

Úvod do Petriho sítí

Úvodní poznámky

- **Petriho sítě** (Petri Nets, PN) představují matematický nástroj pro modelování a simulaci diskrétních systémů (např. systémů hromadné obsluhy apod.).
- Základy položil německý matematik Carl Adam Petri ve své disertační práci „Kommunikation mit Automaten“ v roce 1962.
- Od té doby byly Petriho sítě postupně rozvíjeny a byla tak zvyšována jejich modelovací schopnost.
- Cílem této prezentace je ve stručnosti si představit jednotlivé typy Petriho sítě a ukázat si základní principy Petriho sítí.

Úvodní poznámky

- Postupně byly vytvořeny následující typy Petriho sítí:
 - 1) C/E (Condition / Event) Petriho síť.
 - 2) P/T (Place / Transition) Petriho síť.
 - 3) P/T Petriho síť s inhibičními hranami.
 - 4) P/T Petriho síť s prioritami.
 - 5) TPN Časované (Timed) Petriho síť.
 - 6) CPN Barevné (Coloured) Petriho síť.
 - 7) HPN Hierarchické (Hierarchical) Petriho síť.

C/E Petriho sítě

- Model se skládá z **událostí** (events) a **podmínek** (conditions), které musí být splněny, aby mohla nastat určitá událost. Vazby mezi událostmi a podmínkami jsou znázorněny pomocí **orientovaných hran** (arcs).
- Koncept C/E PN byl vytvořen právě C. A. Petrim.

C/E Petriho síť

- C/E Petriho síť je tedy definována těmito prvky (a na základě nich může být i graficky znázorněna):
 - **Podmínkami** – podmínky zobrazujeme zpravidla kroužky.
 - **Událostmi** – události zpravidla zobrazujeme jako obdélníky, případně usečky.
 - **Orientovanými hranami**, které mohou směřovat buď od podmínky k události nebo od události k podmínce.

C/E Petriho síť

- **Tokeny** (značkami), které zakreslujeme jako tečky v kroužcích odpovídajících jednotlivým podmínkám a které vyjadřují pravdivost jednotlivých podmínek (0 – token se v dané podmínce nenachází, podmínka není splněna, 1 – token se v dané podmínce nachází, podmínka je splněna).
- **Počátečním značením** (initial marking) – počátečním značením rozumíme počáteční rozložení tokenů v síti.

C/E Petriho sítě

- Podmínka c je **vstupní podmínkou** (precondition) události e , pokud je zobrazena orientovanou hranou směřující od podmínky c k události e .
- Podmínka c je **výstupní podmínkou** (postcondition) události e , jestliže je zobrazena orientovanou hranou směřující od události e k podmínce c .
- Každá podmínka může být buď nesplněna (bez tokenu) nebo splněna (s tokenem).

C/E Petriho síť

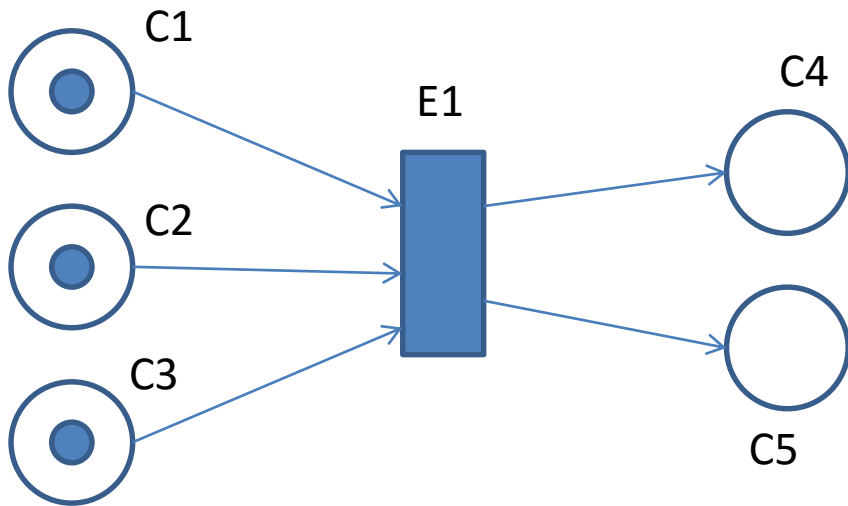
- **Značení síť** (marking) v určitém okamžiku je dána rozložením tokenů v síti (tedy splněním, resp. nesplněním jednotlivých podmínek).
- Změna značení síť (stavu síť) se děje uskutečňováním událostí.
- Událost e je **proveditelná** (enabled), jsou-li splněny všechny její vstupní podmínky a zároveň nesplněny všechny výstupní podmínky.

C/E Petriho sítě

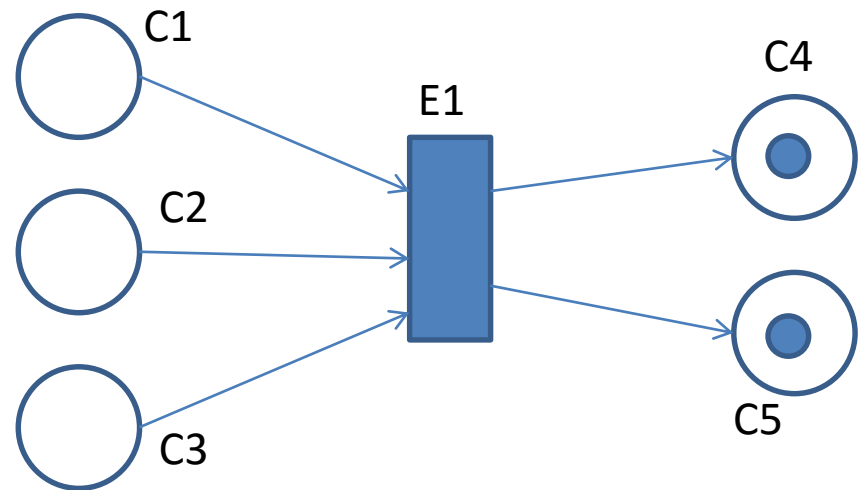
- Po **provedení** proveditelné události se změni značení sítě – všechny vstupní podmínky jsou nesplněny (jsou odebrány příslušné tokeny) a všechny výstupní podmínky jsou splněny (jsou přidány tokeny) – událost byla provedena (fired).
- To, že je událost proveditelná, ještě neznamená, že bude i provedena (konflikt událostí).

C/E Petriho síť

Značení před provedením události E1.

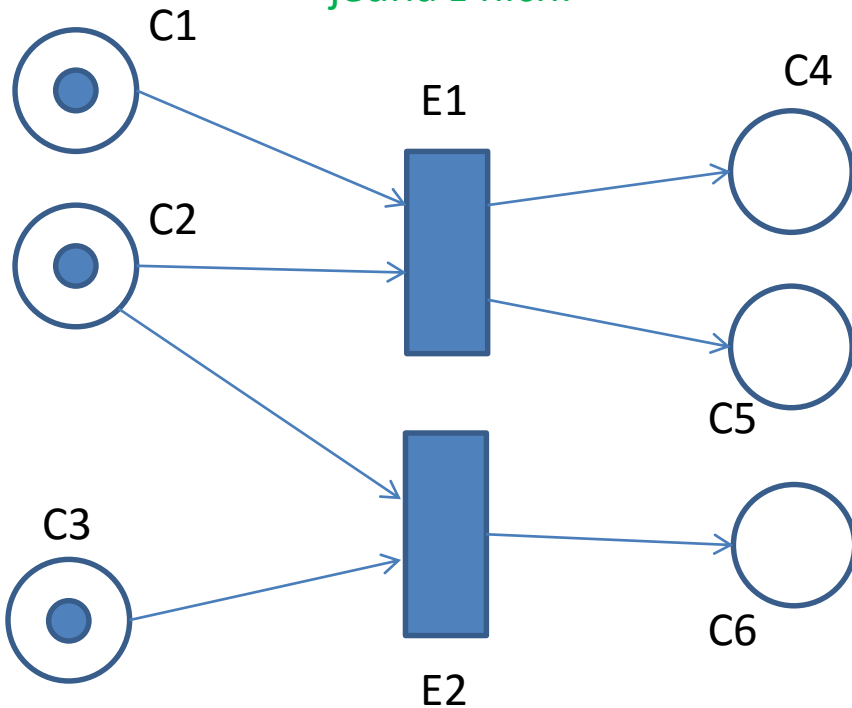


Značení po provedení události E1.

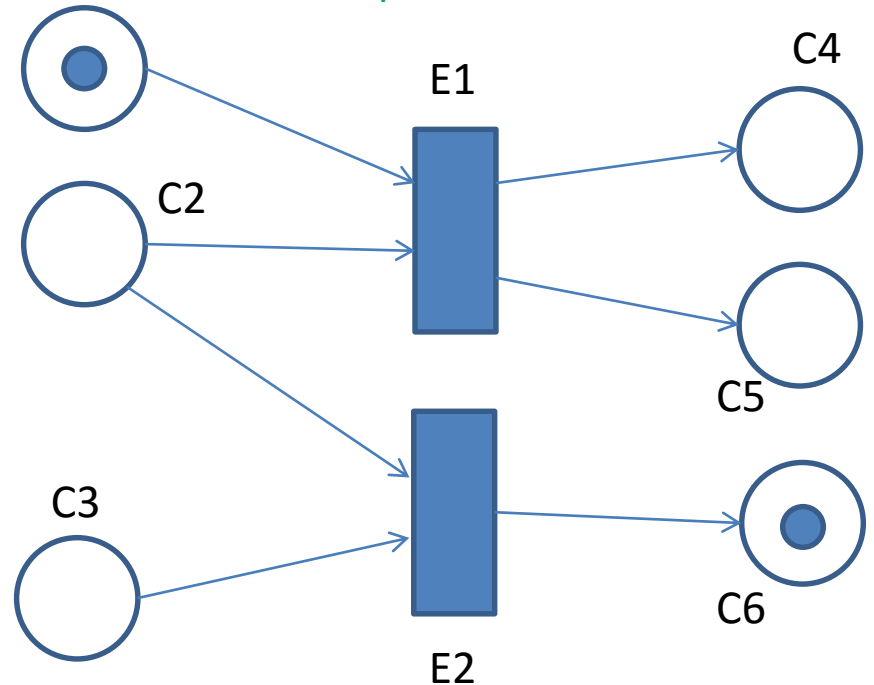


C/E Petriho sítě

Je zřejmé, že obě události jsou v tomto značení proveditelné, provedena může být ovšem pouze jedna z nich.

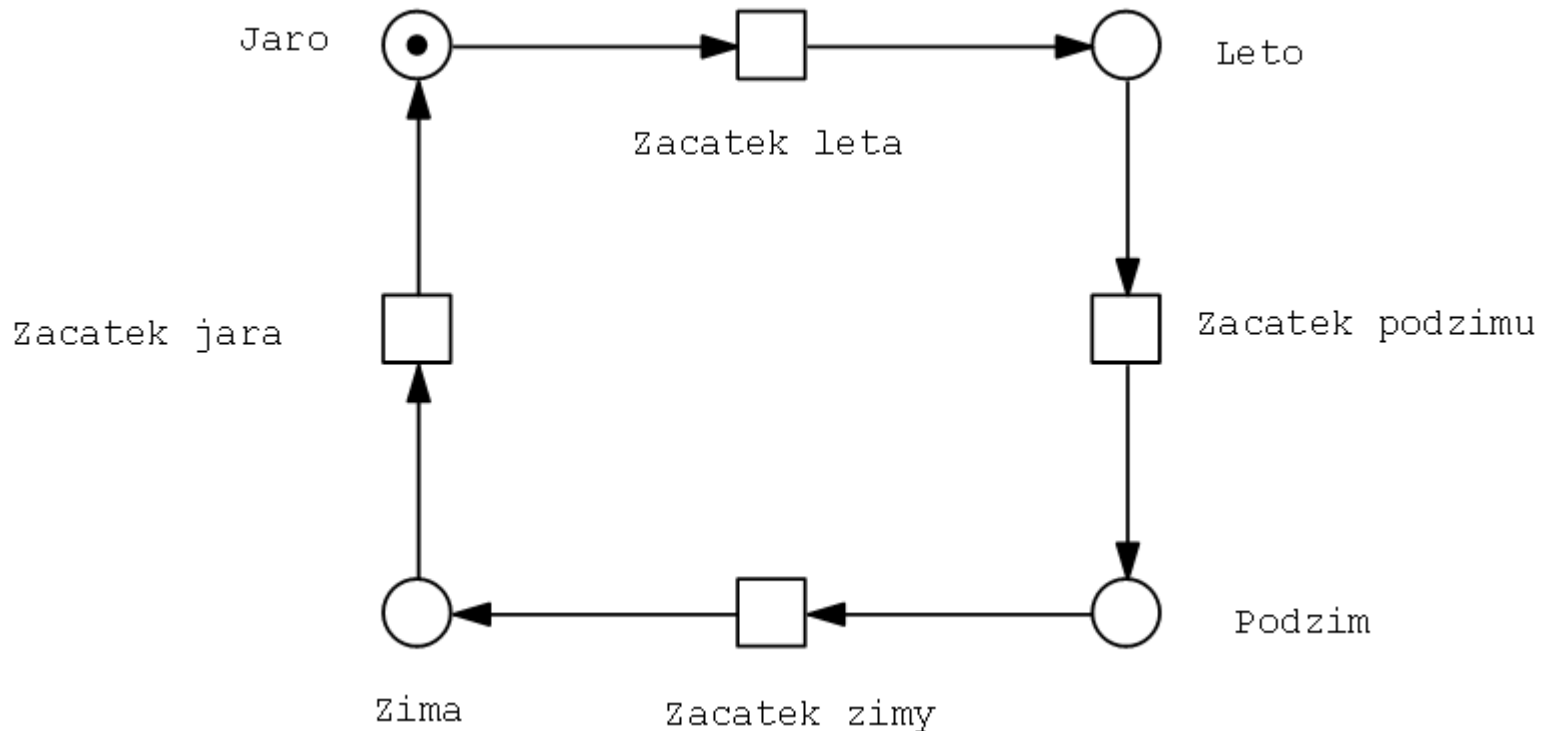


Provedeme-li např. událost E2, změní se značení sítě tak, jak je uvedeno na obrázku. Vidíme, že událost E1 už není proveditelná.



C/E Petriho síť

- Na obrázku je znázorněn jednoduchý model střídání ročních období vytvořený pomocí C/E Petriho sítě.



C/E Petriho síť

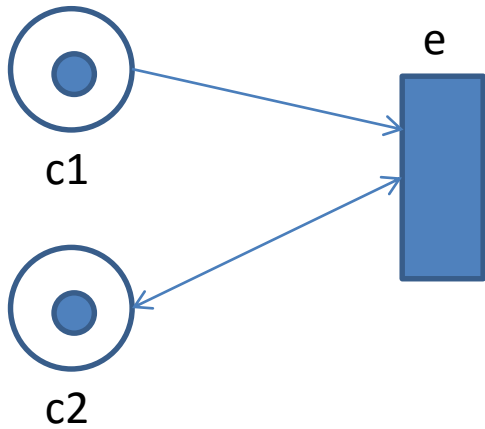
- **Testovací hrana** je smyčka tvořená podmínkou c , událostí e a hranami $[c;e]$ a $[e;c]$. Pomocí testovací hrany událost e testuje podmínku c , aniž by se měnila její platnost. Testovací hranu zpravidla zakreslujeme jednou hranou se šipkami na obou stranách.
- **Čistá síť** je speciálním případem C/E PN bez testovacích hran.

C/E Petriho sítě

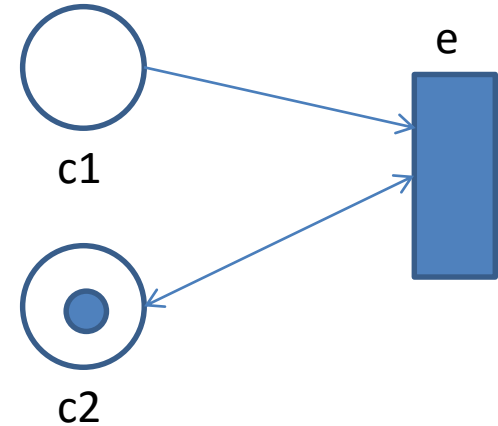


Na obrázku je znázorněna testovací hrana.

C/E Petriho síť



Událost e je proveditelná, neboť obě vstupní podmínky jsou splněny (obsahují token).



Po provedení události e se změnilo značení sítě tak, že podmínka c1 je odebrán token, podmínka c2 token zůstává (je spojena testovací hranou). Situaci lze představit tak, že token je odebrán a ihned vrácen zpět.

P/T Petriho síť

- Původní koncept C/E Petriho síť byl rozšířen na **P/T Petriho síť**.
 - Místo podmínek jsou **místa** (places), místo událostí jsou **přechody** (transitions).
 - Pomocí míst zpravidla modelujeme stavy systému a pomocí přechodů změny stavů systému. Vazby jsou opět znázorněny pomocí orientovaných hran.

P/T Petriho síť

- P/T Petriho síť je tedy tvořena následujícími prvky:
 - **Místa** (places) graficky znázorňované kroužkem.
 - **Přechody** (transitions) graficky znázorňované obdélníky.
 - **Orientovanými hranami** směřujícími buď od místa k přechodu nebo od přechodu k místu.
 - Udáním **kapacity** každého místa sítě (kapacita místa udává maximální počet tokenů, které se mohou v místě v jeden okamžik nacházet).

P/T Petriho síť

- Udáním **váhy** (násobnosti) každé hrany síť. Váha hrany uvádí, kolik tokenů se při provedení přechodu po dané hraně přesouvá.
- Udáním **počátečního značení** síť, jež je dáno počty tokenů v jednotlivých místech síť.
- Není-li kapacita místa uvedena, považuje se kapacita za nekonečnou.
- Není-li váha hrany uvedena, považuje se váha hrany rovna 1.

P/T Petriho síť

- Stav sítě (značení sítě) v určitém okamžiku je dáno počty tokenů v jednotlivých místech sítě.
- Místo p patří do **vstupní množiny** (pre-set) přechodu t , jestliže z místa p vede hrana do přechodu t .
- Místo p patří do **výstupní množiny** (post-set) přechodu t , jestliže z přechodu t vede hrana do místa p .

P/T Petriho síť

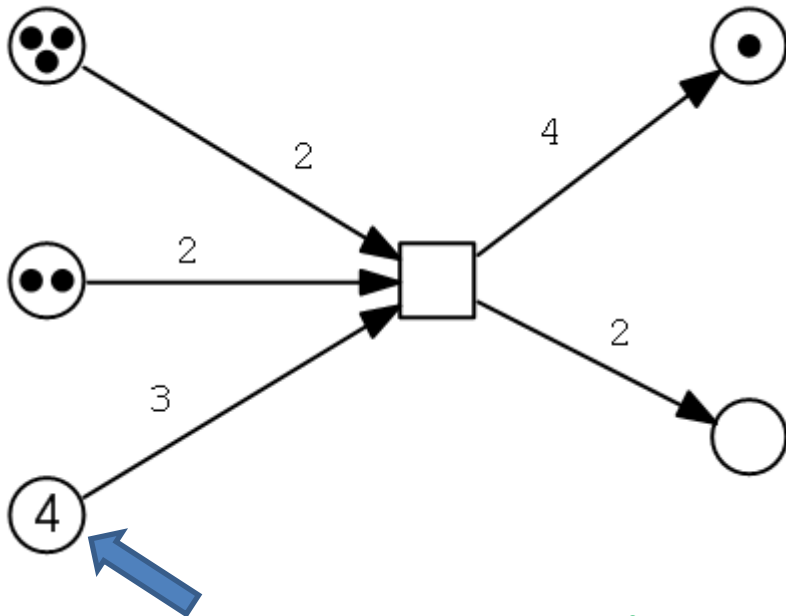
- Přejchod t je **proveditelný** (enabled, activated), pokud:
 - Pro každé místo p vstupní množiny přechodu t platí, že obsahuje alespoň tolik tokenů, kolik činí váha hrany vedoucí z místa p do přechodu t .
 - Pro každé místo p výstupní množiny přechodu t platí, že počet tokenů obsažených v místě p zvětšený o váhu hrany směřující z přechodu t do místa p nepřevyšuje kapacitu místa p .

P/T Petriho síť

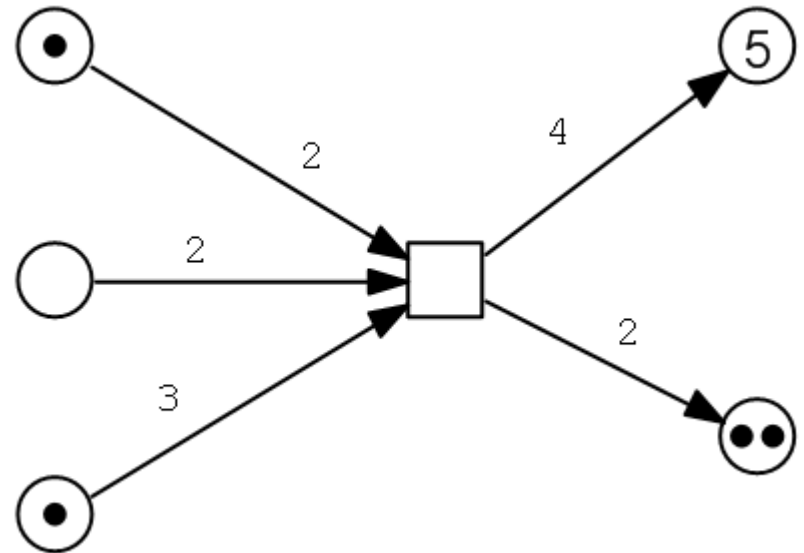
- **Provedením** proveditelného přechodu t dojde ke změně značení sítě následujícím způsobem:
 - V každém místě p vstupní množiny přechodu t se počet tokenů sníží o váhu hrany směřující z místa p do přechodu t .
 - V každém místě p výstupní množiny přechodu t se počet tokenů zvýší o váhu hrany směřující z přechodu t do místa p .

P/T Petriho síť

Značení před provedením přechodu.



Značení po provedení přechodu.



V případě vyššího počtu tokenů v místě se tokeny nezakreslují, ale jejich počet se vyjádří číslem. Kdyby toto místo obsahovalo např. pouze 2 tokeny, přechod by nebyl proveditelný.

P/T Petriho sítě

- Při provádění přechodů obecně neexistuje žádné pravidlo zachování počtu tokenů, tzn., že provedením přechodu mohou tokeny libovolně vznikat či zanikat.
- Stejný počet tokenů na vstupu i na výstupu bude pouze v případě, kdy součet ohodnocení hran spojujících vstupní místa přechodu s přechodem bude roven součtu ohodnocení hran spojujících výstupní místa přechodu s přechodem.

P/T Petriho síť

- C/E Petriho síť je speciálním případem P/T Petriho sítě, kde kapacita každého místa je rovna 1 a váha každé hrany je taky rovna 1. Každá P/T Petriho síť lze převést na ekvivalentní C/E Petriho síť (ovšem se složitější strukturou). C/E Petriho síť tedy pouze umožňují jednodušší zápis.

P/T Petriho sítě

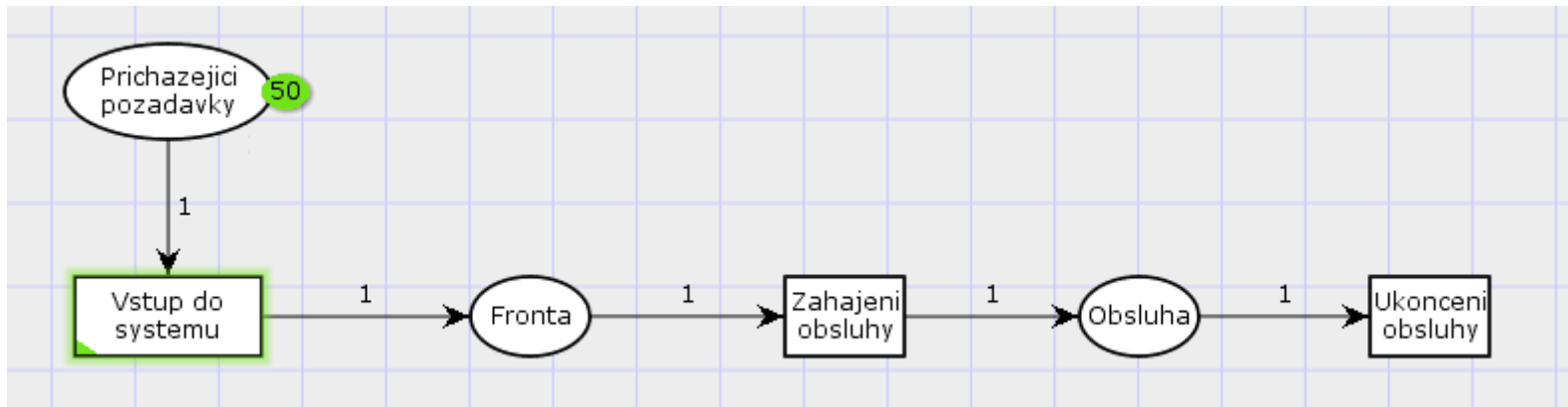
- **Obyčejná Petriho síť** je P/T Petriho síť, ve které je kapacita každého místa nekonečná a násobnost všech hran rovna 1.
- V P/T Petriho síti se mohou samozřejmě vyskytovat i **testovací hrany** jako v C/E Petriho sítích.

P/T Petriho sítě

- Testovací hrany jsou užitečné např. v případě modelování systémů hromadné obsluhy s odmítáním.
- Déle je důležité znát koncept tzv. komplementárních míst, pomocí kterých lze modelovat omezenou kapacitu míst.
- Tyto koncepty si představíme na následujících příkladech.

P/T Petriho síť

- Uvažujme systém hromadné obsluhy, jehož síť je uvedena na obrázku.



P/T Petriho sítě

- Místa mají tento význam:
 - Přicházející požadavky – v tomto místě se nacházejí požadavky, které přichází k systému (na obrázku jich je 50).
 - Fronta – modeluje frontu v SHO.
 - Obsluha – modeluje obsluhu v SHO.

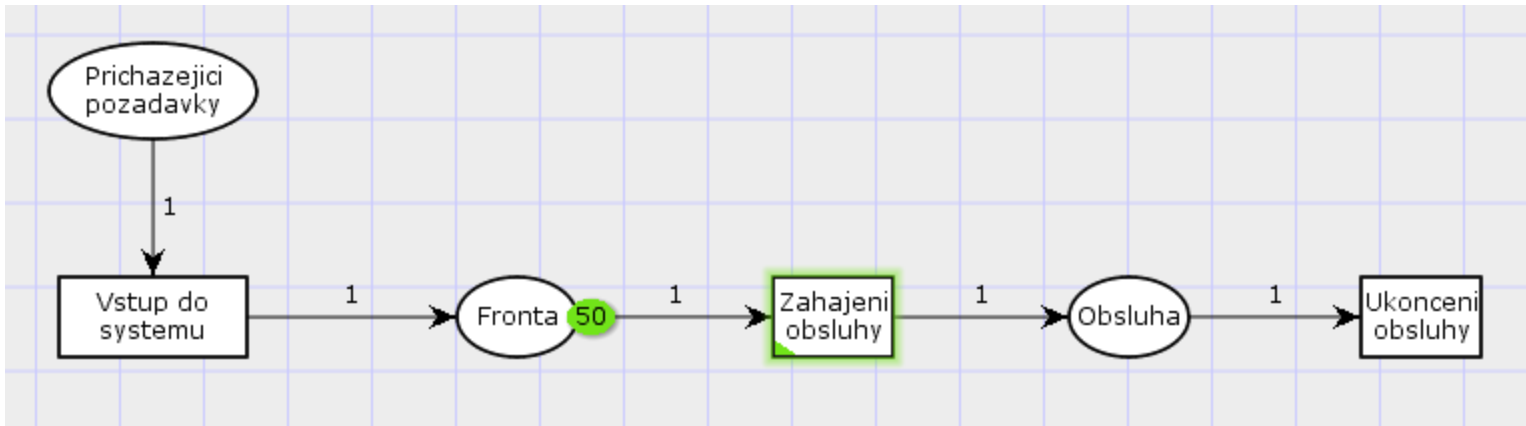
P/T Petriho síť

- Přechody mají následující význam:
 - Vstup do systému – modeluje vstup požadavku do systému.
 - Zahájení obsluhy – modeluje přechod požadavku z fronty do obsluhy.
 - Ukončení obsluhy – modeluje odchod požadavku ze systému.

P/T Petriho sítě

- Jelikož nástroj, ve kterém byla síť vytvořena, neumí definovat omezenou kapacitu míst, jedná se o model SHO, který má neomezený počet míst ve frontě i v obsluze.

P/T Petriho síť



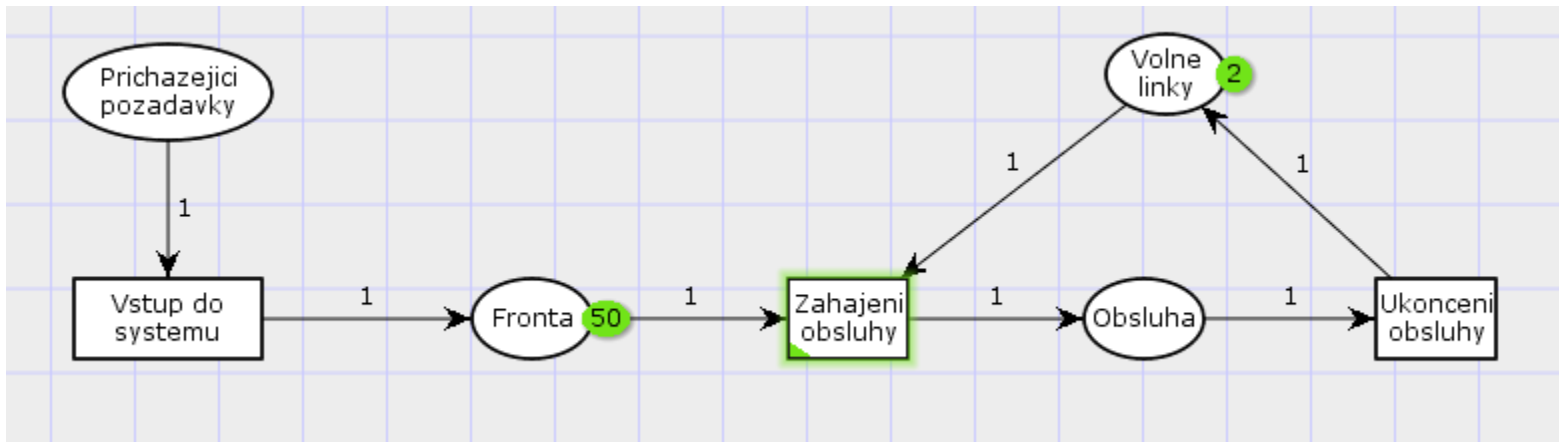
Provedeme-li např. 50 krát přechod Vstup do systému, získáme 50 požadavků ve frontě. Takto bychom mohli pokračovat teoreticky až do nekonečna (pokud bychom v místě Přicházející požadavky měli nekonečně mnoho tokenů). To samé platí i pro místo Obsluha

P/T Petriho síť

- Nyní bychom chtěli docílit stavu, aby model odpovídal systému hromadné obsluhy, který má dvě obslužné paralelně umístěné linky.
- Tento problém vyřešíme pomocí komplementárního místa. Jeho počáteční značení bude odpovídat počtu obslužných linek.

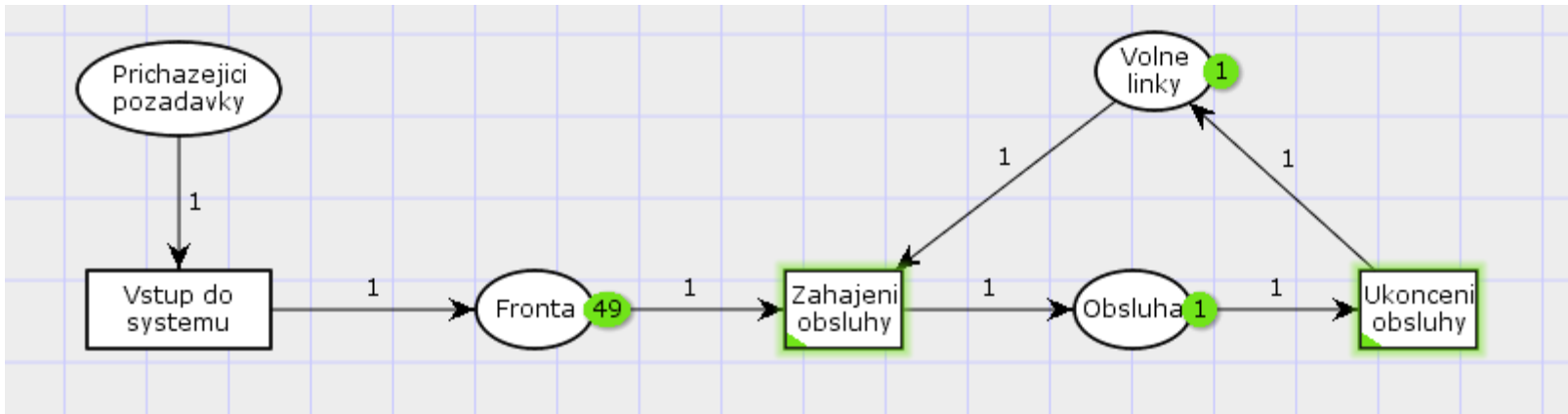
P/T Petriho síť

Místo Volné linky je komplementárním místem k místu Obsluha, počáteční značení tohoto komplementárního místa jsme zvolili 2 (2 volné obslužné linky). Provedením přechodu Zahájení obsluhy je odebrán token z komplementárního místa (ubude nám volná obslužná linka).



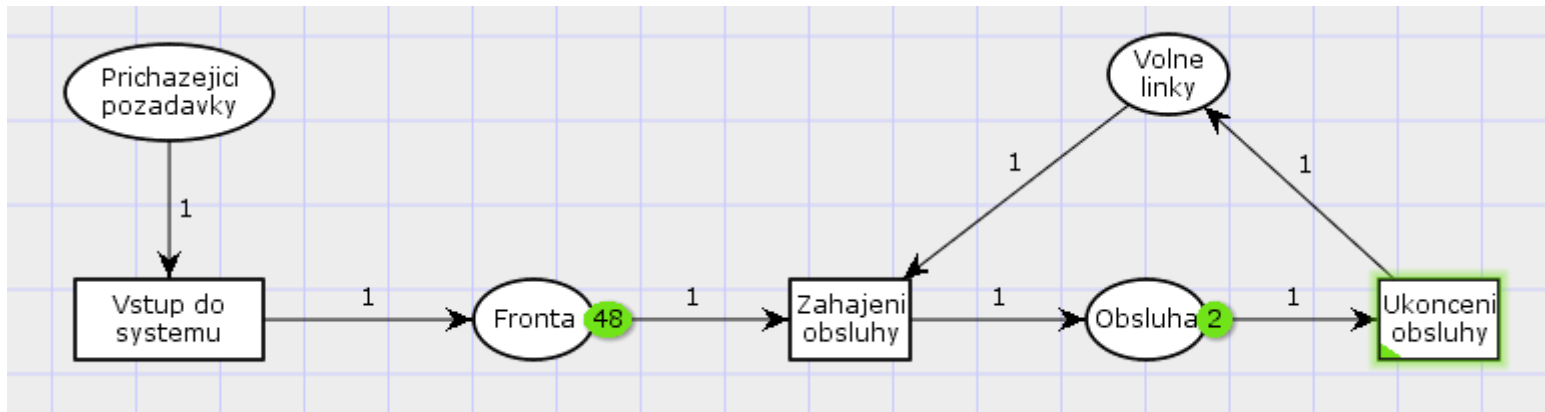
P/T Petriho síť

Provedením přechodu Zahájení obsluhy je odebrán token z komplementárního místa (ubude nám volná obslužná linka), současně dojde k odebrání tokenu z místa Fronta a přidání tokenu do místa Obsluha. V tomto značení provádíme obsluhu jednoho zákazníka a jedna obslužná linka je volná. Vidíme, že přechod Zahájení obsluhy je stále proveditelný, neboť se nachází zákazníci ve frontě a současně máme k dispozici volnou linku.

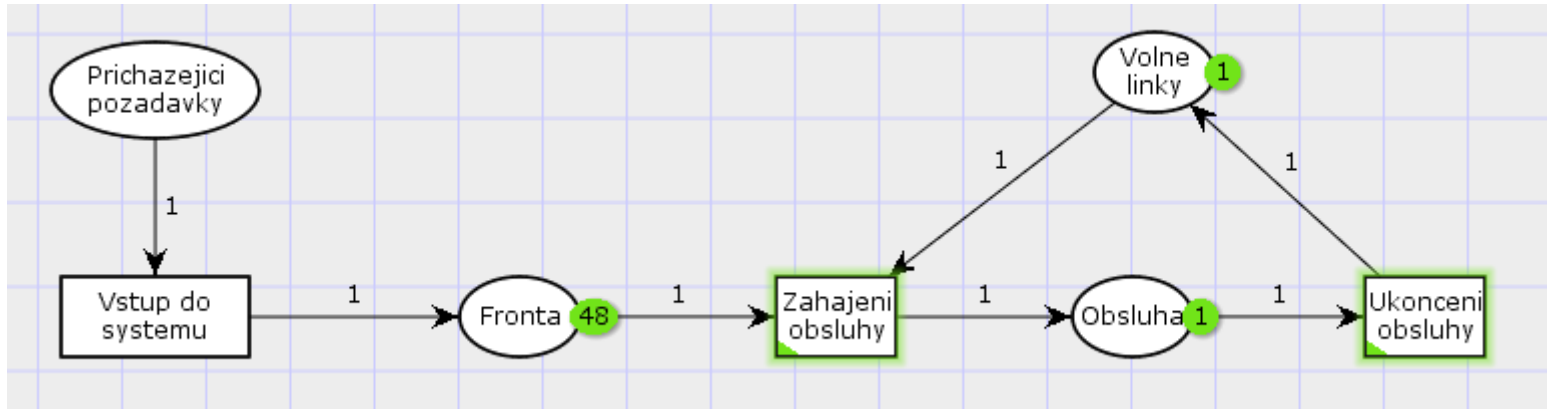


P/T Petriho síť

Dalším provedením přechodu Zahájení obsluhy je odebrán poslední token z komplementárního místa (opět ubude nám volná obslužná linka), současně dojde k odebrání tokenu z místa Fronta a přidání tokenu do místa Obsluha. V tomto značení provádíme obsluhu dvou zákazníků a žádná obslužná linka není volná. Vidíme, že přechod Zahájení obsluhy již není proveditelný, neboť již nemáme k dispozici volnou obslužnou linku. Přechod se stane znovu proveditelný, až ukončíme obsluhu některého ze dvou právě obsluhovaných požadavků.

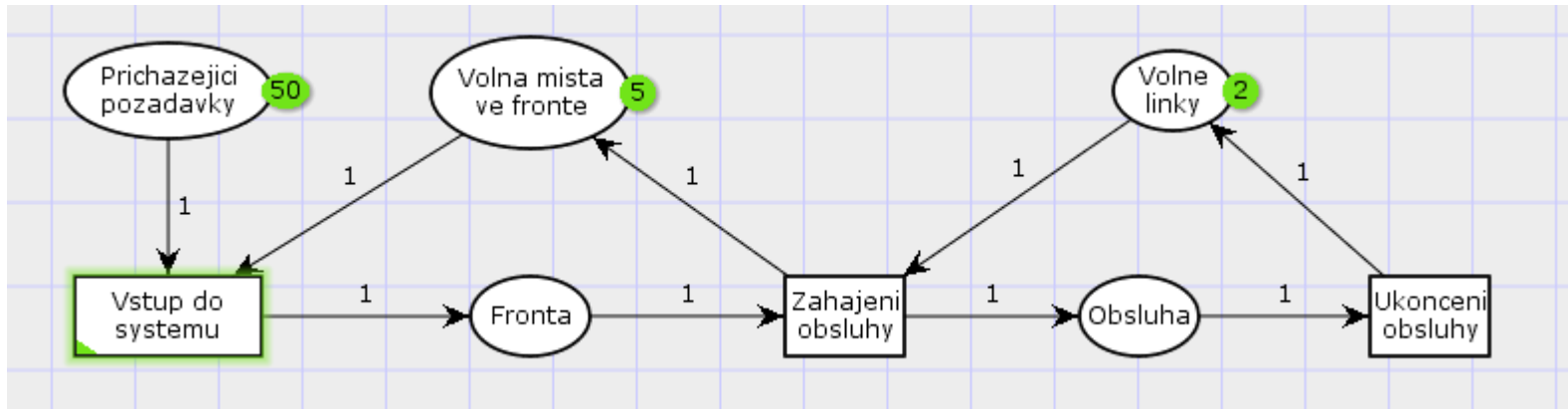


P/T Petriho síť

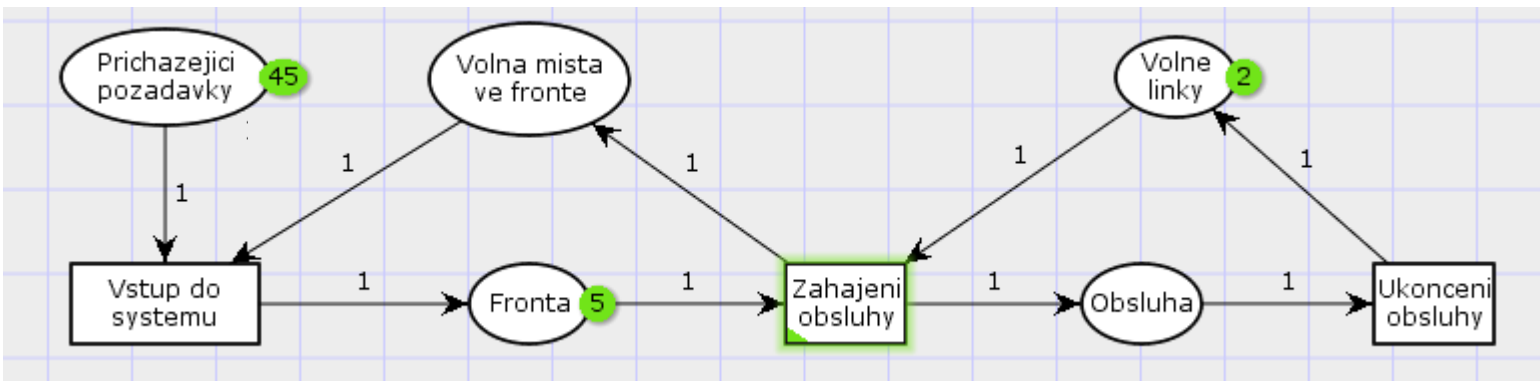


Na tomto snímku vidíte stav sítě po provedení přechodu Ukončení obsluhy. Vidíme, že došlo k vrácení jednoho tokenu do místa Volné linky, což má za následek, že přechod Zahájení obsluhy je znovu proveditelný (samozřejmě za předpokladu, že v místě Fronta je alespoň jeden token).

P/T Petriho síť



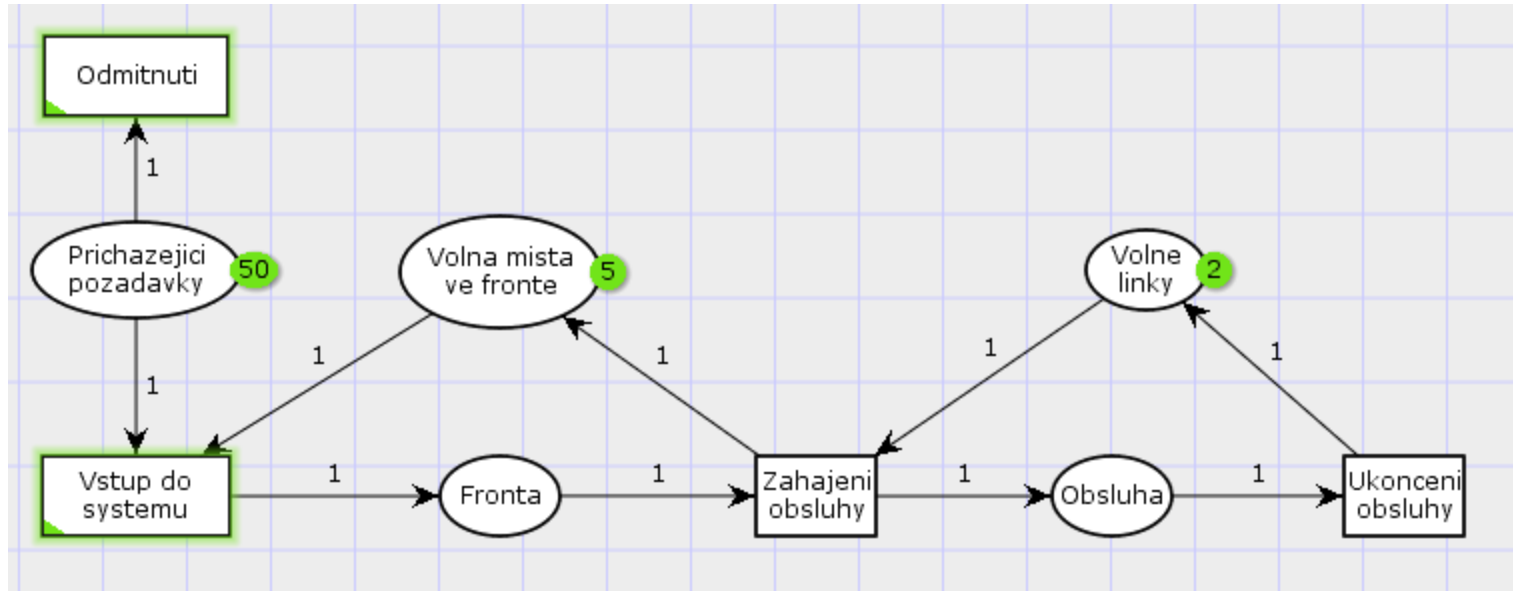
Na tomto snímku vidíte síť, která modeluje systém hromadné obsluhy, který má 5 míst ve frontě a 2 místa v obsluze. V tomto případě nemůže do systému vstoupit požadavek, nachází-li se právě 5 požadavků ve frontě.



P/T Petriho síť

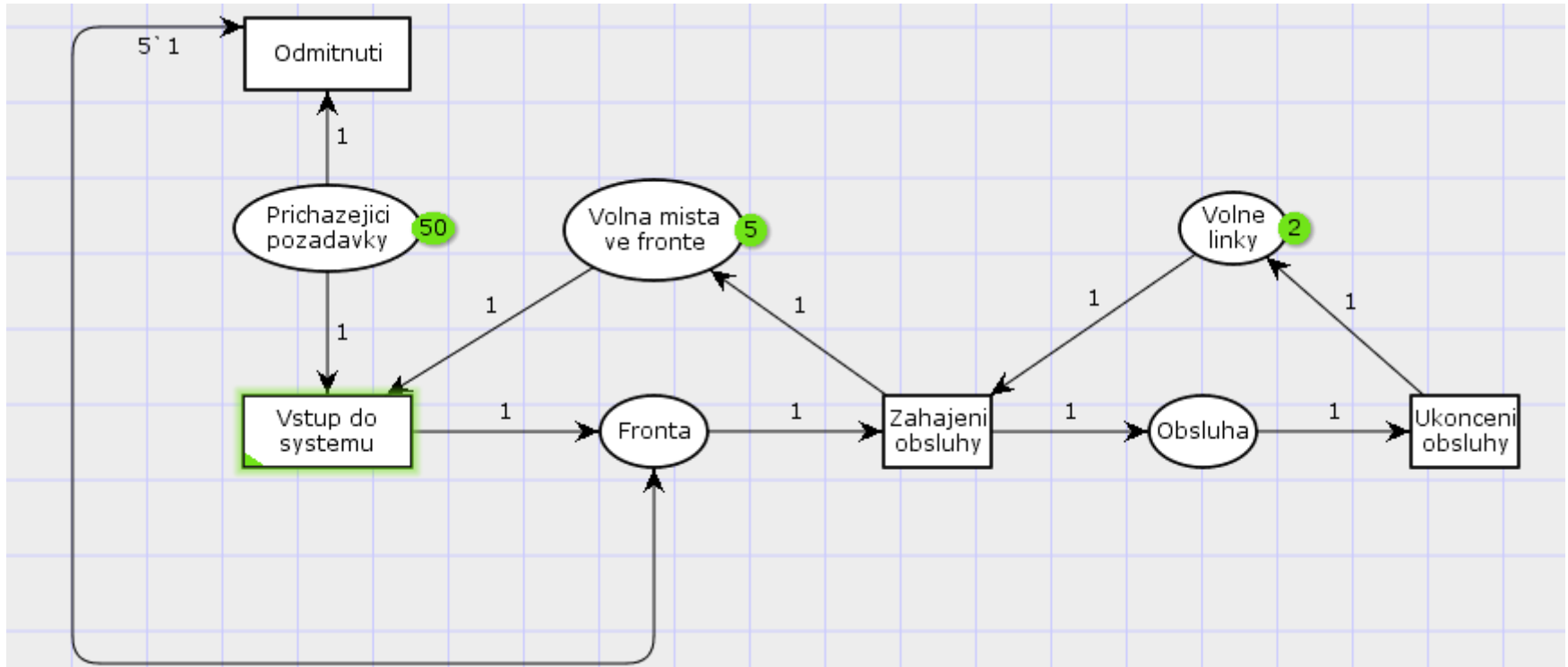
- V případě, že máme systém, který má omezený počet míst, nastává situace, kdy musíme modelovat odmítání požadavků na vstupu do systému. Toto můžeme realizovat novým přechodem Odmítnutí – viz následující snímek.

P/T Petriho síť



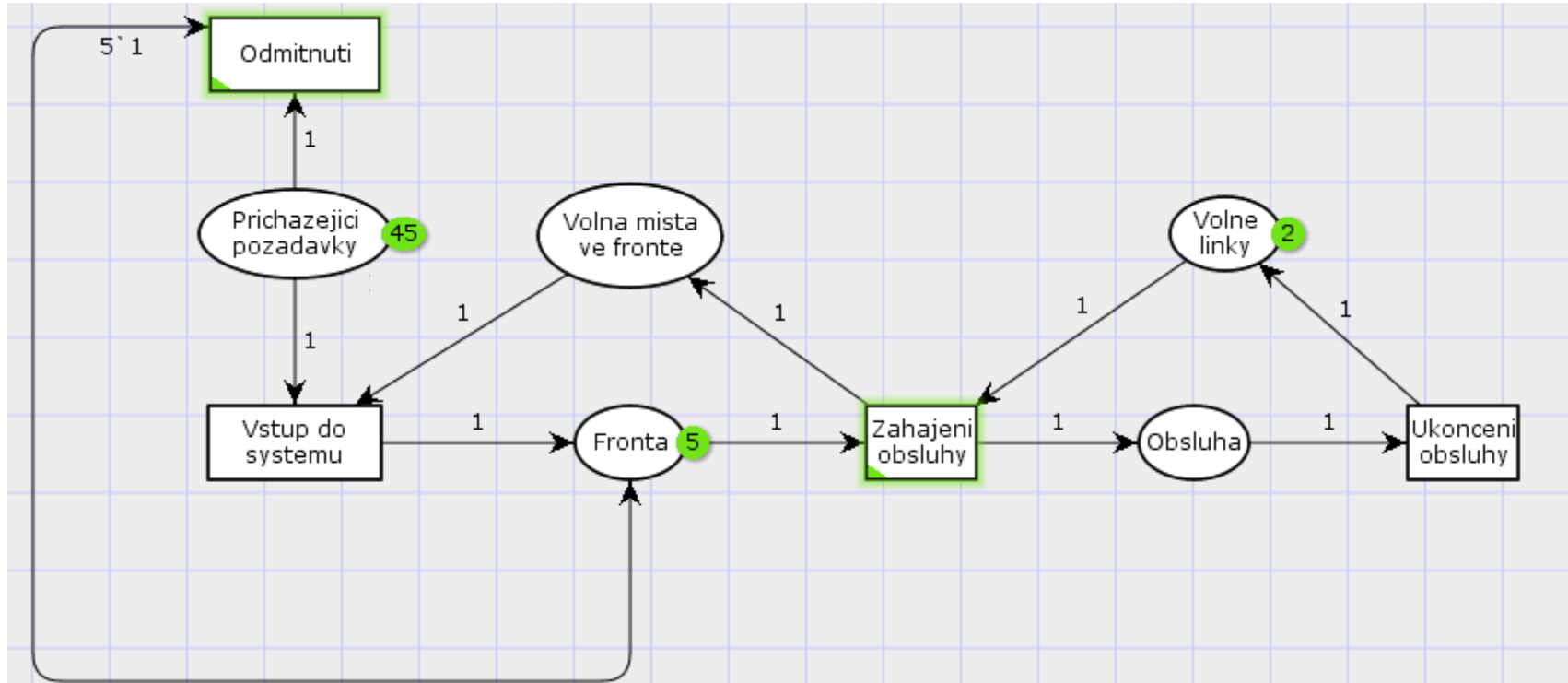
Přidali jsme do modelu přechod Odmítnutí. Nicméně nyní vidíme, že pokud se nachází nějaký token v místě Přicházející požadavky, jsou proveditelné přechody Vstup do systému i Odmítnutí. To znamená, že bychom mohli odmítnout požadavek i tehdy, je-li pro něj v systému místo. Tomu musíme zabránit, k čemuž s úspěchem využijeme právě testovací hrany.

P/T Petriho síť



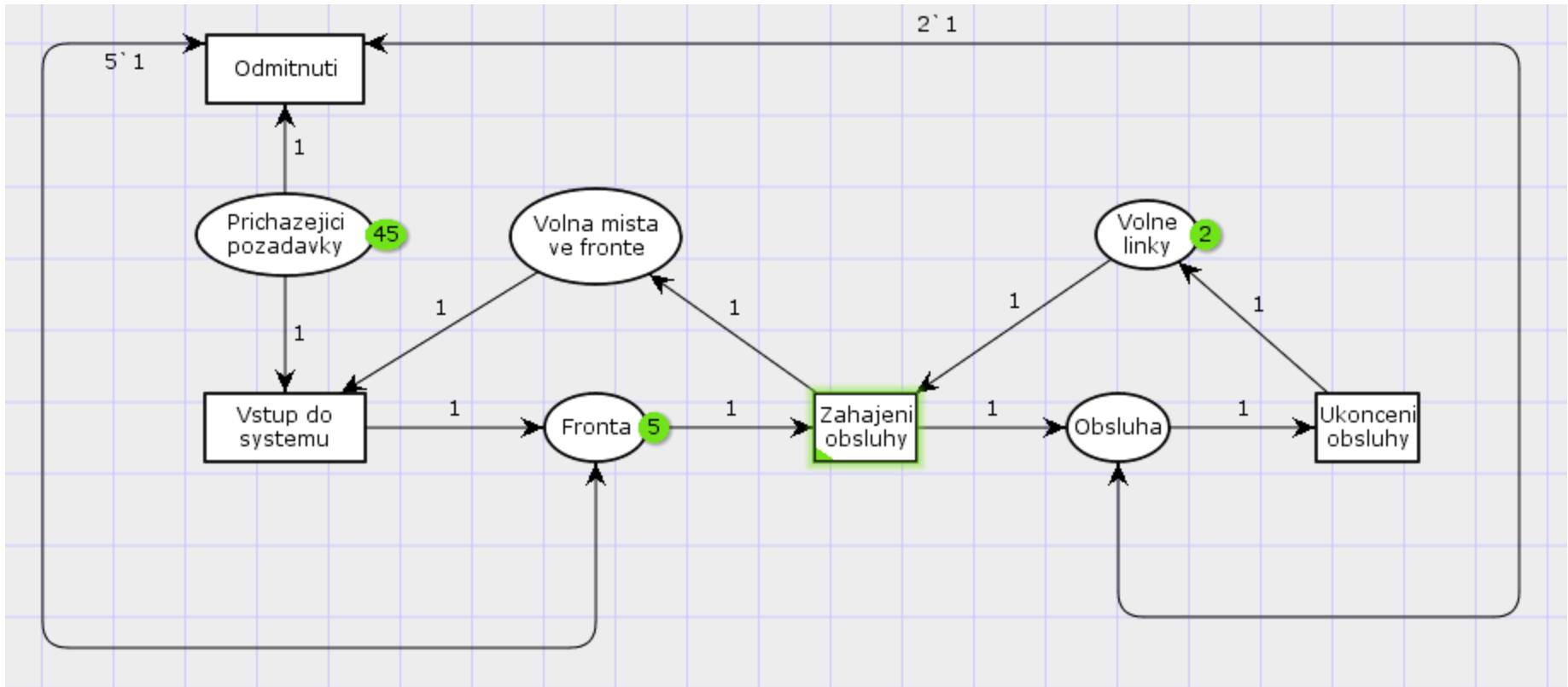
Místo Fronta jsme spojili testovací hranou s přechodem Odmítnutí, váha této hrany je 5, což znamená, že přechod Odmítnutí bude proveditelný pouze tehdy, bude-li se v místě Fronta nacházet alespoň 5 tokenů, což vidíme na následujícím snímku.

P/T Petriho síť



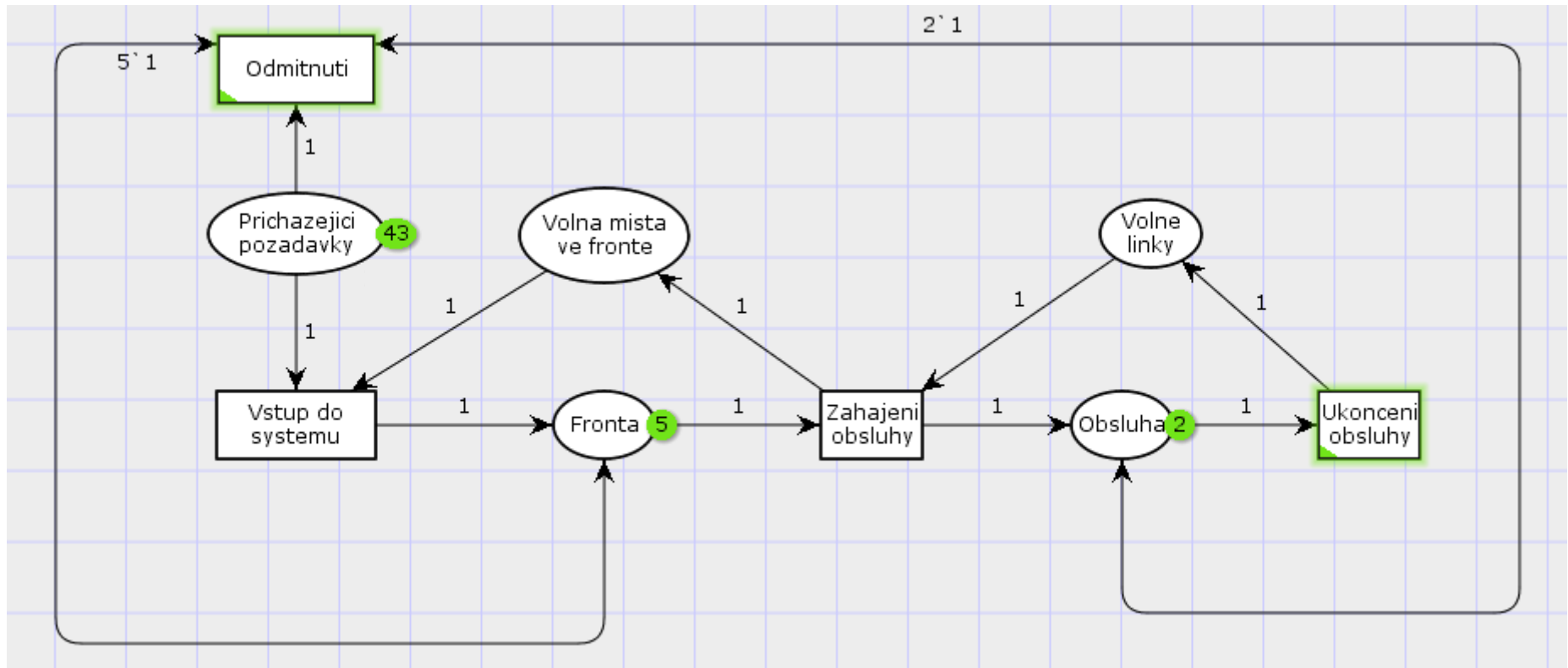
Nyní vidíme, že pokud je 5 tokenů v místě Fronta, přechod Odmítnutí se stal proveditelný. Nicméně vidíme, že v systému se nachází volná místa (konkrétně v obsluze), tudíž bychom ani v tomto značení neměli požadavky odmítat. Toto lze napravit další testovací hranou, kterou spojíme místo Obsluha s přechodem Odmítnutí.

P/T Petriho síť

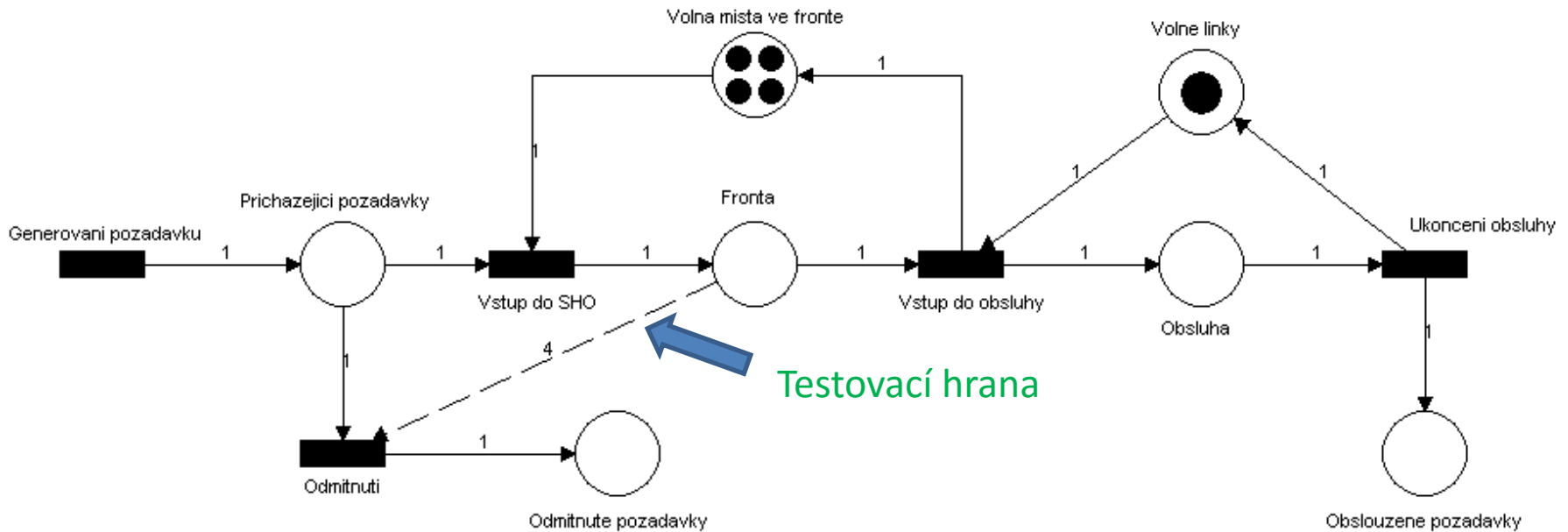


Přidáním této testovací hrany s váhou 2 již přechod Odmítnutí není proveditelný. V tomto případě bude proveditelný pouze tehdy, bude-li v místě Fronta alespoň 5 tokenů a současně v místě Obsluha alespoň 2 tokeny.

P/T Petriho síť



P/T Petriho síť



Na obrázku je model systému hromadné obsluhy se 4 místy ve frontě a 1 obslužnou linkou vytvořený pomocí P/T Petriho sítě. V této síti je uvažována nekonečná kapacita všech míst, proto bylo nutné pro zachování správné funkce modelovaného systému použít pomocná místa „Volna místa ve frontě“ a „Volne linky“ – tzv. **komplementární místa**.

P/T Petriho síť s inhibičními hranami

- **Inhibiční hrana** (inhibitor arc) je speciální hrana, která může směřovat pouze z místa k přechodu. U inhibiční hrany se zpravidla místo šipky zakresluje kolečko. Inhibiční hrany upravují proveditelnost přechodů.
- Místo p patří do **vstupní inhibiční množiny** přechodu t , jestliže z místa p vede inhibiční hrana do přechodu t .

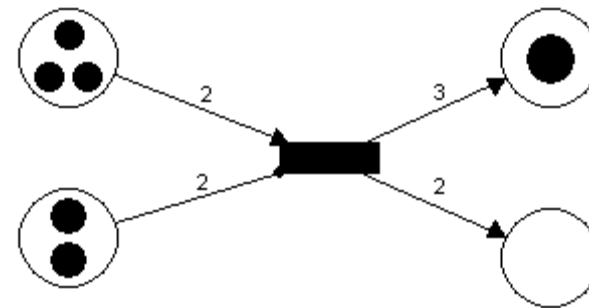
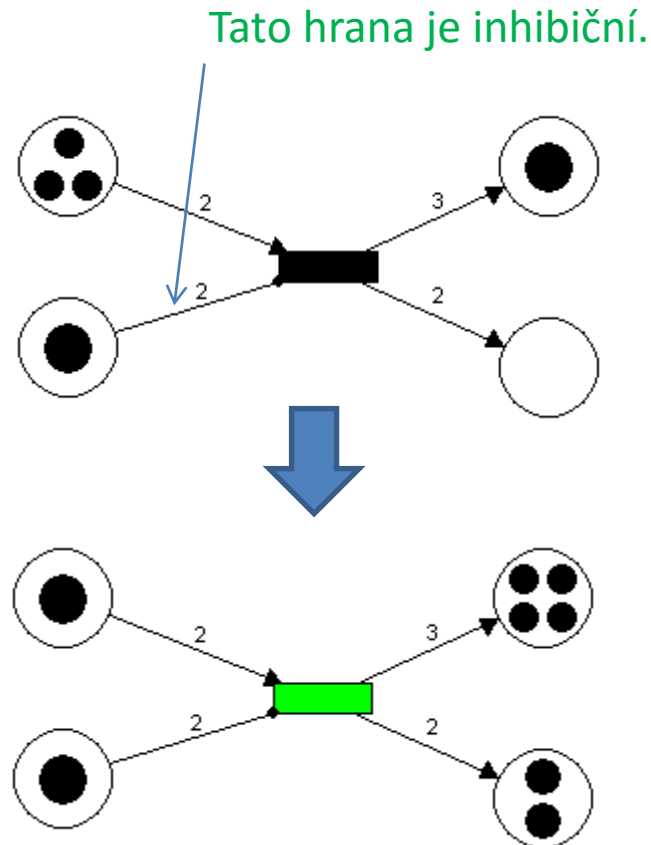
P/T Petriho síť s inhibičními hranami

- Přejchod t je **proveditelný**, jestliže:
 - Je přechod t proveditelný v příslušné P/T Petriho síti (tj. v síti, která vznikne z dané síti odstraněním inhibičních hran).
 - Pro každé místo p vstupní inhibiční množiny přechodu t platí, že obsahuje méně tokenů než kolik činí násobnost inhibiční hrany vedoucí z místa p do přechodu t .

P/T Petriho síť s inhibičními hranami

- Po **provedení** proveditelného přechodu t se změní značení sítě stejným způsobem jako v příslušné P/T Petriho síti. Po inhibiční hraně se tedy tokeny nepřesouvají. Inhibiční hrany můžeme chápat jako negativně pojaté testovací hrany.

P/T Petriho síť s inhibičními hranami



V tomto značení není přechod proveditelný.

P/T Petriho síť s prioritami

- **Petriho síť s prioritami** je P/T Petriho síť, ve které je každému přechodu přiřazeno celé nezáporné číslo udávající prioritu přechodu. Priority přechodů upravují pravidla pro jejich provádění.
- V Petriho síti s prioritami je přechod t **povolen** (has concession), je-li proveditelný v odpovídající P/T síti bez priorit.

P/T Petriho sítě s prioritami

- Přejchod t je **proveditelný** tehdy, pokud:
 - Je povolen.
 - Žádný jiný povolený přechod nemá vyšší prioritu.
- Po **provedení** přechodu t se značení sítě změní stejným způsobem jako v odpovídající P/T síti bez priorit.

Rozšíření Petriho sítí o čas

- Všechny dosud zmíněné typy Petriho sítí mají jednu nevýhodu – nepracují s časem, všechny změny v síti jsou provedeny okamžitě. V případě P/T sítí provedení přechodu odpovídá změně stavu systému a my nyní potřebujeme, aby tyto změny trvaly určitou dobu. Toto předchozí typy Petriho sítí neumožňují.

Rozšíření Petriho sítí o čas

- Trvání dějů může být:
 - 1) Deterministické – v tomto případě se hovoří o **Časovaných Petriho sítích**.
 - 2) Stochastické – v tomto případě hovoříme o **Stochastických Petriho sítích** (SPN – Stochastic Petri Nets).
 - 3) Kombinované – v tomto případě hovoříme o **Zobecněných stochastických Petriho sítích** (GSPN – Generalised Stochastic Petri Nets).

Rozšíření Petriho sítí o čas

- Zavedení času může být spojeno s:
 - **Přechody** (T-timed PN) – provedení přechodu trvá určitou dobu, po kterou token pobývá uvnitř přechodu.
 - **Místy** (P-timed PN) – token pobývá stanovenou dobu ve vstupním místě přechodu, jež má být proveden.
 - **Hranami** (A-timed PN) – přesun tokenu po příslušné hraně trvá určitou dobu.

Rozšíření Petriho sítí o čas

- **Tokeny** (Token Timed PN) – Provádění přechodů v síti je sice okamžité, ale tokeny opouštějící příslušný přechod jsou opatřeny **časovým razítkem** (time stamp), jež udává, kdy může být daný token zase použit. Hodnota časového razítka odpovídá aktuální hodnotě simulárního času zvětšenou o příslušnou hodnotu.

Barevné Petriho sítě

- **Barevná Petriho síť** je obyčejná P/T Petriho síť, ve které:
 - Je možno pracovat s více **typy tokenů** („tokeny různých barev“).
 - Každému místu je přiřazena **třída tokenů** (colour set), která se může v daném místě nacházet (místu nemusí být přiřazen pouze jeden typ tokenů, ale i více).

Barevné Petriho sítě

- Každému přechodu může být přiřazena **podmínka přechodu (strážní podmínka, guard)** tvořená z konstant a proměnných, která po vyhodnocení dává pravdivostní hodnotu (0 nebo 1).
- Každé hraně je přiřazen **hranový výraz (arc expression)** utvořený z konstant a proměnných, který po vyhodnocení představuje multimnožinu tokenů z té třídy, která je přiřazena incidujícímu místu.

Barevné Petriho sítě

- Počáteční značení Barevné Petriho sítě je dáno počátečním značením všech míst sítě – každému místu je přiřazena konkrétní multimnožina tokenů z té třídy, která je místu přiřazena.
-
- pozn. **Multimnožina** je zobecněním pojmu množina. Multimnožina umožňuje výskyt stejného prvku víckrát, např. multimnožina $\{2; 2; 3; 3; 5\}$.

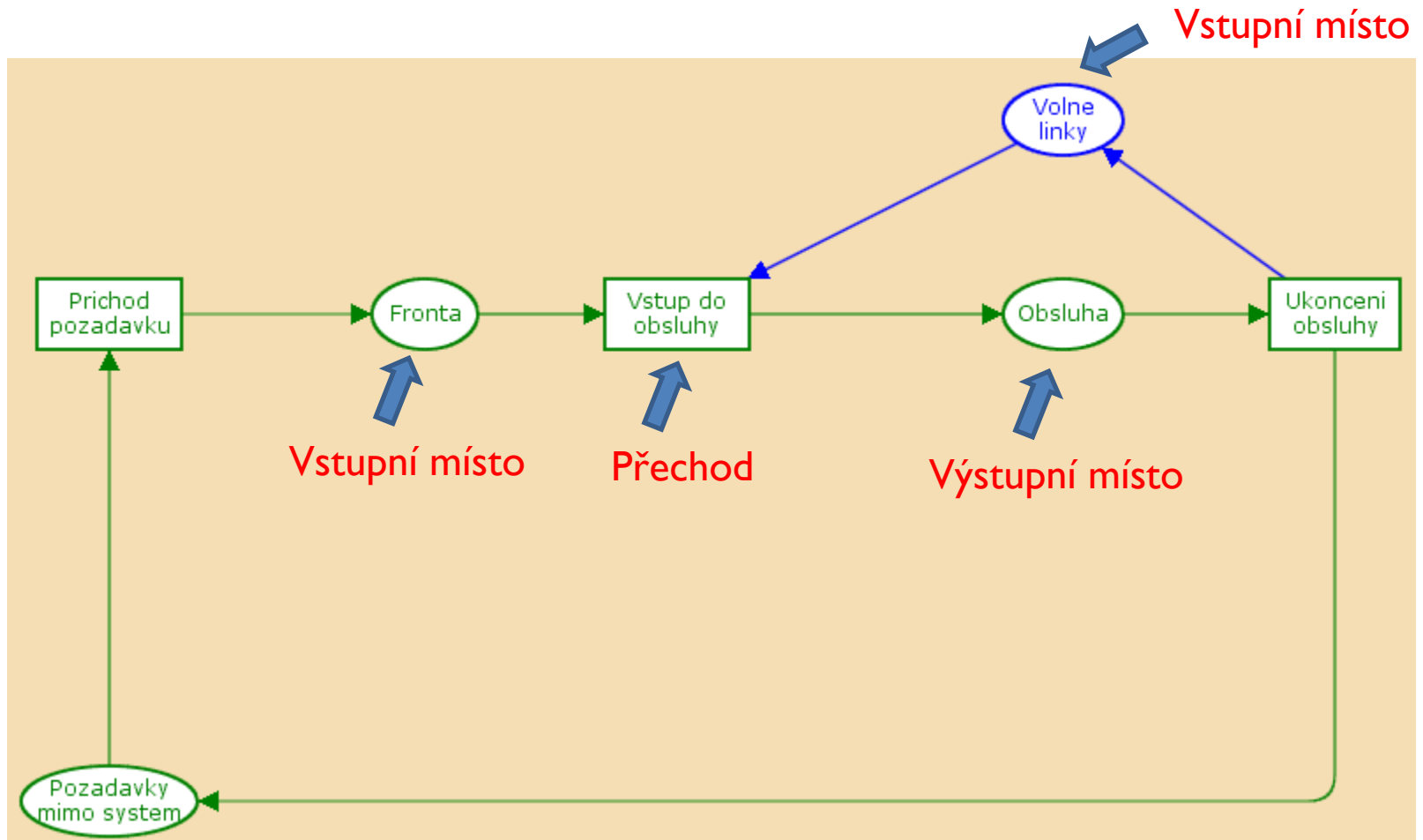
Barevné Petriho sítě v CPN Tools

- Barevná Petriho síť v **CPN Tools** je tvořena ze dvou částí:
 - **Grafické**, která je tvořena grafem Petriho sítě.
 - **Textové**, která je nazývána **inskripcí**.

Barevné Petriho sítě v CPN Tools

- Pro graf platí, že se jedná o bipartitní digraf tvořený:
 - Místy (kreslíme jako kolečka nebo elipsy), množinu všech míst zpravidla označujeme **P**.
 - Přechody (kreslíme jako obdélníky), množinu všech přechodů zpravidla označujeme **T**, přičemž musí platit, že $P \cap T = \{ \}$.
 - Orientovanými hranami, které mohou buď směřovat od míst k přechodům nebo od přechodů k místům.

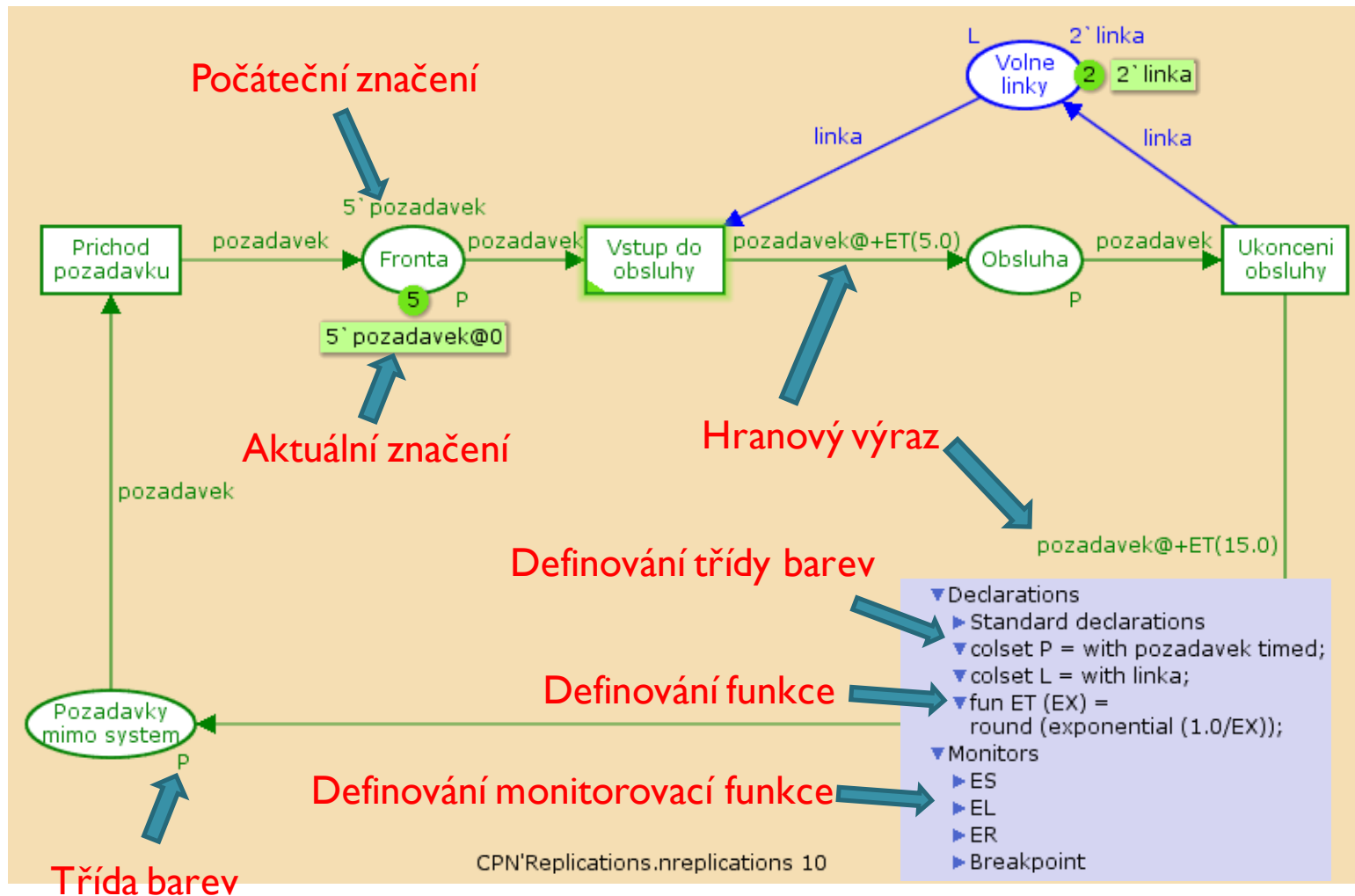
Barevné Petriho síť v CPN Tools



Barevné Petriho sítě v CPN Tools

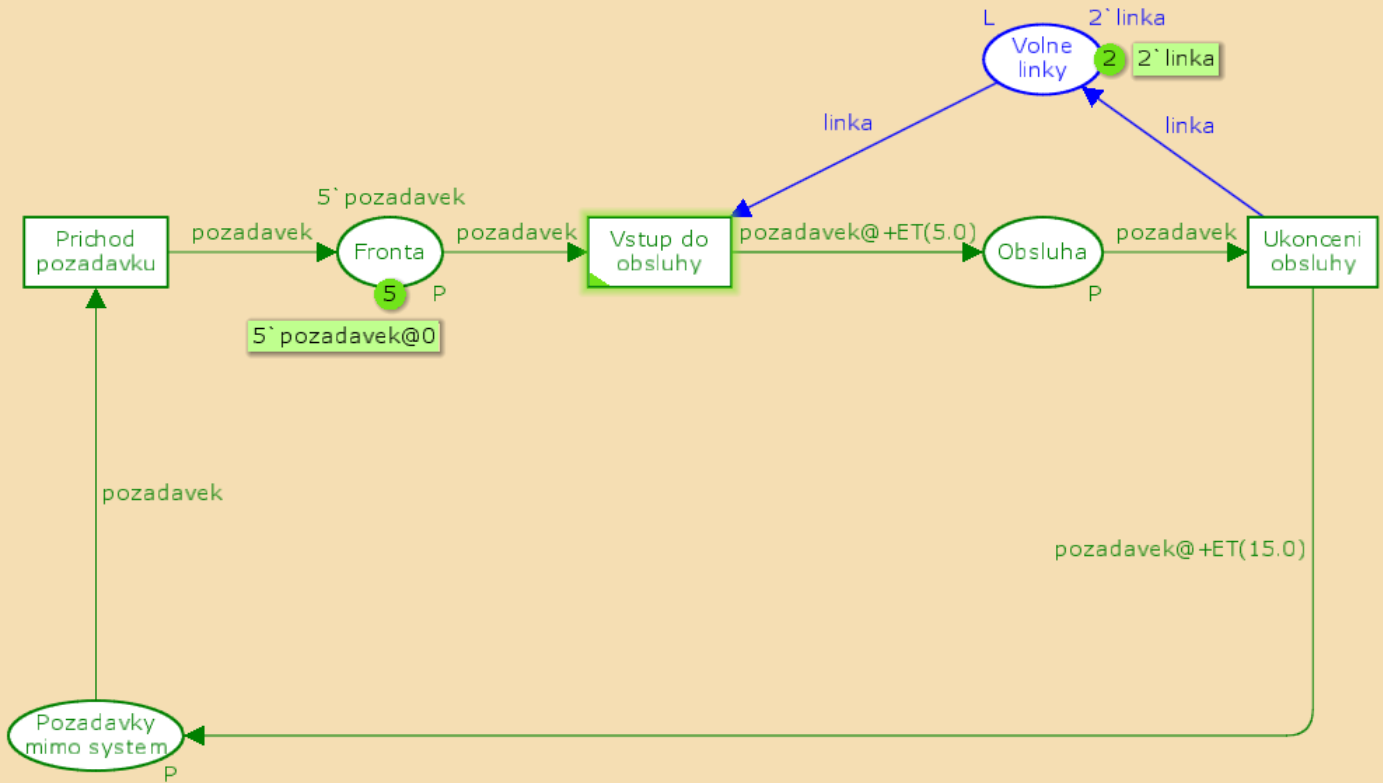
- Inskripční část obsahuje:
 - Deklaraci množin (tříd) barev tokenů.
 - Deklaraci proměnných a funkcí.
 - Deklaraci monitorovacích funkcí.
 - Specifikaci množin barev tokenů přiřazených místům.
 - Hranové výrazy, jež ohodnocují jednotlivé hrany sítě.
 - Počáteční značení, jež definuje rozložení tokenů ve výchozím stavu sítě.
 - Strážní podmínky, které mohou být přiřazeny přechodům, ovlivňující proveditelnost příslušného přechodu.

Barevné Petriho sítě v CPN Tools



- ▼ Tool box
 - Auxiliary
 - Create
 - Hierarchy
 - Monitoring
 - Net
 - Simulation
 - State space
 - Style
 - View
- ▶ Help
- ▶ Options
- ▼ Uzavreny SHO.cpn
 - Step: 0
 - Time: 0
 - ▶ Options
 - ▶ History
 - ▼ Declarations
 - ▶ Standard declarations
 - ▼ colset P = with pozadavek timed;
 - ▼ colset L = with linka;
 - ▼ fun ET (EX) = round (exponential (1.0/EX));
 - ▼ Monitors
 - ▶ ES
 - ▶ EL
 - ▶ ER
 - ▼ Breakpoint
 - Type: Break point
 - Nodes ordered by pages
 - ▼ Predicate
 - fun pred () = IntInf.toInt(time()) > 100000
- [New Page](#)

Binder 0
New Page

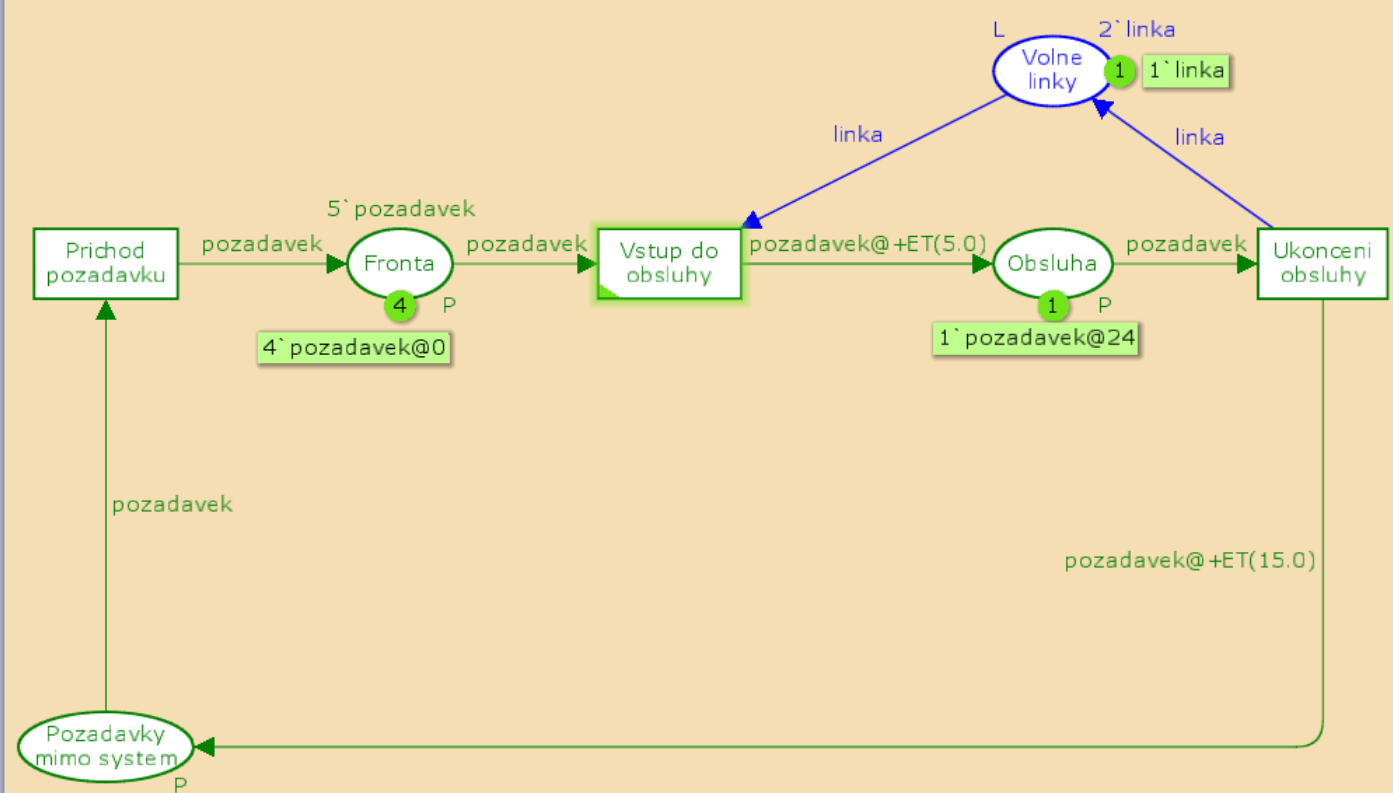


CPN'Replications.nreplications 10

None

- ▼ Tool box
 - Auxiliary
 - Create
 - Hierarchy
 - Monitoring
 - Net
 - Simulation
 - State space
 - Style
 - View
- ▶ Help
- ▶ Options
- ▼ Uzavreny SHO.cpn
 - Step: 1
 - Time: 0
 - ▶ Options
 - ▶ History
 - ▼ Declarations
 - ▶ Standard declarations
 - ▼ colset P = with pozadavek timed;
 - ▼ colset L = with linka;
 - ▼ fun ET (EX) = round (exponential (1.0/EX));
 - ▼ Monitors
 - ▶ ES
 - ▶ EL
 - ▶ ER
 - ▼ Breakpoint
 - Type: Break point
 - Nodes ordered by pages
 - ▼ Predicate
 - fun pred () = IntInf.toInt(time()) > 100000
- [New Page](#)

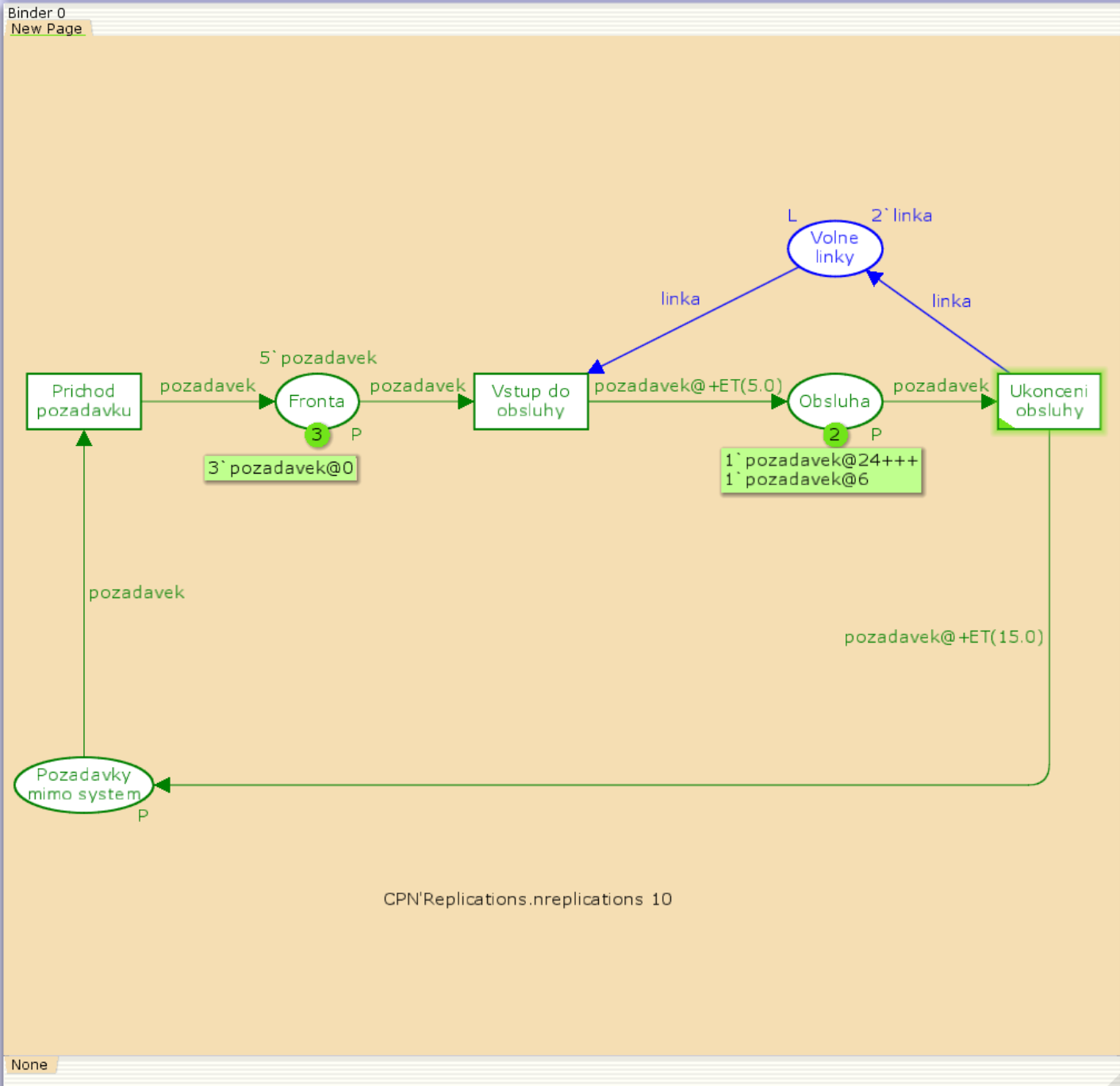
Binder 0
New Page



CPN'Replications.nreplications 10

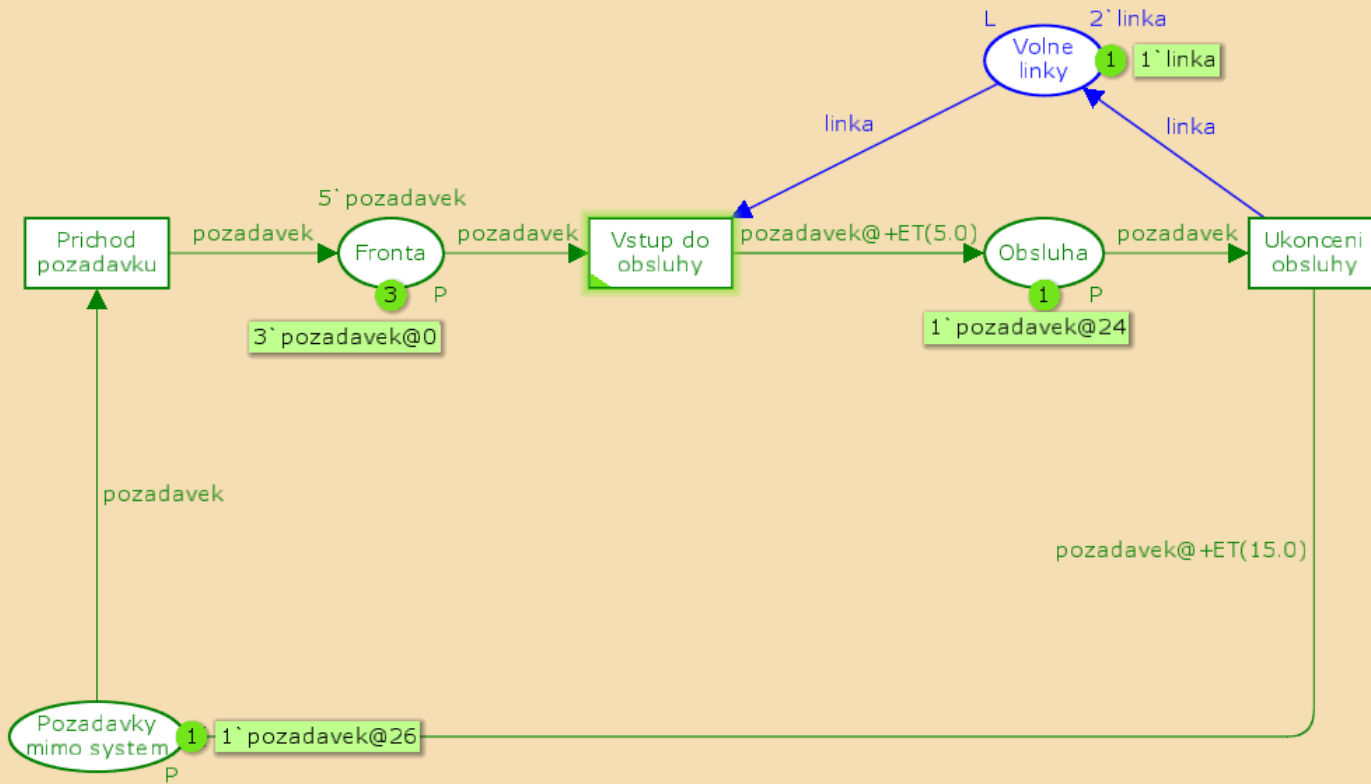
None

- ▼ Tool box
 - Auxiliary
 - Create
 - Hierarchy
 - Monitoring
 - Net
 - Simulation
 - State space
 - Style
 - View
- Help
- Options
- ▼ Uzavreny SHO.cpn
 - Step: 2
 - Time: 6
 - Options
 - History
 - ▼ Declarations
 - Standard declarations
 - ▼ colset P = with pozadavek timed;
 - ▼ colset L = with linka;
 - ▼ fun ET (EX) = round (exponential (1.0/EX));
 - ▼ Monitors
 - ES
 - EL
 - ER
 - ▼ Breakpoint
 - Type: Break point
 - Nodes ordered by pages
 - ▼ Predicate
 - fun pred () = IntInf.toInt(time()) > 100000
- New Page



- ▼ Tool box
 - Auxiliary
 - Create
 - Hierarchy
 - Monitoring
 - Net
 - Simulation
 - State space
 - Style
 - View
- ▶ Help
- ▶ Options
- ▼ Uzavreny SHO.cpn
 - Step: 3
 - Time: 6
 - ▶ Options
 - ▶ History
 - ▼ Declarations
 - ▶ Standard declarations
 - ▼ colset P = with pozadavek timed;
 - ▼ colset L = with linka;
 - ▼ fun ET (EX) = round (exponential (1.0/EX));
 - ▼ Monitors
 - ▶ ES
 - ▶ EL
 - ▶ ER
 - ▼ Breakpoint
 - Type: Break point
 - Nodes ordered by pages
 - ▼ Predicate
 - fun pred () = IntInf.toInt(time()) > 100000
- [New Page](#)

Binder 0
New Page

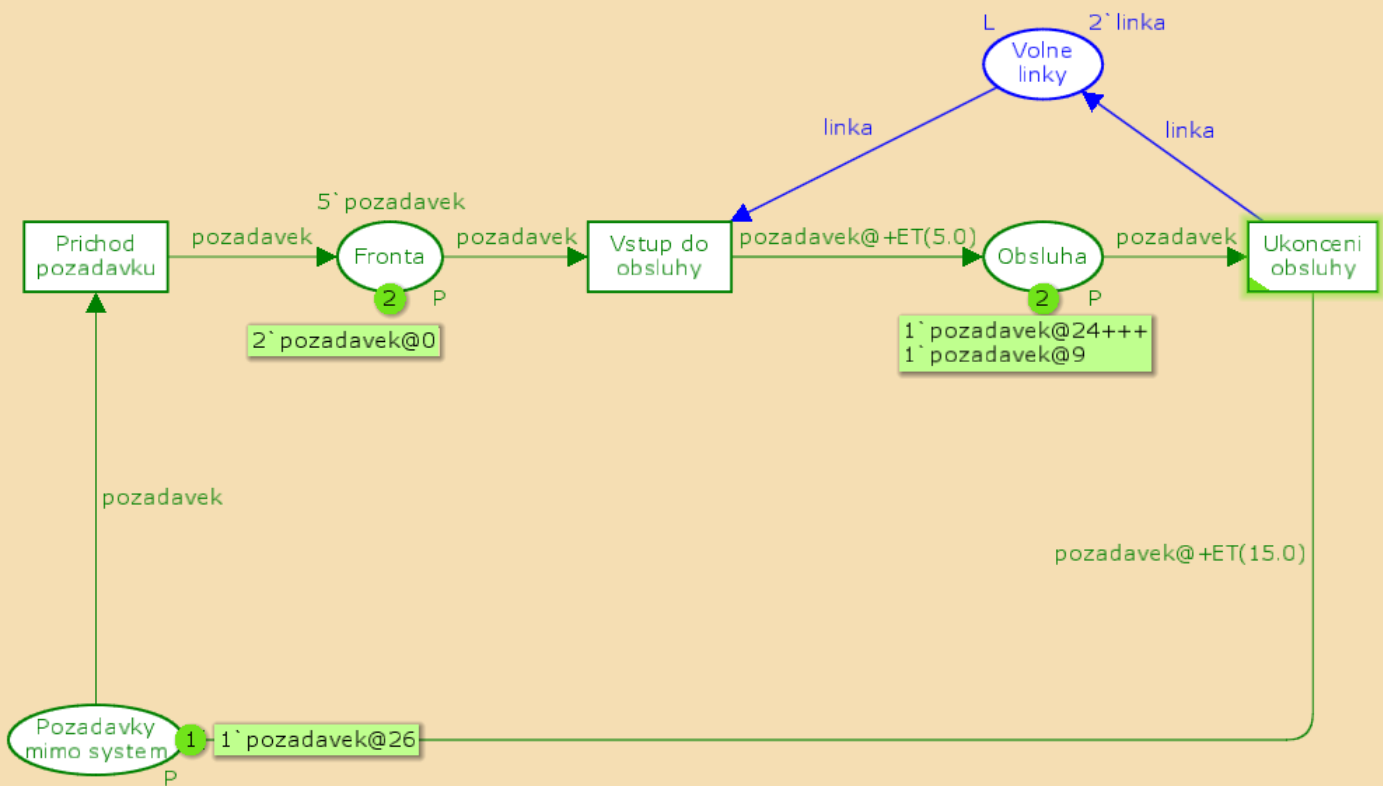


CPN'Replications .nreplications 10

None

- ▼ Tool box
 - Auxiliary
 - Create
 - Hierarchy
 - Monitoring
 - Net
 - Simulation
 - State space
 - Style
 - View
- ▶ Help
- ▶ Options
- ▼ Uzavreny SHO.cpn
 - Step: 4
 - Time: 9
 - ▶ Options
 - ▶ History
 - ▼ Declarations
 - ▶ Standard deklarations
 - ▼ colset P = with pozadavek timed;
 - ▼ colset L = with linka;
 - ▼ fun ET (EX) = round (exponential (1.0/EX));
 - ▼ Monitors
 - ▶ ES
 - ▶ EL
 - ▶ ER
 - ▼ Breakpoint
 - Type: Break point
 - Nodes ordered by pages
 - ▼ Predicate
 - fun pred () = IntInf.toInt(time()) > 100000
 - New Page

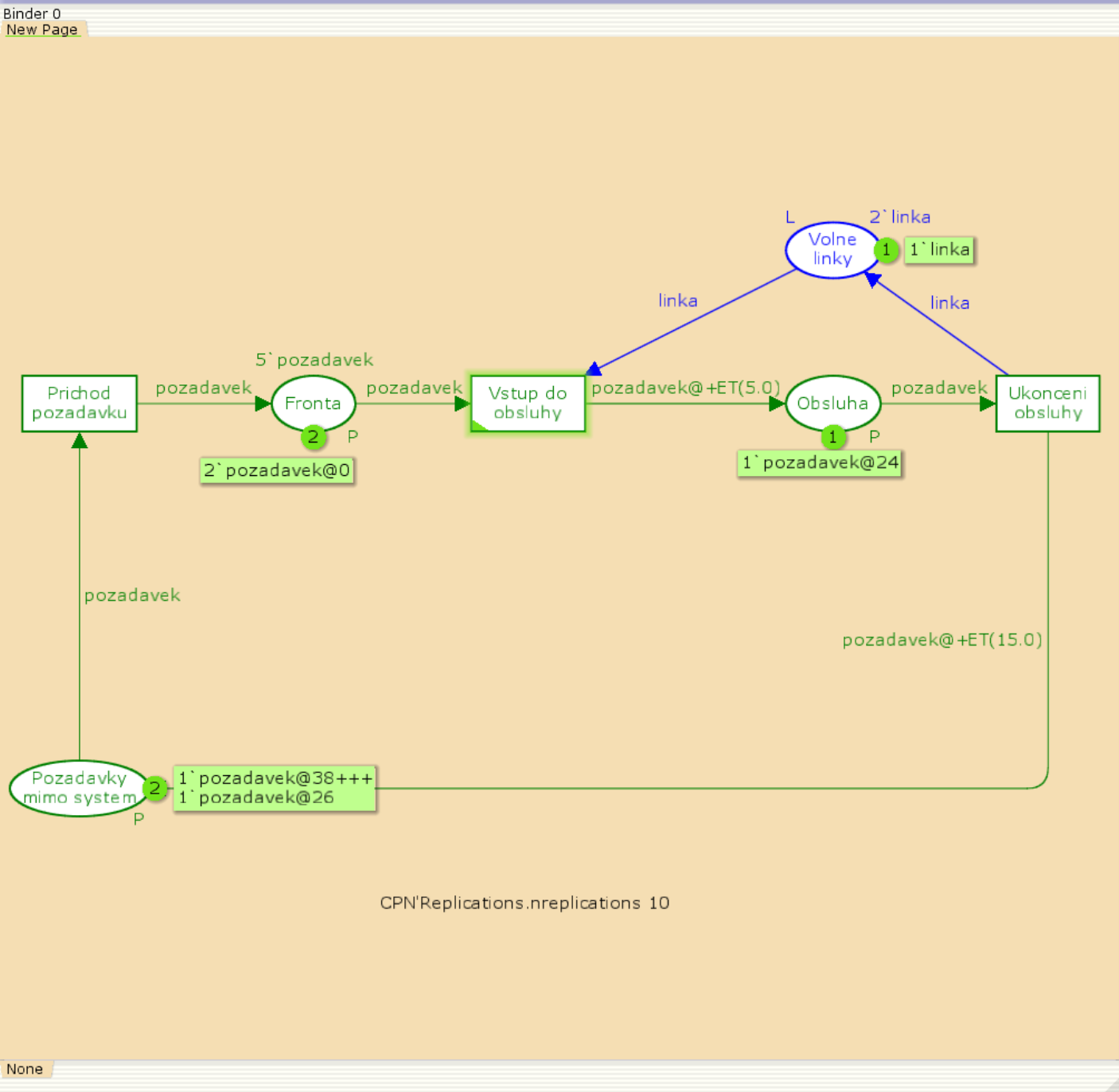
Binder 0
New Page



CPN'Replications.nreplications 10

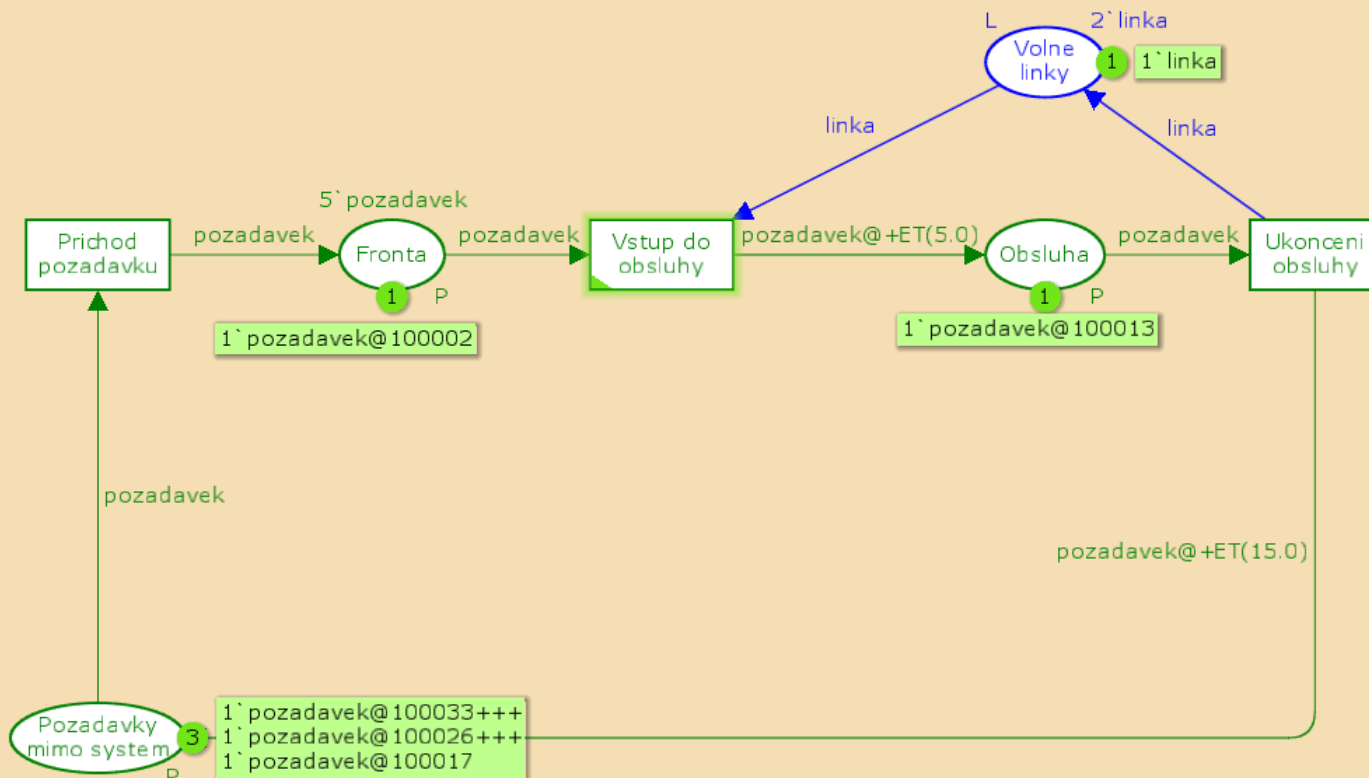
None

- ▼ Tool box
 - Auxiliary
 - Create
 - Hierarchy
 - Monitoring
 - Net
 - Simulation
 - State space
 - Style
 - View
- ▶ Help
- ▶ Options
- ▼ Uzavreny_SHO.cpn
 - Step: 5
 - Time: 9
 - ▶ Options
 - ▶ History
 - ▼ Declarations
 - ▶ Standard declarations
 - ▼ colset P = with pozadavek timed;
 - ▼ colset L = with linka;
 - ▼ fun ET (EX) = round (exponential (1.0/EX));
 - ▼ Monitors
 - ▶ ES
 - ▶ EL
 - ▶ ER
 - ▼ Breakpoint
 - Type: Break point
 - Nodes ordered by pages
 - ▼ Predicate
 - fun pred () = IntInf.toInt(time()) > 100000



- ▼ Tool box
 - Auxiliary
 - Create
 - Hierarchy
 - Monitoring
 - Net
 - Simulation
 - State space
 - Style
 - View
- ▶ Help
- ▶ Options
- ▼ Uzavreny SHO.cpn
 - Step: 71767
 - Time: 100002
 - ▶ Options
 - ▶ History
 - ▼ Declarations
 - ▶ Standard declarations
 - ▼ colset P = with pozadavek timed;
 - ▼ colset L = with linka;
 - ▼ fun ET (EX) = round (exponential (1.0/EX));
 - ▼ Monitors
 - ▶ ES
 - ▶ EL
 - ▶ ER
 - ▼ Breakpoint
 - Type: Break point
 - Nodes ordered by pages
 - ▼ Predicate
 - fun pred () =
 - IntInf.toInt(time()) > 100000

Binder 0
New Page



CPN'Replications.nreplications 10

Barevné Petriho sítě

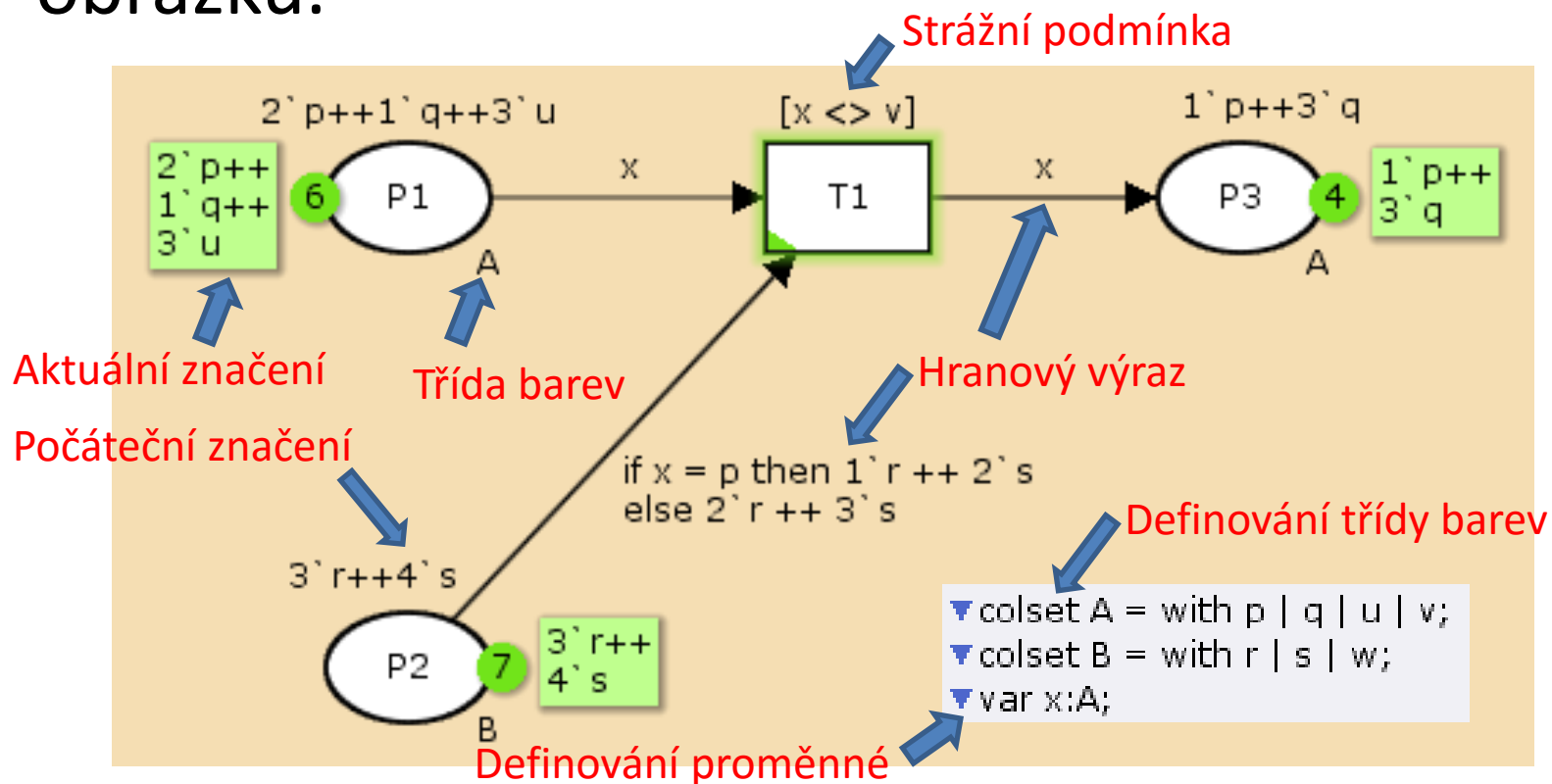
- Přejchod t je v Barevné Petriho síti **proveditelný**, pokud:
 - Multimnožina tokenů obsažená v každém vstupním místě p přechodu t je větší nebo rovna multimnožině, která byla vypočtena při vyhodnocení hranového výrazu příslušícího hraně vedoucí z místa p do přechodu t .
 - Je splněna podmínka přechodu, tj. vyhodnocení strážní podmínky přechodu t dává logickou 1.

Barevné Petriho sítě

- Po **provedení** proveditelného přechodu t je:
 - V každém vstupním místě p odebrána multimnožina tokenů, jež vznikla vyhodnocením hranového výrazu hrany směřující z místa p do přechodu t .
 - V každém výstupním místě p je přidána multimnožina tokenů, jež vznikla vyhodnocením hranového výrazu přiřazeného hraně směřující z přechodu t do místa p .

Barevné Petriho sítě

- Mějme jednoduchou Barevnou síť vytvořenou v nástroji CPN Tools zobrazenou níže na obrázku.



Barevné Petriho sítě

- Síť je tvořena třemi místy P1, P2 a P3 a jedním přechodem T1. Tato síť může představovat fragment modelu systému hromadné obsluhy. V místě P1 čekají požadavky na obsluhu. Uvažujeme 4 typy požadavků (4 barvy tokenů) – p , q , u a v – třída tokenů (colour set) A. Počáteční značení místa P1 jsou 2 požadavky typu r , 1 požadavek typu q a 3 požadavky typu u .

Barevné Petriho sítě

- V místě P2 se nachází materiál potřebný pro obsluhu požadavků. Pro obsluhu slouží 3 druhy materiálu (3 barvy tokenů) – r , s a w zahrnuté v třídě barev B. Počáteční značení místa P2 jsou 3 tokeny barvy r a 4 tokeny barvy s .
- Přejechod T1 může představovat začátek obsluhy požadavku. Proměnná x může nabývat hodnoty z množiny barev A (viz inskripce).

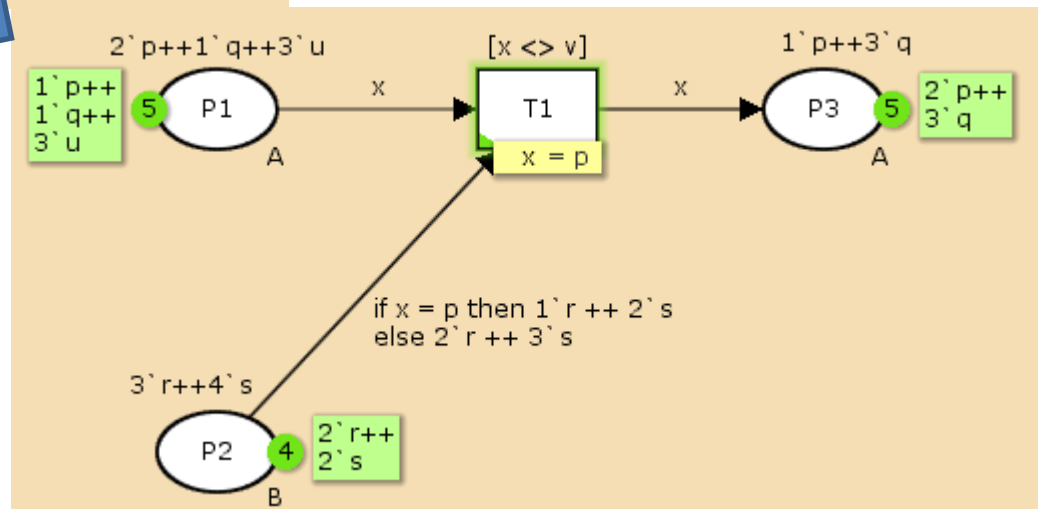
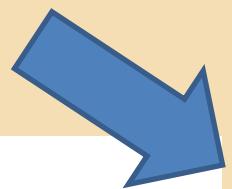
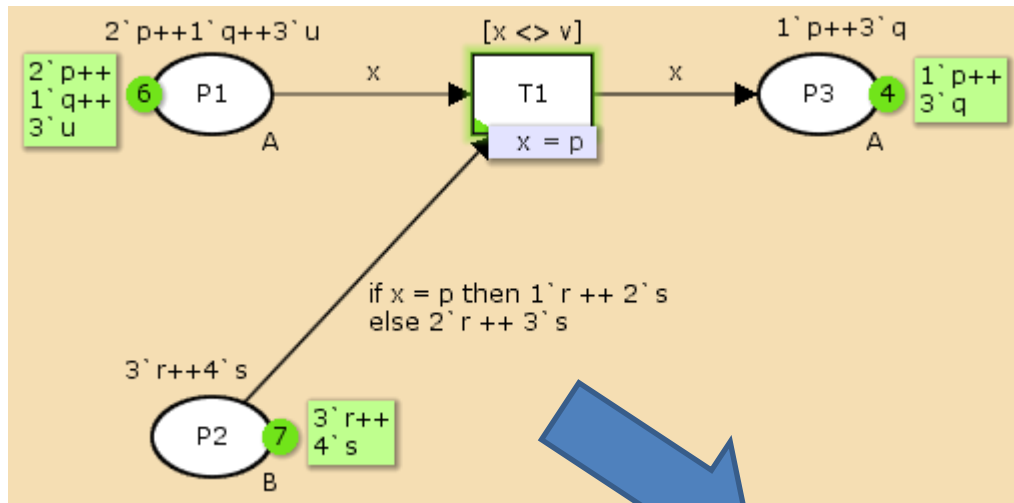
Barevné Petriho sítě

- Hranový výraz přiřazený hraně $[P2;T1]$ nám říká, jaký materiál je třeba pro obsluhu daného požadavku. Je-li $x = p$ (do obsluhy tedy půjde požadavek typu p), potom potřebujeme 1 token r a 2 tokeny s . Pro obsluhu ostatních typů požadavků jsou třeba 2 tokeny r a 3 tokeny s .
- Strážní podmínka přechodu $T1$ nám říká, že přechod je proveditelný pouze tehdy, pokud je x různě od v . Požadavek typu v tedy nebude nikdy obsloužen.

Barevné Petriho sítě

- Místo $P3$ může představovat místo, kam jsou posílány již obsloužené požadavky. Počáteční značení tohoto místa je 1 token p a 3 tokeny q .
- Uvažujme nyní, že $x = p$. V tomto případě je přechod $T1$ proveditelný. Značení sítě po provedení přechodu $T1$ při tomto navázání je znázorněno na následujícím obrázku.

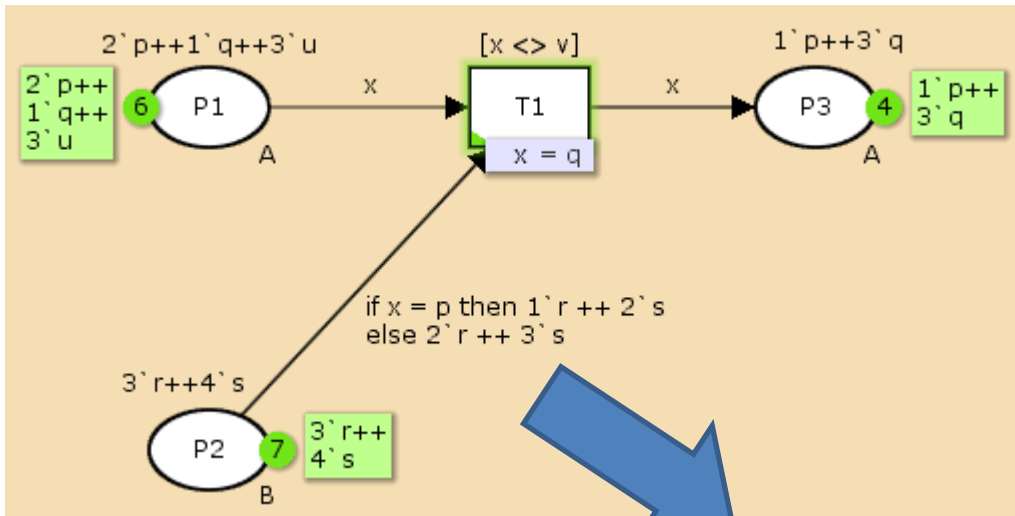
Barevné Petriho sítě



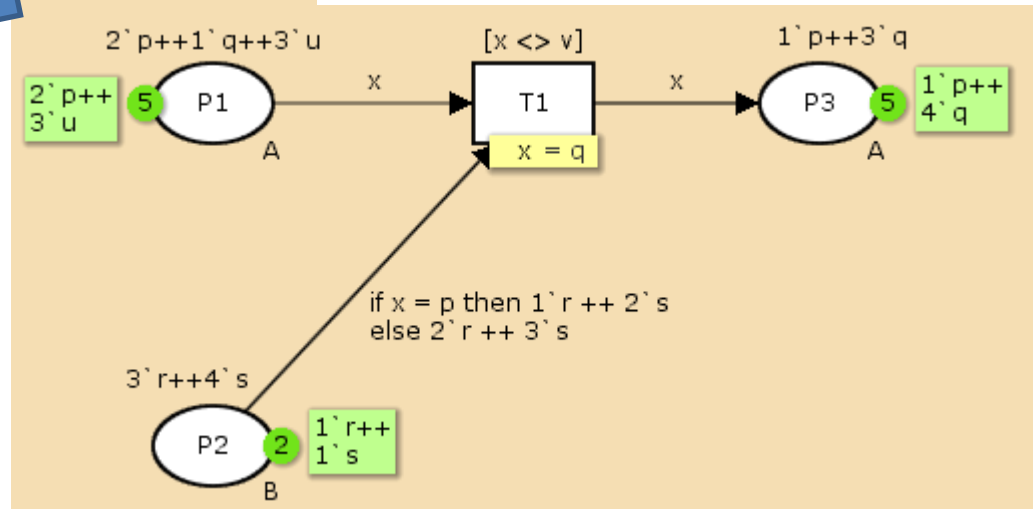
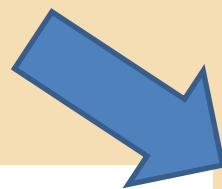
Barevné Petriho sítě

- Přejchod T1 lze provést z počátečního značení i s navázáním $x = q$ nebo s navázáním $x = u$ – viz další obrázky.
- Z počátečního značení (a ani z žádného dalšího značení) není přechod T1 proveditelný s navázáním $x = v$, protože jednak místo P1 neobsahuje žádný token barvy v a rovněž při tomto navázání není splněna strážní podmínka přechodu T1.

Barevné Petriho sítě

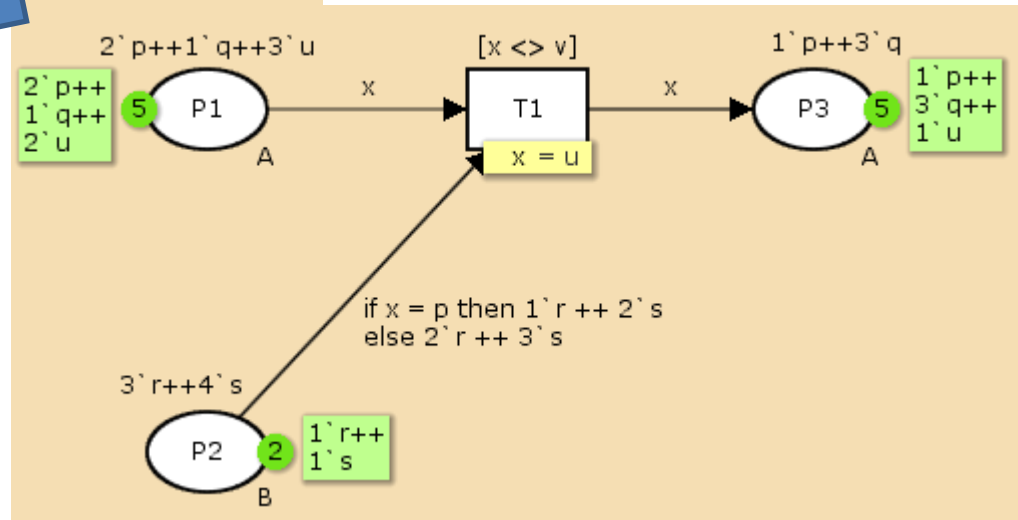
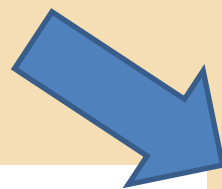
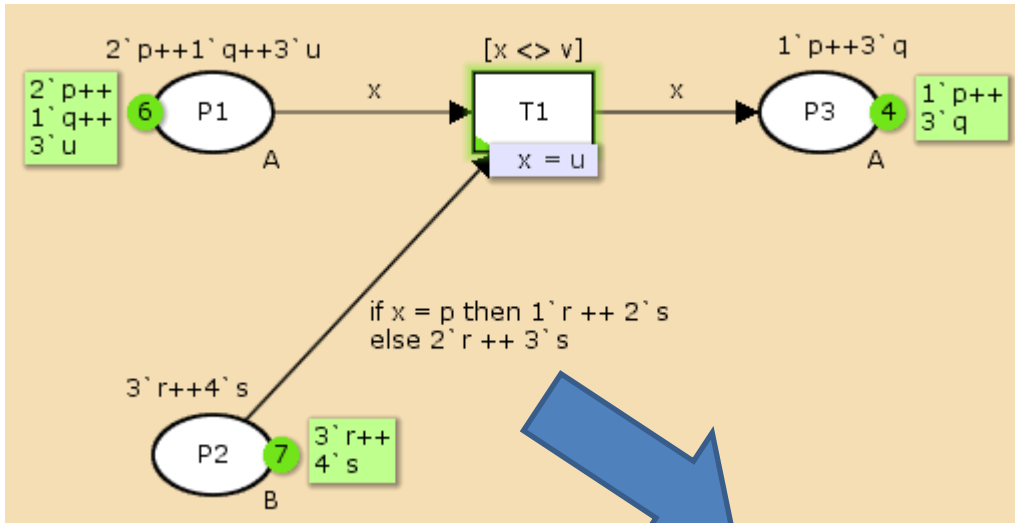


if $x = p$ then $1 \cdot r++$ $2 \cdot s$
 else $2 \cdot r++$ $3 \cdot s$



if $x = p$ then $1 \cdot r++$ $2 \cdot s$
 else $2 \cdot r++$ $3 \cdot s$

Barevné Petriho sítě



Hierarchické Petriho sítě

- Hierarchické Petriho sítě umožňují členit vytvářenou síť na jednotlivé podsítě, které jsou navzájem propojeny.
- Hierarchickou Petriho sítí rozumíme částečně uspořádanou množinu nehierarchických Petriho sítí – tzv. stránek. Stránka B je pod stránkou A, jestliže síť na stránce B rozvíjí některý prvek ze stránky A.

Hierarchické Petriho sítě

- Za tímto účelem se využívají hierarchizační konstrukty:
 - Substituce přechodů – přechod v dané síti je nahrazen substituující sítí.
 - Substituce míst – místo v dané síti je nahrazeno substituující sítí.
 - Volání přechodů.
 - Slučování přechodů.
 - Slučování míst.