

Capítulo

1

Utilizando Redes de Petri para Modelagem de Sistemas Computacionais: Análise, Propriedades e Aplicações

Jean Carlos Teixeira de Araujo e Paulo Romero Martins Maciel

Resumo

Redes de Petri é um formalismo muito útil para tratar sistemas que exibam características de concorrência e distribuição. Além da simplicidade de seus conceitos, este formalismo apresenta uma notação gráfica e matemática que possibilita a apresentação do comportamento dos sistemas modelados, permitindo o uso de seus conceitos de forma intuitiva, podendo ser aplicada nas mais diversas áreas.

Abstract

Petri nets formalism is very useful for treating systems that exhibit characteristics of competition and distribution. Besides the simplicity of its concepts, this formalism provides a graphical notation and mathematics that enables the presentation of the behavior of systems modeled, allowing the use of its concepts intuitively and can be applied in several areas.

1.1. Introdução

Sistemas concorrentes e distribuídos são alguns dos mais desafiantes sistemas que são desenvolvidos na prática. É importante que atividades de depuração e testes de partes dos sistemas sejam levados em consideração antes mesmo da implantação. Uma forma de se ter isso é através da construção de modelos executáveis. A simulação destes modelos pode revelar erros ou inconsistências em fases iniciais do processo de desenvolvimento do sistema, pois [Marranghello, 2005] permite inferir os sistemas modelados sem a necessidade de construí-los, ou seja, quando eles ainda são apenas propostas de novos sistemas; sem a necessidade de perturbá-los, quando seu custo operacional é alto ou os seus requisitos de segurança impedem ou desaconselham

experimentos; e sem o risco de danificá-los, quando se tem que definir os limites do sistema.

Para [Marranghello, 2005] um modelo é uma descrição de um sistema por meio de equações e relações matemáticas bem como de representações gráficas, baseadas em leis ou princípios físicos que governam o sistema, ou seja, o modelo é uma abstração do sistema original. Os modelos podem ser analíticos ou discretos. Os modelos do tipo analítico são formados por uma série de equações matemáticas usadas para prever o comportamento do sistema pela atribuição de valores aos parâmetros do modelo e a solução repetida das equações para cada conjunto de valores. Os modelos do tipo discreto são representados por uma estrutura matemática ou lógica que pode ser exercitada utilizando-se um computador, para simular o comportamento do sistema.

Redes de Petri é um formalismo muito útil para tratar sistemas que exibam características de concorrência e distribuição. Além da simplicidade de seus conceitos, este formalismo apresenta uma notação gráfica e matemática que possibilita a apresentação do comportamento dos sistemas modelados, permitindo o uso de seus conceitos de forma intuitiva. Segundo [Peterson, 1977], o principal uso das redes de Petri é a modelagem de sistemas de eventos em que é possível que alguns eventos ocorrem simultaneamente, mas há restrições sobre o curso, a precedência, ou a frequência dessas ocorrências.

As redes de Petri [Petri, 1966] foram originalmente desenvolvidas e usadas para o estudo das propriedades qualitativas de sistemas, exibindo características de concorrência e sincronização. A PN original não tinha qualquer noção de tempo. Seu objetivo era desenvolver um modelo em que as máquinas de estado fossem capazes de se comunicar. Em relação às aplicações, [Marranghello, 2005] a PN atingiu áreas como a modelagem de componentes de *hardware*, controle de processos, linguagens de programação, sistemas distribuídos e protocolos de comunicação. A característica principal alcançada por este modelo foi a possibilidade de representar a concorrência.

Tal modelo recebeu o nome de redes de Petri em homenagem ao próprio autor, Carl Adam Petri, e hoje encontramos variações do modelo original sendo usados em diversas áreas da ciência para ajudar no estudo do comportamento e desempenho de diferentes sistemas [Penha *et al*, 2004].

1.2. Histórico

Redes de Petri (ou Petri Nets) foram inventadas em agosto de 1939 por Carl Adam Petri, quando ele tinha 13 anos, para descrever processos químicos. Vinte e três anos depois, em 1962, ele documentou o trabalho como parte de sua dissertação de doutorado, intitulada *Kommunikation mit Automaten* (Comunicação com Autômatos), apresentada à Faculdade de Matemática e Física da Universidade Técnica de Darmstadt na então Alemanha Ocidental [Petri, 1966] [Maciel *et al*, 1996]. A dissertação foi preparada enquanto C. A. Petri trabalhou como cientista na Universidade de Bonn, Alemanha, de 1959 a 1962. Este modelo foi baseado nos conceitos de operação assíncrona e concorrente com as peças de um sistema e a percepção de que as relações entre as partes pode ser representada por um gráfico, ou rede [Peterson, 1977].



Figura 1: Foto de Carl Adam Petri

O trabalho de Petri atraiu a atenção de A. W. Holt, o que mais tarde o levou ao *Information System Theory of Project of Applied Data Research, Inc.*, nos Estados Unidos [Murata, 1989] [Peterson, 1977], que em conjunto com outros pesquisadores, desenvolveu muito da teoria, notação e representação das redes de Petri [Maciel, *et al*, 1996].

A década de setenta marcou o desenvolvimento da teoria de redes de Petri e a expansão de seu campo de aplicação. No início daquela década, [Murata, 1989] de 1970 a 1975, o grupo de estrutura da computação do MIT foi o mais ativo na condução da pesquisa sobre redes de Petri, e produziu muitos relatórios e teses sobre *Petri Nets*. Em julho de 1975, houve uma conferência sobre “Redes de Petri e Métodos Relacionados” no MIT, mas nenhum procedimento foi publicado nesse evento. Vale ressaltar que na conferência realizada pelo MIT em 1975, foi a primeira vez em que o nome “Rede de Petri” foi oficialmente utilizado para se referir ao formalismo proposto por Carl Adam Petri treze anos antes, na Alemanha. Em 1979, [Maciel *et al*, 1996] [Murata, 1989] cerca de 135 pesquisadores de vários países europeus reuniram-se por duas semanas em Hamburgo, Alemanha Ocidental, para um curso avançado sobre “Teoria Geral das Redes de Processos e Sistemas”.

Logo no início da década seguinte, em 1980, ocorreu o primeiro Workshop Europeu em Aplicações e Teoria das Redes de Petri. Desde então, uma série de workshops foram realizados todos os anos em diferentes locais da Europa. Contudo, o segundo curso avançado foi realizado na cidade de Bad Honnef, no ano de 1986, também na Alemanha, dando seguimento ao primeiro curso realizado em 1979. No final da década de oitenta, em 1989, foi publicado um artigo sobre redes de Petri pelo Prof. Tadao Murata [Murata, 1989]. Embora este artigo trate primordialmente das redes de baixo nível, ele é um dos artigos mais referenciados sobre o assunto ainda nos dias de hoje.

Somente dez anos depois, em 1996, [World, 2010] veio a acontecer o terceiro curso avançado em redes de Petri, desta vez na cidade de Dagstuhl, como sempre, na

Alemanha, contando com a participação de aproximadamente 80 pessoas, entre os quais 17 palestras foram realizadas, sendo que Carl Adam Petri também visitou o curso por alguns dias.

Em 2003, a cidade de Eichstätt veio a sediar o quarto curso avançado, onde mais de uma centena de pessoas participaram, entre quais 24 palestras foram realizadas. Coincidentemente, o último evento que segue essa cronologia ocorreu entre os dias 13 e 24 de setembro de 2010, na cidade de Rostock, sendo que infelizmente não pôde contar com a presença de C.A. Petri em razão do seu falecimento em julho do mesmo ano, aos 83 anos.

Seguiram-se diversos outros trabalhos propondo alterações ao modelo original, tais como rede com arco inibidor, e redes temporizadas determinísticas e estocásticas. Hoje em dia, redes de Petri é considerada uma técnica para especificação de sistemas concorrentes consolidada. Grupos de pesquisa em todo o mundo têm Redes de Petri como tema, desenvolvendo estudos sobre seus aspectos teóricos e suas aplicações [Maciel *et al*, 1996].

1.3. Classes de Redes de Petri

Em algumas fases do processo de desenvolvimento de um sistema, necessitamos representar o sistema em evolução com maior ou menor detalhamento, de maneira compacta. Em outra fase do desenvolvimento pode ser necessário o esclarecimento de determinados aspectos, que em fases anteriores não se faziam necessários [Maciel *et al*, 1996]. Tendo em vista esses aspectos, a literatura fornece diferentes classificações para as redes de Petri:

1.3.1. Redes de Petri

As redes ordinárias se caracterizam pelo tipo de suas marcas, ou seja, suas marcas são do tipo inteiro e não negativo, enquanto as de alto nível possuem marcas de tipos particulares. As redes ordinárias se subdividem em:

-Redes de Petri Condição-Evento: é a rede mais elementar dentre todas. Essa rede só permite no máximo um *token* em cada lugar, e todos os arcos possuem valor unitário.

1.3.2. Redes de Petri *Place-Transition*

O modelo mais proeminente de uma rede de Petri é a *Place/Transition net*. Essa é o tipo de rede que permite o acúmulo de marcas no mesmo lugar, assim como valores não unitários para os arcos. Contudo, alguns autores fazem distinção entre as redes em que os arcos têm valores diferentes de um e as redes com arcos de peso igual a um. É subdividida em: RdPs Não-Ponderadas e RdPs Ponderadas. Maiores detalhes, vide seção 2.4.

1.3.1. Redes de Petri ou de Alto Nível

São caracterizadas pelos tipos de suas marcas, que não são mais elementos do tipo inteiro positivo. São aquelas cujas marcas incorporam alguma semântica, viabilizando sua diferenciação. Esta semântica pode ir desde a atribuição de valores ou cores às marcas, até a adoção de noções de tipos de dados abstratos. Redes não-ordinárias não aumentam o poder de representação de um modelo. Entretanto, elas permitem uma maior clareza e um maior (ou menor) nível de abstração ao modelo. Segue abaixo algumas das principais subdivisões desse tipo de rede:

-Redes de Petri Temporizadas: existem transições que são temporizadas, e possivelmente transições não temporizadas, nesse caso, transições não temporizadas são prioritárias em relação as temporizadas. Desta forma, a propriedade de tempo também pode ser modelada, não somente a estrutura. A principal vantagem de uma rede de Petri temporizada é a relação de tempo, que antes não havia. A relação de tempo altera os estados/marcações alcançáveis. Por este motivo, é muito importante saber utilizar o tempo na rede para que seu projeto ou modelo não entre por exemplo em um *deadlock* [Penha *et al*, 2004] e [Maciel *et al*, 1996].

-Redes de Petri Coloridas: O seu principal objetivo é a redução do tamanho do modelo, permitindo que marcas individualizadas (cores) representem diferentes processos ou recursos em uma mesma sub-rede. Inicialmente, os *tokens* das CPNs (*Colored Petri Nets*) eram representados por cores ou mesmo por padrões que possibilitam a distinção dos *tokens*. Em trabalhos mais recentes, os *tokens* são representados por estruturas de dados complexas não relacionando cores aos *tokens*, a não ser pelo fato de que estes são distinguíveis [Maciel *et al*, 1996]. Desta maneira, os tokens podem conter informações. Além disso, cada *place* armazena *tokens* de um certo tipo definido e os arcos realizam operações sobre os *tokens*.

-Redes de Petri Hierárquicas: têm por objetivo permitir a representação de modelos de sistemas complexos de forma mais compreensível pelo modelador. Via de regra as extensões hierárquicas não aumentam o poder de representação das RdP, apenas tornam mais fácil para o usuário a compreensão das redes resultantes.

1.4. Redes de Petri Place/Transitions

As redes *Place/Transition* foi o tipo de rede mais utilizado na década de setenta, por permitir uma compactação razoável dos modelos. A partir desse pressuposto e considerando a grande quantidade de redes existentes, passaremos a abordar de agora em diante as características e peculiaridades sobre esse tipo de rede.

1.4.1. Conceitos básicos

O autor [Maciel *et al*, 1996] define rede de Petri como uma técnica de especificação de sistemas que possibilita uma representação matemática e possui mecanismos de análise poderosos, que permitem a verificação de propriedades e verificação da *corretude* do sistema especificado. Já para [Barros, 1996] as redes de Petri é definida como sendo um modelo matemático que possibilita uma representação gráfica ampla proporcionando uniformidade na modelagem, análise e simulação de sistemas a eventos discretos. Além disso, permite uma visualização simultânea da sua estrutura e comportamento. Em

[Bobbio, 1990] encontramos mais uma definição sobre redes de Petri, ele as define como sendo “uma ferramenta gráfica para a descrição formal do fluxo de atividades em sistemas complexos”.

Os autores [Reisig *et al*, 1998] afirmam que uma das principais atrações de redes de Petri é a maneira em que os aspectos básicos de sistemas concorrentes são identificados conceitualmente e matematicamente. O fácil conceito de modelagem torna o modelo de rede de Petri a escolha em muitas aplicações. A forma natural em que uma PN permite capturar formalmente muitas das noções básicas e problemas de sistemas concorrentes contribuiu grandemente para o desenvolvimento de uma rica teoria de sistemas concorrentes baseados em redes de Petri.

Enfim, uma rede de Petri possui um conjunto de elementos capaz de descrever as diversas partes de um sistema com situações típicas, tais como: concorrência, controle, conflitos, sincronização e compartilhamento. Com relação a outras técnicas mais populares de representação gráfica do sistema (como diagramas de blocos lógicos ou árvores), [Bobbio, 1990] PNs são particularmente adequadas para representar de uma forma natural interações lógicas entre as partes ou atividades em um sistema.

1.4.2. Definições formais

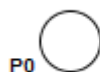
Para definições e notação, geralmente nos referimos para [Peterson, 1981]. Uma rede de Petri (PN) é um grafo dirigido bipartido, consistindo de dois tipos de nós, lugar e transição, e arcos que permitem conectar somente nós de classes diferentes.

Do ponto de vista formal, uma PN marcada é uma quintupla (P, T, I, O, M) , onde [Peterson, 1981]:

- $P = \{p_1, p_2, \dots, p_n\}$ é o conjunto dos n lugares da PN (traçados como círculos na representação gráfica);
- $T = \{t_1, t_2, \dots, t_m\}$ é o conjunto das m transições da PN (traçados como barras na representação gráfica);
- I é a relação de entrada de transição e é representada por meio de arcos dirigidos de lugares para transições;
- O é a relação de saída de transição e é representada por meio de arcos dirigidos de transições para lugares;
- $M = \{m_1, m_2, \dots, m_n\}$ é a marcação. A entrada genérica m_i é o número de marcas (traçados como discos) no lugar p_i na marcação M .

A representação gráfica das redes de Petri tem se mostrado muito útil, pois permite a visualização dos processos e a comunicação entre eles. Uma PN é composta pelos seguintes elementos gráficos:

- Lugares ou *Places*: representam tipicamente uma condição, uma atividade ou um recurso dentro do sistema modelado;



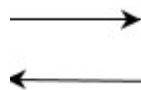
- Transições: representam eventos que, ocorrendo no sistema, podem causar mudança na sua condição;



- Fichas, Marcas ou *Tokens*: representam o estado de um sistema;



- Arcos: indicam os lugares de entrada ou saída para as transições. Eles conectam lugares a transições e transições a lugares.

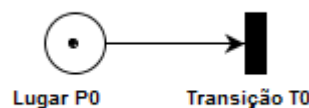


Estes elementos podem ser representados graficamente, formando uma rede de Petri, como podemos observar na figura abaixo:



Figura 2: Componentes básicos de uma rede de Petri

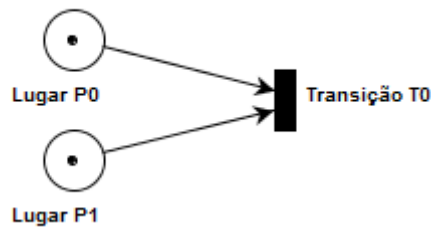
- Arcos de entrada são arcos traçados de lugares para transições, representando as condições necessárias que precisam ser satisfeitas para o evento ser ativado.



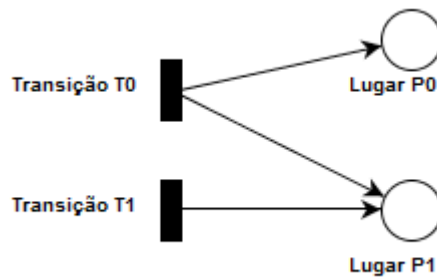
- Arcos de saída são arcos traçados de transições para lugares, representando as condições resultantes da ocorrência do evento anterior.



- *Places* de entrada de uma transição são o conjunto de lugares que estão conectados à transição através de arcos de entradas.



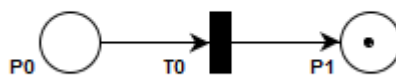
- *Places* de saída de uma transição são o conjunto de lugares para o qual existem arcos vindos da transição.



- Quando os *places* de entrada de uma transição têm o número de marcas necessárias, a transição está habilitada.



- Uma transição habilitada pode disparar (o evento ocorre), removendo um *token* de cada lugar de entrada e depositando um *token* em cada um de seus lugares de saída.



Os vértices de um grafo associado a uma rede de Petri podem ser interligados por múltiplos arcos, ou seja, um lugar pode ser conectado a uma transição através de diversos arcos ou vice-versa. Por conveniência, podemos substituir os múltiplos arcos por um único arco valorado, onde o numeral associado ao arco correspondente ao número de arcos que interligam os vértices [Maciel *et al*, 1996], conforme podemos observar na figura abaixo:

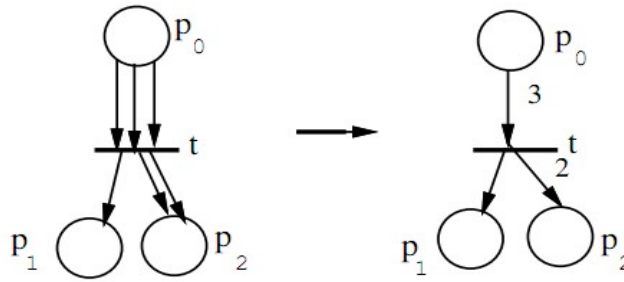


Figura 3: Arcos valorados

Vale ressaltar que não existe uma sequência única para a construção de um modelo em redes de Petri, e que um mesmo problema pode ser modelado de maneiras diferentes, variando de acordo com o ponto de vista e o conhecimento sobre a problemática de quem modelou.

1.4.3. Regras de habilitação e disparo

Para [Murata, 1989] o comportamento de muitos sistemas podem ser descritos em termos de estados do sistema e suas alterações. A fim de simular o comportamento dinâmico de um sistema, um estado ou a marcação de uma rede de Petri é alterado de acordo com a seguinte regra de transição (disparo):

- Uma transição t é dita habilitada se cada lugar de entrada p para t é marcado com pelo menos $w(p, t)$ tokens, onde $w(p, t)$ é o peso do arco do p para t ;
- Uma transição habilitada pode ou não disparar (dependendo se o evento realmente ocorre);
- Um disparo de uma transição habilitada t remove $w(p, t)$ tokens de cada lugar de entrada p de t , e adiciona $w(t, p)$ tokens para cada lugar de saída p de t , onde $w(t, p)$ é o peso do arco de t para p .

Desta maneira, entende-se que uma rede de Petri é executada através do disparo da transição. Uma transição pode disparar se estiver habilitada. Uma transição é habilitada se cada um dos lugares que o seu contributo tem pelo menos um token. A transição dispara removendo um ou mais tokens de cada um dos seus lugares de entrada e então os deposita no *place* de saída.

De modo mais formal podemos dizer que a t_k está habilitada numa marcação M se:

$$\text{para todo } p_i \in I(t_k), m_i \geq 1.$$

Disparar uma transição, em geral, ocasionará uma mudança na marcação da rede de Petri, μ , para p uma nova marcação μ' . Nota-se que somente transições habilitadas podem disparar, o número de tokens em cada lugar permanece sempre não negativo quando uma transição é disparada. O disparo de uma transição nunca pode remover tokens que não estão lá: se qualquer um dos lugares de entrada de uma transição não contém tokens, a transição não pode ser disparada [Peterson, 1977].

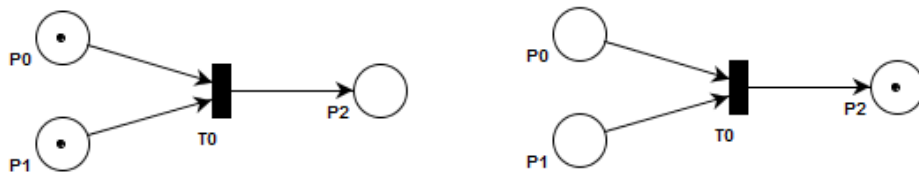


Figura 4: Representação disparo de uma transição (antes e depois)

Se um lugar é uma entrada para a transição, um *token* é removido e, se ele é uma saída, um *token* é adicionado. Nenhuma mudança ocorre se o *place* não é uma entrada ou uma saída.

No entanto, [Murata, 1989] [Maciel *et al*, 1996] existem dois casos particulares de configuração, que têm regras de disparo distintas da apresentada. O primeiro caso corresponde a transições que não possuem lugares de entrada, que são denominadas transições fonte (*source*). A outra situação especial é aquela que apresenta transições



Figura 5: Transição fonte (*source*) antes e após o disparo

sem lugares de saída, chamadas de transição de absorção (*sink*). Uma transição fonte está sempre habilitada e o disparo de uma transição de absorção consome *tokens* nos lugares de entrada, mas não produz nenhum, conforme podemos observar nas figuras abaixo:

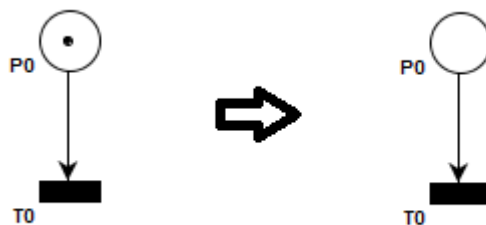


Figura 6: Transição de absorção (*sink*) antes e após o disparo

1.4.4. Árvore de Cobertura

O conjunto de cobertura $C(M_1)$ de uma PN é gerada por meio da árvore de cobertura (*coverability tree*). A primeira marca M_1 é a raiz da árvore de alcançabilidade. A partir da raiz, procure por todas as transições habilitadas; o disparo de uma transição habilitada produz uma nova marcação, que é representado como uma nova folha da árvore, da qual o processo é iterativo.

De acordo com [Peterson, 1977], partir de uma marcação μ , um conjunto de disparos de transição é possível. O resultado do disparo de uma transição de uma marcação μ é uma nova marcação μ' . Dizemos que μ é *imediatamente alcançável* a partir de μ se for possível disparar alguma transição habilitada na marcação μ resultando na marcação μ' . A marcação μ' é alcançável a partir de μ se ele é imediatamente alcançável a partir de μ ou pode ser alcançável de qualquer marcação que é imediatamente alcançável a partir de μ . Define-se o conjunto de alcançabilidade $R(M)$ para uma rede de Petri marcada $M = (P, T, I, O, \mu)$ como o conjunto de todas as marcações, que podem ser alcançadas a partir de μ .

O conjunto de alcançabilidade de uma rede de Petri marcada é o conjunto de todos os estados em que a rede de Petri pode entrar por qualquer possível execução.

Ao identificar adequadamente os nós de fronteira da árvore, a geração da árvore de alcançabilidade envolve um número finito de passos [Peterson, 1981], mesmo que a rede seja ilimitada:

- nós terminais (morto): nós de que as transições não são ativados;
- nós duplicados: nós que já foram gerados na árvore;
- nós infinitamente reproduzíveis. A marcação M é um nó infinitamente reproduzível se $M'' \geq M' (m_i'' \geq m_i', i = 1, 2, \dots, n_p)$ para algum M' já gerados na árvore. Por causa da relação referida, a sequência de transição a partir do qual M'' foi gerada a partir de M' é certamente disparável em M'' . Assim, a sequência $M' \rightarrow M''$ pode ser reproduzida infinitamente, de modo que a contagem de *token* nos lugares para os quais $m_i'' \geq m_i'$ pode aumentar indefinidamente. Nós representamos o número arbitrariamente grande de *tokens* que resulta de nós infinitamente reproduzíveis, definindo um símbolo especial ω com as seguintes propriedades:

$$\omega + a = \omega$$

$$\omega - a = \omega$$

$$a < \omega$$

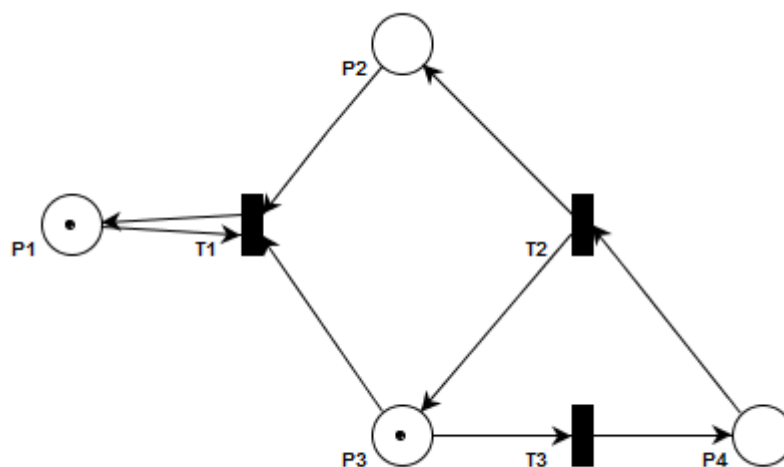


Figura 7: Uma rede de Petri com marcação $(1, 0, 1, 0)$

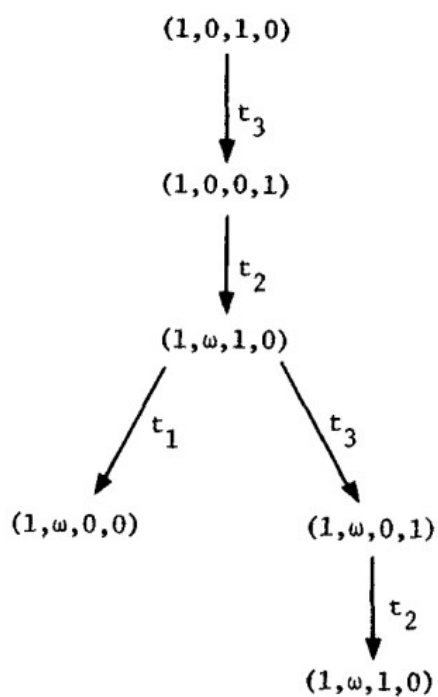


Figura 8: Árvore de Cobertura da rede de Petri anterior

Mais adiante temos uma PN utilizada por [Penha *et al*, 2004] que é um exemplo de conjunto finito de marcações e, posteriormente, sua respectiva árvore de alcançabilidade. O estado inicial é representado pela marcação $M_0 (2, 0, 0, 0)$. Durante a construção da árvore deve ser observado quais são as transições habilitadas para disparo para criação de cada nó (nova marcação).

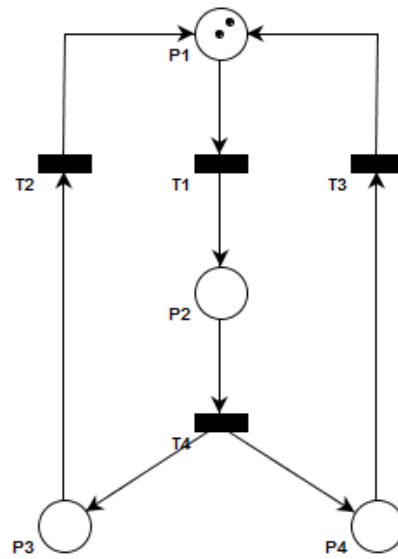


Figura 9: Exemplo de PN para geração da Árvore de Cobertura

Para construirmos a Árvore de Cobertura, devemos observar os seguintes detalhes no exemplo em tela [Penha *et al*, 2004]:

- Após a definição da raiz, devemos observar quais transições estão habilitadas;
- Somente a transição T_1 está habilitada e a marcação resultante é $M_1(0, 1, 0, 0)$;
- O *place* P_2 não recebeu dois *tokens* porque o arco de saída de T_1 tem peso 1;
- A próxima transição habilitada é T_4 resultando em uma marcação $M_2(0, 0, 1, 1)$;
- Neste momento possuímos duas marcações habilitadas, T_2 e T_3 resultando em um nível com dois nós (marcações M_3 e M_4);
- Se dispararmos T_2 a marcação resultante será $M_3(1, 0, 0, 1)$;
- Se dispararmos T_3 a marcação resultante será $M_4(1, 0, 1, 0)$;
- No momento em que rede chega ao estado M_3 ou M_4 passamos a ter dois *places* com *tokens* (M_3 : P_1 e P_4 ou M_4 : P_1 e P_3), porém, a transição T_1 não ficará habilitada enquanto o *place* P_1 não tiver dois *tokens*. Esta situação só vai acontecer quanto houver o disparo de T_3 ou T_4 resultando novamente no estado inicial M_0 .

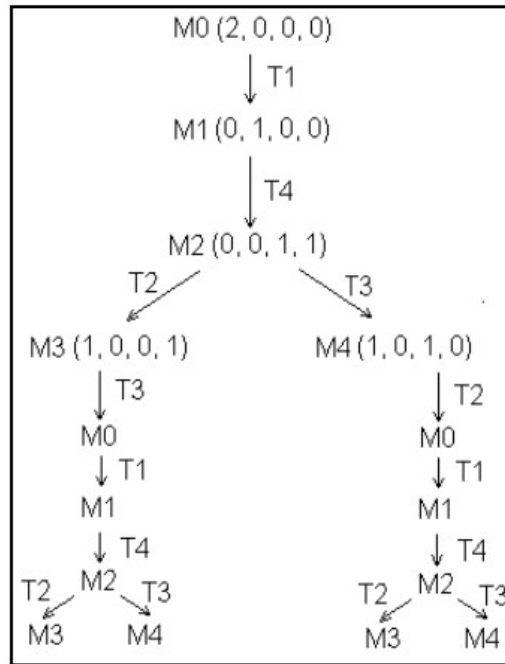


Figura 10: Árvore de Cobertura da PN da figura anterior

Esta PN possui um conjunto de alcançabilidade finito com cinco marcações (M_0, M_1, M_2, M_3, M_4).

1.4.5. Invariantes

Antes de iniciarmos nosso entendimento sobre invariantes de lugar e transição, será realizada um breve comentário sobre análise de redes de Petri com base na abordagem algébrica.

A principal motivação para o uso, em redes de Petri, de uma abordagem algébrica linear é a complexidade comportamental das redes de Petri. Em geral, o conjunto de marcações alcançáveis de uma rede de Petri marcada pode ser explodido, tanto com respeito ao tamanho da rede quanto com respeito ao número inicial de *tokens*. Esses valores podem até chegar ao infinito. Desta forma, na prática, a construção explícita de todas as marcações alcançáveis pode não ser viável. Isso significa que uma análise eficiente das propriedades comportamentais pode não ser baseada em modelos de comportamento, como grafos de marcação. Uma alternativa é explorar as propriedades estruturais da rede, em conjunto a distribuição inicial de *tokens* (marcação inicial da rede). Pode-se observar se uma rede não possui essa propriedade através de invariantes de lugar, uma vez que um invariante de lugar é uma **condição necessária** para a alcançabilidade de uma marcação. Invariantes de lugar e de transição utilizam uma abordagem algébrica linear para tirar conclusões sobre as propriedades comportamentais da rede em análise.

-Invariante de lugar: Caso uma rede possui um ou mais invariantes de lugar, então podemos afirmar que a soma ponderada de *tokens*, desses lugares, quando do

disparo de uma sequência de transições permanece constante. A tarefa de encontrar o invariante de lugar consiste em determinar o valor dos pesos dessa soma ponderada.

Para exemplificar os conceitos de invariante de lugar e de transição, consideraremos a Rede de Petri exibida abaixo:

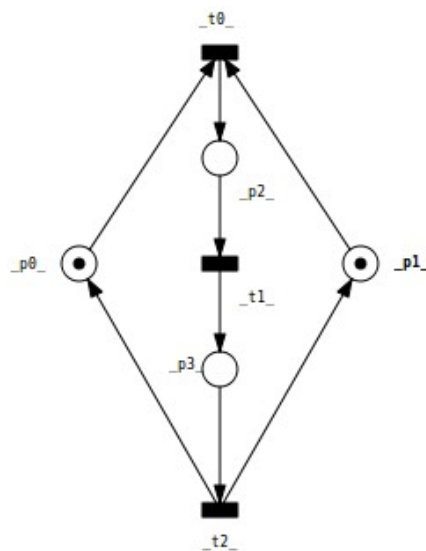


Figura 11. Rede para análise de invariante de lugar e invariante de transição

-Invariante de transição: Trata-se de um vetor característico que representa o número de disparos efetuados por cada transição, em uma sequência de transições, para que se alcance a marcação inicial *original* da rede. Ele possui a dimensão igual ao número de transições na rede. Cada componente do vetor armazena o número de vezes que cada transição pode ser disparada. Pode haver diversos invariantes de transição, sendo que o disparo das transições armazenadas em cada um dos invariantes de transição não altera a marcação da rede.

Considere novamente a rede exibida na anterior. A sequência de disparos t_0, t_1, t_2 (observada nas figuras 12, 13 e 14), faz com que a rede volte a sua marcação inicial. Logo o vetor característico $I_t = (1, 1, 1)$ é um invariante de transição da rede apresentada.

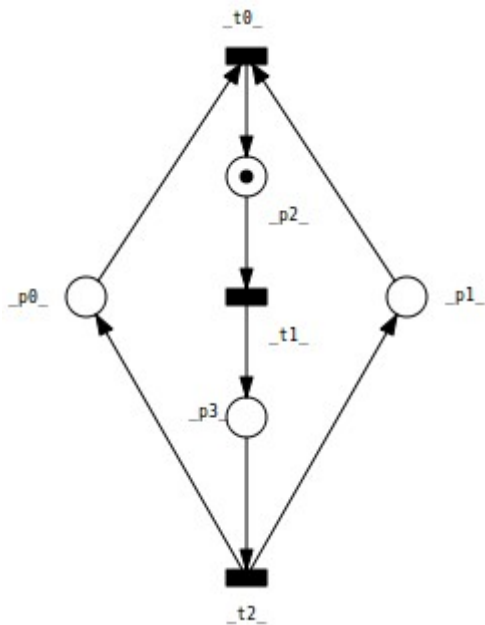


Figura 12: Rede após o disparo da transição t_0

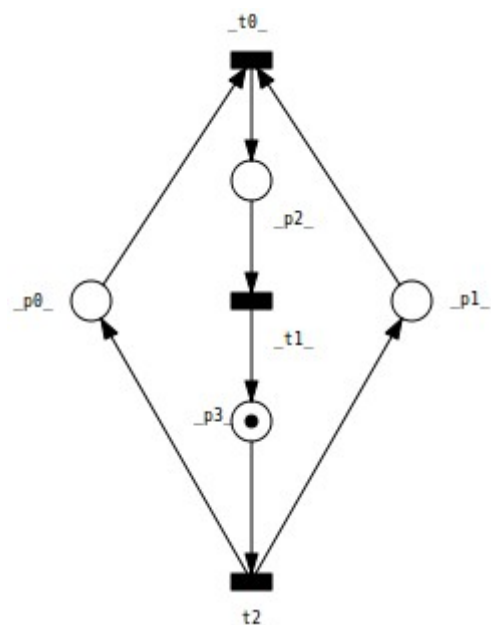


Figura 13: Rede após o disparo da transição t_1

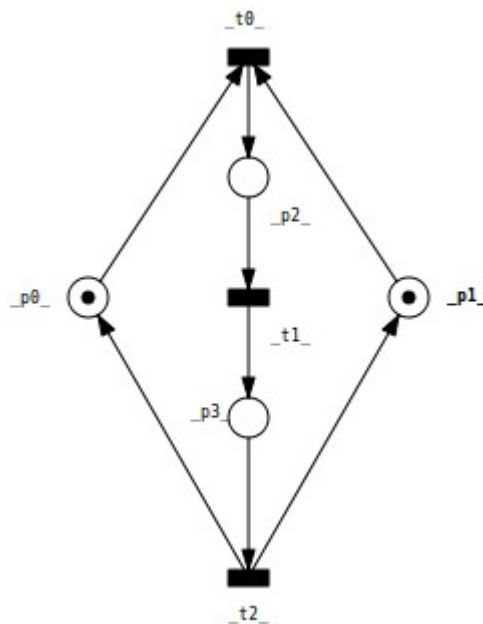


Figura 14. Rede após o disparo da transição t_2

1.4.6. Modelagem

Uma PN utilizada para modelar sistemas reais são muitas vezes referidas como redes de Condição / Eventos. Lugares identificam as condições das partes do sistema (trabalho, ócio, filas, falha), e transições descrevem a passagem de um estado para outro fim (uma tarefa, uma falha, uma reparação, ...). Um evento ocorre (disparo da transição) quando todas as condições forem satisfeitas, (lugares de entrada são marcados). A ocorrência do

evento modifica, no todo ou em parte, o status das condições (marcação). O número de *tokens* em um lugar pode ser usado para identificar o número de recursos daquele *place* [Bobbio, 1990].

1.4.6.1. Processos paralelos

Atividades paralelas ou concorrentes podem ser facilmente modeladas por uma rede de Petri.

Na PN da figura abaixo, as transições T_0 e T_1 são ativadas simultaneamente, o disparo de uma delas não altera o estado da outra. As atividades modeladas por duas transições executam simultaneamente. Na modelagem de confiabilidade, a PN em questão pode representar dois componentes C_1 e C_2 em redundância paralela, neste caso, os lugares p_0 e p_2 representam a condição de trabalho, p_1 e p_3 a condição de falha e T_0 e T_2 caso de falha de C_1 e C_2 , respectivamente [Galdino, 2009].

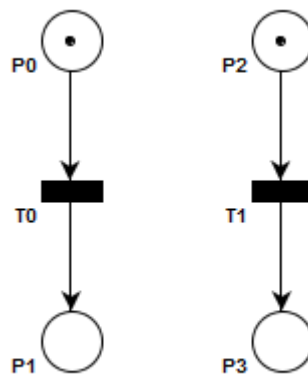


Figura 15: Modelagem de duas atividades paralelas

Em geral, duas transições são ditas *concorrentes* se elas são casualmente independentes, ou seja, uma transição pode disparar antes ou depois ou em paralelo com outro, como no caso de t_2 e t_3 [Murata, 1989].

1.4.6.2. Sincronização

As atividades modeladas por t_0 e t_1 são executadas concorrentemente; entretanto, se elas representarem tarefas de um algoritmo paralelo, ambas devem ser terminadas antes que a execução possa prosseguir. A atividade de sincronização é modelada na rede da figura abaixo por meio da transição t_2 , cujo disparo requer um *token* em p_1 e em p_3 , simultaneamente.

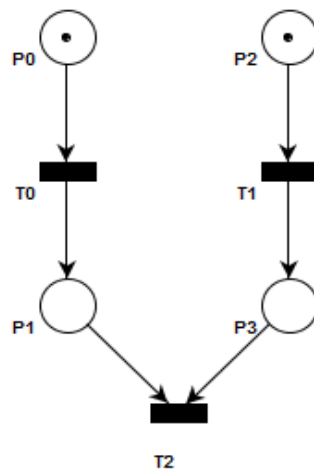


Figura 16: Modelagem de duas atividades paralelas em sincronismo com PN

1.4.6.3. Produtor-Consumidor

Nesta seção, apresentaremos o clássico problema do Produtor/Consumidor, onde podemos observar também alguns aspectos relativos ao compartilhamento de dados. Um processo produtor produz itens que são armazenados em um *buffer* do qual podem ser removidos e adquiridos pelo consumidor. O processo consumidor aguarda até que haja objetos disponíveis nesse *buffer* para que os possa consumir.

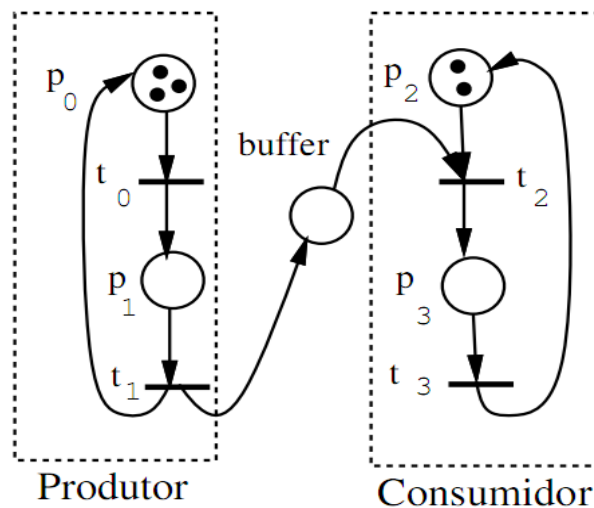


Figura 17: O problema Produtor-Consumidor com *buffer* ilimitado.

Fonte: [Maciel et al, 1996]

O número de marcas nos lugares p_0 e p_2 representa o número de processos produtores e consumidores, respectivamente. Uma marca em p_0 significa que o produtor

está pronto a produzir. Pelo disparo de t_0 e t_1 , que um item é produzido (uma marca é colocada no *buffer*) e o produtor está novamente apto a produzir um novo item. Se o consumidor está pronto a consumir (marca em p_2) e um item está no *buffer*, a transição t_2 pode ser disparada e remover um *token* do *buffer*.

Diante do funcionamento da PN do problema Produtor-Consumidor, notamos que o disparo de t_1 e t_3 armazena uma marca nos lugares p_0 e p_2 , respectivamente. Isso significa que o produtor ou o consumidor está pronto para uma nova execução de suas tarefas. Contudo, da maneira que o problema foi modelado, o produtor poderia ficar produzindo recursos indefinidamente e armazenando *tokens* no *buffer*, enquanto o consumidor aguarda por tempo indeterminado. Uma outra solução para isso é a adoção de um *buffer* com tamanho limitado, que condiciona a produção de recursos na medida em que os mesmos são consumidos, conforme podemos observar na figura a seguir:

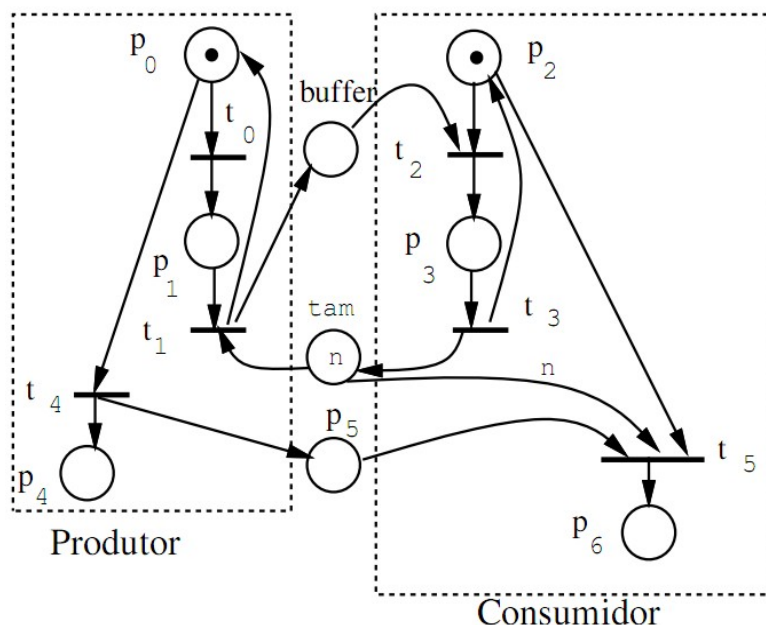


Figura 18: O problema Produtor-Consumidor com *buffer* de tamanho limitado.

Fonte: [Maciel et al, 1996]

1.4.6.4. Jantar dos Filósofos

Apresentaremos neste momento a modelagem por redes de Petri do problema do jantar dos filósofos proposto por Dijkstra¹ em 1965. Este problema descreve uma situação que consiste em uma mesa circular onde sentam um determinado número de filósofos, cada um com seu prato e um garfo entre dois pratos. Os filósofos que podem estar comendo,

1 **Edsger Wybe Dijkstra** foi um cientista da computação neerlandês conhecido por suas contribuições nas áreas de desenvolvimento de algoritmos e programas, de linguagens de programação (pelo qual recebeu o Prêmio Turing de 1972 por suas contribuições fundamentais), sistemas operacionais e processamento distribuído.

pensando ou com fome. Entretanto, para que um filósofo possa comer, ele precisa de dois garfos (o seu e o do filósofo vizinho). O problema é então o fato de que, se todos os filósofos pegarem o garfo da direita em um mesmo instante e aguardarem a liberação do garfo da esquerda, o sistema entrará em *deadlock*. Desta maneira, segue abaixo o modelo proposto por [Maciel *et al*, 1996] que resolve esse problema:

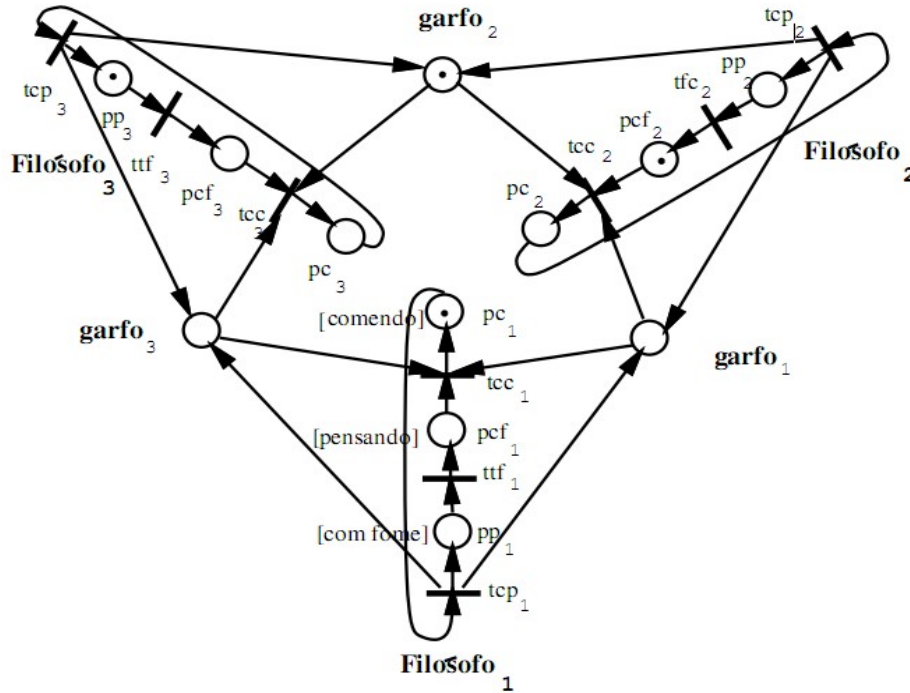


Figura 19: O Jantar dos Filósofos.

Fonte: [Maciel *et al*, 1996]

Nesta solução, os recursos foram representados por marcas nos lugares *garfo₁*, *garfo₂* e *garfo₃*. Os estados de cada filósofo são apresentados pelos lugares, *pp_i*, *pcf_i* e *pc_i*, que identificam os estados *pensando*, *com fome* e *comendo*, respectivamente.

1.5. Propriedades comportamentais

Segue abaixo as Propriedades Comportamentais das Redes de Petri de acordo com [Murata, 1989], [Penha *et al*, 2004] e [Maciel *et al*, 1996].

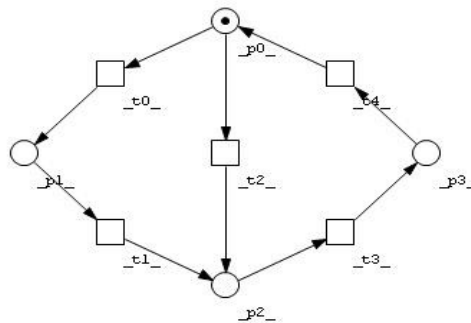
1.5.1. Alcançabilidade (*Reachability*):

Alcançabilidade ou *Reachability* é fundamental para o estudo de propriedades dinâmicas de qualquer sistema. A alcançabilidade indica a possibilidade de atingirmos uma determinada marcação pelo disparo de um número finito de transições, a partir de uma marcação inicial.

Definindo-se formalmente → Alcançabilidade: seja $M_i[t_j > M_k]$ e $M_k[t_h > M_l]$ então $M_i[t_j t_h > M_l]$. Por recorrência designamos o disparo de uma sequência $s \in T$ por $M[s > M']$. O conjunto de todas as possíveis marcações obtidas a partir da marcação M_0 na rede $N=(R, M_0)$ é denotado por $A(R; M_0) = \{M' \in \mathbb{N}^m \mid \exists s \mid M_0[s > M']\}$ onde m é a cardinalidade do conjunto de lugares.

Vejamos um exemplo:

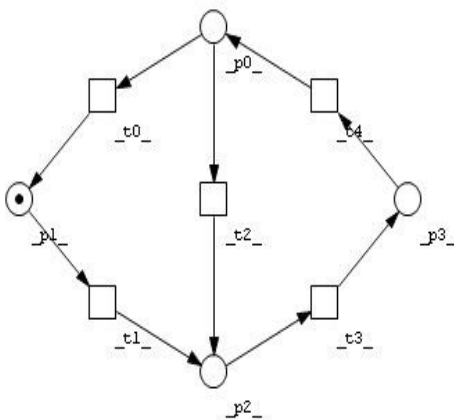
Para a rede de Petri marcada $N = (R, M_0)$, representada na figura abaixo, o disparo de uma transição t_1 , altera a marcação da rede. Uma marcação M' é dita acessível de M_0 , se existe uma sequência de transições que, disparadas, levam a marcação M' . Ou seja se a marcação M_0 habilita a transição t_0 , disparando-se esta transição atinge-se a marcação M_1 . A marcação M_1 habilita t_1 a qual sendo disparada atinge-se a marcação M_2 e assim por diante até a obtenção da marcação M' .



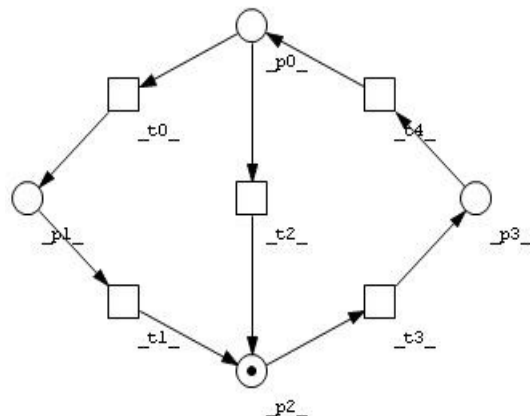
**Figura 20: Análise de Alcançabilidade:
Marcação Inicial**

O problema da análise desta propriedade consiste em determinarmos se uma dada marcação $M' \in A(R; M_0)$ da rede marcada. Na figura vista anteriormente desejamos, por exemplo, saber se a marcação $M' = (0,0,0,1)$ (lugar P_3 marcado) é acessível a partir da marcação inicial apresentada $M = (1,0,0,0)$ (apenas P_0 com marca). Observe a seguir que podemos atingir esta marcação tanto executando a sequência $s' = t_0 t_1 t_3$ como também executando a sequência $s'' = t_2 t_3$, ou seja esta marcação pode ser obtida a partir da marcação inicial.

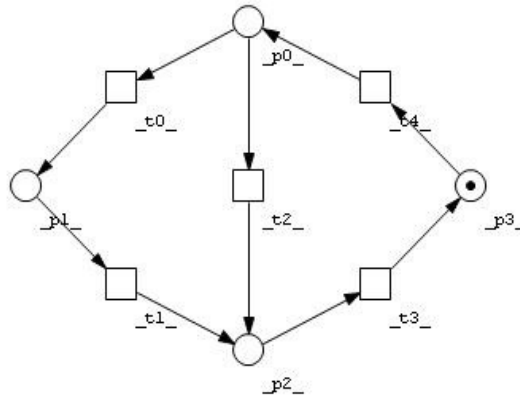
Sequência de transições $s' = t_0 t_1 t_3$:



**Figura 21: Análise de Alcançabilidade:
Disparo de t_0**

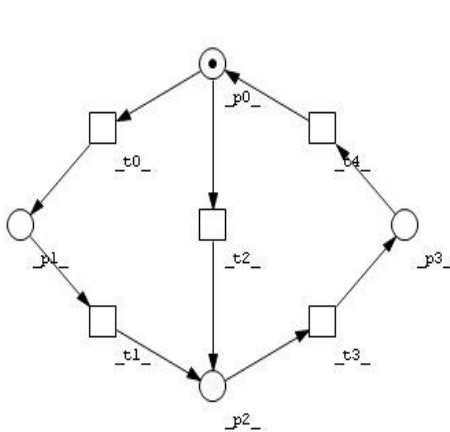


**Figura 22: Análise de Alcançabilidade:
Disparo de t_1**

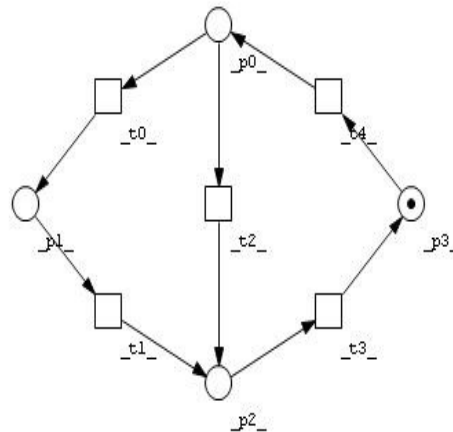


**Figura 23: Análise de Alcançabilidade:
Disparo de t_3**

Sequência de transições $s'' = t_2 t_3$:



**Figura 24: Análise de Alcançabilidade:
Disparo de t_2**



**Figura 25: Análise de Alcançabilidade:
Disparo de t_3**

Em alguns casos, desejamos observar apenas alguns lugares específicos da rede em estudo. Este problema é denominado sub-marcação alcançável. Aqui analisamos se uma dada marcação $M' \in A(R; M_0)$ da rede marcada N , onde M' é qualquer marcação restrita a um subconjunto de lugares. Muitos outros problemas de análise em termos do problema de alcançabilidade. Por exemplo, se uma rede fica em *deadlock* em uma determinada marcação, podemos querer saber se essa marcação é alcançável.

1.5.2. Limitação (*Boundedness*)

Uma definição de rede de Petri *boundedness* (limitação) pode ser encontrada em [Maciel *et al*, 1996].

Rede de Petri limitada: Diz-se que uma rede $N_1 = (P, T, I, O, M_0)$ é limitada (*bounded*) se $k(p_i) \leq \infty$ para todo $p_i \in P$. Isso significa que cada lugar poderá conter um número limitado (finito) de marcações.

Uma discussão que o [Cassandras *et al*, 2008] apresenta é bem oportuna, embora seja trivial: “Em muitos exemplos, os *tokens* representam clientes em um sistema de compartilhamento de recursos.” Por exemplo, os *tokens* no Lugar *Q* da rede de Petri na figura 26 representam clientes que estão entrando em uma fila. Claramente, permitir que as filas cresçam até o infinito é uma situação indesejável, uma vez que isso significa que os clientes irão esperar indefinidamente para serem atendidos. Na teoria clássica de sistemas, uma variável de estado que pode crescer até o infinito indica uma instabilidade no sistema. De forma similar, um crescimento ilimitado no estado dos componentes (marcações [ou seja, *tokens*]) indica alguma forma de instabilidade.

Boundedness se refere a propriedade de um lugar manter um número de *tokens* que nunca excede um dado inteiro positivo. A análise de um problema de *boundedness* consiste de uma rede é limitada e determinar esse limite de *tokens*. Caso essa propriedade comportamental não seja satisfeita, então a tarefa do analista direciona-se à alteração do modelo para garantir essa propriedade.

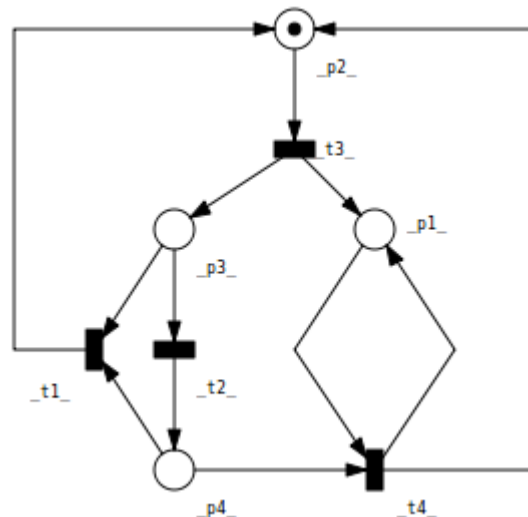


Figura 26: *Boundedness*

1.5.3. Segurança (*Safeness*)

Esta propriedade complementa a propriedade Limitação. Uma PN é dita segura se para todas as marcações alcançáveis todos os lugares da rede possuírem no máximo uma ficha cada um.

Definição:

uma rede $N_1 = (R, M_0)$ é definida como segura se $M(p_i) \leq 1$, para todo $p_i \in P$.

1.5.4. Vivacidade (*Liveness*)

Liveness é uma propriedade comportamental que está relacionada a inexistência de *deadlocks* (*deadlock-freedom*). Uma rede de Petri é dita ser *live* (os termos *live* e *liveness*, em diferentes referências, foram utilizados sem distinções) se, não importa qual marcação tenha sido alcançada a partir da marcação inicial M_0 , é possível disparar

qualquer transição da rede, através de qualquer sequência adicional de transições. Isso significa que, uma rede de Petri *live* garante a operação livre de *deadlock*, independentemente da sequência de transições selecionada.

Como a propriedade *liveness* é extremamente cara, os invariantes de transição podem ser utilizados para verificar a inviabilidade do disparo de transições, apontando para elementos repetitivos na rede e apresentando indícios de existência de transições mortas. Isto é feito através de cálculos sobre a matriz de incidência e sobre o conjunto de lugares ou transições com características especiais (como a conservação de *tokens* ou o funcionamento cíclico). Caso seja apresentada alguma transição não disparável, trata-se de uma possível evidência de não *liveness* na rede.

Além do citado acima, toda rede de Petri *live* e limitada é necessariamente coberta por invariantes de transição. Isto permite negar a propriedade de *liveness*, uma vez que se a rede for limitada e não for coberta por invariantes de transição, não podem ser *liveness*.

Como exemplo das afirmações acima, considere a rede exibida na figura abaixo:

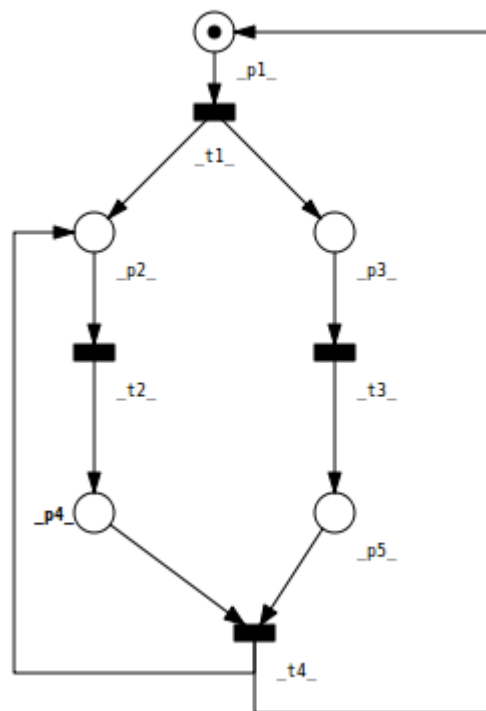


Figura 27. Liveness X Invariantes de Transição

1.5.5. Cobertura (*Coverability*)

Quando uma determinada marcação pode ser obtida através de uma outra marcação. Este é o problema cobertura de uma marcação;

1.5.6. Persistência

Se para qualquer par de transições habilitadas, o disparo de uma delas não desabilita o disparo da outra;

1.5.7. Reversibilidade

Quando existe o retorno à marcação inicial ou uma outra marcação qualquer.

1.5.8. Justiça (*Fairness*)

Justiça (*fairness*) é uma das principais propriedades comportamentais das redes de Petri. [Crepaldi, 2006] afirma que uma rede de Petri é justa se, para duas transições quaisquer, o número de vezes que uma é executada é finito em relação às vezes que a outra não é. Já em [Marranghello, 2005] encontramos a definição de que uma rede é dita justa se, para quaisquer duas ações, o número de vezes que uma é executada enquanto a outra não é executada é finito. Esta propriedade também é conhecida como justiça limitada. Se, por outro lado, uma dada sequência de disparos s for finita ou se cada ação da rede aparecer nesta sequência um número infinito de vezes, s é dita incondicionalmente justa. Uma rede de Petri é dita incondicionalmente justa se todas as sequências de disparo originárias em alguma marcação de $A(R, M_0)$ forem incondicionalmente justas.

De acordo com o que fora exposto anteriormente, percebe-se que existem dois pontos de vista para o significado de justiça, **Justiça limitada** e **Justiça incondicional**, que são apresentados a seguir:

Rede com justiça limitada: seja uma rede marcada $N = (R; M_0)$. N é dita *B-fair* se para todo par de transições $(t_i, t_j) \in T^2$, onde $i \neq j$, são *B-fair* [Maciel *et al*, 1996].

Segundo este ponto de vista, duas transições t_i e t_j são classificadas como *B-fair*, se o número de vezes que uma dispara, enquanto a outra não dispara, é limitado.

Exemplo:

Na figura a seguir temos para cada disparo da transição t_0 o disparo da transição t_1 e vice-versa. A mesma relação vale entre as transições t_2 e t_3 . Observamos também que se tomarmos os pares de transições: (t_0, t_2) , (t_0, t_3) , (t_1, t_2) e (t_1, t_3) disparamos t_0 ou t_1 duas vezes para cada disparo de t_2 ou t_3 , portanto verificamos que todos os pares de transição desta rede são *B-fair*. Veja:

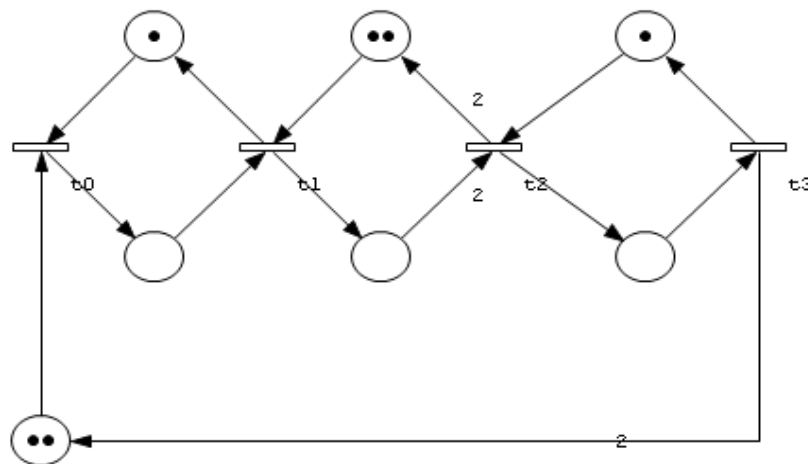


Figura 28: Exemplo de Justiça limitada

Rede Fair Incondicional: seja uma rede marcada $N = (R; M_0)$ e si uma sequência disparável de transições. N é dita Fair incondicional se para toda sequência si a partir de $M_i \in A(R; M_0)$ é Fair incondicional [Maciel et al, 1996].

Sob este ponto de vista, uma sequência de transições s_1 é classificada como *fair* incondicional se essa sequência é finita ou se todas as transições da rede aparecem um número infinito de vezes nessa sequência.

Exemplo:

Na figura a seguir as transições que compõem a rede (t_0, t_1) aparecem um número ilimitado de vezes na única sequência disparável de transições ($t_0, t_1, t_0, t_1, t_0, t_1, \dots$). Veja:

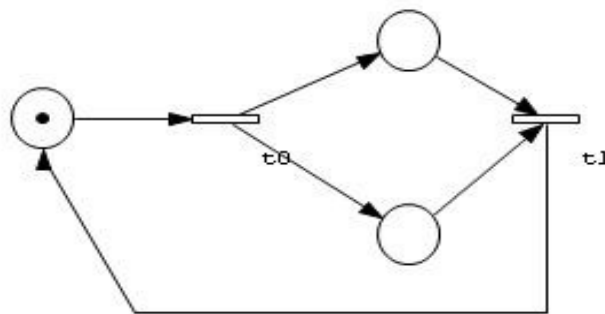


Figura 29: Exemplo de Justiça Incondicional

1.6. Propriedades estruturais

Segundo [Murata, 1989], [Penha et al, 2004] e [Maciel et al, 1996], segue abaixo as Propriedades Estruturais:

1.6.1. Limitação Estrutural

Uma rede de Petri é estruturalmente limitada se é limitada (*structural bounded*) para qualquer marcação inicial. Ou seja, independentemente da marcação inicial, o acúmulo do número de *tokens* em cada Lugar da rede é finito.

Dada uma rede de Petri $R = (P, T, I, O, M_0)$, pode-se verificar esta propriedade multiplicando-se um vetor de inteiros positivos W com dimensão $\#P$, tal que $C \cdot W \leq 0$. Ou seja, é possível realizar a verificação em uma rede através da resolução de um sistema de inequações lineares.

Exemplo: Considere a rede de Petri exibida na Figura abaixo:

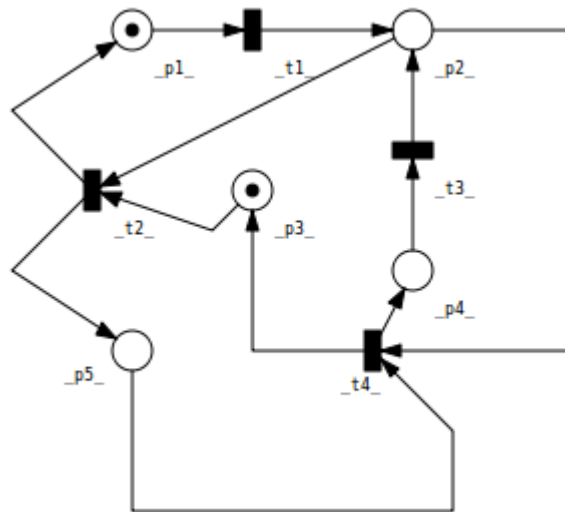


Figura 30. Rede Estruturalmente Limitada

Dada a matriz de incidência:

$$\begin{array}{c}
 \begin{array}{ccccc}
 P1 & P2 & P3 & P4 & P5 \\
 \hline
 t1 & -1 & 1 & 0 & 0 & 0 \\
 t2 & 1 & -1 & -1 & 0 & 1 \\
 t3 & 0 & 1 & 0 & -1 & 0 \\
 t4 & 0 & -1 & 1 & 1 & -1
 \end{array}
 \end{array}$$

Considere um vetor transposto $W^T = [w_1, w_2, w_3, w_4, w_5]$.

Fazendo o produto da matriz de incidência com o vetor transporte W^T , chega-se ao seguinte conjunto de inequações:

- i. $-w_1 + w_2 \leq 0$
- ii. $w_1 - w_2 - w_3 + w_5 \leq 0$
- iii. $w_2 - w_3 \leq 0$
- iv. $-w_2 + w_3 + w_4 - w_5 \leq 0$

Uma possível solução que torna as relações i. a iv. verdadeiras é:

$$W^T = [1 \ 1 \ 1 \ 1 \ 1]$$

Com esse resultado para W^T , podemos concluir que não há evidências que comprovem que essa rede não é estruturalmente limitada.

1.6.2. Conservação

Uma rede de Petri é estruturalmente conservativa se, para cada marcação inicial, a rede é conservativa com relação a um vetor W . Uma rede é conservativa com relação a um vetor $W = (w_1, w_2, \dots, w_n)$ se:

$$\sum w_i \bullet M_k(p_i) = \sum w_i \bullet M_0(p_i)$$

O somatório de *tokens* é igual, desde a marcação inicial até quaisquer das marcações acessíveis, ou seja, todas as transições adicionam exatamente a mesma quantidade de *tokens* dos lugares após à transição, bem como subtraem exatamente a mesma quantidade de marcas dos lugares que antecedem a transição. Desta forma, podemos afirmar que recursos não são criados nem destruídos em redes estruturalmente conservativas. Exemplo:

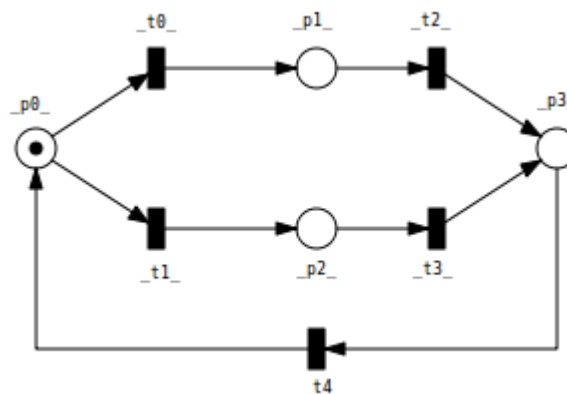


Figura 31: Rede estruturalmente conservativa.

1.6.3. Repetitividade

Vejamos agora algumas definições sobre Redes Repetitivas:

[Marranghello, 2005] afirma que uma rede de Petri é repetitiva se existir uma sequência de disparos, associada a uma dada marcação, na qual todas as ações são executadas um número de vezes infinito.

Para [Maciel *et al*, 1996] uma rede marcada é classificada como repetitiva se para uma marcação e uma sequência de transições disparáveis, para esta marcação, todas as transições da rede são disparadas ilimitadamente.

[Crepaldi, 2006] diz que se existe uma sequência de disparos, para qualquer marcação alcançável, em que todas as transições são disparadas infinitas vezes então, a rede é repetitiva.

Enfim, seja $N = (R; M_0)$ uma rede marcada e s uma sequência de transições. N é dita repetitiva se existe uma sequência s tal que $M_0[s > M_i$ e toda $t_i \in T$ dispara um número infinito de vezes em s . Vejamos um exemplo de rede Repetitiva

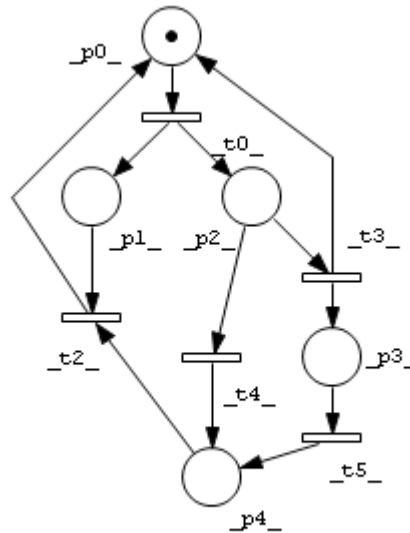


Figura 32: Exemplo de Rede Repetitiva

-Redes Parcialmente Repetitivas:

Segundo [Crepaldi, 2006] se apenas algumas transições, em determinada sequência de disparos, são disparadas infinitas vezes então a rede é parcialmente repetitiva.

[Marranghello, 2005] informa que se a sequência existir, mas apenas algumas ações forem disparadas ilimitadamente, a rede é dita parcialmente repetitiva.

Seja $N = (R; M_0)$ uma rede marcada e s uma sequência de transições. N é dita parcialmente repetitiva se existe uma sequência s tal que $M_0[s > M_i$ e algumas transições t_i disparam um número infinito de vezes em s [Maciel *et al*, 1996].

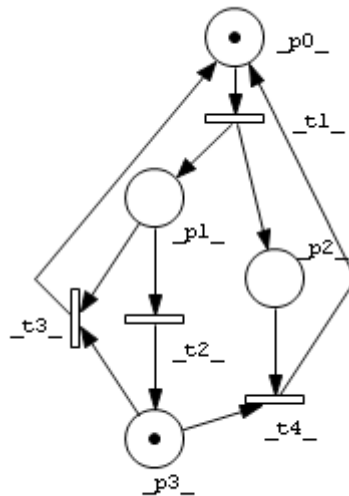


Figura 33: Exemplo Rede Parcialmente Repetitiva

Na figura anterior, podemos observar que as transições t1, t2 e t4 podem ser disparadas infinitamente, caracterizando uma rede repetitiva. Contudo, a transição t3 pode ser disparada apenas uma vez, ou seja, a rede é parcialmente repetitiva.

1.6.4. Consistência

Vejamos algumas definições sobre rede Consistente:

De acordo com [Maciel *et al*, 1996] uma rede de Petri tem a propriedade de consistência se disparando uma sequência de transições a partir de uma marcação inicial M_0 retornar-se a M_0 , porém todas as transições da rede são disparadas pelo menos uma vez.

[Marranghello, 2005] diz que uma rede de Petri é consistente se ao disparar uma sequência de ações a partir de uma marcação inicial, for possível retornar a esta marcação executando, pelo menos uma vez, cada uma das ações da rede. É possível mostrar que uma rede é consistente se, e somente se, existir um vetor δ , não nulo, tal que sua multiplicação pela matriz de incidência da rede de Petri seja igual a zero, isto é, $D \cdot \delta = 0$.

Já para [Crepaldi, 2006] uma rede de Petri é consistente se ao executar uma sequência de disparos “s”, a partir de uma marcação inicial M_0 , for possível retornar a esta marcação disparando pelo menos uma vez cada transição da rede.

Desta maneira, seja $N = (R; M_0)$ uma rede marcada e s uma sequência de transições. N é dita consistente se $M_0[s]M_0$ e toda transição t_i dispara ao menos uma vez em s.

Uma rede $N = (P, T, I, O)$ é consistente se ela é coberta por invariante de transição. Para maiores informações sobre invariantes de transição, vide tópico equivalente.

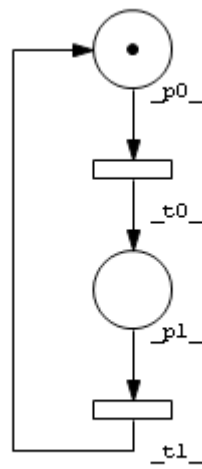


Figura 34: Exemplo de Rede Consistente

1.7. Referências

- [Bobbio, 1990] Bobbio, A. “System Modelling with Petri Nets”. Istituto Elettrotecnico Nazionale Galileo Ferraris. Torino, Italy. Reprinted from: A.G. Colombo and A. Saiz de Bustamante (eds.), System Reliability Assessment, Kluwer p.c., pp 102-143, (1990).
- [Barros, 1996] Barros, J. P. M. P. R. e. “CpPNeTS: uma Classe de Redes de Petri de Auto-Nível – Implementação de um sistema de suporte à sua aplicação e análise”. 221 p. Dissertação (Mestrado em Engenharia Informática) – Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Lisboa, Portugal, 1996.
- [Cassandras *et al*, 2008] Cassandras, C.; G. Lafortune, S. “Introduction to Discrete Event Systems”. Springer Science, 2008. 769p.
- [Crepaldi, 2006] Crepaldi, C. P. Redes de Petri e sua Aplicabilidade na Modelagem de Sistemas Computacionais. Universidade Estadual de Montes Claros. Montes Claros/MG. 2006. 66p.
- [Desel *et al*, 2003] Desel, J. Reisig W. Rozenberg, G. (Eds.). “Lectures on Concurrency and Petri Nets. Advances in Petri Nets”, Lecture Notes in Computer Science, vol. 3098, Springer-Verlag, 2004, ISBN: 3-540-22261-8. Originates from the Advanced Course on Petri Nets held in Eichstätt, Germany, September 2003.
- [Galdino, 2009] Galdino, S. M. L. “ISPN: modelagem e avaliação estocástica intervalar”. Tese de Doutorado. Universidade Federal de Pernambuco, Centro de Informática. Recife/PE. Junho de 2009.
- [Maciel *et al*, 1996] Maciel, P. R. M.; Lins, R. D.; Cunha, P. R. F. “Introdução às Redes de Petri e Aplicações”. X Escola de Computação, Campinas/SP. Departamento de Informática, Universidade Federal de Pernambuco, 1996.
- [Marranghello, 2005] Marranghello, N. “Redes de Petri: Conceitos e Aplicações”. DCCE/IBILCE/UNESP. 21p. Março de 2005.
- [Murata, 1989] Murata, T. “Petri Nets: Properties, Analysis and Applications”, Proceedings of the IEEE, Vol. 77, pp. 541-580. 1989.
- [Penha *et al*, 2004] Penha, D. O., Freitas, H. C., Martins, C. A. P. S. “Modelagem de Sistemas Computacionais usando Redes de Petri: aplicação em projeto, análise e avaliação”. IV Escola Regional de Informática RJ/ES, Vitória/ES, Rio das Ostras/RJ, 2004.
- [Peterson, 1977] Peterson, J. L., “Petri Nets”, Computing Surveys, Vol. 9, No. 3, September, 1977.
- [Peterson, 1981] Peterson, J. L. “Petri Net Theory and the Modeling of Systems”. Prentice-Hall, N.J., 1981. ISBN: 0-13-661983-5.
- [Petri, 1966] Petri, C.A. Kommunikation mit Automaten. Schriften des IIM Nr.2, Institut für Instrumentelle Mathematik, Bonn, 1962. (Available in English as: Communication with automata, Technical Report RADC-TR-65-377, Rome Air Development Center, Griffiss NY, (1966).

[Reisig, 1985] Reisig, W. “Petri Nets, An Introduction”. EATCS, Monographs on Theoretical Computer Science, W.Brauer, G. Rozenberg, A. Salomaa (Eds.), Springer Verlag, Berlin, 1985.

[Reisig *et al*, 1998] Reisig, W. Rozenberg, G. (Eds.). “Lectures on Petri Nets I: Basic Models”. Advances in Petri Nets, Lecture Notes in Computer Science, vol. 1491, Springer-Verlag, 1998, ISBN: 3-540-65306-6. Originates from the Advanced Course on Petri Nets held in Dagstuhl, Germany, October 1996.

[World, 2010] Petri Nets World. Maintained by TGI group at the University of Hamburg, Germany. Disponível em: < <http://www.informatik.uni-hamburg.de/TGI/PetriNets> >. Acesso em: setembro/2010.