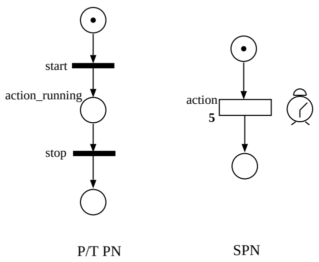
Stochastické Petriho sítě

poznámky do cvičení z MSI

Tomáš Vojnar

1 Úvod

- Navazujeme na cvičení týkající se modelování pomocí P/T a barvených Petriho sítí, tentokrát se však zaměříme na Petriho sítě s explicitním vyjádřením času.
- Časování spojíme s přechody modelujícími akce, neboli změny stavu systému. Ty pak již nebudou atomické v čase, ale od počátku proveditelnosti do vlastního provedení uplyne určitá doba. Tato doba bude odvozována z pravděpodobnostních rozložení, a proto hovoříme o **stochastických Petriho sítích** (SPN).
- Srovnání popisu neatomických akcí v P/T sítích a SPN je uvedeno na obrázku 1, přičemž je zapotřebí si uvědomit, že popis pomocí SPN je přesnější (v P/T sítích se neříká nic o *délce* trvání akcí).



Obrázek 1: Srovnání modelování pomocí klasických a stochastických Petriho sítí

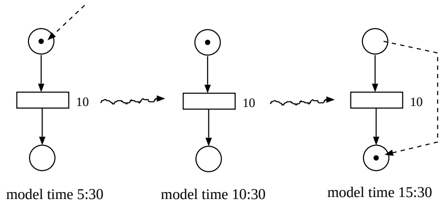
2 Základní pojmy černobílých SPN

- V SPN budou časované i nečasované (okamžité) přechody.
- Časované přechody budou spojeny s pravděpodobnostním rozložením popisujícím dobu provádění a budou se provádět takto:

  1. Jakmile se přechod stane proveditelným, vygeneruje se hodnota z příslušného rozložení.
  2. Čeká se po dobu odpovídající vygenerované hodnotě.

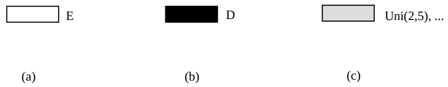
  3. Jakmile příslušná doba čekání uběhne a přechod je po celou tuto dobu bez přerušení proveditelný, provede se – tj. hodnota spojená s přechodem udává, **jak dlouho musí být systém v určitém stavu, než může dojít k jeho změně určitým způsobem**.

Popsaný mechanismus provádění je demonstrován na obrázku 2.



Obrázek 2: Provedení časovaného přechodu (předpokládáme, že byla náhodně vygenerována právě průměrná doba provádění, tj. v našem případě 10)

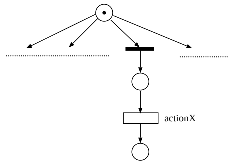
Poznámka: Pro znázornění časovaných přechodů zavedeme konvenci dle obrázku 3.



Obrázek 3: Reprezentace časovaných přechodů: (a) exponenciální se střední hodnotou E, (b) deterministický se zpožděním D, (c) jiný typ rozložení (např. rovnoměrné) – nutno blíže specifikovat (např. Uni(2,5))

- Co se ovšem stane, dojde-li v průběhu čekání na provedení přechodu k přerušení proveditelnosti? To závisí na specifikované **paměťové politice**:
  - **enabling memory policy** – Po přerušení a znovuobnovení proveditelnosti, se generuje nová doba zpoždění, tj. jedná se o restart časovače.
  - **age memory policy** – Odpočet se zastaví a po obnovení proveditelnosti běží dále k nule od toho bodu, kde došlo k přerušení.

  Poznámka: Enabling memory policy budeme pokládat za default, age memory policy budeme specifikovat explicitně.
- U časovaných přechodů je rovněž možné specifikovat tzv. **kapacitu serveru**. Ta udává kolik požadavků může být současně obsluhováno, neboli kolik odpočtů může běžet současně. Kapacita se pohybuje od 1 (default), přes ostatní přirozená čísla (nutno specifikovat) až k nekonečnu (rovněž nutno specifikovat explicitně).
- Chceme-li modelovat situaci, kdy se nějaká operace zahájí (tj. odebere se vstupní značení), pak běží bez možnosti přerušení a skončí po určité době, lze to modelovat tak, jak je ukázáno na obrázku 4.

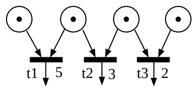


Obrázek 4: Realizace nepřerušitelné akce

- SPN budeme uvažovat s prioritami, časované přechody budou vždy na úrovni 0, ostatní na úrovni alespoň 1. Časovaný přechod tedy může být proveden pouze tehdy, není-li proveditelný žádný okamžitý přechod. To vede na dva druhy značení: **viditelné** – žádný okamžitý přechod není proveditelný a **dočasné** – existují po nulový čas, neboť se provede nějaký okamžitý přechod.
- U okamžitých přechodů budeme specifikovat tzv. **váhy**, které nám umožní říci, s jakou pravděpodobností se systém bude vyvíjet jednou ze vzájemně se vylučujících cest.

  Příklad: Na obrázku 5 specifikujeme, že je možné pokračovat třemi cestami, jejichž pravděpodobnosti jsou po řadě 0.5, 0.3 a 0.2. Tyto pravděpodobnosti vyplývají ze vztahů:

$$
\begin{aligned}
p(t_1) &= \tfrac{5}{5+3+2} = 0.5 \\
p(t_2) &= \tfrac{3}{5+3+2} = 0.3 \\
p(t_3) &= \tfrac{2}{5+3+2} = 0.2
\end{aligned}
$$



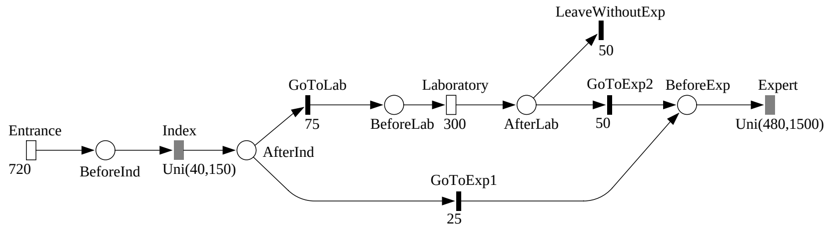
Obrázek 5: Váhy okamžitých přechodů

Poznámka: Všimněte si použití tranzitivity konfliktnosti přechodů při výpočtu pravděpodobnosti možných cest v předchozím příkladě.

3 Příklad na použití SPN

- Budeme modelovat jednoduché zdravotní středisko:
  - Příchod pacienta každých 12 minut s exp. rozložením.
  - Všichni pacienti projdou kartotékou, což jim zabere rovnoměrně 40 až 150 sekund.
  - Poté 25% pacientů jde přímo k lékaři.
  - Zbývajících 75% pacientů jde nejdříve na odběr do laboratoře. To jim zabere v průměru 6 minut, načež polovina lidí odchází a druhá polovina pokračuje k lékaři.

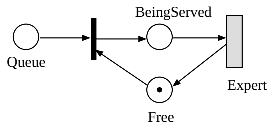
  - U lékaře pacienti stráví rovnoměrně 8 až 25 minut.
- Řešení je na obrázku 6.



Obrázek 6: Model zdravotního střediska

4 Získávání výsledků

- Nástroje pro práci s SPN umožní simulací nebo analýzou získání pravděpodobností jednotlivých viditelných stavů (tj. značení) v ustáleném stavu. Z nich lze pak odvodit další ukazatele jako jsou:
  - pravděpodobnost platnosti nějaké podnímky (např. zařízení je volné),
  - průměrný počet požadavků v (sub)systému a délku front,
  - výkon zařízení – počet provedení nějaké akce za časovou jednotku
  - a další.
- Komplikovanější je to se získáváním informací o přechodném ději. Tímto se zde blíže zabývat nebudeme, pouze poznamenáme, že vždy je možné alespoň sledovat časy a pořadí provádění přechodů při simulaci.
- Příklad: Uvažme náš model lékařského střediska, pak můžeme získat následující informace:
  - pravděpodobnost, že někdo je vyšetřován doktorem, neboli využití doktora: $\sum_{M:M(p_{BeforeExp})>0} p(M)$
  - délku fronty u doktora lze určit ze vztahu $\sum_{M:M(p_{BeforeExp})>1} (M(p_{BeforeExp})-1)p(M)$ nebo je možné upravit SPN dle obrázku 7 a použít vztah $\sum_M M(p_{Queue})p(M)$

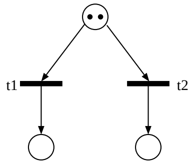


Obrázek 7: Úprava SPN pro snadné určení délky fronty

  - počet pacientů ošetřených doktorem za jednotku času: $\sum_{M:M(p_{BeforeExp})>0} (\frac{480+1500}{2})^{-1} p(M)$

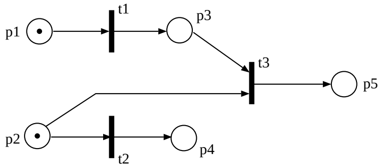
5 Některé vlastnosti PN významné pro SPN

- **Konflikt** (efektivní): provedením jednoho přechodu znemožníme provedení jiného přechodu, nebo snížíme jeho stupeň proveditelnosti – viz obrázek 8. Konflikty se v SPN ošetřují pomocí vah přechodů, jak jsme již ostatně viděli.



Obrázek 8: Dva přechody v efektivním konfliktu

- **Zmatení** (confusion): při provedení těchže přechodů v různém pořadí se řeší různý počet konfliktů, což je v SPN nežádoucí, chceme-li specifikovat pravděpodobnosti jednotlivých cest. V síti na obrázku 9 například nevíme ve stavu daném značkami v místech p_3 a p_4, zda se řešil konflikt t_2, t_3 nebo ne.



Obrázek 9: Příklad zmatení

- Ovlivnění prioritou: Některé konflikty se rozřeší, na druhou stranu některé mohou vzniknout. Jedná se o tzv. **nepřímé konflikty**. Nepřímý konflikt t_1 a t_2 vznikne, když po provedení t_1 následuje sekvence vysoce prioritních přechodů, která končí t_3, který je v konfliktu s t_2.

- **Živost** (ve smyslu Petriho sítí): V libovolném značení je možné provést sekvenci přechodů končících libovolným zvoleným přechodem. Jestliže systém není živý, pak buďto nastane deadlock, některé přechody (a také stavy, v nichž jsou proveditelné) se projeví pouze před ustavením ustáleného stavu a v tomto se již neprojeví, a nebo je několik možných disjuktních ustálených stavů, které se musí řešit odděleně.

- **Reprodukovatelné počáteční značení**: Z libovolného stavu je možné se vrátit do stavu počátečního. Jestliže počáteční značení není reprodukovatelné, pak opět existují značení, které se v ustáleném stavu neprojeví.

- **Tyto vlastnosti jsou detailně zkoumány nástroji pracujícími na bázi SPN.**

6 Stochastické časování a CPN

- Stochastické časování je možné do CPN zavést v podstatě stejně jako do P/T sítí, tj. pomocí časovaných přechodů vyžadujících, aby jejich („barevné“) vstupní značení bylo ve vstupních místech alespoň po určitou dobu.

- Nicméně jistá specifika časování ve spojitosti s CPN přece jen má:

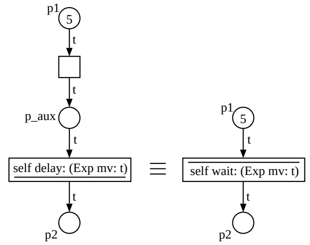
  – Zpoždění je možné odvozovat od barev značek.

  – V případě přechodů s „age memory policy“ se zapamatované časy vztahují ke konkrétním navázáním.

- Pro konkrétní případ CPN v PNtalku je možné časování ze syntaktického hlediska zavést tak, že umožníme ve strážích přechodů používat výrazy typu `self delay: expr` se speciální sémantikou odpovídající časování. Vlastní zpoždění zadáváme jako parametr `delay:` a může jím být: konstanta, vstupní proměnná přechodu, či náhodné rozložení specifikované příslušnou třídou a parametry předanými v konstruktoru (např. `Exp mv: 120` nebo `Uni l: 10 h: 100`). Další speciální zprávy by pak mohly být zavedeny pro specifikaci kapacity serveru a paměťové politiky.

- Výše uvedený mechanismus časování však v PNtalku bohužel nebyl doposud implementován. Místo něj je možné používat zjednodušené časování (bez možnosti přerušení běžících akcí), které je založeno na použití výrazů tvaru `self wait: expr` v akcích přechodů. Vlastní zpoždění se pak specifikuje parametrem `wait:` podobně jako u `delay:`. Sémantika `wait:` je taková, že časovaný přechod odebere vstupní značení a terpve po uplynutí příslušného zpoždění vyprodukuje výstupní.

- Syntax a sémantika `delay:` a `wait:` je demonstrována a srovnána na obrázku 10, který obsahuje dvě (z hlediska časování) ekvivalentní sítě. (Princip je podobný tomu na obrázku 4.)



Obrázek 10: Časování v CPN v PNtalku