UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA

DEPARTAMENTO DE TELEMÁTICA

ANALISADOR AUTOMÁTICO DE REDE DE PETRI TEMPORIZADA PARA VALIDAÇÃO DE PROTOCOLOS DE COMUNICAÇÃO

AUTOR : Mauro Marton

ORIENTADOR: Prof. Dr. Walter da Cunha Borelli

Este exemplar corresponde à redação final da tese defendida por Mauro Marton e aprovada pela Comissão Julgadora em 29/2/1989

W. Borelli 14/3/90

Tese apresentada à Faculdade de Engenharia Elétrica como parte dos requisitos exigidos para a obtenção do Título de Mestre em Engenharia Elétrica.

FEVEREIRO - 1989

A minha esposa

Cristina

AGRADECIMENTOS

Agradeço à minha esposa Cristina pelo carinho, compreensão e apoio nas presenças e ausências.

Agradeço ao meu orientador pela sua orientação, incentivo e sobretudo pela compreensão e amizade, fatores que contribuiram para a realização deste trabalho.

Agradeço à indispensável ajuda financeira da CAPES, FAPESP e CPqD\TELEBRAS, que me deram suporte.

Gostaria também de agradecer à amiga Berenice Camargo Damasceno pelas trocas de idéias necessárias para que chegasse à conclusão desta tese e, também a Gualberto Rabay Filho pela sua colaboração na programação.

Ao meu irmão Marco pelas conversas e incentivo.

Aos amigos que me acompanharam no decorrer de meu trabalho: Magda, Graça, Roberto, Mario, Samuel, Dinho, Saulo, Felipe minha amizade e gratidão.

Aos amigos Cláudio, Sandra, Rita obrigada pelo apoio e amizade.

Meu muito obrigado a todo o pessoal amigo da Telemática: colegas, professores e funcionários.

Agradeço a todos que direta ou indiretamente me ajudaram a chegar a realizar esta tese e, que pelo espaço não mencionei nominalmente.

A minha família, meu carinho especial.

ÍNDICE PÁG.

CAPÍTULO 1

1 - Introdução

A especificação, a validação e a utilização de sistemas complexos exigem o desenvolvimento de modelos e de ferramentas que permitam ao projetista adquirir um conhecimento satisfatório dos problemas encontrados. Por exemplo para sistemas puramente seqüênciais, o modelo automata de estado finito é largamente utilizado devido a facilidade de emprego tanto para análise quanto para a síntese.

O problema se complica a partir do momento onde é necessário conceber sistemas que possam ser capazes de realizar tarefas simultâneas ou pseudo- simultâneas independentes umas das outras, com exceção de certos pontos de cooperação. Estes tipos de situações são encontradas por exemplo quando são considerados a cooperação de tarefas concorrentes ou na concepção de protocolos em rede de comunicação.

Uma rede de comunicação entre outras aplicações está presente tanto nas atuais Redes locais de computadores como estarão presente nas futuras Redes Digitais de Serviços Integrados (RDSI).

Com o objetivo de se diminuir a complexidade dos

projetos de redes, estas têm sido organizadas de forma hierárquica, em camadas ou níveis. As camadas sucessivas usam o serviço oferecido pela camada imediatamente inferior, acrescentando novas funções que são oferecidas à camada imediatamente superior na forma de um serviço mais sofisticado.

O número de camadas, o nome de cada camada e a função de cada camada pode diferir de rede para rede.

Entre cada par de camadas adjacentes existe uma interface. Esta interface define as operações (serviços) disponíveis, como acessá-los e quais os formatos e convenções usados. Enquanto as interações entre as camadas adjacentes são regidas pela interface, as interações entre os processos da mesma camada são regidas pelo protocolo desta camada.

A um determinado conjunto de interfaces e protocolos que definem as camadas da rede chamamos de arquitetura da rede.

A ISO, Organização Internacional para a Padronização (do inglês "International Organization for Standardization") definiu uma arquitetura para redes, chamada arquitetura OSI, Interconexão de Sistemas Abertos, (do inglês "Open Systems Interconection"), em sete camadas [1, 2]: Física, Enlace de Dados, Rede, Transporte, Sessão, Apresentação e Aplicação.(veja fig.1.1). Esta arquitetura em sete camadas é aceita pela CCITT (do inglês "International Telegraph and Telephone Consultative Committee"), ECMA (do inglês " European Computer Manufacturers Association) e por muitas organizações de padronização nacionais, incluindo a ANSI (do inglês " American National Standards Institute").

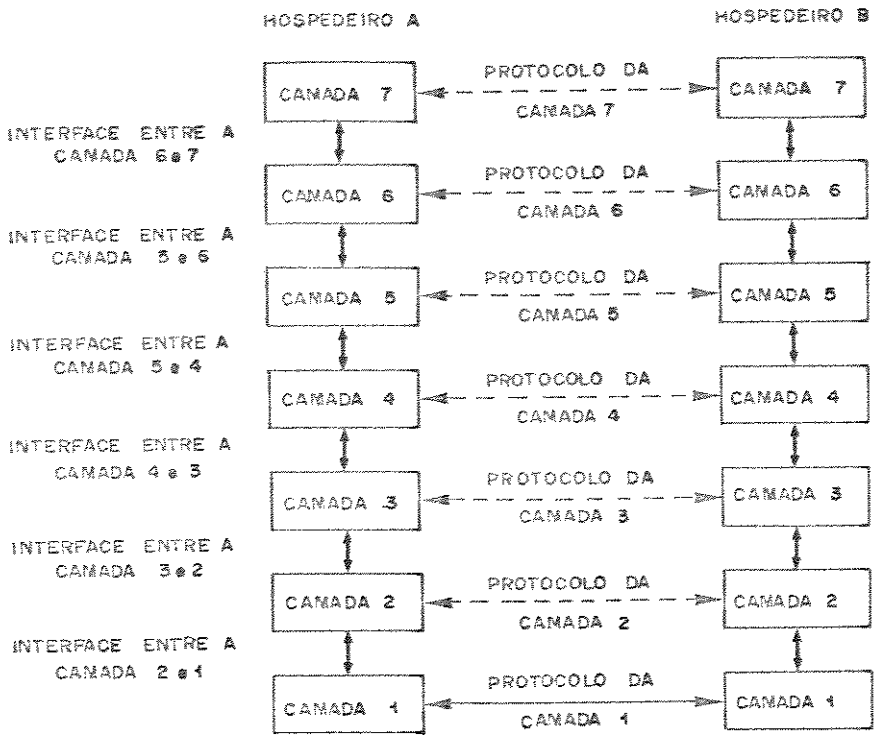

Fig.1.1 - Arquitetura para Interconexão de Sistemas Abertos da ISO.

Os sistemas com que iremos lidar são os protocolos de comunicação.

Um protocolo de comunicação é uma convenção seguida por dois ou mais participantes (nodos) para o intercâmbio eficiente de informação através de um meio de comunicação sujeito a erros [3].

No trecho onde haja erros de transmissão de mensagem, incluíndo perdas totais de mensagens, o protocolo deve ser um sistema tolerante a falhas. Para estar seguro de que suas mensagens estão chegando bem, o transmissor espera a "chegada de

reconhecimento" ou acknowledgment (ack), enviados pelo receptor.

Desta maneira as falhas de transmissão são detectadas pela não chegada do "ack" correspondente. Devido a isto, o transmissor tem que decidir em algum instante que o "ack" não chegará, e retransmitir a mensagem.

O momento em que isto é decidido é dado pelo evento chamado temporização (timeout). Note que pode haver várias causas da ausência do "ack" no momento de ocorrer o timeout, que são : o "ack" não foi enviado, foi enviado e se perdeu, ou simplesmente se atrasou. O transmissor não tem forma de distinguir entre estas possibilidades.

Está claro, então, que toda boa especificação de um protocolo deve incluir o tempo como elemento básico. A especificação deste tempo é um problema em que estamos interessados em estudar.

É indispensável que a especificação ou a descrição de um protocolo sejam precisas e exatas, totalmente ausentes de ambigüidades. Para este objetivo existem vários métodos propostos para a especificação formal de protocolos.

De uma forma geral, podemos separar os métodos propostos em três tipos principais [4] : modelos de transição, modelos baseados em linguagem de programação e modelos híbridos.

Os modelos de transição se baseiam no reconhecimento de que as entidades participantes de um protocolo têm seu comportamento regido por reações a determinados eventos, tais como a chegada de uma mensagem, uma temporização, etc... Um exemplo deste modelo é a máquina de estado finito.

Os modelos baseados em linguagem de programação não

deixam de ser um tipo de algoritmo concorrente e, portanto, podem ser especificados através de linguagem de programação. Várias linguagem já foram usadas com este propósito tais como PASCAL (estendido), PROLOG, etc..

Os métodos híbridos procuram combinar os métodos anteriores. Assim, o comportamento das entidades é descrito por uma pequena máquina de estado finita, agregada a um conjunto de especificações em alguma linguaguem de programação onde os efeitos dos eventos sobre as variáveis locais de cada entidade são descritos.

Para garantir que a especificação esteja completa no sentido de cobrir todas as possíveis situações de operação do protocolo e com isenção de erros é necessário utilizar-se de métodos formais de validação de protocolos.

A validação de protocolos pode ser entendida de várias formas : a verificação de propriedades do protocolo, a verificação de que o protocolo satisfaz a especificação do serviço e a verificação de que um determinado programa de fato implementa o protocolo corretamente.

A maior parte dos trabalhos que têm sido feitos [4], se concentra na verificação de propriedades do protocolo. Dentre as propriedades podemos citar ausência de impasses total ou parcial, ausência de loop, exclusão mútua de seções críticas, etc...

Os métodos baseados em modelos de transição fazem a validação basicamente através da geração sistemática de todos os estados possíveis do sistema. A maior dificuldade encontrada por

estes métodos é a chamada "explosão de estados" que é causada pelo grande número de combinações de transições que podem ocorrer.

Os métodos baseados em Linguagens de programação empregam técnicas usadas em verificação de programas. Estas técnicas podem, potencialmente, lidar com todos os tipos de propriedades que se deseja provar, mas a sua automação é bastante difícil.

Os métodos híbridos usam uma combinação das duas técnicas descritas acima; o uso de variáveis simbólicas, por exemplo, permite a redução do número total de estados possíveis do sistema. Estas técnicas são difíceis de automatizar completamente.

Neste trabalho o método formal de validação de protocolo de comunicação é baseado no modelo de transição. O modelo utilizado foi a Rede de Petri que é o formalismo que tem sido muito usado para modelamento e análise de protocolos [5, 6, 7, 8]. Isto se deve ao fato que desde sua criação a Rede de Petri tem sido muito usada na descrição e análise de sistemas concorrentes assíncrono que se comunicam, devido a sua simplicidade de funcionamento e a sua característica gráfica que permite uma visualização fácil dos sistemas modelados.

Alguns pacotes de Software para análise automática de Rede de Petri tal como OGIVE/OVIDE (do francês " Outil Graphique Interactif pour la Vérification des Systémes a Evolution Parallèle Decrits par Réseaux de Petri") [9, 10], SIPRO - Analisador automático de Rede de Petri [11, 12], tornam a rede de Petri ainda mais interessante para o modelamento e validação

de Protocolos de Comunicação.

Entretanto um problema surge ao se usar a rede de Petri clássica (RP)para modelar protocolos de comunicação, quando o tempo é levado em conta nas especificações, por exemplo em sistemas tolerantes a falha [13]. O funcionamento de tais sistemas, são baseados em grande parte em dispositivos de segurança cuja ação é exclusivamente dependente do tempo. Estes dispositivos são chamados de temporizadores. Os temporizadores são dispositivos de segurança cuja ação é exclusivamente dependente do tempo que pòdem iniciar em um tempo local pré definido e disparar ou reiniciar depois de transcorrido um certo tempo.

A dificuldade da rede de Petri clássica está na sua limitação em modelar parâmetros temporais, como por exemplo, o fato de um evento só ocorrer após um determinado tempo de espera suficiente para que um outro evento ocorra. Além disso o modelo é fraco para modelar prioridades entre eventos [14].

Do levantamento bibliográfico (não exaustivo) realizado sobre extensões da RP que tratam com temporizadores, o modelo que nos pareceu melhor para modelar protocolos, que utilizam largamente o uso de temporizadores é o da Rede de Petri temporizada [15, 16, 17]. Em [18] são citadas três modelos da Rede de Petri temporizada e que foram propostas na literatura(Rede de Ramchandani, Sifakis e de Merlin) . Estas extensões surgiram para modelamento de sistemas temporizados com características diferentes.

Aqui será dado um breve resumo sobre estas três extensões.

1 - Rede temporizada de Ramchandani.

Esta extensão chamada Rede de Petri temporizada (TIMED Petri Nets - TdPNs) foi proposta por Ranchandani em 1973. Neste modelo uma duração do disparo é associada a cada uma das transições da RP. Assim, o disparo da transição não é mais um evento instantâneo como na RP.

A rede temporizada de Ramchandani é um par (RP,δ) onde:

- RP : é a Rede de Petri Clássica < L, T, E, S > [20] associada a rede temporizada sem a consideração dos tempos envolvidos (RP associada);
- δ : é uma função que associa um número relativo a duração do disparo a cada transição ti ∈ T.

As transições são habilitadas do mesmo modo que em RP. O disparo de uma transição habilitada ti causa uma mudança da marcação M para a marcação M", que pode ser escrita em duas etapas:

1) Na primeira etapa tão logo a transição ti estiver habilitada há uma diminuição na marcação, isto é:

$\forall l \in L,\ M'(l) = M(l) - E(l,ti)$

onde :

M'(l) é o n.o de senhas do lugar l depois de completada a primeira etapa.

M(l) é o n.o de senhas anterior,

E(l,ti) é o peso do arco que vai do lugar l para a

transição ti.

2) a segunda etapa ocorre δ (ti) unidades de tempo mais tarde e provoca um incremento na marcação, isto é:

$\forall l \in L,\ M''(l) = M'(l) + S(l,ti)$

onde,

M"(l) é o n.o de senhas do lugar l obtido depois do disparo de ti,

S(l,ti) é o peso do arco que vai da transição ti para o lugar l.

Quando uma transição é habilitada, um disparo é iniciado a menos que o disparo de uma outra transição desabilite-a.

De acordo com a referência [18], esta rede não é apropriada para o modelamento de protocolos de comunicação, onde temporizadores são freqüentemente empregados.

2 - Rede temporizada de Sifakis.

Esta extensão (TIMED PLACE -TRANSITION NETS - TdPTNs) foi proposta por Sifakis em 1977. Neste modelo um atraso é associado a cada lugar da RP.

A rede Temporizada de Sifakis é um par <RP,D> onde :

-RP : é a RP associada.

-D : é a função que associa um atraso (Delay) para cada lugar da RP.

Sifakis em 1979 demonstrou a equivalência de seu modelo com o de Ramchandani e portanto também não é apropriado para modelo de protocolos de comunicação.

3 - Rede temporizada de Merlin.

Esta extensão (TIME PETRI NETs - TPNs) foi proposta por Merlin em 1974. neste modelo um intervalo de tempo é associado a cada uma das transições da RP.

A rede temporizada é um par <RP,Ie> onde :

-RP : é a RP associada ;

-Ie : é a função que associa um intervalo fechado [ai, bi] (0>=ai>=bi) para cada transição ti $\in$ T e é chamado de intervalo de disparo estático da transição ti.

As transições são habilitadas do mesmo modo que em RP. O disparo de uma transição ocorre apenas dentro do intervalo [ai,bi] contado a partir do momento que a transição for habilitada. O disparo de uma transição é uma ação instantânea e provoca exatamente o mesmo efeito que no modelo clássico (RP).

Segundo a referência [18] a rede de Merlin é a mais apropriada e tem sido usada para modelamento e análise de protocolos de comunicação que possuem temporizadores e/ou atrasos variáveis na troca de informação. Um pacote de software utilizando este modelo foi implementado em [19].

Assim, o modelo escolhido foi o de Merlin, pois mostrou-se ser, segundo essas referências, o mais apropriado para se implementar um pacote de software para análise e validação de protocolos com temporizadores.

No capítulo 2 são apresentadas as principais definições e propriedades da rede de Petri clássica e em seguida as da rede

de Petri temporizada. É também apresentado um método de análise de rede de Petri temporizada. Isto tudo é apresentado com o objetivo de se implementar o ARPT que é o objetivo do capítulo seguinte.

No capítulo 3 são apresentados os algoritmos implementados para a verificação das propriedades da RPT, que será usado em validação de protocolos de comunicação, e é apresentado a estrutura do programa em cardápio de opções ao usuário.

No capítulo 4, o protocolo de bit alternante [21] é usado como exemplo de utilização do ARPT e para o qual se mostra as facilidades e potencialidades de tal ferramental na análise e validação de protocolos de comunicação.

Finalmente, no capítulo 5 são apresentadas as principais conclusões do trabalho.

No apêndice 1 são apresentadas as definições gerais das estruturas de dados e as variáveis globais do programa ARPT.

No apêndice 2 são apresentados a descrição e os resultados da análise, feita pelo ARPT, do protocolo de bit alternante.

CAPÍTULO 2

Rede de Petri Temporizada (RPT) : Definições e Conceitos básicos

2.1 - Introdução

Neste capítulo é apresentado inicialmente as principais definições do modelo da Rede de Petri (RP) clássica e as suas principais propriedades [9, 11, 20, 22] tendo em vista a implementação em computador.

Em seguida é apresentado a RPT [16, 17, 25, 31] e o método de análise de RPT por enumeração de estados[17, 18, 24, 25, 28] e suas principais propriedades. Estas definições e conceitos são apresentadas com o objetivo de implementação do pacote ARPT (veja cap.3) para análise e validação de protocolos de comunicação.

2.2 Rede de Petri - Definições e sua representação

Definições Básicas :

Definição 1:

Uma rede de Petri é uma quádrupla RP = <L,T,E,S> onde :

L : é um conjunto finito não vazio de lugares (do inglês "Places") {l1,l2,.....,lnl}.

T : é um conjunto finito não vazio de transições (do

inglês "Transitions") {t1,t2,t3,....tnt}

E : é uma função "Entrada das transições" E : TxL --> N

S : é uma função "Saída das transições" S : TxL --> N

onde N é o conjunto dos números naturais.

Obs. E e S são também denominadas de função incidência direta e função incidência reversa [11] [15]. Estas funções podem ser representadas de forma matricial, o que facilita bastante a manipulação da rede em computador. Desta forma teremos as seguintes matrizes :

E : matriz "entrada da transição"

S : matriz "saida da transição"

Representação gráfica :

A RP possui uma representação gráfica que é útil para mostrar os conceitos da teoria de rede de Petri.

O grafo da RP possui dois tipos de nodos : um círculo representando um lugar e uma barra representando uma transição. Como os círculos representam lugares, os denominaremos de lugares. Similarmente, denominaremos as barras de transições.

Os nodos são ligados por arcos direto. Estes ligam um lugar l a uma transição t se somente se E(t,l) $\neq$ 0, e denominaremos este lugar l de lugar de entrada.

E os nodos ligam uma transição t a um lugar l se somente se S(t,l) $\neq$ 0, e denominaremos este lugar l de lugar de saída. Se E(t,l) $\neq$ 0 e S(t,l) $\neq$ 0, numa mesma rede, então denominaremos l de lugar realimentado [11].

O valor de E(t,l) é chamado de peso do arco de entrada e

o valor de S(t,l) é chamado de peso do arco de saída. Na representação gráfica, o peso de um arco é explicitamente marcado somente se ele é superior a 1 (um). Uma outra forma de representação não muito usada é ter tantos arcos de peso 1 quanto o peso do arco [20].

Definição 2:

A matriz "entrada da transição" (matriz E) é formada por nt linhas e nl colunas. Ela dá uma indicação da estrutura da rede quanto às informações dos lugares de entrada que incidem sobre cada transição específica e seu respectivo peso da entrada.

O elemento da linha i e coluna j (i= 1,2,...,nt; j= 1,2,...,nl) denotado por Eij contém as seguintes informações:

a) se Eij =0, o lugar lj não é lugar de entrada da transição ti.

b) se Eij $\neq$ 0, o lugar lj é um lugar de entrada de ti com peso Eij.

Definição 3:

A matriz "saida da transição" (matriz S) tem a mesma dimensão que a matriz E, isto é, com nt linhas e nl colunas. Com a diferença que ela dá uma indicação da estrutura da rede quanto às informações dos lugares de saída que são atingidos por cada transição específica e seu respectivo peso de saida.

O elemento da linha i e coluna j (i= 1,2,...,nt; j= 1,2,...,nl) denotado por Sij contém as seguintes informações:

a) se Sij =0, o lugar lj não é lugar de saída da transição ti.

b) se Sij $\neq$ 0, o lugar lj é um lugar de saída de ti com peso Sij.

Definição 4:

Uma marcação de uma rede de Petri é uma função M que associa para cada lugar um número natural que representa a

quantidade de senha ou ficha (do inglês "Token"). Na representação gráfica, a senha é representada por um ponto cheio.

M : L --> N

A marcação M pode ser representada por um vetor de dimensão nl igual ao número de lugares, M = (n1,n2,n3.....nl) onde ni ∈ N,i=1,...,l. O número de senhas de cada componente do vetor será relacionada por M(li) = ni, i=1,...,l.

A marcação inicial de uma rede é dada por M0, e representa a distribuição inicial de senhas pelos lugares da rede.

Definição 5 :

Para K >= 1 (um), uma transição t é considerada " K habilitada " para uma marcação M se e somente se :

$\forall$ l ∈ L, M(l) >= K.E(t,l) e $\exists$ l ∈ L tal que

M(l) = K.E(t,l) p/ E(t,l)≠0

Qualquer que seja K >= 1 (um), toda transição K habilitada por M é chamada simplesmente "habilitada" por M.

Definição 6:

Uma transição t é "disparável" numa marcação M, se e somente se ela é habilitada por esta marcação.

Definição 7:

O "disparo" de uma transição disparável t numa marcação M, é uma operação que conduz a marcação M a marcação M' tal que :

$\forall$ l ∈ L M'(l) = M(l) - E(t,l) + S(t,l)

Notação

$$M \xrightarrow{t} M'$$

Definição 8:

Uma seqüência de disparo (SD) é uma seqüencia de transições disparáveis à partir de Mi que conduz a uma marcação Mj tal que :

$$Mi \xrightarrow{ta} Mi+1 \xrightarrow{tb} \ldots\ldots \xrightarrow{tk} Mj$$

ou resumidamente

$$Mi \xrightarrow{SD} Mj \text{ , onde } SD = \{ta, tb, \ldots . tk\}$$

Definição 9 :

O conjunto das marcações decorrente de M0, denotado por {D(M0)}, é o conjunto das marcações Mj gerado por pelo menos uma seqüência de disparo SD disparável a partir de M0 e que conduz a Mj.

$$Mj \in \{D(M0)\} \iff \exists\ SD \text{ tal que } M0 \xrightarrow{SD} Mj$$

Definição 10:

Uma marcação M' $\in$ {D(M0)} é dita superior à uma marcação M $\in$ {D(M0)} se somente se

$\forall\ l \in L,\ M'(l) >= M(l)$ e

$\exists\ lj \in L,\ M'(lj) > M(lj)$;

onde L é o conjunto de lugares que pertencem a rede. Notação : M' > M.

Definição 11 :

A "tabela de marcação" é uma tabela contendo todas as marcações decorrentes obtidas a partir da marcação inicial. Estas marcações obtidas aplicando a definição 5, podem ser colocadas na forma matricial com o número de linhas igual a quantidade de marcações decorrentes mais uma (marcação inicial) e com nl colunas. No cruzamento da linha com as colunas indica o n.o de senhas em cada lugar da marcação que representa a linha.

Definição 12 :

O "grafo de marcações " de uma rede de Petri, para uma marcação inicial M0, é um grafo orientado tal que os nodos representam as marcações decorrente à partir de M0 e um arco liga um nó Mi a um nó Mj se e somente se existe t $\in$ T tal que

$$Mi \xrightarrow{t} Mj$$

O grafo das marcações representa um diagrama de estado do comportamento dinâmico da rede de Petri correspondente onde cada estado é representado por uma marcação decorrente. O grafo de marcação é também chamado de máquina de senha (do inglês "Token machine").

Para se implementar no computador torna-se melhor representar o grafo de marcação na forma matricial com número de linhas igual a quantidade de marcações decorrentes mais uma (marcação inicial) e com nt colunas. A linha define o estado que a rede está no momento e a coluna representa a barra disparada. No cruzamento destes indica a próxima marcação atingida.

Definição 13 :

Uma rede de Petri é considerada "1 limitada" ou simplesmente "limitada" para uma marcação inicial M0 se e somente se:

$\exists$ K $\in$ N tal que $\forall$ Mj $\in$ {D(M0)} e $\forall$ l $\in$ L, Mj(l) =<K

Se K = 1 (um), a rede de Petri é chamada segura. Se K >1 a rede é chamada "K 1 limitada" ou simplesmente "k limitada".

Definição 14 :

Uma rede de Petri, é considerada "t limitada " para uma marcação inicial M0 se e somente se:

$\exists$ K $\in$ N tal que $\forall$ Mj $\in$ {D(M0)} e $\forall$ t $\in$ T,

$\nexists$ l $\in$ L tal que Mj(l) < K.E(t,l)

Se K = 1 (um), a rede de Petri é chamada "t segura ". Se K >1 (um), a rede é chamada "K t limitada ".

Note que se a rede é l limitada então ela é t limitada, enquanto que a recíproca nem sempre é verdadeira.

Definição 15 :

Uma rede de Petri é considerada viva para uma marcação inicial M0 se e somente se para toda transição t e toda marcação Mj $\in$ {D(M0)}, existir uma seqüência de disparo SD a partir de Mj que contenha t, ou seja:

$\forall$ Mj $\in$ {D(M0)} e $\forall$ t $\in$ T, $\exists$ SD a partir de Mj que habilita e dispara t

Definição 16 :

Uma rede de Petri é considerada reiniciável para uma marcação inicial M0 se para qualquer marcação decorrente M $\in$ {D(M0)} existir uma seqüência de disparos tal que faça a rede voltar a marcação inicial M0. Isto é

$$\forall Mj \in \{D(M0)\}, \quad \exists SD / Mj \xrightarrow{SD} M0$$

Após a conceituação básica da rede de Petri, passaremos agora a um exemplo visando o modelamento de protocolos e o tratamento em computador.

A rede de Petri da fig2.1 representa a comunicação entre um transmissor e um receptor por meio de um protocolo do tipo chamada/resposta.

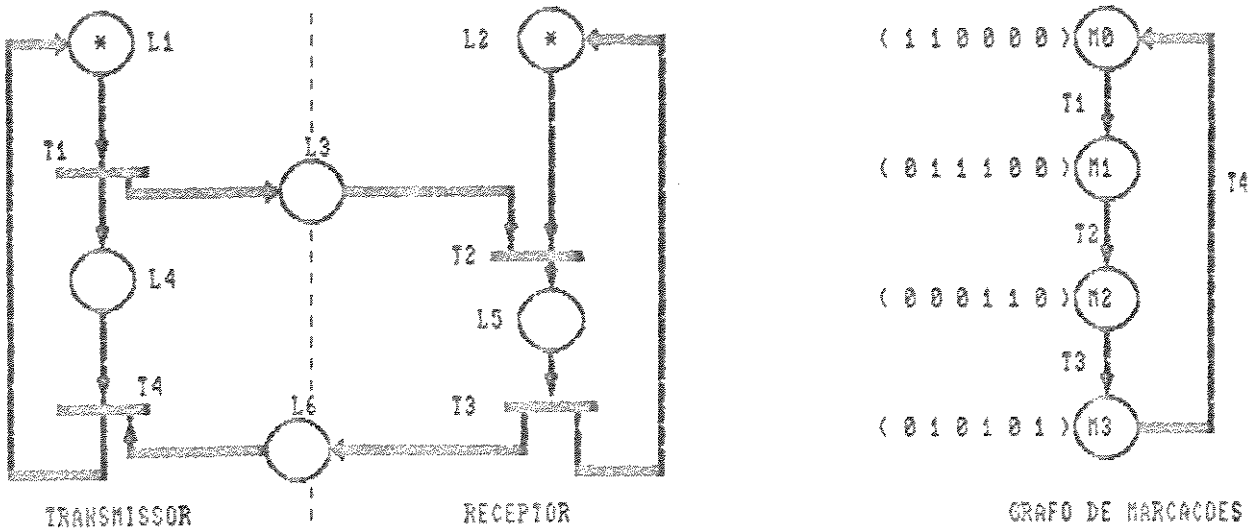

Fig.2.1 - Protocolo tipo chamada / resposta

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \qquad S = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig.2.2 - Matriz entrada de transicao e matriz saida de transicao da rede de Petri da fig.2.1 (definicao 2 e 3)

Marcacao\Lugar	l1	l2	l3	l4	l5	l6
M0	1	1	0	0	0	0
M1	0	1	1	1	0	0
M2	0	0	0	1	1	0
M3	0	1	0	1	0	1

Fig.2.3 - Tabela de marcacoes da rede de Petri da fig.2.1 com M0 = (1 1 0 0 0 0). (definicao 11)

Marcacao\Transicao	t1	t2	t3	t4
M0	1	-	-	-
M1	-	2	-	-
M2	-	-	3	-
M3	-	-	-	0

Fig.2.4 - Grafo de marcacoes da rede de Petri da fig.2.1 representada na forma matricial (definicao 12)

A figura 2.1 mostra a rede de Petri e seu grafo de marcações. Como se pode constatar a rede é limitada (segura), viva e reiniciável para a marcação M0.

2.3 Rede de Petri Temporizada (RPT)

Definições e sua representação

2.3.1 - Introdução ao modelo da RPT e ao método de enumeração de estados

No modelo da rede de Petri temporizada adotado (modelo de Merlin), um intervalo de tempo fechado (incluso os valores extremos) é associado a cada transição de uma rede de Petri clássica. Este intervalo de tempo é constituído por duas extremidades correspondente ao intervalo de tempo no qual uma transição pode disparar, a contar do instante em que foi habilitada.

Este modelo proposto para o tratamento de protocolos é o

que mais se presta para representar eventos cuja ocorrência no tempo não pode ser precisada com absoluta certeza, mas sim numa faixa de tempo possível. Tais eventos podem ser, por exemplo, consequencia de dispositivos assíncronos situados em lugares distintos e que portanto não possuem a mesma referência temporal como por exemplo os temporizadores.

O intervalo estático de tempo é definido por dois valores extremos ALFA e BETA (ALFA, BETA $\in$ Q, onde Q representa os números racionais, ALFA $\leq$ BETA e ALFA $\neq$ oo (infinito)). ALFA corresponde ao tempo mínimo que a respectiva transição deve esperar após sua habilitação para poder disparar e BETA corresponde ao tempo máximo que ela pode permanecer habilitada sem ser disparada.

O intervalo de tempo,chamado aqui de Ie, para uma transição t é um intervalo fechado representado por:

Ie(t) = [ALFA,BETA] onde 0=<ALFA=<BETA; ALFA,BETA $\in$ Q

O limite inferior ALFA é chamado tempo disparo inicial (TDI) estático (do inglês "Static Earliest Firing Time") ou simplesmente TDI estático.

O limite superior BETA é chamado tempo de disparo final (TDF) estático (do inglês "Static Latest Firing Time") ou simplesmente TDF estático.

Definição de uma rede de Petri temporizada[16]:

Uma rede de Petri Temporizada é uma dupla RPT = <RP,Ie> onde :

RP : é uma rede de Petri clássica RP =<L,T,E,S>.

Ie : é um mapeamento chamado Intervalo estático.

Ie : T --> Q x Q onde Q é o conjunto dos números racionais.

A forma geral de um estado S numa RPT é formada por um par S = (M,I) onde :

- M : é a marcação da RP associada.

- I : é a função de intervalos de disparo

A função I associa, a cada transição habilitada pela marcação M, um intervalo de tempo na qual a transição é permitida disparar. Estes intervalos de tempo são chamados de intervalos de disparo dinâmico (do inglês "Dynamic firing intervals") e correspondem ao total ou restante do intervalo no qual a respectiva transição ainda pode disparar.

A figura 2.5 apresenta um exemplo de RPT onde o estado inicial é representado pelo par S0 = (M0,I0).

Note que os lugares l1 e l6 são marcados com uma senha, M0 = l1(1),l6(1) e que a transição t1 e t6 estão habilitadas, I0 = [0,2],[2,3].

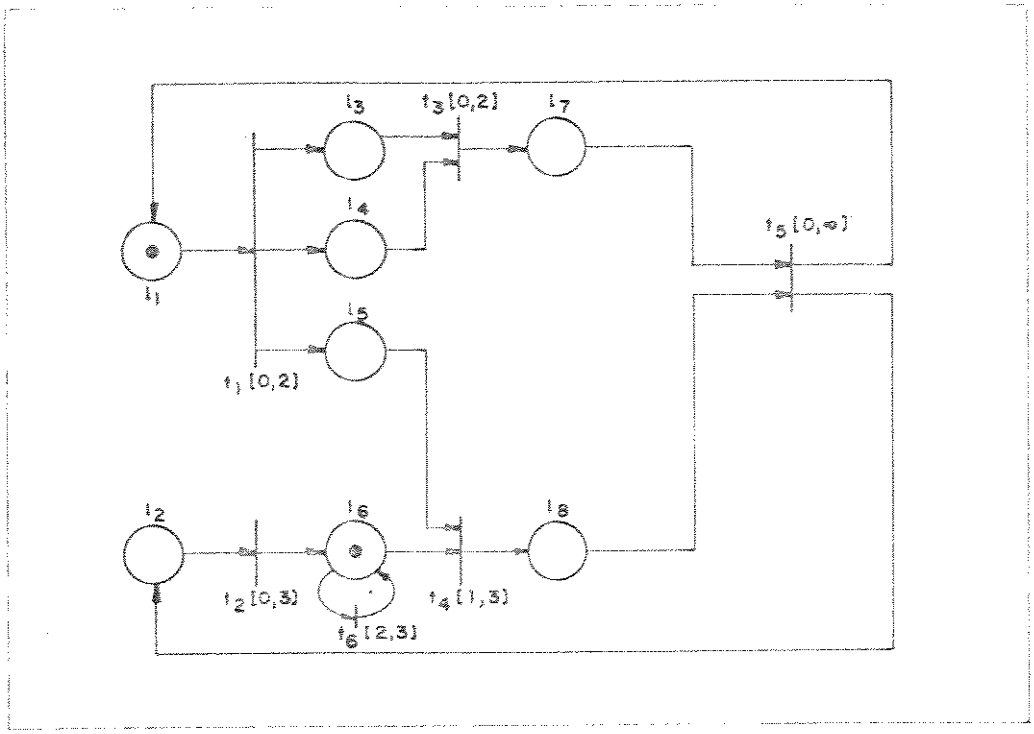

fig.2.5 - Exemplo de RPT com 8 lugares e 5 transições

2.3.1.1 Condições de disparo de uma transição (entre estados).

Assume-se que a RPT esteja em um estado S=(M,I) e que algumas transições sejam habilitadas pela marcação M.

A identificação de quais transições podem disparar é dado abaixo :

Uma transição t é disparável no instante tp do estado S conduzindo a um estado S', isto é : $S \xrightarrow{(t,tp)} S'$, se existem as seguintes condições :

a) t é habilitada pela marcação M

$$(\forall\, l) \quad (M(l) \geq E(t,l))$$

onde $l \in L$

b) tp não é menor que o TDI da transição t e não é maior que o menor dos TDF de todas as transições habilitadas ti pela marcação M.

$$TDI(t) \leq tp \leq \min_{\forall\, i \text{ habilitada}} (TDF(ti))$$

onde TDI(t) representa o TDI dinâmico da transição t e o TDF(ti) é o TDF dinâmico da transição ti.

A condição (a) é a condição usual [19] de habilitação de uma transição em RP.

A condição (b) representa o fato de uma transição habilitada não poder disparar antes de seu TDI dinâmico e deve disparar antes ou no instante de seu TDF dinâmico a menos que outra transição dispare e a desarme.

O tempo tp é contado a partir do momento em que o estado S é alcançado. Assim o tempo absoluto (tempo contado a partir do estado S0) é igual a tp acrescido do tempo necessário para alcançar o estado S.

2.3.1.2 Regras para mudança de estado [17]

Assumido que uma transição t é disparável de um estado S = (M,I), então o estado S' = (M',I') alcançado de S pelo disparo de t no instante tp é gerado como se segue:

a) a nova marcação M' é gerada como em RP[20].

$(\forall l)\ (M'(l) = M(l) - E(t,l) + S(t,l))$

b) o novo domínio I' é gerado em três passos:

1 - retira-se da expressão do domínio I os intervalos relativos às transições desabilitadas quando t dispara. Estas transições desabilitadas são aquelas habilitadas por M e não habilitadas por M*, que é definida como [$\forall l$, $M^*(l) = M(l) - E(t,l)$]. Retira-se inclusive o intervalo relativo a t.

2 - os intervalos de disparo que restam na expressão do domínio I são deslocados em direção à origem dos tempos de um valor tp e truncados, se necessários, para não incluir valores negativos.

3 - introduz-se, ordenadamente, na expressão de domínio os intervalos de disparo estático das novas transições habilitadas por M' (mas não por M*).

Para o exemplo da RPT da fig.2.5 :

O estado inicial é S0 = (M0,I0) onde :

M0 = l1(1),l6(1)

I0 = [0,2],[2,3] (t2 e t6 habilitadas)

No estado inicial, t1 pode disparar em qualquer instante no intervalo [0,2] e t6 pode disparar apenas no instante 2 (pois o TDI de t6 é igual ao TDF de t1 que é o menor dos TDF).

1)Disparando t1 num tempo tp1 dentro do intervalo [0,2], tp1 $\in$ [0,2], conduz-se ao estado S1 = (M1,I1) onde :

M1 = l3(1),l4(1),l5(1),l6(1)

I1 = [0,2],[1,3],[2-tp1,3-tp1] onde $0 =< tp1 =< 2$

(t3,t4 e t6 habilitadas)

Note que a transição t6 é remanescente quando t1

dispara, assim o intervalo foi deslocado em direção à origem do tempo de um valor igual ao instante na qual t1 disparou (tp1).

2) Disparando t6 no instante 2, conduz-se ao estado S2 = (M2,I2) onde:

M2 = l1(1),l6(1)

I2 = [0,0],[2,3] (t1 e t6 habilitadas)

Os disparos de t3, t4 e t6 em S1 e de t1 em S2 levariam aos estados (S4,S5,S6) e S7, respectivamente. Estes estados podem ser determinados conforme as regras de mudança de estado e assim por diante. Como veremos a seguir, estaremos interessados não na geração de estados, mas sim na geração do que vamos caracterizar como classes de estado.

2.3.1.3 Caracterização do comportamento da RPT.

Uma seqüência de disparos SD é definida em RPT tal qual é em RP. Uma seqüência temporizada de disparo (do inglês "firing schedule ") (SD,θ) na RPT é o par constituído de uma seqüência de disparo SD e uma seqüência de tempos θ. Isto é :

(SD,θ) = (t1,tp1).(t2,tp2).(t3,tp3),(t4,tp4).....(tn,tpn)

onde t1,t2,.....,tn são transições e tp1,tp2,.....,tpn são os instantes de disparos das transições.

Uma seqüência (SD,θ) conduz um estado S a um estado Sn, se as transições ti na seqüência SD são sucessivamente disparáveis em seus correspondentes instantes de tempo tpi da

seqüência θ.

Isto é :

$$S \xrightarrow{(SD,\theta)} Sn$$

ou

$$S \xrightarrow{(t1,tp1)} S1 \xrightarrow{(t2,tp2)} S2 \ldots\ldots Sn\text{-}1 \xrightarrow{(tn,tpn)} Sn$$

onde S1,S2,.....,Sn-1 são estados intermediários.

O conjunto de estados alcançáveis a partir do estado inicial S0 e as correspondentes seqüências temporizadas de disparos, seqüências (SD,θ), caracterizam o comportamento da RPT do mesmo modo que o conjunto de marcações decorrentes e as seqüências de disparo caracterizam o comportamento da RP [11]. Este conjunto de estados poderia ser usado com o objetivo de análise de validação do protocolo correspondente, entretanto, devido o fato de tpl poder assumir uma infinidade de valores dentro do intervalo [TDI,TDF] de cada transição habilitada ti pela marcação M , isto levaria a uma infinidade de estados (Sn1,Sn2,.....Snoo).

Um método para solução deste problema é apresentado em [17]. Este método é baseado na derivação de classes de estado, que consiste em agrupar todos os estados em um número menor de classes de estado e na enumeração dessas classes.

2.3.2 Método de enumeração de classes de estados.

Esta seção introduz o conceito de classes de estado e um método para a sua enumeração. Este método permite gerar uma

representação do comportamento dinâmico da RPT. Este comportamento será representado através de um grafo orientado onde os nós são as classes de estados e os arcos representam a relação de acesso (disparo de uma transição) entre classes. Tal grafo é equivalente ao grafo de marcações na RPT.Este grafo será chamado de grafo de classes de estado.

Ao invés de se considerar o estado alcançado a partir do estado inicial pela seqüência temporizada (SD,θ), será considerado o conjunto dos estados alcançados por todas as seqüências temporizadas com a mesma seqüência SD e diferentes seqüências de tempo θ. Este conjunto de estados será chamado de classe de estado associado com a seqüência de disparo SD.

As classes são pares C = (M,D) onde :

- M é a marcação da classe C, isto é, todos os estados em C têm a mesma marcação.
- D é o domínio de disparo da classe, definido como a união dos domínios de todos os estados em C.

A figura 2.6 mostra a passagem do conceito de "estados" para o de "classes de estados". Nesta figura Sni representa o estado alcançado do estado inicial S0 pela seqüência temporizada (SD,θi). A classe de estado C é o conjunto de todos os estados Sni (i = 1,2,..oo).

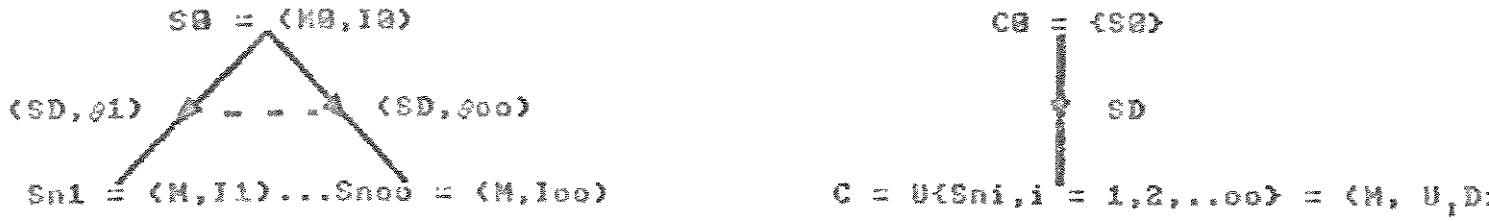

Fig.2.6 - Classe de estado C = (M,D) ; D = U,Di

2.3.2.1 - Condições de disparo de uma transição (entre classes).

Assumindo-se para a RPT que a classe de estado atual na rede seja C = (M,D), onde D é o conjunto solução de um certo sistema de inequações lineares na qual a variável de tempo xi é associada com a transição ti habilitada pela marcação M :

$$D = \{\vec{x} \;/\; A.\vec{x} =< B\}$$

onde A é uma matriz, B um vetor de constantes e $\vec{x}$ um vetor cujas componentes são as variáveis xi.

Em contraposição as expressões de domínio de disparo para estados, as expressões de domínio para as classes de estados permite introduzir relações de interdependência entre os tempos de disparo de diversas transições.

Uma transição t é disparável da classe C = (M,D) se e somente se duas condições forem sastifeitas :

a) t é habilitada pela marcação M.

$$(\forall\, l) \quad (M(l) >= E(t,l))$$

b) Assumindo que a transição t é a K-ésima transição habilitada pela marcação M, então o seguinte sistema de inequações é consistente

$$A.\vec{x} =< b$$

$$x(k) =< x(j) \quad \text{para } \forall\ j,\ j \neq k$$

onde x(k) e x(j) denotam a K-ésima e a j-ésima componente do vetor $\vec{x}$.

A condição (a) é a condição usual de habilitação de uma transição em RP. A condição (b) significa que no conjunto dos intervalos de tempos de disparos (intervalos das variáveis xi) que constitui o domínio da classe C, a transição tk pode ser disparada antes, ou no mesmo instante que qualquer outra transição habilitada tj poderia ser disparada.

2.3.2.2 - Regras para mudança de classe de estado.

Seja a classe C' = (M',D') sucessora da classe C = (M,D) pelo disparo da transição habilitada tj. A marcação M' é calculada, como em RP [20] :

$$(\forall\ l)\ (M'(l) = M(l) - E(tj,l) + S(tj,l)$$

Seja o domínio D dado da seguinte forma :

$$TDI(ti) =< xi =< TDF(ti)$$

$$TDI(tj) =< xj =< TDF(tj)$$

$$\ldots\ldots$$

$$\ldots\ldots$$

$$TDI(ty) =< xy =< TDF(ty)$$

$$xk - xn =< Hkn$$

$$xn - xk =< Hnk$$

$$\ldots\ldots\ldots$$

onde :

- ti,tj,...tk,...tn,...ty são as transições habilitadas por M,
- TDI(ti).....TDI(ty) são os seus TDI dinâmicos,
- TDF(ti).....TDF(ty) são os seus TDF dinâmicos,
- xk - xn, xn - xk, ... são as relações de interdependência entre os tempos de disparos
- Para a transição tj vale

$$TDI(tj) =< \min_{\forall\ ta\ \text{habilitada}} TDF(ta) \quad e \quad p/\ \forall\ ta,\ Haj >= 0$$

O domínio D' pode ser determinado a partir do domínio D, segundo certas regras como as descritas na referência [17].

Aqui apresentamos estas regras de maneira a facilitar a implementação do algoritmo computacional.

As regras de geração do domínio D' são :

a) Os TDI e TDF em D' são determinados para cada transição que não é desabilitada (por ex. a transição tk) pelo disparo de tj da seguinte forma :

TDI(tk) = maior entre {[0]; [TDI(tk) em D - (menor TDF de todas as variáveis em D)]; -Hjk se existir em D]}.

TDF(tk) = menor entre [(TDF(tk) em D - TDI(tj) em D); Hkj se existir em D]

b) Para todas as transições não desabilitadas, além de duas, (por ex. tk e tn) pelo disparo de tj, obtêm-se as seguintes relações :

xk - xn =< Hkn

onde :

Hkn = menor entre { Hkn se existir, TDF(tk) em D - TDI(tn) em D}

xn - xk =< Hnk

onde :

Hnk = menor entre { Hnk se existir, TDF(tn) em D -TDI(tk) em D}

c) As variáveis relativas as transições desabilitadas pelo disparo de tj não aparecerem em D'.

d) Também farão parte do domínio D' as variáveis correspondentes as transições que não eram habilitadas por M mas que são pela nova marcação M'. Cada uma dessa variável terá o seu intervalo de tempo determinado pelo TDI e TDF estáticos da transição correspondente.

e) Finalmente os TDI e os TDF dos intervalos das variáveis são subtraídos de um valor igual ao menor dos TDI (normalização dos intervalos).

A seguir usaremos as regras de (a) até (e) para gerarmos as classes de estados (fig.2.7) para a RPT da fig.2.5.

A fig.2.8 apresenta em forma de tabela o grafo das classes estados relativos a RPT da fig.2.5.

A classe inicial é C0 = [M0,D0] onde :

M0 = l1(1), l6(1)

D0 : 0=<x1=<2

2=<x6=<3

Da classe C0, t1 e t6 podem disparar, pois o menor dos TDF dinâmico é 2 (TDF de x1) e o TDI dinâmico de t6 é 2 :

a) Disparando t1 é gerada a classe C1 = [M1,D1]:

- A marcação M1 foi obtida da maneira usual, isto é da mesma forma como o seria na RP [20].

- Para a obtenção de D1 foram usados os ítens :

  - ítem (a) : novos valores para TDI(t6) e TDF(t6)

  - ítem (c) : exclusão de x1

  - ítem (d) : x3 e x4 são as novas variáveis

b) Disparando t6 é gerada a classe C2 = [M2,D2]:

- A marcação M2 foi obtida da maneira usual [20].

- Para a obtenção de D2 foram usados os ítens :

  - ítem (a) : novos valores para TDI(t1) e TDF(t1)

  - ítem (d) : x6 é a nova variável

Da classe C1, t3, t4 e t6 podem disparar.

a) Disparando t3 é gerada a classe C3 = [M3,D3]:

-Para a obtenção de D3 foram usados os ítens :

  - ítem (a) : novos valores para TDI(t4),TDF(t4),TDI(t6) e TDF(t6)

  - ítem (b) : relação "x4 - x6 =< 3" e "x6 - x4 =(2"

  - ítem (c) : exclusão de x3

b) Disparando t4 é gerada a classe C4 = [M4,D4]:

- para a obtenção de D4 foram usados os ítens :

  - ítem (a) : novos valores para TDI(t3) e TDF(t3)

- ítem (c) : exclusão de x4 e x6

c) Disparando t6 é gerada a classe C5 = [M5,D5]:

- Para a obtenção de D5 foram usados os ítens :

-ítem (a) : novos valores para TDI(t3) e TDF(t3)

-ítem (c) : exclusão de x4 e x6

-ítem (d) : x4 e x6 são as novas variáveis

De maneira semelhante obtém-se as outras classes de estados para RPT da fig.2.5.

##

Classe 0	Classe 1	Classe 2
M0= l1(1),l6(1)	M1= l3(1),l4(1),l5(1) l6(1)	M2= l1(1),l6(1)
D0: 0=<x1=<2	D1: 0=<x3=<2	D2: 0=<x1=<0
2=<x6=<3	1=<x4=<3	2=<x6=<3
	0=<x6=<3	

Classe 3	Classe 4	Classe 5
M3= l5(1),l6(1) l7(1)	M4= l3(1),l4(1),l8(1)	M5= l3(1),l4(1),l5(1) l6(1)
D3: 0=<x4=<3	D4: 0=<x3=<1	D5: 0=<x3=<2
0=<x6=<3		1=<x4=<3
x4 - x6=<3		2=<x6=<3
x6 - x4=<2		

Classe 6	Classe 7	Classe 8
M6= l7(1),l8(1)	M7= l5(1),l6(1),l7(1)	M8= l5(1),l6(1),l7(1)
D6: 0=<x5=<oo	D7: 0=<x4=<2	D8: 0=<x4=<3
	1=<x6=<2	0=<x6=<3
		x4 - x6=<1
		x6 - x4=<2

Classe 9

M9= 13(1),14(1)
15(1),16(1)

D9: 0=<x3=<0
1=<x4=<3
2=<x6=<3

Classe 10

M10= 11(1),12(1)

D10: 0=<x1=<2
0=<x2=<3

Classe 11

M11= 15(1),16(1),
17(1)

D11: 0=<x4=<2
1=<x6=<2
x4 - x6 =<1
x6 - x4 =<2

Classe 12

M11= 12(1),13(1)
14(1),15(1)

D11: 0=<x2=<3
0=<x3=<2

Classe 13

M12= 12(1),15(1)
17(1)

D12: 0=<x2=<3

Fig.2.7 - Lista das classes de estados para a RPT da fig.2.5.

##########

Classe de estado atual --> transição dispara/ Próxima classe de estado

C0 ---> t1/ C1, t6/ C2

C1 ---> t3/ C3, t4/ C4, t6/ C5

C2 ---> t1/ C5

C3 ---> t4/ C6, t6/ C7

C4 ---> t3/ C6

C5 ---> t3/ C8, t4/ C4, t6/ C9

C6 ---> t5/ C10

C7 ---> t4/ C6, t6/ C7

C8 ---> t4/ C6, t6/ C7

C9 ---> t3 / C11

C10 --> t1/ C12, t2/ C0

C11 --> t4/ C6, t6/C7

C12 --> t2/ C5, t3/ C13

C13 --> t2 / C7

Fig.2.8 - Grafo das classes de estados para a RPT da fig.2.5

##

Um ponto importante a se notar nas classes de estados é as relações de tempo entre as transições não desabilitadas como por exemplo em C8.

Tomando C8 verifica-se no grafo das classes de estado, fig.2.8, que pelo disparo de t3 na classe C5 resulta a classe C8.

A marcação da classe C8 não habilita nenhuma nova transição e também não desabilita as da classe C5 (t4 e t6), com exceção de t3. Como em C5, t3 pode disparar entre 0 e 2 e a nova origem do tempo para C8 é tomada no instante do disparo da transição t3, então qualquer que seja o instante θ deste disparo, restará para t4 e t6 em C8, um intervalo de :

$$1 - \theta =< x4 =< 3 - \theta \quad (a) \quad e$$

$$2 - \theta =< x6 =< 3 - \theta \quad (b), \text{ respectivamente.}$$

Como θ pode assumir valores entre 0 e 2, faremos inicialmente uma análise para alguns valores de θ, por exemplo, $\theta = 0,1,2$.

Para $\theta = 0$, o intervalo de t4 e t6 na classe C8 será de :

$1 =< x4 =< 3$ e $2 =< x6 =< 3$, respectivamente.

Para $\theta = 1$, o intervalo de t4 e t6 na classe C8 será de :

$0 =< x4 =< 2$ e $1 =< x6 =< 2$, respectivamente.

Para $\theta = 2$, o intervalo de t4 e t6 na classe C8 será de :

$-1 =< x4 =< 1$ e $0 =< x6 =< 1$, respectivamente.

Como não existe tempo negativo, o intervalo de t4 será $0 =< x4 =< 1$.

Generalizando, para θ variando entre 0 e 2, temos que, o

lado esquerdo da expressão (a) irá variar entre -1 e 1. Como dependemos do instante do disparo de t3, tomamos o menor valor numérico não negativo de (1 - θ) ou seja o valor zero (0). Em relação ao lado direito da expressão (a), esta irá variar entre 1 e 3, neste caso, tomamos o maior valor de (3 - θ), ou seja o valor 3.

De maneira análoga para a expressão (b), o lado esquerdo da expressão, tem valor mínimo igual a 0 e o lado direito, o valor máximo igual a 3.

Neste exemplo, podemos então notar que o intervalo de t4 em C8 será $0 =< x4 =< 3$ e o intervalo de t6 em C8 será $0 =< x6 =< 3$.

Todavia, podemos ver em D5 de C5, que a seguinte relação entre os instantes inicial e final de disparo de t4 e t6 é dada por $x4 - x6 =< 1$ e $x6 - x4 =< 2$.

Gerando os novos intervalos de disparo dinâmico para t4 e t6 em C8 obtêm - se : $0 =< x4 =< 3$ e $0 =< x6 =<3$, como exemplificado acima, e agora a relação entre os instantes de disparo final e inicial é dado por $x4 - x6 =< 3$ e $x6 - x4 =<3$.

A relação em C8 é menos apertada que a existente em C5, o disparo de uma transição não deve permitir que a relação entre os intervalos seja modificada para as transições que não são influenciadas (habilitadas, desabilitadas ou reabilitadas) por este disparo seja modificado.

Isto resulta que a informação fornecida pelos intervalos de tempo dinâmicos associados a transição t4 e t6 em C8 é insuficiente para descrever a evolução do sistema. O domínio de disparo de C8 para levar em conta as restrições implicítas

existentes em D5 de C5 deve ser tal que as restrições sejam explicitamente mantidas ou seja devemos acrescentar à C8 as relações x4 - x6 =<1 e x6 - x4 =<2.

Para uma dada classe de estado inicial C0 = [M0,D0] um grafo de classes de estado pode ser construída usando a regra de geração de classes.

Este grafo de classes pode ser usado para representar o comportamento da RPT (e assim o protocolo desejado). O grafo das classes de estado permite checar as propriedades que caracterizam o comportamento representado pela RPT. Para se implementar no computador torna-se melhor representar o grafo de classes de estados na forma matricial, semelhante ao grafo de marcações em RP (definição 12), com números de linhas igual a quantidade de classes de estados decorrentes e com nt colunas.

A linha define o estado em que a rede está no momento e a coluna representa a barra disparada e o cruzamento destes indica a próxima classe de estados atingida.

Como na definição 11 em RP será necessário gerar uma tabela para armazenar as classes de estados. Esta tabela será chamada de lista de classes de estados que será composta por três matrizes. A primeira matriz será a tabela de marcação em RP, armazenando as marcações M das classes de estado. A segunda matriz, armazenará os intervalos de tempo dinâmico do domínio de disparo da classe e a terceira matriz, armazenará as relações de interdependência entre as variáveis que associam o tempo das transições não desabilitadas.

Não entraremos em detalhe sobre a forma destas tabelas pois como veremos no capítulo 3, a melhor forma de armazenar

estas tabelas é na forma de alocação dinâmica de memória (apontadores).

2.4 Propriedades da RPT

2.4.1 - Introdução

Esta seção destinada as propriedades da Rede de Petri temporizada será dividida em duas partes.

Na primeira parte serão apresentadas as propriedades inerentes a RPT (de 1 a 4).

Na segunda parte serão definidas as propriedades específicas da RPT similares às da RP clássica tal como vivacidade, reiniciabilidade e limitabilidade (de 5 a 7). Um destaque especial será dado a propriedade de limitabilidade, devido apresentar algumas restrições na sua aplicabilidade.

2.4.2 - Propriedades inerentes a RPT [17]

Propriedade 1 :

O conjunto das marcações decorrente de uma RPT é um subconjunto do conjunto de marcações decorrentes da RP clássica associada. A RP clássica associada é a RPT sem os intervalos de tempo associados as transições.

Isto provem do fato que toda transição habilitada na RPT mantem a mesma restrição lógica para o disparo de uma transição, mas com adição de restrições temporais.

Propriedade 2:

Dependendo dos intervalos de tempo, uma mesma marcação pode aparecer em diferentes estados. Isto vem do fato que uma marcação na RPT não define um estado, devido ser considerado o aspecto do tempo.

Propriedade 3:

A RP clássica é um caso particular da RPT. Isto próvem do fato de que na RP clássica, se mais de uma transição é habilitada simultâneamente por uma marcação, os disparos delas não são submetidos à alguma ordem prèvia, não existe prioridade entre as transições.

Por outro lado, em relação a RPT, o fato de juntar um intervalo de tempo permíssivel para o disparo das transições pode introduzir prioridades quando duas ou mais transições são sensibilizadas. É necessário e suficiente num estado onde duas transições ti e tj são habilitadas, que BETAi seja menor que ALFAj para que ti seja prioritária em relação a tj. De fato, se ti deve ser disparada até BETAi e tj só pode disparar a partir de ALFAj, então tj não pode ser disparada antes de ti.

Sendo assim, existem algumas maneiras de descrever o funcionamento da RP clássica através do seu modelo em RPT.

Uma delas é mais intuitiva e é de associar a todas as transições, intervalos de tempos correspondente a hipótese implícitas sobre os instantes de tempo na RP clássica isto é, intervalos [0,oo(infinito)].

Uma outra maneira é associar as todas transições, intervalos do tipo [a,b] onde a=<b. Pois este tipo de intervalo

qualquer prioridade entre as transições. É possível enunciar outros tipos de transformações equivalente como mostrado em [17].

Propriedade 4 :

Toda RPT segundo o modelo de Ramchandani pode ser convertida na RPT segundo o modelo de Merlin, que é o que adotamos, mantendo as mesmas seqüências de disparos.

Para isto é suficiente trocar todas as transições da Rede de Ramchandani por um módulo contendo na ordem, uma transição (início), um lugar e uma transição (fim) como mostra a figura 2.9.

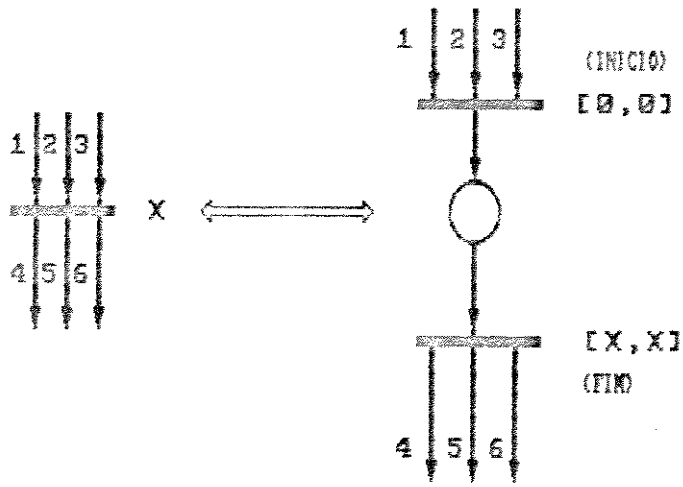

Fig.2.9 - Conversao da rede de Ramchandani para a de Merlin

Uma outra possibilidade para representar a RPT de Ramchandani pela de Merlin, consiste para todas as transições cujo lugar de entrada tenha apenas um arco de saída, de fazer a substituíção do parâmetro temporal x por um intervalo de tempo

[x,x], sem modificar o gráfico. Para as outras transições a substituição deve ser feita como acima. veja fig.2.10.

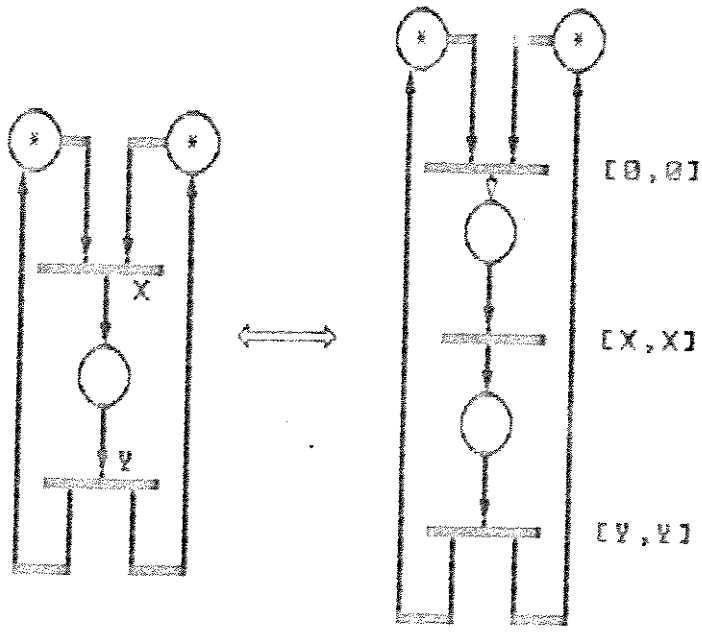

Fig.2.10 - Outra possibilidade de conversão da rede de Ramchandani para a rede de Merlin

2.4.3 - Propriedades específicas da RPT.

Propriedade 5 (Reiniciabilidade):

Uma RPT é chamada reiniciável a partir de uma classe de estado inicial C0, se somente se qualquer que seja a classe Ci pertencente a {D(C0)}, existir uma seqüência de disparo de transições SD a partir de Ci tal que faça a rede voltar à classe inicial ou seja :

$$\forall\ Ci \in \{D(C0)\},\quad \exists\ SD/\ Ci \xrightarrow{SD} C0$$

Esta propriedade se traduz na capacidade do sistema modelado de retornar a seu estado inicial após ter executado uma ou mais tarefas.

Propriedade 6 (Vivacidade)

Uma RPT é viva para um estado inicial C0 se e somente se qualquer que seja o estado decorrente Ci pertencente a {D(C0)} e qualquer que seja a transição t $\in$ T, existir uma seqüência disparável de transições SD a partir de Ci que contenha t, ou seja :

$\forall$ Ci $\in$ {D(C0)} e $\forall$ t $\in$ T, $\exists$ SD a partir de Ci que habilita e dispara t

Um ponto a se notar é que não há nenhuma relação entre vivacidade na RPT e na sua RP clássica associada. Esta propriedade é ilustrada na fig.2.11.

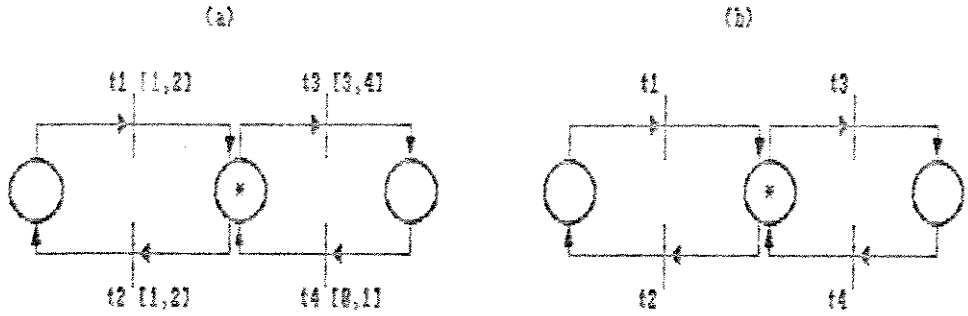

Fig.2.11 - Uma RPT (a) nao viva com a sua Rp (b) classica associada.

As transições t3 e t4 nunca disparam no modelo da RPT, enquanto que na sua RP associada todas as transições disparam.

A característica da propriedade de vivacidade é muito importante pois assegura duas condições :

1) Assegura que a rede está sem impasse (do inglês "deadlock"). Um impasse corresponde a uma marcação onde nenhuma transição é diparável, o que significa evidência de um defeito do sistema representado.

2) Assegura igualmente que toda parte do sistema

representado é acessível.

Propriedade 7 (limitabilidade):

O conceito de limitabilidade para a RPT é o mesmo usado para a RP clássica isto é, o número de senhas em qualquer lugar e para qualquer marcação é limitado. Esta propriedade foi demonstrada como indecídivel [28], mas felizmente, condições suficientes podem ser formuladas, que são discutidas a seguir.

As condições necessárias e suficientes dada em [30] para a limitabilidade da RP clássica fornece uma condição suficiente mas não necessária para a limitabilidade na RPT isto é, se a RP associada a RPT é limitada, a RPT também será, porém se a RP associada for não limitada, não podemos afirmar nada sobre a RPT.

Esta condição tem permitido determinar a limitabilidade para um grande número de RPT, porém tem se provado fraca para o uso na nossas aplicações [14].

As seguintes condições suficientes [17] são mais fortes:

Uma RPT é limitada se nenhum par de classes de estado C = (M,D) e C' = (M',D') decorrentes de C0 for tal que

i) C' é decorrente de C

ii) M' > M (definição 10 da RP)

iii) D' =D;

A classe C' que satisfizer estas condições chamaremos de classe superior. e denotaremos por C' > C.

Como antes, estas condições são suficientes mas não necessárias ou seja, pode se encontrar uma RPT limitada que tenha esses pares de classes.

Felizmente, para o nosso tipo de RPT que é t segura e com peso dos arcos igual a 1, estas condições são suficientes e necessárias para a rede ser limitada.

O tipo de RPT que usaremos é t segura pois consideramos que há apenas uma temporização em cada transição, ou seja equivale a associar a cada transição t um lugar com senha que seja lugar de entrada e de saída da transição t. Veja fig.2.12.

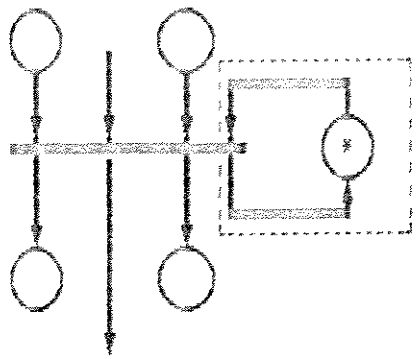

Fig.2.12 - Forma de garantir que a rede será t segura.

Uma rede não limitada traduz em geral um erro de descrição pois há uma acumulação de senhas num lugar e portanto a implementação do sistema não será possível.

2.5 - Conclusão

Nós mostramos a noção de estado na RPT, que é definida por uma marcação M como na RP clássica acrescida de uma função de intervalos de disparos I.

Devido a dificuldade em se manipular estados foi apresentado um método de análise de RPT por enumeração de estados que consiste em agrupar estados em classes, que é definida por uma marcação M como na RP acrescida de um domínio de disparo D. O método é aplicado em um exemplo para melhor compreensão.

Em relação as propriedades inerentes foram mostrado que a RP clássica , a RPT de Ramchandani e a de Sifakis podem ser descritas por intermédio de uma RPT de Merlin. Nesta seção foi também definido as propriedades de reiniciabilidade, vivacidade e limitabilidade.

Em relação à propriedade de limitabilidade apresentamos para a RPT condições suficientes e necessárias para uma RPT t segura e com peso do arcos igual a 1 (um), seja limitada. Isto do ponto de vista prático nos pareceu satisfatório para o exemplo prático que analisamos [13, 14, 32].

CAPÍTULO 3

Analisador Automático de Rede de Petri Temporizada (ARPT)

3.1 - Introdução

Neste capítulo é apresentado o Analisador Automático de Rede de Petri Temporizada (ARPT) para analisar protocolos de comunicação modelados em RPT.

O ARPT é um pacote de software que foi desenvolvido em linguagem PASCAL (Turbo - versão 3.01), e é executado em microcomputador tipo PC e utiliza alocação dinâmica de memória (apontadores).

As definições gerais das estruturas de dados e as variáveis globais do programa estão apresentadas no Anexo 1.

Neste capítulo são apresentados os algoritmos implementados para a verificarão das propriedades da RPT, e por último é apresentado a estrutura do programa em cardápios de opções ao usuário.

3.2 - Algoritmos para teste das propriedades da RPT.

3.2.1 - Limitabilidade

Em uma rede limitada o número de senhas em qualquer lugar, para qualquer marcação das classes de estado, não deve ultrapassar o valor "n" estabelecido a priori.

A técnica usada para analisar as classes de estado neste aspecto é a baseada na árvore de marcações acessíveis (reachability tree) proposta por Peterson [20].

Usando esta técnica vamos gerar uma árvore de classes de estado acessíveis. Nesta árvore, as classes de estado representam os nós da árvore. Os nós são interligados por arcos orientados rotulados com a transição cujo disparo levou àquele nó.

Esta técnica apresenta procedimentos para lidar com a explosão de estados que ocorre quando a rede é não limitada. Esta técnica visa reduzir a geração do número de classes de estado do grafo de classes.

Antes de apresentarmos o algoritmo para a geração da árvore de classes de estado, é conveniente apresentarmos algumas definições.

Nó fronteira : é cada novo nó gerado pelo disparo de uma transição.

Nó terminal : é um nó onde nenhuma transição é habilitada, este nó representa uma classe morta.

Nó duplicado : é um nó que já existe na rede e portanto

seu sucessores não precisam ser analisados.

Nó interior : é um nó que já foi gerado e analisado.

Nó de classe superior é um nó que representa uma classe superior (cap.2, propriedade 7). Quando este tipo de nó aparece significa que a rede é não limitada, ou seja existe um número infinito de classes de estado a partir desta classe.

Para limitar o número de classes de estado, uma alternativa é usar um símbolo especial, w ,que representará este número infinito de classes de estado. Este símbolo especial, w , será colocado em todos os lugares para os quais os números de senhas cresçam indefinidamente.

Este símbolo w é suficientemente grande, e para qualquer constante a, define-se :

$$w + a = w$$

$$w - a = w$$

$$a < w$$

$$w =< w$$

Estas são as operações necessárias para a construção da árvore no caso da rede ser não limitada.

Uma outra definição necessária é a de ramo ascendente. O ramo ascendente de uma classe Ci é o conjunto de todas as classes da árvore, que a partir de uma seqüência de disparo conduz a Ci. Por exemplo o ramo ascendente da classe de estado C11 da RPT da fig.2.5 do Cap.2 (fig.2.8) é { C9, C5, C2, C1, C0 }.

3.2.1.1 - Algoritmo de geração da árvore de classes acessíveis

O algoritmo começa com uma classe de estado inicial que é a raiz da árvore e um nó fronteira. A medida que aparece o nó fronteira, eles são processados pelo algoritmo.

Seja X um nó fronteira a ser processado.

1. Se nenhuma transição é habilitada na classe Cx, então X é um nó terminal.

2. Para toda transição t ∈ T que é disparável em Cx é criado um novo nó Z na árvore de classes acessíveis de acordo com as regras para mudança de classes de estado (seção 2.3.2.2), acrescida da seguintes condições.

(a) Se Mx(li) = w, então Mz(li) =w

(b) Se existe um nó Y que pertence ao ramo ascendente de Z e Cz > Cy, então para os lugares i onde Mz(li) > My(li), faz-se Mz(li) = w.

3. Para cada novo nó Z criado é verificado se existe outro nó Y na árvore com a mesma classe, isto é Cz = Cy, então o nó Z é um nó duplicado.

A cada nó Z gerado, o nó X é redefinido como nó interior.

Quando todos os nós forem classificados como terminal, duplicado ou interior, o algoritmo para.

3.2.1.2 - Implementação do algoritmo.

Quando a rede é não limitada, o número de classes é infinito. Assim, não seria possível calcular todas as classes acessíveis desta rede, em um computador. Desse modo, é usado a técnica da árvore de classes acessíveis (seção 3.2.1.1) para gerar o grafo de classes de estado.

O fluxograma do algoritmo implementado é mostrado na fig.3.1. Este fluxograma permite além da verificação da propriedade de limitabilidade, a construção simultânea da lista e do grafo de classes de estado.

3.2.2 - Reiniciabilidade

Uma RPT é reiniciável a partir de uma classe de estado inicial C0, se somente se qualquer que seja a classe Ci pertencente a {C0}, existir uma seqüência de disparo de transições, a partir de Ci, tal que faça a rede voltar a classe inicial.

Um método para verificar esta propriedade é implementado com o auxílio do grafo de classes de estado.

O método consiste em encontrar todas as classes de estado que a partir de uma seqüência de disparo conduz a C0.

Para isso, verifica-se no grafo de classes de estado, todas as classes Ci que com um disparo de transição conduzem a

C0, depois todas as Cj que com um disparo de transição conduzem as Ci e assim sucessivamente até que se tenha verificado todas as classes do grafo.

Utilizando esse procedimento, pode-se verificar a propriedade de reiniciabilidade da rede.

Como exemplo, pode se utilizar o grafo de classes de estado (fig.2.8) da RPT da fig.2.5.

Utilizando o algoritmo acima temos:

1) Classes que conduzem a C0 : C10

2) Classes que conduzem a C10 : C6

3) Classes que conduzem a C6 : C11, C8, C7, C4, C3.

4) Classes que conduzem a C11, C8, C7, C4, C3 : C13, C9, C5, C1.

5)Classes que conduzem a C13, C9, C5, C1 : C12, C2.

Como todas as classes do grafo possuem uma seqüência de disparo que conduz a C0, então a rede é reiniciável.

A fig.3.2 mostra o fluxograma do algoritmo para a verificação dessa propriedade utilizando-se o grafo de classes.

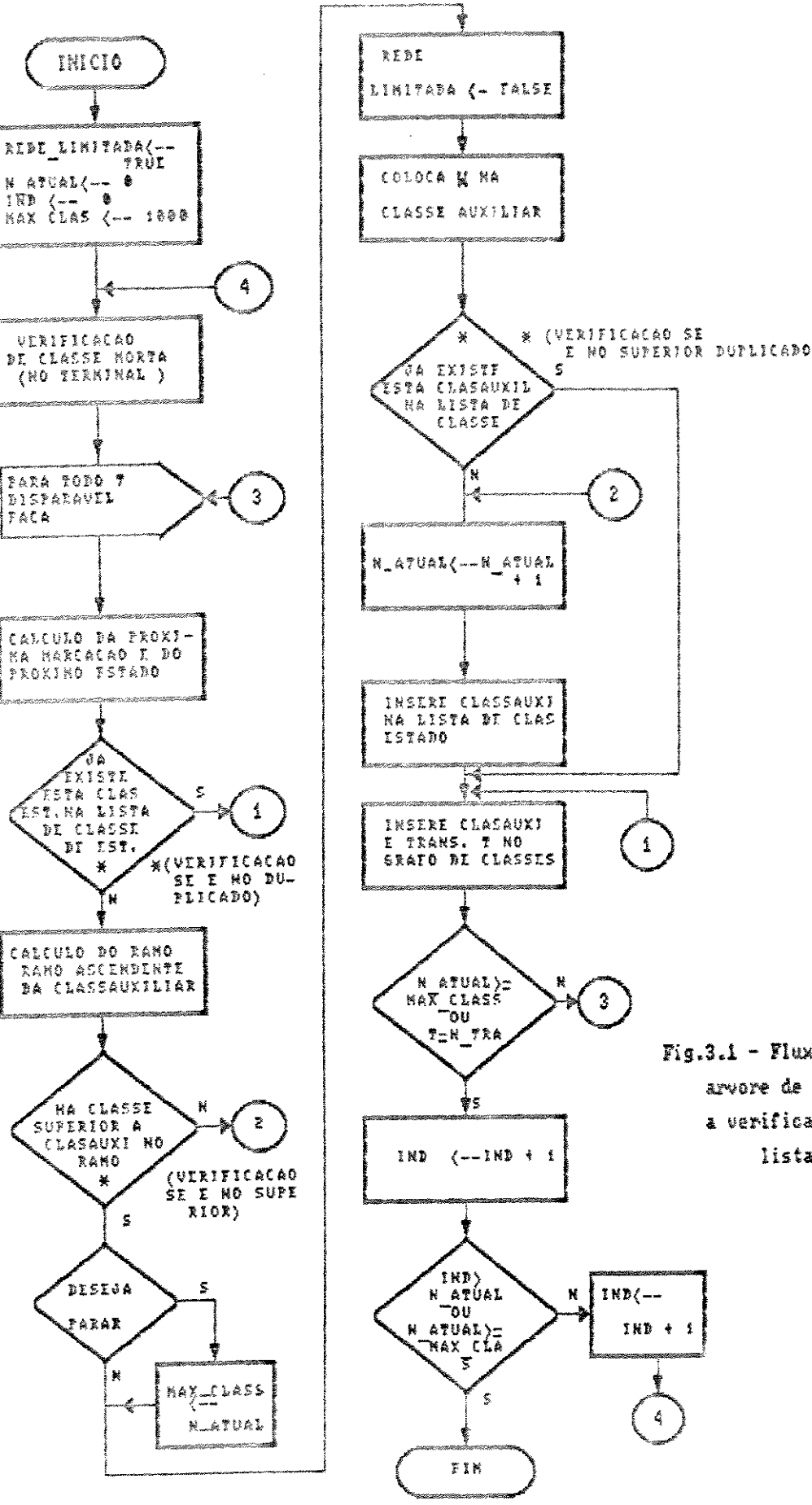

Fig.3.1 - Fluxograma do algoritmo para gerar a arvore de classes de estado acessiveis contendo a verificacao da limitabilidade e construcao da lista e do grafo de classes de estado

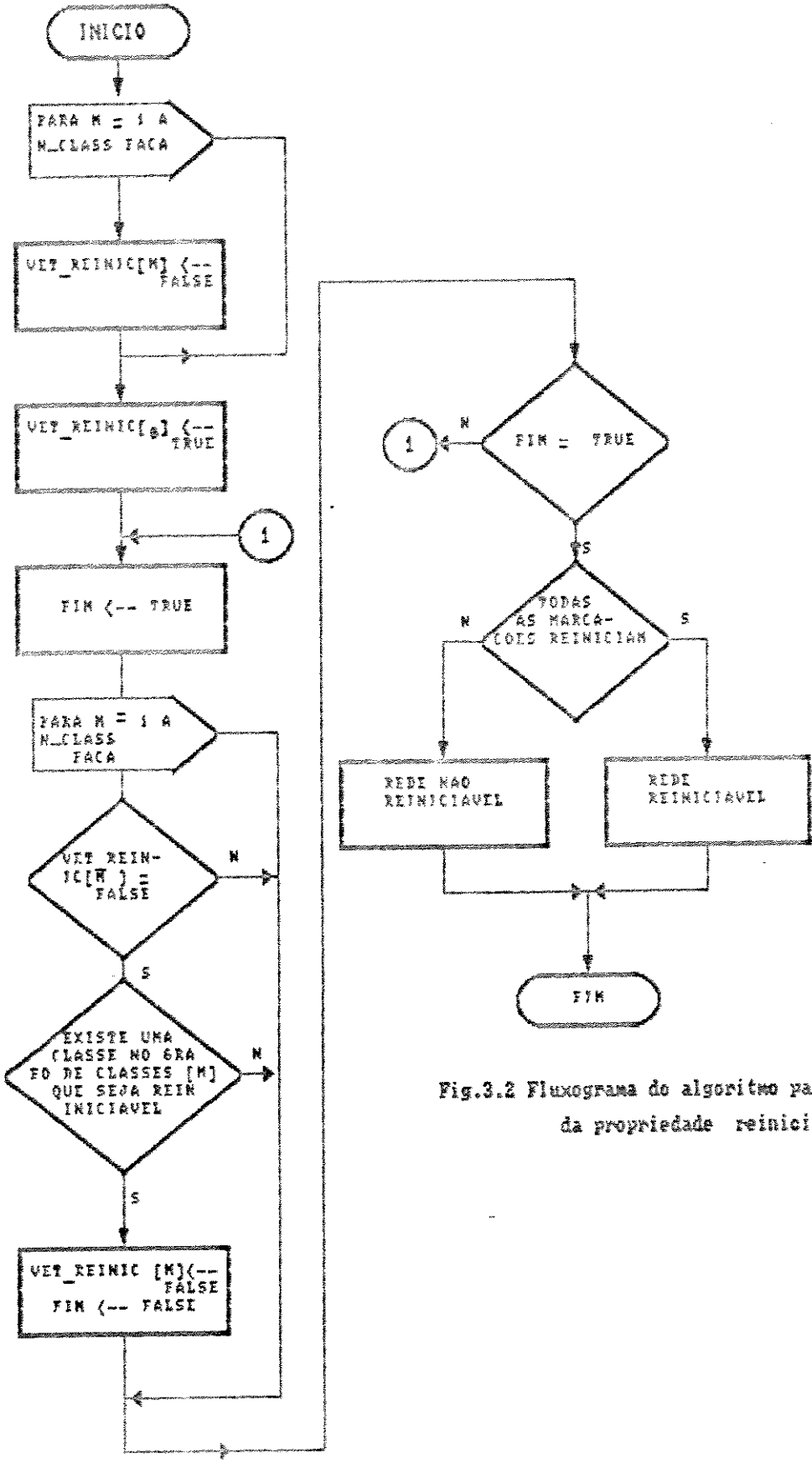

Fig.3.2 Fluxograma do algoritmo para a verificacao da propriedade reiniciabilidade

3.2.3 - Vivacidade

Para a verificação dessa propriedade, as RPT são divididas em dois casos : redes reiniciáveis e não reiniciáveis.

a) Redes de Petri temporizadas reiniciáveis.

Quando a RPT for reiniciável e todas as transições forem disparáveis, a rede é viva.

Isso pode ser verificada com o auxílio do grafo de classes de estado. Tomemos como exemplo a RPT da fig.2.5.

Na seção anterior verificou-se que esta rede é reiniciável para a dada classe inicial C0, portanto à partir de qualquer classe Ci $\in$ { C0 }, existe uma seqüência de disparo que conduz a qualquer Cj $\in$ { C0 }.

Assim sendo, para verificar a propriedade de vivacidade desta rede, basta verificar se todas as transições pertencentes a RPT são disparáveis.

O conjunto de todas as transições da fig.2.5 é {t1, t2, t3, t4, t5, t6}.

Na fig.2.8, pode-se verificar que na classe C0, as transições t1 e t6 disparam para atingir C1 e C2, respectivamente, e na classe C1 as transições t3 e t4 disparam para atingir C3 e C4, respectivamente.

Repetindo-se este procedimento, pode-se encontrar o conjunto de transições que são disparáveis pelo menos uma vez em qualquer Ci $\in$ {C0}. No nosso exemplo, verifica-se que todas as transições que pertencem a RPT disparam, portanto esta rede é viva.

b) Redes de Petri temporizadas não reiniciável

Para a análise das redes não reiniciáveis temos que apresentar algumas definições [9].

Definição 1 : Grafo de classes de estado fortemente conexo.

Um grafo de classes de estado de uma RPT com classe inicial C0 é chamado fortemente conexo se existir, para duas classes quaisquer i e j consideradas nessa ordem, uma seqüência de disparo com início em i e com término em j.

Uma RPT com um grafo de classes de estado fortemente conexo satisfaz a condição de reiniciabilidade.

Como,também queremos lidar com as RPT não reiniciáveis, então vamos subdividir o grafo de classes de estado de uma RPT não reiniciável em dois ou mais subgrafos fortemente conexos. Estes subgrafos fortemente conexos serão chamados de componentes fortemente conexas (CFC).

Para se obter a componente fortemente conexa a qual pertence a classe Ci é feito a interseção do conjunto de sucessoras e antecessoras da classe Ci.

Definição 2 : Uma componente fortemente conexa é isolada se para toda a classe que pertence a esta componente, o conjunto de sucessoras está incluido no conjunto de predecessoras.

Teorema 1 [9]: Uma RPT com uma classe inicial C0 é viva se e somente se cada transição da rede é ao menos uma vez disparada a partir das classes de cada uma das componentes fortementes conexas isoladas do grafo de classes de estado.

Demonstração :

A toda classe Ci $\in$ {C0} e não pertencente a uma componente fortemente conexa isolada (CFCI) é possível através de uma seqüência de disparo conduzir Ci a uma classe pertencente a uma CFCI.

Se todas as transições da rede é ao menos uma vez disparada nesta CFCI, então toda transição é disparável a partir de Ci, portanto a rede é viva.

Um exemplo de grafo de classes de estado que possui 4 transições é apresentado na fig.3.3.

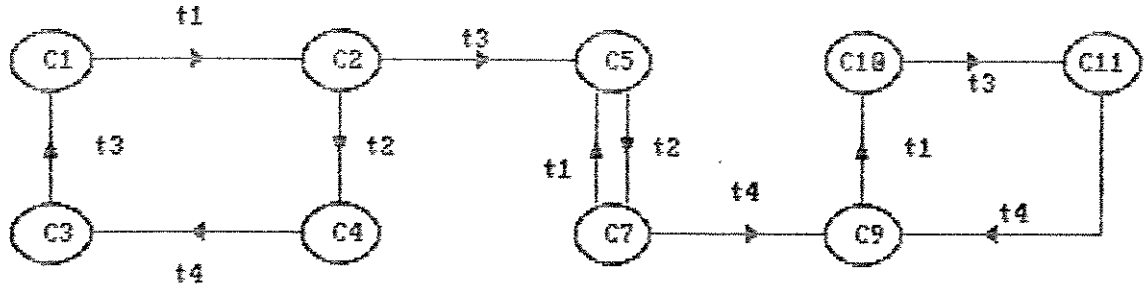

FIG.3.3 - Grafo de classes de estado

As componentes fortemente conexas são três :

CFC1 = { C1, C2, C3, C4 }

CFC2 = { C5, C7 }

CFC3 = { C9, C10, C11 }

Destas componentes apenas CFC3 é uma componente fortemente conexa isolada, pois o conjunto de sucessoras está incluido no conjunto de predecessoras.

Como t2 não dispara em CFC3, a RPT deste exemplo é não viva.

A fig.3.4 mostra o fluxograma do algoritmo implementado para verificar a propriedade de vivacidade.

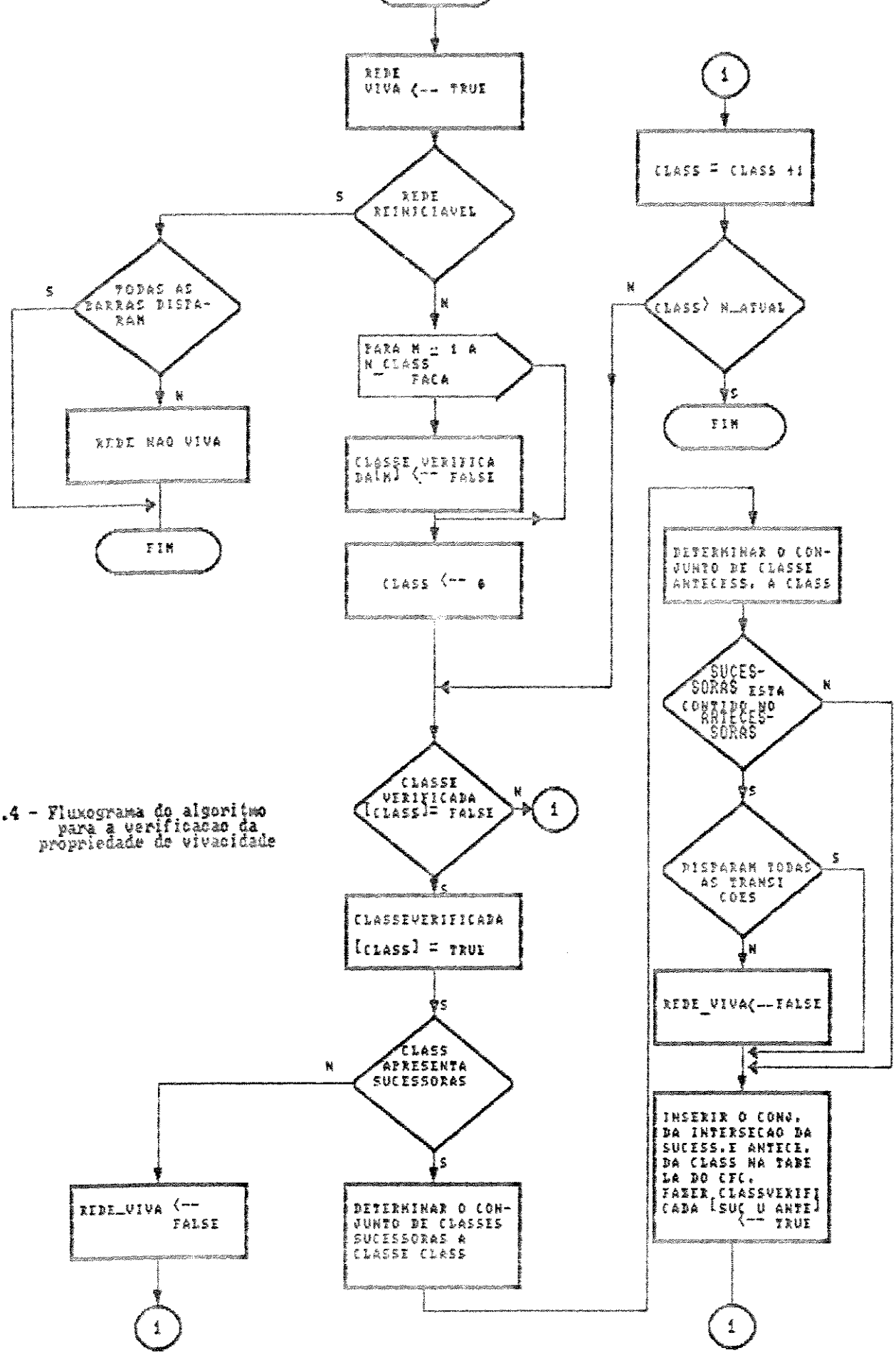

Fig.3.4 - Fluxograma do algoritmo para a verificacao da propriedade de vivacidade

3.3 - Estrutura do programa

3.3.1 - Concepção

O ARPT é um pacote de software para auxiliar na validação de protocolos de comunicação. Para isso ele analisa as propriedades específicas da RPT correspondente ao protocolo e fornece algumas informações (diagnósticos) para a análise.

O ARPT aceita como dados de entrada, o protocolo modelado em RPT. Como dados de saída, ele fornece os resultados da análise das propriedades da RPT, o grafo das classes de estado, a lista das classes de estados e informações para auxiliar na análise da rede como as componentes fortemente conexas e classes antecessoras a uma classe dada.

O ARPT apresenta um cardápio de opção composto por 6 ítens. Algumas destes ítens geram subcardápio de opções.

A fig.3.5 mostra o cardápio de opções do ARPT em diagrama de bloco. As interações entre as opções são feitas por intermédio de arquivos de interfaceamento criado pelo próprio programa, cabendo ao usuário definir os nomes destes arquivos.

As opções descrição, alteração e documentação da rede possibilita de um modo interativo descrever, alterar e documentar respectivamente uma ou mais RPT.

A opção execução gera a lista e o grafo das classes de estado, as componentes fortemente conexas e analisa as propriedades específicas da RPT (limitabilidade, vivacidade e

reiniciabilidade).

A opção documentação de saída faz a documentação dos resultados da análise feito pela opção execução, sendo que o usuário pode especificar o nível de detalhamento desejado. A seguir é apresentado com mais detalhes cada opção.

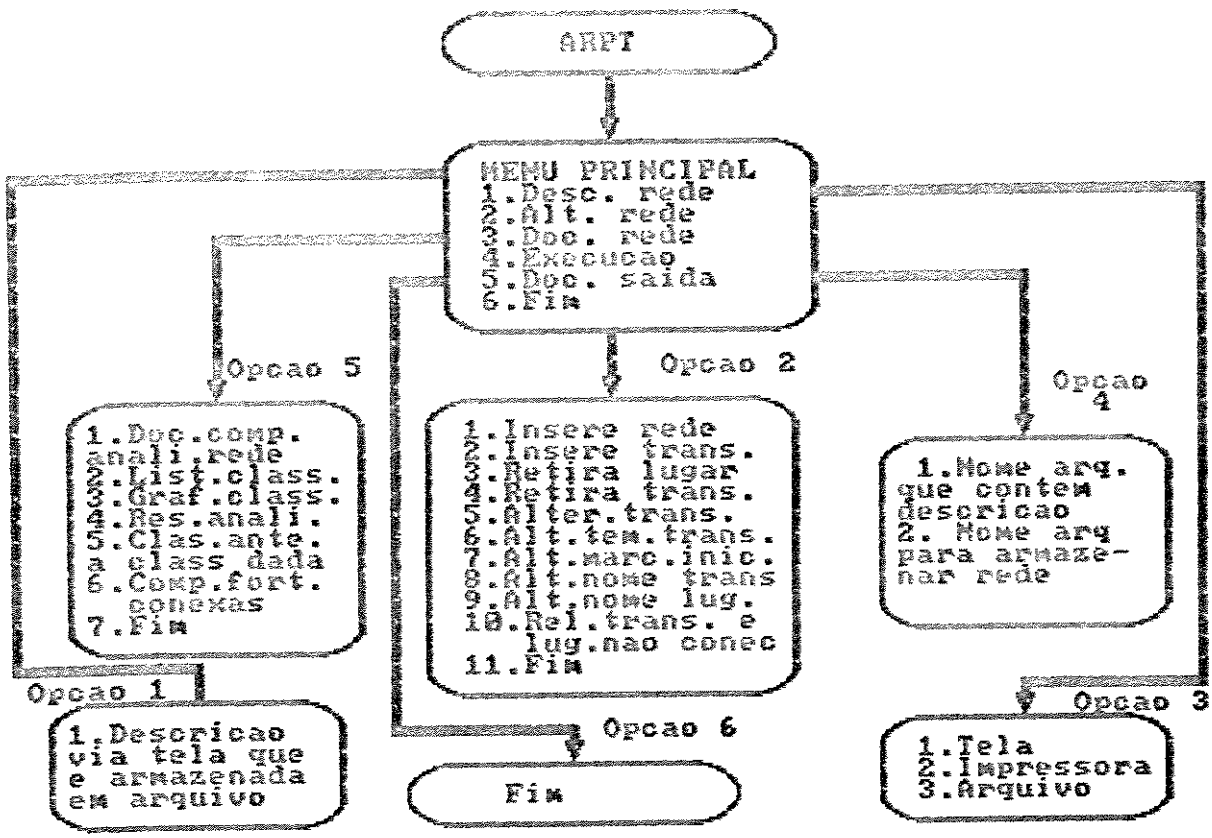

Fig.3.5 Cardapio de opcoes do ARPT em diagrama de bloco

3.3.2 - Opção Descrição da rede

A opção descrição permite gerar a descrição da rede . Para realizar esta operação os procedimentos desta opção interagem com as seguintes variáveis : matriz de incidência (m_inc), marcação inicial (marc_inic), vetor tempo estático inicial/final (tab_tdi,tab_tdf), tabela de nomes de transições (t_nome), tabela de nomes de lugares (l_nome).

Esta opção foi estruturada de forma que a rede a ser descrita deverá ser descrita inicialmente pelo usuário via teclado.

O usuário fornecerá os dados da rede respondendo ao questionário apresentado na tela.

Os dados pedidos pelo ARPT são os seguintes :

- número de lugares que compõem a rede.
- número de transições que compõem a rede
- para cada transição é pedido
  - . lugares de entrada
  - . lugares de saída
  - . TDI (tempo de disparo estático inicial)
  - . TDF (tempo de disparo estático final)
- lugares com senhas (marcação inicial)
- nomes das transições quando houver
- nomes dos lugares quando houver
- nome do arquivo para armazenar a descrição da rede

Como os modelos de protocolos de comunicação em RPT apresentam os pesos dos arcos de entrada e de saída das

transições iguais a 1(um), resolveu-se fixar os pesos com este valor.

Esta opção permite, a qualquer momento que o usuário desejar, de se abandonar a edição da descrição da RPT e voltar ao cardápio de opções do ARPT.

3.3.3 - Opção Alteração da rede

A opção alteração permite realizar a alteração da descrição da rede . Para realizar esta operação os procedimentos desta opção interagem com as seguintes variáveis : matriz de incidência (m_inc), marcação inicial (marc_inic), vetor tempo estático inicial/final (tab_tdi,tab_tdf), tabela de nomes de transições (t_nome), tabela de nomes de lugares (l_nome).

Esta opção foi estruturada em forma de um subcardápio. Ao ser escolhida, é pedido o nome do arquivo onde está a descrição da rede para que se possa ter acesso ao subcardápio.

As opções deste subcardápio são mostradas na fig.3.6.

As opções insere lugar e insere transição permitem introduzir na rede mais lugares e transições, respectivamente.

As opções retira lugar e retira transição permitem desconectar lugares ou transições da rede, respectivamente.

A opção altera transição permite redefinir os lugares de entrada e de saída da transição.

A opção altera tempo das transições permite alterar os

valores de TDI estático e TDF estático.

A opção **altera marcação inicial** permite alterar os lugares com senha inicial.

As opções **altera nome da transição** e **altera nome do lugar** permitem que se faça a alteração dos nomes da transição e do lugar, respectivamente.

A opção **relata transições e lugares não conectados** permite saber quais transições e lugares estão desconectados da rede.

A opção **fim** faz com que se volte para o cardápio do ARPT, mas antes pede o arquivo em que se deseja armazenar a rede alterada.

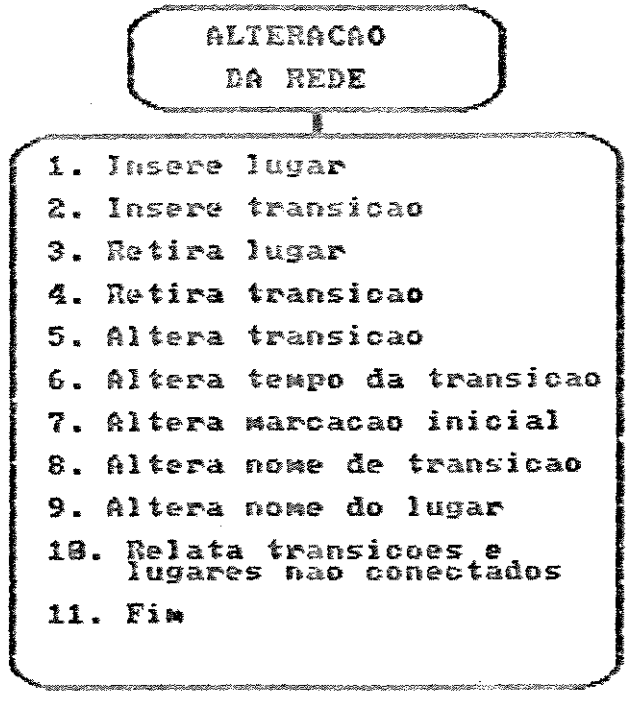

Fig.3.6 Subcardapio da opcao alteracao da rede

3.3.4 - Opção Documentação da rede

A opção documentação permite editar a rede via tela, via impressora e via arquivo. Ao iniciar a documentação o usuário deve fornecer o nome do arquivo onde está a descrição da rede.

Para poder editar via :

1 - Tela

O usuário deverá teclar "con:" e então, será apresentado na tela :

. o número de lugares

. o número de transições

. Para cada transição

- os lugares de entrada

- os lugares de saída

- TDI estático

- TDF estático

. a marcação inicial

. os nomes das transições

. os nomes dos lugares

2 - Impressora

O usuário deverá teclar "lst:" e os dados acima serão impressos em seguida.

3 - Arquivo

Os dados acima da rede serão armazenados num

arquivo, cujo nome é pedido ao usuário.

3.3.5 - Opção Execução

Esta opção permte montar a lista de classes de estado, que é composta pelas variáveis : tabela de marcações (tab_marca), matriz de tempos inicial/final (m_ini_fim), matriz de diferenças de tempo (m_dif_tempo); o grafo de classes de estado (grafo_ce) e a tabela das componentes fortemente conexa (m_cfc) e verificar as propriedades da RPT.

Ao iniciar a execução, o usuário deve fornecer o nome do arquivo onde está a descrição da rede criado pela opção descrição ou alteração.

Em seguida deve digitar o número máximo de classes de estados que se deseja gerar.

Ao verificar a ocorrência da primeira classe superior (rede não limitada) a execução do programa é interrompida e o usuário deve optar em continuar a análise ou o término da execução.

Se o número de classes de estado desejado for inferior ao número total de classes de estados da RPT. O programa avisa que não serão verificadas as propriedades de vivacidade e reiniciabilidade pois a lista e o grafo de classe de estados estão incompletos.

Após a montagem completa da lista e do grafo de classes de estado, o programa passa a verificar as propriedades de reiniciabilidade e vivacidade seguindo os algoritmos apresentados nas seções 3.2.2 e 3.2.3.

Em seguida é pedido o nome do arquivo para armazenar o resultado da análise

3.3.6 - Opção documentação de saída.

Esta opção permite realizar a documentação dos resultados da análise da RPT. Ao iniciar a documentação, o usuário deve fornecer o nome do arquivo onde está o resultado da análise da RPT.

Esta opção foi estruturada em forma de um subcardápio. As opções deste subcardápio são mostrados a fig.3.7.

Ao digitar a opção é pedido se deseja-se que a documentação seja tela, impressora ou arquivo.

Para isso basta digitar:

1 - "con:", para via tela

2 - "lst:", para via impressora

3 - Nome do arquivo, para via arquivo

A opção documentação completa da análise da rede fornece todos os resultados obtidos da análise.

A opção lista das classes de estado fornece todas as classes de estado geradas pela análise.

A opção grafo das classes de estado fornece o grafo das

classes de estados

A opção resumo da análise apresenta os resultados da análise de limitabilidade, reinicibilidade e vivacidade.

Na opção classes antecessora a uma classe dada, o usuário deve digitar o código de uma classe de estado Ci e obterá todas as classes de estado que apresentam uma seqüência de disparo que conduz a Ci.

A opção componentes fortemente conexas apresenta cada compomente fortemente conexa com suas respectivamente classes de estado e as seqüências de disparo. A seguir é apresentado as ligações entre as componentes fortemente conexas.

Também são apresentados as classes mortas ou seja as classes onde não há transições habilitadas o que implica em situações de impasse (do inglês deadlock).

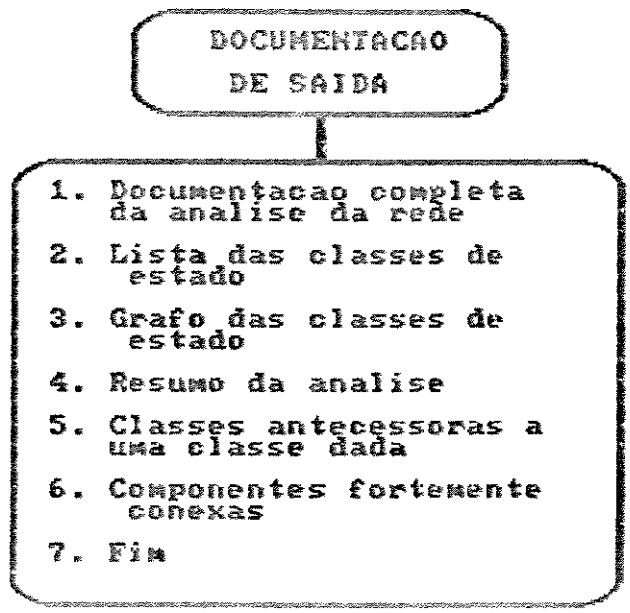

Fig.3.7 Subcardapio de opcao da documentacao de saida

3.4 - Conclusão

Neste capítulo foram discutidos os algoritmos utilizados para a verificação das propriedades de limitabilidade, vivacidade e reiniciabilidade, bem como a geração da lista e do grafo de classes de estado.

Foi apresentado a estrutura do ARPT em forma de Cardápio de opções, para facilitar a interação com o usuário.

O conhecimento da estrutura do ARPT é importante para utilizá-lo na validação de protocolos que é o que será visto no próximo capítulo.

CAPÍTULO 4

Utilização do ARPT na análise e validação do protocolo de bit alternante.

4.1 - Introdução

Neste capítulo é apresentado uma metodologia de análise para a validação de protocolos de comunicação utilizando o analisador automático de rede de Petri temporizada (ARPT). Em seguida é apresentado o protocolo de bit alternante [21] para transferência de dados e testes pelo ARPT de casos modificados deste protocolo.

4.2 - Metodologia de análise para validação de protocolo usando o ARPT.

A metodologia de análise para a validação de protocolos usando o ARPT consiste das seguintes etapas.

a) Converte-se o protocolo de uma especificação em linguaguem natural para o modelo da RPT. Se o protocolo estiver

especificado em uma linguaguem formal de projeto (diagrama de estados, SDL : Functional Specification and Description Language - CCITT, etc...) converte-se o protocolo para o modelo da RPT equivalente.

Tal conversão deve garantir a manutenção da integridade do protocolo. As regras para a conversão de SDL para RPT consistem das utilizadas na conversão para a RP clássica apresentadas em [32] acrescidas dos modelamentos dos temporizadores e de perdas de mensagens.

b) A partir das especificações do protocolo (que devem conter estimativas aproximadas para os tempos mínimos e máximos necessários para a execução de suas diversas ações) associa-se os intervalos estáticos para cada transição da rede, modela-se os temporizadores e as possíveis perdas de mensagens durante as trocas de informação pelos diversos processos do protocolo.

c) Através do ARPT, verifica-se as propriedades específicas da RPT (limitabilidade, reiniciabilidade, vivacidade)

d) A não verificação (imperfeição) de uma de suas propriedades pode ser indicativo da existência de erros na especificação do protocolo ou de um modelamento em RPT não satisfatório (por exemplo : um mau dimensionamento dos intervalos estáticos atribuídos às transições da RPT inclusive aqueles relativos aos temporizadores).

A análise de um mau dimensionamento da RPT equivalente e\ou de falhas de projeto do protocolo podem em muito ser enriquecidas por diagnósticos mais detalhados, que o ARPT pode fornecer sobre as imperfeições da rede (ver seção 4.2.1). Feitas as modificações necessárias volta-se a etapa anterior e

assim até a correção total das eventuais falhas do protocolo.

Esta metodologia de validação de protocolo é mostrada na fig.4.1.

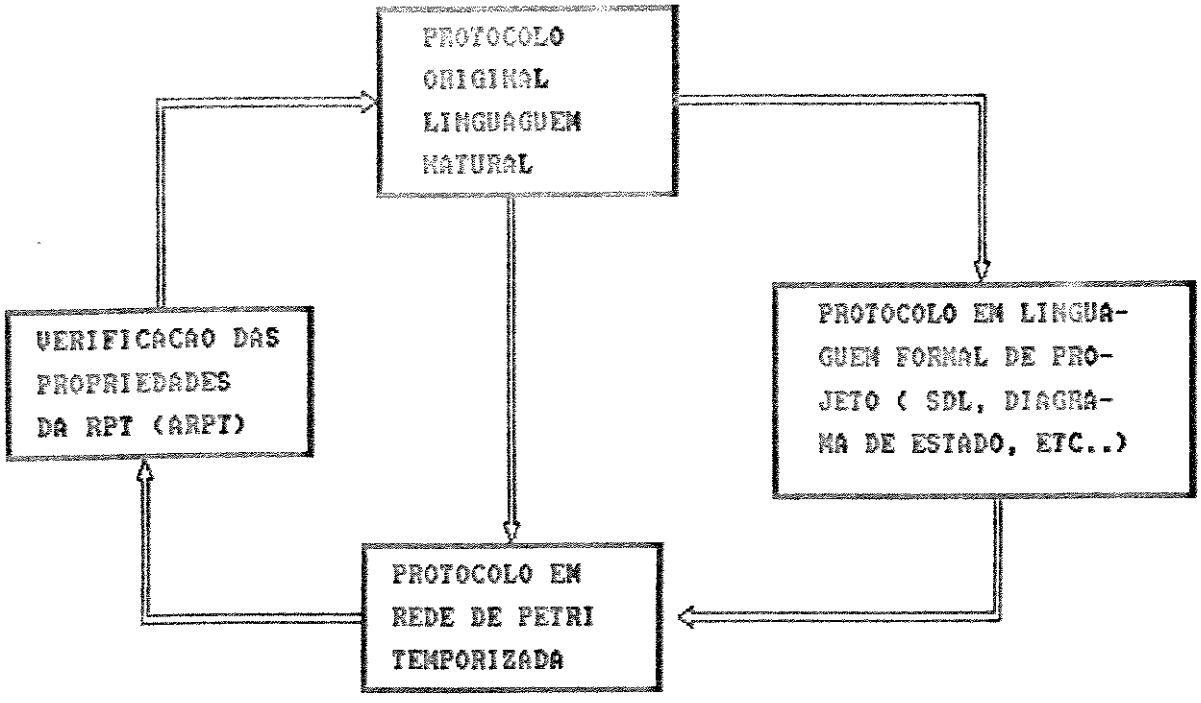

Fig.4.1 - Metodologia de validacao de protocolos

4.2.1 - Utilização e interpretação dos diagnóticos fornecidos pelo ARPT.

Assim que a lista e o grafo de classes de estados forem construídos e as propriedades gerais testadas pelo ARPT podemos interpretar os diágnosticos que são fornecidos pelo subcardápio " Documentação de saída ". No caso do diagnótico ser :

A) A rede é não limitada : isto significa que existe no mínimo uma classe de estado superior a uma outra em um de seus ramos ascendentes.

Assim existe uma seqüência de ações (seqüência de disparo), que repetida faz com que o sistema não consiga processar todas as informações no tempo definido.

B) A rede é não reiniciável , então 2 casos podem se apresentar :

1) A rede está com impasse (do inglês "deadlock"), ou seja existe no mínimo uma classe morta na rede, fazendo com que nenhuma evolução na rede seja possível a partir desta classe de estado.

2) Existe mais de uma componente conexa no grafo de classes e no mínimo uma delas é uma componente fortemente conexa isolada (CFCI).

Neste caso o sistema pode cair num loop improdutivo de forma que, embora a cada instante exista algum processamento sendo feito, o sistema não atinge o seu objetivo final.

C)A rede é não viva, então 3 casos podem se apresentar :

1) A rede está com impasse.

2) Pelo menos uma transição nunca é disparada, ou seja algum evento do sistema jamais ocorrerá pois as condições para tal nunca estarão satisfeitas.

3) Existe mais de uma componente fortemente conexa no grafo de classes e no mínimo uma delas é uma CFCI para a qual nem todas as transições disparam. Neste caso o sistema pode cair num loop onde algumas ações da rede não serão executadas.

Em todos os casos acima, as imperfeições podem ser fornecidas pelo ARPT : lugares não limitados, transições não disparáveis, classes não reiniciáveis, as classes mortas e os CFCI existentes.

Afim de auxiliar o usuário na análise das causas dos diversos mau funcionamentos (que pode ser devido a erro de concepção ou erros introduzidos na fase de modelamento da rede); a opção classes antecessoras a uma classe dada da opção " Documentação de saída " fornece para uma dada classe o seu ramo ascendente que inclui as classes antecessoras e a seqüência de disparo.

Fornecendo a classe que apresenta o problema (classe morta, classe superior, classe não reiniciável, etc....) obtemos os possíveis caminhos que a partir de C0 conduz a tal situação. Analisando estes caminhos podemos propor as alterações devidas.

Além disso, a opção "componentes fortemente conexas" fornece todos as CFC e CFCI com suas respectivas classes e

seqüências de disparo, como também todas as ligações (transições que disparam) entre as componentes fortemente conexas.

Assim com essas opções e mais algum conhecimento sobre o protocolo em estudo consegue-se aplicar a metodologia de análise para a validação do protocolo.

4.3 Protocolo de transferência de dados (bit alternante)

Como mencionado na introdução, os protocolos de comunicação, em geral,possuem restrições temporais em suas especificações. Por exemplo, os mecanismos de reconfiguração para a perda de mensagem são muitas vezes implementados com a ajuda de temporizadores. O método de validação apresentado permite verificar se os valores destes temporizadores estão corretamente escolhidos.

O exemplo a ser usado para a validação de um protocolo de comunicação, através do ARPT será o protocolo de bit alternante.

Este protocolo é um protocolo de transferência de dados do tipo envia - recebe . Antes de emitir uma nova mensagem, o processo emissor espera a chegada do reconhecimento da mensagem que foi enviada.

As hipóteses sobre o funcionamento do meio de comunicação são que as mensagens ou seus reconhecimentos (do inglês "acknowledgements") podem ser perdidos ou degenerados, mas não duplicados nem reordenados.

Um mecanismo de temporização e de retransmissão de mensagem permite corrigir as perdas de mensagens. Um temporizador é inicializado quando uma mensagem é transmitida. Se o reconhecimento da mensagem não chegar em um dado tempo pré-estabelecido (expiração da temporização) (do inglês "time-out"), suficiente para que se garanta que a mensagem esteja perdida, será retransmitida a mensagem e se iniciará uma nova contagem. Se um reconhecimento chegar, o temporizador é desativado.

Este mecanismo básico permite sanar as perdas de mensagens, mas pode introduzir duplicação de mensagens indesejáveis.

Assim, em relação a recepção de uma mensagem, o processo do receptor não pode decidir se a mensagem que é recebida é uma nova mensagem ou uma copia da última mensagem que foi aceita.

Para se garantir que o receptor possa detectar as mensagens duplicadas, elas são numeradas a priori pelo transmissor, com número de seqüência módulo 2 e para todas mensagens (mens.) recebidas pelo receptor um reconhecimento (rec.) é enviado com o número da seqüência da mensagem recebida. Assim são enviadas alternadamente mensagens com o número da seqüência igual a 0 "mens. 0" ou 1 "mens.1". Para ser enviada uma após outra o enviador espera o reconhecimento "rec.0" ou "rec.1" no caso da última mensagem ter sido "mens.0" ou "mens.1", respectivamente. A retransmissão só será feita caso o temporizador correspondente expirar.

O protocolo de bit alternante pode ser representado pela RPT da fig.4.3. A fig.4.2 associa os significados das ações elementares previstas no protocolo com as respectivas transições

que as representam na RPT da fig.4.3.

t1 : transmite " mens.0"	t9 : recebe/rejeita "mens.0"
t2 : retransmite "mens.0"	t10 : recebe/libera "mens.1"
t3 : recebe "rec.0"	t11 : transmite "rec.1"
t4 : envia "mens.1"	t12 : recebe/rejeita "mens.1"
t5 : retransmite "mens.1"	t13 : perde "mens.0"
t6 : recebe "rec.1"	t14 : perde "rec.0"
t7 : recebe/ libera "mens.0"	t15 : perde "mens.1"
t8 : transmite "rec.0"	t16 : perde "rec.1"

Fig.4.2 - Significado das transicoes para o protocolo de bit alternante

Uma simplificação é adotada, as mensagens degeneradas são comparadas as mensagens perdidas e é suposto que a temporização de retransmissão é suficientemente longa para que o meio de transmissão não contenha mais que uma mensagem ou reconhecimento de cada vez.

Nesta rede, as perdas de mensagens ou de reconhecimento são representadas por transições que não têm lugares de saída. Isto é, na fig.4.2 as transições t13 e t14 representam as perdas da "mens.0" e do "rec.0" enquanto as perdas da "mens.1" e do "rec.1" são representadas pelas transições t15 e t16 respectivamente. Estimativas aproximadas (intervalo de disparo estático para cada transição) para a duração de cada ação elementar devem ser fornecidas.

A retransmissão de uma mensagem (temporizadores relativos as transições t2 ou t5) ocorrerá num tempo compreeendido entre 5 e 6 unidades de tempo após a emissão de sua

última cópia. No caso de perda de mensagem (transições t13 ou t15) e de reconhecimento (transições t14 ou t16), os tempos são estimados entre 0 e 1. Para recepção de mensagem (t7 ou t10) e de reconhecimento (t3 e t6), os tempos são estimados entre 0 e 1. Para a recepção e rejeição de mensagem (transições t9 e t12), os tempos são estimados entre 0 e 1 . Para a transmissão de reconhecimento (transições t8 e t11) os tempos são estimados entre 0 e 2.

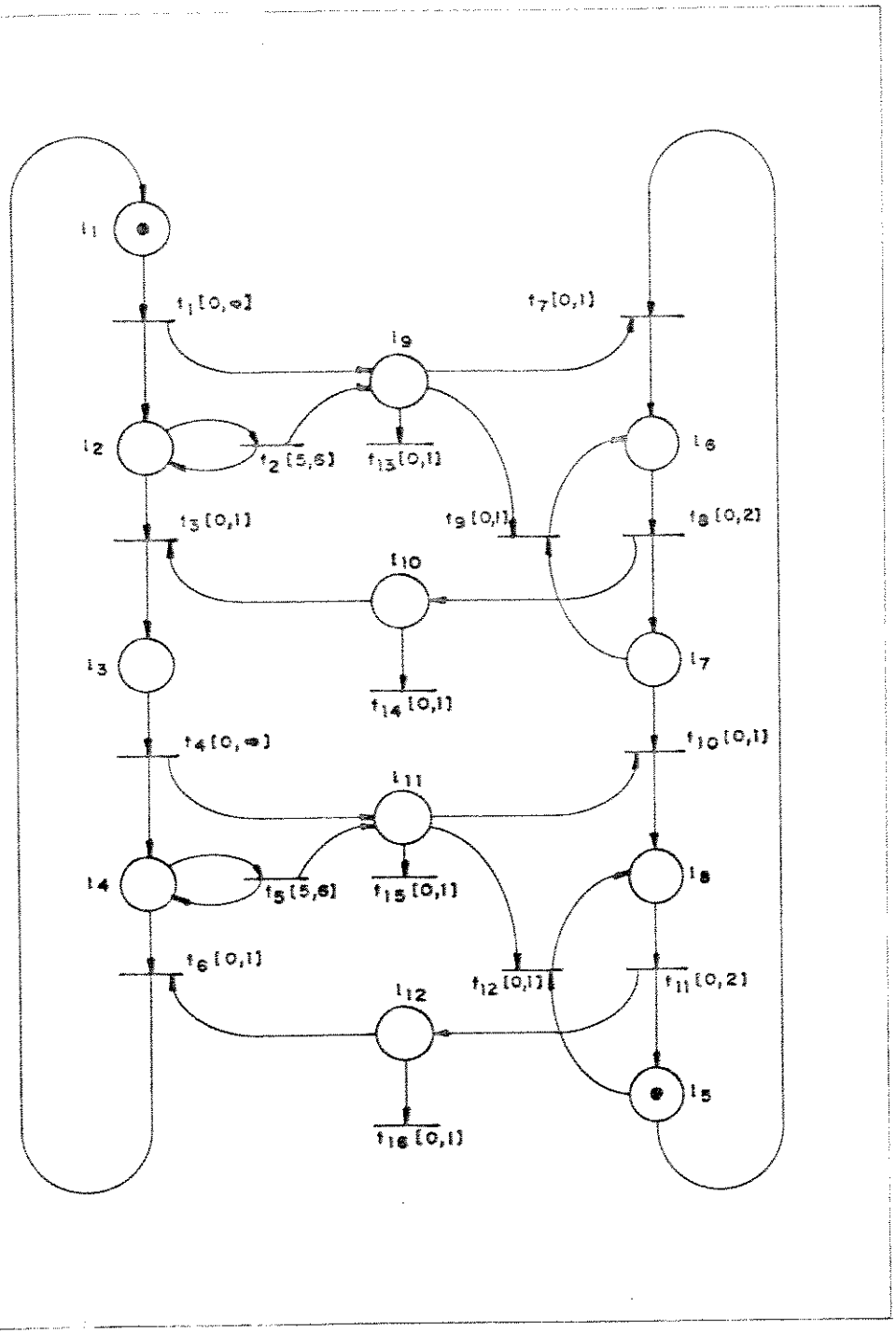

Fig.4.3 RPT para o protocolo de bit alternante

4.4 - Análise pelo ARPT do protocolo de bit alternante

Utilizando o ARPT na análise do protocolo do bit alternante representado pela RPT da fig.4.3, chega-se ao resultdo de que a rede é limitada, viva e reinícIavel e portanto sem imperfeições. A listagem de resultados da saída do ARPT é apresentada no Apêndice 2.

O grafo da classes de estado apresenta todas as seqüências de disparo admissíveis para este protocolo com os temporizadores, representados por t2 ou t5, compreendidos entre 5 e 6 unidades de tempo.

Analisando, o protocolo na parte relativa a transmissão e recepção do bit do tipo 0,que é válido também para o bit tipo 1, notamos que o tempo mínimo de disparo do temporizador é maior que o tempo máximo de recebimento e liberação da "mens.0" somado ao tempo máximo da transmissão de "rec.0" e somado ao tempo máximo para recebimento de "rec.0", isto é :

TDI de t2 > TDF de t7 + TDF de t8 + TDF de t3

Neste caso, (5 > 1 + 2 + 1).

Podemos nos perguntar o que acontece quando o tempo mínimo do temporizador (TDI de t2) é igual a soma dos outros tempos máximos dado acima ou seja TDI = 4.

Para tal pode-se usar o ARPT para fazer a análise desta nova rede. Entretanto o modelo utilizado na fig.4.3 não é o mais adequado para este caso.

Isto é devido, ao fato de que é possível que para esta rede o temporizador dispare e exista um reconhecimento "rec." no

meio de comunicação (lugar 10). Quando o disparo ocorre, é reabilitada a transição t3 que em conseqüência representa um atraso maior para a chegada de "rec." que não corresponde ao funcionamento real.

A RPT que apresentamos na fig.4.3 é um modelo simplificado do protocolo de bit alternante. A fig.4.4 apresenta a parte a ser substituída na rede da fig.4.3, a RPT resultante é um modelo mais abrangente do protocolo por representar todas as possíveis situações de valores associados as durações previstas no protocolo.

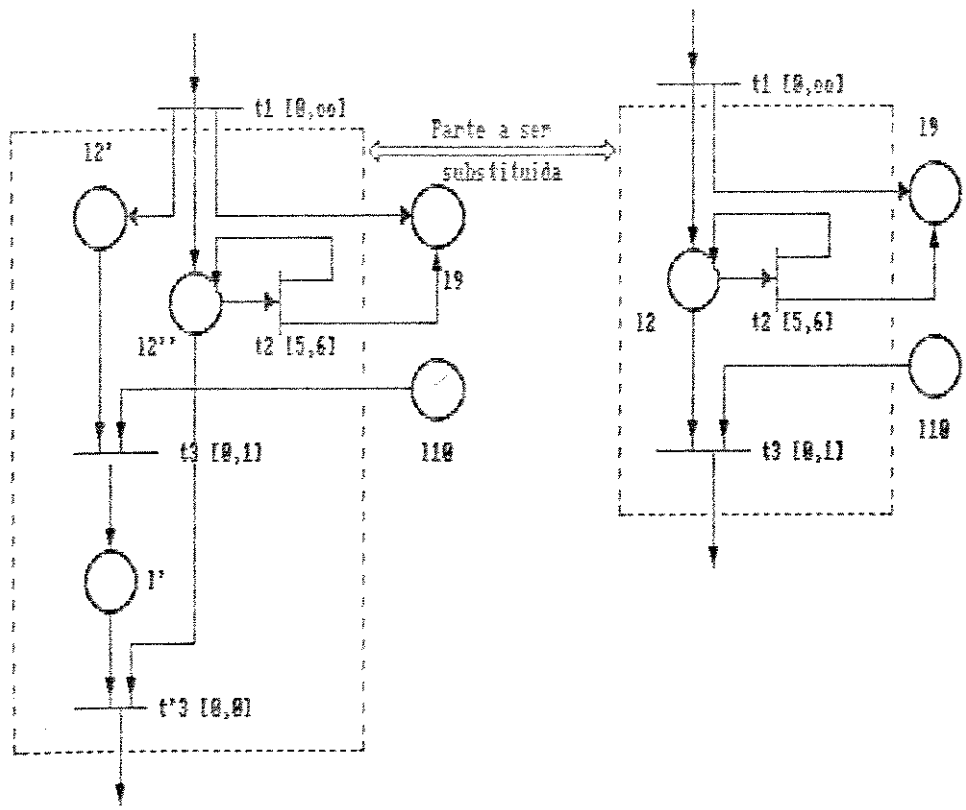

Fig.4.4 - Parte a ser substituida na RPT da fig.4.3
(vale para o temporizador t5)

Este modelo é obtido da separação do lugar 12 em dois lugares 12' e 12'' que servem respectivamente para iniciar o temporizador e para receber o reconhecimento. A transição t3 é também separada em duas transições em série, na qual a última será responsável para desarmar o temporizador.

Este modelo é mais real pois o disparo do temporizador não influência no atraso da recepção do reconhecimento.

A análise desta nova rede pelo ARPT conduziu a uma rede limitada. Assim podemos concluir que não é necessário que o tempo mínimo do temporizador seja superior aos tempos máximos da via de comunicação, pois todo reconhecimento é consumido pelo receptor ou se perde no meio de comunicação.

Se desejarmos saber o que acontece quando o tempo mínimo dos temporizadores é menor ainda, por exemplo TDI =3 usamos novamente o ARPT.

Neste caso a análise deste novo protocolo conduziu a uma nova rede não limitada com W aparecendo no lugar 19 e 111, o que significa o risco do receptor confundir uma nova mensagem com uma velha, devido não conseguir processar todas as informações no tempo definido.

Pela variação dos intervalos estáticos de algumas transições da rede estudamos as implicações em relação ao temporizadores e pudemos notar que existem três faixa de valores:

1) O temporizador está superestimado, neste caso a rede se mantém limitada, viva e reiniciável porém a eficiência do protocolo fica baixa.

2) O temporizador está bem estimado, neste caso a rede se mantém limitada, viva e reiniciável porém a eficiência

aumenta em relação ao anterior.

3) O temporizador está subestimado, neste caso temos dois tipos de situações:

a) Há mensagem e reconhecimento simultâneo no meio de comunicação, mas os processos conseguem identificá-los.

b) Há mensagens e reconhecimentos simultâneo no meio de comunicação, mas os processos não conseguem indentificá-los, ocorrendo a não limitabilidade da rede.

4.5 - Conclusão.

Neste capítulo apresentou- se uma metodologia de análise e validação de protocolos de comunicação através de seu modelo em rede de Petri Temporizada (RPT).

Também apresentamos a utilização e interpretação dos diagnósticos fornecidos pelo ARPT.

Como exemplo de aplicação consideramos o protocolo de bit alternante para o qual a RPT equivalente foi apresentada e testada pelo ARPT. Pela variação dos intervalos estáticos de algumas transições da rede estudamos suas implicações no comportamento do protocolo.

CAPÍTULO 5

Conclusão Final

Neste trabalho o método formal de validação de protocolo é baseado em Rede de Petri.

Como estamos interessados em protocolos que possuam tempo como elemento básico, optamos por usar o modelo de Rede de Petri Temporizada proposto por Merlin que é o mais apropriado e tem sido usado para modelamento e análise de protocolos de comunicação que possuem temporizadores e/ou atrasos variáveis na troca de informação.

Com este modelo implementou-se um Analisador Automático de Rede de Petri Temporizada (ARPT), que fosse conversacional com o usuário.

Para a sua implementação foi apresentado um método de análise de RPT por enumeração de classes de estado e testes de suas propriedades.

Em relação à propriedade de limitabilidade apresentamos uma condição necessária e suficiente para que uma RPT t segura e com peso dos arcos igual a 1 (um), seja limitada.

O ARPT é um pacote de software que foi desenvolvido em linguagem Pascal e é executado em microcomputador tipo PCaT.

Na primeira versão do ARPT [26], que foi feito totalmente em alocação estática, sua capacidade para analisar redes estava limitada à análise de redes de até 20 lugares e 20 transições gerando no máximo 65 classes de estado.

A atual versão, que foi feita em alocação dinâmica, está

capacitada para analisar de até 130 transições e 500 lugares e com uma geração de classes de estados até agora testada de 600 classes.

Uma limitação do programa para RPT maiores está no número de transições, pois o programa apresenta ainda algumas estruturas em alocação estática.

A estrutura do programa é feita em forma de cardápio de opções ao usuário, sendo capaz de fazer a descrição via tela ou arquivo, alteração e documentação da rede, além da análise com seus resultados.

O conhecimento da estrutura do ARPT é importante para utilizá-lo na análise e validação de protocolos.

Uma metodologia para validação de protocolos se torna necessária como apresentada na seção 4.2 e a utilização e interpretação dos diagnósticos fornecidos pelo ARPT apresentados na seção 4.2.1 contribui bastante para a validação de protocolos.

Sugestões para futuros trabalhos:

- A estrutura do ARPT é tal a facilitar a implementação de um módulo de simulação de RPT, que muito auxiliaria na análise dos protocolos. Onde o usuário definiria o conjunto de ações do protocolo que ele queira verificar. Em uma simulação passo a passo para uma dada classe escolhida o programa forneceria as transições habilitadas, das quais o usuário escolheria uma e assim por diante, até que ele desejasse parar.

- Visando a utilização do ARPT para redes maiores e para um menor tempo de execução torna-se necessária a sua implementação num computador de grande porte tipo VAX. O que em

grande parte já foi facilitado pela linguagem de alto nível escolhida e pela estrutura dada ao pacote.

- Uma redução do número de classes de estado através de verificação de classes equivalentes é importante para se facilitar análises detalhadas sobre os dados e diagnósticos fornecidos pelo programa.

- Outra forma de se facilitar a análise da RPT é a de pesquisarmos propriedades estruturais de uma maneira análoga a que existe para a RP Clássica, de tal maneira a que possam permitir a obtenção de suas propriedades sem ter que fazer uma enumeração exaustiva das classes de estado. Tais propriedades seriam relativas às regras de redução e à de obtenção de invariantes [9,10] para a rede temporizada.

Bibliografia.

/1/ Zimmermann, H.

OSI reference model - The ISO of architecture for Open Systems Interconnection, IEEE Transactions on Communications, Vol com - 28, N.o 4, pp 425 - 432, Abril 1980.

/2/ Borelli, W.C. ; Marton,M. ; Guimarães L.

Arquiteturas de Redes de Computadores e Protocolos de comunicação Publ.Interna FEC 067/84 , 1984.

/3/ Tanembaun, A.S.

Computer Networks, Prentice Hall, WC, N. Jersey, 1981.

/4/ Menascé, D. A. ; Schwabe, D.

Redes de Computadores. Aspectos Técnicos e operacionais. Editora Campus LTDA, R.J 1984.

/5/ Courtiat, J. P. ; Ayache, J.M. ; Algayes, B.

Petri nets are good for protocols, Communications Architeture and protocols, ACM sigcomm 84, Montreal, junho 1984.

/6/ Diaz, M. ; Courtiat, J.P. ; Berthomieu, B.; Ayache, J. M.

Status of using Petri net based models for protocols. IEEE Transaction on Computer, 1983 p.p. 1301 - 1305.

/7/ Merlin P.

Specification and validation of protocols. IEEE Transaction on Communication Vol.com 27 N.o 11 Novembro 1979 p.p 1671 - 1680

/8/ Berthelot, G. ; Terrat R.

Petri nets theory for the correcteness of Protocols IEEE Transactions on Communications Vol com - 30 N.o 12 Dezembro 1982

p.p.2497 - 2505.

/9/ Chezalviel-Pradin, B.

Un outil graphique interactif pour la verification des systèmes à evolution paralèle decrit par réseaux de Petri, Thèse Docteur Ingénieur Université Paul Sabatier, Toulouse, Dezembro 1979.

/10/ Dufau, J.

Un outil pour la verification des protocoles decrits par reseaux de Petri, Thèse Docteur Ingénieur Université Paul Sabatier, Toulouse, Janeiro 84.

/11/ Arantes, M.P.C.

Análisador Automático de rede de Petri para validação de protocolo de comunicação, Tese de Mestrado, Unicamp, Fevereiro 1988.

/12/ Tamura, R.T.

Um método para validação de protocolo de comunicação especificado em SDL utilizando rede de Petri, Tese de Mestrado, Unicamp, Campinas, Agosto 1988.

/13/ Marton, M. ; Damasceno, B.C. ; Borelli, W.C.

Validação sem temporizações do procedimento de acesso ao enlace de dados do canal D, TRÓPICO R ; Publ. interna FEC 030/86, 1986

/14/ Damasceno, B.C. ; Marton, M. ; Borelli, W.C.

Rede de Petri clássica com modelamentos de temporizadores (RPMT) ; Publ. interna FEC 013/87 ,junho de 1987.

/15/ Diaz, M.

Modeling and analysis of communication and cooperation protocol using Petri net based models. Computer networks 6,

p.p.419 - 441, 1982.

/16/ Merlin, P. ; Farber, D. J.

Recoverability of communications protocols- Implications of a theorical study, IEEE Transactions on communications, Vol com-24, N.o 9, pp 1036 a 1043, setembro 1976.

/17/ Menasche, M.

Analyse des reseaux de Petri Temporisés et application aux systémes distribués, Thèse, Université Paul Sabatier Tolouse, Novembro 1982.

/18/ Berthomieu, B. ; Menasche, M.

A state enumeration aproach for analyzing time Petri nets, 3.rd european workshop on aplications and theory of Petri nets, Varenna, Italy, setembro 1982.

/19/ Menasche, M.

PAREDE : An Automated Tool For the Analysis of Time(d) Petri Nets, Workshop on Timed Petri Nets,pp 162-169, jul85, Torino, Itália

/20/ Peterson, J.L.

Petri net theory and modeling of systems. Prentice Hall,1981.

/21/ Bartlett, K.A. ; Seantlebury, R. A. ; Wilkinson, P.T.

A note on realiable full duplex transmission over half-duplex link , Communications of the ACM, Vol 12, N.o 5, Maio 1969

/22/ Peterson, J.L.

Petri nets, Computing Surveys, Vol. 9, N.o 3, 1977

/23/ Berthelot, G.

Verification des réseaux de Petri. Thèse Docteur Ingénieur,

Université Paris VI, Paris,1978.

/24/ Berthomieu, B. ; Menasche, M.

Une aproache par enumeration pour l'analyse des réseaux de Petri Temporels, 9th IFIP Congress 1983, Paris, setembro 1983, pp. 41 a 46.

/25/ Marton, M. ; Borelli, W.C.

Rede Petri Temporizada : Modelos e exemplo ; Publ. interna FEC 077/86, 1986.

/26/ Marton,M. ; Damasceno, B.C.; Rabay F., G.; Borelli, W.C.; Motoyama, S.

SIPROTl. Manual e exemplos ; Publ. interna 001/88, fev.88.

/27/ Collins, W.J.

Programação Estruturada com Estudos de Casos em PASCAL - Mcgraw-Hill , 1986.

/28/ Menasche, M. ; Berthomieu, B.

Time Petri nets for analyzing and verifying time dependent protocols. 3.rd International Workshop on protocol Specification, Testing and Verification.

/29/ Jones, N.D. ; Landweber, L.H. ; Lien, Y.E

Complexity of some problems in Petri nets. Theoretical Computer Science n.o 4 , 1977 p.p. 277 - 299.

/30/ Karp, R.M. ; Miller, R.E.

Parallel program schemata. Journal of Computer and System Sciences n.o 3, p.p. 147 - 195 , 1969.

/31/ Merlin P.

Methodology for the design and implementation of communication Protocols. IEEE transaction on Communications. Vol com 24 N.o 6 junho 1976

/32/ Borelli, W.C. ; Marton M., Motoyama S.

Conversão de SDL para Rede de Petri ; Publ. interna FEC 100/85, outubro de 1985.

APÊNDICE 1

ESTRUTURAS DE DADOS DO ANALISADOR AUTOMATICO DE REDE DE PETRI TEMPORIZADA

ESTRUTURAS DE DADOS DO ANALISADOR AUTOMATICO DE REDE DE PETRI TEMPORIZADA

1 - As estruturas de dados deste analisador são :

- Matriz de Incidência (m_inc)
- Tabela de Marcações(tab_marca)
- Matriz de Diferenças de Tempo (m_dif_tempo)
- Matriz de Tempos Inicial/Final (m_ini_fim)
- Grafo de Classes de Estado (grafo_ce)
- Marcação Inicial (marc_inic)
- Vetor Tempo Estático Inicial/Final(tab_tdi,tab_tdf)
- Tabela de nomes de transições (t_nome)
- Tabela de nomes de lugares (l_nome)
- Tabela das componentes fortemente conexa (m_cfc)

Na análise da RPT correspondente a um protocolo, verificou-se que a maior parte das posições destas matrizes apresentavam elementos nulos caracterizando matrizes esparsas [26], então optou-se pela melhor utilização da memória que foi feita atráves de alocação dinâmica pelo uso de apontadores [27].

Para armazenar as informações das matrizes na forma de alocação dinâmica de memória, foi utilizada a técnica de filas, onde cada posição da fila é acessada pelo apontador do início da fila.

No ARPT quatro tipos de apontadores foram definidos :

- Apontador 1, possui um registro de 4 campos que é utilizado

pelo grafo de classes (grafo_ce), tabela de marcações(tab_marc) e matriz de incidência (m_inc).

- Apontador 2, possui um registro de 5 campos que é utilizado pela matriz de diferenças de tempo (m_dif_tempo) e matriz de tempo inicial e final (m_ini_fim).

- Apontador 3, com um registro de 3 campos que é utilizado pela tabela das componentes fortemente conexos (m_cfc).

- Apontador 4, com um registro de 3 campos que é utilizado pela tabela de nomes das transições (t_nome) e pela tabela de nomes de lugar (l_nome).

Os campos dos quatros registros destes apontadores são mostrados nas figuras 1, 2, 3 e 4 a seguir.

CAMPO\VAR	grafo_ce	tab_marc	m_inc
X1	classe	classe	transição
X2	transição	lugar	lugar
X3	nova classe	num senhas	peso
X4	next	next	next

Fig.1 Campos do apontador 1 para cada variável.

CAMPO\VAR	m_dif_tempo	m_inic_fim
Y1	classe	classe
Y2	transição b	transição b
Y3	transição k	tdi(b)
Y4	Xb - Xk	tdf(b)
Y5	next	next

Fig.2 Campos do apontador 2 para as duas variáveis

CAMPO\VAR	m_cfc
Z1	num cfc
Z2	classe
Z3	next

Fig.3 Campos do apontador 3

CAMPO\VAR	t_nome	l_nome
W1	transição	lugar
W2	nome	nome
W3	next	next

Fig.4 Campos do apontador 4 para as duas variáveis.

2 - Descrição das variaveis globais.

n_atual - indica a classe que está sendo executada num determinado instante.

n_lug - representa o número de lugares da rede.

n_tra - representa o número de transições da rede.

tra_disp - indica se uma transição b é disparável numa determinada classe.

tab_tdi - tabela que armazena os tempos de disparos iniciais estáticos da rede.

tab_tdf - tabela que armazena os tempos de disparos finais estáticos da rede.

marc_inic - vetor que armazena a marcação inicial da rede.

grafo_ce - apontador que armazena o grafo de classes de estado da rede.

pgrafo_ce - apontador auxiliar que aponta sempre para o começo da fila do grafo de classes.

tab_marca - apontador que armazena a tabela de marcações da rede.

ptab_marca - apontador auxiliar que aponta para o começo da fila da tabela de marcações.

m_inc - apontador que armazena a matriz de incidência.

pm_inc - apontador auxiliar que aponta para o começo da fila da matriz de incidência.

m_ini_fim - apontador que armazena os tempos de disparos iniciais e finais das transições de uma classe.

pm_ini_fim - apontador auxiliar que aponta para o começo da fila que contém os tempos iniciais e finais.

fm_ini_fim - apontador auxiliar que que aponta para o fim da fila que contém os tempos iniciai e finais.

m_dif_tempo - apontador que armazena a matriz das diferenças de tempo das transições numa determinada classe.

pm_dif_tempo - apontador auxiliar que aponta para o começo da fila que contém as diferenças de tempo.

fm_dif_tempo - apontador auxiliar que aponta para o fim da fila que contém as diferenças de tempo.

t_nome - apontador que armazena a tabela de nomes das transições.

pt_nome - apontador auxiliar que aponta sempre para o começo da fila da tabela de nomes das transições.

l_nome - apontador que armazena a tabela de nomes dos lugares.

pl_nome - apontador auxiliar que aponta para o começo da fila da tabela de nomes de lugares.

m_cfc - apontador que armazena a tabela das componentes fortemente conexas.

pm_cfc - apontador auxiliar que aponta para o começo da fila da tabela das componentes fortemente conexas.

APÊNDICE 2

DESCRIÇÃO E RESULTADOS DA ANÁLISE DA RPT

DO BIT ALTERNANTE

```
DESCRICAO DA REDE

NUM LUGAR   : 12

NUM TRANSICOES : 16


TRANSICAO : 1

LUGARES ENTRADA:
   1 ,

LUGARES DE SAIDA :
  2 ,    9 ,

 TDI   : 0          TDF :  999


TRANSICAO : 2

LUGARES ENTRADA:
   2 ,

LUGARES DE SAIDA :
  2 ,    9 ,

 TDI   : 5          TDF :  6


TRANSICAO : 3

LUGARES ENTRADA:
   2 ,     10 ,

LUGARES DE SAIDA :
  3 ,

 TDI   : 0          TDF :  1
```

TRANSICAO : 4

LUGARES ENTRADA:
3 ,

LUGARES DE SAIDA :
4 , 11 ,

TDI : 0 TDF : 999

TRANSICAO : 5

LUGARES ENTRADA:
4 ,

LUGARES DE SAIDA :
4 , 11 ,

TDI : 5 TDF : 6

TRANSICAO : 6

LUGARES ENTRADA:
4 , 12 ,

LUGARES DE SAIDA :
1 ,

TDI : 0 TDF : 1

TRANSICAO : 7

LUGARES ENTRADA:
5 , 9 ,

LUGARES DE SAIDA :
6 ,

TDI : 0 TDF : 1

TRANSICAO : 8

LUGARES ENTRADA:
6 ,

LUGARES DE SAIDA :
7 , 10 ,

TDI : 0 TDF : 2

TRANSICAO : 9

LUGARES ENTRADA:
7 , 9 ,

LUGARES DE SAIDA :
6 ,

TDI : 0 TDF : 1

TRANSICAO : 10

LUGARES ENTRADA:
7 , 11 ,

LUGARES DE SAIDA :
8 ,

TDI : 0 TDF : 1

TRANSICAO : 11

LUGARES ENTRADA:
8 ,

LUGARES DE SAIDA :
5 , 12 ,

TDI : 0 TDF : 2

TRANSICAO : 12

LUGARES ENTRADA:
5 , 11 ,

LUGARES DE SAIDA :
8 ,

TDI : 0 TDF : 1

TRANSICAO : 13

LUGARES ENTRADA:
9 ,

LUGARES DE SAIDA :

TDI : 0 TDF : 1

TRANSICAO : 14

LUGARES ENTRADA:
10 ,

LUGARES DE SAIDA :

TDI : 0 TDF : 1

TRANSICAO : 15

LUGARES ENTRADA:
11 ,

LUGARES DE SAIDA :

TDI : 0 TDF : 1

TRANSICAO : 16

LUGARES ENTRADA:
12 ,

LUGARES DE SAIDA :

TDI : 0 TDF : 1

MARCACAO INICIAL :
1(1), 5(1),

DOCUMENTACAO DOS NOMES DAS TRANSICOES

TRANSICAO T1 NOME = transmite "mens.0"

TRANSICAO T2 NOME = retransmite "mens.0"

TRANSICAO T3 NOME = recebe "rec.0"

TRANSICAO T4 NOME = envia "mens.1"

TRANSICAO T5 NOME = retransmite "mens.1"

TRANSICAO T6 NOME = recebe "rec.1"

TRANSICAO T7 NOME = recebe/libera "mens.0"

TRANSICAO T8 NOME = transmite "rec.0"

TRANSICAO T9 NOME = recebe/rejeita "mens.0"

TRANSICAO T10 NOME = recebe/libera "mens.1"

```
TRANSICAO   T11   NOME   =   transmite "rec.1"

TRANSICAO   T12   NOME   =   recebe/rejeita "mens.1"

TRANSICAO   T13   NOME   =   perde "mens.0"

TRANSICAO   T14   NOME   =   perde "rec.0"

TRANSICAO   T15   NOME   =   perde "mens.1"

TRANSICAO   T16   NOME   =   perde "rec.1"
```

ANALISE DA RPT - DOCUMENTACAO

NUMERO DE TRANSICOES : 16

NUMERO DE LUGARES : 12

NUMERO TOTAL DE CLASSES : 16

TABELA DAS CLASSES DE ESTADO

(Classe)

(Marcacao = Lugar)

(Dominio : X)

Classe : 0
Marcacao : 1(1) 5(1)
0 =<X1=< 999

Classe : 1
Marcacao : 2(1) 5(1) 9(1)
5 =<X2=< 6
0 =<X7=< 1
0 =<X13=< 1

Classe : 2
Marcacao : 2(1) 6(1)
4 =<X2=< 6
0 =<X8=< 2

Classe : 3
Marcacao : 2(1) 5(1)
0 =<X2=< 2

Classe : 4
Marcacao : 2(1) 7(1) 10(1)
2 =<X2=< 6
0 =<X3=< 1
0 =<X14=< 1

```
Classe :  5
Marcacao :  3(1)  7(1)
0 =<X4=< 999


Classe :  6
Marcacao :  2(1)  7(1)
0 =<X2=< 5


Classe :  7
Marcacao :  4(1)  7(1)  11(1)
5 =<X5=< 6
0 =<X10=< 1
0 =<X15=< 1


Classe :  8
Marcacao :  2(1)  7(1)  9(1)
5 =<X2=< 6
0 =<X9=< 1
0 =<X13=< 1


Classe :  9
Marcacao :  4(1)  8(1)
4 =<X5=< 6
0 =<X11=< 2


Classe :  10
Marcacao :  4(1)  7(1)
0 =<X5=< 2


Classe :  11
Marcacao :  2(1)  7(1)
0 =<X2=< 2


Classe :  12
Marcacao :  4(1)  5(1)  12(1)
2 =<X5=< 6
0 =<X6=< 1
0 =<X16=< 1


Classe :  13
Marcacao :  4(1)  5(1)
0 =<X5=< 5
```

```
Classe :  14
Marcacao :  4(1)  5(1)  11(1)
5 =<X5=< 6
0 =<X12=< 1
0 =<X15=< 1


Classe :  15
Marcacao :  4(1)  5(1)
0 =<X5=< 2


GRAFO DAS CLASSES DE ESTADO
----- --- ------- -- ------

          (Classe ---> Transicao que dispara/Proxima Classe)

     C  0 --->t  1/C  1,
     C  1 --->t  7/C  2,       C  1 --->t 13/C  3,
     C  2 --->t  8/C  4,
     C  3 --->t  2/C  1,
     C  4 --->t  3/C  5,       C  4 --->t 14/C  6,
     C  5 --->t  4/C  7,
     C  6 --->t  2/C  8,
     C  7 --->t 10/C  9,       C  7 --->t 15/C 10,
     C  8 --->t  9/C  2,       C  8 --->t 13/C 11,
     C  9 --->t 11/C 12,
     C 10 --->t  5/C  7,
     C 11 --->t  2/C  8,
     C 12 --->t  6/C  0,       C 12 --->t 16/C 13,
     C 13 --->t  5/C 14,
     C 14 --->t 12/C  9,       C 14 --->t 15/C 15,
     C 15 --->t  5/C 14,




REDE          LIMITADA,    REINICIAVEL  E     VIVA
```