Petri Net

Leo Rutten

Wijzigingen
Herziening $Revision: 1.6 $ $Date: 2008/11/14 09:22:09 $ $Author: lrutten $

Samenvatting

Dit artikel beschrijft de elementaire werking van PetriNet.

Inhoudsopgave

1. Petri Net

1.1. Inleiding

PetriNet is een grafische methode om het gedrag van systemen waarin parallelle gebeurtenissen voorkomen, te beschrijven. Deze techniek heeft als voordeel dat het gedrag van het systeem op een precieze wijze beschreven kan worden en zelfs met software gesimuleerd kan worden. In simulatie verschijnt exact hetzelfde gedrag zoals dat in de werkelijkheid zou voorkomen.

Meestal kan je met een softwarepakket het model van de werkelijkheid tekenen en daarna met een ingebouwde simulator uitgevoerd worden.

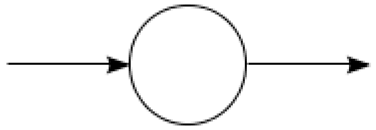
PetriNet kan als techniek gebruikt worden om allerlei systemen te beschrijven: productiesystemen met machines en transportbanden, communicatiesystemen met netwerken en protocols, digitale systemen, processen binnen een besturingssysteem.

1.2. Elementen

1.2.1. Plaatsen

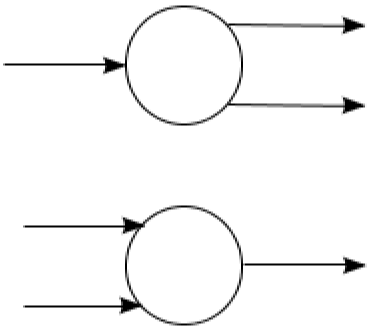
Een plaats wordt grafische weergegeven als een cirkel. Elke plaats heeft één of meerdere ingangen en/of één of meerdere uitgangen. De volgende figuur (Figuur 1) geeft een plaats weer met één ingang en één uitgang.

Figuur 1. Een plaats



De in- en uitgangen worden door pijlen weergegeven. De richting van de pijl geeft aan of je met een in- of een uitgang te maken hebt. Je kan zelf kiezen hoeveel in- en uitgangen een plaats heeft.

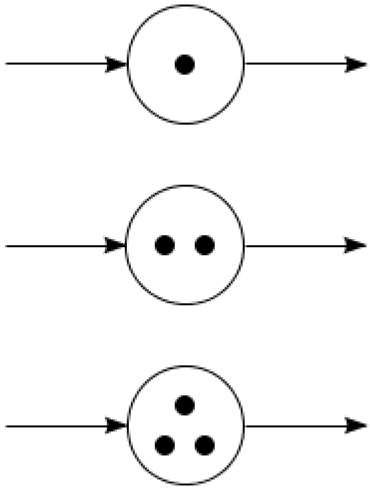
Figuur 2. Meerdere in- en uitgangen bij een plaats



1.2.2. Plaatsen met tokens

Elke plaats heeft geen, een of meerdere tokens. Een token wordt grafisch voorgesteld door een bolletje. Een plaats met een aantal tokens stelt een toestand van een systeem voor. Als je een plaats gebruikt om het aantal vrije parkeerplaatsen in een parkeergarage voor te stellen, dan komt het aantal tokens in de plaats overeen met het aantal vrije plaatsen in de parkeergarage. Je kan zelf kiezen welke betekenis het aantal tokens in een bepaalde plaats krijgt.

Figuur 3. Tokens in een plaats

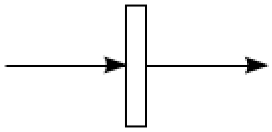


In het voorbeeld Figuur 3 zie je 1, 2 en 3 tokens.

1.2.3. Overgangen

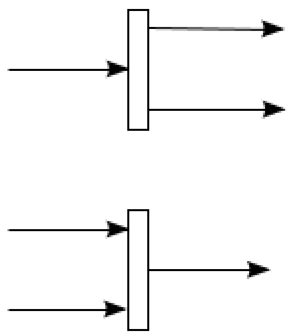
Een overgang (of transitie) wordt grafisch voorgesteld door een rechte lijn. Een overgang in PetriNet stelt een gebeurtenis in de werkelijkheid voor.

Figuur 4. Een overgang



Elke overgang kan één of meerdere ingangen en/of één of meerdere uitgangen hebben. Zoals bij de plaats worden deze in- en uitgangen door pijlen voorgesteld.

Figuur 5. Meerdere in- en uitgangen bij een overgang



1.3. Circuits in Petri Net

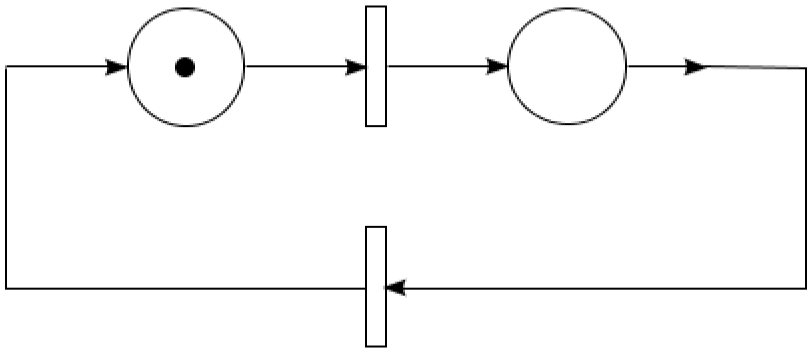
Met behulp van plaatsen en overgangen kan je een compleet circuit bouwen. Je tejent meerdere plaatsen en overgangen. Deze elementen worden verbonden door pijlen. Hierbij gelden een aantal regels:

- Een pijl mag alleen maar getekend worden van plaats naar overgang of omgekeerd. Je mag geen pijl tekenen van een plaats naar een andere plaats. Hetzelfde geldt voor de overgangen.
- Het aantal pijlen naar en van een plaats of overgang is onbeperkt. Denk erook aan dat de pijlen een richting hebben.
- Teken in sommige plaatsen een aantal tokens. Hoeveel en waar hangt af van welke werkelijkheid wilt modelleren.

Een circuit is een model van de werkelijkheid. Dit betekent dat je het gedrag van een deel van de werkelijkheid wil beschrijven in een circuit. Met PetriNet is het daarom ook mogelijk om dit gedrag te simuleren. Het simuleren bestaat erin dat een aantal tokens zich verplaatsen van plaats via overgang. Met andere woorden, een PetriNet circuit is eigenlijk een grote knikkerbaan. De plaatsen en overgangen, verbonden met pijlen, zijn de wegen die de knikkers mogen volgen. Er is wel een verschil tussen knikkers en tokens: een token kan zich vermenigvuldigen of halveren.

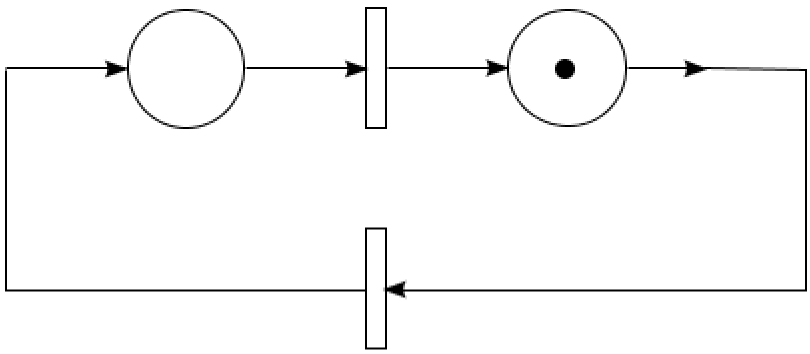
Figuur 6 toont een circuit vóór het vuren van een overgang. De linkerplaats bevat één token. Van de bovenste overgang kan je nu zeggen dat ze gevuurd kan worden. Anders geformuleerd, elke plaats aan de ingang van een overgang heeft ten minste één token. Als deze voorwaarde vervuld is, kan er gevuurd worden. Bij het vuren wordt de plaats aan de ingang met één token verminderd en de plaats aan de uitgang wordt met één token vermeerderd.

Figuur 6. Een circuit vóór het vuren



Figuur 7 toont het circuit na het vuren. Het token bevindt zich nu in de rechterplaats. Dit betekent dat de onderste overgang nu klaar is om gevuurd te worden. Na deze bewerking zijn we weer bij de startpositie aanbeland. Je ziet dat in dit eenvoudig circuit het token continue ronddraait.

Figuur 7. Een circuit na het vuren



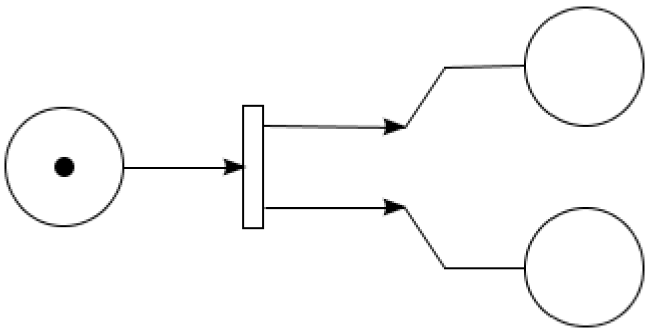
1.4. Meerdere in- en uitgangen

Wanneer er meerdere in- of uitgangen zijn, treedt er verdubbeling of halvering op wanneer tokens door een overgang gevuurd worden.

1.4.1. Een overgang met twee uitgangen

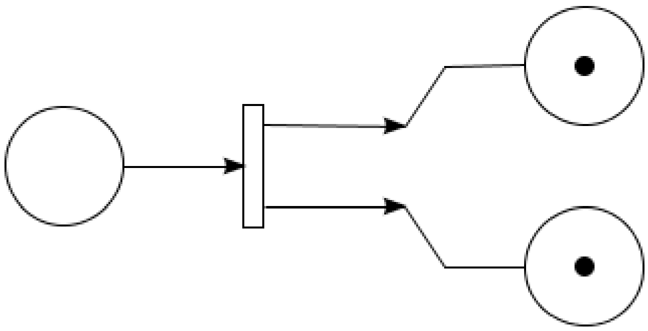
Vóór het vuren ziet de situatie eruit zoals in Figuur 8.

Figuur 8. Twee uitgangen vóór het vuren



Na het vuren is de situatie zoals in Figuur 8.

Figuur 9. Twee uitgangen na het vuren

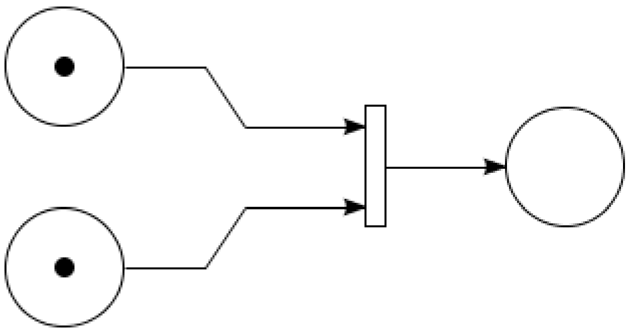


Je ziet dat het aantal tokens verdubbelt.

1.4.2. Een overgang met twee ingangen

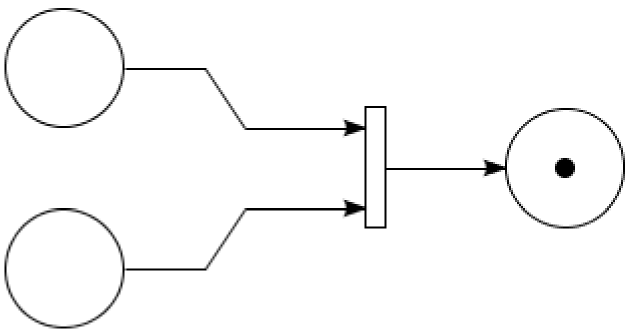
Vóór het vuren ziet de situatie eruit zoals in Figuur 10. Beide plaatsen links hebben elk één token; alleen hierdoor kan de overgang gevuurd worden. Een overgang met twee ingangen werkt als een en-poort.

Figuur 10. Twee ingangen vóór het vuren



Na het vuren ziet de situatie eruit zoals in Figuur 11.

Figuur 11. Twee ingangen na het vuren

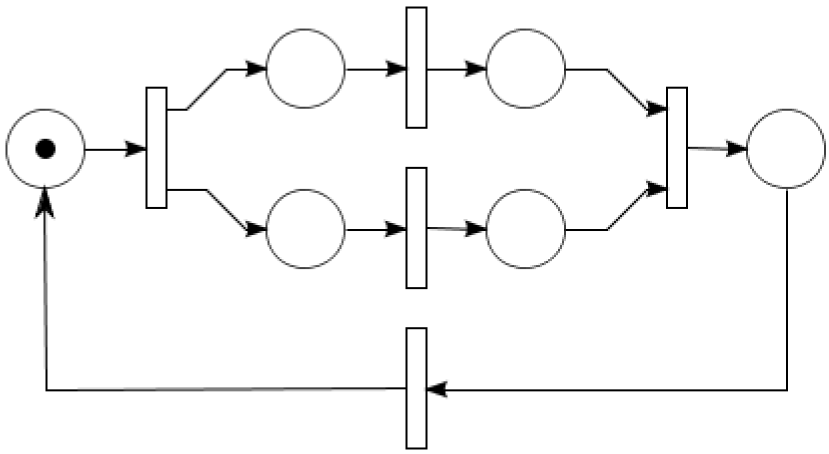


Je ziet dat het aantal tokens halveert. Uit de twee plaatsen links is er en token weggenomen. Merk op dat bij een overgang met meerdere ingangen de voorwaarde tot vuren strenger is: aan elke ingang moet er een token klaar staan om te vuren.

1.5. Een circuit met parellelle wegen

In Figuur 12 komen plaatsen voor met meerdere ingangen als met meerdere uitgangen. Dit circuit modelleert een programma dat met `fork()` een kindproces creëert en dan wacht op het beëindigen van het kindproces.

Figuur 12. Een circuit met parellelle wegen

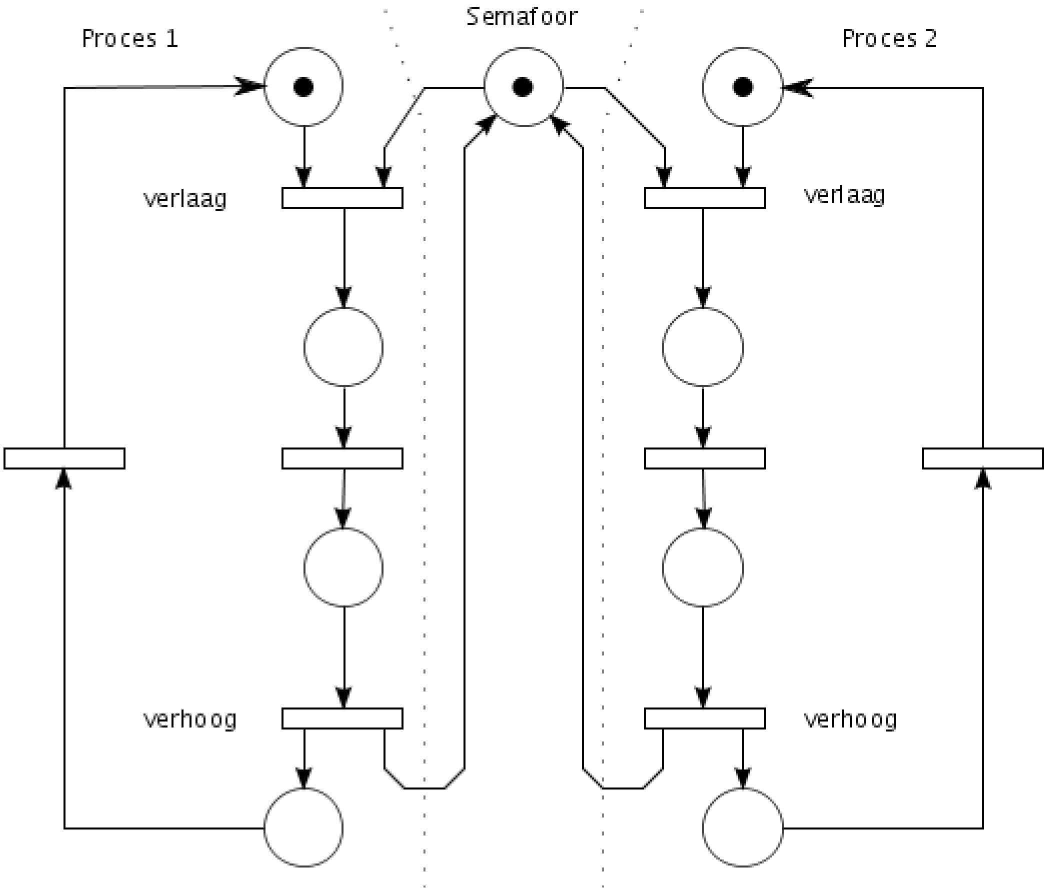


1.6. De semafoor in Petri Net

In Figuur 13 zie je het model van twee processen die een semafoor gebruiken voor wederzijdse uitsluiting. In het midden, tussen de twee verikale stippellijnen, zie je de semafoor voorgesteld met een plaats met één token. Dit komt overeen met een semafoorteller gelijk aan 1.

Het circuit bevindt zich in de startpositie. Zowel de overgang van het linkerproces als van het rechterproces zijn bereid om gevuurd te worden. Het is wel niet mogelijk dat het vuren van beide overgangen doorgaat: het is immers nodig om zowel bij het vuren van de rechter- als linkerovergang een token weg te nemen bij de semafoor. En dit kan natuurlijk niet omdat er slechts één token in de semafoor aanwezig is. Van zodra een van beide overgangen gevuurd is, is de semafoor zonder token en daardoor is de andere overgang geblokkeerd.

Figuur 13. Semafoor in PetriNet



Bij het bereiken van de verhoog overgang komt het token terug bij de semafoor en is de cyclus rond. Bij het simuleren van dit circuit zie je dat de keuze van welke overgang gevuurd wordt, willekeurig is.

1.7. De filosofen aan tafel

Het filosofen aan tafel probleem kan je ook modelleren met PetriNet. Maak voor elke vork een semafoor en voor elke filosoof een proces.

Figuur 14. Filosofen aan tafel