

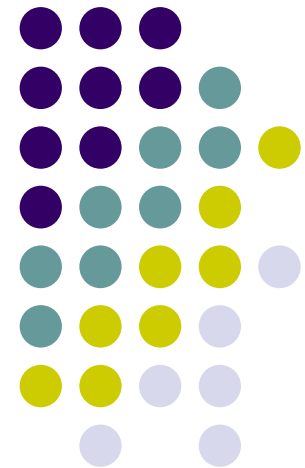
Lekcija 14:

Modeliranje distribuiranih sistema Petrijevim mrežama

Prof.dr.sc. Jasmin Velagić
Elektrotehnički fakultet Sarajevo

Kolegij: Distribuirani sistemi

2012/2013



Sadržaj poglavlja:

+ Modeliranje sistema Petrijevim mrežama

- Uvod

- Postupak sinteze GSPN-a

- Primjer sinteze sistema GSPN-om

- Analiza rada sistema

- Funkcionalna analiza

- Analiza performansi



14.1. Uvod

- ❑ Sinteza Petrijevih mreža za modeliranje, analizu i ispitivanje performansi složenih dinamičkih sistema.
- ❑ Ovo je omogućeno algoritmom koji prevodi gramatički prikaz složenih dinamičkih sistema u ekvivalentni model generalizirane stohastičke Petrijeve mreže.
- ❑ Složeni dinamički sistemi obuhvaćaju: **automatizirane proizvodne sisteme, komunikacijske sisteme, montažne i proizvodne linije, sisteme upravljanja zračnim prometom, elektrane, navigacijske sisteme, itd.**
- ❑ Tradicionalno, složeni dinamički sistemi mogu se modelirati kao diskretni dinamički sistemi, upotrebom Markovljevih lanaca, Petrijevih mreža i drugih postupaka.



Uvod

- ❑ Petrijeve mreže predstavljaju veoma moćan alat za modeliranje i analizu složenih dinamičkih sistema, ali je njihova upotrebljivost jako smanjena zbog **složenosti sadržanih u njihovom kreiranju, rukovanju i analizi.**
- ❑ Da bi se savladale navedene složenosti, Petrijeve se mreže mogu modificirati na jedan od sljedeća dva načina:
 - Njihova izrazita moć je smanjena ograničenjem na model obične Petrijeve mreže, koji lahko analizira svojstva zadanog sistema.
 - Njihova izrazita moć je proširena dodacima na model obične Petrijeve mreže, koji usložnjavaju analizu modela.



Uvod

- ❑ Međutim, niti ograničenja niti proširenja Petrijeve mreže ne mogu efektivno smanjiti nastojanje zahtijevano tokom složenog raščlanjivanja složenih dinamičkih sistema u njihove ekvivalentne Petrijeve mreže.
- ❑ Zhou je ukazao na snažnu potrebu za automatiziranom sintezom Petrijeve mreže kako bi se proširila upotrebljivost analize temeljene na Petrijevim mrežama.



Uvod

Karakteristike dinamičkih sistema

- ❑ **Ovisnosti**: privremene i logičke.
- ❑ **Paralelizam i konkurentnost** (istovremenost) : sposobnost istovremenog izvođenja više zadataka.
- ❑ **Složena sinhronizacija** (otežana analiza korektnosti i performansi).
- ❑ **Sekvencijalizam**: nametanje izvršavanja redosljeda obavljanja zadataka (naprimjer, montaža prije pakiranja).
- ❑ **Sukobljavanje**: nastaje usljed ovisnosti, paralelizma i sekvencijalizma (dva podsistema ne mogu obavljati posao sa istim komponentama u isto vrijeme).



Uvod

Karakteristike dinamičkih sistema

- ❑ Moderni računarski sistemi mogu obavljati operacije na **paralelan** (konkurentan) i **distribuiran** način, za razliku od ranijih sistema koji su obavljali operacije u sekvencijalnom (uzastopnom) poretku.
- ❑ Konkurentnost i sinhronizacija su u takvim sistemima višestruko uvećane i kao posljedica toga, otežana je analiza njihove korektnosti i performansi.
- ❑ **Moderni paralelni i distribuirani sistemi uključuju u sebi složenu sinhronizaciju. Zbog toga je potrebno, prije implementacije sistema, dizajnirati model na kojem bi se odredile performanse i korektnost sistema.**



Uvod

Karakteristike dinamičkih sistema

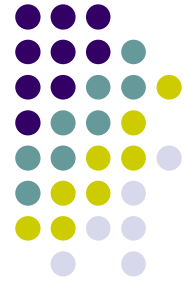
- ❑ Postoji nekoliko metoda i pristupa modeliranja i analize složenih dinamičkih sistema, kao što su:
 - ❑ **operacijska istraživanja,**
 - ❑ **metode temeljene na umjetnoj inteligenciji,**
 - ❑ **hibridni sistemi,**
 - ❑ **Markovljevi lanci (članci),**
 - ❑ **formalne metode,**
 - ❑ **objektno-orijentirane metode**
 - ❑ **i konačno Petrijeve mreže.**
- ❑ Fleksibilni proizvodni sistemi se mogu modelirati pomoću kombinacije Petrijeve mreže, mreže čekanja i različitih programskih pristupa za matematički opis sistema.



Uvod

Potrebe za modelima Petrijevih mreža

- ❑ Petrijeve mreže imaju sljedeće prednosti u odnosu na druge metodologije, kao što su strojevi konačnih stanja, formalni jezici ili konačni rekurzivni procesi:
 - **Sposobnost generiranja upravljačkog koda, tj. prevođenja Petrijeve mreže u Petrijev regulator.**
 - **Petrijeve mreže mogu lahko modelirati DEDS karakteristike, odnosno, Petrijeve mreže su vrlo pogodne za modeliranje konkurentnosti, sinhronih i asinhronih karakteristika, sukobljavanja, uzajamne isključivosti, relacija prednosti, nedeterminizama i zastoja sistema.**
 - **Petrijeve mreže predstavljaju dobru vizualizaciju sistemskih ovisnosti.**



Uvod

Potrebe za modelima Petrijevih mreža

- **Mogućnost detektiranja neželjenih svojstava sistema, kao što su zastoje i nestabilnost.**
 - **Analiza performansi se može obaviti, sa ili bez simulacije, kako bi se postigli porast proizvodnje, korištenja resursa i pouzdanosti.**
 - **Simulacija diskretnih događaja može se obaviti na modelu Petrijeve mreže.**
- Petrijeve mreže su također korisne u planiranju jer one mogu inherentno prikazati relacije prednosti, ograničenja, te mogu sadržavati informacije o vremenu.



Uvod

Potrebe za modelima Petrijevih mreža

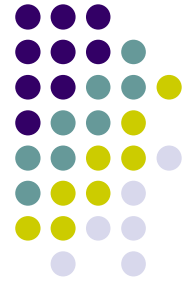
- ❑ Generalizirane stohastičke Petrijeve mreže osim navedenih prednosti imaju i **sposobnost prikaza vremenskih informacija**, što im omogućuje analizu performansi zadanog složenog dinamičkog sistema.
- ❑ Analiza smetnji, teorija mreža čekanja i Markovljevi procesi koristili su se u analizi DEDS modela (uključujući modele Petrijevih mreža).
- ❑ Međutim, mreže čekanja i Petrijeve mreže mogu modelirati sadržaj resursa, ali jedino Petrijeve mreže mogu modelirati **sinhronizaciju između istovremenih aktivnosti**.



Uvod

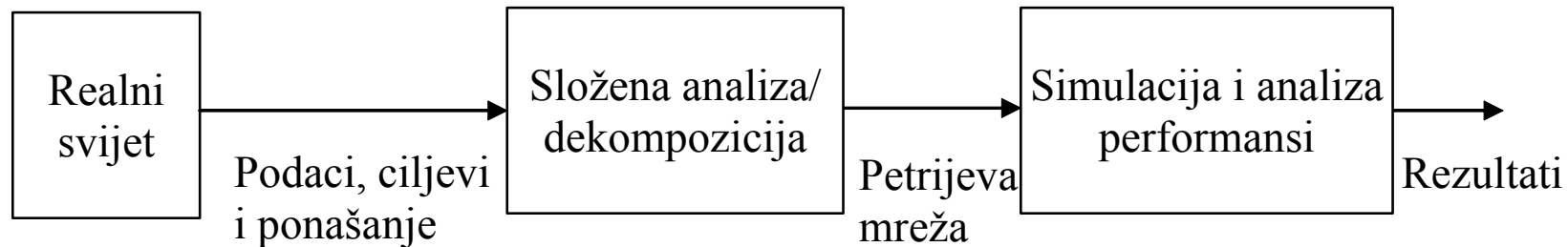
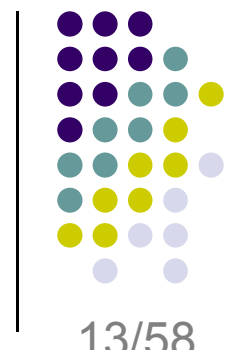
Potrebe za modelima Petrijevih mreža

- Dakle, Petrijeve mreže su formalizam i alat za izbor mogućnosti sistema kako bi se prikazala njihova statička i dinamička svojstva, kao i grafički i formalno ispravni prikaz.
- Raščlanjivanje složenih dinamičkih sistema često može proizvesti ogromne i složene modele Petrijevih mreža.
- Postupci koji vode ka složenim i nerazumljivim modelima su skloniji ljudskim greškama i stoga, manje pouzdani.



Uvod

Standardno modeliranje zasnovano na Petrijevim mrežama



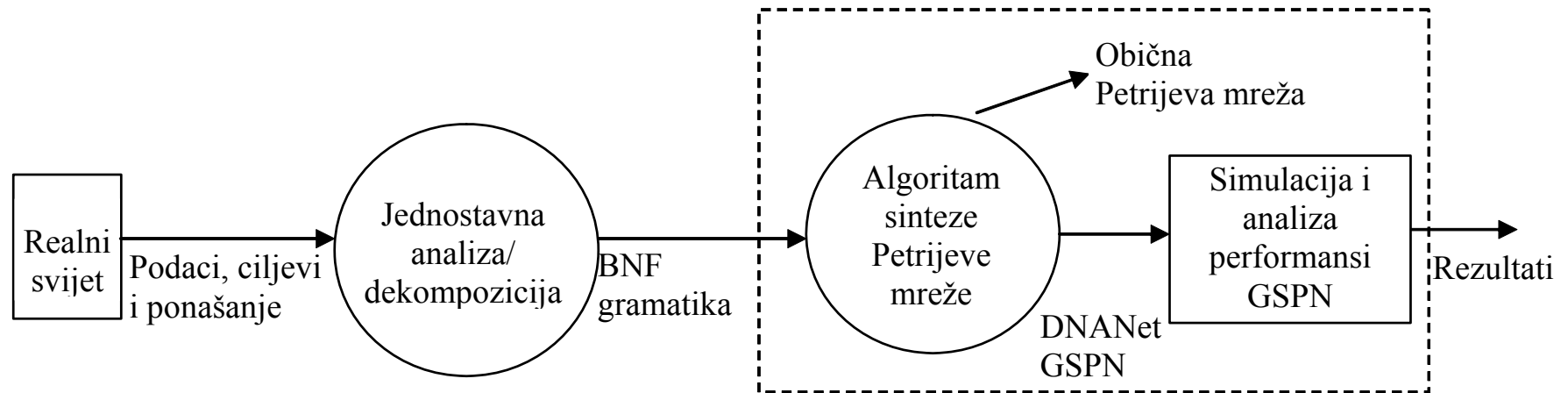
- ❑ **Problem: model sistema se dobiva složenom dekompozicijom obične Petrijeve mreže.**
- ❑ **Kako izbjeći složenu dekompoziciju?**
- ❑ **Složeni dinamički sistem se prvo modelira upotrebom gramatike, a zatim, primjenom odgovarajućeg algoritma prevodi u model Petrijeve mreže.**
- ❑ **Ovaj model se može implementirati u nekom od postojećih softverskih alata i pomoću njega obaviti analiza i ispitivanje performansi složenih dinamičkih sistema.**

Uvod

Poboljšani postupak modeliranja zasnovan na GSPN-u



- Poboljšani postupak, prikazan na slici ispod, lahko kreira, manipulira i analizira složeni dinamički sistem pojednostavljenjem složene dekompozicije u jednostavnu dekompoziciju, koji se nakon toga automatski prevodi u GSPN model.



Rezultirajuća Petrijeva mreža se može zatim simulirati i analizirati upotrebom softverskih alata kao što je **DNANet** (University of Cape Town, South Africa).

- Legenda:
- Predloženi postupci/alati
 - Postojeći postupci/alati
 - ⋯ Automatizirane procedure

14.2. Postupak sinteze GSPN-a

Modeliranje sistema GSPN mrežom

- ❑ Da bi se obavila analiza i sinteza dinamičkog sistema potrebno je načiniti **generički model** pomoću Petrijeve mreže.
- ❑ Ovdje su dane formalne definicije vezane za dinamički sistem.
- ❑ Na temelju tih definicija kreira se ekvivalentna gramatika koja se primjenom odgovarajućih algoritama prevodi u model Petrijeve mreže.
- ❑ **BNF** (Backus Naur Form, John Backus & Peter Naur) gramatika omogućuje opis specifikacija i predstavljanje sistema.



Postupak sinteze GSPN-a

Modeliranje sistema GSPN mrežom

- Definiranje modela dinamičkog sistema.
- Dekompozicija sistema.
- Formiranje BNF gramatike.
- Izbor algoritma sinteze.
- Sinteza Petrijeve mreže iz gramatičkih specifikacija primjenom algoritma sinteze.
- Funkcionalna analiza, ispitivanje korektnosti rada Petrijeve mreže.
- Simulacija Petrijeve mreže.
- Analiza performansi.



Postupak sinteze GSPN-a

Definiranje modela i dekompozicija sistema

□ Određivanje:

- skupa podsistema, *US*
- skupa zadataka, *UT*
- skupa procesa, *UP*
- matrice sposobnosti, *CPM*
- matrice povezanosti, *CTM*
- matrice zadaćne prednosti, *TPM*
- vremena očekivanja, *TD*
- srednjeg vremena čekanja, *MWT*
- funkcije gubitaka, *CF*
- vjerojatnosti kvara, *FAIL*
- vremena popravka kvara, *RPT*



Postupak sinteze GSPN-a

Generički model



Def. (Sistem) Sistem US sastoji se od skupa podsistema S_i :

$$US = \{S_1, S_2, \dots, S_i, \dots, S_{|US|}\}, \quad 1 \leq i \leq |US|$$

Def. (Zadaci) UT je skup svih primitivnih zadataka T_j koje sistem US može obaviti:

$$UT = \{T_1, T_2, \dots, T_j, \dots, T_{|UT|}\}, \quad 1 \leq j \leq |UT|$$

Def. (Procesi) UP je skup svih procesa P_k koje sistem US može izvršiti:

$$UP = \{P_1, P_2, \dots, P_k, \dots, P_{|UP|}\}, \quad 1 \leq k \leq |UP|$$

Postupak sinteze GSPN-a

Generički model

Svaki se proces P_k može raščlaniti u niz primitivnih, neponavljajućih zadataka T_{k_j} :

$$P_k = \{T_{k_1}, T_{k_2}, \dots, T_{k_j}, \dots, T_{k_n}\}, \quad n \geq 1, \quad T_{k_j} \in UT.$$

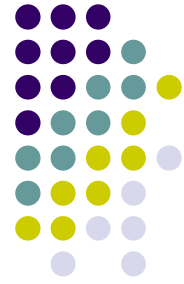
Def. (Matrica sposobnosti) Matrica sposobnosti CPM označava sposobnost svakog podsistema S_i da obavi zadani primitivni zadatak T_j :

i

$$CPM : US \times UT \rightarrow \mathbf{N}$$

$$CPM(S_i, T_j) = \tau_{ij}, \quad \tau_{ij} \in \mathbf{N}, \quad S_i \in US, \quad T_j \in UT$$

gdje je $\tau_{ij} > 0$ ako podsistem S_i može obaviti zadatak T_j u vremenu τ_{ij} , u ostalim slučajevima je $\tau_{ij} = 0$ (podsistem S_i ne može obaviti zadatak T_j).



Postupak sinteze GSPN-a

Generički model

Def. (Matrica povezanosti) Matrica povezanosti CTM definira podsistemsku vezu između podsistema S_{i_1} i S_{i_2}

$$CTM: US \times US \rightarrow \{0,1\}$$

i

$$CTM(S_{i_1}, S_{i_2}) = 1$$

ako je izlaz podsistema S_{i_1} povezan sa ulazom podsistema S_{i_2} , u suprotnom vrijedi:

$$CTM(S_{i_1}, S_{i_2}) = 0$$

Def. (Matrica zadaćne prednosti) Matrica zadaćne prednosti TPM definira uzročno-posljedične veze između dva primitivna zadatka T_{j_1} i T_{j_2} , tako da se zadatak T_{j_1} izvrši prije nego započne zadatak T_{j_2} :

$$TPM: UT \times UT \rightarrow \{0,1\}$$



Postupak sinteze GSPN-a

Generički model

Ako su zadaci T_{j_1} i T_{j_2} neovisni, tada vrijedi:

$$TPM(T_{j_1}, T_{j_2}) = 0$$

u suprotnom:

$$TPM(T_{j_1}, T_{j_2}) = 1$$

Def. (Vrijeme očekivanja) Vrijeme očekivanja $TD: UP \rightarrow \mathbf{N}$ je definirano za svaki proces P_k .

$TD(P_k)=t_d$ znači da je proces P_k kompletiran za vrijeme t_d ili prije njega.

Postupak sinteze GSPN-a

Generički model



Def. (Srednje vrijeme čekanja) Srednje vrijeme čekanja MWT definirano je sa $MWT : US \rightarrow \mathbf{N}$ za svaki podsistem S_i .

22/58

Ono označava zahtijevano vrijeme za svaki podsistem da bude ponovno dostupan nakon kompletiranja prethodne zadaće.

Def. (Vjerojatnost kvara) Vjerojatnost kvara $FAIL : US \rightarrow [0,1]$ pridružena je svakom podsistemu S_i . Ona označava slučajnu vjerojatnost kvara sistema.

Def. (Vrijeme popravka) Vrijeme popravka $RPT : US \rightarrow \mathbf{N}$ definirano je za svaki podsistem S_i .

Ono označava prosječno vrijeme zahtijevano za popravak podsistema (koji je bio u kvaru).

Postupak sinteze GSPN-a

BNF gramatika za formalni prikaz sistema



23/58

<system>	→ system { <sys_name> <sys_entry> }
<sys_name>	→ <string>
<sys_entry>	→ <subsystem> <task> <process> <cpm> <ctm> <tpm>
<subsystem>	→ subsystem { (<subsys_entry>) }
<subsys_entry>	→ <sid> <sname> <cost> <mwt> <fail> <repair_time>
<sid>	→ <string>
<sname>	→ <string>
<cost>	→ <float>
<mwt>	→ <integer>
<fail>	→ <percentage>
<repair_time>	→ <integer>
<task>	→ task { (<task_entry>) }
<task_entry>	→ <tid> <tname>
<tid>	→ <string>
<tname>	→ <string>
<process>	→ process { (<process_entry>) }
<process_entry>	→ <pid> <pname> <due_time> { (<task_pair>) }

Postupak sinteze GSPN-a

BNF gramatika za formalni prikaz sistema



24/58

<pid>	→ <string>
<pname>	→ <string>
<due_time>	→ <integer>
<task_pair>	→ (<tid_pre> , <tid_post>)
<tid_pre>	→ <tid>
<tid_post>	→ <tid>
<cpm>	→ cpm { (<cpm_entry>) }
<cpm_entry>	→ <tid> <sid> <cpm_time>
<cpm_time>	→ <integer>
<ctm>	→ ctm { (<ctm_entry>) }
<ctm_entry>	→ <sid_out> <sid_in>
<sid_out>	→ <sid>
<sid_in>	→ <sid>
<integer>	→ bilo koja cjelobrojna konstanta u ANSI C zapisu
<float>	→ bilo koja realna konstanta u ANSI C zapisu
<percentage>	→ bilo koja realna konstanta u ANSI C zapisu između 0 i 1

Postupak sinteze GSPN-a

BNF gramatika za formalni prikaz sistema



<string> → “ (<character>) ”

<character> → **A | B | C | D | E | F | G | H | I | J | K | L | M |**
N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
a | b | c | d | e | f | g | h | i | j | k | l | m |
n | o | p | q | r | s | t | u | v | w | x | y | z |
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
(|) | { | } | . | , | / | | : | _ | + | - | * | =

Napomena:

U BNF gramatici, terminali su označeni podebljano (**bold**), neterminali su smješteni između trokutastih zagrada <>, konačna iteracija je specificirana sa (...) za nula ili više iteracija i sa (...) za jednu ili više iteracija. Standardna BNF gramatika je modificirana zamjenom znaka := sa → i označavanjem terminala podebljano. C++ oblici komentara (//) su omogućeni. Početni simbol (startni) gramatike je <system>.

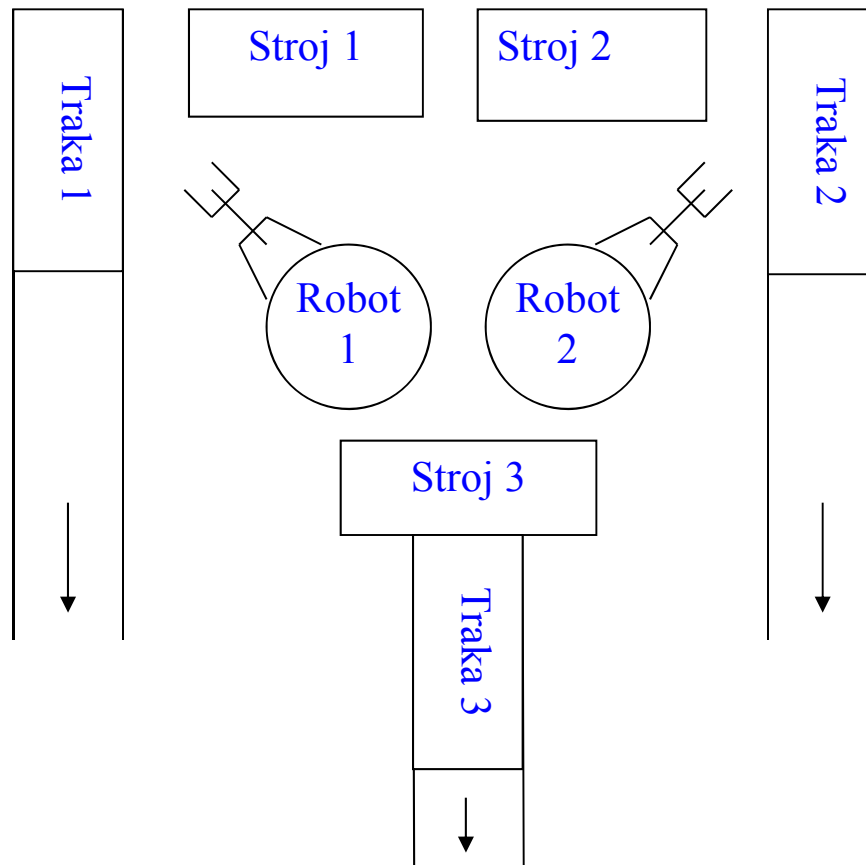
14.3. Primjer sinteze sistema GSPN-om

Fleksibilni proizvodni sistem

Fleksibilni proizvodni sistem predstavlja skupinu povezanih automatiziranih strojeva i alata koji operiraju kao integrirani sistem sa računarskim upravljanja.



26/58



Dvije različite operacije:

1. Obrada dijelova
2. Montaža dijelova

Primjer sinteze sistema GSPN-om

Formalna dekompozicija

Skup podsistema

$US = \{S1, S2, S3, S4, S5, S6, S7, S8\}$

- S1 – Proizvodna traka 1
- S2 – Stroj 1
- S3 – Robot 1
- S4 – Proizvodna traka 2
- S5 – Stroj 2
- S6 – Robot 2
- S7 – Stroj 3
- S8 – Proizvodna traka 3



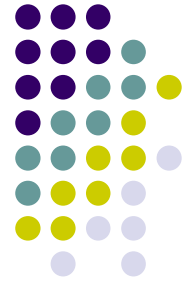
Primjer sinteze sistema GSPN-om

Formalna dekompozicija

Skup primitivnih zadataka

$UT = \{T1, T2, T3, T4, T5, T6, T7, T8, T9, T10\}$

- T1 – Postavljanje dijela A u startnu poziciju
- T2 – Prenosjenje dijela A do Stroja 1
- T3 – Obrada dijela A
- T4 – Prenosjenje dijela A do Stroja 3
- T5 – Postavljanje dijela B u startnu poziciju
- T6 – Prenosjenje dijela B do Stroja 2
- T7 – Obrada dijela B
- T8 – Prenosjenje dijela B do Stroja 3
- T9 – Montaža (spajanje) dijelova A i B
- T10 – Isporučivanje gotovog proizvoda



Primjer sinteze sistema GSPN-om

Formalna dekompozicija

Skup procesa

$UP = \{P1, P2, P3\}$

gdje su:

- P1 – obrada dijela A
- P2 – obrada dijela B
- P3 – montaža dijelova A i B

i

- $P1 = \{T1, T2, T3, T4\}$
- $P2 = \{T5, T6, T7, T8\}$
- $P3 = \{T9, T10\}$



Primjer sinteze sistema GSPN-om

Formalna dekompozicija

Matrica sposobnosti

- označava sposobnosti svakog pojedinačnog podsistema S_i da obavi zadani primitivni zadatak T_j u τ_{ij} vremenskih jedinica.

Task	Subsystem							
	S1	S2	S3	S4	S5	S6	S7	S8
T1	3							
T2			2					
T3		10						
T4			3.5					
T5				3				
T6						2		
T7					8			
T8						3.5		
T9							6	
T10								4

Primjer sinteze sistema GSPN-om

Formalna dekompozicija

Ostala važna svojstva formalnog modela su:

- **Vrijeme očekivanja (due time (DT)).**
- **Funkcija kakvoće (cost function (COST)).**
- **Srednje vrijeme čekanja (mean wait time (MWT)).**
- **Vjerojatnost kvara (failure probability (FAIL)).**
- **Vrijeme popravka, odnosno oporavka (repair time (RPT)).**



Primjer sinteze sistema GSPN-om

BNF gramatički opis



34/58

System { “Manufacturing process”

//

// BNF grammar

//

Subsystem {

//	SID	SNAME	COST	MWT	FAIL	REPAIR
	“S1”	“Conveyor 1”	1	0.1	0.02	10
	“S2”	“Machine 1”	2	0.5	0.05	30
	“S3”	“Robot 1”	3	0.5	0.03	60
	“S4”	“Conveyor 2”	1	0.1	0.02	10
	“S5”	“Machine 2”	2	0.5	0.04	25
	“S6”	“Robot 2”	3	0.5	0.03	60
	“S7”	“Machine 3”	2	0.5	0.05	30
	“S8”	“Conveyor 3”	1	0.1	0.02	10

{

Primjer sinteze sistema GSPN-om

BNF gramatički opis

Task {

- “T1” “Setting the part A in start position”
- “T2” “Loading the part A to Machine 1”
- “T3” “Processing the part A”
- “T4” “Load the part A to Machine 3”
- “T5” “Setting the part B in start position”
- “T6” “Loading the part B to Machine 2”
- “T7” “Processing the part B”
- “T8” “Loading the part B to Machine 3”
- “T9” “Assembling the parts”
- “T10” “Delivering the final product”

}



Primjer sinteze sistema GSPN-om

BNF gramatički opis



36/58

Process {

```
//  PID      PNAME          DUE_TIME    TPM
    "P1"     "Processing the part A"    18.5        {
                ("T1","T2") ("T2","T3") ("T3","T4") ("T4","T9")
            }
    "P2"     "Processing the part B"    16.5        {
                ("T5","T6") ("T6","T7") ("T7","T8") ("T8","T9")
            }
    "P3"     "Assembling the parts A and B"  10          {
                ("T9","T10")
            }
}
```

Primjer sinteze sistema GSPN-om

BNF gramatički opis



37/58

```
Cpm { //Capability Matrix
    //
    TID    SID    CPM_TIME
    "T1"   "S1"   3
    "T2"   "S3"   2
    "T3"   "S2"   10
    "T4"   "S3"   3.5
    "T5"   "S4"   3
    "T6"   "S6"   2
    "T7"   "S5"   8
    "T8"   "S6"   3.5
    "T9"   "S7"   6
    "T10"  "S8"   4
}
```

Primjer sinteze sistema GSPN-om

BNF gramatički opis



38/58

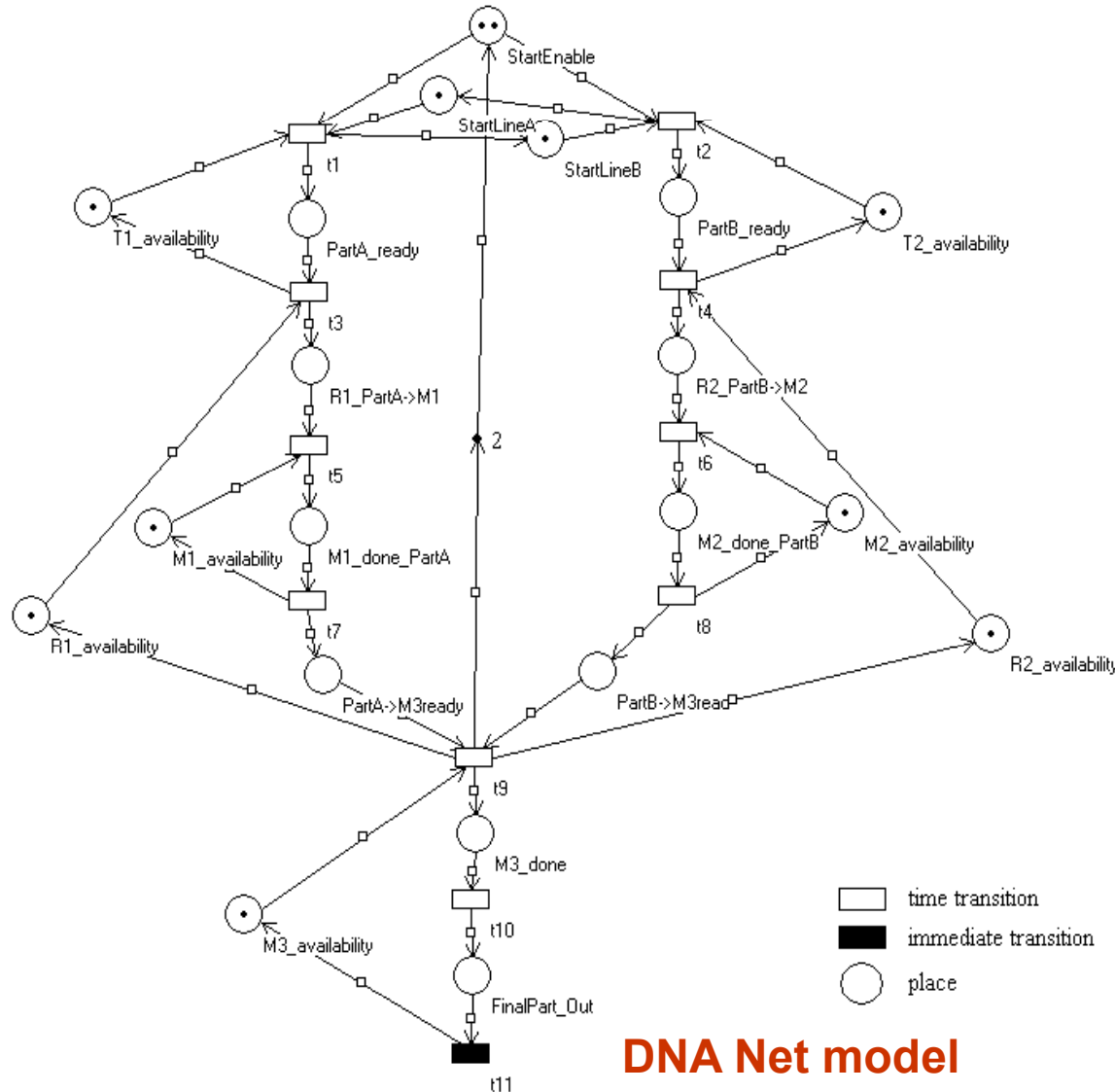
```
Ctm { //Connectivity Matrix
    // SIDout SIDin
    "S1" "S3"
    "S2" "S3"
    "S3" "S2"
    "S3" "S7"
    "S4" "S6"
    "S5" "S6"
    "S6" "S5"
    "S6" "S7"
    "S7" "S8"
}
```

Primjer sinteze sistema GSPN-om

GSPN model fleksibilnog proizvodnog sistema



39/58



DNA Net model

StartEnable, StartLineA, StartLineB – control places to enable start of process and two manufacturing lines. Those places also disable a conveyor to start twice in a row.

PartA_ready – Conveyor 1 loading the part A into Robot 1 workspace

T1_availability – availability of Conveyor 1

R1_PartA->M1 – Robot 1 unloading Conveyor 1 and moving the part A to Machine 1

M1_done_PartA – Machine 1 processed the part A

M1_availability – availability of Machine 1

R1_availability – availability of Robot 1

PartA->M3ready – Robot 1 is ready to moving the part A to Machine 3

PartB_ready – Conveyor 2 loading the part B into Robot 2 workspace

T2_availability – availability of Conveyor 2

R2_PartB->M2 – Robot 2 unloading Conveyor 2 and moving the part B to Machine 2

M2_done_PartB – Machine 2 processed the part B

M2_availability – availability of Machine 2

R2_availability – availability of Robot 2

PartB->M3ready – Robot 2 is ready to moving the part B to Machine 3

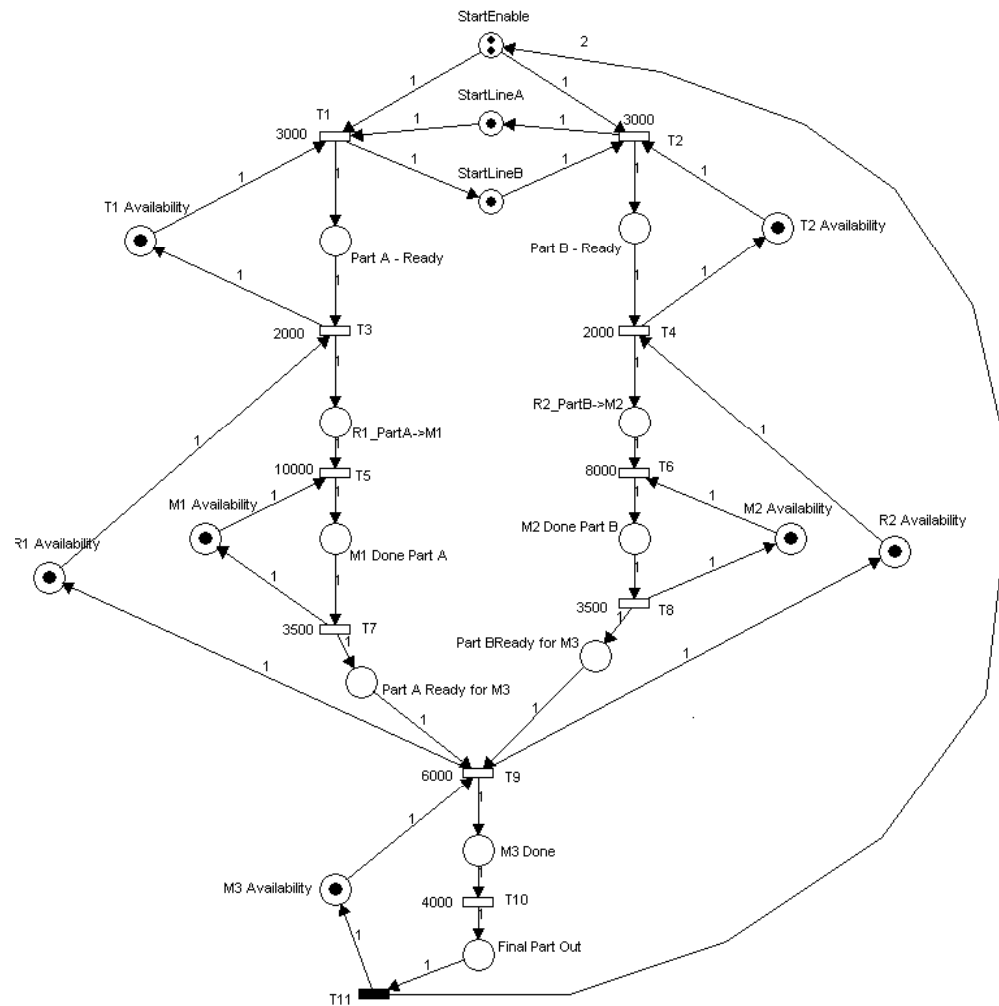
M3_done – Machine 3 assembled the part A with the part B

M3_availability – availability of Machine 3

FinalPart_Out – moving the final product to the Conveyor 3

Primjer sinteze sistema GSPN-om

GSPN model fleksibilnog proizvodnog sistema



HP Sim model

StartEnable, StartLineA, StartLineB – control places to enable start of process and two manufacturing lines. Those places also disable a conveyor to start twice in a row.

PartA_ready – Conveyor 1 loading the part A into Robot 1 workspace

T1_availability – availability of Conveyor 1

R1_PartA->M1 – Robot 1 unloading Conveyor 1 and moving the part A to Machine 1

M1_done_PartA – Machine 1 processed the part A

M1_availability – availability of Machine 1

R1_availability – availability of Robot 1

PartA->M3ready – Robot 1 is ready to moving the part A to Machine 3

PartB_ready – Conveyor 2 loading the part B into Robot 2 workspace

T2_availability – availability of Conveyor 2

R2_PartB->M2 – Robot 2 unloading Conveyor 2 and moving the part B to Machine 2

M2_done_PartB – Machine 2 processed the part B

M2_availability – availability of Machine 2

R2_availability – availability of Robot 2

PartB->M3ready – Robot 2 is ready to moving the part B to Machine 3

M3_done – Machine 3 assembled the part A with the part B

M3_availability – availability of Machine 3

FinalPart_Out – moving the final product to the Conveyor 3

14.4. Analiza rada sistema

Funkcionalna analiza i analiza performansi



41/58

1. Funkcionalna analiza ili analiza korektnosti

Određivanje funkcionalnih svojstava sistema (aktivnost, ograničenost, prekrivanje, blokada, itd)

a) analiza invarijantnosti (algebarski postupci)

b) analiza pomoću grafa prekrivanja (istraživanje svih prekrivenih stanja sistema)

2. Analiza performansi

Mjera kvalitete rada sistema

a) analiza Markovljevim lancem (egzaktan analitički postupak)

b) analiza simulacijom (estimacija nadziranjem neinteraktivne animacije mreže).

Analiza rada sistema

Funkcionalna analiza i analiza performansi

1. Funkcionalna analiza

- a) detekcija blokada
- b) pronalaženje neograničenih ciklusa
- c) provjeravanje da li označavanje odgovara semantički korektnim stanjima sistema

2. Analiza performansi

- a) prosječan broj oznaka po mjestu
- b) puna distribucija oznaka
- c) vjerojatnost omogućenosti prijelaza
- d) propusnost prijelaza
- e) Steady state distribucija



Analiza rada sistema

Funkcionalna analiza i analiza performansi

□ Određivanje funkcionalnih svojstava sistema

- Aktivnost
- Ograničenost
- Zastoj
- Home states

□ Metode funkcionalne korektnosti modela Petrijeve mreže

□ Određuju funkcionalnu korektnost modela Petrijeve mreže

□ Analiza grafom prekrivanja

□ Analiza invarijantnosti (structurally-based technique):

- P- invarijantna analiza
- T-invarijantna analiza



14.5. Funkcionalna analiza

Analiza invarijantnosti

- ❑ Analiza invarijantnosti koristi linearne algebarske tehnike za određivanje invarijantni tvrdnji koje su validne za sva inicijalna označavanja mreže.
- ❑ Općenito **invarijantnosti mjesta** specificiraju težinsku sumu oznaka (tokens) koja ostaje konstantna za sva moguća, izvodiva označavanja mreže.
- ❑ **Invarijantnosti prijelaza** specificiraju redno neovisne kružne sekvence prijelaza koje omogućuju da označavanje mreže ostane nepromijenjeno.
- ❑ Prekrivanje svih mjesta sa **P-invarijantnostima** je dovoljan uvjet ograničenosti modela Petrijeve mreže.
- ❑ Postojanje **T-invarijantnosti** prekriva sve prijelaze u mreži i predstavlja nužan, ali ne i dovoljan uvjet, aktivnosti (živosti) mreže.



Funkcionalna analiza

Analiza invarijantnosti

- Invarijantna analiza koristi matricu slučajeva (dogadaja) \mathbf{C} čiji su elementi definirani na sljedeći način:

$$C_{ij} = I^+(p_i; t_j) - I^-(p_i; t_j)$$

$$\forall p_i \in P, t_j \in T$$

- Dopustivo označavanje μ iz inicijalnog označavanja μ_0 je:

$$\mu = \mu_0 + \mathbf{C}f \quad (*)$$



Funkcionalna analiza

Analiza invarijantnosti

gdje $\mathbf{f} = f_1, \dots, f_{|T|}$ vektor brojeva vremenskih prijelaza t_i propaljenih u sekvenci od μ_0 do μ .



□ Množenjem obje strane jednačbe (*) slijeva sa $\mathbf{v}^T \in Z^n$:

$$\mathbf{v}^T \boldsymbol{\mu} = \mathbf{v}^T \boldsymbol{\mu}_0 + \mathbf{v}^T \mathbf{C} \mathbf{f}$$

Pronalaženje P-invarianata uključuje rješavanje po \mathbf{v} -u jednačbe:

$$\mathbf{v}^T \mathbf{C} = \mathbf{0}$$

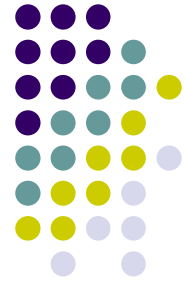
Pronalaženje T-invarianata uključuje rješavanje po \mathbf{f} -u jednačbe:

$$\mathbf{C} \mathbf{f} = \mathbf{0}$$

Funkcionalna analiza

Analiza grafom prekrivanja

- ❑ Uključuje brojanje svih dostupnih oznaka (ili označavanje prekrivanja u slučaju neograničene mreže) koristeći analizu stroge povezanosti komponenti u grafu.
- ❑ Za zadano početno (inicijalno) označavanje mreže μ_0 , sljedeće označavanje u stablu prekrivanja (dostupnosti) se pronalazi kroz prsten svakog omogućenog prijelaza.
- ❑ Iz svakog od ovih novih označavanja, postoje ponovo omogućeni prijelazi koji mogu primiti više oznaka (neki od njih su bili dostupni ranije).
- ❑ Ovaj proces rezultira grafom dostupnih oznaka (označavanja) čiji je korijen u μ_0 ; čvorovi predstavljaju oznake i grane predstavljaju prsten omogućenih prijelaza.
- ❑ Kada se pronađe graf prekrivanja, on se može analizirati kako bi se odredila ograničenost, aktivnost, zastoje i postojanje home states.



Funkcionalna analiza

Rezultati funkcionalne analize GSPN proizvodnog fleksibilnog procesa



48/58

Rezultati analize korektnosti:

Net is bounded.

Deadlock is not possible.

Net is live.

Net has home states.

Coverability graph generation statistics:

75 unique markings strongly connected components

73 hash table entries used

2 was the longest hash list length

1.027397 was the average hash list length

10 was the maximum stack height

1 was the maximum component stack height

Funkcionalna analiza

Rezultati funkcionalne analize GSPN proizvodnog fleksibilnog procesa



P-invariants:

1 0 0 0 0 0 0 0 0 0 0 1 0	(main.T1_availability)
1 0 0 0 0 0 0 1 0 0 2 0 1	(main.PartA_ready)
0 1 0 0 0 0 0 1 2 0 0 1 0	(main.PartB_ready)
0 1 0 0 0 0 0 0 0 0 0 0 1	(main.T2_availability)
0 0 0 0 0 1 0 1 0 0 2 0 1	(main.R1_PartA->M1)
0 0 0 0 1 0 0 1 2 0 0 1 0	(main.R2_PartB->M2)
0 0 1 0 0 1 0 1 0 0 2 0 1	(main.M1_done_PartA)
0 0 0 1 1 0 0 1 2 0 0 1 0	(main.M2_done_PartB)
0 0 1 0 0 0 0 0 0 0 0 0 0	(main.M1_availability)
0 0 0 1 0 0 0 0 0 0 0 0 0	(main.p15)
0 0 0 0 0 1 0 1 0 0 2 0 1	(main.PartA->M3ready)
0 0 0 0 1 0 0 1 2 0 0 1 0	(main.PartB->M3read)
0 0 0 0 1 0 0 0 0 0 0 0 1	(main.R2_availability)
0 0 0 0 0 1 0 0 0 0 0 1 0	(main.R1_availability)
0 0 0 0 0 0 1 0 0 0 0 0 0	(main.M3_done)
0 0 0 0 0 0 1 0 0 0 0 0 0	(main.FinalPart_Out)
0 0 0 0 0 0 1 0 0 0 0 0 0	(main.M3_availability)
0 0 0 0 0 0 0 1 1 0 1 0 0	(main.StartEnable)
0 0 0 0 0 0 0 0 0 1 1 0 1	(main.StartLineA)
0 0 0 0 0 0 0 0 1 1 0 1 0	(main.StartLineB)

ie.

Funkcionalna analiza

Rezultati funkcionalne analize GSPN proizvodnog fleksibilnog procesa



50/58

```
M(main.T1_availability) + M(main.PartA_ready)
M(main.PartB_ready) + M(main.T2_availability)
M(main.M1_done_PartA) + M(main.M1_availability)
M(main.M2_done_PartB) + M(main.p15)
M(main.R2_PartB->M2) + M(main.M2_done_PartB) + M(main.PartB->M3read) + M(main.R2_availability)
M(main.R1_PartA->M1) + M(main.M1_done_PartA) + M(main.PartA->M3ready) + M(main.R1_availability)
M(main.M3_done) + M(main.FinalPart_Out) + M(main.M3_availability)
M(main.PartA_ready) + M(main.PartB_ready) + M(main.R1_PartA->M1) + M(main.R2_PartB->M2) + M(main.M1_done_PartA)
+ M(main.M2_done_PartB) + M(main.PartA->M3ready) + M(main.PartB->M3read) + M(main.StartEnable)
2*M(main.PartB_ready) + 2*M(main.R2_PartB->M2) + 2*M(main.M2_done_PartB) + 2*M(main.PartB->M3read) +
M(main.StartEnable) + M(main.StartLineB)
M(main.StartLineA) + M(main.StartLineB)
2*M(main.PartA_ready) + 2*M(main.R1_PartA->M1) + 2*M(main.M1_done_PartA) + 2*M(main.PartA->M3ready) +
M(main.StartEnable) + M(main.StartLineA)
M(main.T1_availability) + M(main.PartB_ready) + M(main.R2_PartB->M2) + M(main.M2_done_PartB) + M(main.PartB-
>M3read) + M(main.R1_availability) + M(main.StartLineB)
M(main.PartA_ready) + M(main.T2_availability) + M(main.R1_PartA->M1) + M(main.M1_done_PartA) + M(main.PartA-
>M3ready) + M(main.R2_availability) + M(main.StartLineA)
```

All places are covered by P-invariants.

Funkcionalna analiza

Rezultati funkcionalne analize GSPN proizvodnog fleksibilnog procesa



T invariants:

```
1 (main.t11)
1 (main.t2)
1 (main.t1)
1 (main.t3)
1 (main.t4)
1 (main.t5)
1 (main.t6)
1 (main.t7)
1 (main.t8)
1 (main.t9)
1 (main.t10)
```

ie.

```
main.t11 ; main.t2 ; main.t1 ; main.t3 ; main.t4 ;
main.t5 ; main.t6 ; main.t7 ; main.t8 ; main.t9 ;
main.t10
```

All transitions are covered by T-invariants.

14.6. Analiza performansi

- ❑ Analiza performansi konkurentnih sistema ima za cilj da procjeni (izmjeri) kako sistem dobro radi.
- ❑ Dvije osnovne tehnike:
 - **Ugrađeni Markovljevi lanci (analitička metoda).**
 - **Simulacija.**
- ❑ Tehnika ugrađenih Markovljevih lanaca ima dva ograničenja:
 - ne mogu se koristiti za analizu neograničenih Petrijevih mreža,
 - ne mogu se koristiti za analizu mreža sa veoma velikim prostorima stanja.
- ❑ Simulacija nema navedenih restrikcija i može omogućiti ispitivanje performansi kada analitičke metode zakažu, ali se zahtijeva veliki broj simulacijskih ciklusa.



Analiza performansi

Simulacijski ciklus

- ❑ Na početku svakog simulacijskog ciklusa, trenutno označavanje mreže se ispitiva kako bi se odredilo koji prijelazi su omogućeni.
- ❑ Ako nema omogućenih prijelaza, tada je sistem u zastoju i simulacija se završava.
- ❑ Inače, jedan omogućeni prijelaz mora biti označen za propaljivanje (aktiviranje); ako je trenutno označavanje nakon k prijelaza jednako μ_k , tada se omogućeni prijelaz t_j aktivira sa vjerojatnošću:

$$\frac{W_j(\mu_k)}{\sum_{i \in EN(\mu_k)} W_i(\mu_k)}$$

Analiza performansi

Mjere performansi



□ Pr (pojedinačan uvjet C vrijedi):

□ $PROB(C) = \sum \pi_i$, S_1 = skup označavanja za koji uvjet C vrijedi.

□ Pr (mjesto p_i ima tačno k oznaka):

□ $PROB(p_i, k) = \sum \pi_i$, S_2 = skup označavanja za koje mjesto p_i ima k oznaka - korisno za računanje **duljine reda čekanja**.

□ **E(broj oznaka u mjestu p_i)** = može se izračunati iz 2.

□ Brzina propusnosti neposrednih prijelaza

□ Iz eksponencijalnih prijelaza i strukture GSPN-a.

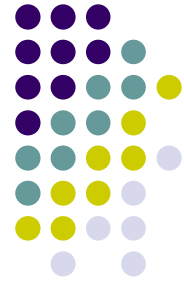
Analiza performansi

Mjere performansi

- Brzina propusnosti eksponencijalnog prijelaza t_j :

$$TR(t_j) = \sum \pi_i F(\mu_i, t_j) q_{ij}, \quad i \in S_3$$

- S_3 = skup označavanja za koji je prijelaz t_j omogućen.
 - $q_{ij} = 1$ ako t_j nije u sukobu sa bilo kojim drugim omogućenim prijelazima.
 - $q_{ij} =$ vjerojatnost da se t_j aktivira između omogućenih sukobljenih prijelaza u označavanju μ_i .
- Šta su sukobom omogućeni eksponencijalni prijelazi?
- Aktiviranje omogućenog prijelaza blokira (onemogućuju) druge omogućene prijelaze.
-- poništavanje procesiranja zbog zastoja.



Analiza performansi

Mjere performansi

- Srednje vrijeme čekanja u mjestu p_i :

$$WAIT(p_i) = \frac{EF(p_i)}{\sum_{t_j \in IT(p_i)} TR(t_j)} = \frac{EF(p_i)}{\sum_{t_j \in OT(p_i)} TR(t_j)}$$

IT: ulazni prijelazi, OT: izlazni prijelazi.



Analiza performansi

Rezultati analize performansi predložene GSPN



57/58

Run time :

2796.333333

Transition Throughput :

main.t11 0.162355 +/- 0.01424

main.t2 0.162713 +/- 0.014255

main.t1 0.162713 +/- 0.014255

main.t3 0.162713 +/- 0.014255

main.t4 0.162713 +/- 0.014255

main.t5 0.162355 +/- 0.01424

main.t6 0.162713 +/- 0.014255

main.t7 0.162355 +/- 0.01424

main.t8 0.162713 +/- 0.014255

main.t9 0.162355 +/- 0.01424

main.t10 0.162355 +/- 0.01424

Analiza performansi:

- Prosječan broj oznaka po mjestu.**
- Prosječna propusnost prijelaza.**

Analiza performansi

Rezultati analize performansi predložene GSPN



58/58

Mean Markings :

main.T1_availability 0.83538 +/- 0.009759

main.PartA_ready 0.16462 +/- 0.005345

main.PartB_ready 0.156574 +/- 0.005189

main.T2_availability 0.843426 +/- 0.009708

main.R1_PartA->M1 0.162177 +/- 0.005973

main.R2_PartB->M2 0.165812 +/- 0.005907

main.M1_done_PartA 0.163726 +/- 0.006571

main.M2_done_PartB 0.166587 +/- 0.006552

main.M1_availability 0.836274 +/- 0.008963

main.p15 0.833413 +/- 0.009028

main.PartA->M3ready 0.34724 +/- 0.010722

main.PartB->M3read 0.355108 +/- 0.010838

main.R2_availability 0.312493 +/- 0.005975

main.R1_availability 0.326857 +/- 0.006068

main.M3_done 0.164263 +/- 0.004335

main.FinalPart_Out 0.163786 +/- 0.004887

main.M3_availability 0.671951 +/- 0.011573

main.StartEnable 0.318155 +/- 0.007313

main.StartLineA 1.006318 +/- 0.009301

main.StartLineB 0.993682 +/- 0.009399