Poglavje 3

Petrijeve mreže

Petrijeve mreže predstavljajo univerzalno orodje za modeliranje in simulacijo dinamičnih sistemov. Na osnovi postavljenega modela in izvedene simulacije lahko pridemo do analize *dinamike* v sistemu, ki nam pove, ali je slednja v sistemu ustrezna ali ne. Pod pojmom dinamike v sistemu smatramo časovno sekvenco *stanj sistema*. Dinamičen sistem torej skozi čas spreminja svoja stanja. Analiza dinamike nam pomaga pri odločitvah ali je sistem potrebno popraviti, dopolniti, itd.

Temelje Petrijevih mrež je postavil Carl Adam Petri leta 1962. Definicijo njegovih mrež bomo v tem delu imenovali za *osnovne Petrijeve mreže* [5]. Poleg osnovnih Petrijevih mrež poznamo tudi *razširjene*. Mednje npr. sodijo *barvne*, *časovne*, *stohastične* in *mehke* Petrijeve mreže. Na področju računalništva se uporabljajo na področjih modeliranja komunikacijskih protokolov, analize zmogljivosti in zanesljivosti računalniških sistemov, analize verifikacije pravilnega delovanja algoritmov itd.

3.1 Gradniki Petrijevih mrež

Struktura Petrijeve mreže je sestavljena iz sledečih štirih tipov gradnikov:

- *pogojev* (angl. *places*),
- *akcij* (angl. *transitions*),
- *usmerjenih povezav* med akcijami in pogoji (ter obratno) in
- *žetonov*, s katerimi so definirane kratnosti izpolnjenosti posameznih pogojev.

Dinamiko v kakršnemkoli sistemu lahko interpretiramo kot zaporedje izvedenih akcij, za izvedbo katerih morajo biti izpolnjeni neki *predhodno* določeni pogoji. Proženje posamezne akcije je torej pogojeno z vnaprej določeno množico izpolnjenih pogojev, rezultat proženja akcije pa povzroči izpolnjenost novega ali

več novih pogojev. Slednje relacije so ponazorjene z usmerjenimi povezavami, ki vodijo iz pogojev v akcije in s povezavami, ki vodijo iz akcij v pogoje. Posamezen pogoj v Petrijevi mreži je lahko izpolnjen ali pa ne. V primeru izpolnjenosti pogoja govorimo o kratnosti izoplnjenosti pogoja. Tako je lahko pogoj izpolnjen enkrat, dvakrat ali večkrat, lahko pa ni izpolnjen. Žetone v Petrijevi mreži uporabljamo za označevanje kratnosti izpolnjenosti posameznih pogojev. En žeton v pogoju tako predstavlja enkratno izpolnjen pogoj, dva žetona dvakratno izpolnjen pogoj itd.

3.2 Osnovne Petrijeve mreže

Osnovne Petrijeve mreže v svoji definiciji ne predvidevajo časa trajanja akcij (čas trajanja akcij je hipen), niti ne predvidevajo stohastično pogojene dinamike. Informacijska vrednost žetona govori le o kratnosti izpolnjenosti pogoja, sam žeton pa ne nosi nobene druge dodatne informacijske vsebine. V nadaljevanju pričujočega razdelka bomo osnovnim Petrijevim mrežam rekli kar Petrijeve mreže.

3.2.1 Formalna definicija Petrijeve mreže

Zapišimo formalno definicijo Petrijeve mreže povzeto po viru [5].

Definicija 3 *Petrijeva mreža je definirana kot četvorček C=(P,T,I,O), pri čemer P predstavlja končno množico pogojev, T končno množico akcij, I vhodno in O izhodno funkcijo. Množici P in T sta si tuji ($P \cap T = \emptyset$).*

Vhodna in izhodna funkcija se nanašata na relacije med akcijami in pogoji. Vhodna funkcija I za akcijo t_j tako določa množico pogojev $I(t_j)$, iz katerih vodijo povezave proti akciji t_j, izhodna funkcija O pa za akcijo t_j določa množico pogojev $O(t_j)$, v katere vodijo povezave iz akcije t_j. Glede na povedano so možne povezave le iz akcij v pogoje in obratno, niso pa dovoljene povezave iz pogoja v pogoj ali iz akcije v akcijo. Običajno obe funkciji zapišemo v obliki prehajalnih matrik reda $m \times n$, pri čemer m predstavlja število akcij ($m = |T|$) in n število pogojev ($n = |P|$), ali v obliki *posplošenih* množic.

V izrazih (3.1) do (3.4) je predstavljen primer formalnega zapisa Petrijeve mreže s tremi akcijami, štirimi pogoji in desetimi povezavami najprej v matrični notaciji, nato pa še v notaciji posplošenih množic. Pri tem izraza (3.1) in (3.2) predstavljata *matrični način* formalne definicije, izrazi (3.1), (3.3) in (3.4) pa formalno definicijo na osnovi posplošenih množic.

$$C = (P, T, I, O), P = \{p_1, p_2, p_3, p_4\}, T = \{t_1, t_2, t_3\}, \tag{3.1}$$

$$I = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \; O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.2}$$

$$I(t_1) = \{p_1, p_2, p_3\}, I(t_2) = \{p_4\}, I(t_3) = \{p_3\}, \tag{3.3}$$

$$O(t_1) = \{p_1\}, O(t_2) = \{p_2, p_2, p_3\}, O(t_3) = \{p_4\}. \tag{3.4}$$

V našem delu se bomo v nadaljevanju držali zgolj matrične notacije.

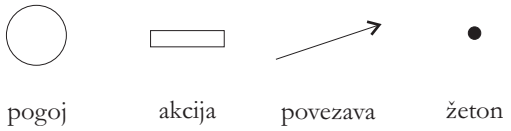
3.2.2 Definicija grafa Petrijeve mreže

Grafično ponazoritev formalno definirane Petrijeve mreže imenujemo *graf Petrijeve mreže*. Njegova definicija je po [5] sledeča:

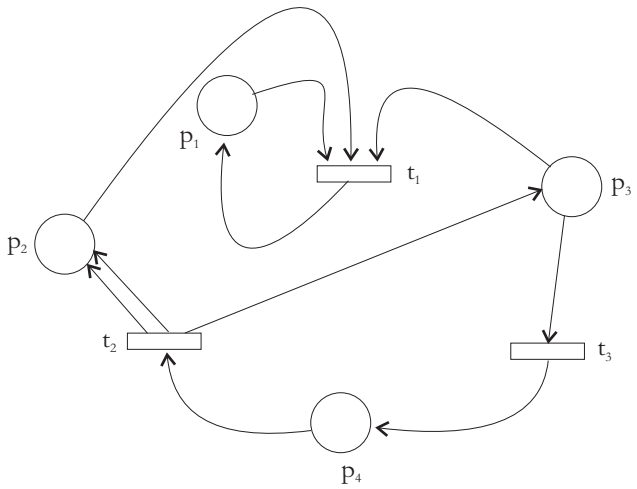
Definicija 4 *Graf Petrijeve mreže je bipartitni usmerjeni graf G=(V,A), kjer V ($V = P \cup T$) predstavlja množico vozlišč in A množico povezav med njimi ($a_i = (v_j, v_k)$). Velja naslednja relacija:*

$$\forall i : a_i = (v_j, v_k),\ (v_j \in T)\&(v_k \in P) \vee (v_j \in P)\&(v_k \in T). \tag{3.5}$$

Za tvorbo grafov Petrijevih mrež uporabljamo grafične primitive, ki so prikazani na sliki 3.1, graf Petrijeve mreže, ki smo jo formalno zapisali v izrazih (3.1) in (3.2) pa na sliki 3.2.



Slika 3.1: Grafični primitivi za tvorbo grafa Petrijeve mreže.



Slika 3.2: Primer grafa Petrijeve mreže.

3.2.3 Označitve pogojev v Petrijevih mrežah

Kot smo povedali že v uvodu z označitvami pogojev v Petrijevih mrežah *določamo* ali *odčitavamo* kratnost izpolnjenosti pogojev. Formalno *označitev* Petrijeve mreže zapišemo kot vektor

$$o(k) = (o_1(k), o_2(k), ..., o_n(k)), \forall i : (o_i(k) \geq 0) \& (o_i(k) \in \mathbb{N} \cup \{0\}), i = 1, .., n, \tag{3.6}$$

pri čemer n predstavlja število pogojev ($n = |P|$), $o_i(k)$ pa število žetonov v i-tem pogoju ali kratnost izpolnjenosti i-tega pogoja. Skozi čas se v sistemu sprožajo različne akcije in načeloma se s tem spreminjajo tudi izpolnjenosti posameznih pogojev. Iz tega razloga je elementom vektorja kot tudi vektorju označitve dodan diskretni časovni atribut k. Oglejmo si še formalno definicijo označitve.

Definicija 5 *Označitev Petrijeve mreže C=(P,T,I,O) je funkcija, ki množico stanj P preslika v vektor nenegativnih celih števil.*

$$o : P \to \mathbb{N} \cup \{0\}. \tag{3.7}$$

Z označitvijo vseh pogojev pridemo do *označene Petrijeve mreže*, ki jo zapišemo kot petorček $M = (P, T, I, O, o(k))$. Glede na to, da definicija označitve (3.6) ne predvideva omejitve števila žetonov v posameznem pogoju, za vsako Petrijevo mrežo obstaja neskončno možno označitev, če ima mreža vsaj en pogoj.

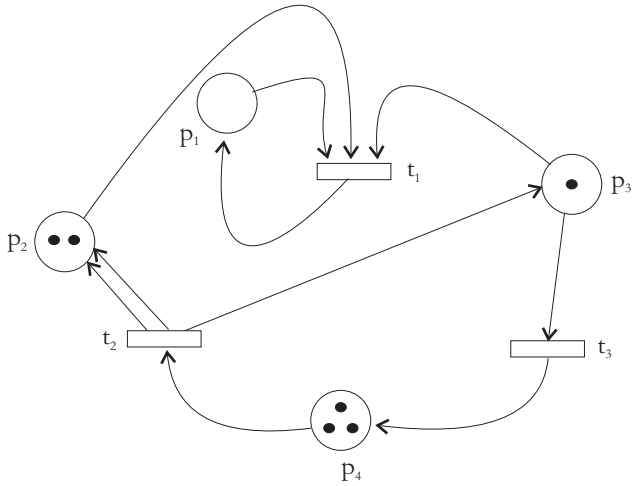
Za zgled mreže formalno definirane v izrazih od (3.1) do (3.4) predpostavimo začetno označitev $o(k_0) = (0, 2, 1, 3)$. V tem primeru bi prišli do grafične ponazoritve mreže, ki jo prikazuje slika 3.3. V primeru, da bi bilo žetonov v posameznem pogoju preveliko, bi v takšen pogoj zapisali število, ki bi predstavljalo kratnost izpolnjenosti pogoja. Označitev mreže $o(k)$ lahko enačimo s *stanjem*, v katerem se nahaja mreža na k-tem diskretnem koraku.

3.2.4 Proženje akcij v Petrijevih mrežah

Proženje posameznih akcij je pogojeno s trenutno označitvijo v Petrijevi mreži. Posamezna akcija se lahko sproži, če je *omogočena*. Akcija t_i je omogočena, če se po vsaki povezavi, ki vstopa v to akcijo, lahko pripelje žeton iz pogoja, ki predstavlja izvor povezave (vhodni pogoj). To lahko formalno zapišemo z izrazom

$$o(k) \geq e(t_i) * I, \ i = 1, ..., m, \tag{3.8}$$

pri čemer je $e(t_i)$ enotski vektor dolžine m ($m = |T|$), enica pa se nahaja na i-ti poziciji, ki jo definira indeks akcije. Sprožitev akcije torej v splošnem odvzema žetone iz vhodnih pogojev - pogojev iz katerih vodijo povezave proti opazovani akciji t_i. Po hipnem trajanju akcije (predpostavljamo, da je čas trajanja ničen ali zanemarljivo majhen), pride do posledičnega izražanja na izhodnih pogojih - pogojih do katerih vodijo izhodne povezave iz opazovane akcije t_i. Praviloma se po hipnem proženju akcije t_i odpravi po vsaki izhodni povezavi proti izhodnim



Slika 3.3: Primer označenega grafa Petrijeve mreže.

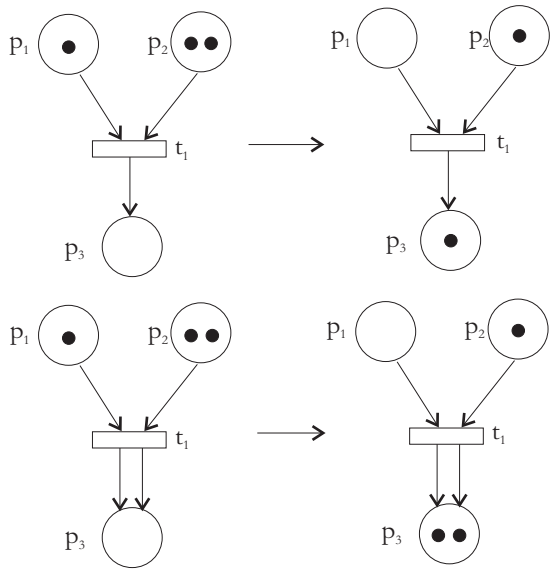
pogojem en žeton. Vsak od njih se uskladišči v izhodnem pogoju. Prehajanje žetonov od opazovane - prožene akcije t_i proti izhodnim pogojem formalno in s tem posledično spremenjeno označitev sistema $o(k+1)$ zapišemo z izrazom

$$o(k+1) = o(k) + e\,[t_i]\,(O - I). \tag{3.9}$$

Posebno je potrebno poudariti, da je potovanje posameznega žetona od vhodnega pogoja preko prožene akcije do izhodnega pogoja hipno (brez časovnega trajanja). V nadaljevanju si oglejmo nekaj konkretnih zgledov dinamik, ki so predstavljene na sliki 3.4. Iz slike je razvidno, da ni nujno, da je število žetonov, ki vstopijo v akcijo, enako številu žetonov, ki iz nje izstopijo.

Za primer povzemimo mrežo definirano formalno z matrikama I in O v izrazu (3.10) s slike 3.5 ob začetni označitvi $o(t) = (1, 0, 0, 0)$. Glede na začetno označitev so omogočene akcije t_1, t_2 in t_3. Ob predpostavki, da se istočasno ne moreta sprožiti dve akciji, imamo tri možne toke dogajanj, ki jih bomo ponazorili z *drevesom označitev*, predstavljenim na sliki 3.6. Do drevesa označitev lahko pridemo z izračunavanjem po izrazih (3.8) in (3.9), ali pa s postopnim prestavljanjem žetonov po grafu Petrijeve mreže. Izračun za levo vejo prehajalnega drevesa je predstavljen v izrazih od (3.12) do (3.15).

$$O = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \; I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$



Slika 3.4: Zgledi dinamike žetonov v Petrijevih mrežah.

$$O - I = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -2 & 1 \\ 1 & 0 & 0 & -1 \end{bmatrix} \tag{3.11}$$

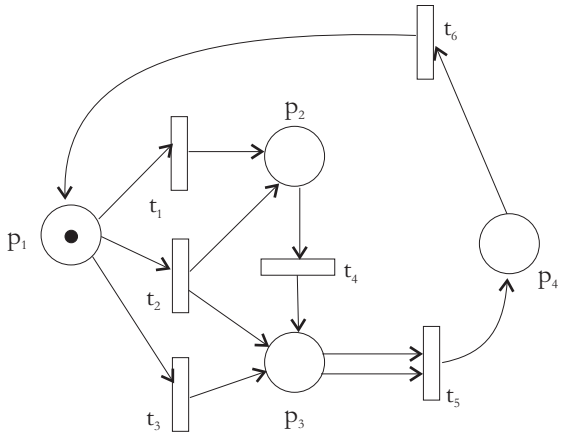
$$t_1 : o(t+1) = (1,0,0,0) + (1,0,0,0,0,0) * \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -2 & 1 \\ 1 & 0 & 0 & -1 \end{bmatrix} = \tag{3.12}$$

$$= (1,0,0,0) + (-1,1,0,0) = (0,1,0,0), \tag{3.13}$$

$$t_4 : o(t+2) = (0,1,0,0) + (0,0,0,1,0,0) * \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -2 & 1 \\ 1 & 0 & 0 & -1 \end{bmatrix} = \tag{3.14}$$

$$= (0,1,0,0) + (0,-1,1,0) = (0,0,1,0) \tag{3.15}$$

Ob analizi proženja akcij moramo biti pozorni na akcije, ki nimajo vhodnih pogojev. Tovrstne akcije so vedno omogočene. Tipičen primer takšne akcije je



Slika 3.5: Primer grafa Petrijeve mreže za izdelavo drevesa označitev.

predstavljen na sliki 3.7. Akcije, ki se lahko ves čas prožijo, tvorijo v drevesu označitev *neskončne veje*. Poleg neskončnih vej moramo biti v drevesu pozorni na *končna stanja* (liste drevesa), kjer se proženje akcij ustavi in na stanja, ki se skozi dinamiko ponavljajo (*ciklična stanja*).

S proženjem akcij v Petrijevi mreži se porajata dve vrsti sekvenc. Prva je sekvenca označitev, druga pa sekvenca akcij. Stanje Petrijeve mreže se odslikuje s trenutno označitvijo.

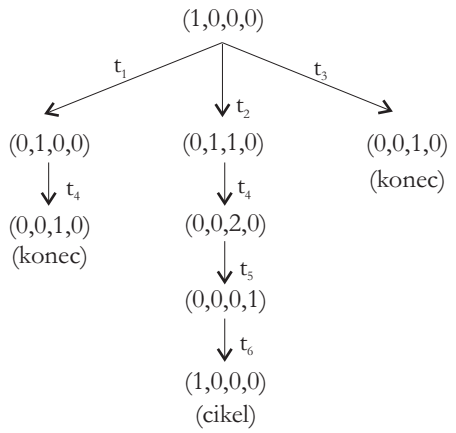
Označitev o' je *neposredno dosegljiva* iz označitve o, ko obstaja takšna akcija t_j, ki nas z izvajanjem pripelje iz označitve o v označitev o'. Povedano drugače je o' neposredno dosegljiva takrat, ko velja $\delta(o, t_j) = o'$, pri čemer δ predstavlja zapis preslikovalne funkcije v novo označitev (stanje).

Množica *dosegljivih stanj* $R(C, o)$ je množica vseh označitev (stanj Petrijeve mreže), ki so dosegljiva iz označitve o. Označitev o' pripada množici $R(C, o)$, če obstaja zaporedje akcij, ki nas pripelje iz označitve o v označitev o'. Velja naslednja relacija:

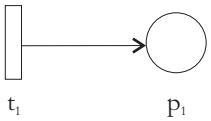
$$if\ (o' \in R(C,o))\ and\ (\exists t_j \in T : o'' = \delta(o', t_j))\ then\ o'' \in R(C,o). \qquad (3.16)$$

3.3 Zgledi modeliranja s Petrijevimi mrežami

V pričujočem razdelku si bomo ogledali nekaj konkretnih primerov modeliranja različnih sistemov s Petrijevimi mrežami. Večina primerov je povzeta po viru [5].



Slika 3.6: Drevo označitev za primer Petrijeve mreže s slike 3.5.



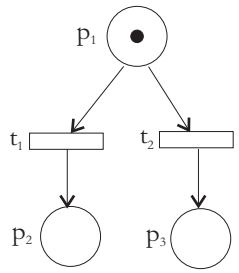
Slika 3.7: Primer akcije brez vhodnih pogojev, ki se neprestano proži.

3.3.1 Nedeterminizem in konkurenčnost v Petrijevih mrežah

Na sliki 3.8 je predstavljen graf Petrijeve mreže, ki ponazarja *nedeterminističnost* izbire poti žetona in *konkurenčnost* akcij. V kontekstu prvega pojma se žeton nedeterministično odloča o izbiri akcije, v kontekstu drugega pojma pa smatramo konkurenco ali kompeticijo med akcijama t_1 in t_2, ki skušata za svojo izvedbo pridobiti žeton.

3.3.2 Problem dveh kitajskih meditatorjev

Dva meditatorja sedita za okroglo mizo ter se izmenično prehranjujeta in meditirata. Za prehranjevanje potrebuje posamezni meditator dve paličici. Na mizi sta na začetku dve paličici za prehranjevanje s kitajsko hrano. Ena je na levi in ena na desni med sedočima, tako da imata oba občutek, da imata dostop do obeh paličic, ki jih nujno potrebujeta za prehranjevanje. Seveda se lahko glede na število razpoložljivih paličic prehranjuje le eden. Na sliki 3.9 je predstavljen graf Petrijeve mreže, ki ponazarja omenjeno situacijo. Predstavljena rešitev podpira nedeterministično izbiro meditatorja pri dodelitvi dveh paličic naenkrat. Navedimo pomene posameznih pogojev in akcij:



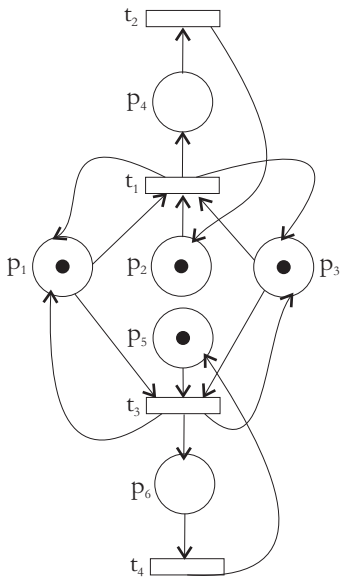
Slika 3.8: Graf Petrijeve mreže nedeterministične izbire poti žetona in konkurenčnosti dveh akcij.

- p_1: prva paličica je prosta,
- p_2: prvi meditator je pripravljen za prehranjevanje,
- p_3: druga paličica je prosta,
- p_4: prvi meditator je pripravljen za meditacijo,
- p_5: drugi meditator je pripravljen za prehranjevanje,
- p_6: drugi meditator je pripravljen za meditacijo,
- t_1: prvi meditator se prehranjuje,
- t_2: prvi meditator meditira,
- t_3: drugi meditator se prehranjuje,
- t_4: drugi meditator meditira.

3.3.3 Računalniški strežni sistem

Računalniški strežni sistem sestavljajo računalniki R1, R2 in R3. Vsaka zahteva, ki vstopi v sistem, mora biti najprej obdelana na R1, nato pa na R2 ali R3. Na sliki 3.10 je predstavljen graf Petrijeve mreže, ki ponazarja omenjeno situacijo. Rešitev podpira nedeterministično izbiro vira (resursa) R2 ali R3. Povezave narisane s prekinjenimi črtami označujejo vhodno in izhodno točko strežnega sistema. Navedimo pomene posameznih pogojev in akcij:

- p_1: v sistem je vstopila zahteva,
- p_2: računalnik R1 je prost,
- p_3: zahteva čaka na obdelavo v drugi fazi,
- p_4: zahteva je dokončno sprocesirana,



Slika 3.9: Graf Petrijeve mreže za problem dveh kitajskih meditatorjev.

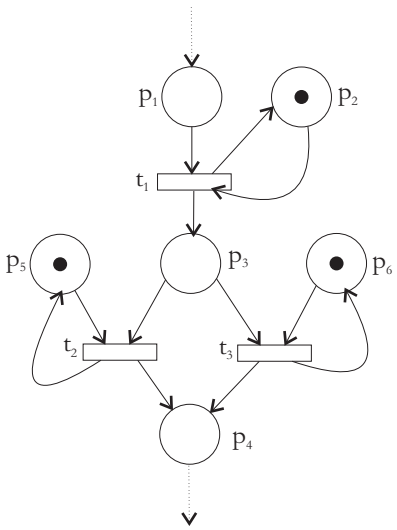
- p_5: računalnik R2 je prost,
- p_6: računalnik R3 je prost,
- t_1: zahteva se obdeluje na R1,
- t_2: zahteva se obdeluje na R2,
- t_3: zahteva se obdeluje na R3.

3.3.4 DO WHILE programski stavek s čuvajem

DO WHILE programski stavek definira izvajanje programskega stavka S, pri čemer se ta prvič izvede brezpogojno, vse nadaljnje izvedbe pa so pogojene z veljavnostjo pogoja P. Formalno ga lahko zapišemo z izrazom

$$do\ \{S\}\ while(P). \tag{3.17}$$

Na sliki 3.11 je predstavljen graf Petrijeve mreže, ki ponazarja opisani sistem. Povezava zaključena s krožcem (p_5-t_3) predstavlja njeno *inhibirno* naravo. Slednje pomeni, da se izhodna akcija t_3 lahko sproži le pod pogojem, da v izvornem pogoju inhibirne povezave ni žetonov. Na tem mestu omenimo, da inhibirna povezava predstavlja razširjavo klasičnih Petrijevih mrež. Nenazadnje nam ob njeni vpeljavi ne služita več izraza (3.8) in (3.9).

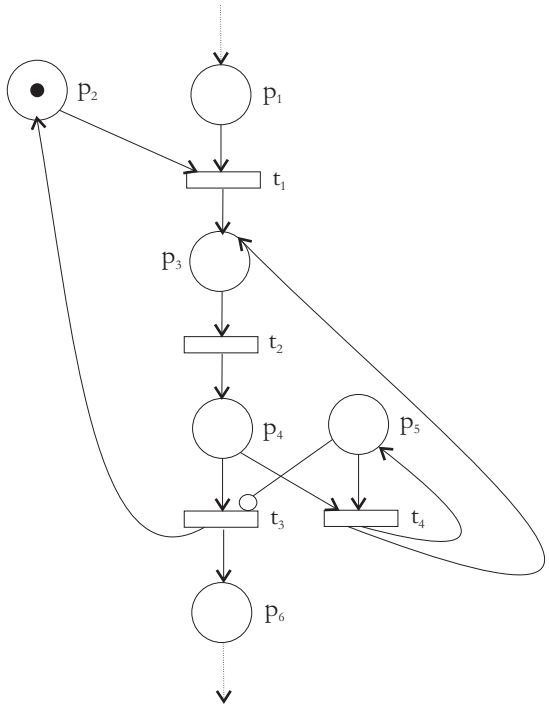


Slika 3.10: Graf Petrijeve mreže za ponazoritev računalniškega strežnega sistema.

Pogoj p_2 predstavlja funkcijo *čuvaja*, ki onemogoča vstope v DO WHILE sekvenco, če je le ta že v uporabi. Navedimo pomene posameznih pogojev in akcij:

- p_1: v sistem je vstopila zahteva za izvajanje prog.stavka,
- p_2: predstavlja funkcijo čuvaja, ob prisotnosti katerega je možno izvršiti zanko,
- p_3: zahteva čaka na izvedbo S stavka,
- p_4: izvedba S stavka je opravljena,
- p_5: pogoj P,
- t_1: izvede se vstop v sekvenco (ob prisotnosti čuvaja),
- t_2: izvede se S stavek,
- t_3: ker pogoj P ni izpolnjen, se izvajanje zaključi in čuvaj se vrne v razpoložljivo stanje,
- t_4: ker pogoj P je izpolnjen, se vrnemo na ponovno brezpogojno izvajanje stavka S.

Kot smo povedali, p_5 predstavlja preverjani pogoj P. Akcija t_4 se izvaja vse dotlej, dokler je v p_5 kakšen žeton (dokler je pogoj izpolnjen). Ker s tem t_4 odvzema žetone iz p_5 in s tem „ruši“ pogoj, vpeljemo povezavo $t_4 - p_5$, ki žetone p_5 vrača. Tako samo preverjanje pogoja P slednjega ne spreminja.



Slika 3.11: Graf Petrijeve mreže za ponazoritev DO WHILE stavka.

3.3.5 IF programski stavek s čuvajem

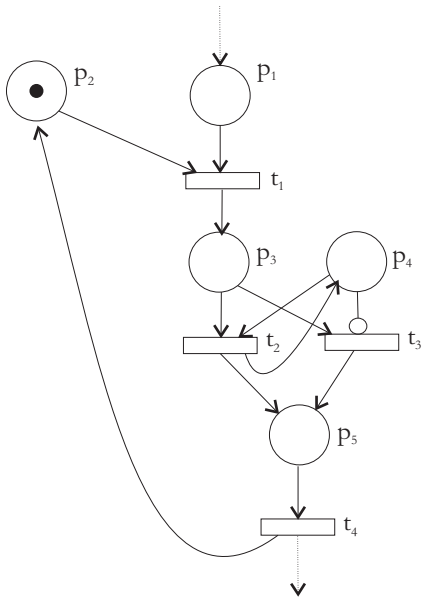
IF programski stavek definira izvajanje programskega stavka S_1, če je izpolnjen pogoj P, v nasprotnem primeru pa izvajanje stavka S_2. Formalno ga lahko zapišemo z izrazom

$$if\ (P)\ then\ \{S_1\}\ else\ \{S_2\}\,. \tag{3.18}$$

Na sliki 3.12 je prikazan graf Petrijeve mreže, ki ponazarja omenjeni sistem. Pogoj p_2 zopet predstavlja funkcijo čuvaja, povezava p_4 - t_3 pa nestandardno inhibirno povezavo. Navedimo še ostale pomene posameznih pogojev in akcij:

- p_1: v sistem je vstopila zahteva za izvajanje IF stavka,
- p_2: predstavlja funkcijo čuvaja,
- p_3: zahteva je pred odločitvijo, kateri stavek izvesti,
- p_4: predstavlja pogoj P,
- p_5: eden od stavkov je izveden,
- t_1: izveden je vstop v stavek,

- t_2: izvede se S_1,
- t_3: izvede se S_2,
- t_4: stavek je dokončno izveden.



Slika 3.12: Graf Petrijeve mreže za ponazoritev IF stavka.

3.3.6 FOR programski stavek s čuvajem

FOR programski stavek predvideva izvajanje programskega stavka S v n ponovitvah. Formalno ga lahko zapišemo z izrazom

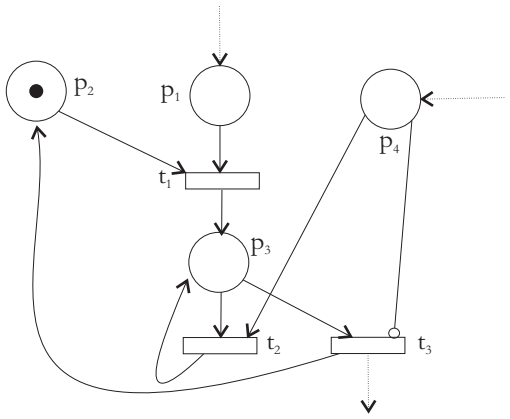
$$for\ (i = 1\ to\ n)\ do\ \{S\}\,. \tag{3.19}$$

Na sliki 3.13 je prikazan graf Petrijeve mreže, ki ponazarja omenjeni sistem. Navedimo pomene posameznih pogojev in akcij:

- p_1: v sistem je vstopila zahteva za izvajanje FOR stavka,
- p_2: predstavlja funkcijo čuvaja,
- p_3: zahteva je pred odločitvijo ali stavek S izvesti ali ne,
- p_4: predstavlja števec n, ki mora biti predhodno inicializiran z n žetoni,
- t_1: izveden je vstop v stavek,

- t_2: izvede se stavek S,
- t_3: FOR stavek je dokončno izveden v celoti.

Črtkana povezava, ki vstopa v p_4 brez izvora, nas opozarja na to, da moramo pred samo izvedbo FOR stavka inicializirati število žetonov (število n). Ključni točki stavka sta v akcijah t_2 in t_3. Prva se izvaja vse dotlej, dokler je v p_4 še kakšen žeton ($n>0$), ko pa le teh zmanjka ($n = 0$), se izvede akcija t_3.



Slika 3.13: Graf Petrijeve mreže za ponazoritev FOR stavka.

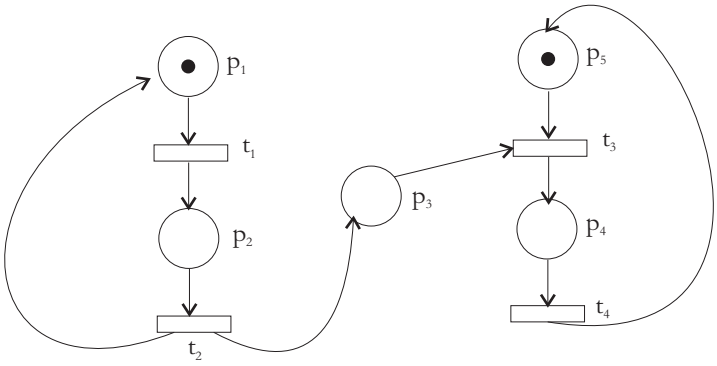
3.3.7 Proizvodno porabniški problem

Problem proizvajalca in porabnika (angl. *producer - consumer problem*) na eni strani definira proces proizvodnje, na drugi strani pa proces porabe. Na sliki 3.14 je predstavljen graf Petrijeve mreže, ki ponazarja omenjeno situacijo. Navedimo pomene posameznih pogojev in akcij:

- p_1: proizvodni proces je pripravljen za proizvodnjo ene enote,
- p_2: enota je proizvedena,
- p_3: funkcija vmesnika (angl. *buffer*), kjer proizvedene enote čakajo na porabo,
- p_4: enota je prevzeta iz vmesnika,
- p_5: proces porabe je pripravljen na porabo ene enote,
- t_1: proizvede se enota,
- t_2: proizvodni proces preda enoto v vmesnik in inicializira se pripravljenost na ponovno proizvodnjo (žeton preide v p_1),

- t_3: proces porabe prevzame enoto iz vmesnika,
- t_4: porabi se enota in inicializira ponovna pripravljenost na porabo (žeton preide v p_5).

Pomembno je, da začetna označitev inicializira pogoja p_1 in p_5.



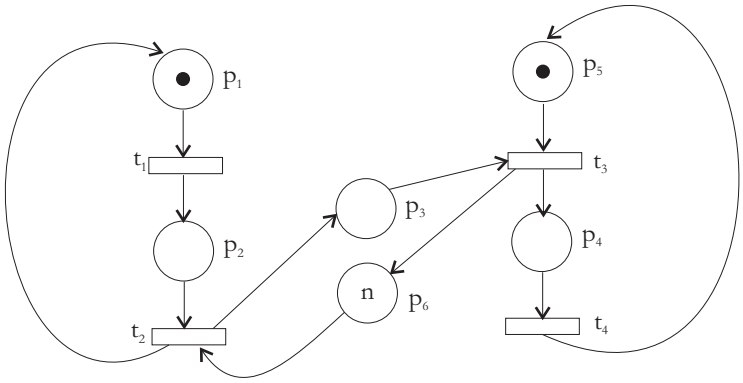
Slika 3.14: Graf Petrijeve mreže za ponazoritev proizvodno porabniškega sistema.

Omenjeni sistem ni najbolj realističen, ker lahko pride do prekoračitve kapacitete realiziranega vmesnika, ki praviloma nikdar nima neskončne kapacitete. V tem smislu na sliki 3.15 predstavimo dopolnitev predhodnega grafa Petrijeve mreže, ki omejuje prekomerno polnjenje vmesnika. V njem nastopa nov pogoj p_6, v katerem je na začetku n žetonov, pri čemer n predstavlja kapaciteto vmesnika. Ko se iz p_6 odstrani vse žetone, akcija t_2 ne more več odlagati enot v vmesnik in je potrebno za nadaljevanje procesa proizvodnje počakati fazo porabe, da p_6 napolni z vsaj enim žetonom (odvzame iz vmesnika vsaj eno enoto).

3.3.8 Problem branja in pisanja

Problem branja in pisanja (angl. *read - write problem*) je aktualen pri kakršnemkoli paralelnem dostopanju do podatkov. Pod pojmom dostopanja ločujemo med procesoma branja in pisanja. V primeru branja načeloma lahko omogočamo n paralelnih dostopov (pravic do branja), pri pisanju pa moramo predhodno podatke zakleniti in s tem onemogočiti vsa aktivna branja in morebitna ostala pisanja. Na sliki 3.16 je predstavljen graf Petrijeve mreže, ki ponazarja omenjeno situacijo. Navedimo pomene posameznih pogojev in akcij:

- p_1: pripravljena je zahteva za branje,
- p_2: omogočen je dostop do branja podatka,
- p_3: vmesnik, v katerem je shranjenih n dostopnih pravic,



Slika 3.15: Graf Petrijeve mreže za ponazoritev realnega proizvodno porabniškega sistema.

- p_4: omogočen je dostop do pisanja,
- p_5: pripravljena je zahteva za pisanje,
- t_1: izvede se zaseganje enkratne pravice za branje,
- t_2: izvede se branje in enkratna pravica do branja se vrne v p_3,
- t_3: izvede se zaseganje n kratne pravice za pisanje (zaseg n žetonov),
- t_4: izvede se pisanje in n kratna pravica za dostop se vrne v p_3.

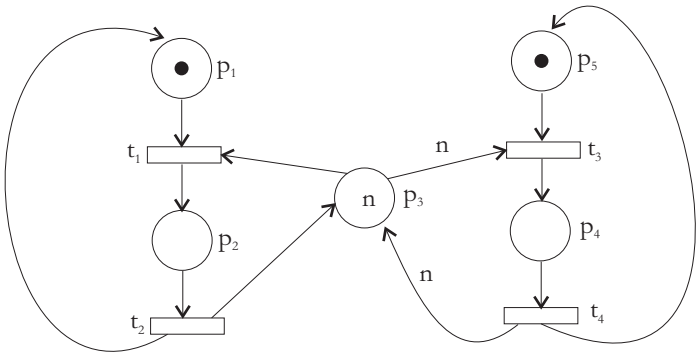
Z vidika začetne označitve je pomembna prisotnost vsaj enega žetona v levem (bralnem) ciklu, prisotnost vsaj enega žetona v desnem (pisalnem ciklu) in označitev števila pravic (p_3).

Posebno pozornost gre nameniti povezavama $p_3 - t_3$ in $t_4 - p_3$, ki sta n - kratni povezavi, kar pomeni, da se po takšni povezavi prelije n žetonov hkrati.

3.3.9 Problem medsebojnega izključevanja

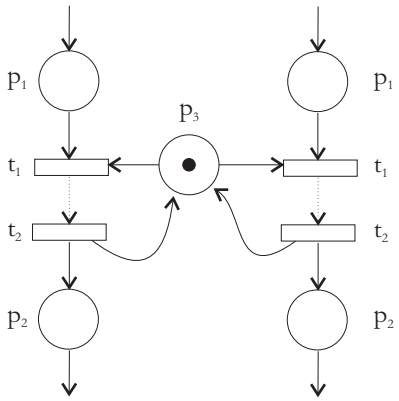
Problem medsebojnega izključevanja (angl. *mutual exclusion problem*) opozarja na možnost obstoja *kritičnih sekcij* sistema, ki ne smejo biti zasedene istočasno. Na sliki 3.17 je prikazan graf Petrijeve mreže, ki ponazarja model omenjenega problema s podeljevanjem pravice do dostopa do sekcije (stražarja). Z vidika začetne označitve je pomembna prisotnost žetona v p_3. Navedimo pomene posameznih pogojev in akcij:

- p_1: do kritične sekcije pristopi zahteva,
- p_2: zahteva zapusti kritično sekcijo,
- p_3: prisotnost enkratne pravice dostopa do kritične sekcije,



Slika 3.16: Graf Petrijeve mreže za ponazoritev problema branja in pisanja.

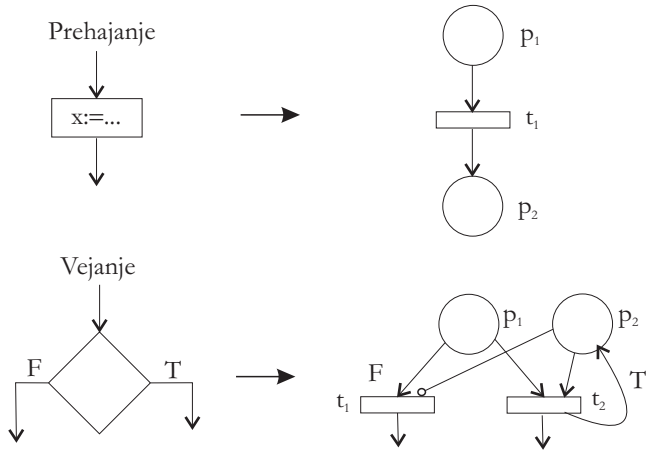
- t_1: dodelitev pravice za dostop do kritične sekcije in njen začetek,
- t_2: konec kritične sekcije in vračanje pristopne pravice v p_3.



Slika 3.17: Graf Petrijeve mreže za ponazoritev eliminacije možnosti istočasnega nahajanja zahtev v kritičnih sekcijah.

3.3.10 Modeliranje diagrama poteka

S Petrijevimi mrežami lahko modeliramo tudi diagrame poteka, ki so ena od osnovnih metod za enolično ponazarjanje algoritmov. Osnovna konstrukta diagramov poteka sta *prehajanje* in *vejanje*. Na sliki 3.18 najdemo ponazoritve, v kakšne Petrijeve konstrukte se preslikajo aktivnosti in v kakšne vejanja. Pri tem pogoj p_2 pri vejanju predstavlja preverjani pogoj, na osnovi katerega se vejanje vrši.



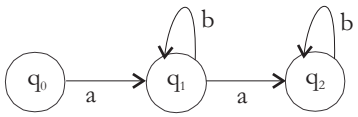
Slika 3.18: Prehod iz gradnikov diagramov poteka na osnovne konstrukte Petrijevih mrež.

vhodna črka - stanje	q_0	q_1	q_2
a	q_1 (t_{11})	q_2 (t_{12})	- (t_{13})
b	- (t_{21})	q_1 (t_{22})	q_2 (t_{23})

Tabela 3.1: Tabela prehajanj v avtomatu s slike 3.19.

3.3.11 Prehod med končnim avtomatom in Petrijevo mrežo

Predpostavimo, da imamo opravka s končnim avtomatom, predstavljenim na sliki 3.19. Omenjeni avtomat lahko ponazorimo s prehajalno tabelo 3.1.



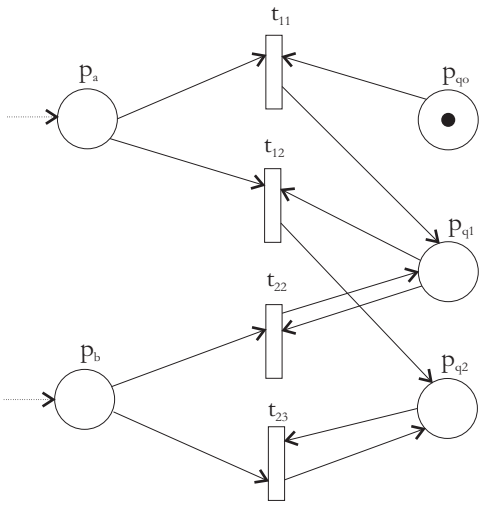
Slika 3.19: Primer končnega avtomata.

Pravila za formiranje ekvivalentne Petrijeve mreže lahko strnemo v naslednje alineje:

- črke vhodne abecede se preslikajo v pogoje (za naš primer tako velja $X = \{a, b\} \Rightarrow P_1 = \{p_a, p_b\}$),
- stanja avtomata se preslikajo v pogoje (za naš primer tako velja $Q = \{q_0, q_1, q_2\} \Rightarrow P_2 = \{p_{q_0}, p_{q_1}, p_{q_2}\}$),

- prehajanja med stanji se preslikajo v akcije (za na naš primer tako velja $T = \{t_{11}, t_{12}, t_{22}, t_{23}\}$),
- formira se množica pogojev P ($P = P_1 \cup P_2$) (za naš primer tako velja $P = \{q_0, q_1, q_2, p_a, p_b\}$).

Na sliki 3.20 je predstavljen graf Petrijeve mreže, ki je ekvivalentna avtomatu s slike 3.19. Z vidika izvajanja mreže je pomembno, da v enega od pogojev v mreži, ki ponazarjajo stanja (q_0, q_1, q_2), vstavimo en žeton, s čimer inicializiramo začetno stanje avtomata.



Slika 3.20: Pretvorba končnega avtomata s slike 3.19 v graf Petrijeve mreže.

3.3.12 Petrijeve mreže kot generatorji jezikov

V predhodnih razdelkih smo bili pozorni predvsem na pogoje, označitve in sekvence označitev. V pričujoče razdelku bomo pozornost preusmerili na sekvenco proženih akcij, ki okarakterizirajo modelirani sistem. V kontekstu jezikov Petrijevih mrež posamezna sprožena akcija predstavlja *znak*, sekvenca sproženih akcij *besedo*, vse možne sekvence proženih akcij pa *jezik* Petrijeve mreže. Ob tem smo že predhodno predpostavljali, da se dve akciji ne moreta sprožiti istočasno. Dve Petrijevi mreži sta *ekvivalentni*, če imata enake jezike. V nadaljevanju zapišimo formalno definicijo jezika.

Definicija 6 *Jezik L je jezik Petrijeve mreže, če obstajajo Petrijeva mreža C=(P,T,I,O), označitev akcij $\rho : T \to \Sigma$, začetna označitev $o(t_0)$ in končna množica končnih označitev F, tako da velja:*

$$L = \{\rho(\beta) \in \Sigma^* : \beta \in T^* \text{ and } \delta(o(t_0), \beta) \in F\}\,. \tag{3.20}$$

Pri tem Σ predstavlja množico vseh črk jezika, Σ^* množico vseh možnih tvorb besed glede na množico črk jezika, F množico vseh besed jezika, β poljubno zaporedje akcij, δ pa preslikovalno funkcijo, ki na osnovi začetne označitve $o(t_0)$ in zaporedja akcij β formira veljavno besedo jezika.

Oglejmo si primer jezika $L = L_1 \cup L_2, L_1 = a(a \vee b)b, L_2 = a^n b^n,\ n \geq 1$. Rešitev generatorja jezika v obliki grafa Petrijeve mreže podajamo na zaporedju slik 3.21 (L_1 in L_2) in 3.22 ($L_1 \cup L_2$). Začetna inicializacija ni potrebna na nobeni od obeh navedenih slik. Na tem mestu povdarimo, da se tako izbira med tipom besede, kot tudi izbira dolžine sekvence n vršita nedeterministično.

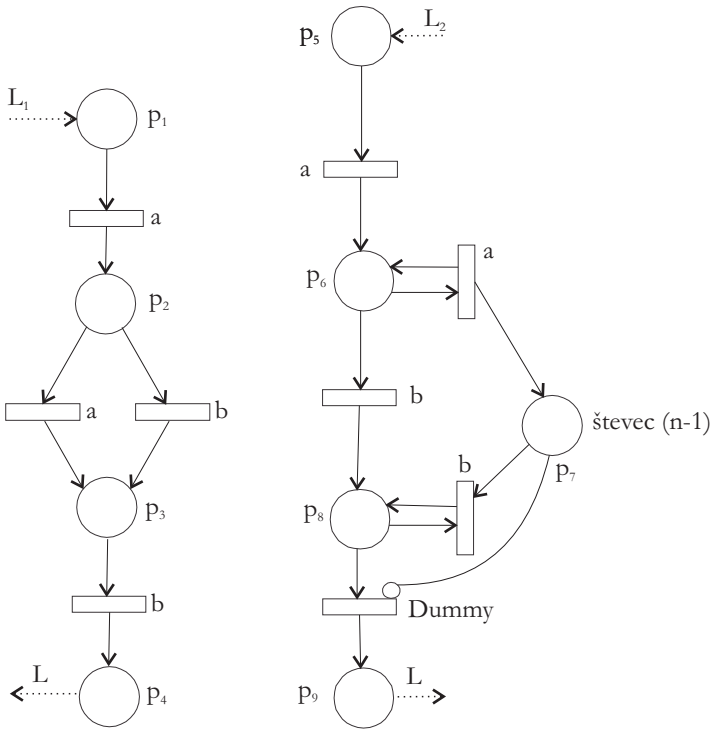
Opišimo še pomene pomembnejših pogojev in akcij:

- p_1: v sistem L_1 je vstopila zahteva za generiranje besede jezika L_1,
- p_2: nedeterministično odločanje med generiranjem črke „a“ ali „b“,
- p_4: beseda jezika L_1 je bila uspešno generirana,
- p_5: v sistem L_2 je vstopila zahteva za generiranje besede jezika L_2,
- p_6: izhodišče za nedeterministično generiranje $(n-1)$ pojavitev črke „a“,
- p_7: pomnjenje števila $(n-1)$ generiranj črke „a“,
- p_8: izhodišče za deterministično generiranje $(n-1)$ pojavitev črke „b“,
- p_9: beseda jezika L_2 je bila uspešno generirana,
- p_{11}: v sistem L je vstopila zahteva za generiranje besede jezika L in pride do nedetrminističnega izbiranja besede iz jezika L_1 ali L_2,
- p_{12}: beseda jezika L je bila uspešno generirana,
- Dummy: vse akcije s to oznako nimajo zmožnosti generiranja črke,
- a: akcije s to oznako generirajo črko „a“,
- b: akcije s to oznako generirajo črko „b“,.

Na sliko 3.22 bi bilo umestno dodati tudi čuvaja, ki bi zagotavljal, da ne bi v sistem (kritično sekcijo) vstopilo več zahtev za generiranje, kar bi vodilo do neustrezne sekvence akcij in posledično do odčitavanja nepravih besed jezika.

3.4 Analiza Petrijevih mrež

Rezultati modeliranja in simulacij s Petrijevimi mrežami nas vodita do možnosti analize dinamike v Petrijevih mrežah. Predhodno smo se že spoznali z metodo analize s pomočjo *drevesa dosegljivosti*, v nadaljevanju pa se bomo seznanili še z analitičnimi ocenami lastnosti *varnosti*, *omejenosti* in *konservativnosti*.



Slika 3.21: Primer Petrijevih mrež - generatorjev jezika L_1 (levo) in L_2 (desno).

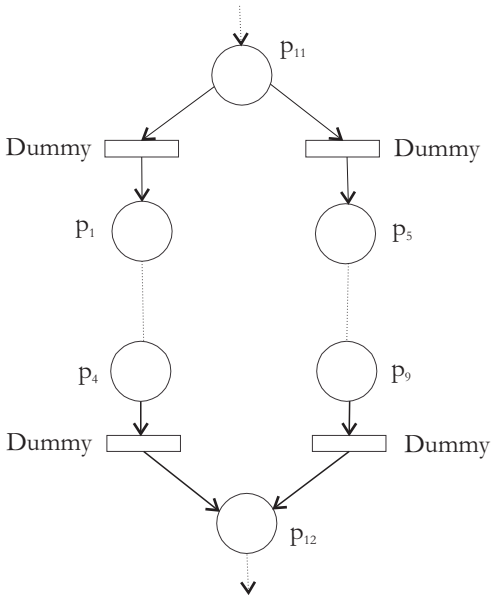
3.4.1 Varnost Petrijeve mreže

Ena od možnih značilnosti opazovane Petrijeve mreže je njena *varnost* (angl. *safeness*). Posamezen pogoj v mreži je *varen*, če se skozi simulacijo število žetonov v tem pogoju nikdar ne dvigne nad 1. Petrijeva mreža kot celota je *varna*, če so v njej varni tudi vsi pogoji. Zapišimo definicijo še formalno.

Definicija 7 *Pogoj $p_i \in P$ Petrijeve mreže C=(P,T,I,O) z začetno označitvijo o je varen, če za vse o'$\in R(C,o)$ velja o'$(p_i) \leq 1$. Petrijeva mreža C je varna, če so v njej varni vsi pogoji.*

Omenjena značilnost opazovane Petrijeve mreže je dobrodošla predvsem na področju modeliranja logičnih enačb, logičnih preklopnih struktur in podobnih sistemov, ki lahko zasedajo le dve možni stanji. Na ta način posamezni pogoj v Petrijevi mreži obravnavamo kot dvovrednostni logični pogoj.

V primeru, da pogoj p_i v Petrijevi mreži ni varen in nima tako večkratnih vhodnih, kot tudi večkratnih izhodnih povezav, obstaja način, da tudi ta pogoj spremenimo v varen pogoj. To dosežemo tako, da vpeljemo nov pogoj p_i'. Postopek vpeljave novega pogoja je zapisan v izrazih



Slika 3.22: Primer Petrijeve mreže - generatorja jezika $L_1 \cup L_2$.

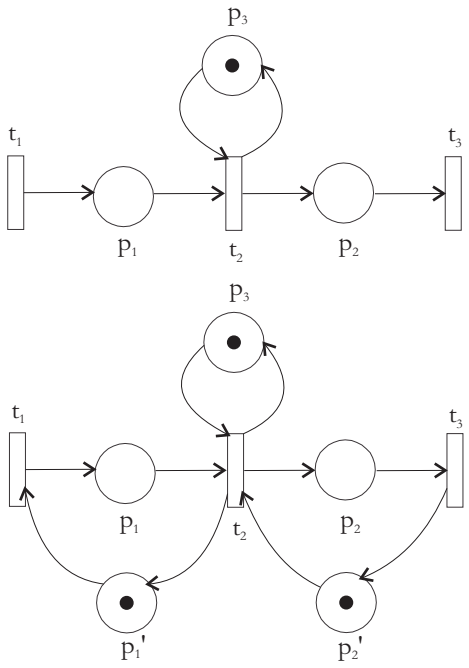
$$if\ (p_i \in I(t_j))\ and\ (p_i \notin O(t_j))\ then\ add\ p_i'\ to\ O(t_j), \tag{3.21}$$
$$if\ (p_i \in O(t_j))\ and\ (p_i \notin I(t_j))\ then\ add\ p_i'\ to\ I(t_j), \tag{3.22}$$

razložimo pa ga lahko na naslednji način. Če je pogoj p_i, ki ni varen, vhodni pogoj za akcijo t_j, ni pa njen izhodni pogoj, uvedemo nov izhodni pogoj p_i' akciji t_j. če pa je pogoj p_i, ki ni varen, izhodni pogoj za akcijo t_j, ni pa njen vhodni pogoj, uvedemo za akcijo t_j nov vhodni pogoj p_i'. Pogoj p_i' je komplementaren pogoju p_i, pomensko pa predstavlja izjavo, da je pogoj p_i neveljaven (v p_i ni žetonov). Po vpeljavi novega pogoja moramo poskrbeti tudi za spremembo začetne označitve, ki mora vsebovati žeton bodisi v p_i ali v p_i'. Na sliki 3.23 je prikazan primer pretvorbe pogojev p_1 in p_2 v varna pogoja.

Lastnost varnosti je zanimiva iz vidika pomnjenja, saj nam zniža število možnih stanj sistema in s tem tudi poceni realizacijo modela v obliki strojne realizacije.

3.4.2 Omejenost Petrijeve mreže

Omejenost je posplošena značilnost varnosti. Petrijeva mreža je k-omejena, če se število žetonov v posameznih pogojih skozi simulacijo nikdar ne dvigne čez mejo k, jo pa vsaj občasno in v vsaj nekaterih pogojih dosega.



Slika 3.23: Primer nevarne Petrijeve mreže (zgoraj) in njene pretvorbe v varno (spodaj).

Definicija 8 *Pogoj $p_i \in P$ Petrijeve mreže C=(P,T,I,O) z začetno označitvijo o je k-omejen, če za vse o'$\in R(C,o)$ velja o'$(p_i) \leq k$. Petrijeva mreža C je k-omejena, če je vrednost k maksimalna gledano po vseh pogojih.*

Značilnost omejenosti porajanja števila žetonov v posameznem pogoju je pogoj za možnost stabilne (končne) realizacije modeliranega sistema.

3.4.3 Konservativnost v Petrijevih mrežah

V mnogih primerih uporabe Petrijevih mrež za namene modeliranja nam pogoji predstavljajo vsebinske resurse, ki morajo biti razpoložljivi za izvedbo akcij. Takšnih resursov skozi dinamiko mreže ne moremo niti uničiti, niti povečevati njihovega števila. Zanje velja značilnost ohranjanja ali *konservacije* v smislu njihovega števila. Slednje lahko zapišemo tudi formalno.

Definicija 9 *Petrijeva mreža C=(P,T,I,O) z začetno označitvijo o je striktno konservativna, če za vse o'$\in R(C,o)$ velja relacija $\sum_{p_i \in P} o'(p_i) = \sum_{p_i \in P} o(p_i)$.*

Relacija v definiciji zahteva, da se vsota žetonov skozi dinamiko mreže ne spreminja. Striktna konservativnost v mreži zahteva, da je $\forall j, j = 1, ..., n, n =$

$|T|, |I(t_j)| = |O(t_j)|$. Povedano drugače mora za vsako akcijo veljati, da je število vhodnih povezav enako številu izhodnih povezav.

V večini modelov realnih sistemov striktna konservativnost ni potrebna. Nekateri pogoji npr. lahko predstavljajo števce, v katerih se število žetonov skozi čas spreminja. V takšnih primerih skušamo s primerno obravnavo eliminirati pogoje, v katerih imamo motečo dinamiko in ohraniti pod drobnogledom samo tiste pogoje, v katerih želimo skozi čas imeti vsotno gledano konstantno število žetonov. Zapišimo to formalno.

Definicija 10 *Petrijeva mreža C=(P,T,I,O) z začetno označitvijo o je konservativna glede na utežni vektor* $w=(w_1, w_2, ..., w_n), n = |P|, w_i \in \{0, 1\}$, *če za vse* $o' \in R(C, o)$ *velja relacija* $\sum_{p_i \in P} w_i \cdot o'(p_i) = \sum_{p_i \in P} w_i \cdot o(p_i)$.

Petrijeva mreža je tako lahko striktno konservativna le ob upoštevanju vektorja uteži $w = (1, 1, ..., 1)$, poljubna mreža pa venomer konservativna z vektorjem uteži $w = (0, 0, ..., 0)$. Zaradi slednjega pogoj za konservativnost zaostrimo in sicer zahtevamo, da je vektor w neničeln ($\exists w_i = 1$).

3.5 Zgledi modeliranja s področja računalniških omrežij

V pričujočem razdelku si bomo ogledali še nekaj specifičnejših zgledov modeliranja s Petrijevimi mrežami na področju računalniških omrežij.

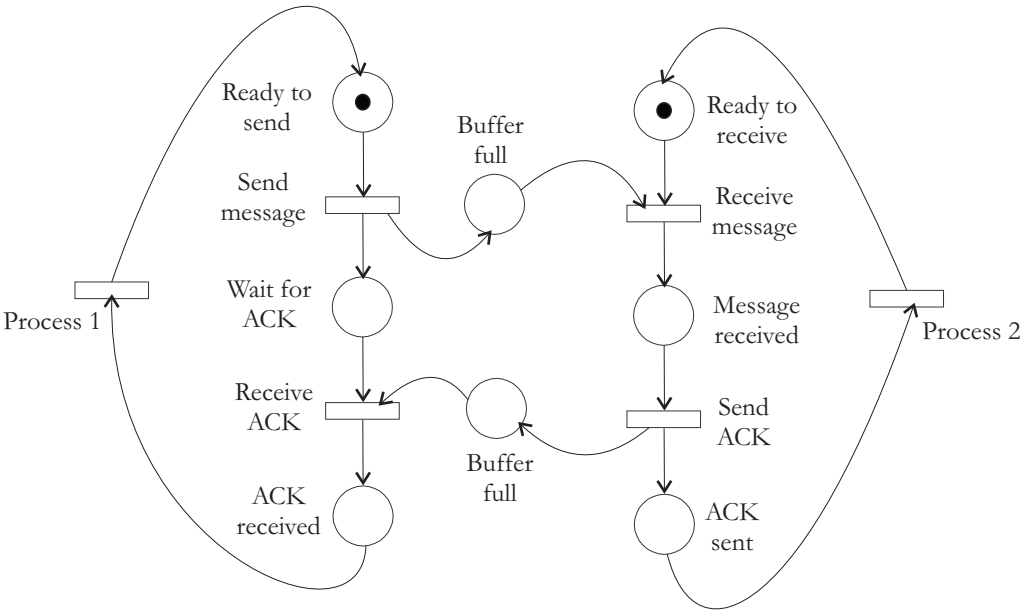
3.5.1 Poenostavljen model protokola med oddajnikom in sprejemnikom

Predpostavimo, da imamo opravka z enostavnim primerom protokola med oddajnikom in sprejemnikom. Prvi sporočila oddaja in počaka na potrditev vsake oddaje, drugi pa sporočila prejema in generira potrditve vseh prejemov. Model tovrstnega sistema je predstavljen na sliki 3.24, povzeti po [6].

V modelu bodimo pozorni na pogoja Bufferfull, saj je iz slike ob podani začetni označitvi razvidno, da sta varna. Enako velja tudi za vse ostale pogoje.

3.5.2 Model nedeterminističnega čakalnega procesa

Predpostavimo, da imamo opravka s sistemom pošiljanja sporočil, pri čemer se odpošiljanje izvede na osnovi nedeterministične izbire akcij t_{r1} (pošiljanje naslovniku 1), t_{r2} (pošiljanje naslovniku 2) ali t_{out} (sporočilo se pošlje v ponovno oddajo, ker predhodno poslanega sporočila ni potrdil nobeden od naslovnikov - simbolična predstavitev stanja TimeOut). Model tovrstnega sistema je predstavljen na sliki 3.25 povzeti po [6].



Slika 3.24: Poenostavljen model protokola med oddajnikom in sprejemnikom povzet po [6].

3.6 Razširitve Petrijevih mrež

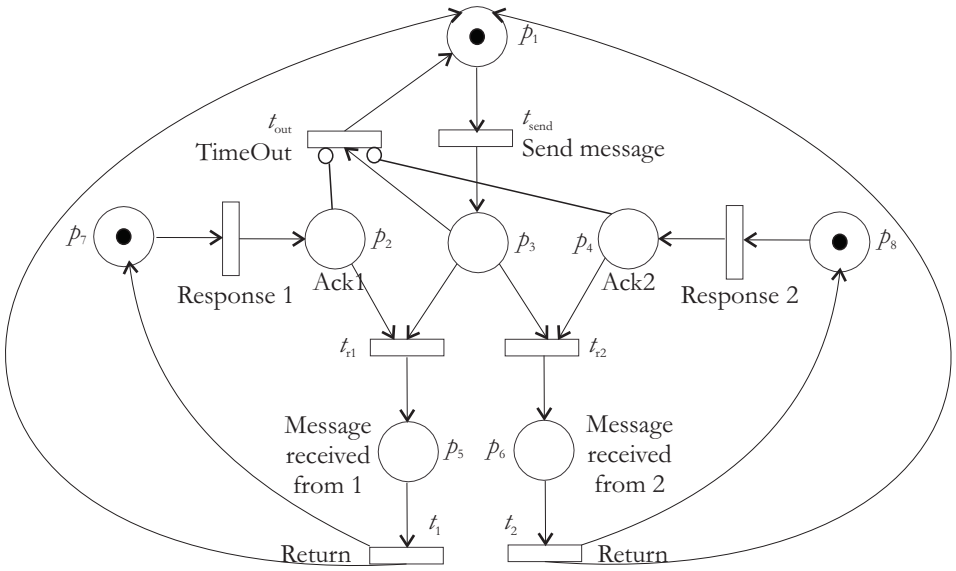
V predhodnjih razdelkih smo si ogledali Petrijeve mreže, kot jih je definiral njihov avtor C. A. Petri. Pri tem smo se v zgledu modela IF stavka in nekaterih kasnejših zgledih že seznanili s konstruktom *inhibirne povezave*, ki ne sodi v osnovno definicijo Petrijevih mrež. Ostale razširjene konstrukte navajamo v naslednjih razdelkih.

3.6.1 Posebni gradniki v Petrijevih mrežah

V razširitvah Petrijevih mrež pogosto zasledimo naslednji vsebinski razširjavi:

- stohastično pogojenost vejanja ali izvedbe akcije, s čimer pridemo do stohastičnih Petrijevih mrež,
- mehkost (angl. *fuzziness*) izvedbe akcije ali izpolnjenosti pogoja, s čimer pridemo do mehkih Petrijevih mrež.

Zaradi splošnosti našega dela omenjenih razširitev ne bomo obravnavali. Poseben primer Petrijevih mrež so *barvne Petrijeve mreže* (angl. *coloured Petri nets*), ki pa jih bomo obravnavali v naslednjem poglavju.



Slika 3.25: Model nedeterminističnega čakalnega sistema povzet po [6].

3.6.2 Časovne Petrijeve mreže

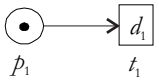
V časovnih Petrijevih mrežah vpeljemo *čas trajanja* posamezne akcije. Njihovo definicijo po [7] zapišemo na naslednji način:

Definicija 11 *šestorček $PN = (P, T, I, O, o(t_0), D)$ imenujemo razširjena časovna Petrijeva mreža, kjer P predstavlja končno množico pogojev, T končno množico akcij, I vhodno matriko in O izhodno matriko, pri čemer obe matriki glasita na akcije. $o(t_0)$ predstavlja začetno porazdelitev žetonov po indeksirani množici pogojev, D pa vektor nenegativnih števil vključujoč ničlo, ki posameznim akcijam določajo časovno trajanje.*

Trajanje posamezne akcije si interpretiramo na sledeči način. Akcija t_j s trajanjem d_j časovnih enot se začne izvajati, ko so izpolnjeni vsi pogoji, iz katerih vodijo vhodne povezave v opazovano akcijo t_j. Pogoji so (in morajo biti) izpolnjeni vse do konca trajanja akcije. šele po d_j časovnih enotah se njihova izpolnjenost razveljavi - žetoni se hipno prenesejo na izhodno stran opazovane akcije. Povedano drugače, žetoni v vhodnih pogojih ostanejo na svojih mestih d_j časovnih enot, šele nato pa se prenesejo preko akcije t_j v izhodne pogoje na nam že znani način. Seveda to ne pomeni, da se število žetonov skozi čas ohranja.

Na sliki 3.26 je prikazan enostaven primer Petrijeve mreže z akcijo t_1 s predvidenim trajanjem d_1 urinih period. Če žeton v času t_0 prispe v pogoj p_1, se bo iz pogoja preko akcije t_1 hipno preselil v časovni točki $t_0 + d_1$. Seveda bo to tega prišlo, če v intervalu $]t_0 + d[$ omenjenega žetona ne bo uporabila (prevzela)

kaka druga akcija, ki ji omenjeni pogoj tudi predstavlja potreben vhod. Če bi takšna konkurenčnost ali kompeticija obstajala, uspe z odvzemom žetona tista akcija, ki se preje izvede. če bi bilo torej proženje več akcij odvisno le od enega pogoja (žetona v njem), bi se izvedla tako le ena in sicer tista, z najkrajšim časom izvedbe.

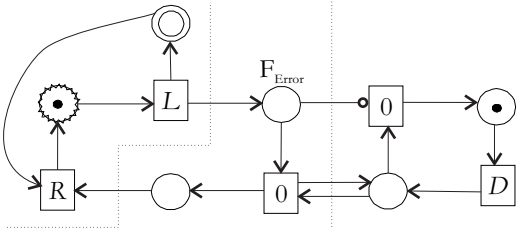


Slika 3.26: Graf Petrijeve mreže s časovnim trajanjem akcije t_1.

Oglejmo si zgled uporabe časovne Petrijeve mreže na primeru servisiranja popravljivega računalniškega sistema.

Zgled 6 *Predvideni čas do odpovedi sistema je L, čas servisiranja pa R časovnih period. Kontrolor periodično (s periodo D) kontrolira sistem in če je sistem v odpovedi, pokliče serviserja.*

Rešitev: Petrijeva mreža opisanega sistema je predstavljena na sliki 3.27, pri čemer levi del slike predstavlja sistem, osrednji del delovanje kontrolorja, desni del slike pa ciklus odsotnosti kontrolorja. Čas zadrževanja kontrolorja v osrednjem delu je hipen. Pri tem pogoj z nazobčanim robom naznačuje delujoče stanje celotnega sistema ($System_{ON}$), pogoj ponazorjen z dvojnim krogom pa začasno nedelujoče stanje celotnega sistema ($System_{OFF}$). Sistem v tem primeru ne vrši predvidene funkcije.



Slika 3.27: Graf Petrijeve mreže za periodično diagnostiko popravljivega računalniškega sistema.

Literatura

[1] M. Anu, "Introduction to modeling and simulation," in *Proceedings of 1997 Winter Simulation Conference*, 1997.

[2] L. Kleinrock and R. Gail, *Queuing systems, problems and solutions.* John Wiley & Sons, 1996.

[3] N. Zimic and M. Mraz, *Temelji zmogljivosti računalniških sistemov.* Založba FE in FRI, 2006.

[4] N. C. Hock, *Queuing Modelling Fundamentals.* Joh Wiley & Sons, 1996.

[5] J. L. Peterson, *Petri Net Theory and the Modeling of Systems.* Prentice Hall, 1981.

[6] T. Murata, "Petri nets: Properties, analysis and applications," 1989.

[7] W. G. Schneeweiss, *Petri Nets for Reliability Modeling.* LiLoLe Verlag, 1999.

[8] N. Jensen and L. Kristensen, *Coloured Petri Nets.* Springer, 1998.

[9] K. Jensen, "A brief introduction to coloured petri nets," in *Proceedings of the Third International Workshop on Tools and Algorithms for Construction and Analysis of Systems (TACAS '97)*, 1997.

[10] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks.* Prentice Hall, 2011.

[11] D. Božić, *Analiza in zgled uporabe programskega orodja CPNTools za postavljanje modelov dinamičnih sistemov.* Diplomsko delo FRI-UL, 2012.

[12] "CPNTools." `http://cpntools.org/`, 2012.

[13] M. Dolenc, *Verifikacija komunikacijskih protokolov na osnovi barvnih Petrijevih mrež.* Diplomsko delo FRI-UL, 2015.

[14] "Matematična ocena latence proizvajalca Rugged." `https://w3.siemens.com/mcms/industrial-communication/en/rugged-communication/Documents/AN8.pdf/`, 2015.

[15] "Matematična ocena latence proizvajalca O3b Networks." http://www.o3bnetworks.com/media/40980/white%20paper_latency%20matters.pdf/, 2015.

[16] "Internet dveh hitrosti." http://webfoundation.org/2015/10/net-neutrality-fails-to-load-web-foundation-response-to-todays-eu-vote//, 2016.

[17] "About Cacti." http://www.cacti.net//, 2016.

[18] P. Antončič, *Monitoriranje računalniških omrežij.* Diplomsko delo FRI-UL, 2012.

[19] "OMNeT++." www.omnetpp.org/, 2012.

[20] A. Varga, *Modeling and Tools for Network Simulation*, ch. OMNeT++, pp. 35–59. Springer-Verlag, 2010.

[21] A. Varga, *OMNeT++ User Manual, Version 4.1.* OpenSim Ltd., 2010.

[22] "INET." http://inet.omnetpp.org/, 2012.