

# Petrijeve mreže

Petrijeve mreže so orodja za modeliranje raznovrstnih sistemov. Tako jih lahko uporabimo za modeliranje:

- računalniških sistemov,
- programske opreme (diagramov poteka),
- kemijskih reakcij,
- rodovnikov...

## 1 Elementi Petrijeve mreže

Petrijeva mreža je četvorka

$$C = (P, T, I, O)$$

$P$  je množica *mest* (*places*)

$$P = (p_1, p_2, \dots, p_n)$$

$T$  je množica *prehodov* (*transitions*)

$$T = (t_1, t_2, \dots, t_m)$$

$I$  in  $O$  sta vhodna in izhodna funkcija, ki povezujeta prehode z mesti.

Vhodna funkcija  $I$  je preslikava prehoda  $t_i$  v množico mest  $I(t_i)$ , ki jih imenujemo vhodna mesta za prehod  $t_i$ .

Izhodna funkcija  $O$  je preslikava prehoda  $t_i$  v množico mest  $O(t_i)$ , ki jih imenujemo izhodna mesta za prehod  $t_i$ .

V množicah  $I(t_i)$  in  $O(t_i)$  se lahko določeno mesto pojavi večkrat – opraviti imamo s posplošeno množico. Naj bo  $p_j$  neko mesto iz množice vhodnih mest prehoda  $t_i$ . Izraz

$$\#(p_j, I(t_i)), p_j \in I(t_i)$$

pomeni število nastopanj mesta  $p_j$  v množici vhodnih mest prehoda  $t_i$  (če mesta  $p_j$  ni v množici vhodnih mest prehoda  $t_i$ , je  $\#(p_j, I(t_i)) = 0$ ). Podobno pomeni izraz

$$\#(p_j, O(t_i)), p_j \in O(t_i)$$

število nastopanj mesta  $p_j$  v množici izhodnih mest prehoda  $t_i$ .

Zdaj že lahko opišemo primer Petrijeve mreže:

$$C = (P, T, I, O)$$

$$P = (p_1, p_2, p_3, p_4, p_5)$$

$$T = (t_1, t_2, t_3, t_4)$$

$$I(t_1) = (p_1) \quad O(t_1) = (p_2, p_3, p_4, p_4)$$

$$I(t_2) = (p_2, p_3, p_4) \quad O(t_2) = (p_2)$$

$$I(t_3) = (p_4, p_4) \quad O(t_3) = (p_5)$$

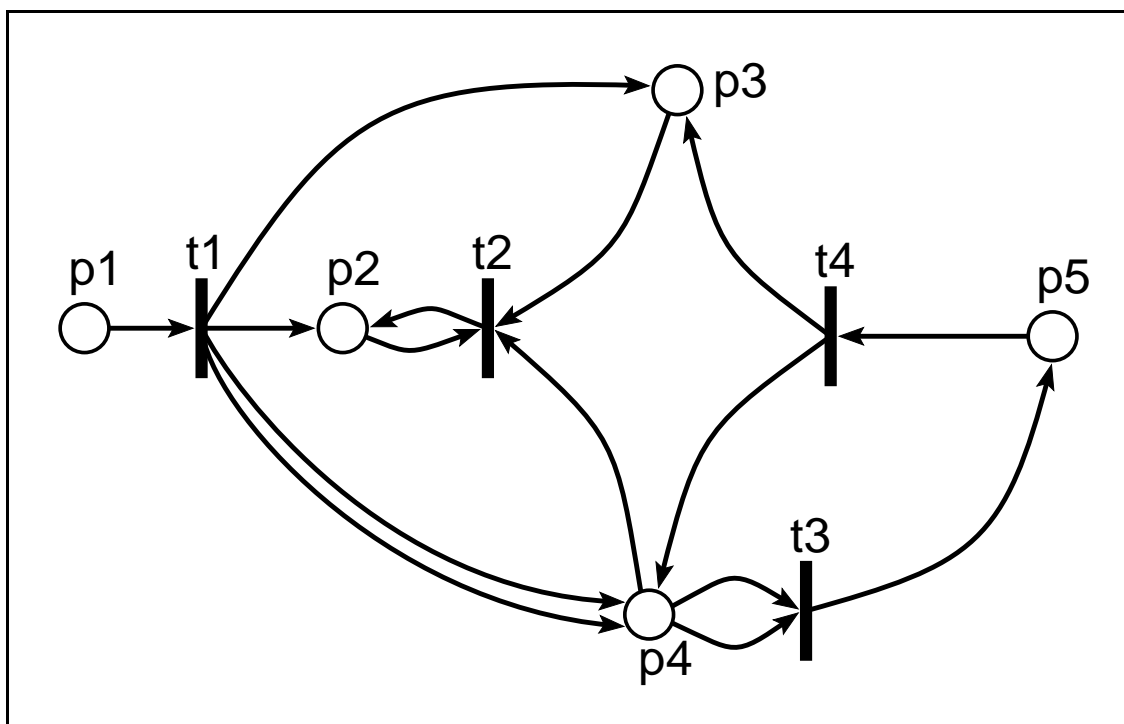
$$I(t_4) = (p_5) \quad O(t_4) = (p_3, p_4)$$

## 2 Predstavitev Petrijeve mreže z grafom

Petrijevo mrežo lahko predstavimo z grafom (omrežjem). Tako predstavitev bomo imenovali *Petrijev graf*. V Petrijevem grafu predstavimo mesta in prehode s točkami omrežja (mesta s krogi, prehode s pravokotniki). Vhodno funkcijo predstavimo z usmerjeno povezavo od mesta

do prehoda, izhodno funkcijo z usmerjeno povezavo od prehoda do mesta. V tem primeru imamo opraviti z *dvodelnim grafom* – množico točk grafa lahko razdelimo v dve podmnožici (v našem primeru mesta in prehode), tako da vsaka povezava v grafu vodi iz točke iz ene od podmnožic v točko v drugi podmnožici (med točkami znotraj podmnožic ni nobenih povezav).

Petrijevo mrežo iz prejšnjega primera predstavimo s Petrijevim grafom na sliki 1.



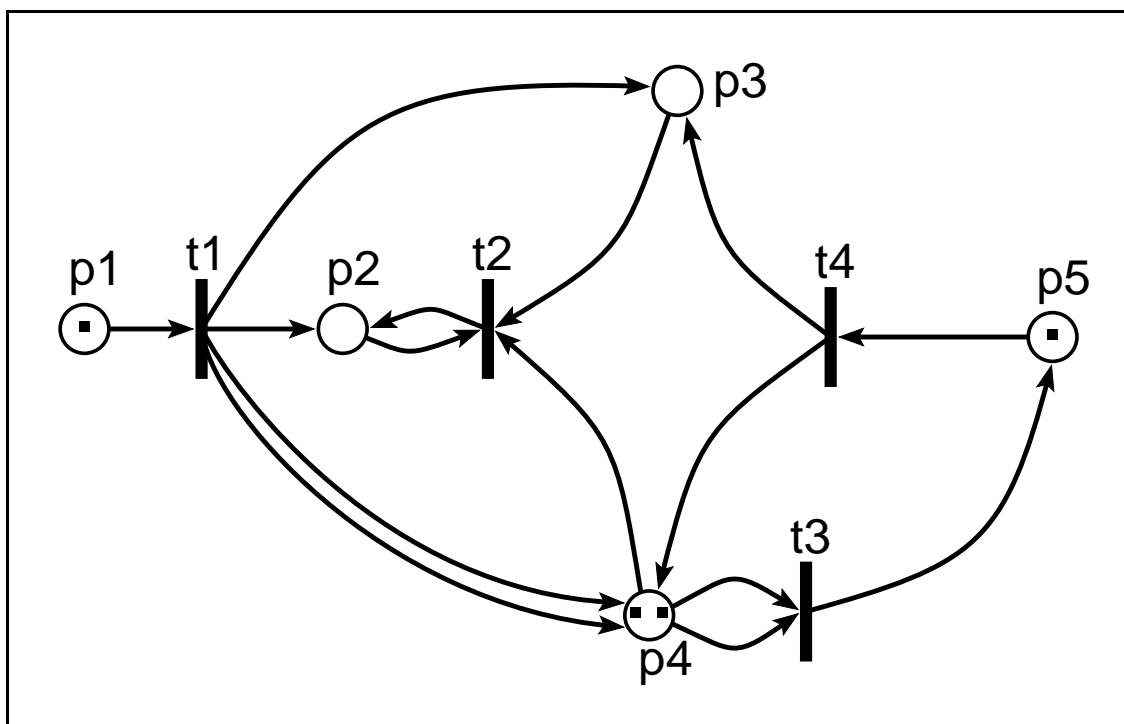
Slika 1: Petrijevi graf.

### 3 Označevanje Petrijeve mreže

Z označitvijo ( $\mu$ ) dodelimo mestom v Petrijevi mreži žetone (*tokens*). Vsakemu mestu lahko dodelimo poljubno število žetonov. Žetoni spadajo med osnovne elemente Petrijeve mreže, ki jo zato lahko razširjeno zapišemo kot petorko:

$$C = (P, T, I, O, \mu)$$

Označevanje  $\mu$  je torej preslikava  $\mu : P \rightarrow \mathbb{N}$ , ki jo lahko zapišemo kot vektor  $\mu = (\mu_1, \mu_2, \dots, \mu_n)$ . Število žetonov pri vsakem mestu označimo v Petrijevem grafu s številom pik ali ustreznim številom. Vzemimo Petrijevo mrežo na sliki 1. Slika 2 prikazuje Petrijevo mrežo, označeno z vektorjem  $\mu = (1, 0, 0, 2, 1)$ .



Slika 2: Primer označitve Petrijevega grafa.

## 4 Izvajanje Petrijeve mreže

Petrijevo mrežo z dano porazdelitvijo žetonov lahko *izvajamo*. Izvajanje Petrijeve mreže je sestavljeno iz zaporedij vžigov prehodov (*transition fire*) med katerimi se nekateri žetoni prenesejo z vhodnih na izhodna mesta izbranega prehoda.

Prehod lahko vžge samo, če je *omogočen*. Prehod je omogočen, če vsako od njegovih vhodnih mest vsebuje vsaj toliko žetonov, kot je povezav od mesta do prehoda. Pogoji za vžig mesta  $t_j$  je torej:

$$\mu(p_i) \geq \#(p_i, I(t_j)), p_i \in P$$

Ob vžigu omogočenega prehoda se žetoni, ki so omogočili vžig, odstranijo iz vhodnih mest in se prenesejo na vsako od izhodnih mest. Na vsako od izhodnih mest se prenese toliko žetonov, kolikor povezav vodi od prehoda do izbranega mesta. Na ta način preide označitveni vektor  $\mu$  v vektor  $\mu'$ . Število žetonov pri mestu  $p_i$  se po vžigu prehoda  $t_j$  spremeni po naslednjem obrazcu:

$$\mu'(p_i) = \mu(p_i) - \#(p_i, I(t_j)) + \#(p_i, O(t_j))$$

Z izvajanjem nadaljujemo, dokler je možen vžig vsaj pri enem prehodu.

## 5 Primeri izvajanja Petrijeve mreže

Vzemimo primer na sliki 2. V začetnem stanju lahko vžgejo prehodi  $t_1, t_3$  in  $t_4$ . Če vžge prehod  $t_1$ , preide vektor  $(1, 0, 0, 2, 1)$  v vektor  $(0, 1, 1, 4, 1)$ . V drugem koraku lahko vžgejo prehodi  $t_2, t_3$  in  $t_4$ . Če izberemo prehod  $t_2$ , dobimo označitveni vektor  $(0, 1, 0, 3, 1)$ .

Celotno zaporedje vžigov in ustreznih označitvenih vektorjev je:

	$(1, 0, 0, 2, 1)$
$t_1$	$(0, 1, 1, 4, 1)$
$t_2$	$(0, 1, 0, 3, 1)$
$t_3$	$(0, 1, 0, 1, 2)$
$t_4$	$(0, 1, 1, 2, 1)$
$t_2$	$(0, 1, 0, 1, 1)$
$t_4$	$(0, 1, 1, 2, 0)$
$t_2$	$(0, 1, 0, 1, 0)$

Napisali smo samo eno možno celotno izvajanje mreže. Pojavi se vprašanje, kaj storiti v primerih, ko je istočasno

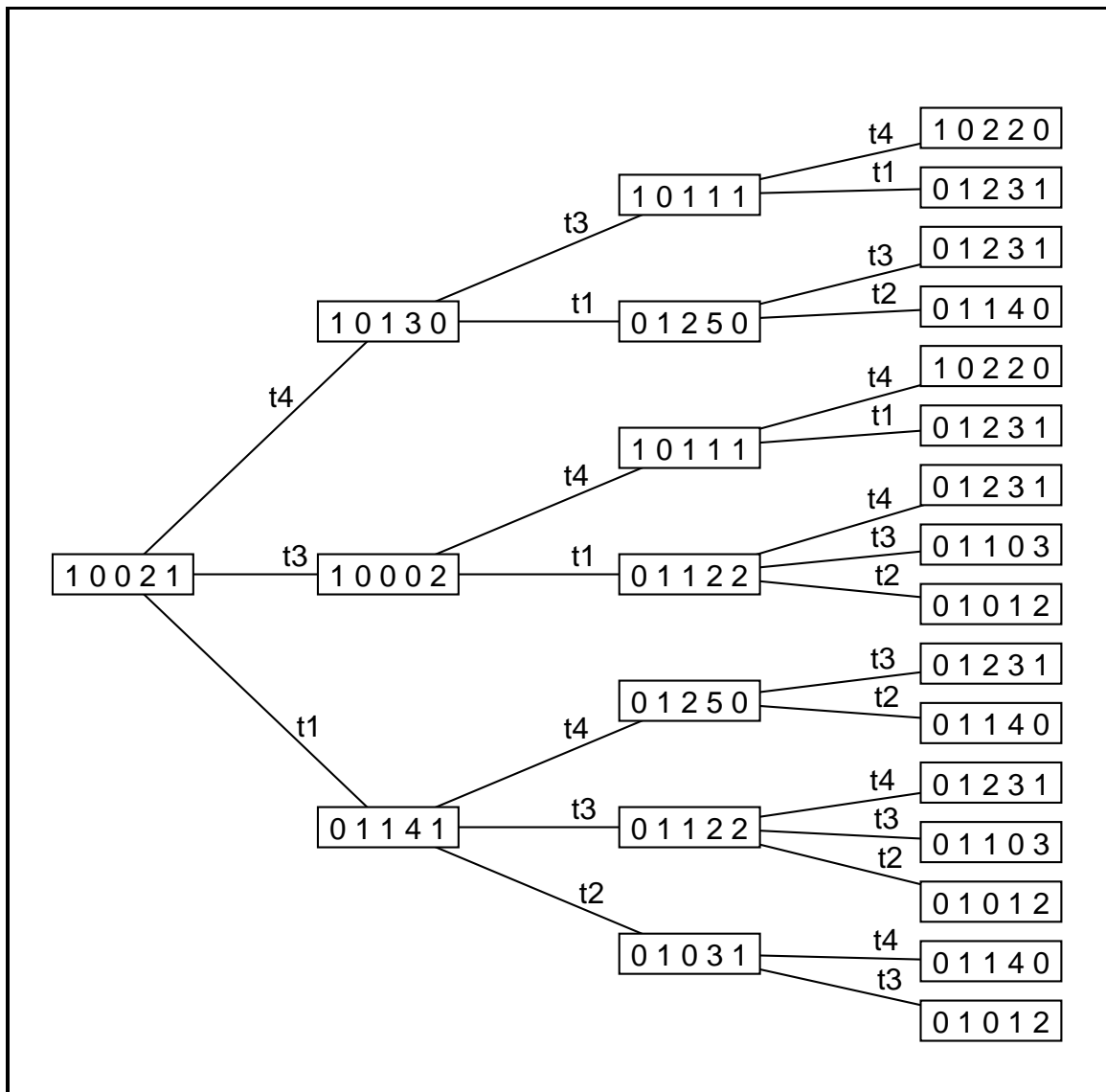


možnih več vžigov. V našem primeru smo vedno izbrali prvi možni vžig, v splošnem pa lahko vžig izberemo naključno.

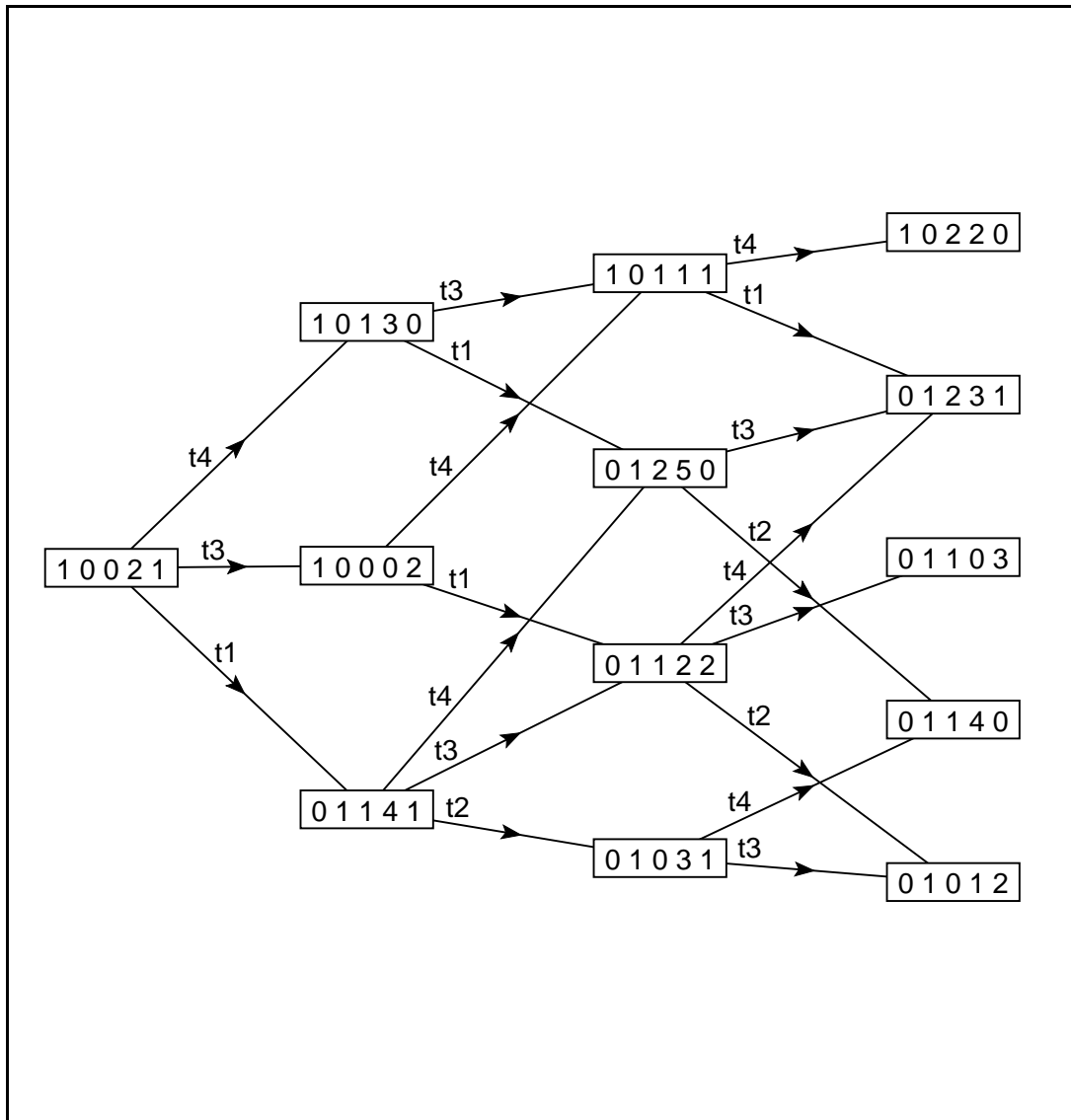
Zanimivo pa je seveda dobiti sliko vseh možnih izvajanj Petrijeve mreže. Za prikaz vseh možnih izvajanj Petrijeve mreže lahko uporabimo hierarhijo (drevo).

Koren drevesa predstavlja začetni vektor, njegovi sinovi pa so vsi možni vektorji, ki jih lahko dobimo z vžigom nekega omogočenega prehoda. Izbrani prehod je predstavljen z usmerjeno povezavo med očetom in sinom. V drevesu se določeni označitveni vektorji lahko pojavijo na večih mestih, saj lahko z različnimi zaporedji prehodov pridemo do istega označitvenega vektorja. Včasih je zato primerneje drevo predelati v usmerjeni graf, kjer se vsak označitveni vektor pojavi samo enkrat.

V našem primeru je vseh možnih izvajanj mreže kar 398, zato ne bomo prikazali celotnega drevesa. Sliki 3 in 4 prikazujeta vsa možna izvajanja z največ tremi vžigi (16 različnih izvajanj).



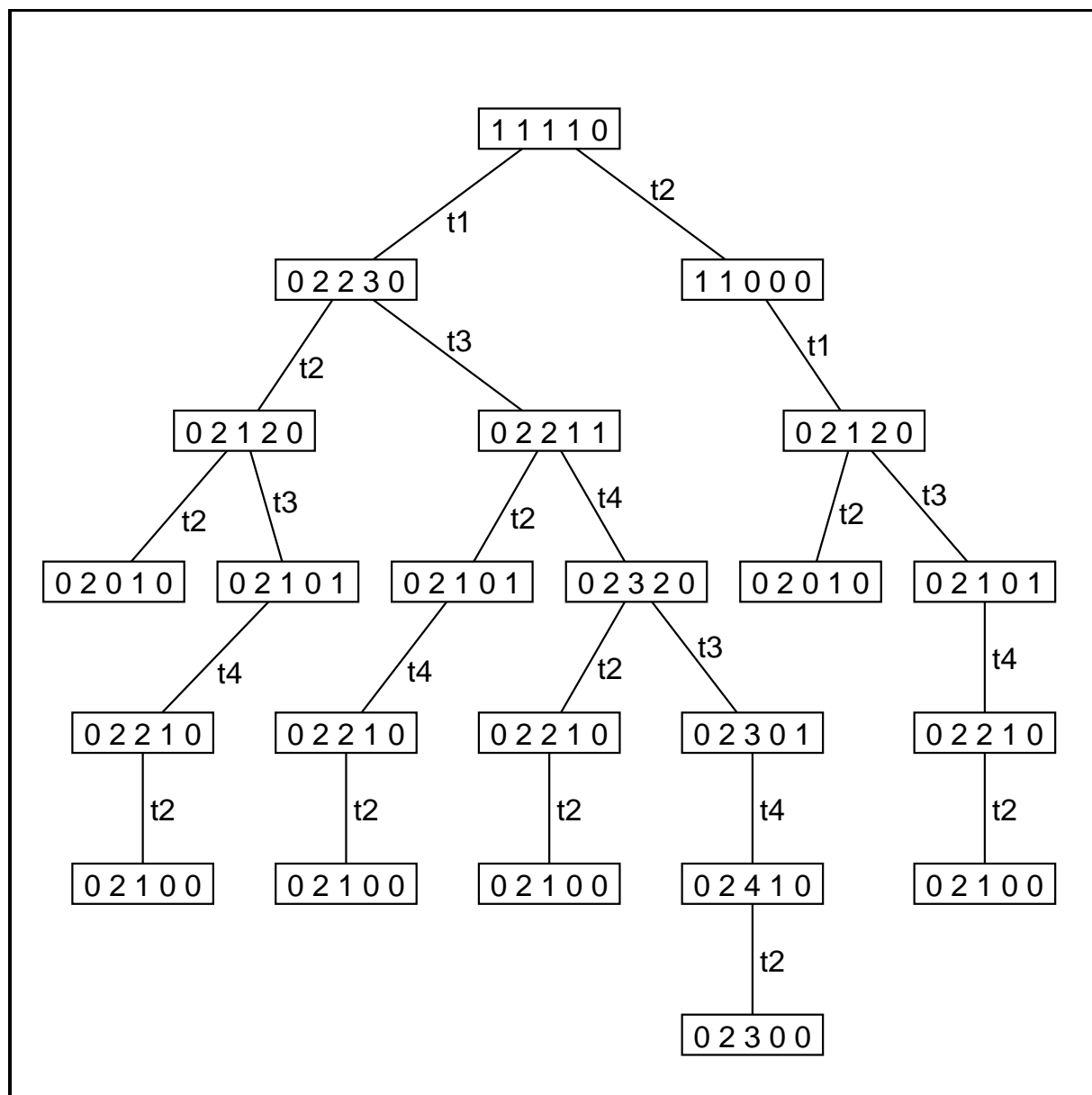
Slika 3: Drevo izvajanja Petrijeve mreže z začetnim vektorjem  $(1, 0, 0, 2, 1)$  do globine 3.



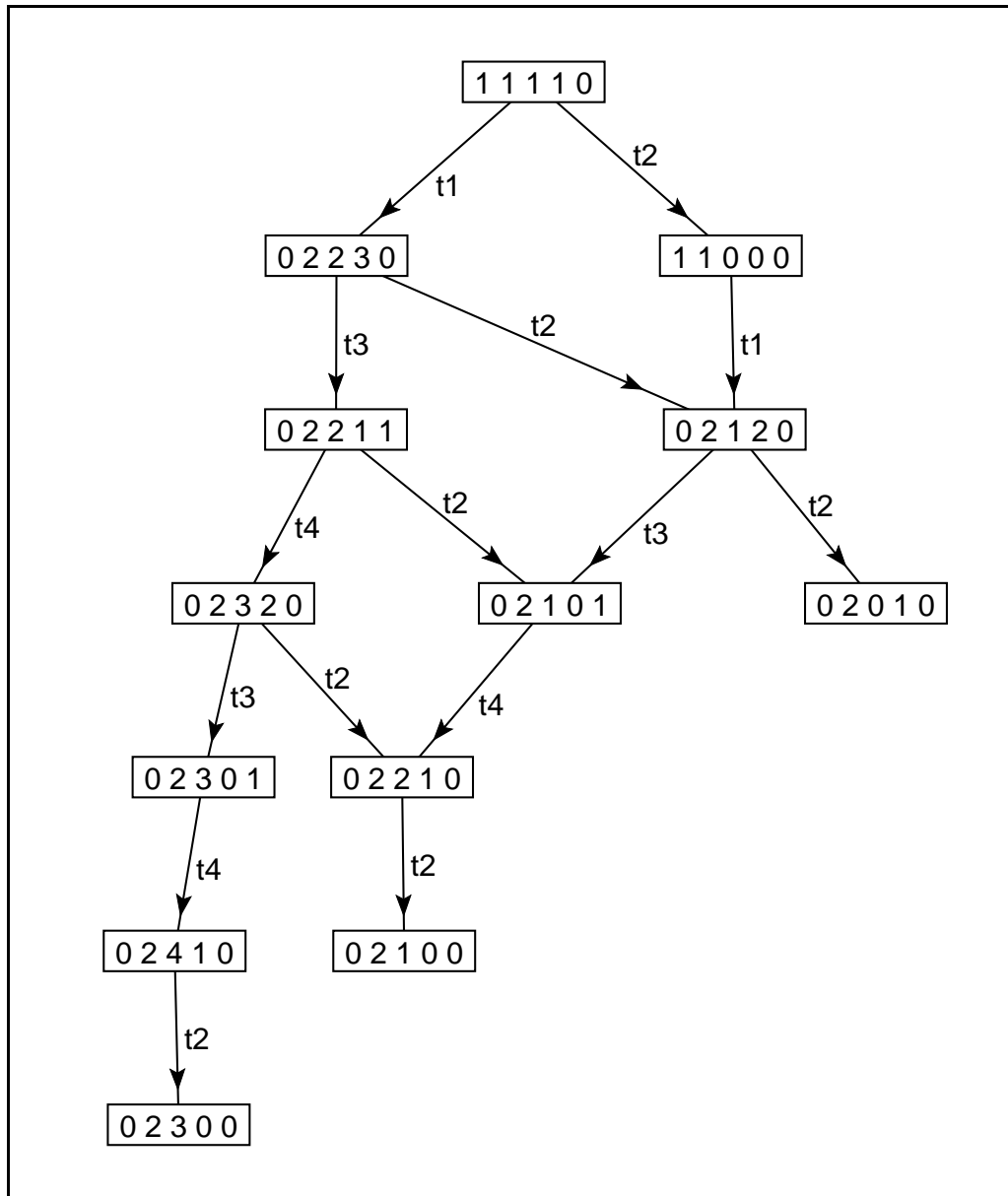
Slika 4: Usmerjeni graf izvajanja Petrijeve mreže z začetnim vektorjem  $(1, 0, 0, 2, 1)$  do globine 3.

Izberimo sedaj še začetni vektor, za katerega bomo lahko izpisali drevo vseh možnih izvajanj, npr. vektor  $(1, 1, 1, 1, 0)$ . Slika 5 prikazuje vsa možna izvajanja (7

različnih izvajanj).



Slika 5: Vsa možna izvajanja z začetnim vektorjem  $(1, 1, 1, 1, 0)$  – drevo izvajanj.



Slika 6: Vsa možna izvajanja z začetnim vektorjem  $(1, 1, 1, 1, 0)$  – usmerjeni graf.

Obstajajo seveda označitve Petrijevih mrež, na katerih izvajanje ni mogoče in označitve, ki povzročijo neskončno

izvajanje. V tem primeru dobimo neskončno drevo, če pa namesto drevesa uporabimo usmerjeni graf, tako da se vsak označitveni vektor pojavi le enkrat, obstajajo v ustreznem grafu cikli (graf ni acikličen).

## 6 Primer: Pet filozofov na večerji

Problem petih filozofov, ki želijo večerjati za isto mizo, je prvič omenil Dijkstra.

### Opis problema

Problem je naslednji:

Pet filozofov sedi za okroglo mizo, na kateri se nahaja veliko kitajske hrane. Med vsakima filozofoma se nahaja ena paličica (skupaj pet paličic). Če želimo jesti, potrebujemo dve paličici – torej mora filozof, če želi jesti, vzeti paličici s svoje leve in desne strani. Problem se pojavi, če vsak filozof vzame najprej paličico na svoji levi in čaka na paličico na svoji desni. V tem primeru pride do *zagate* (*deadlock*) – vsi filozofi bodo večno čakali in stradali. Naloga je poiskati zaporedje, po katerem naj filozofi jemljejo in vračajo paličici (na isto mesto od koder so ju vzeli), tako da ne pride do zagate.

Problem je treba predstaviti na tak način, da bo vsak filozof lahko jemal in vračal samo obe paličici hkrati (ali pa nobene).

## Rešitev s pomočjo Petrijeve mreže

Problem lahko predstavimo z naslednjo Petrijevo mrežo:

- Petrijevo mrežo sestavlja 15 mest. Vsak filozof je predstavljen z dvema mestoma,  $E_i$  predstavlja stanje, ko  $i$ -ti filozof je (stanje prehranjevanja, *Eating*),  $M_i$  predstavlja stanje, ko  $i$ -ti filozof čaka (stanje meditiranja, *Meditating*). Poleg teh desetih mest obstajajo še mesta  $C_1, \dots, C_5$ , ki predstavljajo prisotnost oziroma odsotnost paličice na svojem mestu.
- Petrijevo mrežo sestavlja 10 prehodov in sicer za vsakega filozofa dva prehoda: prehod iz stanja meditiranja v stanje prehranjevanja ( $M E_i$ ) in obratno ( $E M_i$ ). Filozof lahko preide iz stanja meditiranja v stanje prehranjevanja, če sta hkrati prosti obe njegovi paličici. V tem primeru označimo paličici za zasedeni.



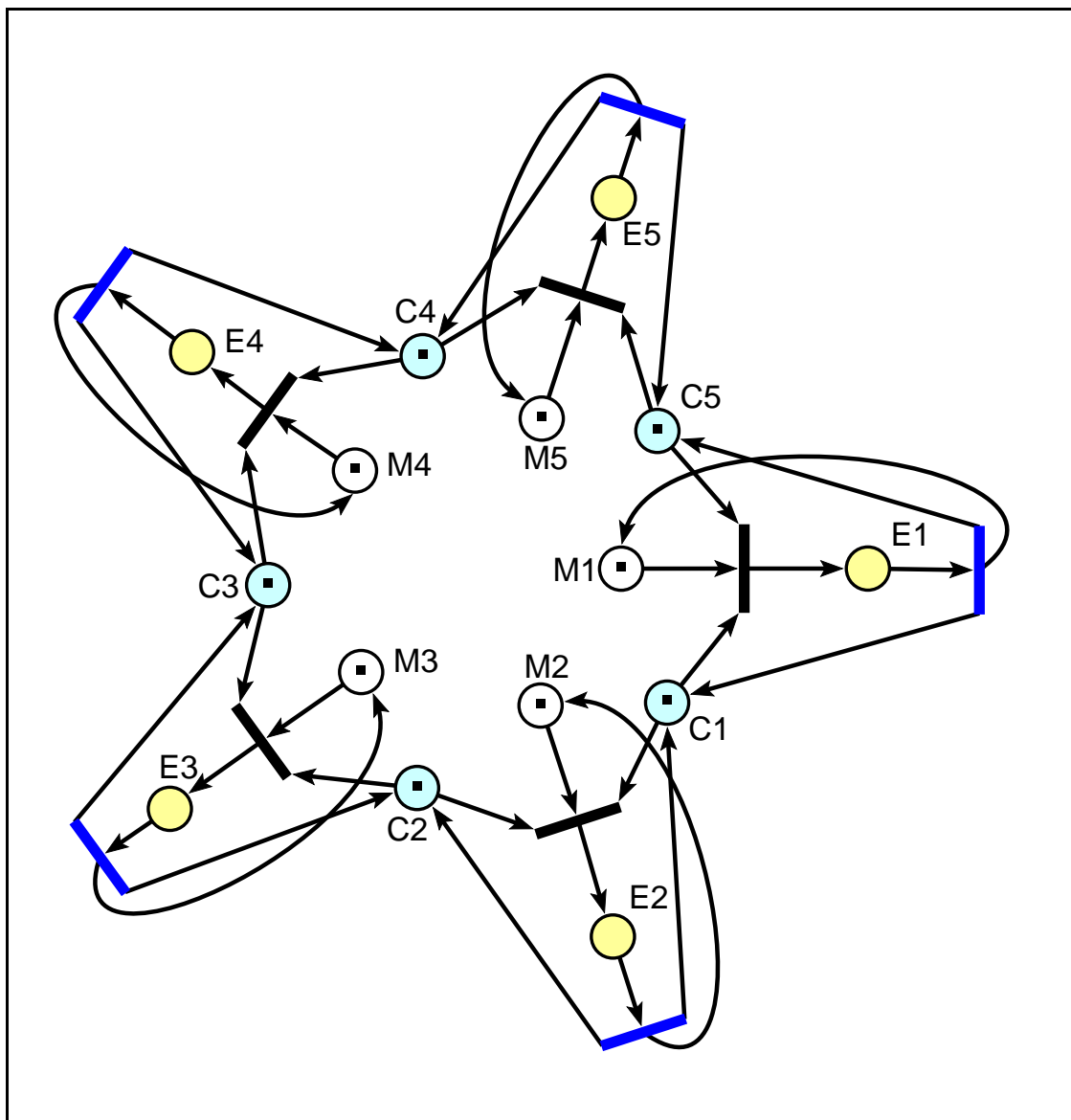
Ko pa filozof neha jesti, preide v stanje meditiranja, tako da vrne paličici spet na svoje mesto (paličici prosti).

- Kot rečeno, sestavlja označitveni vektor 15 mest:

$$(E_1, E_2, E_3, E_4, E_5, M_1, M_2, M_3, M_4, M_5,$$
$$C_1, C_2, C_3, C_4, C_5)$$

Začetna označitev predstavlja vse filozofe v stanju meditiranja in vse paličice na svojih mestih:

$$(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$$



Slika 7: Problem petih filozofov z začetno označitvijo.

Opisana Petrijeva mreža z začetno označitvijo je predstavljena na sliki 7. Taka označitev Petrijeve mreže povzroči neskončno izvajanje – vsak filozof lahko večkrat prehaja

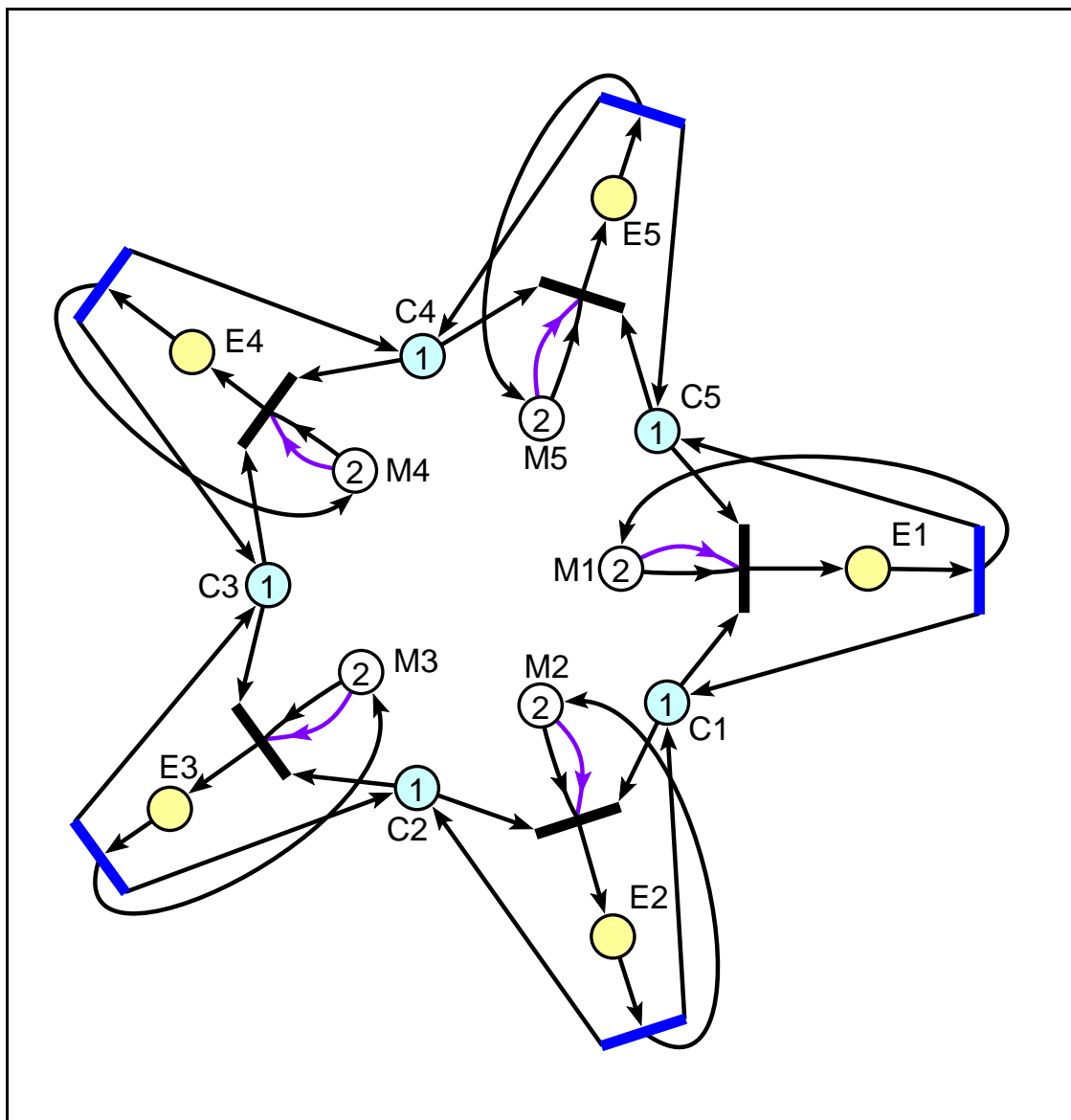
iz stanja prehranjevanja v stanje meditiranja in obratno. Omejitev, ki jo opisuje Petrijeva mreža, je le v tem, da dva sosednja filozofa ne moreta jesti istočasno.

### **Uvedba dodatnih omejitev**

Predpostavimo, da želimo 'nasititi' vseh pet filozofov ter da pustimo vsakemu filozofu jesti samo enkrat. V tem primeru uvedemo v izvajanje še dodaten pogoj, ki zagotovi končnost izvajanja: Vsakemu filozofu damo v začetnem stanju meditiranja 2 žetona ( $M_i = 2$ ), ki sta potrebna pogoja (poleg razpoložljivosti paličic), da lahko preide v stanje prehranjevanja (do ustreznega prehoda vodita tudi dve povezavi in ne ena kot prej). Ko se filozof vrne v stanje meditiranja, se na ustrezno mesto prenese (vrne) samo en žeton ( $M_i = 1$ ), torej filozof ne bo mogel še enkrat jesti. Začetni označitveni vektor je v tem primeru:

$$(0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1)$$

Opisana Petrijeva mreža z začetno označitvijo je predstavljena na sliki 8.



Slika 8: Začetna označitev Petrijeve mreže za problem petih filozofov z dodatno omejitvijo.

S tako omejitvijo dobimo 1760 različnih rešitev tega problema – vsako rešitev sestavlja 10 prehodov, vsak

filozof enkrat vzame paličici in ju enkrat kasneje vrne. V tem primeru se izvajanje Petrijeve mreže konča, ko so vsi filozofi enkrat jedli in so vse paličice spet na svojih mestih.

### **Odprava nepotrebnega vračanja paličic na koncu**

Verjetno pa drugi pogoj (vse paličice na svojih mestih) ni tako bistven – zanima nas, kako hitro lahko vseh pet filozofov spravimo do faze prehranjevanja. Z izvajanjem Petrijeve mreže torej zaključimo, čim zadnji filozof začne jesti.

V Petrijevo mrežo dodamo še eno mesto ( $A_1$ ), ki ima na začetku 5 žetonov. Vsakič, ko nek filozof začne jesti, zmanjšamo število žetonov na mestu  $A_1$  za 1.

Začetni označitveni vektor ima sedaj 16 členov:

$$(A_1, E_1, E_2, E_3, E_4, E_5, M_1, M_2, M_3, M_4, M_5, \\ C_1, C_2, C_3, C_4, C_5)$$

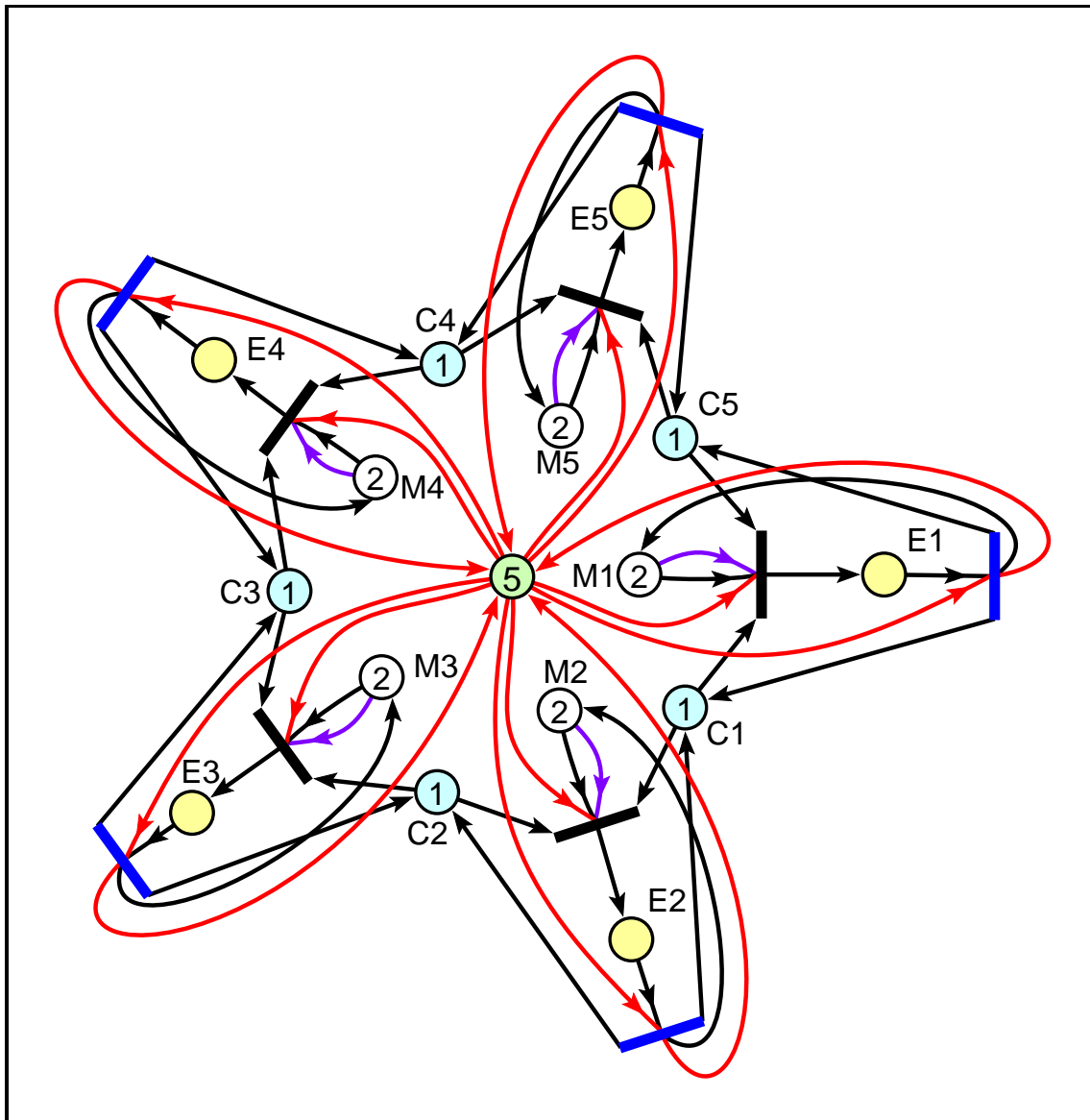
in je v tem primeru:

$$(5, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1)$$

Odločitev, ali dovolimo vračanje paličic (niso še vsi jedli,  $A_1 \neq 0$ ) ali pa z izvajanjem zaključimo (vsi so že siti,  $A_1 = 0$ ), uvedemo nato z dvema dodatnima povezavama pri vsakem filozofu:

- povezava med mestom  $A_1$  in prehodom  $EM_i$ ;
- povezava med prehodom  $EM_i$  in mestom  $A_1$ .

Na ta način preverimo, če število žetonov na mestu  $A_1$  ni enako 0. Po vžigu prehoda  $EM_i$  se število žetonov na mestu  $A_1$  ne spremeni (en žeton se s tega mesta odstrani, a se takoj vrne). Dopolnjena Petrijeva mreža je predstavljena na sliki 9).



Slika 9: Odpravljeno nepotrebno vračanje paličic na koncu.

S tem dodatkom postanejo tudi rešitve bolj zanimive. Število različnih izvajanj Petrijeve mreže je 1300, in sicer 460 z dolžino 8 in 840 z dolžino 9. Rešitve z dolžino

8 so tiste, ki jih pričakujemo za rezultat, saj se rešitve z dolžino 9 ločijo od njih samo po tem, da je nek filozof v eni od vmesnih faz vrnil paličici na svoje mesto, pa to ni bilo potrebno – v kasnejšem izvajanju teh paličic ni nihče potreboval. Za konec izpišimo še primer izbranega izvajanja z dolžino 8:

začetno stanje	(5, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1)
filozof 1 začne jesti	(4, 1, 0, 0, 0, 0, 0, 2, 2, 2, 2, 0, 1, 1, 1, 0)
filozof 3 začne jesti	(3, 1, 0, 1, 0, 0, 0, 2, 0, 2, 2, 0, 0, 0, 1, 0)
filozof 1 neha jesti	(3, 0, 0, 1, 0, 0, 1, 2, 0, 2, 2, 1, 0, 0, 1, 1)
filozof 5 začne jesti	(2, 0, 0, 1, 0, 1, 1, 2, 0, 2, 0, 1, 0, 0, 0, 0)
filozof 3 neha jesti	(2, 0, 0, 0, 0, 1, 1, 2, 1, 2, 0, 1, 1, 1, 0, 0)
filozof 2 začne jesti	(1, 0, 1, 0, 0, 1, 1, 0, 1, 2, 0, 0, 0, 1, 0, 0)
filozof 5 neha jesti	(1, 0, 1, 0, 0, 0, 1, 0, 1, 2, 1, 0, 0, 1, 1, 1)
filozof 4 začne jesti	(0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1)