Instituto Politécnico de Beja
Escola Superior de Tecnologia e Gestão

Introdução à modelação de sistemas utilizando redes de Petri

João Paulo M. P. Ramos e Barros

Beja, 6 de Janeiro de 2001

Nota Introdutória

Este texto é uma breve uma introdução às redes de Petri adaptada do primeiro capítulo de [Barros, 96].

Índice

Introdução

As redes de Petri (RdP) devem o seu nome ao trabalho de Carl Adam Petri que na sua dissertação de doutoramento, submetida, em 1962, à Faculdade de Matemática e Física da Universidade Técnica de Darmstadt na Alemanha, apresentou um tipo de grafo bipartido[1] com estados associados, com o objectivo de estudar a comunicação entre autómatos [Murata, 89][2]. O seu desenvolvimento posterior foi catalisado pelas suas numerosas potencialidades de modelação, designadamente: sincronização de processos, concorrência, conflitos e partilha de recursos. Desde essa data, têm sido desenvolvidos trabalhos teóricos e aplicações sobre RdP tendo este estudo levado, quer a um desenvolvimento das técnicas de análise das RdP e sua aplicação prática, quer ao desenvolvimento de muitas variantes ao modelo seminal das RdP tendo em vista aplicações específicas.

Como ferramentas matemáticas e gráficas, as RdP oferecem um ambiente uniforme para a modelação, análise formal e simulação de sistemas a eventos discretos, permitindo uma visualização simultânea da sua estrutura e comportamento. Mais especificamente, as RdP modelam dois aspectos desses sistemas, eventos e condições, bem como, as relações entre eles. Segundo esta caracterização, em cada estado do sistema verificam-se determinadas condições. Estas podem possibilitar a ocorrência de eventos que por sua vez podem ocasionar a mudança de estado do sistema [Peterson, 77: 228]. Como veremos, é possível relacionar, de uma forma intuitiva, condições e eventos com os dois tipos de nós da rede, respectivamente lugares e transições.

1. A estrutura da rede - lugares, transições, arcos e marcas

Os elementos dos dois conjuntos em que se podem dividir os nós constituintes de uma RdP denominam-se, respectivamente, *lugares* e *transições*. Os lugares são normalmente representados por circunferências ou elipses, e as transições por segmentos de recta, rectângulos ou barras. Os lugares encontram-se ligados às transições, e estas aos lugares, através de arcos dirigidos (Figura 1).

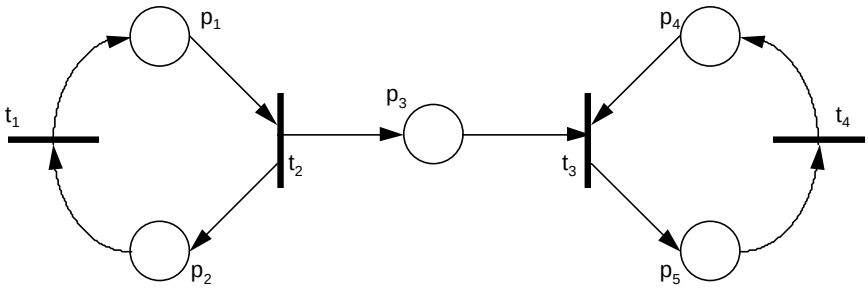


Figura 1 - Uma rede de Petri.

A estrutura de uma rede de Petri pode, pois, ser definida como um tuplo R = (L, T, AE, AS) [Peterson, 77: 235]conforme apresentado no Quadro 1

[1] Um grafo *G* denomina-se bipartido quando os seus nós podem ser divididos em dois conjuntos N_1 e N_2, tais que nenhum nó contido em N_1 ou N_2 se encontra ligado a outro nó contido no mesmo conjunto.

[2] Este trabalho contêm um breve resumo histórico sobre a origem das RdP.

R = (L, T, AE, AS) onde:

$L = \{ p_1, p_2, \ldots, p_m \}$	é um conjunto de lugares[3]
$T = \{ t_1, t_2, \ldots, t_n \}$	é um conjunto de transições
$L \cap T = \varnothing$	os conjuntos L e T são disjuntos
AE: $L \times T$	é um conjunto de arcos de entrada nas transições
AS: $T \times L$	é um conjunto de arcos de saída das transições

Por exemplo, para a Figura 1 temos:

$L = \{ p_1, p_2, p_3, p_4, p_5 \}$

$T = \{ t_1, t_2, t_3, t_4 \}$

$AE = \{ (p_2, t_1), (p_1, t_2), (p_3, t_3), (p_4, t_3), (p_5, t_4) \}$

$AS = \{ (t_1, p_1), (t_2, p_2), (t_2, p_3), (t_3, p_5), (t_4, p_4) \}$

Quadro 1 Definição de uma Rede de Petri.

Dado uma RdP constituir um modelo abstracto, podem ser dadas várias interpretações distintas a lugares e transições, conforme o tipo de aplicação em causa. No entanto, dadas as suas características intrínsecas, os lugares podem, de uma forma muito genérica e muito próxima da sua visualização gráfica, ser vistos como depósitos de recursos e as transições como acções que manipulam esses recursos. Os recursos são representados graficamente por pequenos círculos pretos dentro dos lugares. A cada um desses círculos dá-se o nome de *marca*. Os lugares surgem como representantes do estado da rede e as transições como responsáveis pela mudança de estado; desta forma, a RdP é simultaneamente orientada para os estados e para as acções. Ao estado da rede é usual chamar-se marcação, dado ser definido pela marcação de cada um dos seus lugares, ou seja, pela quantidade de marcas presentes em cada lugar. As marcas são representadas graficamente como pequenos círculos pretos, dentro dos lugares. A função das transições consiste em destruir e/ou criar estas marcas. Como as transições estão obrigatoriamente entre lugares é através da sua acção (denominada disparo) que um lugar altera a sua marcação. Os arcos indicam, para cada transição, os lugares sobre os quais estas actuam. Na Figura 2 vemos um exemplo de uma RdP marcada[4] correspondente à RdP da Figura 1.

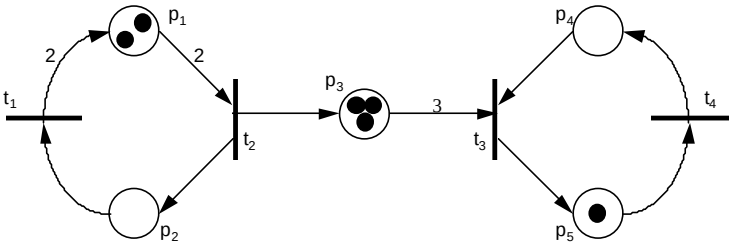


Figura 2 – Uma rede de Petri marcada.

Sendo as marcações nos lugares as representantes dos estado da rede, as transições surgem como os agentes que fazem a rede evoluir de estado para estado. Seguidamente veremos como se processa essa evolução.

2. A evolução da rede — o disparo das transições

Conforme já referido, são as transições que podem destruir e criar as marcas contidas nos lugares. Os arcos ligados a cada transição indicam exactamente sobre que lugares actuam. Um arco com origem num lugar e término numa transição (a partir daqui designado por arco de entrada), indica que essa transição subtrai, aquando do seu disparo, uma marca desse lugar. De forma simétrica, um arco com origem numa transição e fim num lugar (daqui em diante designado por arco de saída), indica que essa transição adiciona, aquando do seu disparo, uma marca a esse lugar. Assim sendo, podemos pensar nos arcos como indicando o sentido do movimento das marcas de um lugar para outro, atravessando a transição.

Daí resulta que uma transição só pode disparar se cada lugar de entrada contiver pelo menos uma marca, de forma a poder ser retirada no disparo da transição, pelo respectivo arco. Quando tal sucede diz-se que a transição

[3] Dado a a letra "l" ser facilmente confundida com o algarismo um ("1") optou-se pela utilização da letra "p" para denominar elementos do conjunto "L".

[4] Por enquanto, podemos ignorar ou números que surgem junto de três dos arcos.

está *habilitada*[5]. O disparo de uma transição é instantâneo, ou seja, as duas acções citadas são efectuadas ao mesmo tempo. Assim sendo, não existe nenhum instante no qual todos, ou parte, dos lugares de entrada já contêm menos uma marca e todos, ou parte, dos lugares de saída ainda não contêm mais uma marca, vide Figura 3

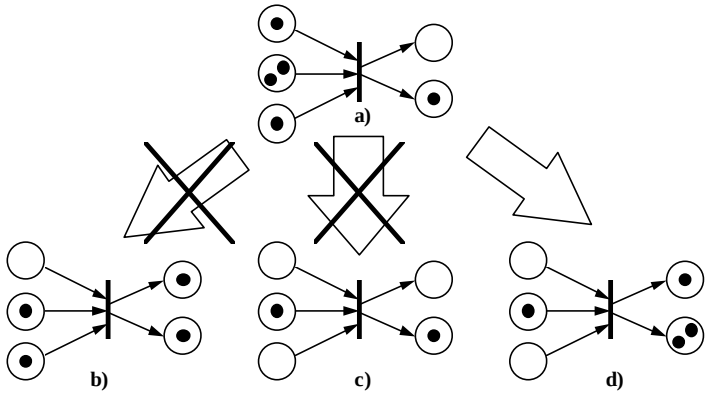


Figura 3 - Disparo de uma transição. a) corresponde à situação inicial. b) e c) <u>não</u> correspondem ao resultado do disparo da transição. Apenas d) representa a rede obtida após o disparo.

É muito importante notar que apesar de, por vezes, se falar em "movimento de marcas", tal é incorrecto do ponto de vista formal. O que acontece, aquando do disparo de uma transição, não é a mudança de marcas, dos lugares de entrada da transição para os lugares de saída, mas sim, a remoção de marcas nos lugares de entrada e a criação de novas marcas nos lugares de saída. Os arcos de saída são, portanto, criadores de novas marcas e não simples depositantes das marcas retiradas pelos arcos de entrada. Tal seria contraditório com o facto de uma transição poder possuir diferentes quantidades de arcos de entrada e de saída.

Um tipo de redes[6], muito frequente, denominado RdP generalizada, permite a existência de múltiplos arcos entre nós. Tal equivale a associar um inteiro a cada arco, constituindo, dessa forma, uma generalização do tipo de arcos já descrito. Nessas redes, cada arco não "transporta" necessariamente uma marca mas sim a quantidade especificada. A essa quantidade associada a cada arco dá-se o nome de *peso*.

A RdP até aqui descrita corresponde à apresentada em [Peterson, 77: 235]. Conforme já aí referido: "A maioria dos investigadores utilizam Redes de Petri generalizadas nos seus trabalhos, muitas vezes ignorando a distinção entre elas e aquelas que definimos como Rede de Petri" [Peterson, 77: 246]. Por exemplo, em [Murata, 89: 543] o autor apresenta, com a designação de Redes de Petri uma Rede de Petri generalizada. Por outro lado, a marcação surge como parte da definição de RdP, não sendo feita a distinção entre RdP e RdP marcada. Daí resulta uma definição mais extensa (Quadro 2) onde, para além dos pesos dos arcos, se inclui já a marcação. Por outro lado, efectuou-se uma compactação ao reunir os conjuntos relativos aos arcos.

$R = (L, T, A, PA, M_0)$ onde:	
$L = \{ p_1, p_2, \ldots, p_m\}$	é um conjunto de lugares
$T = \{ t_1, t_2, \ldots, t_n\}$	é um conjunto de transições
$L \cap T = \emptyset \ \wedge \ L \cup T \neq \emptyset$	os conjuntos L e T são disjuntos e não vazios
$A: (L \times T) \cup (T \times L)$	é o conjunto dos arcos
$PA: A \rightarrow \mathbf{N}$	são os pesos dos arcos
$M_0: L \rightarrow \mathbf{N_0}$	é a marcação inicial

[5] *enabled* na bibliografia em língua inglesa.

[6] O glossário do Apêndice A apresenta alguns dos principais tipos de RdP em ordem alfabética, bem como uma sua breve descrição informal.

Por exemplo, para a Figura 2 temos:
$L = \{ p_1, p_2, p_3, p_4, p_5 \}$
$T = \{ t_1, t_2, t_3, t_4 \}$
$A = \{ (p_2, t_1), (p_1, t_2), (p_3, t_3), (p_4, t_3), (p_5, t_4), (t_1, p_1), (t_2, p_2), (t_2, p_3), (t_3, p_5), (t_4, p_4) \}$

$PA((p_2, t_1)) = 1$	$PA((t_1, p_1)) = 2$	$M_0((p_1) = 2$
$PA((p_1, t_2)) = 2$	$PA((t_2, p_2)) = 1$	$M_0((p_2) = 0$
$PA((p_3, t_3)) = 3$	$PA((t_2, p_3)) = 1$	$M_0((p_3) = 3$
$PA((p_4, t_3)) = 1$	$PA((t_3, p_5)) = 1$	$M_0((p_4) = 0$
$PA((p_5, t_4)) = 1$	$PA((t_4, p_4)) = 1$	$M_0((p_5) = 1$

Quadro 2- Definição de uma Rede de Petri generalizada (com pesos) com marcação inicial.

Com base nesta definiçao diz-se que uma transição se encontra habilitada se e só se todos os arcos de entrada da transição tiverem associado um peso menor que a quantidade de marcas do respectivos lugares de saída. Formalmente: $(\forall (p, t) \in A)\ PA((p, t)) \leq M(p)$.

Daqui em diante uma Rede de Petri generalizada e marcada será denominada simplesmente por Rede de Petri ou RdP.

Tal como o sistema modelado, a RdP não prevê nem impõe uma ordem de ocorrência dos eventos. Tal significa que se em determinado instante, mais do que uma transição se encontra apta a disparar, qualquer uma pode disparar. O estado seguinte é resultado do disparo de um qualquer multiconjunto[7] de transições habilitadas. A este multiconjunto de transições aptas dá-se o nome de passo. A quantidade de estados seguintes possíveis corresponde pois à quantidade existente de diferentes passos.

3. Modelação com Redes de Petri

Quando modelamos um sistema através de uma RdP, estamos necessariamente a criar uma interpretação da rede[8]. É essa interpretação ou significação que efectua a ligação do modelo abstracto que qualquer RdP representa, com o sistema concreto que se pretende modelar. Por exemplo, uma possível interpretação da rede da Figura 2, é a que se apresenta na Figura 4 onde temos a modelação de um sistema produtor-consumidor [Reisig, 92].

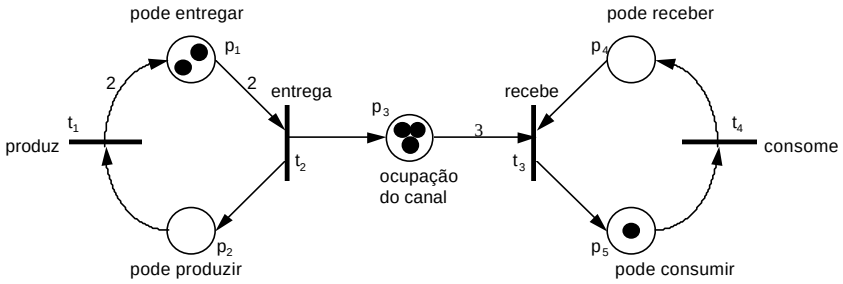


Figura 4 - Rede de Petri com pesos associados aos arcos (RdP generalizada) e marcada, com interpretação.

Podemos ver que os produtos são produzidos aos pares e que para cada um destes pares é entregue uma unidade. Por outro lado, o consumidor (modelado pelos lugares p_4 e p_5 e pelas transições t_3 e t_4) necessita de receber três unidades e, posteriormente, de as consumir, antes de poder receber mais unidades.

[7] Um multiconjunto é uma estrutura idêntica a um conjunto mas com uma única e importante diferença: pode conter mais do que uma ocorrência de um mesmo elemento.

[8] Interpretação no sentido de significação. Não confundir com "RdP interpretada". Essa designação pode ser utilizada com vários sentidos [Gomes, 91]. Por exemplo em [Peterson, 77] uma rede não-interpretada é uma rede em que não se atribuíram significados aos lugares, transições, etc.. Nesse caso apenas estamos interessados nas características abstractas da rede. A rede não tem significado concreto. Já em [David, 91], uma rede interpretada denomina uma classe específica de redes (ver Apêndice A).

As RdP permitem a modelação e visualização de diversos conceitos e relações, designadamente: paralelismo e concorrência, partilha de recursos, sincronização, memorização, limitação de recursos e leitura. Em [David et al., 92: 64-7] é possível encontrar as construções que permitem esse tipo de modelações.

Além do fluxo de controlo, as RdP podem também representar o fluxo de dados (*dataflow*). Num modelo de fluxo de dados os operadores são activados pela chegada dos operandos. Numa RdP os operandos são representados pela presença de marcas nos lugares e os operadores estão associados a transições.

As RdP não oferecem apenas uma representação para a estrutura e funcionamento de um sistema. Permitem a visualização do comportamento do sistema através do "movimento" das marcas. Conforme referido por Robert Valette[9], as RdP capturam a dinâmica dos sistemas a eventos discretos tornando-se dessa forma particularmente úteis para a sua simulação.

4. Representação de Redes de Petri

As RdP são grafos. Os grafos são estruturas de dados clássicas cujas técnicas de representação são já bem conhecidas [Sedgewick, 88] [Cormen et al., 90]. Existem duas formas fundamentais de representar grafos: através de uma matriz de incidências ou utilizando uma lista de adjacências. Como nas RdP, é frequente muitos dos nós não se encontrarem ligados entre si, a lista de adjacências constitui, em principio, uma melhor opção tendo em vista a poupança de memória. Consideram-se duas perspectivas quando à melhor representação de uma RdP: "a perspectiva do simulador/executor" e a "perspectiva do editor".

Para o simulador/executor, as transições constituem os elementos da rede que apresentam maior importância. São as transições que determinam a evolução da rede pelo que a estrutura que é necessário percorrer é a que contem as transições. Como para determinar a habilitação, ou não, das transições, são necessárias as marcações dos seus lugares de entrada e os pesos dos seus arcos de entrada, será conveniente que cada transição contenha a lista dos seus arcos de entrada que por sua vez contêm o lugar a que se encontram ligados. Os lugares contêm as respectivas marcações (quantidade de marcas). As transições conterão, também, a lista dos seus lugares de saída que por sua vez contêm o lugar respectivo. Importa assinalar que estas relações de inclusão podem corresponder apenas à inclusão de referências. Por exemplo, cada arco pode conter apenas uma referência para o lugar respectivo que se encontra numa lista de lugares. Essa lista pode ser percorrida de forma a consultar a marcação da rede de forma mais eficiente.

Em [Silva, 85: 315-72] são apresentadas várias representações possíveis (na "perspectiva do simulador/executor") para RdP seguras. Todas procuram maximizar a relação *rapidez/ocupação de memória*.

Na segunda perspectiva considerada, isto é, para um editor gráfico de RdP que interage com os elementos gráficos constituintes, não existe a noção de maior ou menor importância de um tipo de nó da rede face a outro. A adição ou remoção de lugares ou transições é igualmente custosa pois pode ser necessário remover um conjunto de arcos em ambos os casos. Ou seja, para o editor o facto do grafo ser bipartido apenas se reflecte na necessidade de evitar ligações inválidas de arcos. Interessa encontrar, inserir e/ou remover nós da rede. Como à medida que se adiciona mais informação a estrutura se torna mais complexa esse tipo de operações pode dar origem a apontadores que deixam de apontar para endereços válidos (*dangling pointers*). Uma das formas de evitar essa situação consiste em adicionar mais informação à estrutura, nomeadamente transformando listas ligadas simples em listas ligadas duplamente. Outra forma, consiste em marcar os elementos a remover e seguidamente percorrer a estrutura reconstruindo-a[10].

5. Propriedades das Redes de Petri

As propriedades das RdP encontram-se extensamente documentadas [Peterson, 77][Peterson, 81][Silva, 85][Murata, 89][Gomes, 91][David et al., 92][Zurawsky et al., 94]. Em [Murata, 89] as propriedades das RdP são classificadas em comportamentais ou estruturais conforme sejam ou não dependentes da marcação inicial.

---

[9] Em resposta à pergunta "Que propriedades das RdP as tornam utilizáveis na simulação?" colocada na petri net mailing list por Andreas Mittermayr (mitterma@fmi.uni-passau.de, andreas@newciv.org).

[10] Esta técnica foi sugerida por Henrik Hulgaard (henrik@cs.washington.edu) em resposta a uma pergunta de outro leitor da petri net mailing-list.

Dentro das propriedades comportamentais temos [Murata, 89: 547-50]: a alcançabilidade, a limitabilidade, a vivacidade, a reversabilidade, a cobertura, a persistência, a distância sincrónica e a justiça.

Dentro das propriedades estruturais refiram-se como de especial interesse [Murata, 89: 566-9]: vivacidade estrutural, controlabilidade, limitabilidade estrutural, conservabilidade, repetibilidade, consistência, justiça-B estrutural.

6. Métodos de Análise

Uma possível classificação dos métodos de análise de RdP considera a existência de três grupos [Murata, 89: 550]:

1. Análise por enumeração.

2. Técnicas de redução ou decomposição (transformações).

3. Matriz de incidência e equação de estado.

É ainda possível considerar um quarto tipo de análise de RdP: a simulação da RdP [Silva, 85]. Esta não permite obter garantias quanto a uma determinada propriedade do sistema modelado, mas por vezes pode ser a única técnica aplicável, dadas as limitações das restantes. Nomeadamente em RdP de alto-nível (vide §7.1) em que as restantes técnicas de análise ainda se encontram relativamente pouco desenvolvidas, a simulação apresenta uma grande importância.

A análise por enumeração baseia-se na construção de um grafo que representa todas as marcações que a RdP pode alcançar. Cada nó corresponde a uma marcação e cada arco corresponde ao disparo de um conjunto não vazio de transições, também denominado *passo*. Se a RdP é limitada, é possível construir este tipo de grafo e nesse caso ele denomina-se grafo de ocorrências [Jensen, 92]. Caso a RdP não seja limitada, o grafo de ocorrências é infinito. Nesse caso ainda é possível construir um grafo que se denomina *grafo de cobertura*[11]. Em [Murata, 89] e [David et al., 92: 36-7] encontra-se detalhado o algoritmo de construção da árvore, ou grafo, de cobertura.

Conforme notado em [Peterson, 77: 240] muitas questões podem ser reduzidas ao problema da alcançabilidade. No entanto, apesar das suas óbvias potencialidades e aplicabilidade[12], este método é muitas vezes dificilmente aplicável dada a sua natureza fortemente combinatória [Silva, 85: 101] com a consequente "explosão" de estados.

A análise por redução (ou transformação) consiste na obtenção de uma RdP mais simples mas que mantem as propriedades que se pretendem analisar. Este tipo de análise pode revelar-se muito útil mas apenas é aplicável a alguns tipos especiais de RdP ou em algumas situações particulares [Murata, 89: 550]. Em [Murata, 89: 553] apresentam-se, de forma resumida, as técnicas de redução mais simples. Em [Silva, 85: 110-28] e [David et al., 92: 46-60] é feita uma apresentação mais detalhada das principais técnicas de análise por redução.

A matriz de incidências e a equação de estados constituem a tentativa de utilização de álgebra linear na análise de RdP. Tal como as técnicas de análise por redução, estes métodos não são aplicáveis a muitos tipos de RdP. Nas RdP onde é possível a sua aplicação, nomeadamente os grafos marcados [Silva, 85: 101], a sua utilização permite, em alguns tipos de análise, evitar as técnicas de enumeração. Em [Murata, 89] encontra-se uma boa introdução a este tipo de técnicas de análise.

---

[11] Evitou-se, propositadamente, a designação "árvore de alcançabilidade" (*reachability tree*) que apresenta pelo menos dois significados distintos na literatura. Em [Murata, 89] é lhe atribuído um significado idêntico ao de grafo de alcançabilidade (igual a grafo de ocorrências [Jensen, 92]) embora utilizando representações gráficas distintas. Já em [Silva, 85] e [Peterson, 77][Peterson, 81], árvore de alcançabilidade tem o significado de árvore de cobertura de [Murata, 89] e [David et al., 92: 34-7]..

[12] É aplicável a todas as classes de RdP [Murata, 89: 550].

7. Algumas classes de Redes de Petri

Desde o trabalho seminal de Petri têm surgido muitas e diversas variantes ao seu modelo de Redes de Petri. Pode-se afirmar que a maior parte destas variantes nasceu da necessidade de adaptação das RdP ordinárias à especificidade da aplicação para as quais a sua utilização era desejada. O apêndice A contem uma lista de alguns dos tipos de RdP mais frequentemente referidos na literatura.

Conforme referido em [Morasca et al., 91], o modelo original das RdP falha na representação de duas importantes características: aspectos funcionais complexos, tais como, condições que determinam o fluxo de controlo, e os aspectos de temporização. Para enfrentar estas duas limitações duas classes de extensões às RdP foram desenvolvidas: as RdP de alto-nível e as RdP temporizadas. A classe de RdP desenvolvida no âmbito deste trabalho, apresenta ambas as características: é de alto-nível e temporizada. Por essa razão, estes dois tipos de extensão são discutidos seguidamente.

7.1 Redes de Petri de alto nível

A característica fundamental que distingue as RdP de alto nível (RdP-AN) das RdP e que as qualifica como de alto nível, é a possibilidade de as marcas não serem iguais entre si, correspondendo a elementos de um domínio. Desta forma, as marcas podem conter muito mais informação. Esta já não se limita à simples presença ou não da marca (ou marcas) num determinado lugar, podendo conter dados relativos à sua caracterização como indivíduo distinto dos restantes. Tal permite uma compactação (eventualmente muito significativa) da RdP, como veremos no exemplo típico do problema dos filósofos.

Dado as marcas não serem necessariamente iguais, torna-se necessário anotar a rede, designadamente os arcos e transições. As transições surgem como modificadores das marcas, podendo gerar marcas de domínios distintos dos das marcas presentes nos lugares de entrada.

Desta forma a complexidade da rede encontra-se dividida: parte é representada pela própria estrutura da rede - tal como sucede nas RdP ordinárias - e outra parte é representada pelas inscrições da rede. É importante notar que estas redes podem ser traduzidas para RdP de baixo nível, visto apresentarem o mesmo poder de modelação.

Numa rede de alto-nível em que cada lugar apenas possa conter, no máximo, uma marca podemos pensar nos lugares como variáveis (*valores-esquerdos*[13]). e nas marcas como valores de variáveis (*valores-direitos*). Este tipo de redes surge como uma generalização das RdP seguras ou binárias. Cada lugar já não corresponde necessariamente a uma variável *Booleana* (presença ou ausência de uma marca) mas pode corresponder a qualquer tipo de variável.

Se cada lugar poder conter mais do que uma marca (caso geral) podemos ver o lugar como uma estrutura de dados que contem um ou mais valores de um determinado tipo. Esta estrutura de dados é geralmente vista como um *multiconjunto*. O multiconjunto é a estrutura de dados que associada ao lugar, menos o afasta do lugar da RdP ordinária. Aí não existem diferenças entre as marcas pelo que não faz sentido falar em estruturas de dados associadas que devolvam diferentes valores consoante os critérios utilizados. Já nas RdP-AN podemos impor variadíssimos critérios resultantes das diferentes estruturas de dados que podemos associar a cada lugar. Por exemplo, alguns trabalhos consideram lugares *FIFO*[14] ou *LIFO*[15] [Pezzé, 94]. Outras estruturas podem ser utilizadas mas o multiconjunto é, de facto, aquela que permite a generalização natural dos lugares das RdP ordinárias e é a estrutura de dados em que se baseiam os dois principais tipos de RdP-AN: as RdP coloridas e as RdP Predicado-Transição. Os arcos de saída correspondem, nessas redes, às funções de adição de dados na

---

[13] Do inglês *left value* ou *l-value*, endereço de memória. O nome deriva da operação de afectação. Aí o endereço corresponde à variável do lado esquerdo e o valor à variável ou constante do lado direito. Por exemplo, na afectação *A = B*, *B* representa um valor-direito (em inglês: *right-value* ou *r-value*) presente na posição de memória correspondente à variável *B* e que será colocado no endereço designado pela variável *A*. O r-*value* corresponde ao valor presente no endereço de memória de uma dada variável [Sethi, 90].

[14] Do Inglês: *First In First Out*, estrutura em fila (*Queue*), primeiro a entrar é o primeiro a sair.

[15] Do Inglês: *Last In First Out*, estrutura em pilha (*Stack*), último a entrar é o último a sair.

estrutura e os arcos de entrada correspondem às funções de subtracção de dados da estrutura. É importante notar que não existe uma correspondência entre um tipo arco e uma função de modificação do valor de um campo da estrutura (uma marca). Esse efeito é conseguido através da conjugação das duas acções referidas: um arco de entrada retira a marca e um arco de saída adiciona uma nova marca.

Nas RdP-AN a cada lugar associamos um domínio ou tipo. A definição deste tem, como já referido, uma parte respeitante aos dados que pode conter. Por exemplo: um dado lugar pode ser responsável pela representação do conjunto dos livros requisitados numa biblioteca caso em que o seu tipo conterá um campo para o nome do livro, outro para o autor, outro para a editora, outra para a data de requisição, etc.. Se à definição do tipo adicionarmos as expressões dos arcos de entrada e de saída de todos os lugares desse tipo obteremos uma definição de tipo que contem quer a parte estática (reserva de memória para os seus dados) quer a parte de dinâmica de alteração desses dados. No exemplo apresentado uma função possível seria a de modificação da data de requisição do livro, a alterar sempre que o livro fosse de novo requisitado. Esta definição de um tipo corresponde ao que usualmente se denomina por tipo de dados abstracto. A definição dos tipos das variáveis como tipos de dados abstractos constitui uma das características da programação orientada por objectos [Meyer, 88][Booch, 94] tornando-a adequada para a implementação de RdP de alto-nível. A linguagem C++ permite a definição de tipos de variáveis que obedecem à definição apresentada em [Meyer, 88].

O facto das marcas poderem representar variáveis de um tipo de dados abstracto, permite que através da utilização de marcas persistentes[16] se consigam especificar lugares que se comportam como essas estruturas[17]. Para tal basta que o lugar contenha uma marca cujo tipo corresponda à estrutura de dados pretendida para o lugar. A Figura 5 mostra a especificação de um lugar *FIFO* utilizando este método. Para as inscrições da rede, utilizou-se um pseudocódigo muito próximo da linguagem C++. Os tipos correspondem a classes (directiva *Class*) e os lugares são declarados utilizando a directiva *Place*.

A maior parte do trabalho prático e teórico na área das RdP de alto-nível tem-se centrado ou nas RdP Predicado-Transição[18] [Genrich, 86] (RdPPr-Tr), ou nas RdP coloridas[19] [Jensen, 94] (RdPCol). Conforme notado em [Jensen, et al., 91], os dois modelos são muito semelhantes, sendo mais correcto vê-los como dois dialectos, superficialmente diferentes, de uma mesma linguagem.

---

[16] Numa RdP, considera-se uma marca persistente quando esta nunca *sai* do lugar, ou seja, para qualquer passo da rede, para cada arco de entrada existe um arco de saída que repõe a marca. No caso das RdP-AN podemos considerar uma definição mais lata em que a marca pode ser modificada. Por exemplo uma marca (de alto-nível) persistente, constituída por um identificador e uma fila de espera pode manter o identificador embora modificando o estado da sua fila de espera.

[17] Estas estruturas têm de ser limitadas caso se pretenda manter a possibilidade de traduçao da rede para uma RdP de baixo-nível.

[18] Em inglês: *Predicate-transition nets* ou *PrT-nets*.

[19] Em inglês: Coloured Petri Nets ou *CP-Nets*.

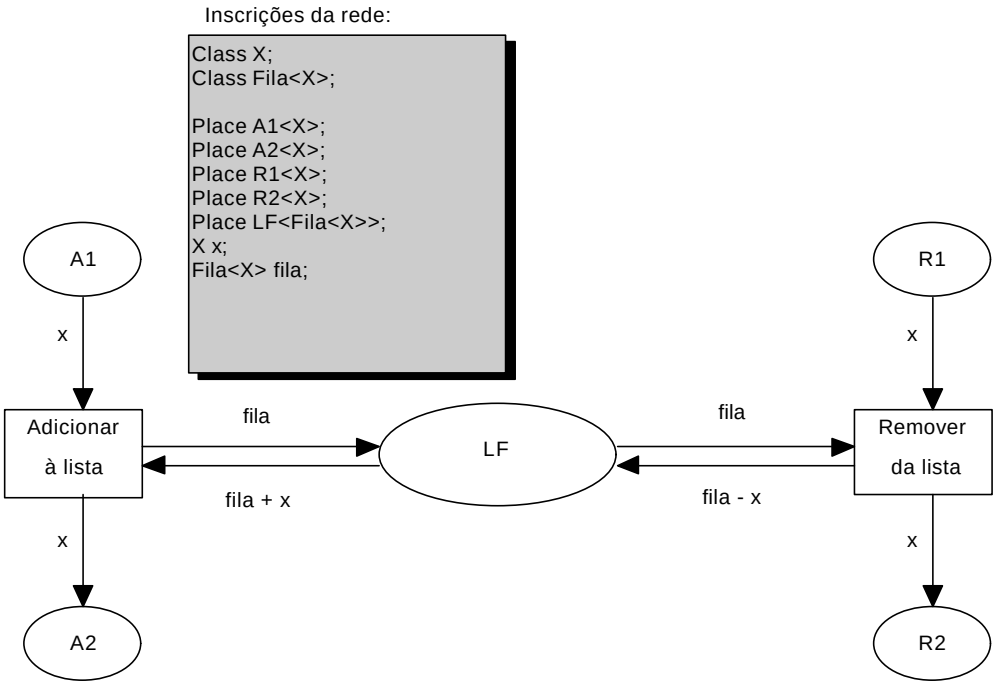


Figura 5 - Especificação de um lugar *FIFO*, com base num lugar contendo um multiconjunto de marcas, numa RdP de alto-nível.

Outra classe de RdP de alto nível é a das Redes de Petri com marcas individuais (RdPcMI) [Reisig, 92]. Como se verá, o nível de abstracção permitido por esta classe de redes não é tão elevado como o das já referidas RdPCol e RdPPr-Tr.

Em seguida faz-se uma breve apresentação das RdP coloridas que são as mais utilizadas.

Redes de Petri Coloridas

As Redes de Petri coloridas são, provavelmente, as RdP-AN que mais interesse têm levantado. O salto que se dá ao passar das RdP para as RdPCol é, mais do que em qualquer outro tipo de RdP de alto nível, extremamente semelhante ao que se dá ao passar da programação em linguagens *Assembly* para linguagens de alto-nível como PASCAL ou C [Kernigham et al., 88]. A utilização de estruturas de dados complexas em vez de *bits* tem o seu paralelo nas RdP, na utilização de marcas complexas em substituição das marcas simples das RdP ordinárias.

O, já citado, exemplo dos filósofos permite uma demonstração do poder de compactação da rede oferecido pelas RdPCol. Em §7.2 veremos como a introdução de uma representação hierárquica nas RdP permite a ocultação de sub-redes que surgem repetidas, através do recurso a instâncias de uma RdP que surgem na rede original como macrolugares ou macrotransições (§7.2). As RdPCol também podem facilmente suportar uma representação hierárquica com todas as vantagens que esta apresenta, mas a sua principal vantagem reside no elevado poder de compactação da rede, resultante, não de uma representação hierárquica mas sim, do facto das marcas corresponderem a variáveis de tipos cuja complexidade apenas se encontra dependente da linguagem utilizada para os definir, normalmente a mesma que é utilizada para todas as inscrições da rede. Na Figura 6 ilustra-se a transformação de uma RdP ordinária numa RdPCol, utilizando o bem conhecido problema dos filósofos [Ben-Ari, 82].

A transformação consiste fundamentalmente na substituição de conjuntos de lugares por um só lugar "colorido", ou seja, contendo marcas coloridas. Essas marcas coloridas, permitem a representação de cada um desses lugares através de valores (neste caso inteiros) distintos. Esta fusão de lugares obriga necessariamente a uma fusão dos respectivos arcos. Esta é conseguida associando funções aos arcos que permitam determinar quais as marcas a subtrair ou adicionar aos lugares. É o que acontece com a função Garfos() Figura 6b). É de salientar que a redução de complexidade obtida através da utilização da rede colorida (Figura 6b)) será tanto mais significativa, quanto o maior for a quantidade de filósofos. Para se obter uma representação de uma mesa com mais filósofos e

garfos, seria apenas necessário alterar a marcação inicial dos lugares `Pensando` e `Garfos disponíveis` e a função `Garfos`. Enquanto que em a) a própria rede cresceria, na quantidade de lugares e arcos, na razão directa do crescimento do número de filósofos e garfos.

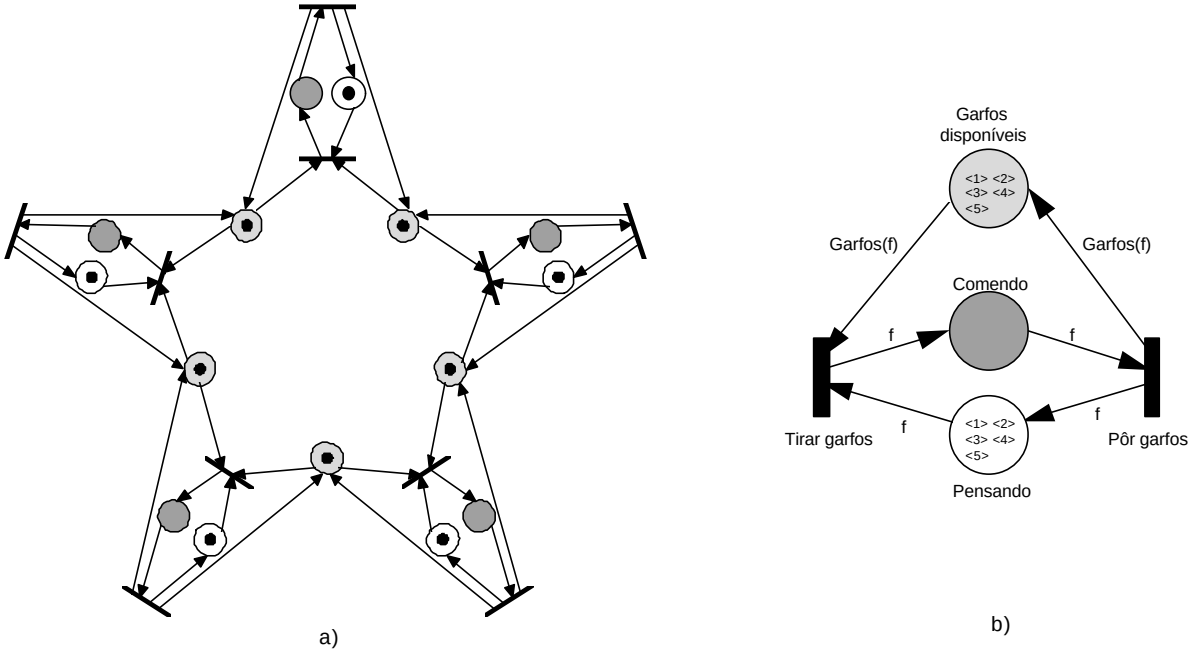


Figura 6 O problema dos filósofos: a) utilizando uma RdP ordinária; b) utilizando uma RdPCol. Os lugares em a) apresentam colorações distintas para evidenciar a sua "fusão" no lugar com idêntica coloração em b).

A selecção das marcas a subtrair ou adicionar aos lugares é suportada por um conjunto de variáveis, também chamadas variáveis de transição visto serem sempre utilizadas no contexto do disparo de uma transição. Como o cálculo do disparo de cada transição é feito em diferentes instantes, as variáveis podem ser declaradas num escopo global (conhecidas portanto de todas as transições) sendo, nesse caso, utilizadas por mais do que uma transição. A necessidade de existência de variáveis advém, tal como as expressões dos arcos, do facto, das marcas não serem todas iguais. A necessidade da existência de variáveis é uma das consequências directas da existência de marcas com valores associados. Esses valores têm de ser conhecidos durante o processo de disparo de uma transição pelo que as variáveis servem de depósito das marcas escolhidas. Podem então ser lidas com vista à avaliação da guarda e da determinação dos valores das marcas a gerar. Nas RdP ordinárias o facto de todas as marcas serem iguais, implica que o único dado importante no disparo da transição é a presença ou não da quantidade de marcas necessária nos arcos de entrada. As marcas geradas são aí apenas determinadas pelos pesos (RdP generalizada) dos arcos de saída. Não existe a noção de *valor* de cada marca.

Conforme referido, as variáveis na RdPCol servem para guardar os valores das marcas, dos arcos de entrada, seleccionadas. O par *(variável, valor selecionado),* denominar-se-á *vínculo*[20] e representar-se-á por: <*variável* ¬ *valor*>. Por exemplo, se a variável *x*, do tipo inteiro, tem o valor *3* atribuído então o vínculo correspondente será <*x* ¬ *3*>.

Dado que as marcas não são necessariamente iguais entre si, é extremamente útil a possibilidade de impor restrições ao disparo de uma transição com base nos valores das suas marcas. Para tal define-se uma função *Booleana* que permite testar os valores das marcas nos lugares de entrada, que à partida poderiam ser utilizadas no disparo de uma transição. A cada transição pode então associar-se uma função desse tipo denominada *guarda*.

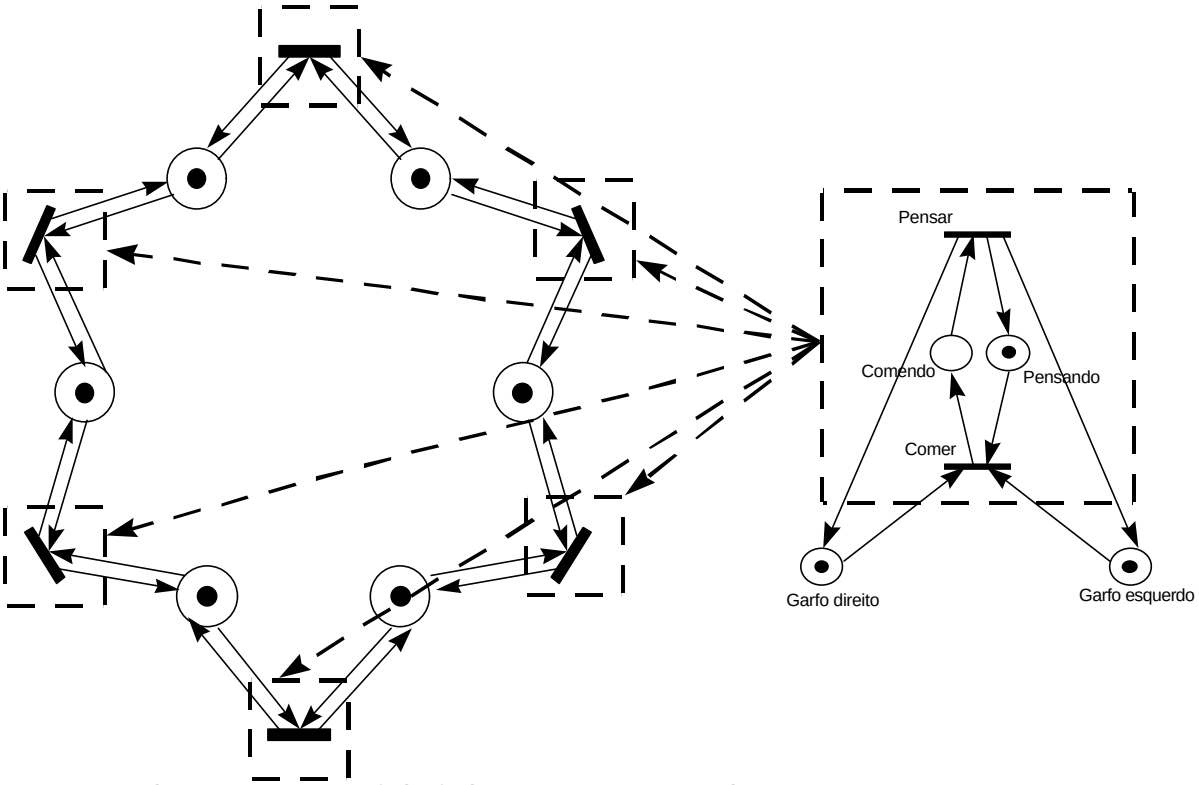
Para uma definição formal das RdPCol e seu comportamento dinâmico recomenda-se a leitura de [Jensen, 92]. Para uma análise formal mais detalhada de alguns aspectos, uma excelente continuação é [Jensen, 94].

[20] Em inglês: *binding*.

7.2 Redes de Petri hierárquicas

Uma RdP que modele um sistema não trivial, rapidamente se torna demasiado extensa para poder ser facilmente compreendida. Nestas situações o velho princípio “dividir para reinar”, surge como a forma mais óbvia de reduzir a complexidade e, desse modo, manter a legibilidade do modelo.

A forma mais natural e intuitiva de decompor uma RdP é a de considerar a existência de sub-redes que se comportam como lugares ou como transições. Desta forma, uma RdP passa a incluir mais dois conjuntos de elementos: os macrolugares e as macrotransições, conforme se tratem de sub-redes que se comportem como lugares ou como transições, respectivamente. Por exemplo, a rede da Figura 7a) pode ser representada através de



uma rede hierárquica em que cada filósofo é modelado por uma instância de uma macrotransição (Figura 7b)).

Figura 7 - Problema dos filósofos modelado por uma RdP com macrotransições. cada macrotransição (à direita na figura) representa um filósofo. A rede é equivalente às da Figura 6.

Numa macrotransição ou num macrolugar, os nós (lugares ou transições) que contactam com o exterior da macro (como acontece com as transições *Comer* e *Pensar* da Figura 7b)), denominam-se portos de entrada, de saída ou de entrada-saída conforme se encontrem ligados ao exterior por arcos de entrada, saída ou ambos, respectivamente. Daqui podemos concluir que os portos de uma macrotransição são necessariamente transições, e os portos de um macrolugar são necessariamente lugares. No entanto, se definirmos dentro de uma macrotransição lugares "fantasma" de interface, estes podem ser considerados como os lugares porto da transição. Estes lugares "fantasma" não são verdadeiros lugares pois são fundidos com os verdadeiros lugares da rede que inclui a macrotransição. Este tipo de lugares é utilizado nas RdPCol.

Também aqui é possível traçar um paralelo com as linguagens de programação: enquanto as RdP-AN e, em particular as RdPCol, têm a sua correspondência nas linguagens de alto-nível, as redes hierárquicas correspondem a uma linguagem *assembly* que disponibilize *macros*[21]. Assim temos:

Redes de Petri	Linguagens de Programação
RdP ordinárias	Linguagem *Assembly*
RdP hierárquicas	Linguagem *Assembly* com *macros*.
RdP alto nível	Linguagem de alto nível

Quadro 3 - Redes de Petri *versus* linguagens de programação.

7.3 Redes de Petri coloridas hierárquicas

Embora teoricamente interessantes e potencialmente úteis as RdP hierárquicas não resolvem eficientemente o problema do crescimento excessivo do número de nós da rede, quando se pretende modelar um sistema não trivial. Como vimos, as RdP-AN e designadamente as RdPCol, apesar de teoricamente possuírem o mesmo poder de modelação das RdP ordinárias, na prática apresentam uma muito maior capacidade de modelação do que as RdP ordinárias, mesmo considerando construções hierárquicas. Conforme já notado em [Huber et al., 90], a diferença é idêntica à que existe entre programar utilizando linguagem *assembly*, e programar utilizando uma linguagem com tipos de dados estruturados.

7.4 Redes de Petri não-autónomas

As RdP denominam-se autónomas quando a sua evolução apenas depende de condicionalismos resultantes da sua estrutura e marcação. As RdP com temporizações associadas são um exemplo de RdP não-autónomas, dado o seu comportamento depender, também, de condições impostas pelas temporizações associadas. Outra forma de redes não-autónomas resulta de considerarmos o disparo, de uma ou mais transições, dependente, não apenas da marcação dos seus lugares de entrada, mas também de uma condição externa associada. Esta vai permitir a sincronização da evolução da rede com eventos externos. Estas RdP denominam-se *sincronizadas*.

8. Conclusão

As RdP oferecem um suporte prometedor para a modelação, análise e simulação de sistemas a eventos discretos. O baixo-nível do modelo seminal das RdP torna extremamente difícil a sua aplicação para sistemas não triviais. Como tal várias RdP de alto-nível foram sendo desenvolvidas e estudadas. Todas apresentam a vantagem de permitirem especificações muito mais compactas tornando dessa forma possível a sua aplicação a sistemas muito mais complexos. Outra limitação do modelo seminal, é a impossibilidade de modelação do tempo. Tal é necessário para a modelação de sistemas com exigências de tempo-real. Em resposta a esta limitação várias forma de 'RdP com tempo" foram surgindo. A decomposição hierárquica pode ser aplicada às RdP e como é ainda mais útil quando se utilizam RdP de alto-nível. Por fim, referiram-se as RdP sincronizadas com eventos externos.

[21] Porção de código com um dado nome. Esse código pode ser invocado multiplas vezes num mesmo programa comportando-se como um sub-programa. A invocação do código (correspondente ao nome deste) é substituída textualmente antes do texto ser compilado.

Alguns tipos de Redes de Petri

Há, efectivamente, uma diferença notável entre o nome da coisa e o que deveras se passa com ela.

Richard P. Feynman

Existe já um número considerável de tipos de Redes de Petri. Por esta razão considerou-se interessante efectuar um seu levantamento que apesar de necessariamente incompleto, permitisse uma sistematização dos tipos de Redes de Petri, e respectiva nomenclatura, que mais frequentemente surgem na literatura. O glossário que em seguida se apresenta, fornece uma breve descrição de vários tipos de rede indicando sempre uma ou mais referências onde se pode encontrar a definição apresentada (e não só).

Autónoma [David et al., 92: 4]: quando descreve o funcionamento de um sistema que evolui de uma forma autónoma, i.e., cujos instantes de disparo ou não são conhecidos ou não são especificados. Esta designação não é realmente necessária mas permite evidenciar a distinção de uma RdP com estas características, de um RdP **não-autónoma**.

Binária [David et al., 92: 21]: O m. q. RdP **segura**.

Capacidade finita [Murata, 89: 543]: a cada lugar está associada uma "capacidade", correspondente à quantidade máxima de marcas que este possa conter. Neste tipo de redes a regra de disparo das transições contem uma condição adicional: o número de marcas nos lugares de saída não pode exceder as capacidades dos respectivos lugares. À regra de disparo com esta condição é dado o nome de *regra forte de transição*; sem esta condição denomina-se *regra fraca de transição*. Demonstra-se facilmente [Murata, 89: 543] que qualquer rede de Petri **pura** e de capacidade finita pode ser transformada numa rede de Petri equivalente onde se pode aplicar a regra fraca de transição.

Capacidade infinita [Murata, 89: 543]: cada lugar pode conter uma quantidade ilimitada de marcas.

Colorida [Murata, 89: 572]: um dos três tipos de redes de alto-nível, vide §0.

Com arco-inibidor [Peterson, 77:247][David et al., 92: 15].: RdP contendo "arcos de inibição". Estes arcos apresentam uma semântica oposta à dos arcos “normais”. Os arcos de inibição permitem testar a ausência de marcas nos lugares e tornam o poder de modelação das RdP igual ao das máquinas de Turing [Peterson, 77: 247].

Com marcas individuais [Murata, 89: 572]: um dos três tipos de redes de alto-nível, vide §7.2.

Com prioridades [David et al., 92: 17]: as transições têm prioridades de disparo, umas sobre as outras. Tal significa que em caso de conflito entre duas ou mais transições, disparará a que tiver uma prioridade mais elevada. Este tipo de redes não pode ser transformado numa RdP **ordinária**, dado o seu poder de modelação ser superior e idêntico ao das máquinas de Turing. Podem ser modeladas, por exemplo, por RdP **estendidas.**

Com teste-zero [David et al., 92]: O m. q. RdP **com arco-inibidor**.

Condição-Evento [Reisig, 92: 11-22]: RdP ordinárias, em que cada lugar contem no máximo uma marca. São, portanto, RdP **seguras**.

Conservativa*:* a quantidade de marcas na rede, é constante. Considerando RdP de baixo-nível, isto apenas pode suceder se o número de arcos de entrada de cada transição for igual ao número de arcos de saída. Nalguns sistemas pode ser conveniente considerar que uma única marca pode representar mais do que um recurso. A conservatividade da rede pode então medir-se com base no número de recursos. Daqui resulta outra possível definição de RdP conservativa [Zurawsky et al., 94: 572]. Se existe um vector $w = [w_1, w_2, ..., w_m]$, onde m é o número de lugares e $w(p) > 0$ para cada lugar $p \in P$, tal que a soma pesada das marcas se mantem em todas as marcações alcançadas partindo de uma marcação inicial M_0.

Contínuas [David et al., 92: 125-79]: A marcação de cada lugar não é um número inteiro positivo, mas sim um número real positivo. O disparo de uma transição corresponde à subtracção e adição de uma quantidade real de marcas aos lugares de entrada e de saída respectivamente. Estas redes permitem a modelação de sistemas que não podem ser modelados pelas RdP ordinárias.

Escolha-assimétrica [Murata, 89: 554][David et al., 92: 8-9]: se quaisquer dois lugares têm transições de saída comuns, então o conjunto de incidência posterior de pelo menos um deles deverá estar contido no conjunto de incidência posterior do outro: $\forall l_1, l_2 \in L : l_1\bullet \cap l_2\bullet \neq \varnothing \Rightarrow l_1\bullet \subseteq l_2\bullet \vee l_1\bullet \supseteq l_2\bullet$ [Murata, 89: 554]. O m. q. RdP **simples** [David et al., 92: 8-9].

Escolha-livre [Peterson, 77: 248][Murata, 89: 554]: cada arco é o único arco de saída de um lugar ou o único arco de entrada de uma transição: $\forall l_1, l_2 \in L: l_1\bullet \cap l_2\bullet \neq \varnothing \Rightarrow |l_1\bullet| = |l_2\bullet| = 1$. Esta restrição significa que se existe uma marca num lugar então ela continuará nesse lugar até a sua única transição de saída disparar ou, se existirem múltiplas transições de saída para o lugar, será possível escolher livremente qual das transições de saída irá disparar. Subconjunto das RdP de **escolha-livre-estendida**.

Escolha-livre-estendida: todos os lugares que partilham transições de saída têm de possuir as mesmas transições de saída: $\forall l_1, l_2 \in L: l_1\bullet \cap l_2\bullet \neq \varnothing \Rightarrow l_1\bullet = l_2\bullet$ [Murata, 89: 554]. Subconjunto das RdP de **escolha-assimétrica**.

Estocástica [Zurawsky et al., 94: 578][Marsan et al., 84][Murata, 89: 570]: redes temporizadas em que o tempo associado aos lugares ou transições é modelado através de variáveis aleatórias. Nestes modelos tornou-se convenção associar as temporizações apenas às transições [Zurawsky et al., 94: 578]. Nas RdP estocásticas, as variáveis aleatórias têm uma distribuição exponencial.

Estendidas [Peterson, 77: 246] [Murata, 89: 546]: segundo [Peterson, 77: 246], *estendidas* é uma designação que engloba as RdP que apresentam um poder de modelação superior ao das RdP ordinárias. São exemplos deste tipo de redes, as RdP **generalizadas** e as RdP com **arco-inibidor**. Segundo [Murata, 89: 546] a designação RdP estendidas resume-se a uma designação para RdP **com arcos-inibidores**[22].

Generalizada [Peterson, 77: 246][David et al., 92: 11]: é permitida a existência de mais do que um arco entre um lugar e uma transição ou entre uma transição e um lugar. Este tipo de rede tem o mesmo poder de modelação de uma RdP **ordinária**. Na sua representação gráfica é usual manter o desenho de apenas um arco entre os nós em causa, adicionando a inscrição de um número que indica a quantidade de arcos (ou peso do arco) que se pretende representar, entre os nós. O m. q. RdP **lugar-transição**.

Grafo de estados [David et al., 92: 6]: o m. q. **máquina de estados**.

Grafo de eventos [David, 91: 146][David et al., 92: 6] o m. q. **grafo marcado** ou **grafo de transições.**

---

[22] Provavelmente por na mesma referência ser dado o nome de *RdP* às *RdP **generalizadas*** de [Petersom, 77: 246].

Grafo de transições [David, 91: 146][David et al., 92] o m. q. **grafo marcado** ou **grafo de eventos**.

Grafo marcado [Peterson, 77: 247] [David et al., 92: 6] cada lugar tem exactamente um arco de entrada e um arco de saída. Um grafo marcado permite a representação de actividades paralelas, mas não permite representar decisões (conflitos) [Peterson, 77: 247]. Um grafo marcado é o dual do **grafo de estados** [David et al., 92: 6].

Híbridas [David et al., 92: 125-79]: contêm lugares e transições continuos (como nas RdP **contínuas**) e discretos.

Interpretada [Peterson, 77 : 230][Peterson, 81: 58-9][David, 91: 143][David et al., 92: 99]: rede à qual se associou uma interpretação, ou significado, aos seus lugares e transições, passando por isso a corresponder a algo real que se pretende modelar. Por oposição uma rede não interpretada não apresenta qualquer significado para os seus lugares e transições, é uma representação totalmente abstracta [Peterson, 77 : 230] e [Peterson, 81: 58-9].

Outro significado totalmente distinto é o apresentado em [David, 91: 143] [David et al., 92: 99]. Aí uma rede interpretada apresenta três características: é sincronizada, temporizada-L e inclui uma parte de processamento de dados cujo estado é definido por um conjunto de variáveis. Este estado é modificado por um conjunto de operações que estão associadas aos lugares e determina o valor das condições (ou predicados) que estão associados com as transições. Numa RdP interpretada, uma transição dispara quando está apta, a condição é verdadeira e o evento ocorre. Conforme afirmado na mesma referência, este modelo está muito próximo do Grafcet [David et al., 92][Novais, 94], um modelo definido para modelar controladores lógicos.

Livre-de-conflitos [David et al., 92]: cada lugar tem, no máximo, uma transição de saída.

Lugar-Transição [Reisig, 92: 25-33][Jensen, 92: 2-8]: O m. q. RdP **generalizada**.

k-limitada ou limitada: RdP na qual, para qualquer marcação alcançável, nenhum lugar apresenta mais de *k* marcas.

Máquina de estado [Peterson, 77: 247]: cada transição tem exactamente um arco de entrada e um arco de saída. Uma máquina de estado permite a representação de decisões (conflitos) mas não permite representar a sincronização de actividades paralelas.

Não-autónoma [David et al., 92: 17, 71-124].: uma rede que permite especificar não apenas *o que* "acontece" mas também *quando* "acontece". São redes sincronizadas com eventos externos (RdP sincronizadas) e/ou cujos passos estão dependentes do tempo (RdP temporizadas).

Ordinária [Murata, 89: 543]: Todos os arcos têm peso um. Em [Peterson, 77: 235] corresponde à definição de Rede de Petri (sem qualquer qualificação, portanto).

Predicado-Transição [Genrich, 86][Murata, 89: 572]: um dos três tipos de redes de alto-nível, vide §1.

Pura [Murata, 89: 543].: rede que não contem auto-ciclos.

Segura [David et al., 92: 21]: Característica de uma RdP em que em todas as marcações alcançáveis (a partir de uma dada marcação) nenhum lugar contem mais do que uma marca. O m. q. RdP **binária**.

Simples [David et al., 92] [Murata, 89].: cada transição é afectada, no máximo, por um conflito. O m. q. RdP de **escolha-assimétrica**.

Sincronizada [David et al., 92:]: numa rede autónoma, uma transição pode disparar se está habilitada. Numa rede sincronizada, cada transição tem um evento associado e disparará, obrigatoriamente, se estiver habilitada quando o evento ocorrer.

Temporizada [Kumar et al., 94: 1500]: permite a descrição de um sistema cujo funcionamento está dependente do tempo. Existem duas formas principais de modelar o tempo numa RdP temporizada: ou este está associado aos lugares (RdP temporizada-L) ou está associado às transições (RdP temporizada-T).

Temporizada-L: vide **temporizada**.

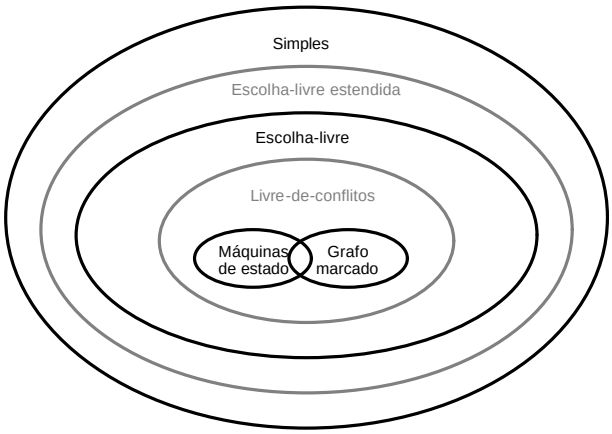
Temporizada-T: vide **temporizada**.

Persistente: se para quaisquer duas transições aptas, o disparo de qualquer delas nao incapacita a outra. Todos os grafos marcados são persistentes mas o contrário nao é verdade.

Reversivel [Zurawsky et al., 94: 573]: rede em que partindo de qualquer marcação é posível regressar à marcação inicial.

Viva: (ou, de forma equivalente, se uma dada marcação é viva) se, em qualquer marcação alcançada, é possível disparar pelo menos uma transição.

Relação de inclusão entre alguns tipos de RdP:



Bibliografia

[Barros, 96] Barros, João Paulo, *CpPNeTS: uma Classe de Redes de Petri de Alto-nível Implementação de um sistema de suporte à sua aplicação e análise*, dissertação apresentada na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, como parte dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, Lisboa, 1996.

[Bastide et al., 95] Bastide, Rémi e Palanque, Philippe, "A Petri Net Based Environment for the Design of Event-Driven Interfaces", in *Proceedings of the 16th International Conference on Application and Theory of Petri Nets*, Giorgio de Michelis Michel Diaz (Eds.), Turin, Italy, LNCS 935, June 1995.

[Ben-Ari, 82] Ben-Ari, M., *Principles of Concurrent Programming*, Prentice-Hall, 1982.

[Booch, 94] Booch, Grady, *Object-Oriented Analysis and Design with Applications*, 2nd edition, The Benjamin/Cummings Publishing Company, Inc., 1994.

[Cormen et al., 90] Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., *Introduction to Algorithms*, second printing, MIT Press, 1990.

[David, 91] David, René, "Modeling of Dynamic Systems by Petri Nets", ECC 91 European Control Conference, Grenoble, France, July 2-5 1991.

[David et al., 92] David, René e Alla, Hassane, *Petri Nets and Grafcet Tools for modelling discrete event systems*, Prentice Hall, 1992.

[Genrich, 86] Genrich, H. J., "Predicate/Transition Nets" in [Jensen, et al., 91:3-43].

[Gomes, 91] Gomes, Luís Filipe dos Santos, *Especificação de sistemas digitais*, trabalho de síntese, submetido para a prestação de provas de aptidão pedagógica e capacidade científica, Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Novembro 1991.

[Huber et al., 90] Huber, P., Jensen, K., e Shapiro, R.M., "Hierarchies in Coloured Petri Nets", in [Jensen, et al., 91:215-43].

[IEC, 92] International Electrotechnical Comission, Draft Int. Standard, Feb. 14, IEC 1131: Programmable Controllers, Part 3: Programming Languages, 1992.

[Inan et al., 88] Inan, Kemal, Varaiya, Pravin, "Finitely Recursive Process Models for Discrete Event Systems", *IEEE Transactions on Automatic Control*, Vol. 33, No. 7, July 1988.

[Itmi, s.d.] Itmi, Mhamed, "Designing Generalized Petri Nets", s.d..

[Jensen, 87] Jensen, "Coloured Petri Nets", *Advances in Petri Nets 1986*, (W. Brauer, W. Reisig, G. Rozemberg Eds), Lecture Notes Computer Science 254, Berlin-Heidelberg-New York: Springer-Verlag.

[Jensen, 90] Jensen, K., "Coloured Petri Nets: A High Level Language for System Design and Analysis", *in* [Jensen, et al., 91:342-417].

[Jensen, et al., 91] Jensen, K. e Rozemberg(Eds.) *High-level Petri Nets theory and application*, Spinger-Verlag, Berlin Heidelberg, 1991.

[Jensen, 92] Jensen, Kurt, *Coloured Petri Nets Basic Concepts, Analysis Methods and Practical Use*, Volume 1, Springer-Verlag, Berlin Heidelberg, 1992.

[Jensen, 94] Jensen, Kurt, "An Introduction to the Theoretical Aspects of Coloured Petri Nets", *A Decade of Concurrency*, Lecture Notes in Computer Science vol. 803, Springer-Verlag 1994, pp. 230-72.

[Jensen, 95] Jensen, Kurt, *Coloured Petri Nets Basic Concepts, Analysis Methods and Practical Use*, Volume 2, Springer-Verlag, Berlin Heidelberg, 1995.

[Keen et al., 93] Keen, C. D., Lakos, C. A., "A Methodology for the Construction of Simulation Models using Object-Oriented Petri Nets", paper presented at the *1993 European Simulation Multiconference on Adapting Object-Oriented Development Methodologies to the Design of Discrete Event Simulation Models*, 1993. Disponível em http://www.cs.utas.edu.au/ Research/ opn.html.

[Kernigham et al., 88] Kernighan, Brian W. e Ritchie, Dennis M., *The C Programming Language*, second edition, AT&T Bell Laboratories, Murray Hill, New Jersey, Prentice Hall, Englewood Cliffs, New Jersey, E.U.A..

[Kumar et al., 94] Kumar, Devendra e Harous, Saad, "Distributed Simulation of Timed Petri Nets: Basic Problems and Their Resolution", *IEEE Trransactions on Systems, Man, and Cybernetics*, vol. 24, no.10, October 1994.

[Levine et al., 92] Levine, John R., Manson, Tony e Brown, Doug, *lex & yacc*, second edition, O'Reilly & Associates, Inc, 1992.

[Lindqvist, 90] Lindqvist, M., "Parametrized Reachability Trees for Predicate/ Transition Nets" *in* [Jensen, et al., 91:351-72].

[Magee et al., 99] Magge, Jeff, e Kramer, Jeff, *Concurrency State Models and Java Programs*, John Wiley & Sons, 1999.

[Manduchi et al., 94] Manduchi, G., Moro, M., "An Object Oriented Approach in Building an Environment for Simulation and Analysis Based on Timed Petri Nets with Multiple Execution Policies", IEEE, 1994.

[Marsan et al., 84] Marsan, A. M., Balbo, G., e Conte, G., "A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems", ACM Trans. Comput. Syst., vol. 2, no. 2, pp. 93-122, May 1984.

[Marsan et al., 86] Marsan, A. M., Balbo, G., e Conte, G., *Performance Evaluation of Multiprocessor Systems using Petri Nets*. Cambridge, M.A.: MIT, 1986.

[Merlin et al., 76] Merlin, P. M. e Farber, D. J., "Recoverability of Communication Protocols", *IEEE Transactions on Communications*, vol. COMM-24.

[Meyer, 88] Meyer, Bertrand, *Object-oriented Software Construction*, Prentice-Hall International (UK) Ltd., 1988.

[Morasca et al., 91] Morasca, Sandro, Pezzé, Mauro e Trubian, Marco, "Timed High-Level Nets", *The Journal of Real-Time Systems*, vol. 3, no. 2, pp. 165-89, May 1991.

[Murata, 89] Murata, Tadao, "Petri Nets: Properties, Analysis and Applications", *Proceedings of the IEEE*, Vol. 77, No. 4, April 1989.

[Murata, 95] Murata, Tomohiro, "Application of Petri Nets to Sequence Control Programming", *in* [Zhou, 95: 43-68]

[Peterson, 77] Peterson, James L., "Petri Nets", *Computing Surveys*, Vol. 9, No. 3, September 1977.

[Peterson, 81] Peterson, James L., *Petri Net Theory and the Modelling of Systems*, Prentice-Hall, 1981.

[Pezzé, 94] Pezzé, Mauro, "Cabernet: A Customizable Environment for the Specification and Analysis of Real-Time Systems", Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy, ftp://ipeset4.elet.polimi.it/dist/Cabernet, June 2, 1994,.

[Ramchandani, 74] Ramchandani, C., "Analysis of asynchronous concurrent systems by timed Petri nets", Project MAC Technical Report MAC-TR-120, Massachusetts Institute of Technology, Cambridge MA, 1974.

[Raymond et al., s.d.] Raymond, Serge & Paludetto, Mario, Université Paul Sabatier/ LAAS-CNRS, Toulouse, France, "C++ et le Paralelism pour une Methode Orientees Objects", s.d..

[Reisig, 92] Reisig, W., *A Primer in Petri Net Design*, Springer-Verlag, Berlin Heidelberg, 1992.

[Sedgewick, 88] Sedgewick, Robert, *Algorithms*, second edition, Addison-Wesley Publishing Company, 1988.

[Sethi, 90] Sethi, Ravi, *Programming Languages Concepts and Constructs*, AT&T, Addison-Wesley Publishing Company, 1990.

[Schöf et al., 95] Schöf, Stefan, Sonnenschein, Michael e Wieting, Ralf, "Efficient Simulaion of THOR Nets", in *Proceedings of the 16th International Conference on Application and Theory of Petri Nets*, Giorgio de Michelis Michel Diaz (Eds.), Turin, Italy, LNCS 935, June 1995.

[Shukla et al., 91] Shukla, Ashish, Robbi, Anthony D., "A Petri Net Simulation Tool", Conference Proccedings 1991 IEEE International Conference on Systems, Man and Cybernetics, "Decision Aiding for Complex Systems, vol. 1, IEEE, New York, NY, USA, 1991.

[Silva, 85] Silva, Manuel, *Las Redes de Petri en la Automática y la Informática*, Madrid, Editorial AC, 1985.

[Venkatesh et al., 95] Venkatesh, Kurapati, Zhou, MengChu, e Caudill, Reggie J., "Discrete Event Control Design for Manufacturing Systems Via Ladder Logic Diagrams and Petri Nets: A Comparative Study", in [Zhou, 95], 1995.

[Zuberek, 91] Zuberek, W. M., "Timed Petri Nets Definitions, Properties, and Applications", *Microelectron. Reliab*., Vol. 31, No. 4, pp. 627-44, 1991.

[Zurawsky et al., 94] Zurawsky, Richard e Zhou, MengChu, "Petri Nets and Industrial Applications: A Tutorial", *IEEE Transactions on Industrial Electronics*, Vol. 41, No. 6, December 1994.