Otázka 2 – 4

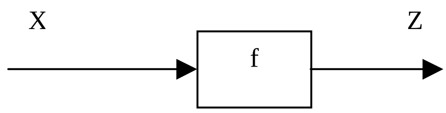
Teorie automatů

Studium chování dynam. systémů s diskrétním parametrem – číslic. počítače, nervové syst., jazyky,….

Cíl: Vytvořit model, jehož matemat. chování aproximuje chování pozorovaného systému.

Konečný automat
Vlastnosti:
1. množiny vstup., výstup. a stavových vektorů jsou konečné
2. čas. množina je zobrazitelná na množinu celých čísel
3. stavově přechodová funkce a výstupní funkce jsou stacionární

Logické systémy - dělení:
1. *Kombinační* – odezva v časovém okamžiku podmíněna výhradně okamžitými hodnotami na vstupech.



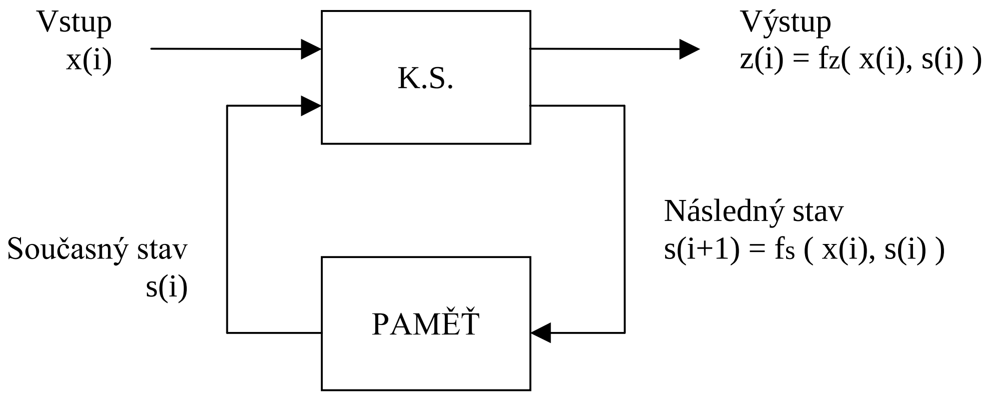
Chování: $f: X \rightarrow Z$
kde X – množina vst. vektorů
Z – množ. výst. vektorů

2. *Sekvenční* – Výst. vektor závisí na vstupech , ale i na sekvenci vstup. vektorů, popř. časových intervalech změn vst. vektorů
   Při stejném vstup. vektoru může generovat jiné výstupní vektory – musí mít paměť (histotie).
   Chování: $F_s : X^* \rightarrow Z^*$

   kde X*, Z* - nekonečné množiny všech posloupností s konečnou délkou vektorů množiny X resp. Z – nazýváme vstup. a výstup. slovo (slovo = sekvence vektorů).

   Funkci sekvenč. syst. Fs lze vyjádřit:
   $s(t+1) = f_S (s(t), x(t))$ násl. stav
   $z(t) = f_Z (s(t), x(t))$ souč. výstup

x(t), z(t), s(t) - vst., výst., stavový vektor systému
s(t+1) - následný stav



Pozn. K.S. = kombinační síť

Boolská algebra: (jen něco, podrobně viz. sylaby přednášek LSY)

Prostředek pro analytický popis chování LS a trfce logických výrazů.

Axiomy: 1.1 = 1 0+0 = 0
atd. (to každý zná…)

Zákony: a+a=a a.a=a tautologie
a+b=b+a a.b=b.a komutativní z.
atd… (to taky každý zná…)

Odvozené zákony:

$a + \bar{a}.b = a + b$ $a . (\bar{a}+b) = a.b$ z absorbce negace

$\overline{(a+b)} = \bar{a} . \bar{b}$ $\overline{ab} = \bar{a} + \bar{b}$ De Morganovy zák.

De Morganovy zák.:

$\overline{a.b.c\ldots} = \bar{a} + \bar{b} + \bar{c} + \ldots$ $\overline{a+b+c+\ldots} = \bar{a} . \bar{b} . \bar{c} \ldots$

Formy popisu logické fce

1. *Pravdivostní tabulka* (to zná taky každý)

2. Logický výraz

Term : součinový – boolský operátor, který neobsahuje +
součtový - boolský výraz, který neobsahuje **.**
Minterm – součinový tvar, který obsahuje všechny proměnné v přímé nebo negované formě

Maxterm – součtový tvar, který obsahuje všechny proměnné v přímé nebo negované formě
Každý minterm nebo maxterm nabývá log. hodnoty 0 a 1 právě pro jeden vstup. vektor.

Normální disjunktivní forma – součet mintermů
Normální konjunktivní forma – součin maxtermů

3. Seznam indexů vstup. vektorů

Index – dekad. hodnota binárně zakódovaného vstup. vektoru
Seznam indexů – seznam vst. vektorů, pro něž je fce =1
- představuje zjednoduš. zápis log. fce

4. Mapa log. funkce

Svobodova (binární), Karnaughova (gray-kód)

Syntéza kombinačních obvodů

Postup:
1. Ze zadání určit počet vstupů, výstupů, log. funkcí
2. Ze zadání pravdivostní tab. resp. mapa
3. Minimalizovat funkce. Algebraický tvar.
4. Převod na tvar vhodný pro realizační soubor, optimaliz.
5. Nakreslit log. síť + úprava + optimalizace.
6. Analýza logické sítě. Hazardy.
7. Ověření činnosti.

Hazardy v kombinačních log. obvodech

Hazard – přechodová chyba v důsledku konečné doby reakce použitých realizačních prvků.

1. Funkční hazardy – hazard nevhodné změny vstupů

   -přechodové stavy vznikají na základě reálného průběhu změn vstupních proměnných
   - správné odezvy na nesprávné vstup. vektory
   - neodstranitelné pomocnými obvody

Východisko :
- Filtr na výstupu
- Zákaz vícenásobných změn
- Synchronizace kombinačního obvodu

2. Logické hazardy – hazard navržené struktury s reálnými prvky

   Vznik – konečná doba reakce realizačních prvků

   a) Statický hazard – při zachování úrovně

Může existovat hazard v 0 nebo v 1. Dočasná změna výstupu v důsledku různě dlouhých zpoždění různě dlouhých větví sítě.

b) Dynamický hazard – při změně výstupu
Při fundamentální změně místo jednoduchého přechodu $0 \rightarrow 1$ (resp. $1 \rightarrow 0$) vznikne posloupnost změn např. $0 \rightarrow 1 \rightarrow 1 \rightarrow 0$.
Důvod: vzájemná časová zpoždění ve větvích kombinač. obvodu.

!!! Existuje-li statický hazard v části obvodu ⇒ existuje i dynamický !!!

Konečný automat jako matemat. model sekvenčního systému.

Chování sekvenčního systému lze popsat algebraickým systémem, který představuje automat (sekvenční stroj). Je to model chování – ne struktury.

Konečný automat – uspořádaná šestice $M = \langle X, S, Z, \delta, \omega, s0 \rangle$

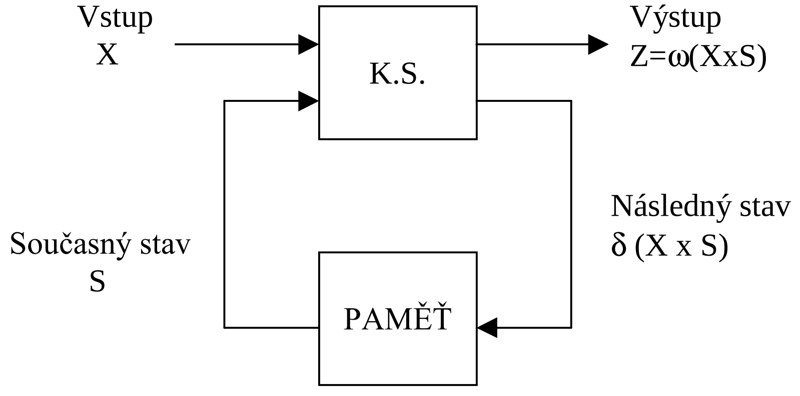
X – konečná množina všech vstup. vektorů
Z – konečná množina všech výstup. vektorů
S – konečná množina všech vnitřních stavů
δ - přechodová funkce – zobrazení $\delta : X \times S \rightarrow S$
ω - výstupní funkce - zobrazení $\omega : X \times S \rightarrow Z$
s0 – počáteční stav $s0 \in S$

Přechodová funkce: $s(i+1) = \delta(x(i), s(i))$

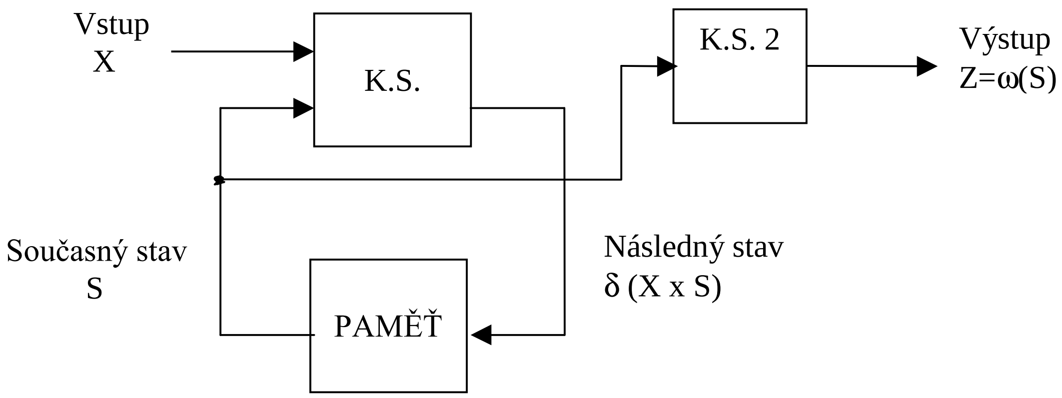
Výstupní funkce: - dvě varianty

1. $z(i) = \omega(x(i), s(i))$ Mealy automat
2. $z(i) = \omega(s(i))$ Moore automat

M E A L Y :



M O O R E :



Formy popisu chování automatu :

1. Tabulka přechodů

2 části – příští stav (δ), výstupy (ω)
sloupce – všechny vstup. vektory X
řádky – všechny vnitřní stavy S
Průsečík řádku a sloupce: následný stav a současný výstup

2. Graf přechodů

Popis chování automatu orientovaným grafem nahrazuje tabulku přechodů.

Uzly – vnitřní stavy
Orientované hrany – možné přechody

3. Matice přechodů

Používá se při velkém počtu stavů – přehlednost .
Při p stavech – čtvercová matice p x p.

řádky – výchozí stav
sloupce – následný stav
průsečík – podmínky přechodu

<u>Sekvenční zobrazení</u>

$Z^* = F_s (X^*)$

X^* - sekvence vstup. vektorů (vstupní slovo)
Z^* - sekvence výstup. vektorů (výstupní slovo)

Vlastnosti:

1. Zachovává délku slova
2. Je závislé na počátečním stavu před X^*

<u>Mealy</u>: Výstup pouze když existuje vstup. vektor.
<u>Moore</u>: Výstup nezávisí na vstupu.

Výstupní sekvence MOORE automatu je o 1 takt zpožděna. První výstup je „falešný“ a odpovídá počátečním podmínkám!!!

Minimalizace množiny vnitřních stavů automatu
redukce nadbytečných stavů – zjednodušení syntézy i realizace

<u>Ekvivalence stavů</u> – stavy s1 a s2 jsou ekvivalentní, jestliže $\omega(X^*, s1) = \omega(X^*, s2)$ pro všechny možné sekvence vst. vektorů X^*.

Jednoduchá ekvivalence - $\omega(x(i), s(i)) = \omega(x(i), s(j))$
$\delta(x(i), s(i)) = \delta(x(i), s(j))$
tj. shodné řádky v přechodové tabulce

Režimy činnosti automatu

1. <u>Asynchronní režim</u> - Přechody mezi stavy inicializují změny vstupních a stavových proměnných ⇒ definují diskrétní čas.
   Fundamentální režim – nová změna vstupních proměnných může nastat až po ustálení stavů po předešlé změně ⇒ systém <u>musí</u> přecházet do stabilních stavů.

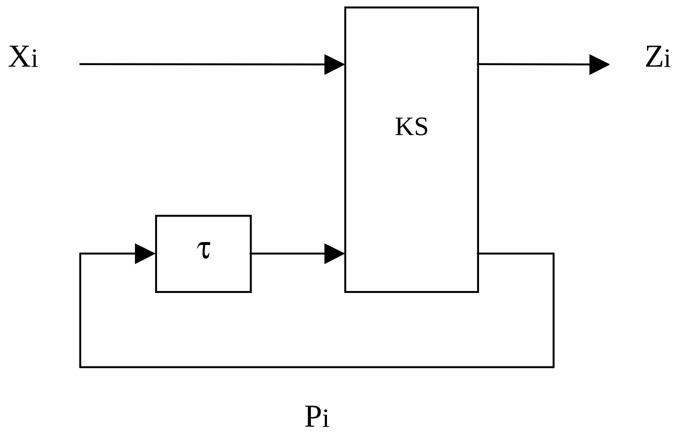
2. <u>Impulsní režim</u> - Přechody mezi stavy jsou inicializované významnou změnou hodnoty vstupní proměnné (hrany, krátké pulsy).
3. <u>Synchronní režim</u> – Celá činnost automatu řízena externím generátorem hodinových impulsů, které definují diskrétní čas.

Syntéza sekvenčních logických obvodů

V praktické syntéze vycházíme z koncepce Moore automatu a v průběhu syntézy je možné přejít na Mealy.

Univerzální postup syntézy:
1. Ze zadání časový diagram základní sekvence
2. Tabulka přechodů, graf přechodů
3. Minimalizace množiny vnitřních stvů
4. Volba vnitřního kódu
5. Odvození map vnitřní funkce (δ zobrazení) a výstupní funkce (ω zobrazení)
6. Volba realizačních prvků (klopných obvodů)
7. Analytické vyjádření vnitřních a výstupních proměnných
8. Test na hazardy a jejich odstranění
9. Konstrukce sekvenčního logického obvodu

Syntéza asynchronních sekvenčních logických obvodů



Paměť tvoří zpětná vazba τ - ustálení přechdového ději, filtr hazardů, zajištění fundamentálního režimu.

Fundamentální režim :
1. vstupní vektor se liší pouze v jedné proměnné
2. změna vstupního vektoru pouze v ustáleném stavu
3. každý přechod končí v ustáleném stavu
4. změna pouze jedné vnitřní proměnné

Kódování vnitřních stavů

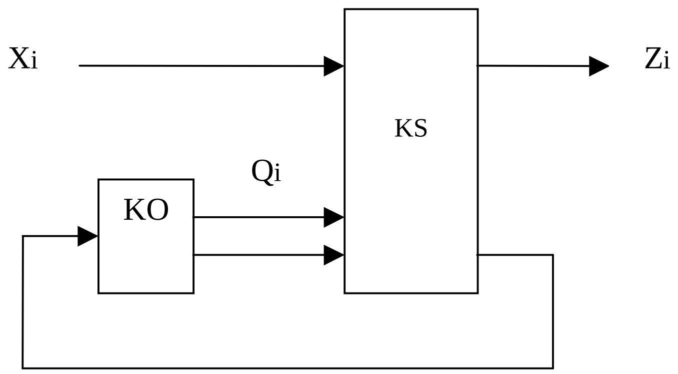
- přiřazení kódu vnitřních proměnných vnitřním stavům sekvenčního obvodu

Sousednost stavů – mezi stavy je definován přechod

Správné kódování – zajistí přiřazení sousedním stavům kód, který se mění pouze v jedné proměnné. !!!

Mapa vnitřní funkce – náhrada výchozích i následných stavů v tabulce přechodů kódovými kombinacemi vnitřních proměnných
pro i- vnitřních proměnných   i- map vnitřní funkce

<u>Syntéza sekvenčních logických obvodů s použitím paměťových členů</u>



S_i, R_i, T_i
Budící proměnné KO

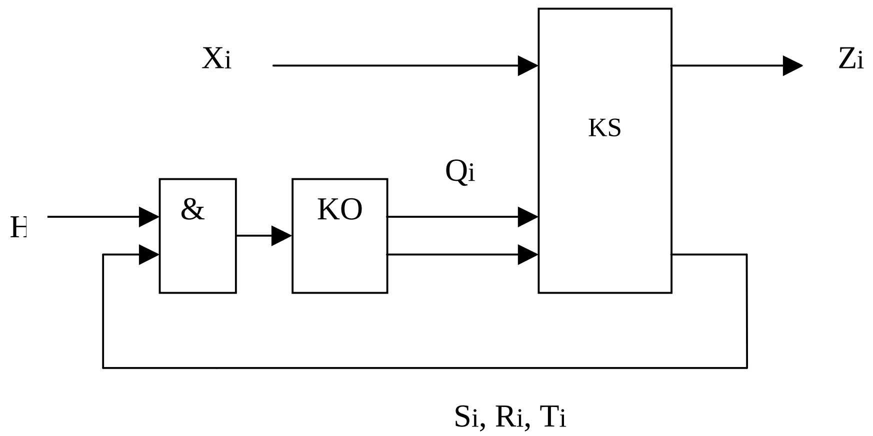
Upravit mapu vnitřní funkce na univerzální tvar vhodný pro čtení budících funkcí podle slovníku KO. Při změně hodnoty se zapisují silné **0** nebo **1**, když hodnota zůstává nechávají se slabé 0 nebo 1.

Přehled pravidel:

Typ KO	vstup	každá	některá
SR	S R	**1** **0**	1,- 0,-
SR-JK	S(J) R(K)	**1** **0**	**0**, 1, - **1**, 0, -
T	T	**1, 0**	-
D	D	**1,** 1	-

Pravidla platí pro asynchronní, synchronní i synchronní hradlové KO.

Syntéza synchronních sekvenčních logických obvodů



Odlišnosti syntézy:
1. libovolná volba vnitřního kódu
2. necitlivost na hazardy (eliminuje všechny kromě funkčního)
3. vstup. proměnné by měly být synchronizovány
4. u asynchronních vstupů problém s dodržením předstihu s přesahu vůči hodinám
5. oscilace a cykly jsou povoleny a přechod mezi stavy určují hodiny

Obvody TTL

U_{ILmax} = 0,8V U_{OLmax} = 0,4V
U_{IHmin} = 2V U_{OHmin} = 2,4V

I_{IL} = -1,6 mA I_{IH} = 40 μA

Logický zisk N – číslo, které udává, kolik elementárních vstupů dalších obvodů může být na výstup daného obvod připojeno při zaručení výrobcem definovaných logických úrovní.
Garantovaná šumová imunita – je definovaná jako rozdíl nehorších mezních hodnot vstupů a výstupů zaručených výrobcem logických obvodů.

Více viz.: Bayer, Šimek : ESY2
Bayer : LSY – sylaby přednášek

Obvody PLD (Programmable Logic Device)

Univerzální logické obvody - funkce programovaná uživatelem
- rozložení vývodů pouzdra měnitelné

Dělení PLD :
1. PROM – progr. paměti
2. PAL – progr. log. pole
3. FPLA – plně progr. log. pole

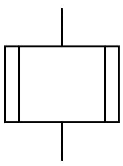
Význam PLD :
1. Redukce počtu obvodů, ploš. spoj, cena
2. Flexibilita – změna funkce
3. Zvětšení spolehlivosti
4. Optimalizace ploš. spoje – změna topologie
5. Kratší doba vývoje
6. Programování pomocí symbolických jazyků

Obvody GAL (Generic Array Logic)

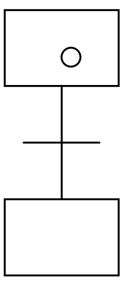
GRAFCET

- grafický nástroj pro popis sekvenčního logického řízení a systémů diskrétních událostí a pro přípravu vyjádření algoritmů řídících systémů.

Dva základní prvky: krok a přechod
Stav systému reprezentován množinou kroků, z nichž každý určitou měrou přispívá do stavového popisu.
Krok může být aktivní a neaktivní. Aktivita je znázorněna značkou (tečka).
Přechod reprezentuje v daném stavu systému možnost měnit svůj stav v závislosti na podmínkách přechodu.



Kroky, které jsou aktivní na začátku řídícího procesu (popisují počáteční stav systému)

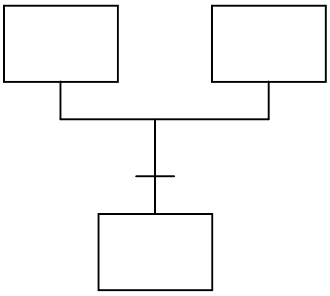


Aktivní krok

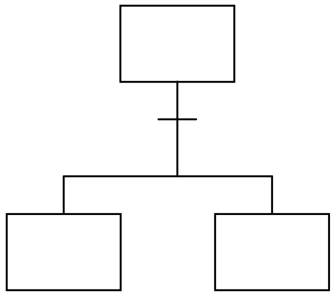
Přechod

Neaktivní krok

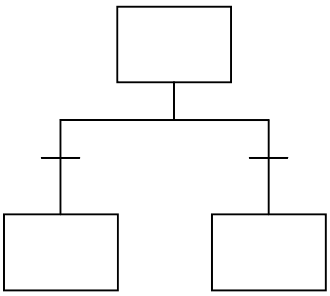
Paralelismus a synchronizace



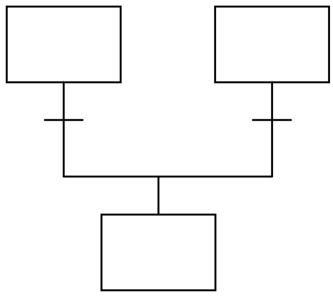
Konvergentní AND



Divergentní AND



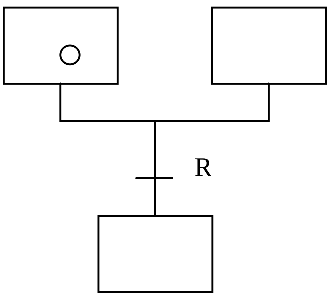
Divergentní OR



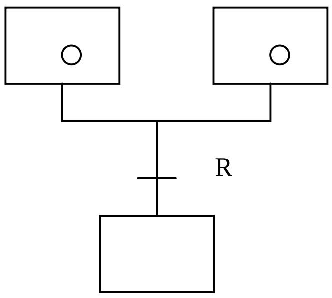
Konvergentní OR

Pravidla:

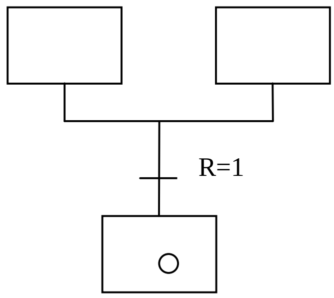
1. počáteční stav je charakterizován počátečními kroky, které jsou definovány jako aktivní na počátku procesu.
2. Každý přechod se může vyskytovat ve dvou stavech: uvolněn nebo neuvolněn. Přechod je uvolněn, když všechny jeho vstupní kroky jsou aktivní. Přechod je aktivován, jestliže je uvolněn a podmínka jeho přechodu je „TRUE“.
3. Aktivování přechodu zároveň uvede do aktivního stavu bezprostředně následující krok(y) a deaktivuje krok(y) bezprostředně předcházející.



Neuvolněný přechod



Uvolněný přechod



Aktivovaný přechod

4. Dvojitou čarou značíme přechody, které mají být aktivovány současně.

Akce (příkazy) spojené s kroky

Příkaz bez paměti (probíhající akce) – vykonává se do té doby, pokud krok s ním spojený je aktivní.
Zpožděný příkaz bez paměti – je vykonáván až po uplynutí zpoždění od aktivace s ním spojeného kroku a do té doby, pokud je krok aktivní.
Časově limitovaný příkaz bez paměti (impulsová reakce) – je vykonáván, jakmile krok s ním spojený je aktivní a pokud trvá zadaný čas od aktivace kroku.
Podmíněný příkaz bez paměti (podmíněná akce) – vykonává se pouze, když je krok aktivní a s ním spojená podmínka je „TRUE".
Akce s pamětí – spojena se dvěma kroky, začíná s náběžnou hranou startovacího kroku a končí s náběžnou hranou ukončovacího kroku.
… jsou ještě další varianty, podrobněji viz. sylaby přednášek LSY (Bayer).

Podmínky přechodu a jejich vyjádření

Podmínka přechodu může být vyjádřena ve 4 formách:
- slovní vyjádření
- boolský výraz
- grafické vyjádření
- časově závislé vyjádření

Výsledek podmínky je vždy boolská proměnná – nabývá hodnot „1" TRUE nebo „0" FALSE.

Aktivace přechodu nemusí záviset jen na log. hodnotě podmínky, ale může záviset i na její změně ($0 \rightarrow 1$ nebo $1 \rightarrow 0$).

Petriho sítě (PS)

Grafický a matematický nástroj promodelování systémů diskrétních událostí (DES).

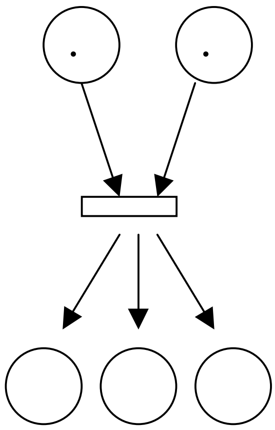
1. Matematický popis:
   Jednorozměrná PS – uspořádaná čtveřice A= <Pre, Post, P, T >
   P – konečná množina míst
   T – konečná množina přechodů
   Pre – matice PxT preconditions – hrany vedoucí z míst do přechodu
   Post – matice PxT postconditions – hrany vedoucí s přechodu do místa
   C = Post – Pre …… incidenční matice

   Označená PS = neoznačená PS + počáteční značení Mo
   M(P1) = počet tokenů v místě P1

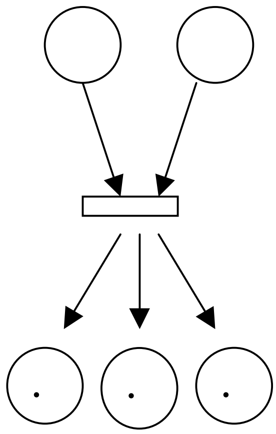
2. Grafický popis
   Bipartitní orientovaný ohodnocený graf.

Vývoj systému je reprezentován přesunem tokenů v síti.
Každé nové značení reprezentuje nový stav systému

<u>Aktivace přechodu</u>
Přechod t je uvolněn, když ve všech jeho vstupních místech je počet tokenů ≥ váze hrany jdoucí z místa do přechodu.
Aktivace odejme ve všech vstupních místech přechodu a vloží tokeny do šech výstupních míst přechodu.



Před aktivací



Po aktivaci

<u>Ohraničená PS :</u>
Pro dané počáteční značení M0 je PS ohraničená, když pro jakékoliv dosažitelné značení Ma pro každý uzel Pi je počet tokenů ≤ k (tzn. konečný počet stavů – lze vytvořit stavový diagram).

<u>Binární PS:</u>
Každé místo obsahuje nejvýše jednu značku.

<u>Pseudoživá PS:</u>
Jestliže všechny přechody jsou pseudoživé.
Přechod je pseudoživý, když z počát. značení M0 existuje sekvence přeskoků, která vede k uvolnění přechodu. (= každý přechod může být alespoň jednou přeskočen)

<u>Živá PS:</u>
Jestliže všechny přechody jsou živé.
Přechod je živý, když po každé změně dosažitelné z M0, existuje sekvence přeskoků, která uvolní daný přechod.

<u>Reverzibilní PS:</u>
PS je reverzibilní, když z libovolného dosažitelného stavu existuje sekvence přeskoků zpět do M0.

Strukturální konflikt

Dva přechody mají SK, jestliže existuje místo, které je vstupním místem obou těchto přechodů.

Efektivní konflikt

Dva přechody mají EK, jestliže:

- mají strukturální konflikt
- značení jejich vstup. míst ≥ váze hran jdoucích z místa do přechodu

Podrobněji viz. Bayer: Sylaby LSY
Přednášky DRS. (Hanzálek)