpodobnostech dostáváme

$$\pi_1(t+dt) = \pi_0(t) \cdot \lambda dt + \pi_1(t) \cdot (1 - (\lambda + \mu) dt) + \pi_2(t) \cdot 2\mu dt$$

Odtud

$$(\pi_1(t+dt) - \pi_1(t))/dt = \pi_0(t) \cdot \lambda + \pi_1(t) \cdot (-(\lambda + \mu)) + \pi_2(t) \cdot 2\mu,$$

neboli

$$\pi_1'(t) = \lambda \pi_0(t) - (\lambda + \mu) \pi_1(t) + 2\mu \pi_2(t), \quad (*)$$

kde $\pi_1'(t)$ je prvá derivace funkce $\pi_1(t)$. Stejným způsobem získáme zbývajících 5 diferenciálních rovnic pro stavy **0, 2, 3, 4, 5**. Dostáváme tak systém 6 lineárních diferenciálních rovnic o 6 neznámých funkcích $\pi_0(t), \pi_1(t), \dots, \pi_5(t)$. Lze ukázat, že tento systém je zákonitě vždy lineárně závislý a představuje tedy jen 5 nezávislých rovnic. Chybějící rovnici nahradíme podmínkou (2). Vztah (3) zadává počáteční podmínky pro tento systém.

V naprosté většině praktických případů existuje limita

$$\pi = \lim_{t \rightarrow \infty} \pi(t) = (\lim_{t \rightarrow \infty} \pi_0(t), \lim_{t \rightarrow \infty} \pi_1(t), \dots, \lim_{t \rightarrow \infty} \pi_5(t)) = (\pi_0, \pi_1, \dots, \pi_5)$$

nezávislá na počátečním značení. Tak je tomu např. tehdy, je-li graf markovského procesu silně souvislým grafem. Číselný vektor $\pi = (\pi_0, \pi_1, \dots, \pi_5)$ nazýváme vektorem ustálených pravděpodobností (steady state probabilities). Při startu náhodného procesu, v okamžiku $t=0$, jsou pravděpodobnosti jednotlivých stavů stanoveny počáteční podmínkou $\pi(0) = (1, 0, 0, 0, 0, 0)$, pak se hodnoty pravděpodobností stavů počnou měnit, aby se po určité době ustálili na svých limitních hodnotách $\pi = (\pi_0, \pi_1, \dots, \pi_5)$. Dále se již s časem nemění.

Pokud nás nezajímá chování systému v počátečním přechodném období (toto období je relativně krátké), ale jen jeho chování v ustáleném stavu ($\pi_0, \pi_1, \dots, \pi_5$ oto období je potenciálně nekonečné), pak cílem řešení není nalézt vektor funkcí $\pi(t) = (\pi_0(t), \pi_1(t), \dots, \pi_5(t))$, ale pouze vektor konstant $\pi = (\pi_0, \pi_1, \dots, \pi_5)$. K tomu není zapotřebí řešit systém diferenciálních rovnic, ale postačí řešit mnohem jednodušší systém algebraických rovnic. Postup ukážeme opět na příkladě stavu **1** markovského procesu. Zapsáním limitního přechodu $t \rightarrow \infty$ na obou stranách rovnice (*) dostaneme

$$\lim_{t \rightarrow \infty} \pi_1'(t) = \lambda \lim_{t \rightarrow \infty} \pi_0(t) - (\lambda + \mu) \lim_{t \rightarrow \infty} \pi_1(t) + 2\mu \lim_{t \rightarrow \infty} \pi_2(t)$$

a jeho provedením pak jednouchou algebraickou rovnicí

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

Obdobným postupem nalezneme dalších 5 algebraických rovnic pro stavy **0, 2, 3, 4, 5**. Získáme tak 6 rovnic o 6 neznámých $\pi_0, \pi_1, \dots, \pi_5$ a k tomu máme navíc rovnici $\pi_0 + \pi_1 + \dots + \pi_5 = 1$, kterou nutně potřebujeme, neboť z prvních 6 rovnic je jich jen 5 lineárně nezávislých). Řešením těchto rovnic nalezneme ustálené pravděpodobnosti stavů (jejich vyjádření pomocí vstupních konstant λ, μ).

Podle standardního postupu (viz příloha B k učebním textům) nalezneme ustálené pravděpodobnosti stavů markovského procesu řešením následujícího systému rovnic:

$$\mathbf{Q}^T \cdot \boldsymbol{\pi} = \mathbf{0}, \quad \pi_0 + \pi_1 + \dots + \pi_5 = 1, \quad \text{kde}$$

$\boldsymbol{\pi}$ je neznámý vektor ustálených pravděpodobností chápaný jako sloupcový vektor (jak je v maticových zápisech zvykem),

$\mathbf{Q} = (q_{ij})$ je čtvercová **matice intenzit přechodů**; je-li $i \neq j$, pak q_{ij} značí intenzitu přechodu z i -tého do j -tého stavu a diagonální prvek q_{ii} je roven záporně vzatému součtu všech ostatních prvků v i -tém řádku,

T je symbol transpozice matice,

$\mathbf{0}$ je sloupcový nulový vektor (všechny jeho prvky jsou nulové).

V našem konkrétním příkladě můžeme stanovit matici intenzit přechodů bezprostředně z obr.6. Nejdříve stanovíme všechny nediagonální prvky matice přepsáním intenzit z obr.6. do matice. Diagonální prvky matice stanovíme pak tak, aby součet prvků v každém řádku byl nulový. Matice intenzit přechodu je zapsána v zarámované části následující tabulky tab.1.

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|------------|--------------------|---------------------|---------------------|---------------------|-----------|
| 0 | $-\lambda$ | λ | 0 | 0 | 0 | 0 |
| 1 | μ | $-(\lambda + \mu)$ | λ | 0 | 0 | 0 |
| 2 | 0 | 2μ | $-(\lambda + 2\mu)$ | λ | 0 | 0 |
| 3 | 0 | 0 | 2μ | $-(\lambda + 2\mu)$ | λ | 0 |
| 4 | 0 | 0 | 0 | 2μ | $-(\lambda + 2\mu)$ | λ |
| 5 | 0 | 0 | 0 | 0 | 2μ | -2μ |

Tab.1 Matice intenzit přechodů

Rovnice (**), kterou jsme výše na ukázkou odvodili, odpovídá 2.sloupci matice \mathbf{Q} , resp. 2.řádku matice \mathbf{Q}^T a rovnice (**) sama je 2.rovnicí systému $\mathbf{Q}^T \cdot \boldsymbol{\pi} = \mathbf{0}$.

Ukážeme zde ještě jeden supervýhodný způsob nalezení ustáleného řešení markovského procesu, který nevyžaduje žádná řešení systému rovnic. Východiskem této metody je opět zjednodušený graf markovského procesu. Pro každý stav i procesu určíme tzv. „globální“ intenzitu γ_i , s intuitivním významem míry (intenzity) směřování náhodného procesu do tohoto stavu. Má-li graf markovského procesu jednoduchou lineární topologii, pak výpočet globálních intenzit pomocí lokálních intenzit (intenzit v dosavadním smyslu, tj. intenzit ohodnocujících jednotlivé hrany grafu) je mimořádně snadný. Pro tento případ je globální intenzita směřování do daného stavu definována jako prostý součin intenzit všech přechodů (zobrazených hranami), jejichž provedení přibližuje stav procesu k danému stavu. Výpočet takto definovaných globálních intenzit pro markovský proces z obr. 6 následuje:

$$\begin{aligned} \gamma_0 &= \mu \cdot 2\mu \cdot 2\mu \cdot 2\mu \cdot 2\mu = 2^4 \mu^5 \\ \gamma_1 &= \lambda \cdot 2\mu \cdot 2\mu \cdot 2\mu \cdot 2\mu = 2^4 \lambda \mu^4 \\ \gamma_2 &= \lambda \cdot \lambda \cdot 2\mu \cdot 2\mu \cdot 2\mu = 2^3 \lambda^2 \mu^3 \\ \gamma_3 &= \lambda \cdot \lambda \cdot \lambda \cdot 2\mu \cdot 2\mu = 2^2 \lambda^3 \mu^2 \\ \gamma_4 &= \lambda \cdot \lambda \cdot \lambda \cdot \lambda \cdot 2\mu = 2 \lambda^4 \mu \\ \gamma_5 &= \lambda \cdot \lambda \cdot \lambda \cdot \lambda \cdot \lambda = \lambda^5 \end{aligned}$$

Ustálené pravděpodobnosti jednotlivých stavů π_i dostaneme tak, že vypočtené globální intenzity jednotlivých stavů γ_i vydělíme součtem všech všech globálních intenzit, tj.

$$\pi_i = \gamma_i / (\gamma_0 + \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 + \gamma_5) \quad \text{pro } i=0,1,2,3,4,5$$

Tato metoda je speciálním případem obecnější metody prezentované a dokázané v článku: J.Markl. *A Graph Method for Markov Models Solving*. Acta Mathematica et Informatica Universitatis Ostraviensis 1 (1993).

7. Ukazatelé chování systému

Známe-li ustálené pravděpodobnosti stavů systému hromadné obsluhy (markovského systému, který systém hromadné obsluhy reprezentuje), můžeme počítat rozmanité další pravděpodobnosti a jiné veličiny charakterizující chování systému. Jedná se zejména o následující charakteristiky:

- Pravděpodobnost přijetí požadavku do systému:

$$\pi_0 + \pi_1 + \pi_2 + \pi_3 + \pi_4 = 1 - \pi_5$$
- Pravděpodobnost odmítnutí požadavku systémem:

$$\pi_5$$
- Pravděpodobnost, že přijatý požadavek bude čekat na obsluhu:

$$\pi_2 + \pi_3 + \pi_4$$
- Střední hodnota počtu obsluhovaných požadavků = střední hodnota počtu pracujících aparátů obsluhy:

$$0\pi_0 + 1\pi_1 + 2\pi_2 + 2\pi_3 + 2\pi_4 + 2\pi_5 = \pi_1 + 2(\pi_2 + \pi_3 + \pi_4 + \pi_5)$$
- Střední hodnota počtu přijatých požadavků čekajících na obsluhu:

$$0\pi_0 + 0\pi_1 + 0\pi_2 + 1\pi_3 + 2\pi_4 + 3\pi_5 = \pi_3 + 2\pi_4 + 3\pi_5$$
- Střední hodnota počtu požadavků v systému obsluhy:

$$0\pi_0 + 1\pi_1 + 2\pi_2 + 3\pi_3 + 4\pi_4 + 5\pi_5 = \pi_1 + 2\pi_2 + 3\pi_3 + 4\pi_4 + 5\pi_5$$
- Střední hodnota počtu volných (nepracujících) aparátů obsluhy:

$$2\pi_0 + 1\pi_1 + 0\pi_2 + 0\pi_3 + 0\pi_4 + 0\pi_5 = 2\pi_0 + \pi_1$$
- Střední hodnota počtu obsazených (pracujících) aparátů obsluhy:

$$0\pi_0 + 1\pi_1 + 2\pi_2 + 2\pi_3 + 2\pi_4 + 2\pi_5 = \pi_1 + 2(\pi_2 + \pi_3 + \pi_4 + \pi_5)$$
- Střední hodnota doby strávené požadavkem v systému obsluhy:

$$(0\pi_0 + 1\pi_1 + 2\pi_2 + 3\pi_3 + 4\pi_4 + 5\pi_5) / \lambda = (\pi_1 + 2\pi_2 + 3\pi_3 + 4\pi_4 + 5\pi_5) / \lambda \quad ????$$
-

Množina stavů $\{0,1,2,3,4,5\}$ obslužného systému (viz obr.6) se přirozeným způsobem rozpadá na tři části (superstavy):

- $\{0,1\}$... počet požadavků v systému < počet obslužných aparátů;
vznikají ztráty z prostoje obslužných aparátů
- $\{2\}$... počet požadavků v systému = počet obslužných aparátů;
nevznikají žádné ztráty
- $\{3,4,5\}$... počet požadavků v systému > počet obslužných aparátů;
vznikají ztráty z prostoje (čekání) požadavků

Ve stavu **5** (systém je plný: všechny aparáty obsluhy pracují, fronta dosáhla maximálně možné délky) vznikají navíc ztráty nedovolením vstupu požadavku do obslužného systému (odmítání možných zákazníků). V ostatních stavech tyto ztráty nevznikají.

Spočítáme celkové ztráty obslužného systému za jednotku času. K tomu potřebujeme znát následující ekonomické parametry (jejich konkrétní hodnoty musí dodat ekonomická analýza):

- c_1 ... ztráty spojené s prostojem jednoho obslužného aparátu za jednotku času
- c_2 ... ztráty spojené s čekáním jednoho požadavku ve frontě na obsluhu za jednotku času (např. ekonomická ztráta zákazníka za jednotku času čekání ve frontě)
- c_3 ... ztráty spojené s odmítnutím jednoho požadavku v důsledku plnosti systému

Na základě těchto parametrů můžeme stanovit očekávané (střední) hodnoty ztrát celého obslužného systému:

- C_1 ... střední hodnota ztrát spojených s prostoji obslužných aparátů za jednotku času,
- C_2 ... střední hodnota ztrát spojených s čekáním požadavků na obsluhu za jednotku času,
- C_3 ... střední hodnota ztrát spojených s odmítáním požadavků (zákazníků) za jednotku času,
- C ... střední hodnota celkových ztrát spojených s provozem daného obslužného systému za jednotku času.

Následuje výpočet:

$$C_1 = c_1(\text{stř.hodn. počtu volných apar.}) = c_1(2\pi_0 + 1\pi_1 + 0\pi_2 + 0\pi_3 + 0\pi_4 + 0\pi_5) = c_1(2\pi_0 + \pi_1)$$

$$C_2 = c_2(\text{stř.hodn. počtu čekajících pož.}) = c_2(0\pi_0 + 0\pi_1 + 0\pi_2 + 1\pi_3 + 2\pi_4 + 3\pi_5) = c_2(\pi_3 + 2\pi_4 + 3\pi_5)$$

$$C_3 = c_3(\text{stř.hodn. počtu odmítnutí}) = c_3(0\pi_0 + 0\pi_1 + 0\pi_2 + 0\pi_3 + 0\pi_4 + 1\pi_5) \quad \lambda = c_3\pi_5\lambda$$

$$C = C_1 + C_2 + C_3 = c_1(2\pi_0 + \pi_1) + c_2(\pi_3 + 2\pi_4 + 3\pi_5) + c_3\pi_5\lambda$$

Poslední vzorec udává míru neefektivnosti daného obslužného systému s hodnotami $S=2$ (počet serverů, aparátů obsluhy), $Q=3$ (maximální přípustná délka fronty). U náhodných systémů nelze neefektivnost zcela potlačit (snížit na nulovou hodnotu), lze však docílit její minimální hodnoty optimální volbou parametrů S a Q .

8. Optimalizační úlohy

Systém hromadné obsluhy s popsanou jednoduchou strukturou je jednoznačně popsán čtveřicí parametrů (S, Q, λ, μ) . Představme si nyní, že jsme v úloze navrhovatele obslužného systému, pro něhož parametry λ, μ jsou dané parametry a který má určit parametry S, Q , tak, aby systém pracoval co nejefektivněji. Parametr λ určuje očekávané množství požadavků, které vzniknou během jednotky času a které budou požadovat obsluhu a parametr μ udává výkonnost obslužných aparátů, které máme k dispozici. Tyto parametry jsou vnější a předpokládáme, že navrhovatel je nemůže ovlivňovat. Naproti tomu volba hodnot parametrů S (počet obslužných aparátů v systému) a Q (kapacita fronty ve které čekají požadavky na obsluhu) je plně v rukách projektanta obslužného systému. Má systém navrhnout tak, aby celkové provozní ekonomické ztráty způsobené prostoji obslužných aparátů, čekáním zákazníků (požadavků) ve frontě a odmítáním zákazníků, byly co nejmenší. Aby to mohl učinit, musí znát ještě hodnoty ekonomických parametrů c_1, c_2, c_3 definovaných v předchozí kapitole. Čili matematicky formulováno, jsou dány hodnoty veličin $\lambda, \mu, c_1, c_2, c_3$ a na základě nich mají být hodnoty veličin S a Q tak, aby jisté kritérium efektivnosti bylo optimalizováno (ekonomické kritérium provozních ztrát bylo minimalizováno). Tu kritériální funkci $C = C(S, Q; \lambda, \mu, c_1, c_2, c_3)$ v dalším sestavíme. (Pro úplnost ještě poznamenejme, že v právě formulovaném zadání optimalizační úlohy abstrahujeme od pořizovacích nákladů na výstavbu obslužného systému – tyto náklady samo sebou také závisí na S, Q .)

Markovský model obslužného systému je obdobný modelu z obr.6; liší se jen kvantitativně a to tím, že místo s konkrétními počty $S=2$ (serverů, obslužných aparátů) a $Q=3$ (míst ve frontě), budeme nyní pracovat s obecnými počty S, Q . Celkový počet stavů je pak nikoliv $(2+3)+1=6$, ale $(S+Q)+1$.

Množina stavů $\{0, 1, \dots, S-1, S, S+1, \dots, S+Q-1, S+Q\}$ obecného systému se přirozeným způsobem opět rozpadá na tři části (superstavy):

- $\{0,1,\dots,S-1\}$... počet požadavků v systému < počet obslužných aparátů; vznikají ztráty z prostojů obslužných aparátů
- $\{S\}$... počet požadavků v systému = počet obslužných aparátů; nevznikají žádné ztráty
- $\{S+1,\dots,S+Q-1,S+Q\}$... počet požadavků v systému > počet obslužných aparátů; vznikají ztráty z prostojů (čekání) požadavků

Ustálené řešení markovského modelu obecného systému s nekonkretizovanými parametry S, Q budeme řešit opět grafovou metodou pomocí globálních intenzit. Globální intenzity přechodů γ_i ($i=0,1,\dots,S+Q$) se přímo odečítají z grafu, který je přímočarým zobecněním grafu z obr.6. Ten, kdo si zobecněný graf nedokáže ihned představit, může si jej (i s nezbytnými s vytečkovanými částmi) snadno nakreslit. Dále následují výsledky pro pro všech $S+Q+1$ globálních globálních intenzit. Každý výsledek je prezentován ve třech formách. Prvá forma odpovídá bezprostřednímu odečtu z grafu. Kvůli větší čitelnosti jsou skupiny činitelů odděleny podtržítka tak, aby rozdělení činitelů (lokálních intenzit) do skupin bylo v souladu s rozdělením stavů do superstavů. Druhá forma vyjadřuje výsledky v kompaktním úsporném tvaru – bez nutnosti vytečkování chybějících činitelů. Konečně třetí forma naznačuje jakým způsobem by bylo možné vyjádřit jedinou formulí všechny globální intenzity.

$$\begin{aligned}
 \gamma_0 &= \mu.2\mu.3\mu\dots(S-1)\mu_.S\mu_.S\mu\dots S\mu &= (S-1)!.S^Q.\mu^{S+Q} &= ((S-1)!/0!).S^Q.\mu^{S+Q}.\lambda^0 \\
 \gamma_1 &= \lambda.2\mu.3\mu\dots(S-1)\mu_.S\mu_.S\mu\dots S\mu &= (S-1)!.S^Q.\mu^{S+Q-1}.\lambda &= ((S-1)!/1!).S^Q.\mu^{S+Q-1}.\lambda^1 \\
 \gamma_2 &= \lambda.\lambda.3\mu\dots(S-1)\mu_.S\mu_.S\mu\dots S\mu &= ((S-1)!/2!).S^Q.\mu^{S+Q-2}.\lambda^2 &= ((S-1)!/2!).S^Q.\mu^{S+Q-2}.\lambda^2 \\
 &\dots & & \\
 \gamma_{S-1} &= \lambda.\lambda.\lambda\dots\lambda.(S-1)\mu_.S\mu_.S\mu\dots S\mu &= (S-1).S^Q.\mu^{Q+1}.\lambda^{S-1} &= ((S-1)!/(S-2)!).S^Q.\mu^{Q+1}.\lambda^{S-1} \\
 &\text{-----} & & \text{-----} \\
 \gamma_S &= \lambda.\lambda.\lambda\dots\lambda.\lambda_.S\mu.S\mu\dots S\mu &= (S.\mu)^Q.\lambda^S &= S^Q.\mu^Q.\lambda^S \\
 &\text{-----} & & \text{-----} \\
 \gamma_{S+1} &= \lambda.\lambda.\lambda\dots\lambda.\lambda_.\lambda_.S\mu.S\mu\dots S\mu &= S^{Q-1}.\mu^{Q-1}.\lambda^{S+1} &= S^{Q-1}.\mu^{Q-1}.\lambda^{S+1} \\
 \gamma_{S+2} &= \lambda.\lambda.\lambda\dots\lambda.\lambda_.\lambda_.\lambda_.S\mu\dots S\mu &= S^{Q-2}.\mu^{Q-2}.\lambda^{S+2} &= S^{Q-2}.\mu^{Q-2}.\lambda^{S+2} \\
 &\dots & & \dots \\
 \gamma_{S+Q-1} &= \lambda.\lambda.\lambda\dots\lambda.\lambda_.\lambda_.\lambda.\lambda\dots\lambda.S\mu &= S.\mu.\lambda^{S+Q-1} &= S^1.\mu^1.\lambda^{S+Q-1} \\
 \gamma_{S+Q} &= \lambda.\lambda.\lambda\dots\lambda.\lambda_.\lambda_.\lambda.\lambda\dots\lambda.\lambda &= \lambda^{S+Q} &= S^0.\mu^0.\lambda^{S+Q}
 \end{aligned}$$

Ustálené pravděpodobnosti jednotlivých stavů jsou dány formulemi:

$$\pi_i = \gamma_i / (\gamma_0 + \gamma_1 + \dots + \gamma_{S-1} + \gamma_S + \gamma_{S+1} + \dots + \gamma_{S+Q-1} + \gamma_{S+Q}) \quad \text{pro } i=0,1,2,\dots,S+Q$$

Známe-li ustálené pravděpodobnosti, můžeme počítat střední hodnoty ztrát tří různých typů za jednotku času

$$\begin{aligned}
 C_1 &= c_1(\text{stř.hodn. počtu volných apar.}) = c_1(S\pi_0 + (S-1)\pi_1 + \dots + \pi_{S-1}) \\
 C_2 &= c_2(\text{stř.hodn. počtu čekajících pož.}) = c_2(\pi_{S+1} + 2\pi_{S+2} + \dots + Q\pi_{S+Q}) \\
 C_3 &= c_3(\text{stř.hodn. počtu odmítnutí}) = c_3\pi_{S+Q} \lambda
 \end{aligned}$$

a také střední hodnotu celkových ztrát

$$C = C_1 + C_2 + C_3 = c_1(S\pi_0 + (S-1)\pi_1 + \dots + \pi_{S-1}) + c_2(\pi_{S+1} + 2\pi_{S+2} + \dots + Q\pi_{S+Q}) + c_3\pi_{S+Q} \lambda$$

představující hledanou kritériální (účelovou) funkci. Tato funkce není tak jednoduchá, jak na první pohled vypadá. Optimalizační proměnné S, Q jsou skryty i ve výrazech pro globální intenzity γ_i a tedy i pro ustálené pravděpodobnosti π_i . Kdybychom v kritériální funkci

rozvedli ustálené pravděpodobnosti pomocí globálních intenzit a globální intenzity rozvedli dále pomocí lokálních intenzit λ , μ a proměnných S , Q , získali bychom velice složitý výraz. Pro hledání extrému funkce $C(S, Q)$ nelze použít klasických metod matematické analýzy nejen pro mimořádnou složitost funkce, ale také proto, že S , Q , nejsou reálné veličiny, ale veličiny nabývající kladných celých hodnot. Nejjednodušší metoda nalezení dvojice (S, Q) minimalizující funkci $C(S, Q)$ je metoda tabelizace této funkce. Tato metoda je vhodná také proto, že množiny hodnot argumentů S a Q , které přicházejí v úvahu pro extrém nebývají příliš rozsáhlé.

Všimněme si ještě, že volíme-li speciálně $S=2$ a $Q=3$, dostáváme z obecných vzorců uvedených v této kapitole výsledky vypočtené v kapitole předchozí.

9. Zobecnění

Systém hromadné obsluhy popsany v 1.kapitole (a následně také modely tohoto systému ve tvaru P/T PN, SPN, GSPN a ekvivalentních markovských systémů), může být v mnoha směrech modifikován a zobecněn:

- Požadavky na obsluhu mohou být vícero typů, které se navzájem liší:
 - intenzitami příchodu
 - intenzitami obsluhy
 - prioritou obsluhy
- Obslužné aparáty mohou být různých typů navzájem se lišících:
 - výkonností, tj. intenzitami obsluhy
 - specializací, tj. použitelností pouze na vybrané typy požadavků
- Z fronty čekajících požadavků mohou odcházet „netrpělivé“ požadavky; proud odcházejících zákazníků je poissonovský s danou intenzitou odchodu
- Fronta může být strukturována do 2 nebo více front podle prioritního stupně obsluhovaných požadavků (nejdříve se obsluhují požadavky s vyšší prioritou obsluhy).

Appendix D:

Algoritmy analýzy P/T Petriho sítí

1. Konstrukce grafu dosažitelnosti

Vstup: PN systém $\langle P, T, I, O, H, M_0 \rangle$

Výstup: Orientovaný graf $\langle V, E \rangle$ představující graf dosažitelnosti PN systému v případě, že PN systém je omezený. V opačném případě indikace neomezenosti PN systému.

Algoritmus:

1. Inicializace: $\langle V, E \rangle = \langle \{ M_0 \}, \emptyset \rangle$, M_0 není označeno.
2. Dokud v množině uzlů V existují neoznačené uzly vykonávej následující operace:
 - 2.1. Vyber neoznačený uzel $M \in V$ a označ jej.
 - 2.2. Pro každý přechod t , který je proveditelný při značení M , prováděj:
 - 2.2.1. Vypočítej značení M' takové, že $M \xrightarrow{t} M'$.
 - 2.2.2. Existuje-li značení $M'' \in V$ takové, že $M'' \xrightarrow{\sigma} M'$ pro nějakou sekvenci σ a přitom M'' je pokrýváno M' , pak algoritmus končí práci s tím, že PN systém je neomezený a tedy neexistuje konečný graf dosažitelnosti.
 - 2.2.3. Neexistuje-li žádné značení $M'' \in V$ takové, že $M'' = M'$ (tj. značení M' se dosud v množině V nevyskytuje), pak $V := V \cup \{ M' \}$, přičemž M' je neoznačený prvek množiny V .
 - 2.2.4. $E := E \cup \{ (M, t, M') \}$
3. Algoritmus končí práci s tím, že PN systém je omezený a jeho graf dosažitelnosti je $RG(M_0) = \langle V, E \rangle$.

2. Vlastnosti bezpečnosti (marking invariances)

Vstup: Graf dosažitelnosti $RG(M_0)$. Vlastnost bezpečnosti II .

Výstup: Odpověď *ano* nebo *ne* podle toho, zda PN systém má nebo nemá vlastnost II .

Algoritmus:

1. Inicializace: všechny prvky množiny dosažitelných stavů $RS(M_0)$ považuj za neoznačené.
2. Dokud v množině $RS(M_0)$ existují neoznačené stavy (značení) prováděj následující:
 - 2.1. Vyber neoznačený stav $M \in RS(M_0)$ a označ ho.
 - 2.2. Jestliže stav M nemá nemá vlastnost II , pak odpověď je *ne* (PN systém nemá vlastnost II).
3. Odpověď je *ano*, PN systém má vlastnost II .

3. Vlastnosti živosti (liveness invariances)

Vstup: Graf dosažitelnosti $RG(M_0)$. Vlastnost živosti II .

Výstup: Odpověď *ano* nebo *ne* podle toho, zda PN systém má nebo nemá vlastnost II .

Algoritmus:

1. Dekomponuj graf dosažitelnosti $RG(M_0)$ na silně souvislé komponenty: C_1, C_2, \dots, C_r .
2. Vytvoř graf silně souvislých komponent RG^c , tj. graf (V_c, E_c) , kde $V_c = \{C_1, C_2, \dots, C_r\}$ a $(C_i, t, C_j) \in E_c$ právě tehdy, když existuje hrana $(M, t, M') \in E$, přičemž M je v C_i , a M' je v C_j , $i \neq j$.
3. Urči množinu F všech terminálních silně souvislých komponent grafu RG^c .
4. Pokud existuje aspoň jedna komponenta C_i v F :
 - 4.1. Jestliže žádné značení M z C_i nemá vlastnost II , pak odpověď zní *ne*, tj. PN systém nemá vlastnost II .
 - 4.2. Odstraň C_i z F .
5. Odpověď zní *ano*, tj. PN systém má vlastnost II .

4. Výpočet minimálních p-invariantů (minimal p-semiflows) PN struktury

Vstup: Incidenční matice C typu $(m, n) = (|P|, |T|)$.

Výstup: Množina všech minimálních p-invariantů PN struktury definované maticí C .

Algoritmus:

1. Inicializace: $A = C$, $B = I_m$, (I_m je jednotková matice řádu $m = |P|$).
2. Pro $j=1, 2, \dots, n$ ($n=|T|$) prováděj:
 - 2.1. Přidej ke složené matici $[A|B]$ všechny řádky, které jsou lineárními kombinacemi s přirozenými koeficienty dvojic řádků z matice $[A|B]$ a které současně anulují j -tý sloupec matice A .
 - 2.2. Vyškrtej z matice $[A|B]$ všechny řádky s nenulovým prvkem v j -tém sloupci matice A .
3. Všechny řádky (p-invarianty) matice B převed' do kanonického tvaru (tj. vyděl každý řádek největším společným dělitelem všech jeho prvků).
4. Odstraň z matice B všechny neminimální p-invarianty (tj. vyškrtej z matice B všechny ty řádky, které ostře pokrývají nějaký jiný řádek matice B).
5. Řádky matice B představují úplný soubor minimálních p-invariantů PN struktury zadané incidenční maticí C .

5. Výpočet minimálních t-invariantů (minimal t-semiflows) PN struktury

Lze vypočítat jako p-invarianty duální PN struktury, tj. místo s incidenční maticí C typu $(m, n) = (|P|, |T|)$ počítáme s transponovanou maticí C^T typu $(n, m) = (|T|, |P|)$.