

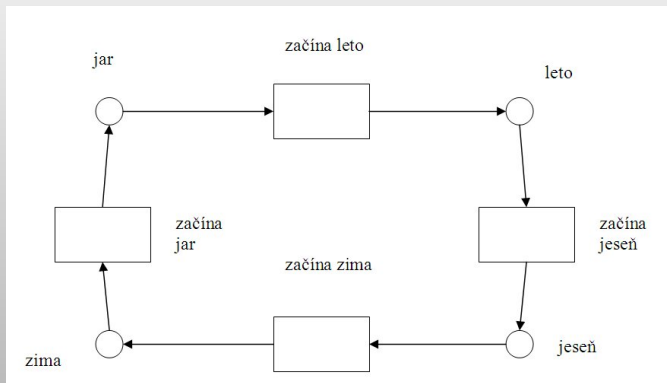
Modely konkurentných systémov

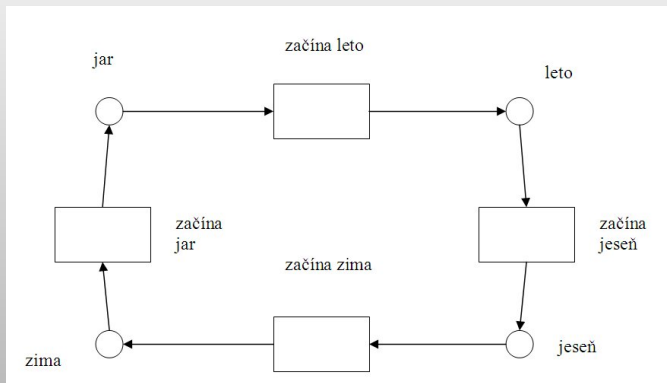
Formálne metódy tvorby softvéru

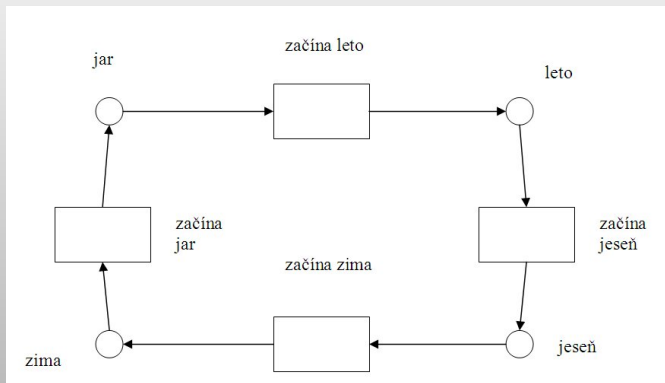
Damas Gruska

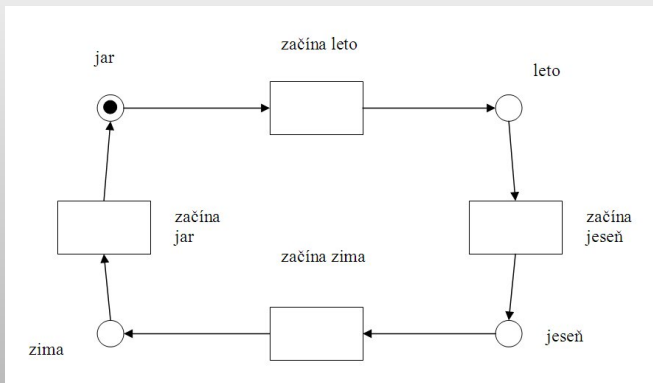
Katedra aplikovanej informatiky, I20, gruska@fmph.uniba.sk

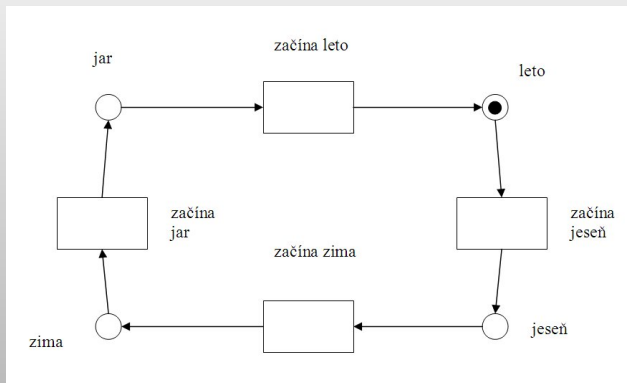
Prednáška 7.

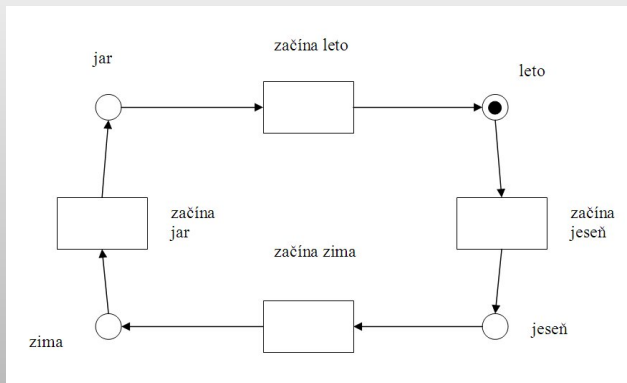


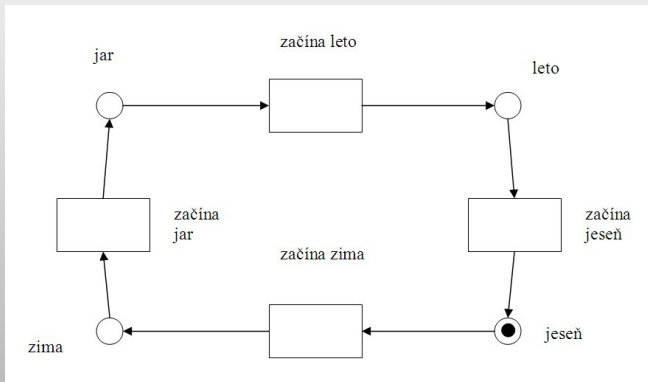


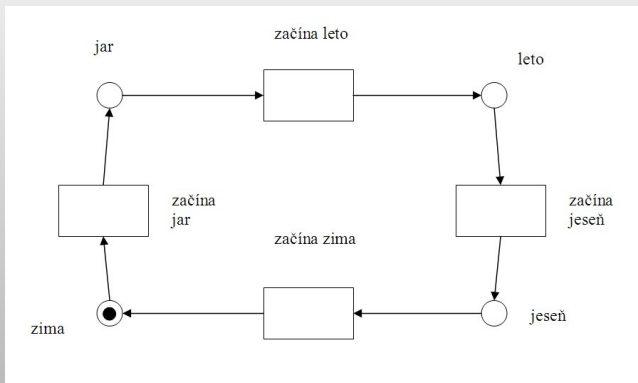


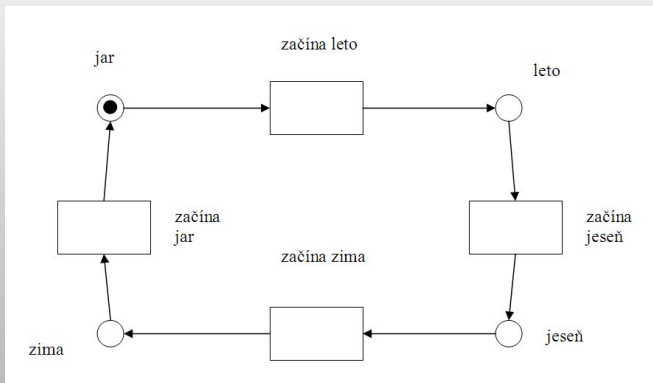


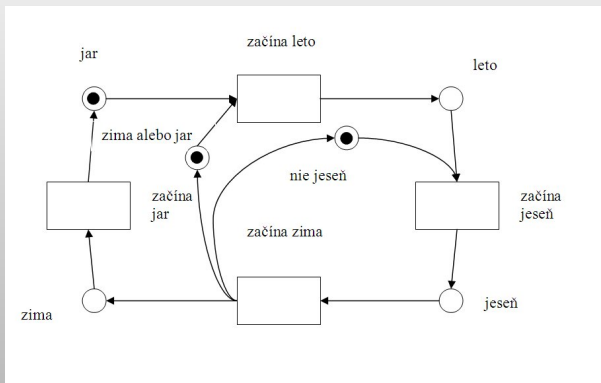




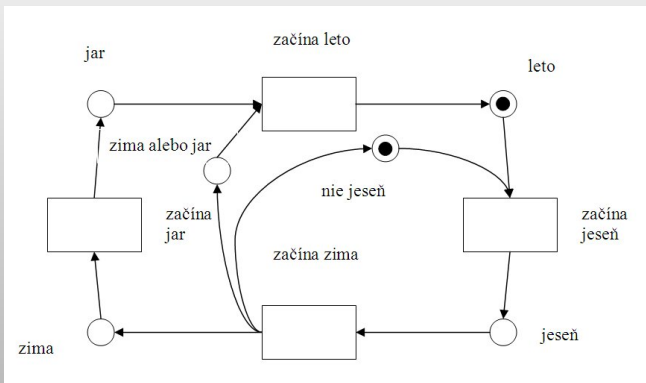




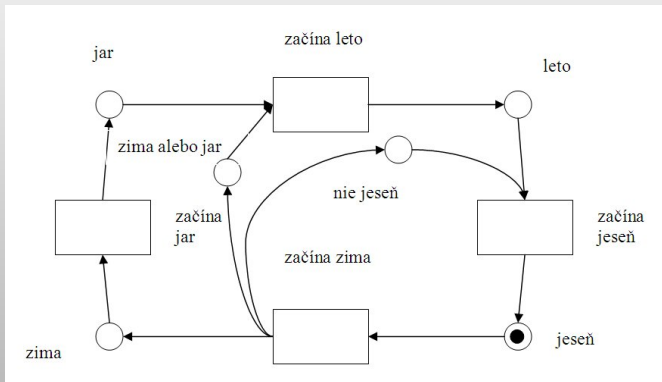


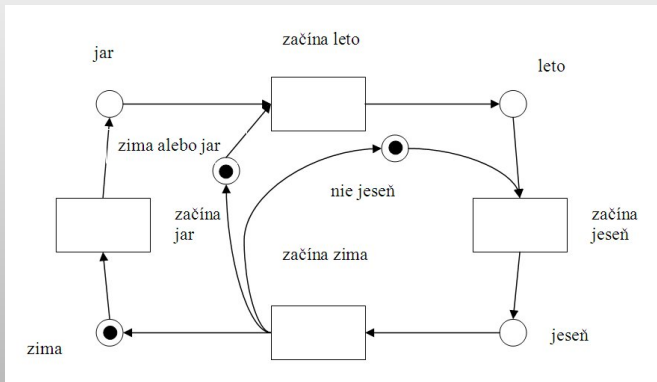


Petriho siete

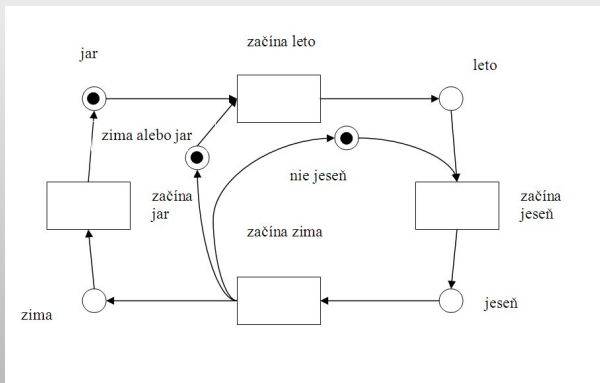


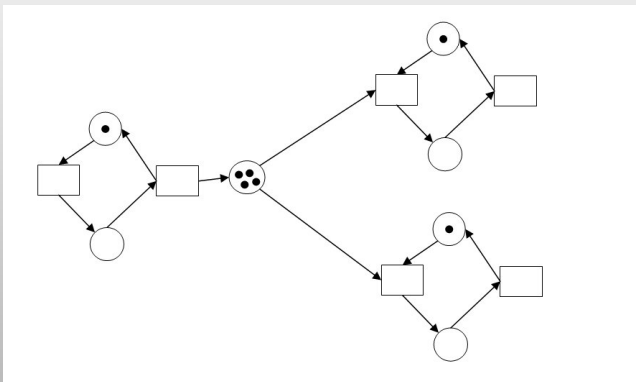
Petriho siete

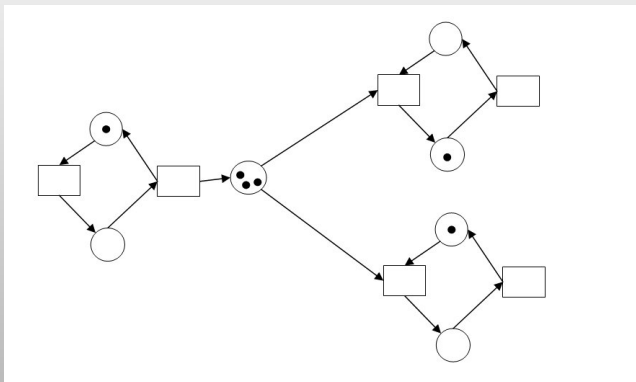


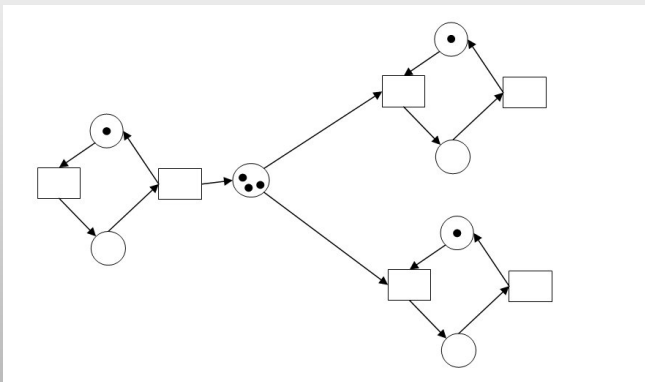


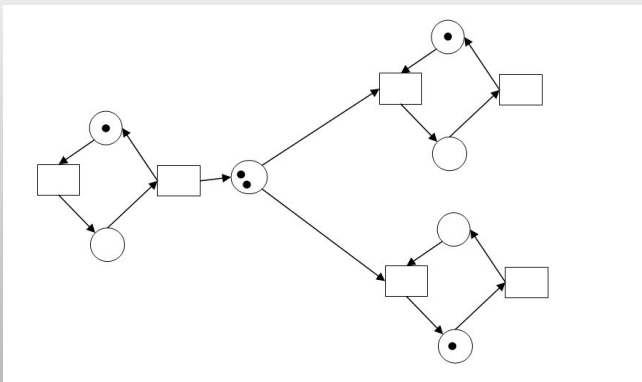
Petriho siete











Definition

Trojica $N = (P, T, F)$ je sieť iff

Definition

Trojica $N = (P, T, F)$ je sieť iff

i) P a T sú disjunktné množiny miest a prechodov

Definition

Trojica $N = (P, T, F)$ je sieť iff

- i) P a T sú disjunktné množiny miest a prechodov
- ii) $F \subseteq (P \times T) \cup (T \times P)$

Definition

Nech $N = (P, T, F)$ je sieť

Definition

Nech $N = (P, T, F)$ je sietė

i) $x \in P \cup T$

Definition

Nech $N = (P, T, F)$ je sietė

i) $x \in P \cup T$

* $x = \{y \mid yFx\}$... preset x

Definition

Nech $N = (P, T, F)$ je sieť

i) $x \in P \cup T$

${}^*x = \{y \mid yFx\}$... preset x

$x^* = \{y \mid xFy\}$... postset x

Definition

Nech $N = (P, T, F)$ je sieť

i) $x \in P \cup T$

${}^*x = \{y \mid yFx\}$... preset x

$x^* = \{y \mid xFy\}$... postset x

${}^*X = \bigcup_{x \in X} {}^*x$

Definition

Nech $N = (P, T, F)$ je sietė

i) $x \in P \cup T$

${}^*x = \{y \mid yFx\}$... preset x

$x^* = \{y \mid xFy\}$... postset x

${}^*X = \bigcup_{x \in X} {}^*x$

$X^* = \bigcup_{x \in X} x^*$

Definition

Nech $N = (P, T, F)$ je sieť

i) $x \in P \cup T$

${}^*x = \{y \mid yFx\}$... preset x

$x^* = \{y \mid xFy\}$... postset x

${}^*X = \bigcup_{x \in X} {}^*x$

$X^* = \bigcup_{x \in X} x^*$

$x \in {}^*y$ iff $y \in x^*$

Definition

Nech $N = (P, T, F)$ je sieť

i) $x \in P \cup T$

${}^*x = \{y \mid yFx\}$... preset x

$x^* = \{y \mid xFy\}$... postset x

${}^*X = \bigcup_{x \in X} {}^*x$

$X^* = \bigcup_{x \in X} x^*$

$x \in {}^*y$ iff $y \in x^*$

ii) dvojica $(p, e) \in P \times T$ je slučka iff pFe a eFp . N sa volá bezslučková, ak neobsahuje slučky.

Definition

Nech $N = (P, T, F)$ je sieť

i) $x \in P \cup T$

${}^*x = \{y \mid yFx\}$... preset x

$x^* = \{y \mid xFy\}$... postset x

${}^*X = \bigcup_{x \in X} {}^*x$

$X^* = \bigcup_{x \in X} x^*$

$x \in {}^*y$ iff $y \in x^*$

ii) dvojica $(p, e) \in P \times T$ je slučka iff pFe a eFp . N sa volá bezslučková, ak neobsahuje slučky.

iii) x je izolovaný iff ${}^*x \cup x^* = \emptyset$

Definition

Nech $N = (P, T, F)$ je sieť

i) $x \in P \cup T$

${}^*x = \{y \mid yFx\}$... preset x

$x^* = \{y \mid xFy\}$... postset x

${}^*X = \bigcup_{x \in X} {}^*x$

$X^* = \bigcup_{x \in X} x^*$

$x \in {}^*y$ iff $y \in x^*$

ii) dvojica $(p, e) \in P \times T$ je slučka iff pFe a eFp . N sa volá bezslučková, ak neobsahuje slučky.

iii) x je izolovaný iff ${}^*x \cup x^* = \emptyset$

iv) N je jednoduchá ak rôzne prvky nemajú rovnaké preset a postset, i.e. $\forall x, y$ platí $({}^*x = {}^*y) \wedge (x^* = y^*) \Rightarrow x = y$

Definition

Nech $N = (P, T, F)$ je sieť

Definition

Nech $N = (P, T, F)$ je sieť

i) $c \subseteq P$ sa volá prípad (case)

Definition

Nech $N = (P, T, F)$ je sieť

i) $c \subseteq P$ sa volá prípad (case)

ii) nech $e \in T$ a $c \subseteq P$, udalosť e je c -umožnená iff $*e \subseteq c$ a $e^* \cap c = \emptyset$

Definition

Nech $N = (P, T, F)$ je sieť

i) $c \subseteq P$ sa volá prípad (case)

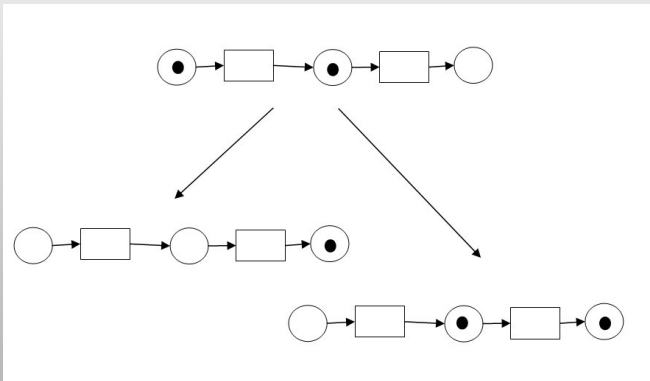
ii) nech $e \in T$ a $c \subseteq P$, udalosť e je c -umožnená iff $*e \subseteq c$ a $e^* \cap c = \emptyset$

iii) nech $e \in T$, $c \subseteq P$ a e je c -umožnená. Potom

$c' = (c \setminus *e) \cup e^*$ je výsledkom vykonania e (označenie $c[e > c']$)

Čo by sa stalo, keby sme vynechali podmienku $e^* \cap c = \emptyset$:

Čo by sa stalo, keby sme vynechali podmienku $e^* \cap c = \emptyset$:



Definition

Nech $N = (P, T, F)$ je sieť

Definition

Nech $N = (P, T, F)$ je sieť

i) množina $G \subseteq T$ je nezávislá iff

$$\forall e_1, e_2 \in G, e_1 \neq e_2 \Rightarrow {}^*e_1 \cap {}^*e_2 = \emptyset \text{ a } e_1^* \cap e_2^* = \emptyset$$

Definition

Nech $N = (P, T, F)$ je sieť

i) množina $G \subseteq T$ je nezávislá iff

$$\forall e_1, e_2 \in G, e_1 \neq e_2 \Rightarrow {}^*e_1 \cap {}^*e_2 = \emptyset \text{ a } e_1^* \cap e_2^* = \emptyset$$

ii) nech c a c' sú prípady a G nezávislá. G sa volá krok z c do c' ($c[G > c'$) iff $\forall e \in G$ je c -umožnená a $c' = (c \setminus {}^*G) \cup G^*$

Definition

Nech $N = (P, T, F)$ je sieť

i) množina $G \subseteq T$ je nezávislá iff

$$\forall e_1, e_2 \in G, e_1 \neq e_2 \Rightarrow {}^*e_1 \cap {}^*e_2 = \emptyset \text{ a } e_1^* \cap e_2^* = \emptyset$$

ii) nech c a c' sú prípady a G nezávislá. G sa volá krok z c do c' ($c[G > c'$) iff $\forall e \in G$ je c -umožnená a $c' = (c \setminus {}^*G) \cup G^*$

Poznámka: prechody z G možno urobiť i sekvenčne v ľubovolnom poradí s rovnakým výsledkom.

Condition / Event Systém nie je plne popísaný pokiaľ neurčíme množinu prípadov C . Táto by mala spĺňať nasledovné podmienky:

Condition / Event Systém nie je plne popísaný pokiaľ neurčíme množinu prípadov C . Táto by mala spĺňať nasledovné podmienky:

- i) Ak krok G je možný v prípade c , $c \in C$ tak aj výsledok bude opäť v C ,

Condition / Event Systém nie je plne popísaný pokiaľ neurčíme množinu prípadov C . Táto by mala spĺňať nasledovné podmienky:

- i) Ak krok G je možný v prípade c , $c \in C$ tak aj výsledok bude opäť v C ,
- ii) naopak, ak prípad c , $c \in C$ je výsledkom kroku G tak aj prípad, z ktorého to vzišlo musí byť opäť v C ,

Condition / Event Systém nie je plne popísaný pokiaľ neurčíme množinu prípadov C . Táto by mala spĺňať nasledovné podmienky:

- i) Ak krok G je možný v prípade $c, c \in C$ tak aj výsledok bude opäť v C ,
- ii) naopak, ak prípad $c, c \in C$ je výsledkom kroku G tak aj prípad, z ktorého to vzišlo musí byť opäť v C ,
- iii) všetky prípady v C môžu byť na seba transformované (striedavo - dopredu i dozadu),

Condition / Event Systém nie je plne popísaný pokiaľ neurčíme množinu prípadov C . Táto by mala spĺňať nasledovné podmienky:

- i) Ak krok G je možný v prípade c , $c \in C$ tak aj výsledok bude opäť v C ,
- ii) naopak, ak prípad c , $c \in C$ je výsledkom kroku G tak aj prípad, z ktorého to vzišlo musí byť opäť v C ,
- iii) všetky prípady v C môžu byť na seba transformované (striedavo - dopredu i dozadu),
- iv) každý prechod je c -umožnený pre nejaké c a každé miesto patrí do nejakého c a nepatrí do nejakého c' .

Definition

Štvorica $\Sigma = (P, T, F, C)$ je Condition / Event Systém (C/E systém) iff

Definition

Štvorica $\Sigma = (P, T, F, C)$ je Condition / Event Systém (C/E systém) iff

i) (P, T, F) je jednoduchá sieť bez izolovaných prvkov a $P \cup T \neq \emptyset$,

Definition

Štvorica $\Sigma = (P, T, F, C)$ je Condition / Event Systém (C/E systém) iff

i) (P, T, F) je jednoduchá sieť bez izolovaných prvkov a

$P \cup T \neq \emptyset$,

ii) $C \subseteq \mathcal{P}(P)$ je trieda ekvivalencie relácie $R_\Sigma = (r_\Sigma \cup r_\Sigma^{-1})^*$ kde $c_1 r_\Sigma c_2$ iff $\exists G, G \subseteq T, C_1[G > c_2$. C je trieda prípadov,

Definition

Štvorica $\Sigma = (P, T, F, C)$ je Condition / Event Systém (C/E systém) iff

- i) (P, T, F) je jednoduchá sieť bez izolovaných prvkov a $P \cup T \neq \emptyset$,
- ii) $C \subseteq \mathcal{P}(P)$ je trieda ekvivalencie relácie $R_\Sigma = (r_\Sigma \cup r_\Sigma^{-1})^*$ kde $c_1 r_\Sigma c_2$ iff $\exists G, G \subseteq T, C_1[G > c_2$. C je trieda prípadov,
- iii) $\forall e, e \in T, \exists c, c \in C$ také, že e je c -umožnený,

Definition

Štvorica $\Sigma = (P, T, F, C)$ je Condition / Event Systém (C/E systém) iff

- i) (P, T, F) je jednoduchá sieť bez izolovaných prvkov a $P \cup T \neq \emptyset$,
- ii) $C \subseteq \mathcal{P}(P)$ je trieda ekvivalencie relácie $R_\Sigma = (r_\Sigma \cup r_\Sigma^{-1})^*$ kde $c_1 r_\Sigma c_2$ iff $\exists G, G \subseteq T, C_1[G > c_2$. C je trieda prípadov,
- iii) $\forall e, e \in T, \exists c, c \in C$ také, že e je c -umožnený,
- iv) $\forall p, p \in P, \exists c, c', c \in C, c' \in C$ také, že $p \in c$ a $p \notin c'$.

Úloha: Napíšte Condition / Event Systémy odpovedajúce nasledujúcim procesom:

Úloha: Napíšte Condition / Event Systémy odpovedajúce nasledujúcim procesom:

a.Nil

Úloha: Napíšte Condition / Event Systémy odpovedajúce nasledujúcim procesom:

a.Nil

a.b.c.Nil

Úloha: Napíšte Condition / Event Systémy odpovedajúce nasledujúcim procesom:

a.Nil

a.b.c.Nil

a.Nil|b.Nil

Úloha: Napíšte Condition / Event Systémy odpovedajúce nasledujúcim procesom:

a.Nil

a.b.c.Nil

a.Nil|b.Nil

a.Nil| $\bar{a}.Nil$

Úloha: Napíšte Condition / Event Systémy odpovedajúce nasledujúcim procesom:

a.Nil

a.b.c.Nil

a.Nil|b.Nil

a.Nil| $\bar{a}.Nil$

a.b.Nil|c.d.Nil

Úloha: Napíšte Condition / Event Systémy odpovedajúce nasledujúcim procesom:

a.Nil

a.b.c.Nil

a.Nil|b.Nil

a.Nil|\bar{a}.Nil

a.b.Nil|c.d.Nil

a.c.Nil|b.\bar{c}.Nil

Úloha: Napíšte Condition / Event Systémy odpovedajúce nasledujúcim procesom:

$a.Nil$

$a.b.c.Nil$

$a.Nil|b.Nil$

$a.Nil|\bar{a}.Nil$

$a.b.Nil|c.d.Nil$

$a.c.Nil|b.\bar{c}.Nil$

$(a.c.Nil|b.\bar{c}.Nil) \setminus \{c\}$

Úloha: Napíšte Condition / Event Systémy odpovedajúce nasledujúcim procesom:

$a.Nil$

$a.b.c.Nil$

$a.Nil|b.Nil$

$a.Nil|\bar{a}.Nil$

$a.b.Nil|c.d.Nil$

$a.c.Nil|b.\bar{c}.Nil$

$(a.c.Nil|b.\bar{c}.Nil) \setminus \{c\}$

$a.Nil|b.Nil|c.Nil|d.Nil$

Úloha: Napíšte Condition / Event Systémy odpovedajúce nasledujúcim procesom:

$a.Nil$

$a.b.c.Nil$

$a.Nil|b.Nil$

$a.Nil|\bar{a}.Nil$

$a.b.Nil|c.d.Nil$

$a.c.Nil|b.\bar{c}.Nil$

$(a.c.Nil|b.\bar{c}.Nil) \setminus \{c\}$

$a.Nil|b.Nil|c.Nil|d.Nil$

$\mu Xtick.X$

Úloha: Napíšte Condition / Event Systémy odpovedajúce nasledujúcim procesom:

$a.Nil$

$a.b.c.Nil$

$a.Nil|b.Nil$

$a.Nil|\bar{a}.Nil$

$a.b.Nil|c.d.Nil$

$a.c.Nil|b.\bar{c}.Nil$

$(a.c.Nil|b.\bar{c}.Nil) \setminus \{c\}$

$a.Nil|b.Nil|c.Nil|d.Nil$

$\mu X tick.X$

$\mu X (tick.X + tick.Nil)$

Definition

Nech $\Sigma = (P, T, F, C)$ je Condition / Event Systém (C/E systém)
a $b, \bar{b} \in P$.

Definition

Nech $\Sigma = (P, T, F, C)$ je Condition / Event Systém (C/E systém)
a $b, \bar{b} \in P$.

i) \bar{b} je **komplement** b iff $*b = \bar{b}^*$ a $*\bar{b} = b^*$

Definition

Nech $\Sigma = (P, T, F, C)$ je Condition / Event Systém (C/E systém) a $b, \bar{b} \in P$.

i) \bar{b} je **komplement** b iff $*b = \bar{b}^*$ a $*\bar{b} = b^*$

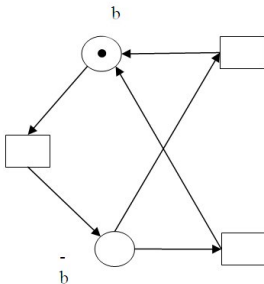
ii) Σ je **úplná** ak pre každé b existuje komplement \bar{b} . (ľahko vidno, že môže byť len jeden, keďže (P, T, F) je jednoduchá.

Definition

Nech $\Sigma = (P, T, F, C)$ je Condition / Event Systém (C/E systém) a $b, \bar{b} \in P$.

i) \bar{b} je **komplement** b iff $*b = \bar{b}^*$ a $*\bar{b} = b^*$

ii) Σ je **úplná** ak pre každé b existuje komplement \bar{b} . (ľahko vidno, že môže byť len jeden, keďže (P, T, F) je jednoduchá.



Definition

Σ_1 a Σ_2 sú **ekvivalentné** ($\Sigma \simeq \Sigma'$) ak existujú bijekcie f, g medzi C_1 a C_2 a medzi T_1 a T_2 také, že

$$c[G > c' \Leftrightarrow f(c)[g(G) > f(c')$$

Definition

Σ_1 a Σ_2 sú **ekvivalentné** ($\Sigma \simeq \Sigma'$) ak existujú bijekcie f, g medzi C_1 a C_2 a medzi T_1 a T_2 také, že

$$c[G > c' \Leftrightarrow f(c)[g(G) > f(c')$$

Definition

Σ je **bezkontaktný** iff $\forall e, e \in T, \forall c, c \in C$:

Definition

Σ_1 a Σ_2 sú **ekvivalentné** ($\Sigma \simeq \Sigma'$) ak existujú bijekcie f, g medzi C_1 a C_2 a medzi T_1 a T_2 také, že

$$c[G > c' \Leftrightarrow f(c)[g(G) > f(c')$$

Definition

Σ je **bezkontaktný** iff $\forall e, e \in T, \forall c, c \in C$:

i) $*e \subseteq c \Rightarrow e^* \subseteq P \setminus c$

Definition

Σ_1 a Σ_2 sú **ekvivalentné** ($\Sigma \simeq \Sigma'$) ak existujú bijekcie f, g medzi C_1 a C_2 a medzi T_1 a T_2 také, že

$$c[G > c' \Leftrightarrow f(c)[g(G) > f(c')]$$

Definition

Σ je **bezkontaktný** iff $\forall e, e \in T, \forall c, c \in C$:

- i) $*e \subseteq c \Rightarrow e^* \subseteq P \setminus c$
- ii) $e^* \subseteq c \Rightarrow *e \subseteq P \setminus c$

Theorem

i) Každý úplný systém je bezkontaktný.

Theorem

- i) Každý úplný systém je bezkontaktný.*
- ii) Pre každý C/E systém Σ existuje bezkontaktný C/E systém Σ' taký, že $\Sigma \simeq \Sigma'$.*

Vrcholy - prvky C (t.j. prípady)

Grafická reprezentácia - case graf

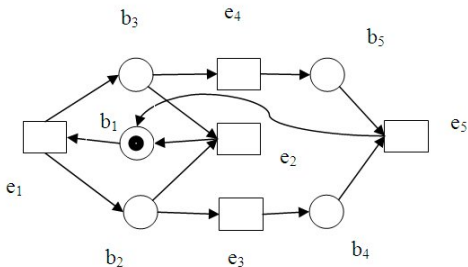
Vrcholy - prvky C (t.j. prípady)

Hrany - kroky medzi prípadmi (označené podmnožinami T)

Grafická reprezentácia - case graf

Vrcholy - prvky C (t.j. prípady)

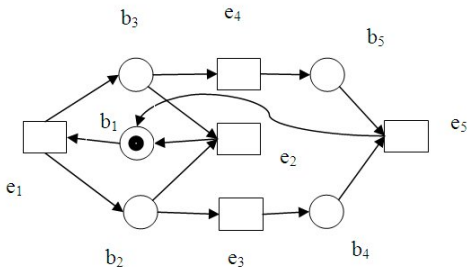
Hrany - kroky medzi prípadmi (označené podmnožinami T)



Grafická reprezentácia - case graf

Vrcholy - prvky C (t.j. prípady)

Hrany - kroky medzi prípadmi (označené podmnožinami T)

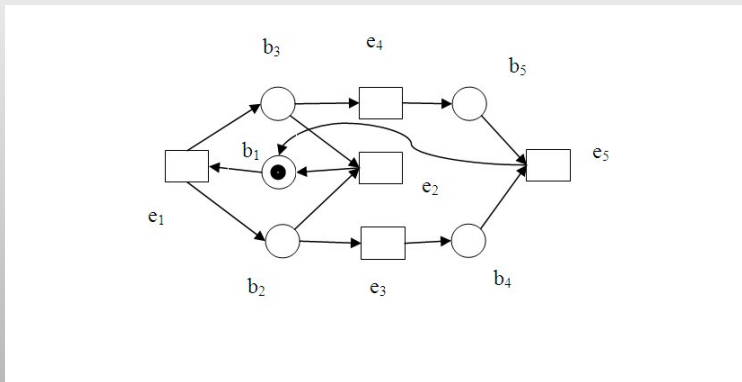


Úloha: Napíšte case graf pre predchádzajúci systém.

Grafická reprezentácia - case graf

Vrcholy - prvky C (t.j. prípady)

Hrany - kroky medzi prípadmi (označené podmnožinami T)



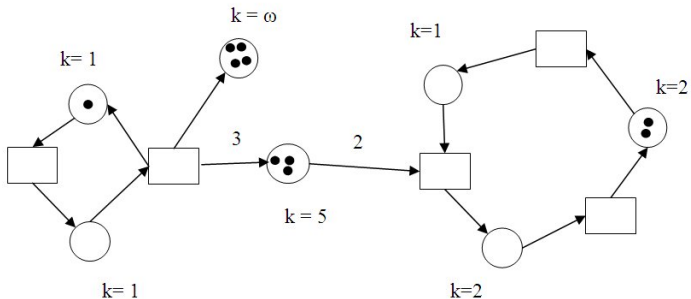
Úloha: Napíšte case graf pre predchádzajúci systém.

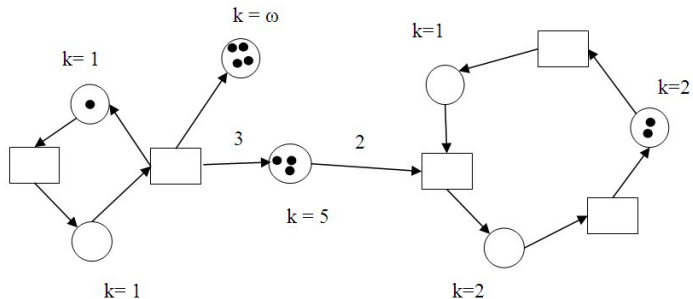
Úloha: Napíšte case graf pre systém odpovedajúci $a.b.Nil|c.d.Nil$.

Theorem

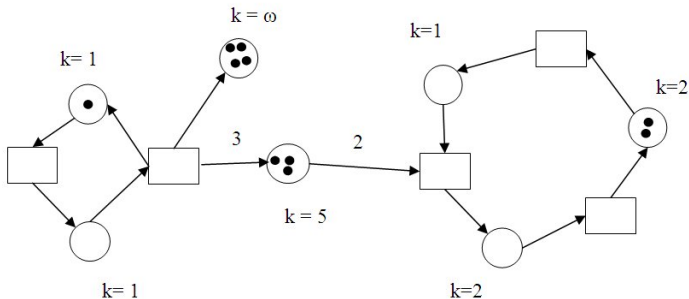
C/E systémy su ekvivalentné, ak ich case grafy sú izomorfné.

Place / Transition systémy



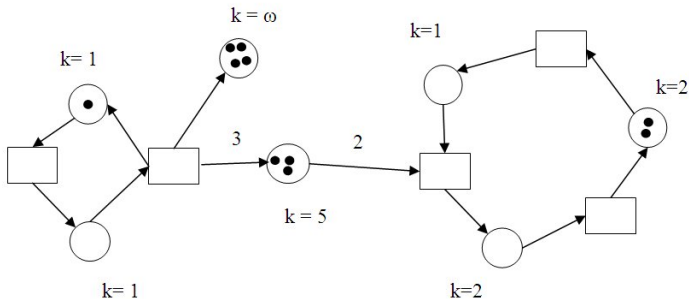


Producent a dvaja konzumenti



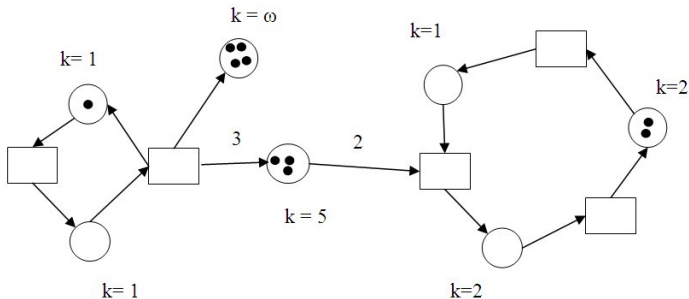
Producent a dvaja konzumenti

1. zásobník môže obsahovať najviac 5 bodiek



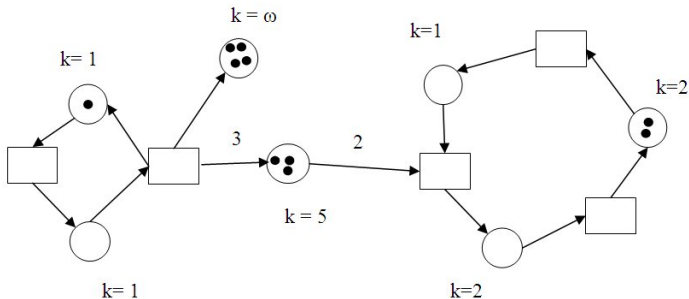
Producent a dvaja konzumenti

1. zásobník môže obsahovať najviac 5 bodiek
2. producent generuje naraz 3 bodky



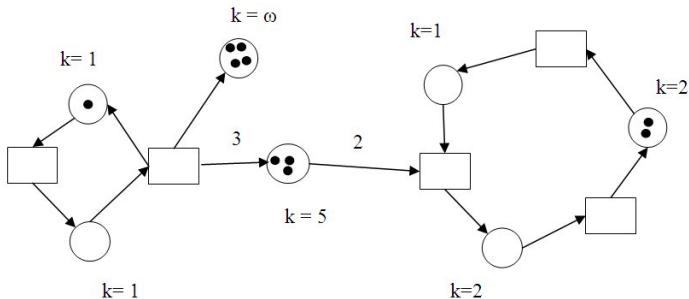
Producent a dvaja konzumenti

1. zásobník môže obsahovať najviac 5 bodiek
2. producent generuje naraz 3 bodky
3. každý konzument odoberie 2 bodky zo zásobníka



Producent a dvaja konzumenti

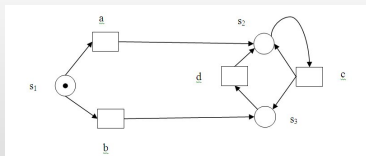
1. zásobník môže obsahovať najviac 5 bodiek
2. producent generuje naraz 3 bodky
3. každý konzument odoberie 2 bodky zo zásobníka
4. najviac jeden konzument má prístup k zásobníku



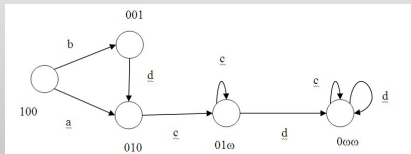
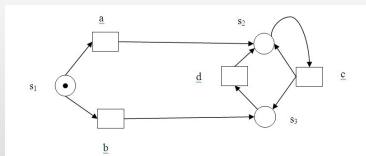
Producent a dvaja konzumenti

1. zásobník môže obsahovať najviac 5 bodiek
2. producent generuje naraz 3 bodky
3. každý konzument odoberie 2 bodky zo zásobníka
4. najviac jeden konzument má prístup k zásobníku
5. kroky producenta sa počítajú

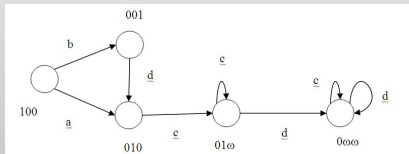
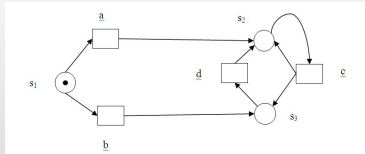
Place / Transition systémy - grafická reprezentácia



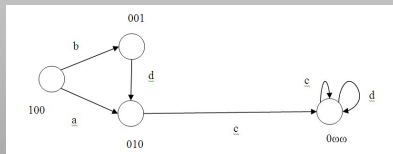
Place / Transition systémy - grafická reprezentácia



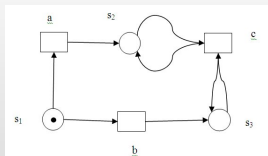
Place / Transition systémy - grafická reprezentácia



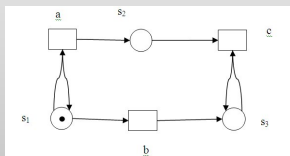
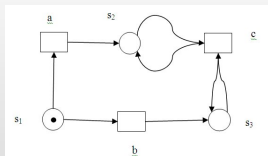
ale aj



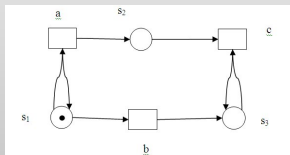
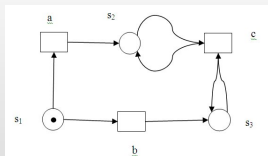
Place / Transition systémy



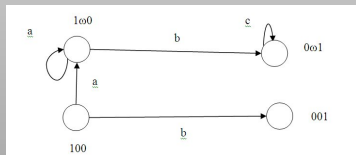
Place / Transition systémy



Place / Transition systémy



oba majú rovnaký case graf



Problémy:

Problémy:

- dosiahnuteľnosť (EXSPACE-hard)

Problémy:

- dosiahnuteľnosť (EXSPACE-hard)
- liveness

Problémy:

- dosiahnuteľnosť (EXSPACE-hard)
- liveness
- boundedness

Problémy:

- dosiahnuteľnosť (EXSPACE-hard)
- liveness
- boundedness

Iné typy:

Problémy:

- dosiahnuteľnosť (EXSPACE-hard)
- liveness
- boundedness

Iné typy:

- reset hrany, vyprázdni miesto - dosiahnuteľnosť sa stáva nerozhodnuteľnou

Problémy:

- dosiahnuteľnosť (EXSPACE-hard)
- liveness
- boundedness

Iné typy:

- reset hrany, vyprázdni miesto - dosiahnuteľnosť sa stáva nerozhodnuteľnou
- inhibitors (test na prázdnoť miesta), sila Turingového stroja

Problémy:

- dosiahnuteľnosť (EXSPACE-hard)
- liveness
- boundedness

Iné typy:

- reset hrany, vyprázdni miesto - dosiahnuteľnosť sa stáva nerozhodnuteľnou
- inhibitors (test na prázdnoť miesta), sila Turingového stroja
- Coloured Petri Nets

Problémy:

- dosiahnuteľnosť (EXSPACE-hard)
- liveness
- boundedness

Iné typy:

- reset hrany, vyprázdni miesto - dosiahnuteľnosť sa stáva nerozhodnuteľnou
- inhibitors (test na prázdnoť miesta), sila Turingového stroja
- Coloured Petri Nets
- hierarchical Petri Nets

Problémy:

- dosiahnuteľnosť (EXSPACE-hard)
- liveness
- boundedness

Iné typy:

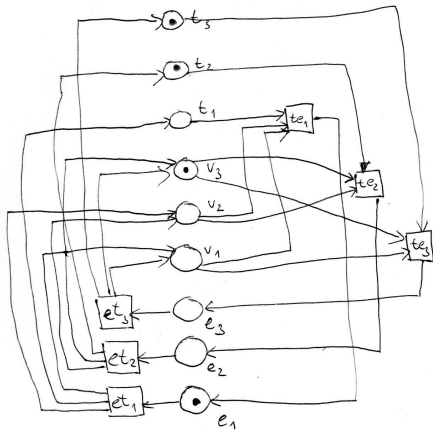
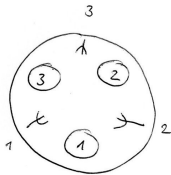
- reset hrany, vyprázdni miesto - dosiahnuteľnosť sa stáva nerozhodnuteľnou
- inhibitors (test na prázdnoť miesta), sila Turingového stroja
- Coloured Petri Nets
- hierarchical Petri Nets
- časové Petri nets

Problémy:

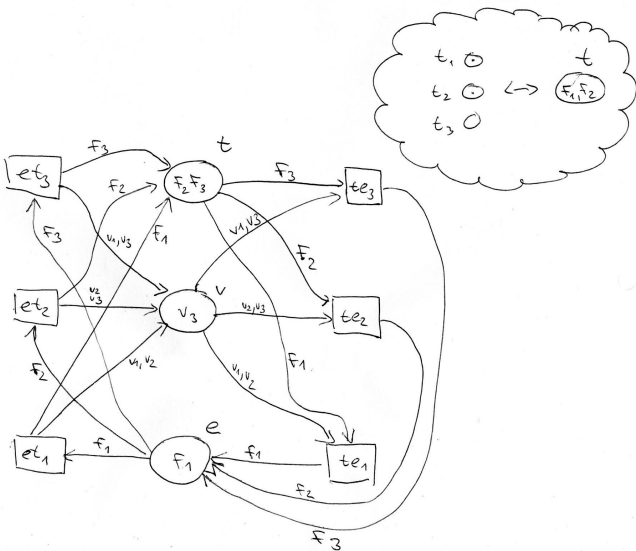
- dosiahnuteľnosť (EXSPACE-hard)
- liveness
- boundedness

Iné typy:

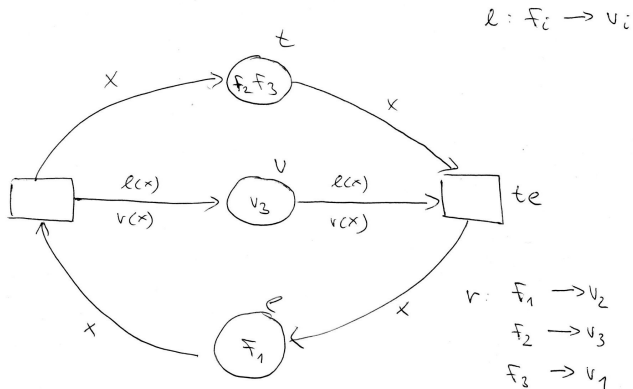
- reset hrany, vyprázdni miesto - dosiahnuteľnosť sa stáva nerozhodnuteľnou
- inhibitors (test na prázdnoť miesta), sila Turingového stroja
- Coloured Petri Nets
- hierarchical Petri Nets
- časové Petri nets
- Petri nets s prioritami
- ...



Predicate / Event systémy



Predicate / Event systémy



Kombinují sílu Petriho sítí a programovací jazyk

Kombinujú silu Petriho sietí a programovací jazyk

Petriho sieť ponúka spôsob vyjadrenia synchronizácie konkurentných procesov.

Kombinujú silu Petriho sietí a programovací jazyk

Petriho sieť ponúka spôsob vyjadrenia synchronizácie konkurentných procesov.

Programovací jazyk ponúka nástroje na definovanie rôznych dátových typov a spôsob manipuláciu ich hodnôt.

Kombinujú silu Petriho sietí a programovací jazyk

Petriho sieť ponúka spôsob vyjadrenia synchronizácie konkurentných procesov.

Programovací jazyk ponúka nástroje na definovanie rôznych dátových typov a spôsob manipuláciu ich hodnôt.

Kombinujú grafický a textový nástroj.

Miesta (places)

Miesta (places)

- Meno

Miesta (places)

- Meno

- Množina farieb (colours), ktoré môžu obsahovať

Miesta (places)

- Meno
- Množina farieb (colours), ktoré môžu obsahovať
- Počiatočné ohodnotenie - multimnožina "colours".

Prechodu (transitions)

Prechodu (transitions)

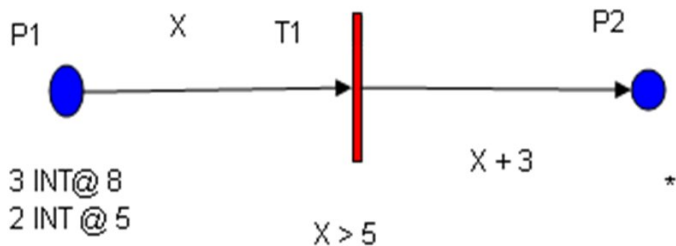
- Meno

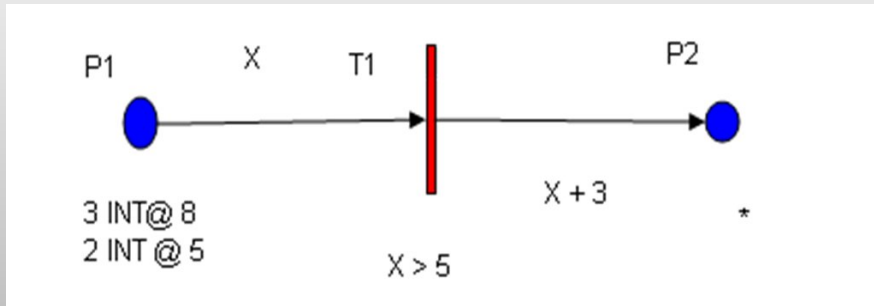
Prechodu (transitions)

- Meno

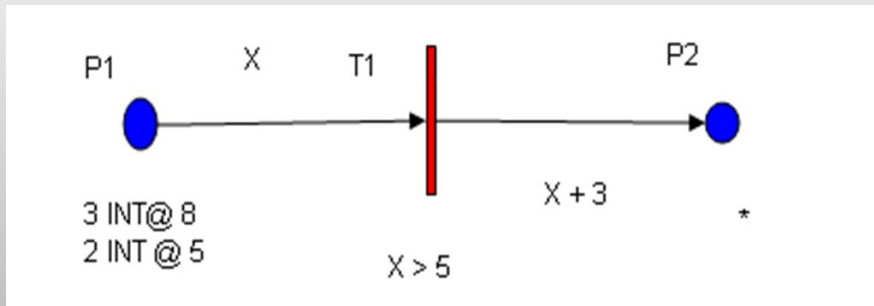
= Guard (stráž) - boolovský výraz, může obsahovat premenné

Coloured Petri nets



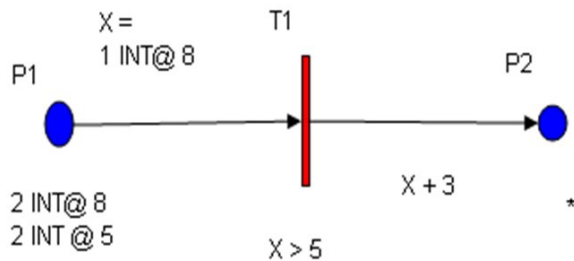


Miesto P1 obsahuje 5 farebných tokenov typu Integer.

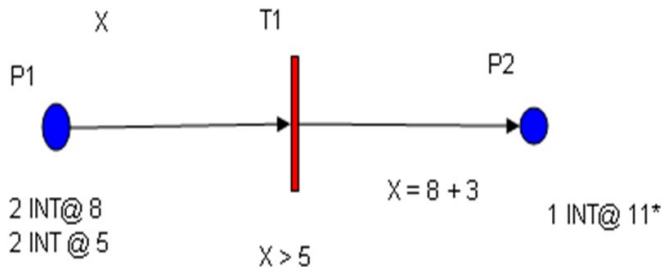


Miesto P1 obsahuje 5 farebných tokenov typu Integer.
3 tokeny majú hodnotu 8 a 2 tokeny hodnotu 5.

Coloured Petri nets



Coloured Petri nets



Coloured Petri nets

