Petri-Nets

Cocheci Cristiana Visan Sebastian

Cuprins

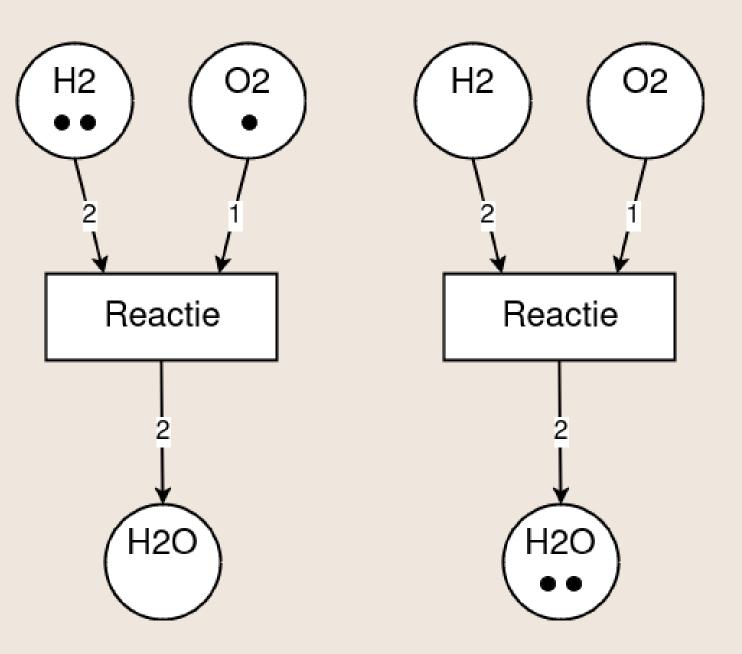
- 1 Introducere
- 2 Structura unui Petri-Net
- 3 State Graph
- 4 Work Clusters
- 5 Detalii Implementare
- 6 Exemple (Mutex, Filozofi & Santa)
- 7 Incheiere

Introducere

Petri Nets au fost create de catre Carl Petri in 1939 cu scopul de a modela reactii chimice.

Petri nets modeleaza programe concurente si contribuie la procesul de design al unui program concurent.

Un Petri Net este un graf directionat bipartit in care unele noduri pot contine tokenuri.



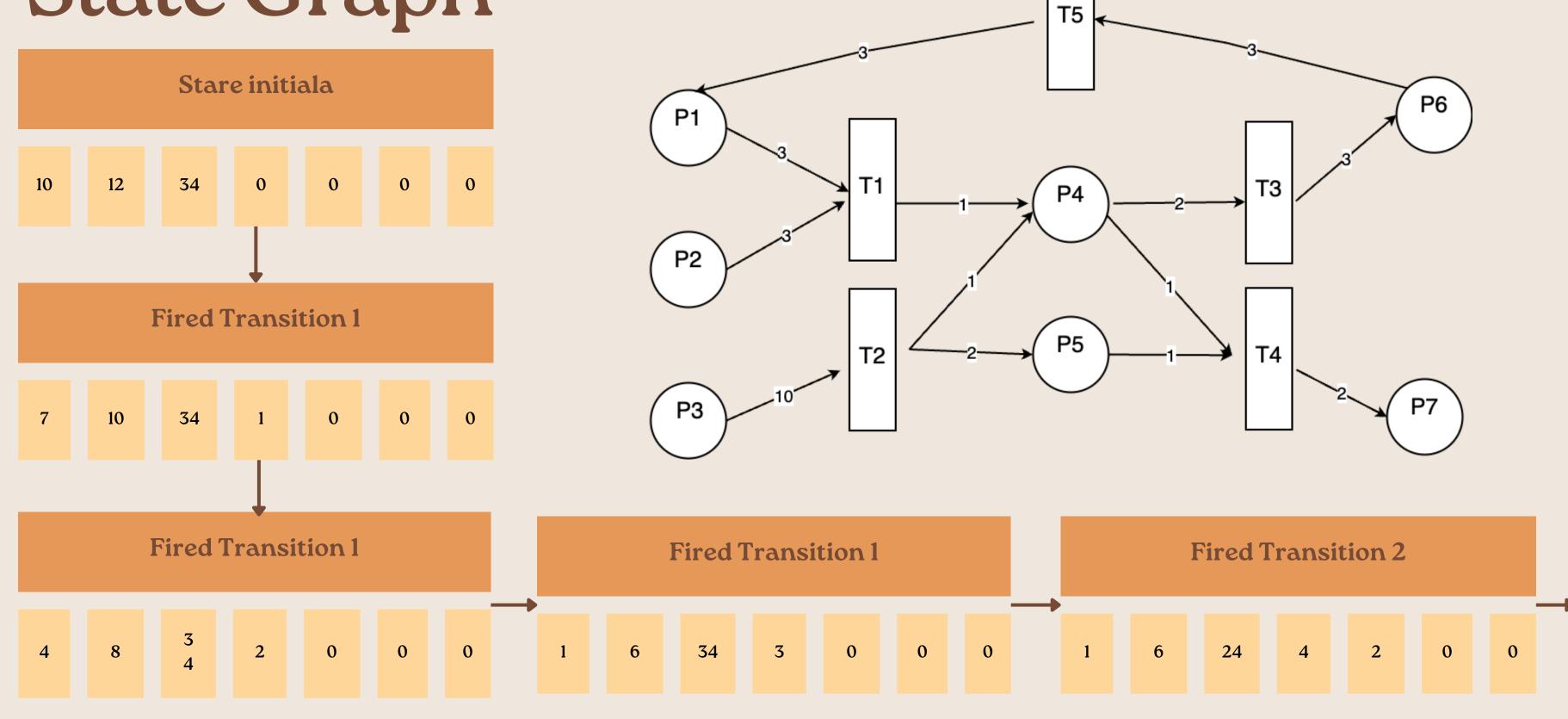
Structura unui Petri-Net

WEIGHTED EDGES PLACE NODES 000 • Pot contine oricate token-uri. • Au muchii catre Transition Nodes. • Atunci cand se activeaza o muchie exterioara, se trimite numarul corespunzator de tokenuri catre nodul tranzitie. 000 • Atunci cand se activeaza o muchie interioara, primeste numarul corespunzator de tokenuri.

TRANSITION NODES

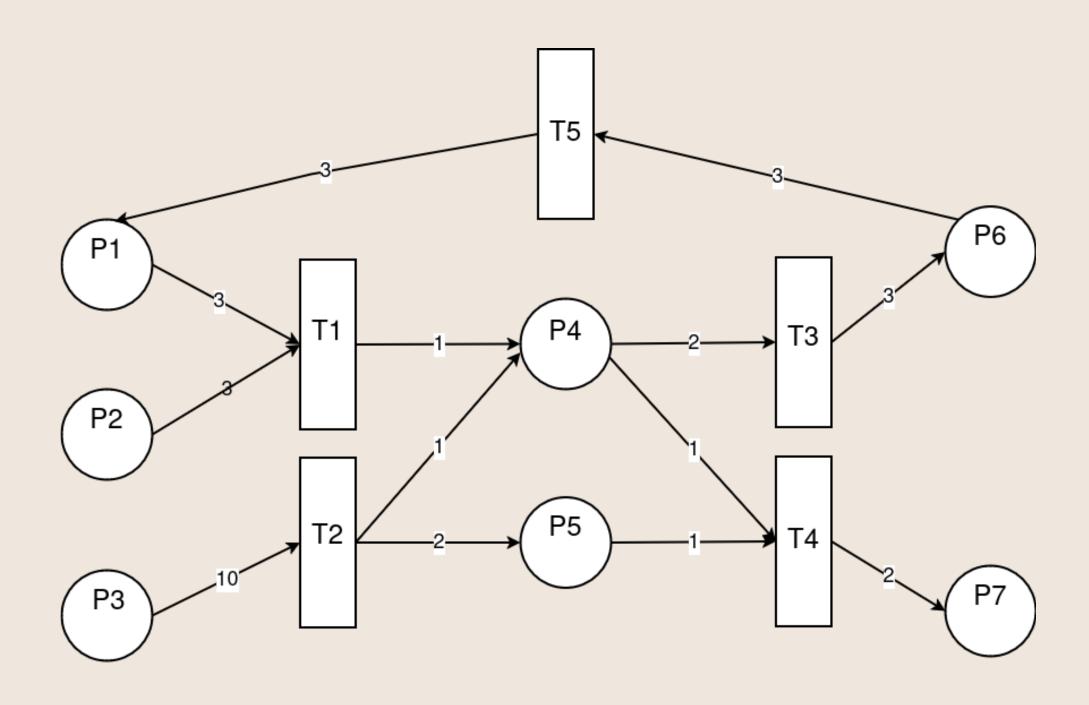
- Nu pot contine tokenuri.
- Un nod se activeaza atunci cand toate Place Nodurile care au muchie catre el au suficiente tokenuri corespunzatoare tranzitiilor lor.
- Observatie: Numarul de tokenuri nu se conserva neaparat.

State Graph



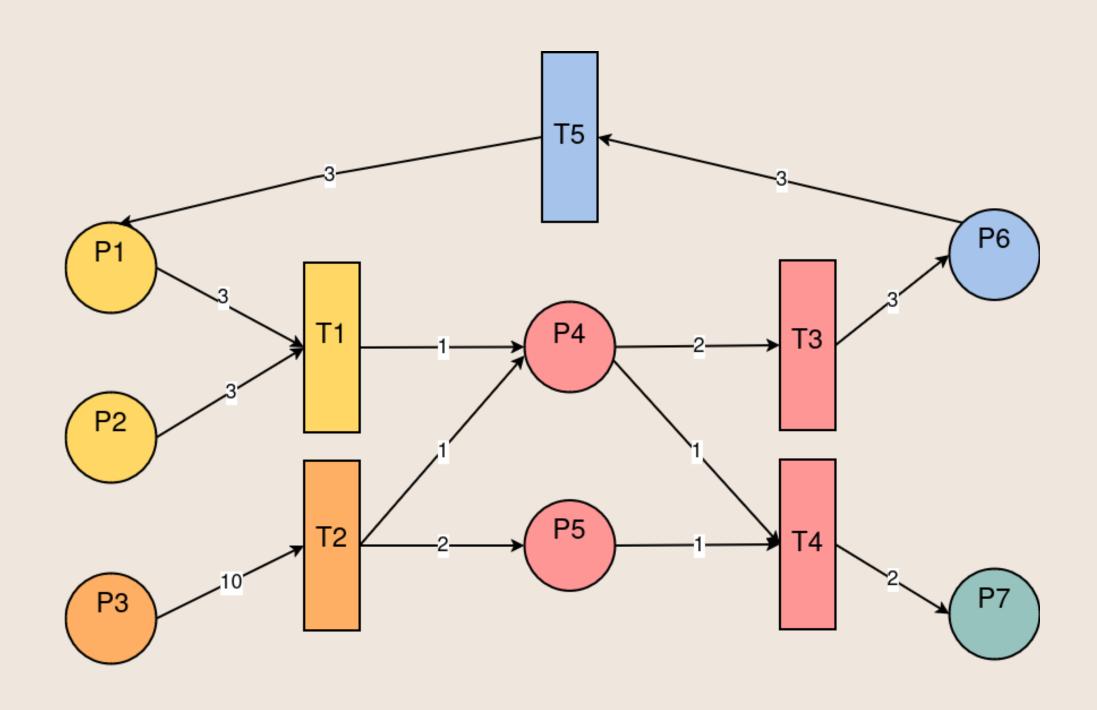
Work Clusters

Nodurile de tranzitie sunt partitionate a.i. un PlaceNode sa nu aiba muchii directionate catre doua partitii diferite.



Work Clusters

Nodurile de tranzitie sunt partitionate a.i. un PlaceNode sa nu aiba muchii directionate catre doua partitii diferite.



NONDETERMINISM

In cazul in care mai multe tranzitii pot fi activate in acelasi momemt de timp, alegerea se face aleator (nedeterminist).

WORK CLUSTERS

- Nodurile de tranzitie sunt partitionate a.i. un PlaceNode sa nu aiba muchii directionate catre doua partitii diferite.
- Fiecare partitie este echivalenta unui Work Cluster, respectiv unui thread unic.
- Fara aceasta partitionare apar race conditions.

Detalii de Implementare

GO CHANNELS

- Atunci cand se declanseaza o tranzitie, notificam prin canale workclusterele afectate ca trebuie sa verifice daca pot da fire la randul lor.
- Pentru a opri executiea unui Petri-Net, folosim un canal pe care anuntam momentul finalizarii:
 - o cand nu se mai produce o tranzitie prea mult timp (patience)
 - o cand expira perioada acceptata de rulare (timeout)

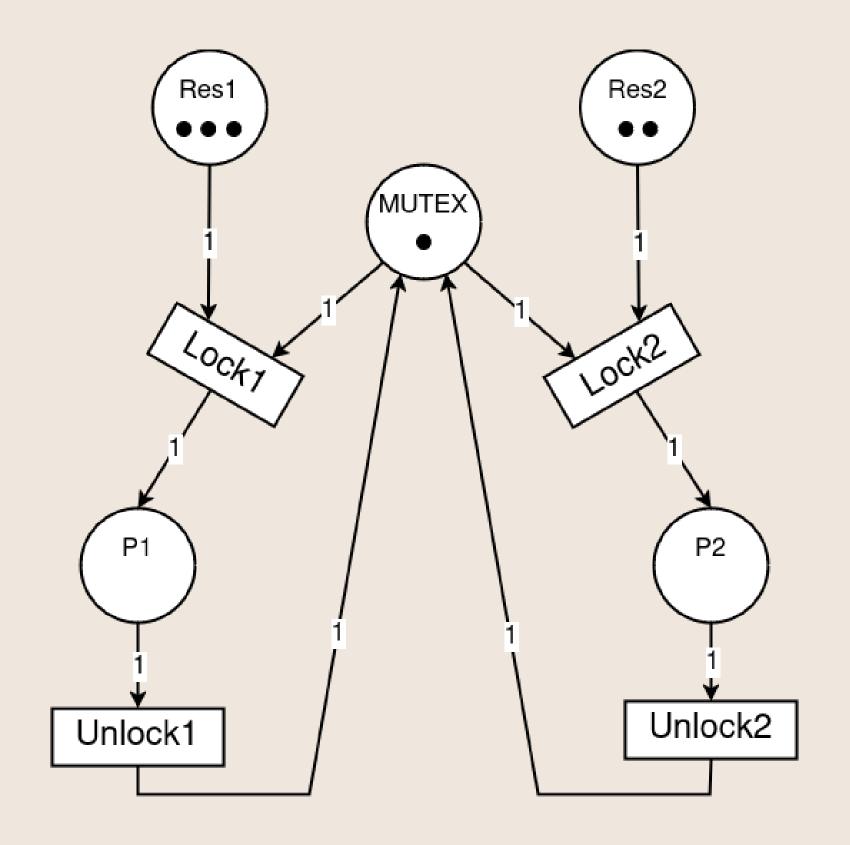
STATE GRAPH

O stare reprezinta numarul de tokenuri care se afla la un moment de timp in fiecare Place nod.

In implementarea nostra, simulam graful de stari afisand intr-un fisier separat toate starile prin care trece Petri-Netul.

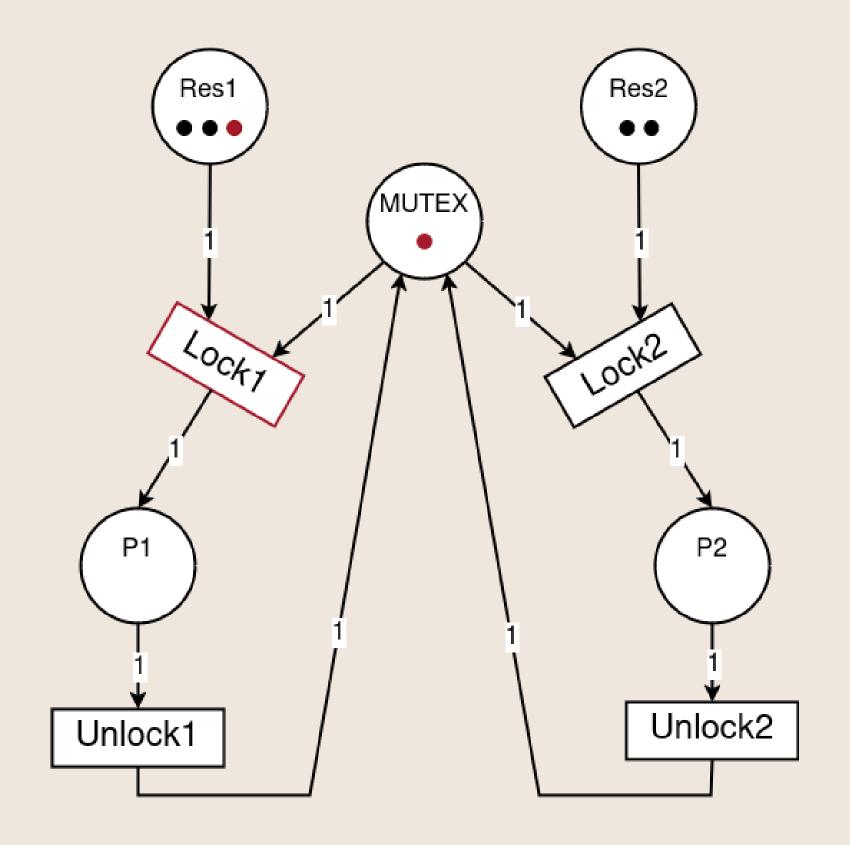
Putem modela excluziunea mutuala folosind un nod de tipul place care are exact un jeton (cheia lacatului).

Odata ce o tranzitie foloseste acest token, toate celelalte tranzitii care depind de cheie sunt blocate.



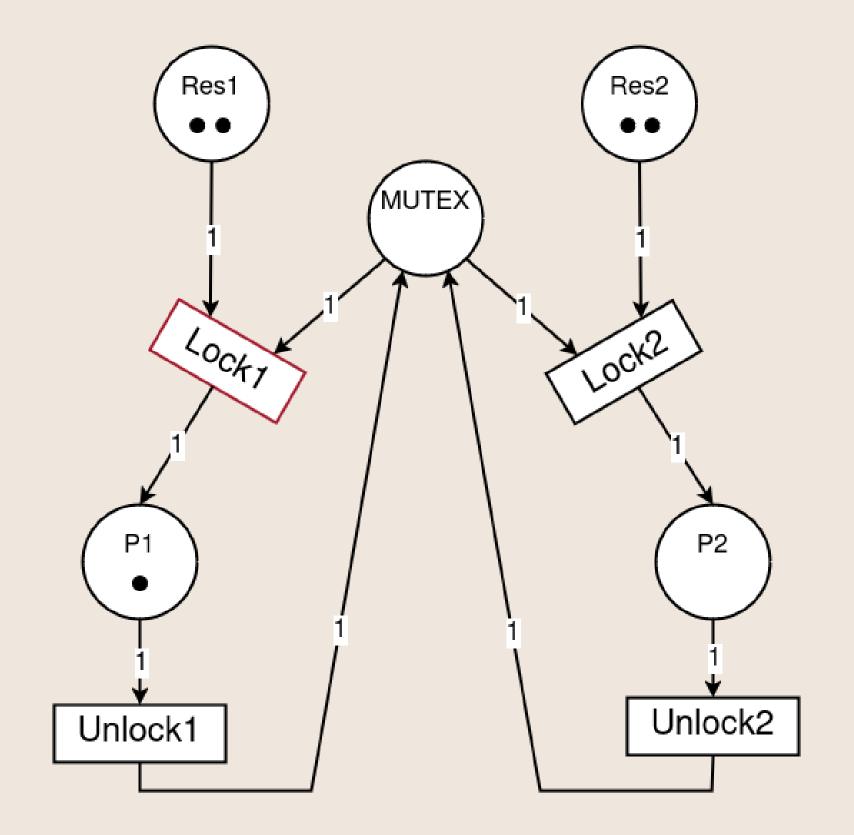
Putem modela excluziunea mutuala folosind un nod de tipul place care are exact un jeton (cheia lacatului).

Odata ce o tranzitie foloseste acest token, toate celelalte tranzitii care depind de cheie sunt blocate.



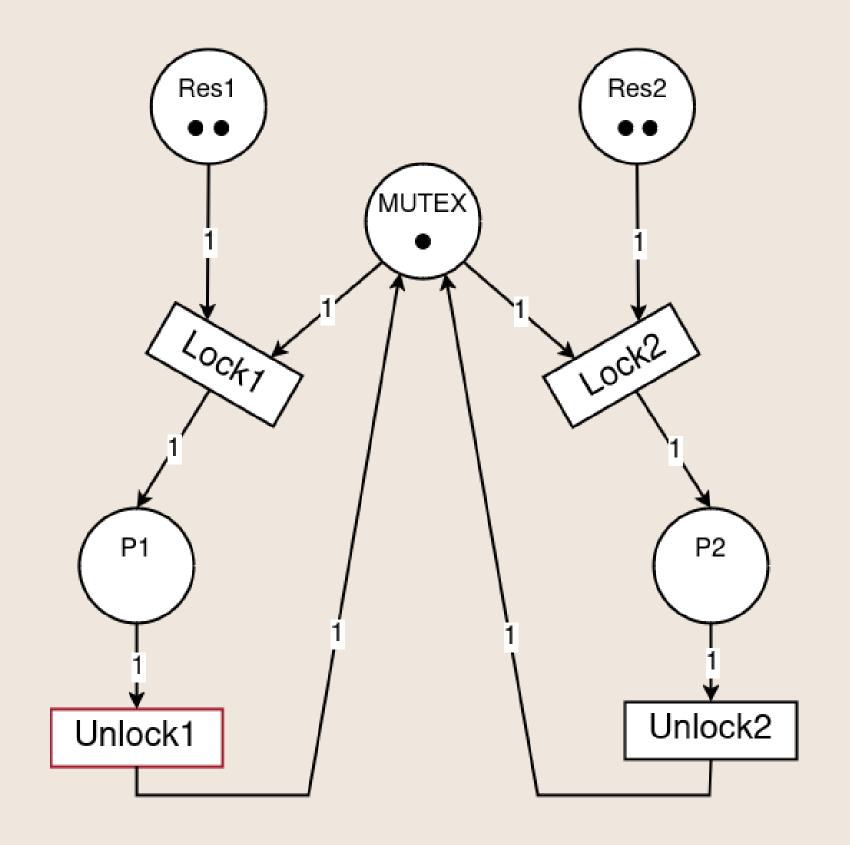
Putem modela excluziunea mutuala folosind un nod de tipul place care are exact un jeton (cheia lacatului).

Odata ce o tranzitie foloseste acest token, toate celelalte tranzitii care depind de cheie sunt blocate.



Putem modela excluziunea mutuala folosind un nod de tipul place care are exact un jeton (cheia lacatului).

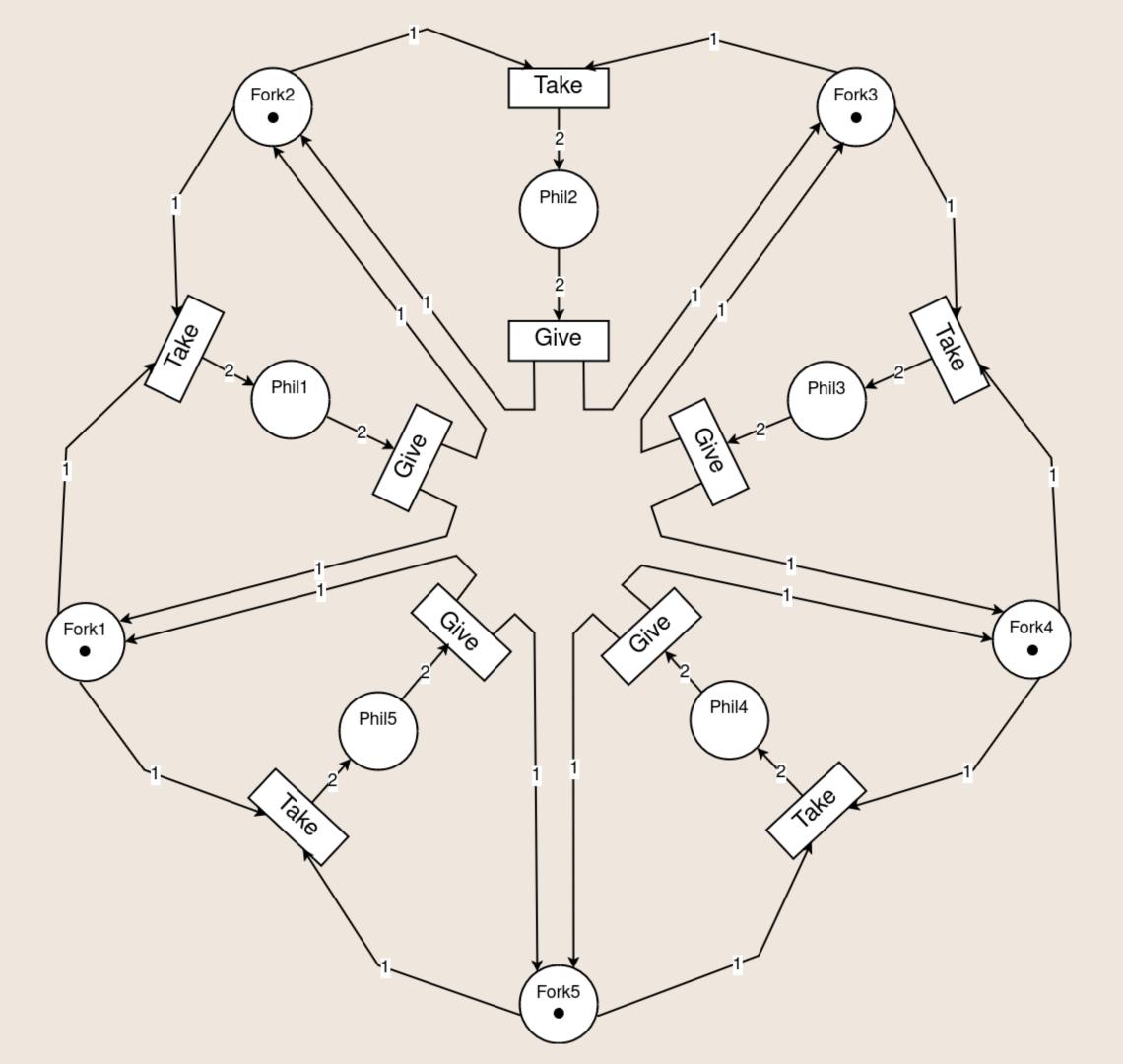
Odata ce o tranzitie foloseste acest token, toate celelalte tranzitii care depind de cheie sunt blocate.



Problema Filozofilor

Problema filozofilor este rezolvata de faptul ca o tranzitie se declaseanza atomic.

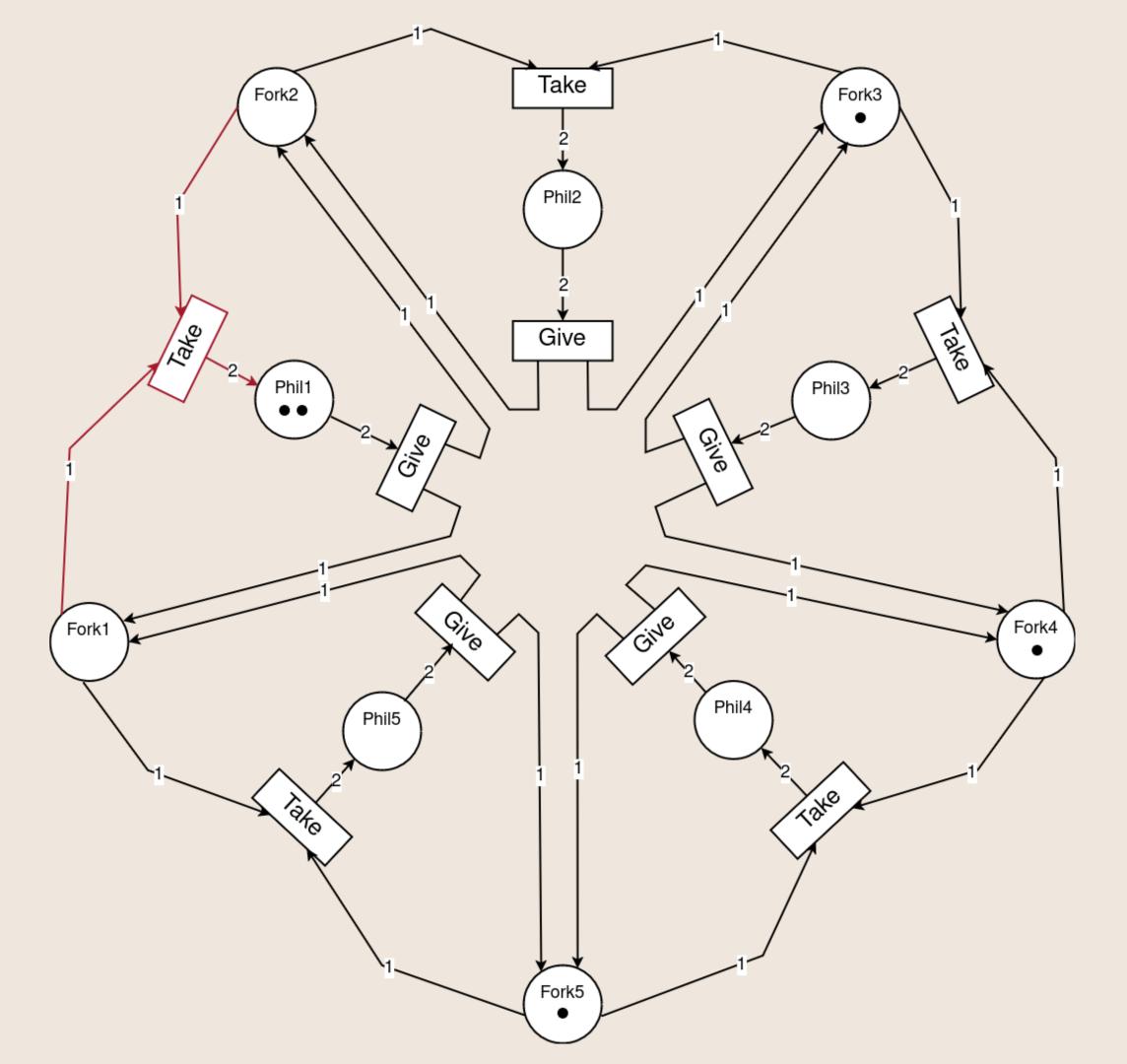
De exemplu, atunci cand primul filosof decide sa ia cele 2 resurse, ia ambele jetoane din nodurile place Forkl si Fork2. Daca un jeton lipseste, atunci tranzitia nu se declanseaza, iar filosoful asteapta.



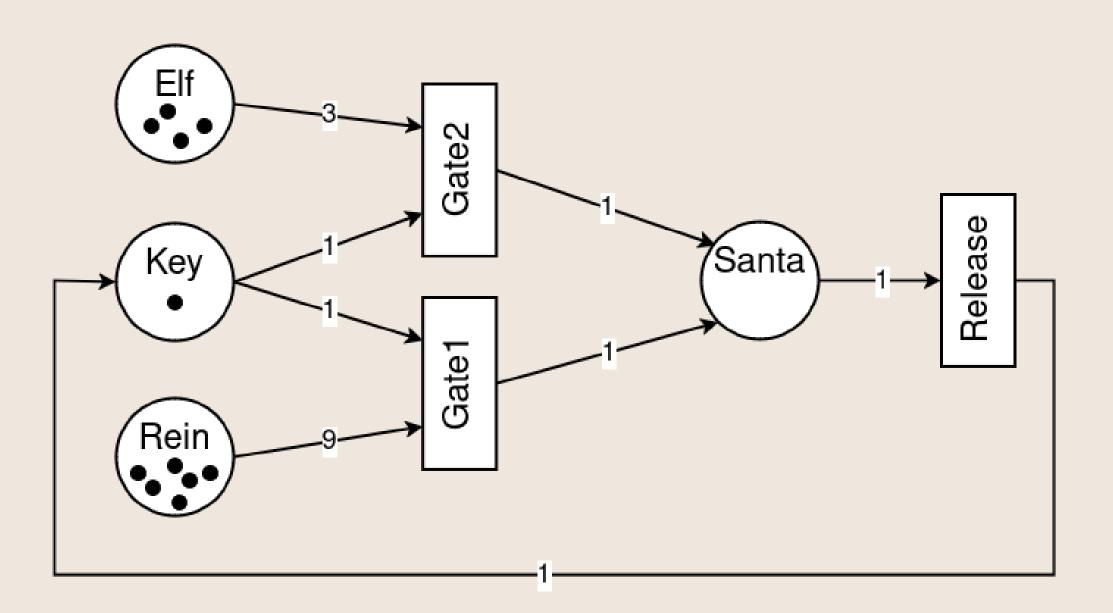
Problema Filozofilor

Problema filozofilor este rezolvata de faptul ca o tranzitie se declaseanza atomic.

De exemplu, atunci cand primul filosof decide sa ia cele 2 resurse, ia ambele jetoane din nodurile place Forkl si Fork2. Daca un jeton lipseste, atunci tranzitia nu se declanseaza, iar filosoful asteapta.



Santa Claus



- Problema Santa Claus se modeleaza foarte natural cu un Petri Net.
- Considerand fiecare poarta ca o tranzitie, putem folosi un nod Key cu un singur jeton pentru excluziunea mutuala (la Santa intra un singur grup odata).
- Capacitatea fiecarui grup este data de muchia exterioara a nodului Elf/Rein.

Continuarea proiectului

JETOANE COLORATE

In articolul original se vorbeste pe scurt despre conceptul de 'colored petri nets'. Adaugarea unei culori pentru fiecare nod, insereaza logica conditionala in structura de date (if).

JETOANE CU INFORMATIE

Fiecare nod place retine doar numarul de jetoane, dar fiecare jeton ar putea contine informatii extra, de exemplu: un numar, un id, o resursa.

TRANZITII CA FUNCTII

Momentan, o tranzitie doar ia jetoane din unele locuri si le pune in altele. Am putea adauga functionalitate: procesarea informatiilor, afisarea unor mesaje (print) sau modelarea timpului (sleep).