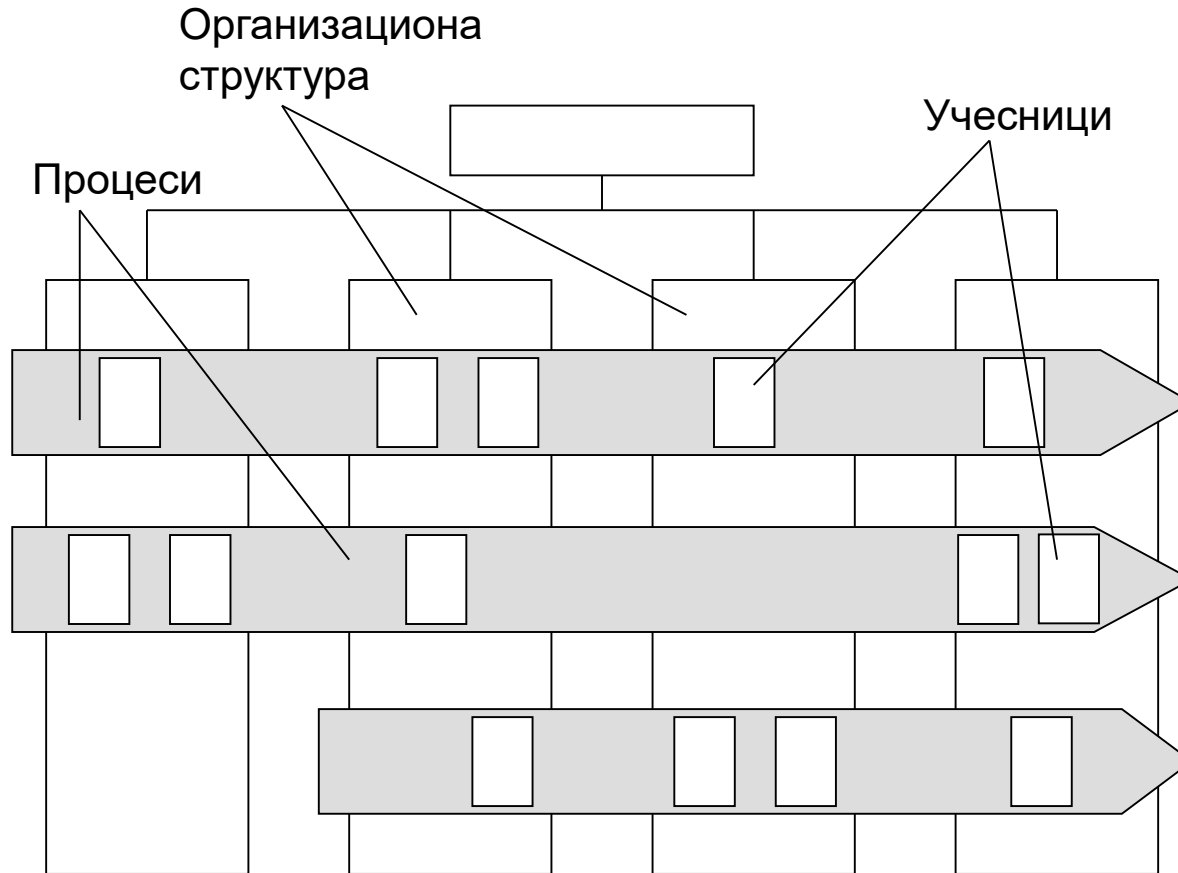


Анализа процеса и Петријеве мреже

Процес

- *Davenport* (1990): “... специфичан **низ активности** у времену и простору, који има свој почетак и крај и прецизно дефинисане **улазе и излазе**.”
- *Harrington* (1991): “... свака активност или **група активности** које узимају **улаз**, повећавају му вредност и тиме производе **излаз** за унутрашњег или спољашњег корисника.”
- *Hammer* и *Champy* (1993): “... **колекција активности** која има више **улаза** и ствара више **излаза** који имају неку вредност за корисника.”
- Standard ISO 8402 (1996) “... скуп међусобно повезаних **ресурса** и **активности** који претварају **улазне** елементе у **излазне**.”
- Standard ISO 9000 (2000) “... скуп међусобно повезаних или међусобно делујућих **активности** који претвара **улазне** у **излазне** елементе.”
- Веорватно најкраћа дефиниција процеса је
“... трансформација улазних величина (*input*) у излазне величине (*output*) .“

Пословни процеси



MIT Process Handbook Project

Сврха: развијање методологије и софтверског алата за представљање пословних процеса на различитим нивоима апстракције и сакупљању, организовању и анализирању примера о томе како различите организације изводе сличне пословне процесе

Основни концепти:

Декомпозиција - специјализација процеса.

Управљање зависностима.

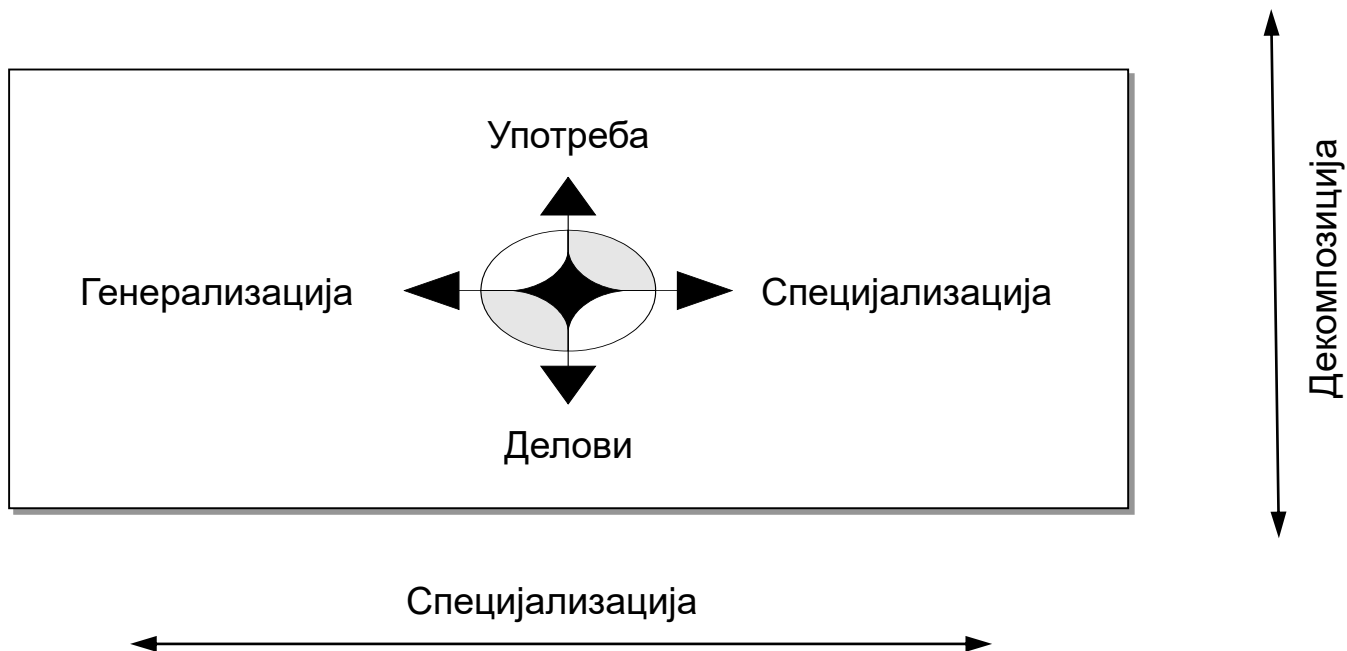
*(Center for Coordination Science,
Massachusetts Institute of Technology, Cambridge, Massachusetts)*

Декомпозиција и специјализација

- Две димензије у опису пословних процеса:
 - **Различити делови (декомпозиција) пословних процеса** Декомпозиција процеса - раздвајање процеса на кораке (активности) који треба да се изврше да би се процес комплетирао, односно остварио његов циљ.
 - **Различити типови (специјализација) пословних процеса** Специјализација процеса – нова димензија у опису процеса која анализира процесе по типу (класи) и даје одговоре на питања: како, шта, коме итд. Класе процеса се могу дефинисати помоћу:
 - Минималног скупа понашања система: сва понашања (активности) које сваки процес из посматране класе мора да садржи. Специјализација процеса се врши додавањем активности које посматрани процес разликује од осталих процеса у класи.
 - Максималног скупа понашања система: сва могућа понашања (активности) које процеси из посматране класе могу да садрже. Специјализација процеса се врши уклањањем активности, односно рестрикцијом овог скупа.

Компас процеса

- Који су различити делови овог процеса? - *Parts* (делови)
- У ком већем процесу се користи овај процес? - *Uses* (употреба)
- Који су други процеси попут овог? - *Generalization* (генерализација)
- На које друге начине може да се изврши овај процес? - *Specialization* (специјализација)



Управљање зависностима

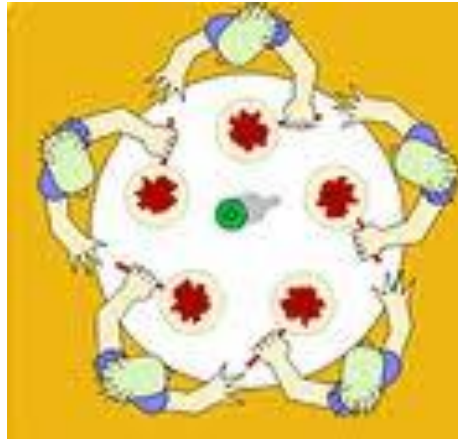
- Три основне врсте зависности:
 1. Ток (произвођач - корисник)
 2. Дељење улаза
 3. Слагање, дељење излаза
- Кад год постоји зависност између две активности потребна је координација.
- Активности у процесу: активности производње(језгро процеса) и активности координације.
 - Идентификација активности координације:
 - директним тражењем међу свим активностима,
 - тражењем учесника у процесу који координирају или
 - елиминисањем активности производње из листе свих активности. Активности које остају су или активности координације или активности које треба избацити из процеса.
 - Често су активности које користе више информација од осталих (информационо интензивне), управо активности координације.

Deadlock

- Блокада, застој, ћорсокак, мртво стање
- Скуп процеса је у блокади ако сваки процес чека на догађај који може да изазове само неки други процес.



Услови за Deadlock



1. Узајамно искључење (*Mutual exclusion*) – само један процес може да користи ресурс у једном тренутку.
2. Држи и чекај (*Hold and wait*) - процес чува ресурс и чека следећи.
3. Нема права пречег (*No preemption*) – заузети ресурс се не може преотети. Процес добровољно ослобађа ресурс након извршења.
4. Кружно чекање (*Circular wait*) – мора постојати ланац од најмање два процеса где сваки чека ресурс заузет суседним процесом.

Управљање блокадом

- Превенција – осигурати да се *deadlock* не деси тако што се обезбеди да се бар један од 4 услова не појави
- Избегавање *deadlock* - осигурати да се *deadlock* не деси коришћењем информација о захтевима ресурсима и динамичким избегавањем несигурних ситуација
- Детекција и опоравак – дозволити *deadlock*, али регистровати када се појави, опоравити процес и наставити
- Игнорисање – најлакши и најчешћи приступ.

Технике за моделирање процеса: захтеви

- Моделирање активности, ресурса и веза између њих.
- Процесна оријентација.
 - Моделирање у хијерархији или у сегментима и приказивање различитих варијанти (типова) процеса.
 - Моделирање зависности и механизма координације.
- Моделирање кашњења (стохастичког).
- Лако прилагођавање и отвореност.
- Алат.
- Разумљива и лака за употребу.
- Могућност анализе процеса.

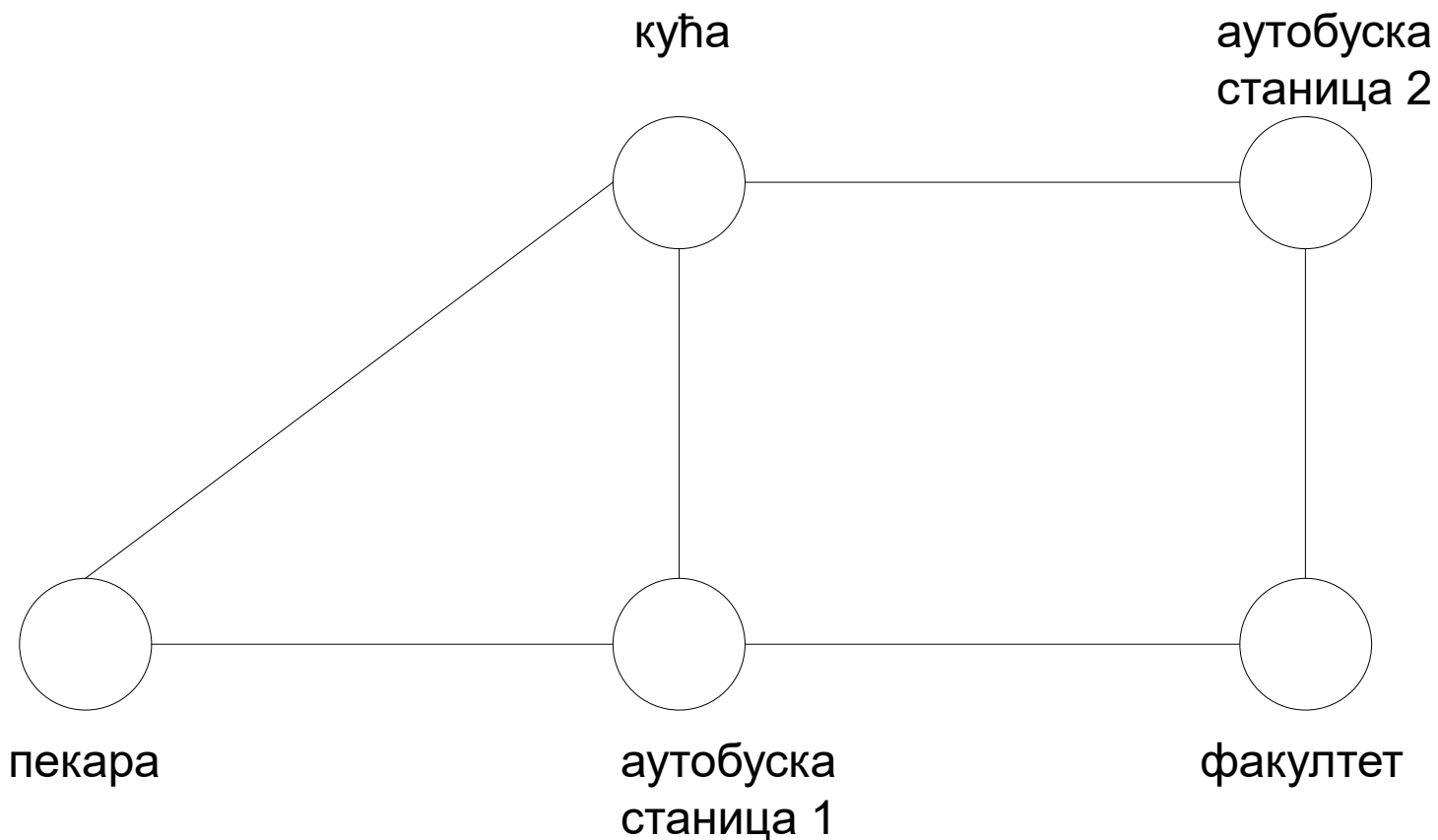
ПЕТРИЈЕВЕ МРЕЖЕ (Petri nets- PN)

- **Петријеве мреже** су графички и математички алат за моделирање и анализу процеса и њихове динамике.
- Концепт PN је увео Carl Adam Petri 1962. године у Докторској дисертацији под називом "*Kommunikation mit Automaten*", на Факултету за математику и физику *Uniadt* у Немачкој.

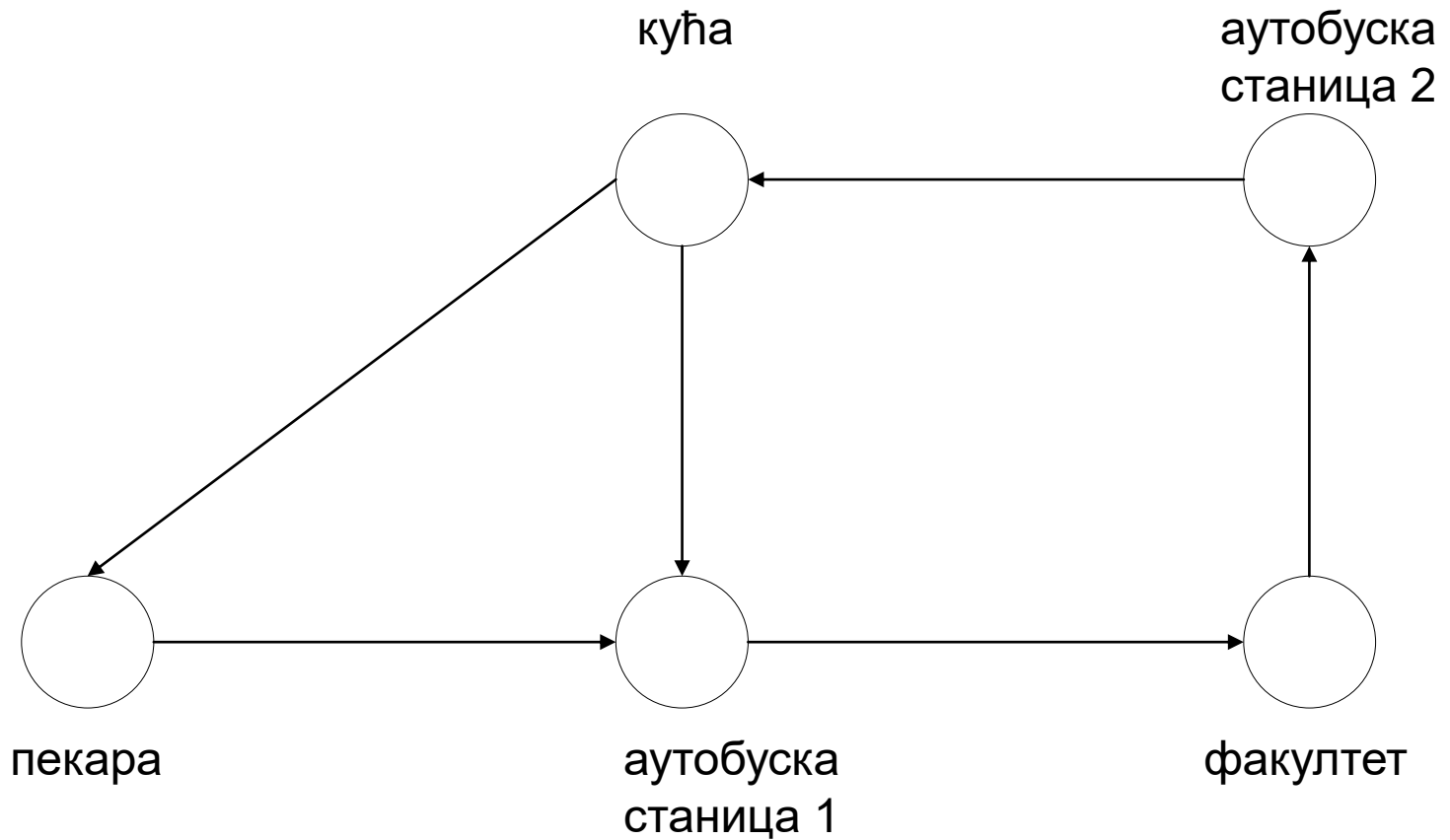
ПЕТРИЈЕВЕ МРЕЖЕ (Petri nets- PN)

- Преко 250 књига посвећених Петријевим мрежама и више од 2000 књига у којима су Петријеве мреже обрађене у оквиру поглавља или наслова.
- Преко 8500 радова посвећених Петријевим мрежама.
- Сталне конференције:
 - *International Conferences on Application and Theory of Petri Nets and Other Models of Concurrency* (1980.)
 - *Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools* (1998.),
 - *The International Conference on Quantitative Evaluation of SysTems (QEST)* у оквиру кога се од 2004. године одржава *the International Workshop on Petri Nets and Performance Models* (PNPM)

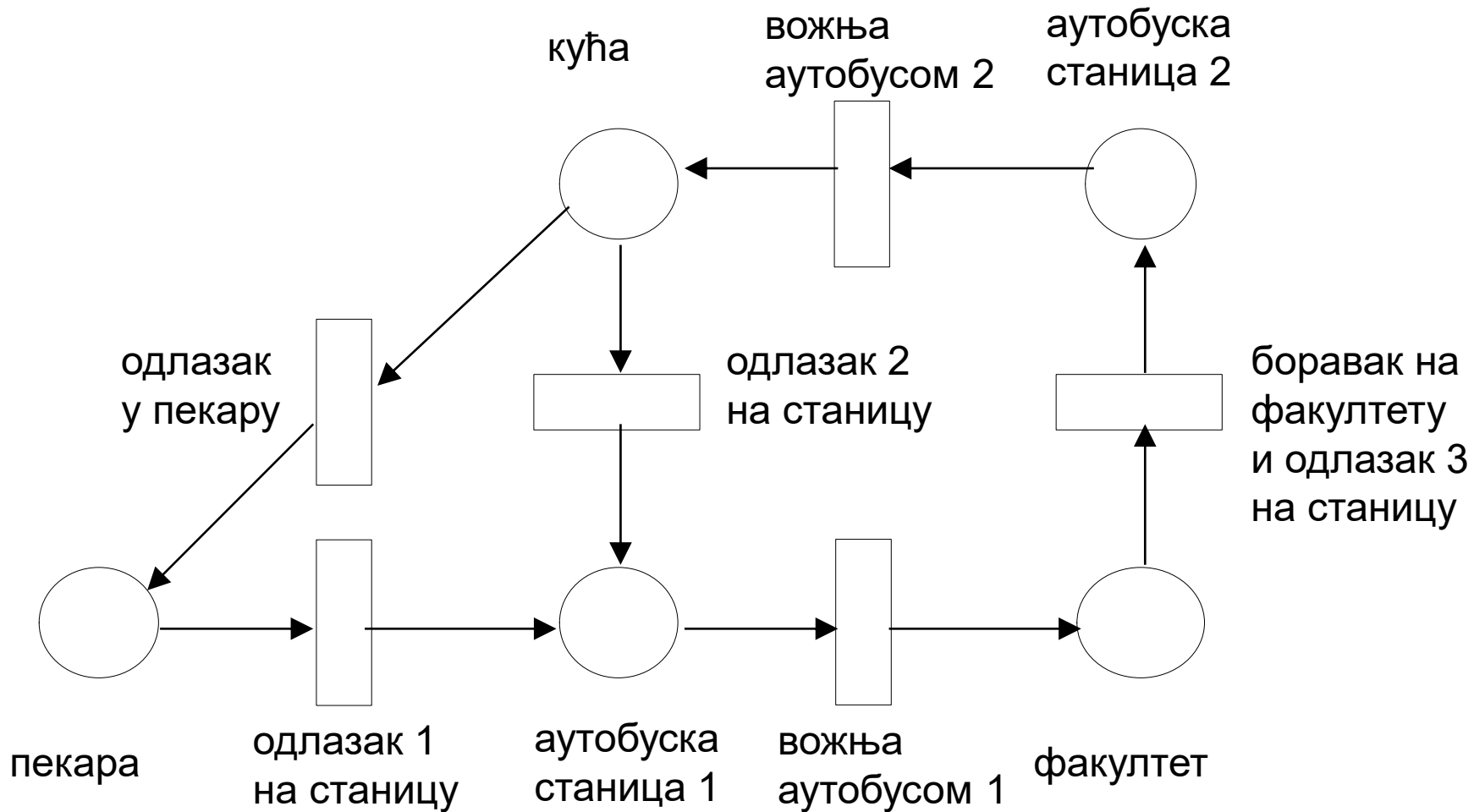
Граф (X,U)



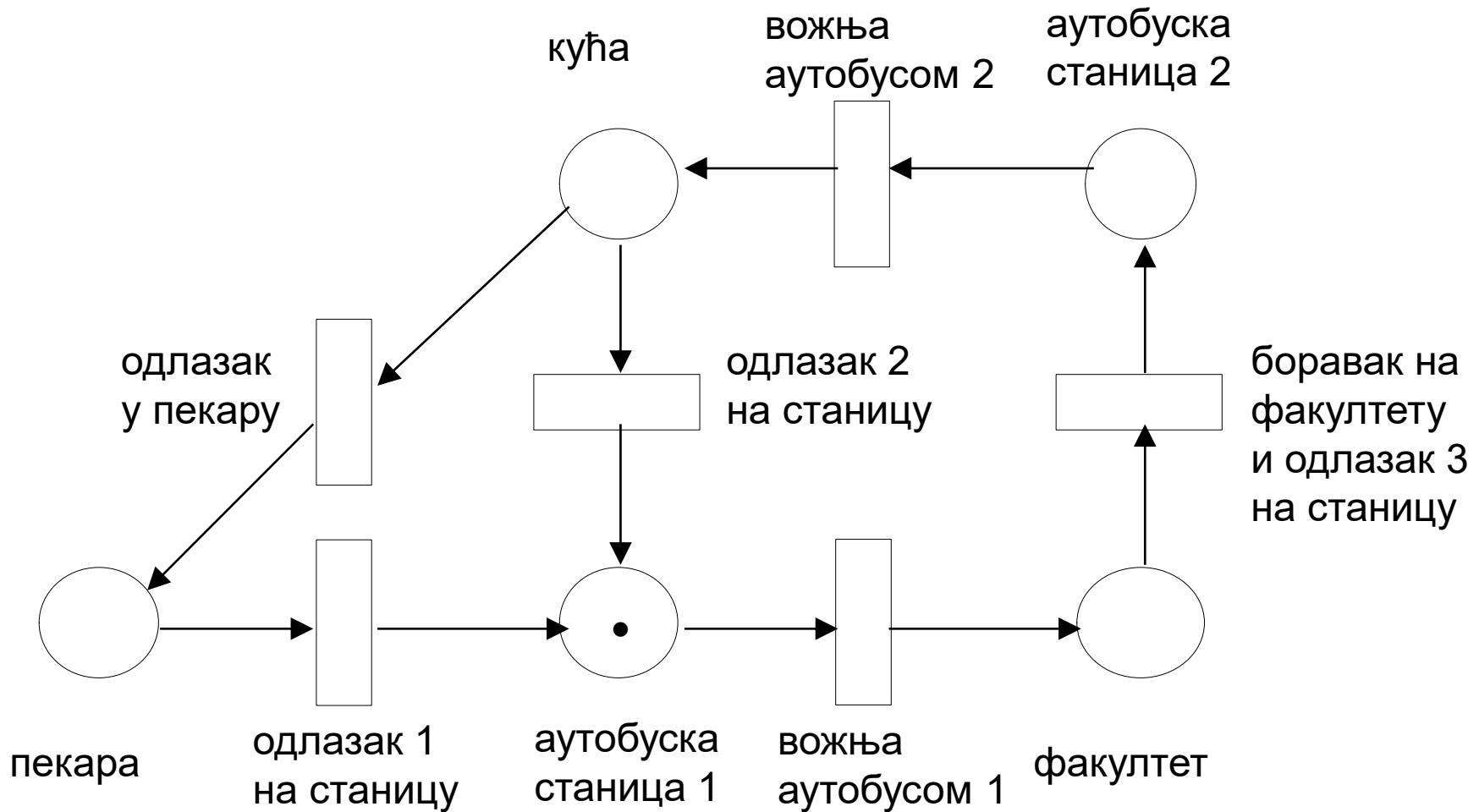
Диграф (X,U)



Бипартитни граф (P,T,U)



Петријева мрежа – Petri Net (PN)



Формална дефиниција РН

Елементарна (црно-бела) РН је петорка $PN = (P, T, A, W, M_0) = (N, M_0)$ где је:

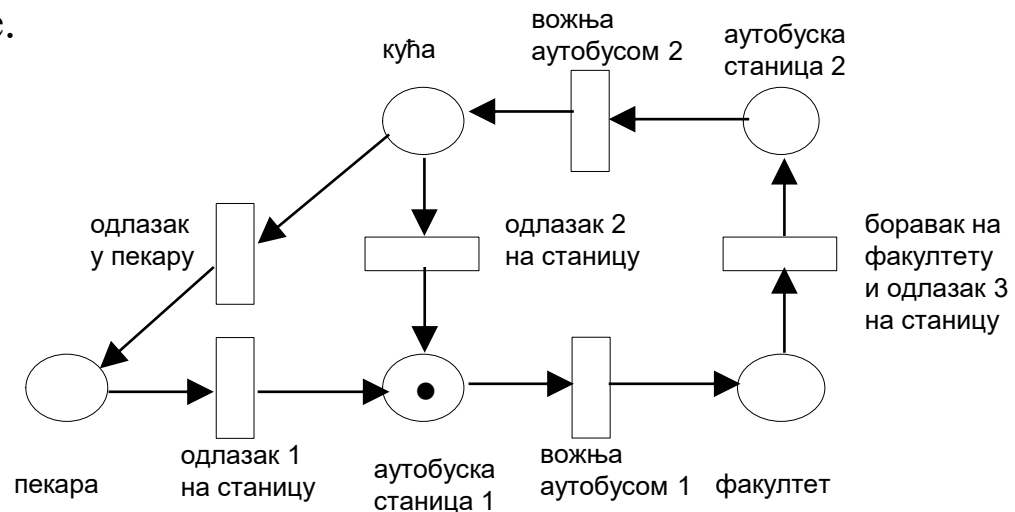
$P = \{p_1, p_2, \dots, p_n\}$ коначан скуп места (чворови графа)

$T = \{t_1, t_2, \dots, t_q\}$ коначан скуп прелаза (чворови графа)

$A \subseteq (P \times T) \cup (T \times P)$ коначан скуп грана, при чему је $(P \cap T) = \emptyset$ и $(P \cup T) \neq \emptyset$

$W: A \rightarrow \{1, 2, \dots\}$ функција тежина придружена гранама,

$M_0: P \rightarrow \{0, 1, 2, \dots\}$ почетно маркирање.



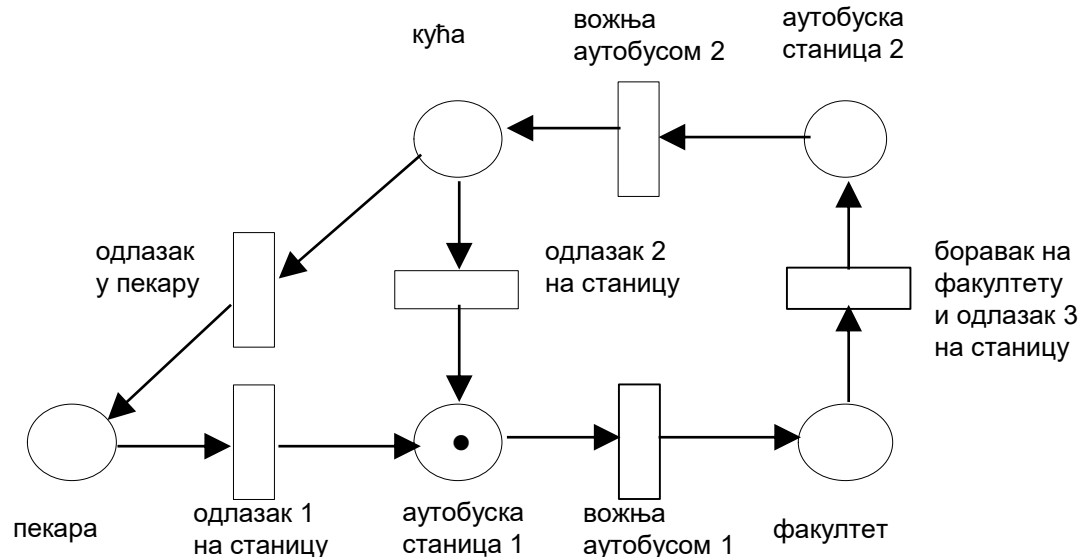
Моделирање динамике помоћу PN

${}^{\circ}t$ – скуп свих улазних места прелаза t , тј. ${}^{\circ}t = \{p \mid p \in P, (p, t) \in A\}$

t° – скуп свих излазних места прелаза t , тј. $t^{\circ} = \{p \mid p \in P, (t, p) \in A\}$

${}^{\circ}p$ – скуп свих улазних прелаза места p , тј. ${}^{\circ}p = \{t \mid t \in T, (t, p) \in A\}$

p° – скуп свих излазних прелаза места p , тј. $p^{\circ} = \{t \mid t \in T, (p, t) \in A\}$



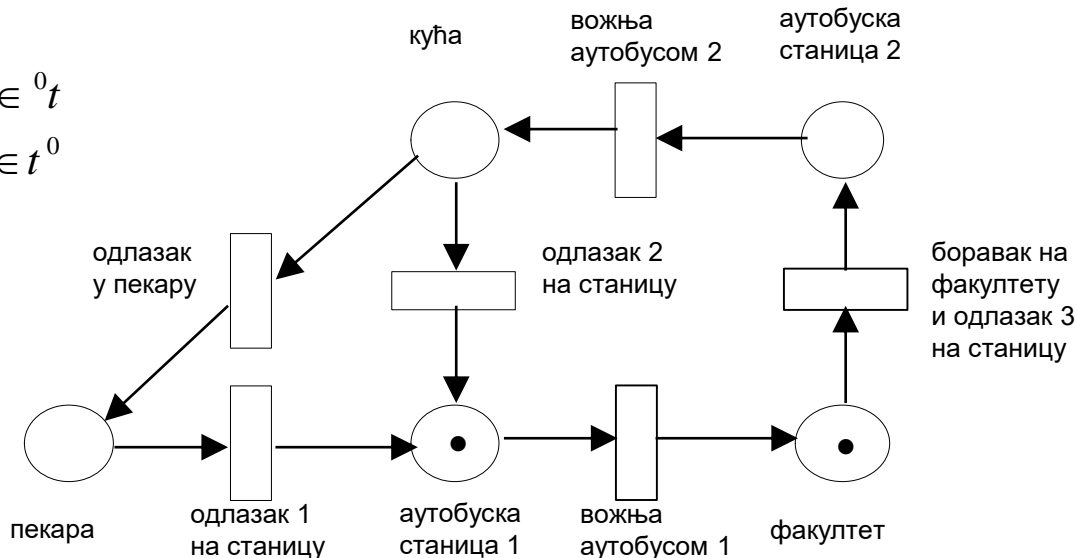
Моделирање динамике помоћу PN

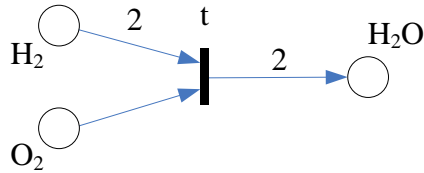
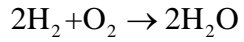
Место “аутобуска станица 1” садржи жетон.
 Прелаз “вожња аутобусом 1” је омогућен.

$$\forall p \in {}^0t, M(p) \geq W(p,t)$$

Формално, паљење прелаза је трансформација неког маркирања M у маркирање M' , дефинисана са:

$$M'(p) = \begin{cases} M(p) - W(p,t) & \text{ако } p \in {}^0t \\ M(p) + W(p,t) & \text{ако } p \in t^0 \\ M(p) & \text{иначе} \end{cases}$$





$$PN = (P, T, A, W, M_0)$$

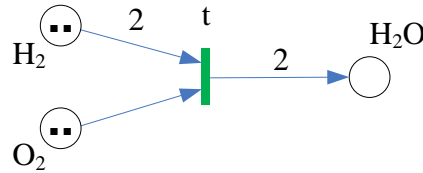
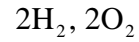
$$P = \{\text{H}_2, \text{O}_2, \text{H}_2\text{O}\}$$

$$T = \{t\}$$

$$A = \{(\text{H}_2, t), (\text{O}_2, t), (t, \text{H}_2\text{O})\}$$

$$W = [2, 1, 2]$$

$$M_0 = [0, 0, 0]$$



$$\forall p \in {}^{\circ}t, M(p) \geq W(p, t)$$

$$M_0 = [2, 2, 0]$$

$${}^{\circ}t = \{\text{H}_2, \text{O}_2\}$$

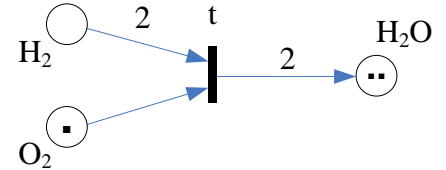
$$M(\text{H}_2) = 2, M(\text{O}_2) = 2$$

$$W(\text{H}_2, t) = 2, W(\text{O}_2, t) = 1$$

$$\Rightarrow M(\text{H}_2) \geq W(\text{H}_2, t)$$

$$\Rightarrow M(\text{O}_2) \geq W(\text{O}_2, t)$$

\Rightarrow прелаз t је омогућен



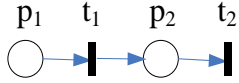
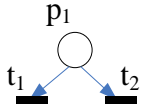
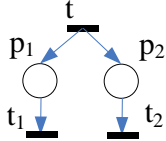
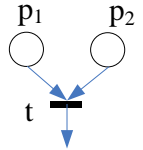
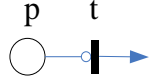
$$M'(p) = \begin{cases} M(p) - W(p, t) & \text{ако } p \in {}^{\circ}t \\ M(p) + W(p, t) & \text{ако } p \in t^{\circ} \\ M(p) & \text{иначе} \end{cases}$$

$$M'(\text{H}_2) = M(\text{H}_2) - W(\text{H}_2, t) = 2 - 2 = 0$$

$$M'(\text{O}_2) = M(\text{O}_2) - W(\text{O}_2, t) = 2 - 1 = 1$$

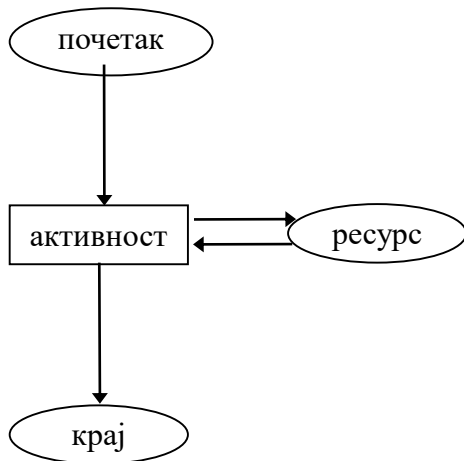
$$M'(\text{H}_2\text{O}) = M(\text{H}_2\text{O}) + W(t, \text{H}_2\text{O}) = 0 + 2 = 2$$

Стандардни модели зависности

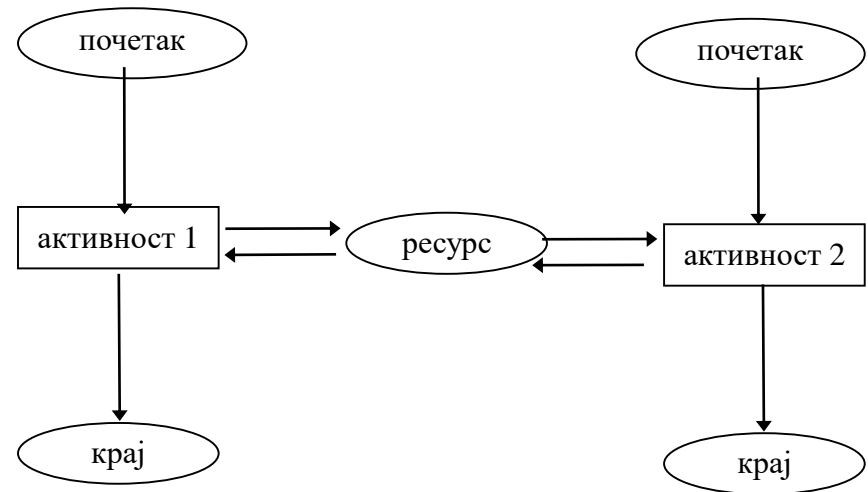
Примитив	Услов	Приказ
Секвенца	$t_1 \in p_1^0, t_1 \in {}^0 p_2, t_2 \in p_2^0$ $M(p_1) \neq \emptyset$	
Конфликт	$t_1, t_2 \in p^0$ $M(p) \neq \emptyset$	
Истовременост (concurrency)	$p_1, p_2 \in t^0, t_1 \in p_1^0, t_2 \in p_2^0$	
Синхронизација	$p_1, p_2 \in {}^0 t$ $M(p_1) \neq \emptyset, M(p_2) \neq \emptyset$	
Инхибитор	$p \in {}^0 t, M(p) = \emptyset$	

Стандардни модели зависности

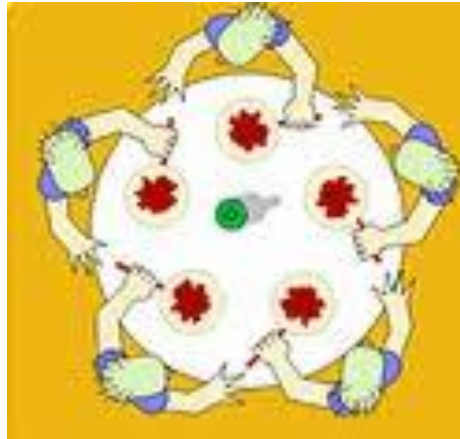
коришћење ресурса



дељење ресурса



Услови за Deadlock



1. Узајамно искључење (*Mutual exclusion*) – само један процес може да користи ресурс у једном тренутку.
2. Држи и чекај (*Hold and wait*) - процес чува ресурс и чека следећи.
3. Нема права пречег (*No preemption*) – заузети ресурс се не може преотети. Процес добровољно ослобађа ресурс након извршења.
4. Кружно чекање (*Circular wait*) – мора постојати ланац од најмање два процеса где сваки чека ресурс заузет суседним процесом.

