

Sieci Petriego

Piotr CHRZĄSTOWSKI, Warszawa

Kiedy Amerykanie na początku lat 60-tych przygotowywali się do programu Apollo, stało się jasne, że bez komputera pokładowego w module lądującym nie ma co marzyć o sukcesie. Łatwo powiedzieć! Pamiętajmy, że w tamtych czasach marzeniem konstruktorów sprzętu było to, żeby komputer zmieścił się do windy i dawał się wnieść do biura bez konieczności burzenia ściany! Urządzenia te ważyły zazwyczaj grubo ponad tonę. Tymczasem wyniesienie na orbitę 1 kilograma ładunku było wielokrotnie droższe niż kilogram złota, a ponadto były pewne ograniczenia co do masy całkowitej modułu lądującego, więc wiadomo było, że to, co polecą liczyć tam na górze, musi być bardzo małe, a wtedy znaczyło to po prostu: bardzo słabe obliczeniowo.

Zatem koncepcja ukształtowała się następująco: na pokładzie statku kosmicznego leci jakieś maleństwo do doraźnych obliczeń, a główne rachunki przeprowadza komputer w Houston. No i pojawił się problem. Księżyc od Ziemi oddalony jest o jedną sekundę świetlną. W obie strony daje to 2 sekundy. Wieczność z punktu widzenia komputera. Jak zsynchronizować działania tak, żeby np. żaden z komputerów nie liczył na przestarzałych danych? Albo, co gorsza, częściowo na przestarzałych, a częściowo na aktualnych w przypadku rozsynchronizowania transmisji?

To wszystko skłoniło Amerykanów do poszukania jakichś prac teoretycznych dotyczących przetwarzania danych jednocześnie przez dwie jednostki centralne (albo i więcej). I tu niespodzianka! Okazało się, że na początku lat 60-tych w zasadzie nikt się tym jeszcze nie zajmował. Ludzie tak byli zajęci ulepszaniem tradycyjnej architektury komputera stojącego w jednym pomieszczeniu, że nie dojrżeli jeszcze po prostu takiej potrzeby. Maszyny były na tyle drogie, że wszelkie zastosowania były ukierunkowane na przetwarzanie zdecydowanie scentralizowane.

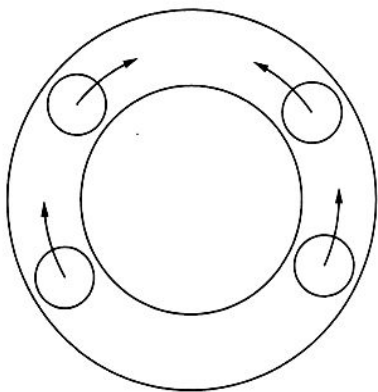
Dogłębne poszukiwania zakończyły się w końcu sukcesem. Odkryto mianowicie, że w Niemczech Carl Adam Petri w 1962 roku obronił pracę doktorską „Kommunikation mit Automaten” [Petri]. W pracy tej Petri przedstawił wyjątkowej urody model współbieżnego przetwarzania informacji. I się zaczęło! Dość powiedzieć, że dziś model ten pod nazwą *sieci Petriego* jest bodaj najpopularniejszym narzędziem do opisu asynchronicznych systemów współbieżnych. Jego popularność spowodowana jest zgrabnym połączeniem prostoty i wizualnej atrakcyjności z elegancką teorią matematyczną, która pozwala uzyskiwać zaskakująco głębokie i nieoczywiste rezultaty. Bibliografia dotycząca sieci Petriego dochodzi już do 10 tysięcy pozycji.

Zacznijmy od przykładu. Wyobraźmy sobie zapętloną rynnę (rys. 1), w której umieszczono 4 idealnie sprężyste kule o identycznej masie. Załóżmy też, że nadano im pewne jednakowe co do wielkości prędkości początkowe. Od zwrotów tych prędkości zależą zdarzenia, które zajądą w tym systemie. Kule, które poruszają się naprzeciw siebie zderzą się po pewnym czasie i zwroty ich prędkości ulegną zmianie. Naszym zadaniem będzie opisanie tego, co może się stać, a konkretniej przeanalizowanie konfiguracji, w których system może się znaleźć w wyniku zachodzących zderzeń. Będziemy przy tym abstrahowali od dokładnego położenia kul, a interesować będzie nas jedynie zwrot każdej z nich.

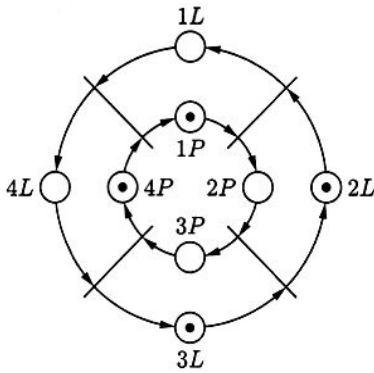
Konfiguracja zatem będzie uporządkowaną czwórką wartości ze zbioru $\{L, P\}$, gdzie i -tym elementem jest L , gdy i -ta kula porusza się w lewo, a P , gdy i -ta kula porusza się w prawo. Przyjmijmy, że stan początkowy, to $PLLP$, czyli kula pierwsza i czwarta poruszają się w prawo, a kula druga i trzecia w lewo. Stan ten nie zmieni się, aż nastąpi jedyne możliwe zdarzenie: kula pierwsza zderzy się z drugą, co doprowadzi nas do konfiguracji $LPLP$. W tej konfiguracji możliwe są już trzy zdarzenia: albo kula pierwsza zderzy się z czwartą, albo druga z trzecią, albo w końcu oba te zdarzenia nastąpią *równocześnie*. Można

W rzeczywistości był to zespół 3 identycznych komputerów. Trzech – na wypadek awarii. Przy rozbieżności wyników rezultat ustalało się przez głosowanie.

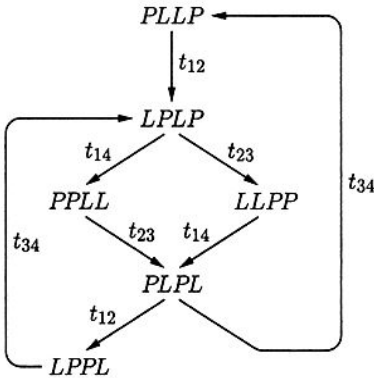
Asynchronicznych, czyli takich, w których akcje równoległe działających procesorów nie są taktowane wspólnym zegarem i są wykonywane niezależnie.



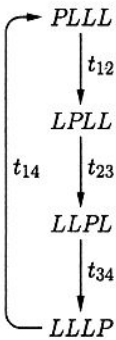
Rys. 1



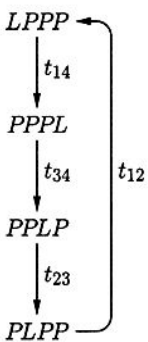
Rys. 2



Rys. 3



Rys. 4



Rys. 5

PPPP

LLLL

Rys. 6

tu mówić o prawdziwej równoległości: trzech niezależnych obserwatorów może te same dwa zdarzenia zaobserwować na te właśnie trzy różne sposoby, na przykład jeżeli rynna jest dostatecznie duża, a oni stoją odpowiednio daleko od siebie. Jest to nota bene jedna z możliwości definiowania równoległości zdarzeń: istnienie obserwatorów, którzy są w stanie zaobserwować dwa zdarzenia w dowolnym porządku. Nie ma wtedy między tymi zdarzeniami żadnego związku przyczynowo-skutkowego i możemy je nazwać *równoległymi*.

Z zasady zachowania momentu pędu wynika, że w każdej konfiguracji osiągalnej z naszej konfiguracji początkowej, całkowity moment pędu układu będzie taki sam, jak początkowy, czyli zero. Dwie kule będą się poruszać w lewo, a dwie w prawo. Pytanie, czy każdy taki układ zostanie osiągnięty?

Model sieci Petriego opisujący podaną sytuację przedstawiony jest na rysunku 2. Mamy tu 8 kółek opisujących 8 możliwych stanów: po dwa dla każdej kuli – nazywamy je *warunkami* – oraz 4 kreseczki opisujące 4 zdarzenia, czyli możliwe zderzenia między sąsiednimi kulami. Aby zdarzenie zaszło, muszą być spełnione oba warunki przypisane kółkom znajdującym się na początkach strzałek prowadzących do każdego z nich. Generalna zasada jest taka: aby aktywować dane zdarzenie, warunki znajdujące się na wejściach do niego muszą zachodzić, a warunki na wyjściach z tego zdarzenia nie mogą zachodzić. Zajście zdarzenia spowoduje, że jego warunki wejściowe przestaną zachodzić, a wyjściowe zaczną zachodzić. Fakt zachodzenia warunku przy symulacji sieci przedstawia się umieszczając pionek w kółku odpowiadającym temu warunkowi, a brak pionka oznacza, że warunek nie zachodzi. Rozmieszczenie pionków z rysunku 2 odpowiada zatem początkowej konfiguracji naszego systemu.

Na rysunku 3 przedstawiony jest *graf osiągalności* naszego układu. Węzłami tego grafu są konfiguracje osiągalne z konfiguracji początkowej, a krawędzie etykietowane są zdarzeniami, które powodują przejścia między odpowiednimi konfiguracjami. Przerwane strzałki odpowiadają zdarzeniom równoległym.

Widzimy, że graf ten zawiera 6 węzłów, czyli każda konfiguracja o zerowym momencie pędu jest osiągalna, gdyż $\binom{4}{2} = 6$, więc wybór dwóch prawych zwrotów spośród 4 można dokonać właśnie na 6 sposobów i wszystkie są osiągalne.

Jeżeli wybierzemy inną konfigurację początkową, np. *PLLL*, to graf osiągalności będzie przedstawiał się jak na rysunku 4. Wyczerpuje to wszystkie konfiguracje, w których 3 kule poruszają się w lewo, a jedna w prawo. Analogiczny graf osiągalności będący 4-elementowym cyklem dostaniemy dla konfiguracji początkowej *LPPP* (rys. 5). Jeśli chodzi o konfiguracje początkowe *LLLL* i *PPPP*, to ich grafy osiągalności przedstawione są na rysunku 6: są one izolowanymi pojedynczymi węzłami bez krawędzi. Faktycznie: przy takich zwrotach początkowych nie wystąpi żadne zderzenie.

Ten model sieci Petriego, historycznie pierwszy, okazał się użyteczny w wielu prostych przypadkach, ale niewystarczający do opisu większości praktycznych sytuacji. Tutaj przy założeniu skończoności zbioru zdarzeń i warunków dostawaliśmy zawsze skończone grafy osiągalności, co dość ograniczało możliwości modelowania praktycznych systemów. Wobec skończoności przestrzeni stanów osiągalnych większość problemów jest tu dość łatwa do rozstrzygnięcia. Zmiany, jakie zostały do tego modelu wprowadzone, spowodowały, że siła wyrazu zwiększyła się znacznie, ale wzrosły też problemy związane z analizą tych systemów. Definicja, którą poniżej przedstawimy, jest obecnie przyjmowana jako standardowy model sieci Petriego i jeżeli nie używa się żadnych przymiotników (takich jak np. sieci warunkowo-zdarzeniowe dla przedstawionego właśnie modelu), to przez sieci Petriego rozumie się właśnie to, co następuje:

Definicja. Sieć Petriego, jest to uporządkowana trójka $\langle P, T, F \rangle$, w której P, T to rozłączne, niepuste i skończone zbiory odpowiednio *miejsc* i *tranzycji*, a $F \subseteq (P \times T) \cup (T \times P)$ jest *relacją przepływu*.

Elementy zbioru P przedstawiamy graficznie jako kółka, elementy T jako kreski, a elementy F jako strzałki.

Ustalmy sieć Petriego $\mathcal{N} = \langle P, T, F \rangle$.

Markowaniem sieci \mathcal{N} nazwiemy każdą funkcję $M: P \rightarrow \mathbb{N}$.

W odróżnieniu zatem od sieci warunkowo-zdarzeniowych, dopuszczamy możliwość umieszczenia większej liczby pionków w każdym z miejsc. Parę (\mathcal{N}, M) , gdzie \mathcal{N} jest siecią Petriego, a M markowaniem tej sieci nazywamy systemem Petriego.

Niech $\bullet x = \{y \in (P \cup T) | (y, x) \in F\}$ i, analogicznie, $x \bullet = \{y \in (P \cup T) | (x, y) \in F\}$. W szczególności dla każdego zdarzenia $t \in T: \bullet t$ jest zbiorem wszystkich miejsc, z których do zdarzenia t prowadzą strzałki.

Powiemy, że zdarzenie $t \in T$ jest aktywne przy markowaniu M wtedy i tylko wtedy, gdy

$$\forall s \in \bullet t : M(s) \geq 1.$$

Aktywne przy markowaniu M zdarzenie może *zajść* lub, jak to się przyjęło mówić w teorii Petriego, *odpalić*. Wtedy markowanie M zmieni się w markowanie M' w następujący sposób:

$$M'(s) = M(s) - 1 \text{ dla } s \in \bullet t; \quad M'(s) = M(s) + 1 \text{ dla } s \in t \bullet \setminus \bullet t$$

oraz

$$M'(s) = M(s) \text{ dla pozostałych miejsc } s.$$

Czyli zdarzenie t zdejmie po jednym pionku ze wszystkich swoich miejsc wejściowych nie będących wyjściowymi, doda po jednym pionku do wszystkich swoich miejsc wyjściowych nie będących wejściowymi i nie zmieni zawartości pozostałych miejsc (w tym tych, które są jednocześnie wejściowe i wyjściowe). Jeśli zdarzenie t przeprowadza M na M' , to zapisujemy to jako $M[t]M'$. Efekt zajścia zdarzenia t ilustruje rysunek 7.

Markowanie M' jest *osiągalne* z markowania M w \mathcal{N} jeśli istnieje sekwencja

$$M_0, t_{i_1}, M_1, t_{i_2}, M_2, \dots, t_{i_k}, M_k$$

taka, że

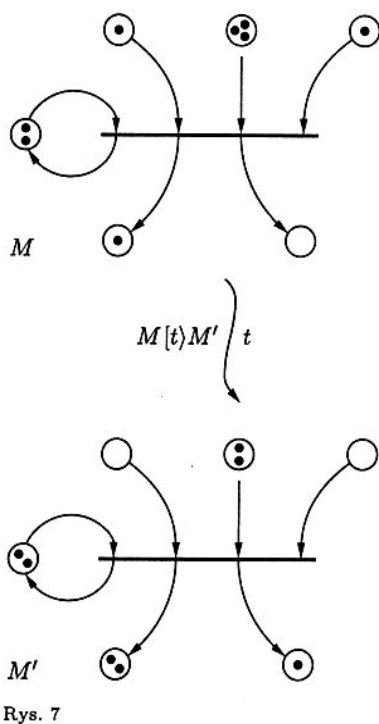
$$M = M_0, \quad M' = M_k, \quad \forall 1 \leq j < k : M_j[t_{i_j}]M_{j+1}.$$

Sytuację tę zapisujemy jako $M[t_{i_1} \dots t_{i_k}]M'$, a fakt, że z markowania M można wykonać sekwencję zdarzeń $t_{i_1} \dots t_{i_k}$ zapisujemy przez $M[t_{i_1} \dots t_{i_k}]$. Nie są to może szczyty inwencji notacyjnej, ale przyjęły się i po krótkim czasie człowiek do tych zapisów przywyka. Zbiór markowań osiągalnych z M oznaczamy przez $[M]$.

Jednym z najciekawszych problemów w teorii sieci Petriego jest postawiony w 1968 roku przez Karpa i Millera *problem osiągalności*. Formułujemy go następująco: dla zadanej sieci Petriego $\mathcal{N} = \langle P, T, F \rangle$ i dwóch markowań M, M' stwierdzić, czy $M' \in [M]$. Rozstrzygalność tego problemu była przez kilkanaście lat problemem otwartym. Problem ten okazał się na tyle trudny, że kilka opublikowanych pod koniec lat 70-tych dowodów okazało się niepoprawnych. Dopiero wynik Kosaraju z 1982 roku [SRK] oparty na wcześniejszych pracach Mayra z późniejszymi poprawkami Muellera rozstrzygnął pozytywnie ten problem. Problem osiągalności jest rozstrzygalny, choć jego złożoność algorytmiczna nie wyraża się funkcją pierwotnie rekurencyjną, jest więc ogromna. Ciekawe, że próby rozszerzenia modelu sieci Petriego o pewne użyteczne i ułatwiające modelowanie mechanizmy mogą popsuć jego rozstrzygalność.

Oto kilka przykładów takich modyfikacji:

- *Sieci z łukami wzbraniającymi*: jeśli taki łuk jest poprowadzony między miejscem s i zdarzeniem t , to aby zdarzenie t było aktywne, poza normalnymi wymaganiami, żąda się, żeby $M(s) = 0$. Czyli do aktywowania t konieczny jest brak pionków w miejscu s .



Rys. 7

- *Sieci z łukami czyszczącymi*: jeśli taki łuk łączy miejsce s ze zdarzeniem t , to zajście t powoduje wyzerowanie markowania w miejscu s , niezależnie od (dodatniej) liczby pionków, która w nim była.
- *Sieci z priorytetami*: ustalamy arbitralnie kolejność wykonywania zdarzeń w chwili, gdy zaistnieje konflikt między aktywnymi zdarzeniami konkurującymi o te same pionki.
- *Sieci z maksymalną regułą zachodzenia zdarzeń*: w każdym ruchu jesteśmy zmuszeni odpalać dowolny z maksymalnych pozdbiorów aktywnych zdarzeń.
- *Sieci samomodyfikujące się*: liczba pionków, którą zdejmujemy z miejsc lub dodajemy do miejsc przy zajściu zdarzenia, jest zależna od markowania, w którym to robimy.

W każdym z tych modeli problem osiągalności okazał się nierozstrzygalny [PCh].

Jednym z podstawowych problemów interesujących badaczy sieci Petriego jest *problem żywotności*. Formułujemy go następująco: problem żywotności polega na tym, by stwierdzić, czy w sieci \mathcal{N} markowanie M jest żywe, a markowanie M jest żywe w \mathcal{N} , jeśli

$$\forall t \in T : \forall M' \in [M] : \exists M'' \in [M'] : M''(t),$$

czyli markowanie jest żywe, jeśli niezależnie od stanu, który z niego osiągniemy (M'), będziemy mogli aktywować dowolne zdarzenie t . Innymi słowy żywotność markowania oznacza, że żadna z części sieci nie zostanie bezpowrotnie wyłączona, niezależnie od tego, jak złośliwie będziemy starali się to zrobić. Można co prawda w nieskończoność ignorować możliwość uruchomienia jakiegoś zdarzenia t , ale nie można odebrać nadziei na aktywowanie t . Jeśli od osiągnięcia dowolnego markowania M' zdarzenie t mogłoby podejmować wybory dotyczące kolejności zachodzenia zdarzeń, to mogłoby pokierować nimi tak, żeby siebie uaktywnić. Pokazano, że problem żywotności jest równoważny problemowi osiągalności, czyli oba są do siebie sprowadzalne. Z punktu widzenia modelowanych systemów brak żywotności oznacza niebezpieczeństwo bezpowrotnego wyłączenia jakiejś części systemu i jest uważany za usterkę.

Kolejnym problemem, którym się zajmiemy jest *problem ograniczoności*. Markowanie M jest *ograniczone* w \mathcal{N} , jeśli zbiór $[M]$ jest skończony. Innymi słowy, istnieje ograniczenie na liczbę pionków w każdym z miejsc. Jeżeli modelowany system Petriego jest nieograniczony (czyli markowanie nie jest ograniczone w sieci), to też nie jest to korzystna sytuacja: w praktyce oznacza to zazwyczaj niebezpieczeństwo przepełnienia buforów danych, które są związane z potencjalnie nieograniczonym miejscem. Czyli modelując systemy rzeczywiste nie chcemy arbitralnie ograniczać liczby pionków, które mogą się pojawić na każdym z miejsc, tylko chcemy przez ograniczoność sieci mieć zapewnione, że i tak nie musimy tego robić, bo reguły odpalenia nie pozwolą na wyprodukowanie zbyt wielu pionków w którymś z miejsc.

Problem ograniczoności jest rozstrzygalny za pomocą dość prostej procedury. Konstruuje się mianowicie graf osiągalności pilnując, żeby przy dodawaniu kolejnego węzła nie było na ścieżce prowadzącej z markowania początkowego markowania mniejszego od właśnie dodawanego. Mniejszego, czyli o wartościach mniejszych równych na wszystkich współrzędnych, a na co najmniej jednej – o wartości ostro mniejszej. Obecność takiego markowania na tej ścieżce oznaczałaby możliwość wygenerowania dowolnej liczby pionków na współrzędnej, która jest ostro większa. Z lematu Koeniga wynika, że nieskończony graf osiągalności musiałby zawierać nieskończoną ścieżkę prowadzącą od markowania początkowego. Z faktu zaś, że w dowolnym zbiorze wektorów z \mathbb{N}^n istnieje zawsze skończona liczba elementów minimalnych (lemat Dicksona), wynika, że przy konstrukcji nieskończonego grafu osiągalności zawsze natrafimy na sytuację, w której dodawany węzeł będzie istotnie większy od swojego poprzednika na ścieżce od markowania początkowego. Zatem albo przy konstrukcji grafu osiągalności natrafimy na węzeł większy od swojego poprzednika i wtedy sieć ograniczona nie jest, albo graf jest skończony i wtedy w skończonym czasie algorytm zakończy działanie przez wyczerpanie wszystkich osiągalnych węzłów.

Powiemy, że sieć \mathcal{N} jest *dobrze zbudowana*, jeśli istnieje dla niej markowanie M , które jest żywe i ograniczone. Pytanie o dobre zbudowanie sieci jest pytaniem strukturalnym: patrząc się na sam graf sieci chcemy wiedzieć, czy jest szansa na znalezienie dla niej dobrego (czyli w tym wypadku żywego i ograniczonego) markowania. Pytanie o dobre zbudowanie jest zasadniczym pytaniem *strukturalnej teorii sieci Petriego*, czyli takiej, która bada własności sieci na podstawie jej grafu, w oderwaniu od markowania początkowego. Najczęściej w problemach stawianych przez strukturalną teorię sieci Petriego markowania występują pod kwantyfikatorami.

Algorytm odpowiadający na pytanie o dobre zbudowanie sieci w ogólnym przypadku nie jest znany, choć sformułowano wiele mocnych warunków koniecznych. Istnieje jednak przepiękna charakteryzacja dobrego zbudowania w pewnej ważnej podklasie sieci Petriego, a mianowicie w *sieciach z wolnym wyborem*.

Często w sieci Petriego pojawia się *konflikt*. Jest to sytuacja, w której dwa zdarzenia współzawodniczą o jakieś pionki (rys. 8). Konflikt jest pojęciem dynamicznym, to znaczy zależnym od markowania. Na rysunku 9 przedstawiona jest sytuacja, w której konfliktu nie ma mimo, że istnieje potencjalna możliwość wystąpienia konfliktu. W strukturalnej teorii sieci Petriego zdefiniowanie konfliktu niezależnego od markowania jest szczególnie istotne. Definiujemy więc sieć Petriego z *wolnym wyborem*, jako sieć, w której wszystkie konflikty są konfliktami niezależnymi od markowania, a dokładniej: sieć \mathcal{N} jest z wolnym wyborem wtedy i tylko wtedy gdy

$$\forall p \in P |p \bullet| > 1 \Rightarrow \forall t \in p \bullet : \bullet t = \{p\},$$

sieć jest więc z wolnym wyborem, jeśli w każdej sytuacji gdy dwa zdarzenia mają wspólne miejsce wejściowe, jest ono jedynym miejscem wejściowym do tych zdarzeń. Rysunek 8 przedstawia sytuację wolnego wyboru, rysunek 9 – nie. Ilekroć pojawi się pionek na konfliktogennym miejscu, wszystkie zdarzenia z niego wychodzące są w konflikcie, bo to miejsce jest jedynym miejscem wejściowym do tych zdarzeń i wszystkie stają się aktywne. Zatem przy jednym pionku w takim miejscu albo wszystkie są parami w konflikcie, albo żadna para zdarzeń wyjściowych nie jest w konflikcie.

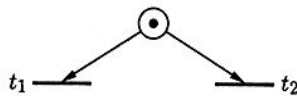
Właśnie w klasie sieci z wolnym wyborem znaleziono wspomnianą charakteryzację. Do wyrażenia jej potrzebujemy trochę matematyki oraz pewnego uproszczenia. Od tej pory zabraniamy miejscu być jednocześnie miejscem wejściowym i wyjściowym do jednego zdarzenia. Nie jest to istotne ograniczenie, gdyż łatwo można dla każdej sieci posiadającej takie miejsca podać równoważną jej sieć, w której taka sytuacja nie występuje. Ograniczenie to umożliwi nam za to bardzo użyteczną reprezentację macierzową sieci. Przez $\vec{1}_k$ oznaczamy wektor przestrzeni \mathbb{N}^k składający się z samych jedynek. Niech $|P| = n, |T| = m$.

Definicja. Macierz incydencji $C = c_{ij}, i = 1, \dots, n; j = 1, \dots, m$ o wartościach w zbiorze $\{-1, 0, 1\}$ określamy następująco:

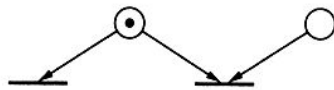
$$c_{ij} = \begin{cases} -1 & \text{jeśli } (s_i, t_j) \in F, \\ 1 & \text{jeśli } (t_j, s_i) \in F, \\ 0 & \text{w przeciwnym razie,} \end{cases}$$

czyli jeśli zdarzenie t_j pobiera pionek z miejsca s_i , to na pozycji (i, j) kładziemy minus jedynekę, jeśli t_j dodaje pionek do s_i , to jedynekę, a w przeciwnym razie 0. Innymi słowy macierz C składa się z m kolumn, przy czym kolumna j opisuje, jak zdarzenie t_j zmienia pionki na wszystkich n miejscach.

Będziemy rozważali słowa (czyli skończone ciągi) nad alfabetem $T = t_1, \dots, t_m$. Zbiór wszystkich takich słów oznaczamy przez T^* . Oznaczmy przez $\#(t, w)$ liczbę wystąpień symbolu t w słowie w . *Wektorem Parikha* dla słowa $w \in T^*$ nazwiemy wektor $\pi(w) = (\#(t_1, w), \#(t_2, w), \dots, \#(t_m, w))$. Na przykład dla $T = \{a, b, c, d\}, w = adcaac, \pi(w) = (3, 0, 2, 1)$.



Rys. 8



Rys. 9

Okazuje się, że jeśli w sieci Petriego \mathcal{N} o macierzy incydencji C zachodzi $M(w)M'$ dla pewnych markowań M, M' oraz słowa w , to równanie liniowe

$$M + Cx = M'$$

ma rozwiązanie w \mathbb{N}^m i wektor Parikha $\pi(w)$ jest jednym z pierwiastków tego równania. Powyższe równanie jest nazywane *równaniem stanu*.

Dowód tego twierdzenia jest dość prosty. Wystarczy zauważyć, że przemnożenie macierzy C prawostronnie przez T -wektor epsilonowy $\varepsilon_j = (0, \dots, 0, 1, 0, \dots, 0)^T$, mający jedynkę na pozycji j , daje kolumnę $C(\cdot, j) = t_j$, a kolumna ta reprezentuje zmianę markowania wywołaną przez zajście zdarzenia t_j . Widać, że przemnożenie macierzy przez dowolny T -wektor

$$a = (a_1, \dots, a_m) = \sum_{j=1}^m a_j \varepsilon_j, \text{ gdzie } \forall 1 \leq i \leq m : a_i \in \mathbb{N}$$

daje zmianę markowania spowodowaną przez a_1 krotne zajście zdarzenia t_1 , \dots , a_m -krotne zajście zdarzenia t_m . Pomijamy tu kwestię, czy zajście takiego ciągu zdarzeń byłoby w ogóle możliwe. Ale gdyby było (czyli gdyby nie zbrakło pionków do wykonania sekwencji zdarzeń o takim wektorze Parikha), to markowanie zmieniłoby się właśnie o Ca .

Tak więc mamy warunek konieczny osiągalności markowania M' z początkowego markowania M : równanie $M + Cx = M'$ musi mieć rozwiązanie w liczbach naturalnych.

Zauważmy też, że jeśli równanie $yC = 0$ ma rozwiązanie dla P -wektora niewiadomych y w liczbach naturalnych, to dla każdego markowania M i każdego markowania M' osiągalnego z M zachodzi równość iloczynów skalarnych $yM = yM'$. Jeśli bowiem M' jest osiągalne z M , to spełnione jest równanie stanu $M + Cx = M'$ dla pewnego wektora naturalnego x . Jeśli przemnożymy teraz lewostronnie to równanie przez y , to otrzymamy $yM + yCx = yM'$ i środkowy wyraz zeruje się na mocy definicji y i łączności mnożenia macierzy. Taki wektor y nazywany jest wektorem wagowym, albo niezmienniczym, systemu (\mathcal{N}, M) , gdyż każde markowanie osiągalne zachowuje niezmienniczo wagę, określaną jako iloczyn skalarny markowania osiągalnego przez y . Jeżeli dodatkowo $y \geq \vec{1}_n$, to taki niezmiennik wiąże ze sobą liczby pionków na wszystkich miejscach i w szczególności nie dopuszcza, aby ich liczba urosła ponad miarę. Każde markowanie w sieci, która posiada taki dodatni niezmiennik y jest więc ograniczone.

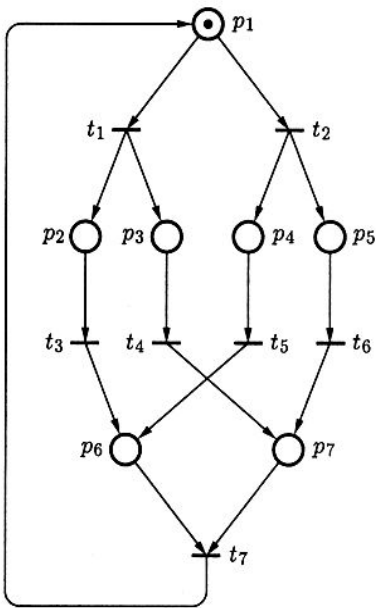
Używamy notacji, w której napis $x \geq y$ dla $x = [x_1, \dots, x_n]$ i $y = [y_1, \dots, y_n]$ oznacza, że dla każdego i zachodzi $x_i \geq y_i$.

Zauważmy, że jeśli chcemy, aby markowanie w sieci było dodatkowo żywe, to musi istnieć sekwencja t_{i_1}, \dots, t_{i_k} zawierająca przynajmniej jedno wystąpienie każdego zdarzenia ze zbioru T i taka, że jej wykonanie nie zmienia markowania. Dzieje się tak dlatego, że jeśli markowanie jest ograniczone, to zbiór markowań osiągalnych jest skończony, a my możemy odpalać nieskończone ciągi zdarzeń. Żywotność gwarantuje nam, że możemy zażądać odpalenia sekwencji zawierającej przynajmniej po jednym wystąpieniu każdego zdarzenia (stosujemy po prostu definicję żywotności kolejno do t_1, t_2, \dots, t_m wydłużając za każdym razem sekwencję). Osiągniemy w ten sposób pewne markowanie M_1 , być może różne od początkowego. Zaczniemy teraz od M_1 i wykonajmy kolejną sekwencję zawierającą wszystkie zdarzenia. Otrzymamy pewne markowanie M_2 . Ale zabawę tę możemy powtarzać tak długo, jak nam się tylko podoba, a markowań, którymi dysponujemy jest skończona liczba, więc któreś z nich się w tym ciągu powtórzy. Sekwencja zdarzeń, która będzie znajdować się między powtarzającymi się markowaniami będzie miała więc dwie cechy:

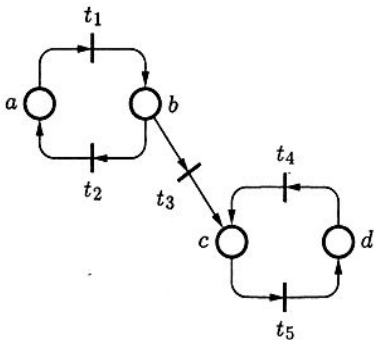
- będzie zawierała wszystkie zdarzenia,
- jej wykonanie nie zmienia markowania, od którego ją zaczęto wykonywać.

Wyrażając w języku algebry te dwie własności stwierdzamy, że istnieje zatem T -wektor $\vec{x} \geq \vec{1}_m$, taki że $M + C\vec{x} = M$, czyli $C\vec{x} = 0$.

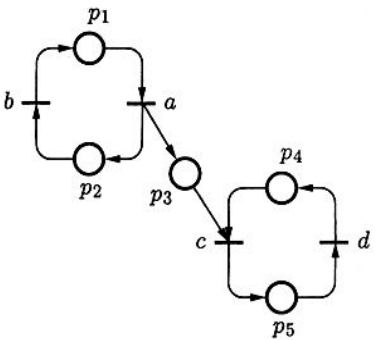
Widać więc, że zarówno lewe, jak i prawe dodatnie anihilatory macierzy C mają związek z dobrym zbudowaniem sieci. Intuicyjnie lewy anihilator jest



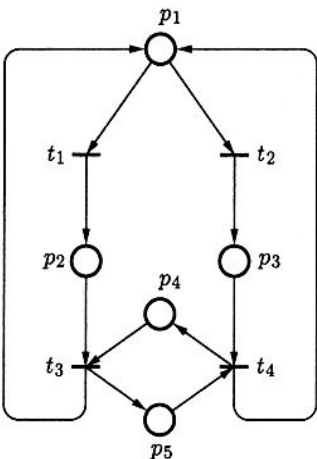
Rys. 10



Rys. 11



Rys. 12



Rys. 13

odpowiedzialny za ograniczonosc, prawy za żywotność (pod warunkiem, że istnieje lewy). Już w końcu lat 70-tych pokazano, że aby sieć była dobrze zbudowana konieczne jest, aby one istniały. Wiadomo było również, że nie są to warunki wystarczające. W roku 1990 M. Silva z Uniwersytetu w Saragossie podał nieoczekiwanie przepiękny warunek uzupełniający tamte dwa, który łącznie z nimi stał się warunkiem koniecznym i wystarczającym dobrego zbudowania sieci z wolnym wyborem. Dowód, który podał był mocno niepełny i niepoprawny, ale dwa lata później J. Desel z Monachium w swoim doktoracie przedstawił nie budzący wątpliwości dowód twierdzenia sformułowanego przez Silvę. Dowód jest kilkunastostrońcowy i nie sposób przedstawić go w tym artykule. Jednak samo sformułowanie twierdzenia niewątpliwie na to zasługuje. Przypomnijmy, że $|P| = n$, $|T| = m$ i zdefiniujmy $a = |F \cap (P \times T)|$, czyli a jest liczbą wszystkich strzałek wychodzących od jakiegoś miejsca do jakiegoś zdarzenia w \mathcal{N} .

Twierdzenie o rzędzie

Spójna sieć z wolnym wyborem \mathcal{N} o macierzy incydencji C jest dobrze zbudowana wtedy i tylko wtedy, gdy zachodzi koniunkcja trzech warunków:

1. $\exists \vec{x} \geq \vec{1}_m: C \vec{x} = 0$ (istnieje prawy dodatni anihilator C)
2. $\exists \vec{y} \geq \vec{1}_n: \vec{y} C = 0$ (istnieje lewy dodatni anihilator C)
3. $rank(C) = m + n - a - 1$ (rzęd macierzy incydencji jest o jeden mniejszy niż łączna liczba miejsc i zdarzeń minus liczba strzałek wychodzących z miejsca do zdarzenia)

Sformułowanie zaiste niezwykle i trudno nie uznać Silvy za autora tego twierdzenia. Wymyślenie takiego warunku, jak 3. nie może być dziełem przypadku.

Komentując to twierdzenie, którego pełny dowód można przeczytać w książce [DE], uzasadnimy konieczność tych trzech warunków. Rozważmy kolejno sieci z rysunków 10–13 (wszystkie oczywiście są z wolnym wyborem).

Sieć z rysunku 10 spełnia założenia twierdzenia. I rzeczywiście, łatwo się przekonać, że markowanie początkowe zawierające pionek w miejscu 1 jest żywe i ograniczone. Zauważmy, że jeśli odwrócimy kierunki strzałek, to rząd macierzy się nie zmieni (macierz sieci odwróconej będzie wyjściową macierzą przemnożoną przez -1). Lewe i prawe anihilatory pozostaną bez zmian. Zmieni się natomiast liczba a . I rzeczywiście ta odwrócona sieć dobrze zbudowana nie jest: choć każde markowanie jest w niej ograniczone, to żadne markowanie nie jest w niej żywe.

Sieć z rysunku 11 spełnia równanie rzędu (3.), ma lewy anihilator dodatni, ale nie ma prawego dodatniego anihilatora. I rzeczywiście ta sieć dobrze zbudowana nie jest: zdarzenie t_3 może zająć co najwyżej skończoną liczbę razy dla każdego markowania początkowego.

Sieć z rysunku 12 z kolei jest siecią dualną do sieci z rysunku 11. Zdarzenia i miejsca zostały zamienione rolami. Odpowiada to transponowaniu macierzy incydencji. Znowu równanie rzędu jest spełnione, tym razem istnieje prawy dodatni anihilator macierzy incydencji, ale nie ma lewego. Faktycznie, każde markowanie zawierające przynajmniej jeden pionek w miejscach 1 lub 2 oraz przynajmniej jeden pionek w miejscach 4 lub 5 jest żywe. Żadne jednak żywe markowanie nie jest w tej sieci ograniczone, gdyż miejsce p_3 nie może być ograniczone: jeśli markowanie ma być żywe, to trzeba niestety pozwolić na niczym nieskrępowane kręcenie się cyklu $t_1 t_2$, którego każdy obrót dodaje pionek do p_3 .

Sieć z rysunku 13 ma oba anihilatory, ale nie spełnia równania rzędu. Gdyby usunąć miejsca p_4 i p_5 wraz ze strzałkami, które do nich przylegają, to równanie byłoby spełnione i markowanie posiadające pionek na p_1 byłoby żywe i ograniczone. Dodanie tego cyklu powoduje, że rząd robi się o jeden za duży. Cykl ten likwiduje jeden stopień swobody, który mieliśmy zagwarantowany przez wolny wybór. Dokonując go trzeba jak gdyby przewidzieć, czy pionek (lub pionki) znajdują się w miejscu p_4 , czy p_5 . Podjęcie złej decyzji i skierowanie

pionka w tę gałąź, której wykonanie będzie wstrzymane przez cykl, może skutecznie zablokować sieć. Zauważmy, że gdybyśmy przesunęli cykl związany z miejscami p_4, p_5 w górę i zaczepili go o zdarzenia t_1, t_2 , to sieć byłaby już dobrze zbudowana, miała oba anihilatory, a mimo to równanie rzędu nadal nie byłoby spełnione. Tak, ale uzyskana sieć nie byłaby z wolnym wyborem!

Doprawdy urzekające, jak tradycyjne metody algebry liniowej pozwalają w elementarny, choć wysoce nieoczywisty sposób odkrywać tajemnice budowy sieci Petriego.

Bibliografia

- [Petri] Carl Adam Petri, Kommunikation mit Automaten., *Schriften des Institutes für Instrumentelle Mathematik*. Bonn, 1962
- [KM] R.M. Karp, R.E. Miller. Parallel Program Schemata. *J. Comput. System Sci.*, 3. 1969.
- [SRK] S.R.Kosaraju, Decidability of reachability in vector addition systems. *Proc. of the 4th Annual ACM Symposium on Theory of Computing*, San Francisco, CA, 1982.
- [DE] J.Desel, J.Esparza, Free Choice Petri Nets, *Cambridge University Press*. 1995.
- [PCh] Piotr Chrzastowski-Wachtel, Testing Undecidability of the Reachability in Petri Nets with the Help of 10th Hilbert Problem, *Lecture Notes in Computer Science 1639*. 1999.