

1. Úvod

Tato dokumentace popisuje koncepci a funkčnost simulátoru P/T Petriho sítí napsané v programovacím jazyce C++. Simulátor umí simulovat přechody založené na pravděpodobnost, prioritě a exponenciálním nebo konstantním časování. Výsledkem simulace jsou statistické informace informující o uskutečněných transakcích v zadané P/T Petriho síti a informace o samotné síti v počátečním a konečném stavu. Běh simulátoru končí ve chvíli kdy není možné provést další transakci nebo vyprší čas simulace.

1.1. Autor

Simulátor vypracoval jediný autor v rámci projektu do bakalářského předmětu IMS. Projekt je na téma „Simulátor P/T Petriho sítí“. Autor projektu:

- Patrik Čada – xcadap00

Při tvorbě simulátoru bylo čerpáno z těchto zdrojů:

- IMS.pdf – Hlavní zdroj teoretických informací.
- ims-demo1.pdf – Zdroj testovacích modelů.

1.2. Testované prostředí

Simulátor byl testován (překlad a funkčnost) v těchto prostředích:

- Linux x86_64 3.16.7-29-desktop – OpenSUSE 13.2 překladač g++
- Linux x86_64 3.12.49 - CentOS release 6.7 překladač g++

2. Rozbor tématu a použitých metod

P/T Petriho síť (IMS opora, strana 123) je matematická reprezentace diskretních distribuovaných systémů. Petriho síť byla vytvořena v rámci disertační práce Carlem Adamem Petrim roku 1962 v rámci jeho disertační práce. Petriho síť je nedeterministická (IMS opora, strana 32).

2.1. Matematická definice (IMS opora, strana 123)

Petriho síť je definována jako šestice $\Sigma = (P, T, F, W, C, M_0)$

Kde:

- P – Množina míst (stavů)
- T – Množina přechodů, $P \cap T = \emptyset$
- Incidenční relace $F \subseteq (P \times T) \cup (T \times P)$
- Váhová funkce $W : F \rightarrow \{1, 2, \dots\}$
- Kapacity míst $C : P \rightarrow \mathbb{N}$
- Počáteční značení $M_0 : P \rightarrow \mathbb{N}$ (M se nazývá značení Petriho sítě)

2.2. Reprezentace P/T Petriho sítě

Petriho síť bývá reprezentována graficky a to jako bipartitní graf s ohodnocením. Takto reprezentovaná síť má dva druhy uzlů které označují místa a přechody. Místa a přechody navzájem spojují orientované hrany

2.3. Grafická reprezentace prvků sítě

- **Místa**

Místo v Petriho síti představuje stav systému. Stav systému je definován pomocí značek, ty vyjadřují a aktuální stav místa. Značky mohou být definovány počátečním stavem a jejich počet v místě může být omezen kapacitou místa.



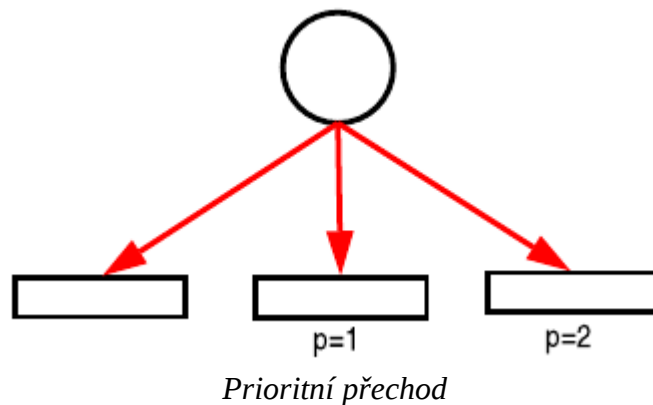
Místo s dvěmi značkami

- **Přechody**

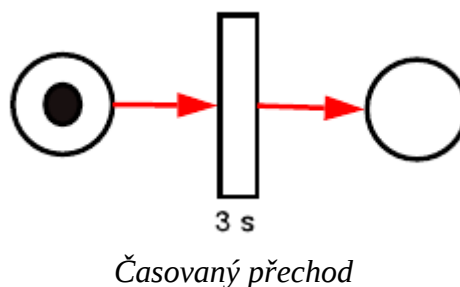
Přechody v P/T Petriho sítích definují možné události. Spojují vstupní a výstupní místo. Přechod je proveden pouze v případě kdy jsou splněny všechny vstupní a výstupní podmínky. Přechod je proveden okamžitě tedy v nulový čas. Pokud je přechod časovaný k přechodu dojde až v době uplynutí času do té doby je značka ve výchozím místě a může přejít jiným přechodem jehož podmínky byly splněny.

Námi použitá Petriho síť je rozšířena o speciální typy přechodů:

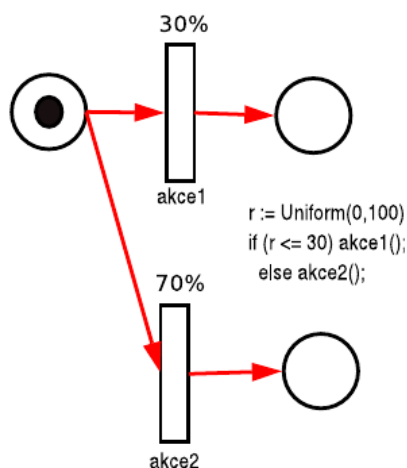
Prioritní přechody (IMS opora, strana 128) – Pokud existují dva možné přechody z aktuálního místa (vstupní i výchozí podmínky jsou splněny), je proveden prioritní přechod. V základu mají všechny přechody prioritu 1.



Časované přechody (IMS opora, strana 132) – Přechod je proveden až po uplynutí časového intervalu, ten může být zadán konstantně, rozsahu nebo některého rozložení v případě tohoto simulátoru exponenciálním. Do doby než je přechod proveden může být značka odebrána jiným přechodem jehož podmínky byly splněny.



Pravděpodobnostní přechody(IMS opora, strana 130) – Přechod je proveden na základě pravděpodobnosti. V rámci validity modelu musí existovat ještě další přechody z vstupního místa tak aby součet jejich pravděpodobnosti byl roven 100%, tedy v programovém vyjádření 1.0.



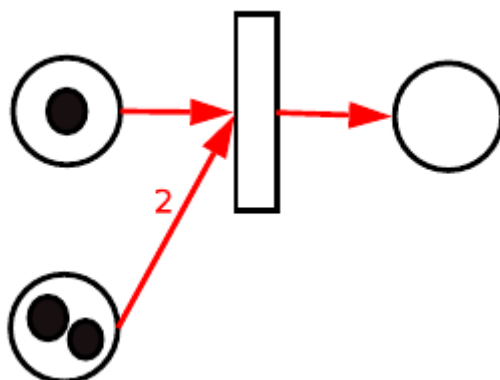
Pravděpodobnostní přechod

Pravidla pro používání rozšířených přechodů (IMS opora, strana 131):

Rozšířená pravidla nelze kombinovat protože by mohlo docházet k jejich nejednoznačnosti tedy by neměl vzniknout přechod s pravděpodobností 30% a časem exp(5), takový simulační model by byl nevalidní.

- **Hrany**

Hrany spojují místa a přechody, jsou orientované a mohou mít kapacitu která říká kolik značek je z místa odebráno a do dalšího místa přidáno, kapacita na vstupní a výstupní hraně může být rozdílná.



Hrana s kapacitou 2

3. Koncepce

Simulátor je navržen tak a aby kladl důraz na co nejpřehlednější zápis simulačního modelu a umožňoval uživateli definovat popisky k jednotlivým Místům/Přechodům/Hranám aby získal rychlejší přehled ve výsledných statistických informacích. Simulátor se skládá z dvou hlavních částí. Simulačního jádra třídy *Simulator* které slouží jako uložisko modelu. Stará se o vytváření Míst/Hran/přechodů a automaticky dealokuje paměť v momentě kdy je třída simulátoru dealokována. Hlavní funkcí této třídy je však řízení a provádění simulace, koncept simulace je provést veškeré možné přechody, časované přechody vložit do kalendáře. V momentě kdy nelze provést žádné další přechody je spuštěna plánovaná událost v kalendáři čímž se změní aktuální čas simulace na čas nejbližší události. Po provedení události se simulátor opět pokusí provést další přechody a celý proces se opakuje dokud je možné provést další přechody (včetně plánovaných) nebo vyprší čas simulace. Druhou hlavní částí je kalendář událostí třídy *Calendar* ten se stará o výše zmíněné plánování časovaných přechodů. Simulátor dále obsahuje třídy reprezentující jednotlivé entity P/T Petriho sítě. Třída *Place* představuje místo, třída *Transition* přechod a třída *Edge* reprezentuje hranu. O zaznamenávání statistik a jejich výpis se stará třída *StatisticPrinter*. Výsledkem simulátoru jsou statistiky popsané níže.

4. Architektura simulačního simulátoru

Simulátor byl navržen tak aby se dal jednoduše používat a uživatel tedy potřebuje znát povědomí pouze o třech třídách a jejich rodiči. Třídy nelze tvořit jinak než přes metody třídy simulátoru, třídy nelze po jejich vytvoření měnit s výjimkou na nastavení popisu.

4.1. Mapování abstraktního modelu na Třídy simulátoru

- Třída *Identificator* – Slouží jako rodič tříd entit. Metody:
 - unsigned int **getId()** - Předá interní ID objektu.
 - std::string &**getDescription()** - Předá uživatelem nastavený popis objektu, v základu vrací prázdný řetězec.
 - void **setDescription**(const std::string &description) - Nastavuje uživatelem definovaný popis objektu.
- Třída *Place* - Dědí z třídy *Identificator*, reprezentuje místo na modelu. Metody:
 - TCapacity **getCapacity()** - Služí k informaci o aktuálním množství značek.
 - TCapacity **getMaxCapacity()** - Slouží k získání informace o maximálním možném množství značek v místě.
- Třída *Transition* – Dědí i třídy *Identificator*, reprezentuje přechod na modelu. Metody:
 - TPriority **getPriority()** - Slouží k získání priority přechodu.
 - double **getChance()** - Slouží k získání pravděpodobnosti přechodu.
 - TTimeTypes **getTimeType()** - Slouží k získání jaký typ časování se použije, v základu je nastaven nečasovaný.
 - TTime **getTime()** - Slouží k získání nastaveného času pro časové přechody.
 - Transition::Type **getType()** - Slouží k získání informace o jaký typ přechodu se jedná.

- Třída *Edge* – Dědí z třídy *Identifier*, reprezentuje hrany v modelu. Metody:
 - *TCapacity* **getCapacity()** - Slouží k získání kapacity hrany.
 - *Edge::Direct* **getDirect()** - Slouží k získání směru hrany.
 - *Place* ***getPlace()** - Slouží k získání místa patřícího k hraně.
 - *Transition* ***getTransition()** - Slouží k získání přechodu patřícího k hraně.

4.2. Použití simulátoru

Při použití simulátoru se využívá výhradně třídy *Simulator* a jeho metod. Metody:

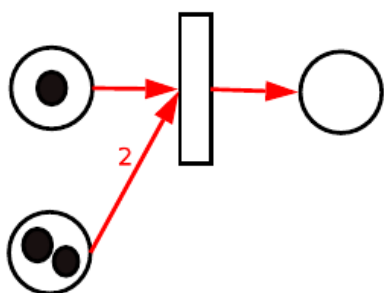
- *Place* ***createPlace**(*TCapacity* initCapacity = 0, *TCapacity* maxCapacity = 0) - Vytvoření místa s počátečním množstvím značek a maximální kapacitou, maximální kapacita 0 znamená bez omezení.
- *Transition* ***createTransition**(double chance) - Vytvoření přechodu závislého na šanci.
- *Transition* ***createTransition**(*TTimeTypes* timeType, *TTime* time) - Vytvoření časovaného přechodu.
- *Transition* ***createTransition**(*TPriority* priority = 0) - Vytvoření přechodu závislého na prioritě.
- *Edge* ***createEdge**(*Place* *place, *Transition* *transition, *TCapacity* capacity = 1) - Vytvoření hrany z místa do přechodu s nastavenou kapacitou.
- *Edge* ***createEdge**(*Transition* *transition, *Place* *place, *TCapacity* capacity = 1) - Vytvoření hrany z přechodu do místa s nastavenou kapacitou.
- void **run**(*TTime* maxTime = DEFAULT_MAX_SIMULATION_TIME) - Spuštění simulace.
- void **printStatistics**(unsigned int types = *StatisticPrinter::All*) - Vypsání statistik, vstupní parametr *types* je bitovým polem složeným z hodnot *StatisticPrinter::TinfoTypes*.

5. Podstata simulačních experimentů a jejich průběh

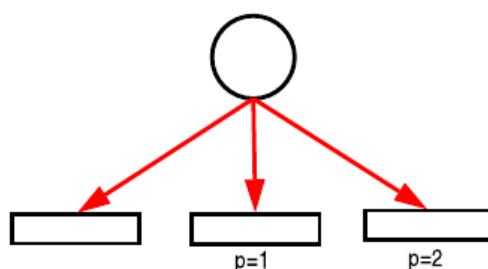
Simulátor je založen na principu plánování časových událostí do kalendáře. Nečasované přechody jsou prováděny okamžitě, v momentě kdy nelze provést další přechody. Je spuštěna událost z kalendáře které má nejnižší čas, vykonané události jsou z kalendáře mazány.

5.1. Postup experimentování

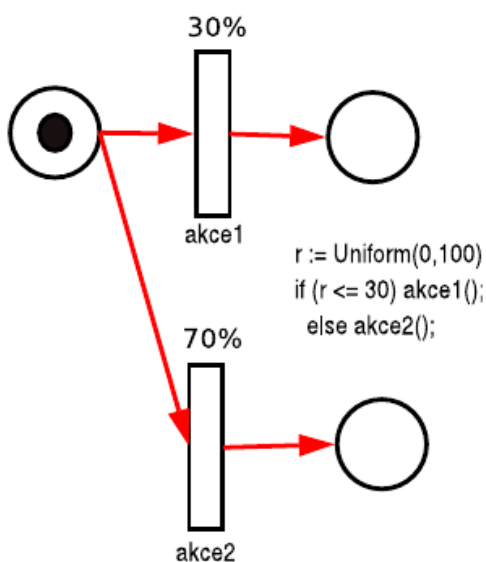
Při vývoji a testování simulátoru byly nejdříve použity jednodušší modely dokazující elementární funkčnost simulátoru.



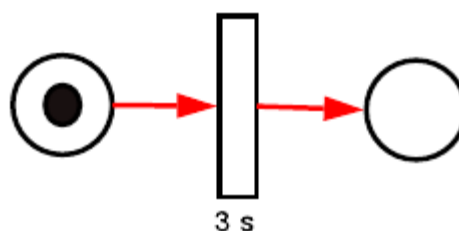
Testovací model reprezentující funkčnost přechodu a kapacity hrany



Testovací model reprezentující funkčnost priority



Testovací model reprezentující funkčnost pravděpodobnosti



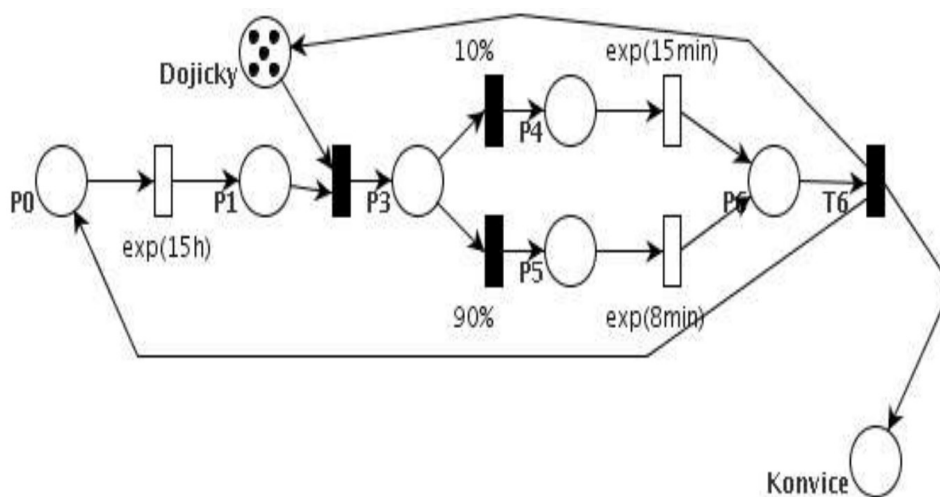
Testovací model časovaného přechodu

5.2. Simulovaný model

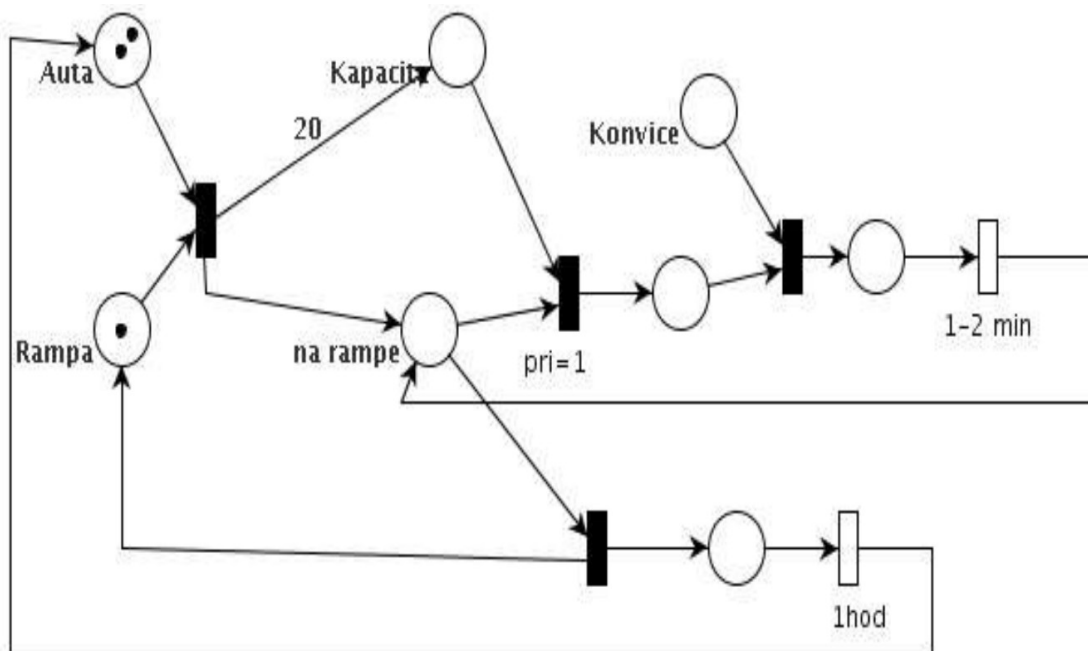
V rámci projektu byly naprogramovány dva simulační modely. Hlavní model představuje model „Krávy dojičky auto“. Druhým(doplňkovým) modelem je pak model vleku. Popíšeme si pouze model krávy. V rámci projektového modelu byla nastavena kapacita místa konvice na 20.

Model byl převzat ze slidů k prvnímu DEMO cvičení strana 23 a 24.

Model:



Model krávy / dojička



Model auta

5.3. Výsledné statistiky modelu

- Celková statistika simulace

Celková statistika simulace zobrazuje souhrné informace o simulaci. Transakcemi je v tomto případě myšleno počet vykonaných přechodů.

```
#####
#                                     #
#                               Statistika:                               #
#####
#      Přechody:      #      Mista:      #      Hrany:      #      Transakci:      #
#####
#           13           #           15           #           34           #           751           #
#####
#                                     #
#                               Čas simulace:                               #
#####
#                                     4.9015                                     #
#####
```

- **Informace i místech**

Informace o místech zobrazují detaily o místech modelu včetně jeho počátečního a konečného stavu, sloupec „ček.“ informuje o tom kolik značek v tomto místě je ve stavu čekajícím tedy naplánovaných v kalendáři.

#	Mista:										#	
#	ID:	#	Popis:	#	Poč stav:	#	Kon stav:	#	Ček.:	#	Kapacita:	#
#	0	#	P0	#	100	#	99	#	99	#	0	#
#	1	#	P1	#	0	#	0	#	0	#	0	#
#	2	#	Dojicky	#	5	#	4	#	0	#	0	#
#	3	#	P3	#	0	#	0	#	0	#	0	#
#	4	#	P4	#	0	#	0	#	0	#	0	#
#	5	#	P5	#	0	#	1	#	1	#	0	#
#	6	#	P6	#	0	#	0	#	0	#	0	#
#	7	#	Konvice	#	0	#	0	#	0	#	20	#
#	8	#	Auta	#	2	#	1	#	0	#	0	#
#	9	#	Rampa	#	1	#	0	#	0	#	0	#
#	10	#	Na rampě	#	0	#	0	#	0	#	0	#
#	11	#	Kapacita	#	0	#	0	#	0	#	0	#
#	12	#	P7	#	0	#	1	#	0	#	0	#
#	13	#	P8	#	0	#	0	#	0	#	0	#
#	14	#	P9	#	0	#	0	#	0	#	0	#

- **Informace o přechodech**

Informace o přechodech zobrazují pouze přehled všech přechodů a jejich atributů jako je zpoždění, šance, priorita.

Přechody:						
ID:	Popis:	Priorita:	Šance:	Časování:	Typ časování:	
0	Exp 15h	0	0	15	#Exponenciální	#
1	T1	0	0	0	# Nečasovaný	#
2	Šance 10%	0	0.1	0	# Nečasovaný	#
3	Šance 90%	0	0.9	0	# Nečasovaný	#
4	Exp 15min	0	0	0.25	#Exponenciální	#
5	Exp 8min	0	0	0.133333	#Exponenciální	#
6	T6	0	0	0	# Nečasovaný	#
7	T7	0	0	0	# Nečasovaný	#
8	Priorita 1	1	0	0	# Nečasovaný	#
9	T9	0	0	0	# Nečasovaný	#
10	T10	0	0	0	# Nečasovaný	#
11	1h	0	0	1	# konstantní	#
12	1,5min	0	0	0.025	# konstantní	#

- **Informace o hranách**

Informace o hranách zobrazují hrany v modelu a stejně jako přechody slouží pouze pro přehled.

Hrany:					
ID:	Popis:	Místo:	Směr:	Přechod:	
0		0	-->	0	#
1		1	<--	0	#
2		1	-->	1	#
3		2	-->	1	#
4		3	<--	1	#
5		3	-->	2	#
6		4	<--	2	#
7		3	-->	3	#
8		5	<--	3	#
9		4	-->	4	#

- **Transakce**

Přehled transakcí je nejdůležitější tabulka informací. Zobrazuje všechny transakce které v modelu proběhly ovšem bývá značně obsáhlá. Směr „-->“ značí že proběhl přenos po hraně od místa k transakci.

Transakce:											
Čas:	#	Místo:	#	Směr:	#	Akt H:	#	váha:	#	Přechod:	#
0	#	8	#	-->	#	2	#	1	#	7	#
0	#	9	#	-->	#	1	#	1	#	7	#
0	#	11	#	<--	#	0	#	20	#	7	#
0	#	10	#	<--	#	0	#	1	#	7	#
0	#	11	#	-->	#	20	#	1	#	8	#
0	#	10	#	-->	#	1	#	1	#	8	#
0	#	12	#	<--	#	0	#	1	#	8	#
0.0534487	#	0	#	-->	#	100	#	1	#	0	#
0.0534487	#	1	#	<--	#	0	#	1	#	0	#
0.0534487	#	1	#	-->	#	1	#	1	#	1	#

6. Shrnutí simulačních experimentů a závěr.

Na zvolených modelech bylo dokázáno že simulátor je funkční. Ze statistických informací bylo zjištěno že simulátor provede za 1000 jednotek simulačního času tedy 1000h modelového času cca 140 000 transakcí dále bylo zjištěno že nejslabším článkem v modelu jsou krávy které nestíhají produkovat dostatečné množství mléka a tedy dojičky a auto často nepracují. Ze získaných výsledků lze předpokládat že zadaný model je nekonečný.