

Memo sur les réseaux de Petri

Un RdP est un modèle mathématique ayant pour but de modéliser et vérifier le comportement de systèmes à événements discrets*.

Principe de modélisation:

composant actif= transporter objet ou information

composant passif=contient/stocke des objets ou de l'information

- composants actifs et passifs d'un système doivent être bien différenciés,

arc= ne relie jamais deux sommets de la même famille, représente une relation abstraite entre composants (et n'est pas une composante réelle).

Réseau de places et transitions:

Il s'agit d'un graphe biparti orienté caractérisé par une famille de places (composant passif) et une famille de transitions (composant actif) reliées par des arcs (relation entre composants).

Les places sont représentées par des cercles et les transitions par des barres.

Les arcs relient les places aux transitions et les transitions aux places. Ils sont valués par un poids (résultat d'une fonction de valuation).

Chaque place contient un nombre entier de jetons (appelés aussi marques) pour modéliser la dynamique du système: une place est donc vide ou marquée.

L'ensemble des places et la distribution des marques forment le marquage.

Le réseau donne les règles de fonctionnement du système et le marquage donne l'état du système.

Notation: On retrouve différentes notations dont :

Un RdP est un 5-uplet: $\langle P, T, W^-, W^+, M_0 \rangle$ tq:

P =ensemble fini non vide de places

T = ensemble fini de transitions

W =application d'incidence (Post=+ arrière, Pre=- avant); soit t transition et p une place, $W^-(p,t)$

défini le nombre de jetons que doit contenir p pour que t soit franchissable ($p \rightarrow t$). $W^+(p,t)$ définit le nombre de jetons à ajouter à p une fois t franchie ($t \rightarrow p$).

Chaque application d'incidence est associée une matrice $|P \times T|$ et matrice d'incidence $C = \text{Post} - \text{Pre}$.

M =marquage initial (nb jetons initial)

Un RdP est un 6-uplet: $\langle P, T, F, M_0, W, K \rangle$ tq:

F = ensemble fini d'arcs (flèches)

K = limite de capacité (nombre maximum de jetons que peut contenir une place)

Franchissement d'une transition:

On dit qu'une transition t est franchissable pour un marquage M si chaque place p en entrée contient un nombre de jetons \geq à la valuation de l'arc qui la relie à t :

$M(p) \geq W^-(p,t)$ (voir Notation).

Le franchissement permet d'atteindre un nouveau marquage M' tq:

$M'(p) = M(p) - W^-(p,t) + W^+(p,t) = M(p) + C(p,t)$ on note alors $M[t > M']$

* discret s'utilise comme synonyme de dénombrable,

un système à événements discrets satisfait 2 propriétés: espace d'état discret, transition d'état déclenchée par un événement

Modélisations de comportements:

Modélisation d'un feu routier:

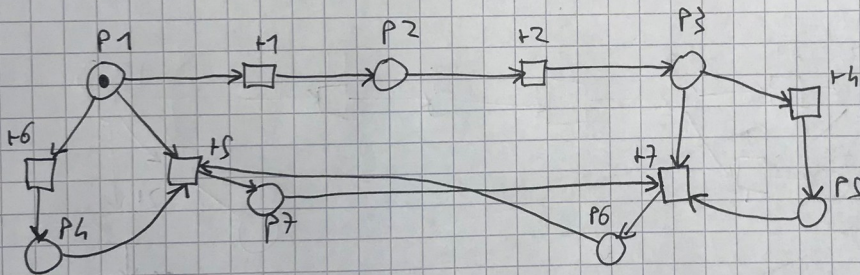
- Les Voitures passent lorsque le feu est Vert ou Orange
Les Piétons ne passent pas _____ et les voitures passent
- Les Voitures ne passent pas lorsque le feu est rouge
Les piétons passent _____ et les voitures sont à l'arrêt

Liste des conditions:

- p1: Le feu est Vert
- p2: _____ Orange
- p3: _____ Rouge
- p4: Les Voitures passent
- p5: Les Voitures sont à l'arrêt
- p6: Les piétons passent
- p7: Les piétons sont à l'arrêt

Liste des actions:

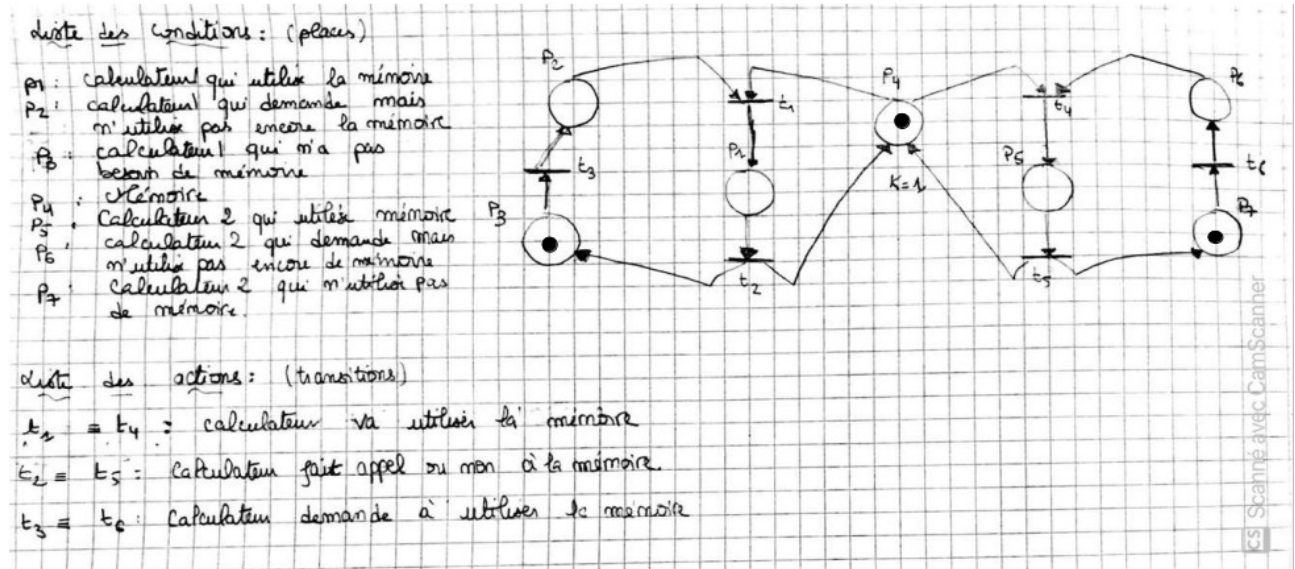
- t1: Le feu passe du Vert à l'Orange
- t2: _____ Orange au Rouge
- t3: _____ Rouge au Vert
- t4: Les Voitures s'arrêtent
- t5: Les piétons s'arrêtent
- t6: Les Voitures démarrent
- t7: Les piétons commencent à marcher



Deux calculateurs utilisent une mémoire commune. On suppose que chaque calculateur peut avoir trois états :

- il n'a pas besoin de la mémoire
- il la demande mais ne l'utilise pas encore
- il l'utilise

Modéliser le fonctionnement de ce système par un RdP:



Mémo sur les tables de hachage distribuées(DHT):

-La Table de Hachage Distribuée (ou DHT pour Distributed Hash Table) est un système de stockage de données réparti et persistant.

-Soit un système réparti, c'est à dire composé de nœuds avec chacun un identifiant associé, et une donnée à stocker. Une fonction de hachage associe une clé à la donnée, la clé indique sur quel nœud la donnée doit être stockée, chaque nœud s'occupant d'une partie des données en fonction de leur clé.

-Lorsqu'un utilisateur recherche une donnée l'algorithme associe une clé à la donnée recherchée, il va demandé à un nœud dans son «entourage» s'il possède la donnée associée à cette clé, si oui la donnée est transmise à l'utilisateur, si non, le nœud sollicité par l'utilisateur va lui communiquer un nœud plus susceptible d'avoir l'information demandée d'après la clé de la donnée, alors l'utilisateur redemande l'information à ce nœud. Ce mécanisme se répète jusqu'à ce que le dernier nœud auquel l'utilisateur a été redirigé ne possède pas de nœud dans son entourage étant plus susceptible que lui-même d'avoir l'information, dans ce cas le nœud possède cette information, sinon l'information n'est pas disponible sur le réseau.

-Il va y avoir une redondance de l'information parmi les nœuds car le nœud où l'information est stockée initialement va communiquer cette information à d'autre nœuds s'occupant du même type de données.

-Un des avantages du DHT est sa robustesse face aux pannes des machines qui sont les nœuds du réseaux : si une partie des machines crash, la redondance des données stockées permet de continuer à pouvoir utiliser le réseaux sans problèmes majeurs (sauf si les pannes sont trop importantes pour le réseau, ex: les 3/4 des machines tombent en pannes).

-En plus de la résistance aux crashes, la redondance permet d'avoir une fiabilité des données stockées.Ex : si L Nœuds ont une information I, alors il faut corrompre $L/2+1$ Nœuds pour corrompre I car l'utilisateur prendra la version de l'information qui apparaîtra en majorité (certains réseaux fonctionnent avec une majorité aux 3/4 ou autres pour plus ou moins de fiabilité).

-Un autre avantage du DHT est son extensibilité au niveau du nombre de nœuds, car le nombre maximum de nœuds parcourus pour la recherche d'une donnée évolue de manière logarithmique avec le nombre de nœuds dans le réseau.

-Une table de hachage parfaite est une table de hachage où il n'y a pas de collisions (2 valeurs différentes ont la même clé) lors du stockage des données. S'il y a une collision dans un nœud, on utilise une table de hachage (ou une LinkedList).

-Différent réseaux utilisant les DHT: Pastry, Chord, Kademlia...

Sources:

RdP:

http://www.moreire.eu/IMG/pdf/cours_petri2.pdf

<http://homepages.laas.fr/francois/RdP/rdp.pdf>

<https://www.emse.fr/~augusto/enseignement/icm/gis1/UP2-2-RdP-slides.pdf>

https://fr.wikipedia.org/wiki/R%C3%A9seau_de_Petri

<http://www.specialautom.net/automatisme/cours%20version%20revue%20pour%20le%20site.pdf>

<http://eric.univ-lyon2.fr/~jdarmon/docs/old/exam0.pdf>

http://www.site.uottawa.ca/~nrahmani/CEG4566_H13/notes_cours/Annexe_1.pdf

DHT:

https://fr.wikipedia.org/wiki/Table_de_hachage_distribu%C3%A9e

https://fr.wikipedia.org/wiki/Table_de_hachage

<https://fr.wikipedia.org/wiki/Kademlia>

<https://www.ietf.org/proceedings/65/slides/plenaryt-2.pdf>

<http://cedric.cnam.fr/~sailhanf/enseignements/p2p/dht.pdf>

<https://www.irif.fr/~jch/software/bittorrent/dht-20130214.pdf>