



COOPERAÇÃO DE SISTEMAS A EVENTOS DISCRETOS EM REDE COM ANÁLISE DE BLOQUEIO

Lucas Antunes Floriano

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadores: Lilian Kawakami Carvalho
Gustavo da Silva Viana

Rio de Janeiro
Janeiro de 2022

COOPERAÇÃO DE SISTEMAS A EVENTOS DISCRETOS EM REDE COM
ANÁLISE DE BLOQUEIO

Lucas Antunes Floriano

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO
ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE
ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU
DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Orientadores: Lilian Kawakami Carvalho
Gustavo da Silva Viana

Aprovada por: Prof. Lilian Kawakami Carvalho, D.Sc.
Prof. Gustavo da Silva Viana, D.Sc.
Prof. Carlos Andrey Maia, D.Sc.
Prof. Paulo Eigi Miyagi, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
JANEIRO DE 2022

Floriano, Lucas Antunes

Cooperação de sistemas a eventos discretos em rede com análise de bloqueio /Lucas Antunes Floriano. – Rio de Janeiro: UFRJ/COPPE, 2022.

X, 65 p.: il.; 29, 7cm.

Orientadores: Lilian Kawakami Carvalho

Gustavo da Silva Viana

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2022.

Referências Bibliográficas: p. 63 – 65.

1. Sistemas a Eventos Discretos. 2. Redes de Petri.
3. Lugares Comuns. 4. Cooperação de sistemas em rede.
5. Análise de bloqueio. I. Carvalho, Lilian Kawakami *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

COOPERAÇÃO DE SISTEMAS A EVENTOS DISCRETOS EM REDE COM ANÁLISE DE BLOQUEIO

Lucas Antunes Floriano

Janeiro/2022

Orientadores: Lilian Kawakami Carvalho
Gustavo da Silva Viana

Programa: Engenharia Elétrica

Cada vez mais, os desafios da indústria exigem flexibilidade e autonomia por parte dos sistemas envolvidos. Neste contexto, temos os sistemas ciber-físicos, caracterizados pela junção de sistemas cibernéticos, redes de comunicação e sistemas físicos. Um dos desafios envolvendo sistemas ciber-físicos é o problema de cooperação de sistemas independentes, em que temos como objetivo fazer com que diversos sistemas, que já possuem alguma autonomia, comecem a trabalhar em conjunto. Neste trabalho, o problema de cooperação de sistemas a eventos discretos em rede é formulado, em que um conjunto de sistemas independentes conectados em rede, denominados módulos, interagem entre si compartilhando recursos. Inicialmente, consideramos que cada um dos módulos opera sozinho, cumprindo suas respectivas especificações locais. Então nosso objetivo é sintetizar um sistema em cooperação em rede, em que os módulos cumpram tarefas específicas em conjunto, sem deixar de desempenhar suas funções locais. Para solucionar o problema proposto, são apresentados dois algoritmos para realizar a cooperação de sistemas a eventos discretos modelados por redes de Petri, utilizando lugares comuns. Em seguida, uma análise de bloqueios é realizada a partir do sistema em cooperação em rede. Por fim, é apresentado um caso de estudo em uma planta mecatrônica didática para montagem de cubos.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

COOPERATION OF NETWORKED DISCRETE EVENT SYSTEMS WITH DEADLOCK ANALYSIS

Lucas Antunes Floriano

January/2022

Advisors: Lilian Kawakami Carvalho
Gustavo da Silva Viana

Department: Electrical Engineering

The challenges faced by the industry require more and more flexibility and autonomy from the involved systems. In this context, we have cyber-physical systems that are characterized by the combination of digital systems, communication networks and physical systems. One of the challenges of cyber-physical systems is the cooperation of independent systems, where many systems that have already had some autonomy work together. In this work, we address the problem of cooperation of networked discrete event systems, where a group of network connected independent systems work together sharing resources. Initially, we will consider that each module works alone, executing its respective local specifications. Our goal is to build a networked cooperating system where the modules can execute a global task together while also executing their local tasks. To solve this problem, we present two algorithms to implement the cooperation of networked discrete event systems modeled by Petri nets, using common places. After that, we make an analysis of deadlocks on the resulting cooperating networked system. Finally, we present a case study in a cube assembling mechatronic plant.

Sumário

Lista de Figuras	viii
Lista de Tabelas	x
1 Introdução	1
2 Fundamentos Teóricos	4
2.1 Eventos e Linguagens	4
2.2 Redes de Petri	5
2.2.1 Propriedades gerais de redes de Petri	5
2.2.2 Lugares comuns em redes de Petri	10
2.2.3 Controle supervisorio em redes de Petri	13
3 Cooperação de sistemas a eventos discretos em rede	16
3.1 Formulação do problema	17
3.1.1 Arquitetura considerada	17
3.1.2 Especificações locais	18
3.2 Especificações globais do sistema em cooperação	18
3.2.1 Cooperação por habilitação	18
3.2.2 Cooperação por impedimento	20
3.3 Síntese do sistema em cooperação com lugares comuns	22
3.3.1 Síntese do sistema em cooperação por habilitação utilizando lugares comuns	24
3.3.2 Síntese do sistema em cooperação por impedimento utilizando lugares comuns	30
3.4 Análise de bloqueio do sistema em cooperação com lugares comuns .	36
4 Aplicação prática	42
4.1 Planta mecatrônica	42
4.1.1 Módulo 1	42
4.1.2 Módulo 2	44
4.1.3 Módulo 3	46

4.1.4	Módulos 4 e 5	46
4.2	Modelagem dos módulos	47
4.3	Cooperações implementadas	50
5	Conclusão e trabalhos futuros	62
	Referências Bibliográficas	63

Lista de Figuras

2.1	Rede de Petri do Exemplo 2.1	5
2.2	Rede de Petri do Exemplo 2.3	7
2.3	Redes de Petri do exemplo 2.6	11
2.4	Redes de Petri do exemplo 2.7	12
2.5	Redes de Petri do exemplo 2.8	14
3.1	Arquitetura do sistema em cooperação em rede	16
3.2	Redes de Petri do exemplo 3.3	23
3.3	Redes de Petri do exemplo 3.4	27
3.4	Redes de Petri do exemplo 3.5	33
3.5	Redes de Petri do Exemplo 3.6: Módulos N_1 e N_2 em preto e lugares comuns p_{c1} e p_{c2} em vermelho.	37
3.6	Redes de Petri do Exemplo 3.7: Módulos N_1 e N_2 em preto e lugar comum p_{c1} em vermelho.	39
4.1	Planta mecatrônica disponibilizada no Laboratório de Controle e Automação (LCA) da UFRJ	43
4.2	Peças plásticas que são montadas na planta	43
4.3	Fluxograma da planta mecatrônica	44
4.4	Módulo 1	45
4.5	Módulo 2	45
4.6	Módulo 3	46
4.7	Rede de Petri do módulo 1	47
4.8	Rede de Petri do módulo 2	48
4.9	Rede de Petri do módulo 3	49
4.10	Rede de Petri do módulo 4	50
4.11	Rede de Petri do módulo 5	52
4.12	Representação da cooperação entre os módulos 1 e 2, respectivamente	53
4.13	Representação da cooperação entre os módulos 4 e 5, respectivamente	54
4.14	Representação da cooperação por habilitação entre os módulos 2 e 3, respectivamente	55

4.15	Representação da cooperação por habilitação entre os módulos 5 e 3, respectivamente	56
4.16	Representação da cooperação por impedimento entre os módulos 2 e 5, respectivamente	58
4.17	Representação da cooperação por impedimento entre os módulos 5 e 2, respectivamente	59
4.18	Sistema em cooperação completo	60

Lista de Tabelas

4.1	Condições e ações da rede de Petri do módulo 1	47
4.2	Condições e ações da rede de Petri do módulo 2	49
4.3	Condições e ações da rede de Petri do módulo 3	50
4.4	Condições e ações da rede de Petri do módulo 4	51
4.5	Condições e ações da rede de Petri do módulo 5	51

Capítulo 1

Introdução

Nos últimos anos temos vivenciado uma grande mudança nos paradigmas do meio industrial. Cada vez mais, os desafios da indústria exigem flexibilidade e autonomia por parte dos sistemas envolvidos, além de terem também um foco muito grande na comunicação entre estes sistemas. Este é o cenário da Indústria 4.0 [1] que tem recebido muita atenção na literatura [2, 3]. Neste contexto, temos os sistemas ciber-físicos, caracterizados pela junção de sistemas cibernéticos, redes de comunicação e sistemas físicos.

Um dos desafios envolvendo sistemas ciber-físicos é o problema de cooperação de sistemas independentes, em que temos como objetivo fazer com que diversos sistemas que já possuem alguma autonomia comecem a trabalhar em conjunto. Estes sistemas podem estar próximos fisicamente, compartilhando recursos entre si [4], mas podem também estar em lugares completamente diferentes, mas conectados em rede e compartilhando informações [5]. Este problema é bastante explorado na literatura, e dentre as formas de representar os sistemas envolvidos, temos os sistemas a eventos discretos (SEDs), que podem ser modelados formalmente por redes de Petri (RPs) [6–9].

Dentre as estratégias utilizadas para abordar o problema de cooperação de SEDs, o controle supervisorio se destaca como uma das mais exploradas na literatura. Os estudos em controle supervisorio giram em torno da criação de um controlador que irá operar em cima do sistema a ser controlado, limitando a sua dinâmica e garantindo o comportamento desejado. Em GIUA [10], a teoria de controle supervisorio em redes de Petri é desenvolvida. Quando um controlador é sintetizado para operar em cima de mais de um sistema simultaneamente, damos a ele o nome de coordenador. Em automatos, este problema é abordado nos seguintes trabalhos [11–14]. Um dos problemas existentes no controle supervisorio é o tamanho dos modelos dos sistemas envolvidos, de forma que a confecção de múltiplos controladores pode resultar em um crescimento não desejado no modelo do sistema global. Além disso, uma vez que os controladores tenham sido modelados, a implementação de uma nova

regra de controle impactará de forma ainda mais extrema no tamanho do modelo do sistema.

Um outro problema que surge com a cooperação de sistemas independentes está na possível criação de bloqueios indesejados nas dinâmicas envolvidas. Idealmente, o sistema em cooperação deve ser capaz de executar todas as tarefas que os módulos individuais eram capazes de executar previamente, garantindo que as operações de um sistema não atrapalhem ou bloqueiem as operações dos demais [15]. Em LIU *et al.* [4] é apresentada a definição de colaboração entre sistemas modelados por uma nova classe de redes de Petri, chamadas Redes de Processos Paralelos (P^2Ns)¹, verificando a existência de bloqueios nestes sistemas. Entretanto, o trabalho não apresenta de forma clara como obter um sistema em colaboração a partir de vários sistemas independentes. Em HUANG *et al.* [16] são abordadas estratégias para a prevenção de bloqueios em redes de Petri do tipo S^3PR , que são redes que permitem o compartilhamento de apenas um recurso em qualquer estado de operação. Contudo, o foco do trabalho não é na obtenção do sistema, mas na prevenção dos bloqueios de um sistema já sintetizado.

Em ZGORZELSKI and LUNZE [5] o conceito de sincronização em redes de Petri conectadas em rede é apresentado, que envolve controlar a dinâmica de múltiplas redes independentes para trabalharem em conjunto. A sincronização possui foco em fazer com que múltiplos sistemas evoluam ao mesmo tempo. Esta preocupação com simultaneidade não ocorre no problema de cooperação.

Neste trabalho, o problema de cooperação de sistemas a eventos discretos em rede é formulado, em que um conjunto de sistemas independentes conectados em rede, denominados módulos, interagem entre si compartilhando recursos. Uma das vantagens desta estratégia será contornar a necessidade da criação de coordenadores, agregando informações diretamente na estrutura dos módulos envolvidos. Inicialmente, consideramos que cada um dos módulos opera sozinho, cumprindo suas respectivas especificações locais. Então nosso objetivo é sintetizar um sistema em cooperação em rede, em que os módulos cumpram tarefas específicas em conjunto, sem deixar de desempenhar suas funções locais. Para solucionar o problema proposto, são apresentados dois algoritmos para realizar a cooperação de sistemas a eventos discretos modelados por redes de Petri, utilizando lugares comuns, cada um implementando um tipo de cooperação: habilitação e impedimento, que são relacionados às especificações globais do sistema. Em seguida, uma análise de bloqueios é realizada a partir do sistema em cooperação em rede. Por fim, é apresentado um caso de estudo em uma planta mecatrônica para montagem de cubos.

O trabalho será dividido nos seguintes capítulos. No capítulo 2, serão apresen-

¹ P^2Ns é um conjunto de redes de Petri, interligadas por um subconjunto de lugares, que possuem um estado inicial e um estado final bem definidos

tados os conceitos teóricos já explorados na literatura que serão usados ao longo do trabalho, tais como redes de Petri, lugares comuns, bloqueios em rede de Petri, sifões e controle supervisorio. No capítulo 3, será apresentada a formulação do problema, assim como as duas cooperações que serão o foco do trabalho e as metodologias criadas para implementar estas cooperações e a correlação destas com eventuais bloqueios nas redes envolvidas. No capítulo 4, será apresentada a implementação dos algoritmos propostos em uma planta mecatrônica de montagem de cubos. No capítulo 5, apresentamos a conclusão deste trabalho, assim como propostas de trabalhos futuros.

Capítulo 2

Fundamentos Teóricos

Neste capítulo, serão apresentados todos os fundamentos teóricos explorados nesse trabalho. Primeiramente, na seção 2.1 serão apresentados os conceitos de eventos e linguagem. Na seção 2.2 será apresentada a teoria de redes de Petri, com um foco maior em sílfões e lugares comuns. Finalmente, na seção 2.3 será apresentada uma revisão de controle supervisório em redes de Petri.

2.1 Eventos e Linguagens

Em sistemas a eventos discretos, a mudança de um estado do sistema para outro se dá através da ocorrência de eventos. Estes eventos são usualmente representados por uma letra ou um número e ocorrem ao longo do tempo.

Uma sequência de eventos representa uma ordenação no tempo na qual o primeiro evento da sequência foi o primeiro a ocorrer, seguido pelo segundo e assim por diante. Sejam a e b dois eventos distintos, podemos escrever por exemplo as sequências aab , $bbbb$ ou a para descrever a ordem em que estes dois eventos ocorreram em determinado sistema. Dadas duas sequências s_1 e s_2 , dizemos que a sequência $s_3 = s_1s_2$ é a concatenação de s_1 e s_2 . A sequência vazia é definida como ε .

Dado um conjunto de eventos Σ , chamamos de linguagem o conjunto de sequências finitas construídas com os eventos de Σ . Note que, embora as sequências contidas na linguagem devem ser finitas, a linguagem em si não precisa ser um conjunto finito. Seja Σ o conjunto de eventos $\Sigma = \{a, b, c\}$, podemos construir por exemplo as linguagens $L_1 = \{a, aa, aaa\}$ e $L_2 = \{\text{Todas as sequências de tamanho finito iniciadas com } a\}$. Uma definição importante para lidar com linguagens é o prefixo-fechamento, que será mostrado a seguir.

Definição 2.1 (*Prefixo e prefixo-fechado*) *Seja $s = uv$ uma sequência resultada da concatenação das sequências u e v , dizemos que u é um prefixo de s e v é um sufixo*

de s . Dizemos que uma linguagem L é prefixo-fechada se todos os prefixos possíveis de todas as sequências de L pertencem a L também.

Após as definições básicas de linguagem e eventos, uma revisão da teoria de redes de Petri será apresentada a seguir.

2.2 Redes de Petri

2.2.1 Propriedades gerais de redes de Petri

Uma rede de Petri é um grafo orientado bipartido [17], [18], ou seja, possui dois tipos de vértices que nunca são ligados entre si. Em outras palavras, uma rede de Petri é um dispositivo que manipula eventos respeitando regras bem definidas, de acordo com o comportamento do sistema modelado [19].

Ela possibilita modelar comportamentos como paralelismo, concorrência, sincronização e compartilhamento de recursos [6]. O grafo de uma rede de Petri é definido a seguir.

Definição 2.2 (*Grafo de uma rede de Petri*) Um grafo de uma rede de Petri é definido como um grafo bipartido (P, T, F, W) , em que P representa o conjunto finito de lugares, representados graficamente por um círculo, T é o conjunto finito de transições, representadas graficamente por uma linha, $F \subseteq (P \times T) \cup (T \times P)$ é o conjunto de arcos ordinários que ligam lugares a transições e vice-versa e $W : F \rightarrow \mathbb{N}^*$ é uma função associada a cada arco que atribui à ele um peso.

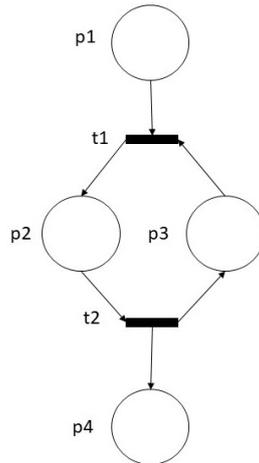


Figura 2.1: Rede de Petri do Exemplo 2.1

Exemplo 2.1 A rede de Petri $N = (P, T, F, W)$ está representada graficamente na figura 2.1, em que $P = \{p_1, p_2, p_3, p_4\}$ é o conjunto de lugares, $T = \{t_1, t_2\}$ é o

conjunto de transições e $F = \{(p_1, t_1), (t_1, p_2), (p_2, t_2), (t_2, p_3), (p_3, t_1), (t_2, p_4)\}$ é o conjunto de arcos. Note que (p_1, t_1) representa um arco direcionado partindo de p_1 para t_1 . Finalmente, os pesos dos arcos são definidos como $W(p_1, t_1) = W(t_1, p_2) = W(p_2, t_2) = W(t_2, p_3) = W(p_3, t_1) = W(t_2, p_4) = 1$, ou seja, todos os arcos possuem peso 1.

A matriz de incidência é uma matriz que possibilita a análise do comportamento dinâmico das redes de Petri e é definida a seguir.

Definição 2.3 (Matriz de incidência) Uma rede de Petri $N = (P, T, F, W)$ pode ser representada por uma matriz de incidência $[N]$, em que $[N]$ é uma matriz $|P| \times |T|$ com $[N](p, t) = W(t, p) - W(p, t)$.

O conjunto de nós sucessores a um nó $x \in (P \cup T)$ será definido por $x\bullet = \{y \in (P \cup T) | \exists (x, y) \in F\}$ enquanto o conjunto de nós antecessores à x será definido por $\bullet x = \{y \in (P \cup T) | \exists (y, x) \in F\}$. Na rede de Petri da figura 2.1, o nó p_2 possui como antecessor t_1 e como sucessor t_2 . Note que se o nó for um lugar, seus antecessores e sucessores sempre serão transições, e vice-versa.

Seja S um conjunto de nós, o conjunto $S\bullet$ é definido como o conjunto de todos os sucessores de cada elemento de S , formalmente $S\bullet := \bigcup_{x \in S} x\bullet$. De forma similar, $\bullet S$ é definido como o conjunto de todos os antecessores de cada elemento de S , isto é, $\bullet S := \bigcup_{x \in S} \bullet x$. Com isso, podemos apresentar a definição de um sifão em uma rede de Petri.

Definição 2.4 (Sifão) Um sifão é um conjunto de lugares S de uma rede de Petri N , tal que $\bullet S \subseteq S\bullet$.

Em palavras, um sifão é um conjunto de lugares em que toda transição que adiciona ficha a um lugar do conjunto, também retira ficha de um lugar do conjunto. Desta forma, o número de fichas dentro de um sifão nunca aumenta, apenas diminui ou se mantém constante.

Exemplo 2.2 A rede de Petri, mostrada na figura 2.1, possui 7 sifões: $S_1 = \{p_1\}$, $S_2 = \{p_2, p_3\}$, $S_3 = \{p_1, p_2\}$, $S_4 = \{p_2, p_3, p_4\}$, $S_5 = \{p_1, p_2, p_3\}$, $S_6 = \{p_1, p_2, p_4\}$ e $S_7 = \{p_1, p_2, p_3, p_4\}$. Note que para o conjunto S_1 , o conjunto de sucessores é $S_1\bullet = \{t_1\}$, enquanto o conjunto de antecessores é vazio, isto é, $\bullet S_1 = \emptyset$. Uma vez que $\bullet S_1 \subseteq S_1\bullet$, S_1 é um sifão. Considere agora o conjunto S_2 , em que $S_2\bullet = \{t_1, t_2\}$ e $\bullet S_2 = \{t_1, t_2\}$. Novamente, $\bullet S_2 \subseteq S_2\bullet$, então S_2 também é um sifão. É importante ressaltar que nem todo conjunto de lugares forma um sifão. Seja o conjunto $S' = \{p_1, p_4\}$, em que $S'\bullet = \{t_1\}$ e $\bullet S' = \{t_2\}$. Como $\bullet S' \not\subseteq S'\bullet$, S' não é um sifão.

Para que uma rede de Petri represente um sistema dinâmico, adicionamos a ela uma marcação, que será representada por um vetor coluna $x = [x(p_1) \ x(p_2) \ \dots \ x(p_n)]^T$. Graficamente, esta marcação será representada por um número de fichas $x(p_i) \in \mathbb{N}$ distribuídas dentro dos lugares.

Definição 2.5 Uma marcação M de uma rede de Petri $N = (P, T, F, W)$ é um mapeamento de P para \mathbb{N} , em que $M(p)$ representa o número de fichas no lugar $p \in P$ [20]. Chamamos de rede marcada a tupla (N, M_0) , onde M_0 é um vetor de tamanho $|P|$ que representa a marcação inicial de N .

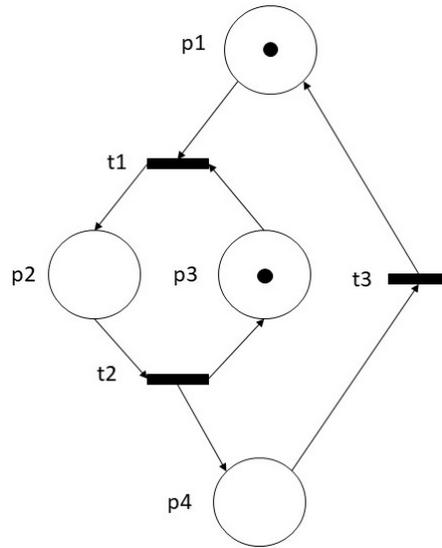


Figura 2.2: Rede de Petri do Exemplo 2.3

Exemplo 2.3 Na figura 2.2 temos uma rede de Petri marcada (N, M_0) , em que $N = (P, T, F, W)$, $P = \{p_1, p_2, p_3, p_4\}$, $T = \{t_1, t_2, t_3\}$, $F = \{(p_1, t_1), (t_1, p_2), (p_2, t_2), (t_2, p_3), (p_3, t_1), (t_2, p_4), (p_4, t_3), (t_3, p_1)\}$, função de peso W igual a 1 para todos os arcos definidos e marcação inicial $M_0 = [1 \ 0 \ 1 \ 0]^T$. No vetor M_0 , a primeira posição representa a quantidade de fichas no lugar p_1 , a segunda posição representa a quantidade de fichas no lugar p_2 e assim por diante. Note que a marcação M_0 indica que temos inicialmente uma ficha em p_1 e uma ficha em p_3 .

Dado um subconjunto $S \subseteq P$, a soma de todas as fichas de todos os lugares de S será denotada como $M(S)$, ou seja, $M(S) = \sum_{p \in S} M(p)$. Diremos que S é vazio se $M(S) = 0$. Estas fichas possuirão a função de habilitar o disparo de uma transição, que por sua vez mudará a marcação da rede assim como seu estado, criando uma dinâmica.

Definição 2.6 (*Habilitação de transições*) Dizemos que uma transição $t \in T$ está habilitada para uma marcação M se $\forall p \in \bullet t, M(p) \geq W(p, t)$, ou seja, se todos os lugares antecessores àquela transição possuem uma quantidade de fichas maior ou igual ao peso do arco ligando aquele lugar à transição. Com o disparo de t , obtemos uma nova marcação M' tal que $\forall p \in P, M'(p) = M(p) - W(p, t) + W(t, p)$.

Em palavras, o disparo de uma transição retira fichas dos lugares antecessores àquela transição e coloca fichas nos lugares sucessores. A quantidade de fichas retiradas e colocadas é igual ao peso do arco ligando a transição e o lugar. Na rede de Petri da figura 2.2, caso a transição t_1 dispare, as fichas de p_1 e p_3 são retiradas, e uma ficha será criada em p_2 . Note que a nova marcação da rede será $M_0 = [0 \ 1 \ 0 \ 0]^T$.

O disparo de uma transição t , assim como sua mudança de marcação indo de M para M' podem ser representados como $M[t]M'$. Uma marcação M'' é dita ser alcançável partindo de M se existirem transições t_0, t_1, \dots, t_n e marcações M_1, M_2, \dots, M_n tal que $M[t_0]M_1[t_1]M_2 \dots M_n[t_n]M''$ seja possível. O conjunto de todas as marcações possíveis para uma rede N com marcação M é chamado de conjunto alcançável e é representado por $R(N, M)$. Note que este conjunto não precisa ser finito.

Em um caso mais geral, podemos adicionar às redes de Petri um arco especial, chamado arco inibidor. Tais redes são chamadas redes de Petri estendidas e serão definidas a seguir.

Definição 2.7 (*Redes de Petri estendidas*) Uma rede de Petri marcada estendida é definida como uma quintupla $N_I = (P, T, F, W, I)$, em que P, T, F, W e M_0 são definidos da mesma forma que na definição 2.5 e $I : (P \times T) \rightarrow \mathbb{N}$ representa o conjunto de arcos inibidores ligando lugares a transições.

Um arco inibidor será representado graficamente como uma linha conectando o lugar a transição, com um círculo na extremidade ligada à transição. O arco inibidor não terá influência nenhuma na dinâmica da rede enquanto não houver fichas em seu lugar. A partir do momento que a quantidade de fichas no lugar de entrada do arco inibidor for igual ou maior que o peso do arco, sua transição será impossibilitada de disparar, independente da marcação dos demais lugares da rede.

Definição 2.8 (*Sequência de disparos*) Definimos como sequência de disparo de transições a sequência $\pi = t_0 t_1 \dots t_n$ [18]. Para uma rede de Petri N , a sequência π será dita possível partindo de uma marcação M se for possível disparar, em ordem, cada uma das transições da sequência (sem o disparo de quaisquer outras transições). Neste trabalho chamaremos de $\Pi(N, M)$ o conjunto de todas as sequências π possíveis de uma rede N partindo de uma marcação M .

Note que, dada uma rede de Petri $N = (P, T, F, W)$ com marcação inicial M , podemos interpretar uma sequência de disparos π de N como uma sequência de eventos, em que o conjunto de eventos Σ é igual ao conjunto de transições T . Com esta interpretação, o conjunto $\Pi(N, M)$ representa a linguagem gerada pela rede de Petri N . Note que esta linguagem sempre será prefixo-fechada, visto que para que uma sequência π seja possível de ocorrer na rede N , todos os seus prefixos devem ser possíveis também.

Considerando a definição 2.8, uma outra forma de obter a nova marcação M' partindo de M , após uma sequência π , é através da matriz de incidência $[N]$. A nova marcação M' será $M' = M + [N]\vec{r}$, em que \vec{r} representa o vetor coluna de tamanho $|T|$ em que o j -ésimo componente corresponde ao número de vezes que a transição t_j disparou em π .

Com as definições de marcação e disparo de transições, podemos definir quando uma rede se encontrará em estado de bloqueio.

Definição 2.9 (*Estado de bloqueio*) Dizemos que uma rede de Petri N com marcação M está em estado de bloqueio quando nenhuma transição de N pode disparar.

Uma vez definido o conjunto alcançável de marcações $R(N, M)$, podemos então apresentar a definição de sifão controlável.

Definição 2.10 (*Sifão controlável*) Dizemos que um sifão S é controlável se e somente se $\forall M \in R(N, M_0), M(S) > 0$. Em outras palavras, um sifão será controlável se para qualquer marcação alcançável partindo de uma marcação inicial M_0 o sifão possua ao menos uma ficha dentro dele.

Exemplo 2.4 A rede de Petri da figura 2.2 possui os seguintes sifões: $S_1 = \{p_2, p_3\}$, $S_2 = \{p_2, p_3, p_4\}$, $S_3 = \{p_1, p_2, p_4\}$ e $S_4 = \{p_1, p_2, p_3, p_4\}$. Com marcação inicial $M_0 = [1 \ 0 \ 1 \ 0]^T$, podemos observar a sequência de disparos $\pi = t_1 t_2 t_3$. Essa sequência resultará na seguinte dinâmica: $M_0[t_1]M_1[t_2]M_2[t_3]M_0$, em que $M_1 = [0 \ 1 \ 0 \ 0]^T$ e $M_2 = [0 \ 0 \ 1 \ 1]^T$. Note que para todas as marcações alcançáveis partindo de M_0 , $M(\{p_2, p_3\}) > 0$ e $M(\{p_1, p_2, p_4\}) > 0$. Ou seja, os conjuntos de lugares $\{p_2, p_3\}$ e $\{p_1, p_2, p_4\}$ contém ao menos uma ficha para qualquer marcação alcançável a partir de M_0 . Como S_1 , S_2 e S_4 contém os lugares $\{p_2, p_3\}$, então os três sifões serão controláveis, visto que sempre existirá ao menos uma ficha dentro dos sifões. O mesmo vale para S_3 com o conjunto de lugares $\{p_1, p_2, p_4\}$, de forma que S_3 também será um sifão controlável.

Teorema 2.1 Uma rede de Petri (N, M_0) com um conjunto de sifões \mathcal{S} é livre de bloqueio quando todos os seus sifões forem controláveis, ou seja $\forall S \in \mathcal{S}, \forall M \in R(N, M_0), M(S) > 0$ ([20]).

De acordo com o teorema 2.1, se todos os sifões da rede possuírem ao menos uma ficha para qualquer marcação alcançável, então a rede é livre de bloqueio. Em outras palavras, para qualquer sequência de disparo de transições π_1 de uma rede (N, M_0) , sempre existirá uma sequência π_2 tal que $\pi_1\pi_2 \in R(N, M_0)$. Note, contudo, que o contrário não será necessariamente verdade, ou seja, uma rede pode ser livre de bloqueios e conter um sifão não-controlável.

Teorema 2.2 *Seja N uma rede de Petri que, para uma marcação M , se encontra em um bloqueio. Então o conjunto $S = \{p \in P | M(p) = 0\}$ é um sifão ([20]).*

Em palavras, para que exista um bloqueio em uma rede de Petri, é necessário que exista um sifão não-controlável na rede que, em determinado momento de sua dinâmica, perderá todas as suas fichas. Note, contudo, que a existência de um sifão vazio não implica necessariamente em um bloqueio na rede.

Definição 2.11 *(Sifão mínimo) Um sifão é dito mínimo se ele não contiver outro sifão com menos lugares.*

A partir da definição 2.11 e do teorema 2.1 podemos enunciar o seguinte corolário.

Corolário 2.1 *Seja uma rede de Petri N que possua um sifão não-mínimo S e um sifão mínimo S_{min} . Além disso, suponha que $S_{min} \subset S$ e que S_{min} seja controlável para uma marcação inicial M_0 . Então o sifão S também será controlável.*

Prova. Como $S_{min} \subset S$, teremos que $\forall M \in R(N, M_0), M(S) \geq M(S_{min})$. Como S_{min} é controlável, $\forall M \in R(N, M_0), M(S_{min}) > 0$. Consequentemente, $\forall M \in R(N, M_0), M(S) > 0$, o que significa que o sifão S é controlável. ■

Exemplo 2.5 *Como visto no exemplo 2.4, a rede de Petri da figura 2.2 possui os seguintes sifões: $S_1 = \{p_2, p_3\}$, $S_2 = \{p_2, p_3, p_4\}$, $S_3 = \{p_1, p_2, p_4\}$ e $S_4 = \{p_1, p_2, p_3, p_4\}$. Como S_1, S_2, S_3 e S_4 são controláveis (exemplo 2.4), a rede da figura 2.2 será livre de bloqueios para uma marcação inicial $M_0 = [1 \ 0 \ 1 \ 0]^T$, de acordo com o teorema 2.1. Note que, de acordo com a definição 2.1, os sifões S_1 e S_3 são mínimos. Como uma consequência do corolário 2.11, é suficiente que S_1 e S_3 sejam controláveis para que a rede de Petri N seja livre de bloqueios para a marcação inicial M_0 .*

2.2.2 Lugares comuns em redes de Petri

Nesta seção apresentaremos a definição de lugares comuns, que serão usados na definição de Rede de Petri com lugares de fronteira [21].

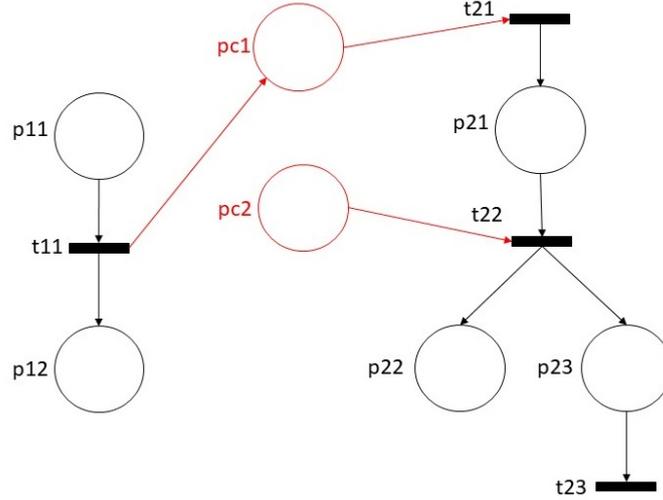


Figura 2.3: Redes de Petri do exemplo 2.6

Definição 2.12 (*Lugares comuns*) Sejam duas redes $N_1 = (P_1, T_1, F_1, W_1)$ e $N_2 = (P_2, T_2, F_2, W_2)$, define-se como conjunto de lugares comuns o conjunto $P_{12} = P_1 \cap P_2$.

Em palavras, um lugar comum é um lugar compartilhado entre N_1 e N_2 . A partir da definição 2.12, o seguinte teorema pode ser apresentado.

Teorema 2.3 Sejam duas redes $N_1 = (P_1, T_1, F_1, W_1)$ e $N_2 = (P_2, T_2, F_2, W_2)$ tais que o conjunto de lugares comuns $P_{12} \neq \emptyset$ e $T_1 \cap T_2 = \emptyset$. Então:

$$\forall p \in P_1 \setminus P_{12}, \forall t \in T_2 \Rightarrow (p, t) \notin (F_1 \cup F_2) \wedge (t, p) \notin (F_1 \cup F_2), \quad (2.1)$$

$$\forall p \in P_2 \setminus P_{12}, \forall t \in T_1 \Rightarrow (p, t) \notin (F_2 \cup F_1) \wedge (t, p) \notin (F_2 \cup F_1), \quad (2.2)$$

em que \setminus representa subtração de conjuntos.

Prova. Suponha que $\exists p \in P_1 \setminus P_{12}$ e $\exists t \in T_2$ tal que $(p, t) \in F_1$, ou seja, existe um lugar da rede N_1 que não é um lugar comum e que se conecta a uma transição t de N_2 . De acordo com a definição 2.2, $F_1 \subseteq (P_1 \times T_1) \cup (T_1 \times P_1)$ e $F_2 \subseteq (P_2 \times T_2) \cup (T_2 \times P_2)$. Sendo assim, $(p, t) \in ((P_1 \times T_1) \cup (T_1 \times P_1))$, o que é uma contradição, visto que por hipótese $t \notin T_1$. O mesmo será válido para um arco no sentido contrário, ou seja, $(t, p) \in F_1$. Sendo assim, podemos concluir que não podem existir lugares que não sejam lugares comuns e que se liguem com transições de outra rede. ■

Exemplo 2.6 A figura 2.3 mostra duas redes de Petri, $N_1 = (P_1, T_1, F_1, W_1)$ e $N_2 = (P_2, T_2, F_2, W_2)$ em que $P_1 = \{p_{11}, p_{12}, p_{c1}, p_{c2}\}$, $T_1 = \{t_{11}\}$, $F_1 = \{(p_{11}, t_{11}), (t_{11}, p_{12}), (t_{11}, p_{c1})\}$, $P_2 = \{p_{21}, p_{22}, p_{23}, p_{c1}, p_{c2}\}$, $T_2 = \{t_{21}, t_{22}, t_{23}\}$, $F_2 = \{(t_{21}, p_{21}), (p_{21}, t_{22}), (t_{22}, p_{22}), (t_{22}, p_{23}), (p_{23}, t_{23}), (p_{c1}, t_{21}), (p_{c2}, t_{22})\}$, tal que todos os arcos definidos têm peso igual a 1. De acordo com a definição 2.12, $P_{12} = \{p_{c1}, p_{c2}\}$.

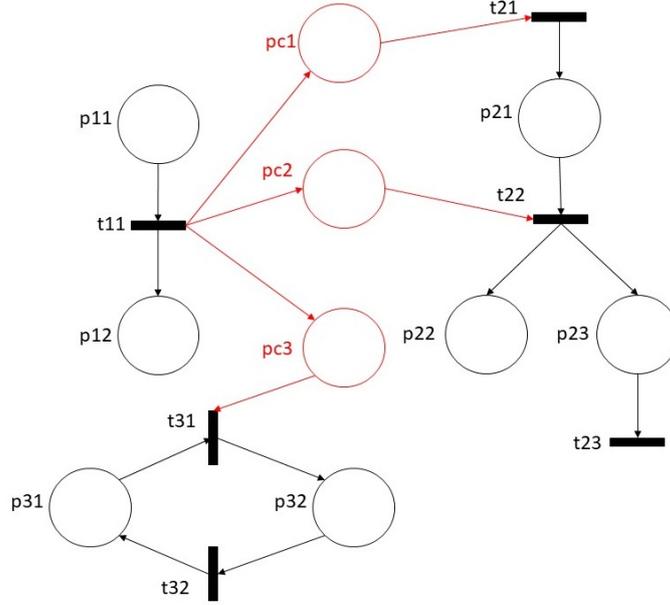


Figura 2.4: Redes de Petri do exemplo 2.7

De acordo com o teorema 2.3, o arco (p_{c1}, t_{21}) pode existir, visto que um dos elementos é um lugar comum. Por outro lado, se um arco (p_{11}, t_{21}) existir, violaria a hipótese inicial do teorema de que $T_1 \cap T_2 = \emptyset$, visto que $p_{11} \in P_1 \setminus P_{12}$ e $t_{21} \in T_2$.

Com isso, podemos definir as redes de Petri com lugares de fronteira [21].

Definição 2.13 (Redes de Petri com lugares de fronteira) Um conjunto de redes de Petri com lugares de fronteira é definido por $\mathcal{F} = \{(N_z, \mathcal{P}_z) : z = 1, 2, \dots, K, K \in \mathbb{N}\}$, em que $N_z = (P_z, T_z, F_z, W_z)$ é uma rede de Petri e $\mathcal{P}_z = \{P_{zi} \subseteq P_z : i = 1, 2, \dots, K \text{ e } i \neq z\}$ é um conjunto P_{zi} , que por sua vez é o subconjunto de lugares comuns entre os módulos N_z e N_i .

Exemplo 2.7 Na figura 2.4, podemos ver um sistema composto por três redes de Petri com lugar de fronteira. Estas redes são $N_1 = (P_1, T_1, F_1, W_1)$, $N_2 = (P_2, T_2, F_2, W_2)$ e $N_3 = (P_3, T_3, F_3, W_3)$ com $P_1 = \{p_{11}, p_{12}, p_{c1}, p_{c2}, p_{c3}\}$, $T_1 = \{t_{11}\}$, $F_1 = \{(p_{11}, t_{11}), (t_{11}, p_{12}), (t_{11}, p_{c1}), (t_{11}, p_{c2}), (t_{11}, p_{c3})\}$ para a primeira rede, $P_2 = \{p_{21}, p_{22}, p_{23}, p_{c1}, p_{c2}\}$, $T_2 = \{t_{21}, t_{22}, t_{23}\}$, $F_2 = \{(t_{21}, p_{21}), (p_{21}, t_{22}), (t_{22}, p_{22}), (t_{22}, p_{23}), (p_{23}, t_{23}), (p_{c1}, t_{21}), (p_{c2}, t_{22})\}$ para a segunda rede, $P_3 = \{p_{31}, p_{32}, p_{c3}\}$, $T_3 = \{t_{31}, t_{32}\}$ e $F_3 = \{(p_{31}, t_{31}), (t_{31}, p_{32}), (p_{32}, t_{32}), (t_{32}, p_{31}), (p_{c3}, t_{31})\}$ para a terceira rede, com peso 1 em todos os arcos definidos. De acordo com a definição 2.12, os lugares p_{c1} e p_{c2} são lugares comuns compartilhados entre as redes N_1 e N_2 , ou seja, pertencem tanto ao conjunto de lugares P_1 quanto ao conjunto P_2 , enquanto o lugar comum p_{c3} é compartilhado entre as redes N_1 e N_3 , pertencendo tanto ao conjunto P_1 quanto

ao conjunto P_3 . Teremos então que $P_{12} = \{p_{c1}, p_{c2}\}$ e $P_{13} = \{p_{c3}\}$. Portanto, pela definição 2.13 o sistema pode ser descrito por $\mathcal{F} = \{(N_1, \mathcal{P}_1), (N_2, \mathcal{P}_2), (N_3, \mathcal{P}_3)\}$, em que $\mathcal{P}_1 = \{P_{12}, P_{13}\}$, $\mathcal{P}_2 = \{P_{12}\}$ e $\mathcal{P}_3 = \{P_{13}\}$.

2.2.3 Controle supervisorio em redes de Petri

A teoria de controle supervisorio se baseia em projetar um agente que atue em malha fechada sobre uma planta, representando um sistema a ser controlado. Este agente, usualmente chamado de supervisor, é feito a partir de uma especificação, que tem como objetivo impor uma restrição no comportamento de malha aberta da planta, limitando o comportamento da mesma. Uma vez sintetizado, o supervisor atuará na planta restringindo sua dinâmica, impedindo que ela atinja determinados estados que violariam a especificação imposta. Note que a adição do supervisor nunca fará com que a planta consiga alcançar estados que não eram alcançáveis em malha aberta.

O controle supervisorio em redes de Petri é um conceito bem explorado na literatura [10], [22]. No geral, a sintetização de um supervisor em rede de Petri consiste na criação de um conjunto de lugares, chamados lugares *monitores*, que serão responsáveis por habilitar ou desabilitar determinadas transições da rede, limitando o comportamento da planta. Estes lugares monitores forçarão na rede de Petri uma restrição da forma (\vec{w}, k) , em que \vec{w} é um vetor de números inteiros e k é um número inteiro, limitando as marcações alcançáveis e criando um conjunto de marcações *permitidas* tal que:

$$\mathcal{M}(\vec{w}, k) = \{M \in \mathbb{N}^{|P|} \mid \vec{w}^T \cdot M \leq k\} \quad (2.3)$$

Restrições deste tipo são chamadas de Restrições generalizadas mutuamente exclusivas (Generalized Mutual Exclusion Constraint - GMEC) e podem ser forçadas a uma rede qualquer através da criação de um ou mais lugares monitores [10]. O significado prático de uma restrição deste tipo é limitar o número de fichas em um conjunto de lugares, garantindo que a quantidade de fichas em cada lugar multiplicada por seu peso definido em \vec{w} nunca será maior que o valor k para qualquer marcação alcançável.

Exemplo 2.8 *A rede de Petri da figura 2.5 (a) representa um sistema a ser controlado. Queremos impor no sistema uma restrição do tipo $\mathcal{M}(\vec{w}, k)$, como definido anteriormente, em que $\vec{w} = [0 \ 1 \ 1 \ 0]^T$ e $k = 1$. Em palavras, queremos garantir que a soma das fichas nos lugares p_2 e p_3 não seja maior que 1, para qualquer marcação alcançável. O lugar monitor p_m , visto na figura 2.5 (b), representa o supervisor que impõe esta restrição na rede.*

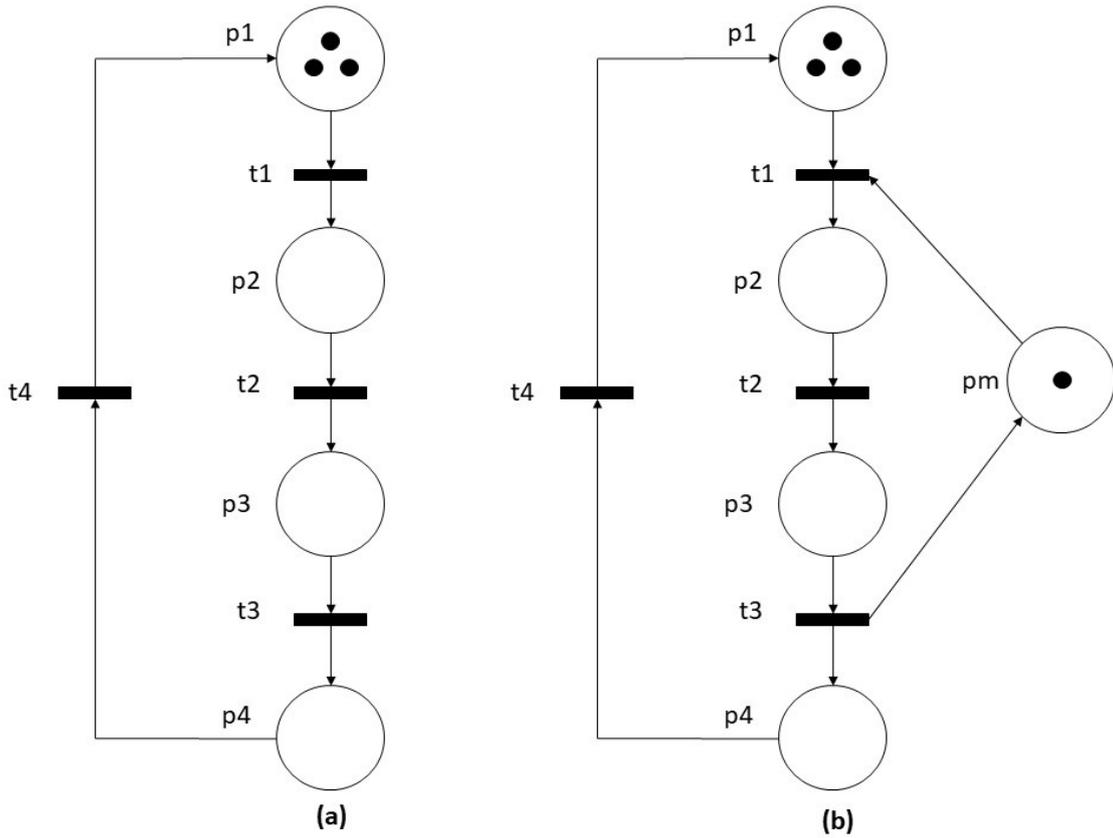


Figura 2.5: Redes de Petri do exemplo 2.8

Podemos notar que antes da adição do lugar monitor, se a transição t_1 disparasse duas vezes antes de t_3 disparar uma vez, a soma de fichas dos lugares p_2 e p_3 seria igual a 2, violando a especificação que queremos impor. A adição do supervisor fará com que esta seqüência de disparos não seja mais possível, visto que o lugar p_m ficará sem fichas e impedirá t_1 de disparar novamente enquanto t_3 não tiver disparado.

Note, contudo, que esta estratégia de controle criando lugares monitores irá apenas impor na rede uma restrição GMEC, como na equação 2.3. Esta restrição na dinâmica da rede tem como foco principal o controle de fichas, mas ela não será suficiente para resolver o problema de cooperação entre duas ou mais redes que será abordado neste trabalho.

Outra aplicação mais complexa do controle supervisorio é na coordenação de módulos representando subsistemas independentes. Neste caso, cada módulo possui um supervisor local próprio e opera de forma independente dos demais. Queremos então fazer com que estes módulos sejam sincronizados e comecem a operar de forma cooperativa como um sistema global. Podemos realizar esta sincronização através da criação de um supervisor de nível mais alto, também chamado de coordenador [23]. O coordenador será sintetizado da mesma forma que um supervisor, com a diferença de que a planta a ser controlada será composta pela junção de cada

um dos módulos, em conjunto com seus respectivos supervisores. Desta forma, as especificações locais de cada módulo serão forçadas pela ação dos supervisores locais, enquanto as especificações globais referentes a operação conjunta serão forçadas pela ação do coordenador.

Estas estratégias de controle envolvendo a síntese de um supervisor ou coordenador possuem alguns problemas. Primeiramente, o tamanho do modelo do sistema cresce significativamente com a síntese dos supervisores e coordenadores. Este problema de crescimento fica ainda mais aparente quando precisamos modificar ou incluir novas especificações em um sistema já controlado. Finalmente, a ideia de existir um coordenador centralizando todo o controle vai contra as metodologias da Indústria 4.0, que tem como foco a descentralização, independência e flexibilidade de sistemas.

Capítulo 3

Cooperação de sistemas a eventos discretos em rede

Neste capítulo abordaremos o problema de estabelecer uma cooperação entre diversos sistemas independentes conectados em rede. Faremos também uma análise do sistema em cooperação resultante para determinar a existência de bloqueios.

Na seção 3.1 apresentaremos a formulação do problema, na seção 3.2 apresentaremos a estratégia proposta para implementar a cooperação entre os sistemas, na seção 3.3 apresentaremos os dois algoritmos para implementar as estratégias propostas e na seção 3.4 apresentaremos as análises de bloqueio do sistema em cooperação.

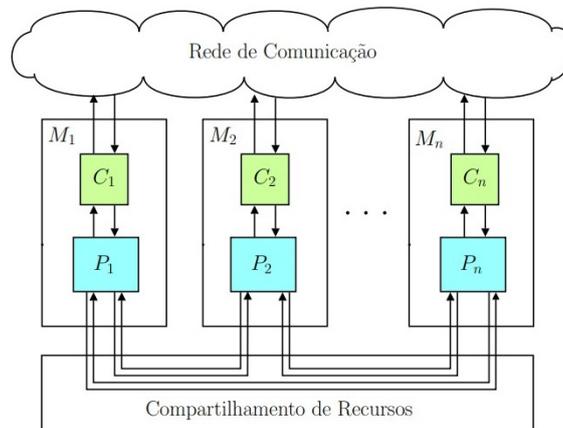


Figura 3.1: Arquitetura do sistema em cooperação em rede

3.1 Formulação do problema

3.1.1 Arquitetura considerada

A arquitetura considerada neste trabalho é mostrada na figura 3.1. O sistema global consiste em um conjunto de módulos \mathcal{M}_i (subsistemas controlados) tal que controladores C_i controlam plantas P_i , $i = 1, 2, \dots, n$, que interagem entre si compartilhando recursos. Os controladores C_i são compostos por supervisores previamente sintetizados, e se comunicam entre si por meio de uma rede de comunicação. Consideram-se as seguintes hipóteses.

H1 Não há um coordenador para o sistema.

H2 Não há atrasos nem perdas de pacotes nos canais de comunicação.

H3 Não existem bloqueios na dinâmica dos módulos controlados.

H4 As redes de Petri dos sistemas envolvidos possuem apenas arcos com peso unitário.

A hipótese H1 visa adequar o problema à realidade da Indústria 4.0, e também contornar os problemas de complexidade na síntese do coordenador. A existência de atrasos em redes de comunicação é um problema bastante recorrente e será estudado em trabalhos futuros (hipótese H2). A hipótese H3 garante que os sistemas não possuirão bloqueios em sua dinâmica. O problema de bloqueio no sistema global será abordado na seção 3.4. A hipótese H4 limita as redes envolvidas a redes com peso 1 em seus arcos.

Inicialmente, consideramos que cada um dos módulos opera sozinho, cumprindo suas respectivas especificações locais. Então nosso objetivo é sintetizar um sistema em cooperação em que os módulos cumpram tarefas específicas em conjunto, sem deixar de desempenhar suas funções locais. Uma forma de obter este sistema global é identificando os momentos nos quais uma concorrência pode acontecer e estabelecendo uma prioridade entre os módulos envolvidos. A estrutura que será usada neste trabalho é conhecida na área de sistemas digitais como estrutura mestre-escravo [24], [25].

Neste trabalho, um método para estabelecer esta prioridade será proposto, utilizando a criação de lugares comuns conectando os módulos envolvidos no compartilhamento de recurso. O lugar comum representará um recurso que será disponibilizado pelo mestre daquela estrutura, permitindo que os escravos consumam tal recurso e sigam com sua dinâmica.

Por fim, queremos verificar se a criação desse sistema em cooperação resulta no surgimento de bloqueios previamente inexistentes em sua dinâmica. Será apresentado um teste para identificar se o sistema em cooperação possui bloqueios.

3.1.2 Especificações locais

Como mencionado na seção anterior, cada módulo possui suas especificações locais previamente definidas. Um dos objetivos deste trabalho é que com a síntese do sistema em cooperação, as especificações locais continuem sendo atendidas.

Seja $N_i = (P_i, T_i, F_i, W_i)$ uma rede de Petri representando um módulo \mathcal{M}_i do sistema, com marcação inicial M_{0i} , e $\Pi(N_i, M_{0i})$ o conjunto de todas as sequências de disparo possíveis de N_i partindo de M_{0i} . Como consideramos que os módulos estão controlados, todas as sequências de $\Pi(N_i, M_{0i})$ respeitam as especificações locais do módulo \mathcal{M}_i .

Suponha que, após a síntese do sistema em cooperação, uma nova rede N'_i representando \mathcal{M}_i , com marcação inicial M'_{0i} seja obtida. Diremos que as especificações locais de N_i foram respeitadas se

$$\Pi(N_i, M_{0i}) = \Pi(N'_i, M'_{0i}). \quad (3.1)$$

Isto é, com a síntese do sistema em cooperação, queremos que todas as sequências de $\Pi(N_i, M_{0i})$ continuem possíveis e nenhuma sequência fora deste conjunto passe a ser possível, garantindo que as especificações locais continuem sendo respeitadas.

3.2 Especificações globais do sistema em cooperação

Nesta seção, apresentaremos duas estratégias para a síntese do sistema em cooperação. Cada uma das estratégias terá como objetivo forçar uma especificação envolvendo dois ou mais módulos do sistema em cooperação, sem modificar as especificações locais destes módulos.

3.2.1 Cooperação por habilitação

Sejam $N_e = (P_e, T_e, F_e, W_e)$ e $N_m = (P_m, T_m, F_m, W_m)$ redes de Petri representando dois módulos do sistema, com marcações iniciais M_{0e} e M_{0m} respectivamente. Cada um dos módulos opera de forma independente do outro e não compartilham entre si lugares ou transições. Queremos agora que um dos módulos, denominado como escravo, tenha uma de suas transições t_e desabilitada independentemente da marcação de sua rede. Esta transição será impedida de disparar até que outra transição t_m do outro módulo, denominado como mestre, dispare.

Em outras palavras, queremos fazer com que a dinâmica do módulo escolhido como escravo seja impossibilitada de disparar t_e enquanto o módulo mestre não permitir, o que por sua vez, será feito com o disparo da transição t_m . Portanto,

o escravo será forçado a esperar a permissão vinda do mestre para seguir com sua dinâmica.

Seja N_e o módulo escravo e N_m o módulo mestre, definiremos como $N_G = (P_e \cup P_m, T_e \cup T_m, F_e \cup F_m, W_g)$ a rede representando o sistema global, com W_g definido com os mesmos pesos de W_e e W_m e com marcação inicial $M_{0g} = [M_{0e}M_{0m}]$. Considere a função

$$\begin{aligned} \Upsilon : \quad T \times \Pi &\rightarrow \mathbb{N}, \\ (t, \pi) &\mapsto \Upsilon(t, \pi) = n \end{aligned} \tag{3.2}$$

que retorna o número de vezes que a transição t ocorre na sequência de disparos π . Na sequência $\pi = t_1t_2t_3t_4t_1t_2$, por exemplo, teremos $\Upsilon(t_1, \pi) = 2$, $\Upsilon(t_2, \pi) = 2$ e $\Upsilon(t_5, \pi) = 0$.

Definição 3.1 (*Especificação da cooperação por habilitação*) *Seja $N_g = (P_e \cup P_m, T_e \cup T_m, F_e \cup F_m, W_g)$ um sistema em cooperação com marcação inicial M_{0g} contendo um módulo mestre N_m e um módulo escravo N_e . Sejam $t_e \in T_e$ uma transição escolhida do módulo escravo e $t_m \in T_m$ uma transição escolhida do módulo mestre, uma especificação por habilitação do sistema em cooperação N_g será definida como:*

$$\forall \pi \in \Pi(N_g, M_{0g}), \Upsilon(t_e, \pi) \leq \Upsilon(t_m, \pi) \tag{3.3}$$

em que π representa uma sequência de disparo qualquer de N_g partindo de M_{0g} e $\Pi(N_g, M_{0g})$ o conjunto de todos os π possíveis para aquela marcação.

Em palavras, a partir da definição 3.1 a especificação permitirá no sistema apenas sequências em que o número de ocorrências da transição t_e seja menor ou igual ao número de ocorrências da transição t_m .

Sendo assim, apenas sequências em que t_e dispare depois de t_m serão permitidas, impondo no sistema o comportamento desejado.

Note que como $\Pi(N_g, M_{0g})$ é prefixo fechado, a especificação da definição 3.1 deve valer para todos os prefixos de π também, visto que ela deve valer para todas as sequências de $\Pi(N_g, M_{0g})$. Sendo assim, é possível que uma sequência π não viole a especificação, mas algum prefixo dela viole, fazendo com que π não seja uma sequência possível.

Finalmente, vale ressaltar que a especificação da definição 3.1 não restringe a ordem em que os disparos podem ocorrer. Cada disparo de t_m representará uma permissão para o módulo escravo. Se t_m disparar múltiplas vezes antes de t_e disparar uma vez, múltiplas permissões serão dadas, e t_e poderá disparar o mesmo número de vezes.

Exemplo 3.1 *Sejam $N_e = (P_e, T_e, F_e, W_e)$ e $N_m = (P_m, T_m, F_m, W_m)$ duas redes de Petri escolhidas como escravo e mestre, respectivamente, e $N_G = (P_e \cup P_m, T_e \cup T_m, F_e \cup F_m, W_g)$ a rede representando o sistema global, com marcação inicial M_{0g} . Considere $t_e \in T_e$ e $t_m \in T_m$ as transições escolhidas dos módulos escravo e mestre, respectivamente, para estabelecer a cooperação.*

Primeiramente, considere a sequência de disparo $\pi = t_e t_m t_m t_e$. Note que $\Upsilon(t_e, \pi) = \Upsilon(t_m, \pi) = 2$, respeitando a especificação da definição 3.1. Contudo, o prefixo $\pi_1 = t_e$ terá $\Upsilon(t_e, \pi) = 1$ e $\Upsilon(t_m, \pi) = 0$, o que viola a especificação. Sendo assim, a sequência π não poderá fazer parte de $\Pi(N_g, M_{0g})$, visto que isso implicaria que π_1 também faz parte, violando a especificação.

Considere agora a sequência $\pi = t_m t_m t_e t_e$. Tanto π quanto qualquer prefixo de π respeitam a especificação da definição 3.1, o que implica que π poderá fazer parte de $\Pi(N_g, M_{0g})$.

3.2.2 Cooperação por impedimento

Mais uma vez, considere $N_e = (P_e, T_e, F_e, W_e)$ e $N_m = (P_m, T_m, F_m, W_m)$ redes de Petri representando dois módulos do sistema, com marcações iniciais M_{0e} e M_{0m} respectivamente. De forma semelhante à cooperação por habilitação, escolheremos um módulo como escravo (N_e) e outro como mestre (N_m). Desta vez, contudo, escolheremos uma transição $t_e \in T_e$ do módulo escravo e duas transições $t_{m1} \in T_m$ e $t_{m2} \in T_m$ do módulo mestre. Queremos agora que a dinâmica do módulo escravo se inicie inalterada. Apenas após o disparo de t_{m1} que começaremos a impossibilitar a transição t_e de disparar. Este impedimento seguirá até o disparo de t_{m2} , após o qual t_e poderá disparar novamente.

Em outras palavras, queremos definir dentro da dinâmica do módulo mestre um período de impedimento, cujo início é dado pela transição t_{m1} e o final pela transição t_{m2} . Enquanto t_{m1} não disparar nenhum dos dois módulos perceberá qualquer alteração em sua dinâmica. Com o disparo de t_{m1} , o módulo mestre entrará no período de impedimento, impedindo o módulo escravo de disparar a transição t_e . Apenas com o disparo de t_{m2} que o mestre sairá do período de impedimento, permitindo que o módulo escravo siga com sua dinâmica.

Definiremos novamente como $N_G = (P_e \cup P_m, T_e \cup T_m, F_e \cup F_m, W_g)$ a rede representando o sistema global, com W_g definido com os mesmos pesos de W_e e W_m e com marcação inicial $M_{0g} = [M_{0e} M_{0m}]$. Considere a função

$$\begin{aligned} \text{Suffix} : \quad \Pi &\rightarrow T, \\ \pi &\mapsto \text{Suffix}(\pi) = t \end{aligned} \tag{3.4}$$

que retorna a última transição que ocorreu na sequência π .

Definição 3.2 (*Especificação da cooperação por impedimento*) Seja $N_g = (P_e \cup P_m, T_e \cup T_m, F_e \cup F_m, W_g)$ um sistema em cooperação com marcação inicial M_{0g} contendo um módulo mestre N_m e um módulo escravo N_e . Sejam $t_e \in T_e$ uma transição escolhida do módulo escravo e $t_{m1} \in T_m$ e $t_{m2} \in T_m$ transições escolhidas do módulo mestre, uma especificação por impedimento do sistema em cooperação N_g será definida como:

$$\forall \pi \in \Pi(N_g, M_{0g}), \text{Sufix}(\pi) \neq t_e \vee \Upsilon(t_{m1}, \pi) = \Upsilon(t_{m2}, \pi) \quad (3.5)$$

e

$$\forall \pi \in \Pi(N_g, M_{0g}), \Upsilon(t_{m2}, \pi) \leq \Upsilon(t_{m1}, \pi). \quad (3.6)$$

em que π representa uma sequência de disparo qualquer de N_g partindo de M_{0g} e $\Pi(N_g, M_{0g})$ o conjunto de todos os π possíveis para aquela marcação.

Em palavras, a primeira equação da definição 3.5 nos garante que todas as sequências terminando em t_e devem conter também o mesmo número de ocorrências de t_{m1} e t_{m2} .

Se interpretarmos o disparo de t_{m1} como uma ordem de impedimento vinda do mestre e o disparo de t_{m2} como uma liberação, t_{m1} disparar mais vezes que t_{m2} implica em impedimentos sem liberação. Desta forma, t_e será impossibilitada de disparar enquanto todos os impedimentos não forem liberados.

Note que, mais uma vez, a especificação deve valer para todos os prefixos de π , de forma que se π não termina com t_e mas algum de seus prefixos termina, este prefixo deve respeitar a especificação para que π seja uma sequência válida.

Finalmente, para que esta especificação imponha o comportamento desejado, precisamos também garantir que t_{m2} não possa disparar antes de t_{m1} , excluindo os casos onde uma liberação foi dada antes do impedimento ocorrer.

A segunda equação da definição 3.5 garante este comportamento. Note que esta segunda equação é semelhante à especificação da definição 3.1, com t_{m2} no lugar de t_e .

Exemplo 3.2 Sejam $N_e = (P_e, T_e, F_e, W_e)$ e $N_m = (P_m, T_m, F_m, W_m)$ duas redes de Petri escolhidas como escravo e mestre, respectivamente, e $N_G = (P_e \cup P_m, T_e \cup T_m, F_e \cup F_m, W_g)$ a rede representando o sistema global, com marcação inicial M_{0g} . Considere $t_e \in T_e$ e $t_{m1}, t_{m2} \in T_m$ as transições escolhidas dos módulos escravo e mestre, respectivamente, para estabelecer a cooperação.

Considere a sequência de disparo $\pi = t_e t_{m1} t_{m2} t_e$. Os prefixos de π terminando em t_e são $\pi_1 = t_e$ e $\pi_2 = \pi = t_e t_{m1} t_{m2} t_e$. Como $\Upsilon(t_{m1}, \pi_1) = \Upsilon(t_{m2}, \pi_1) = 0$ e

$\Upsilon(t_{m1}, \pi_2) = \Upsilon(t_{m2}, \pi_2) = 1$, a primeira equação da definição 3.5 foi respeitada. Além disso, $\Upsilon(t_{m2}, \pi) \leq \Upsilon(t_{m1}, \pi)$ para todos os prefixos de π , respeitando também a segunda equação da definição 3.5.

Considere agora a sequência $\pi' = t_e t_{m1} t_e t_{m2}$. O prefixo $\pi_3 = t_e t_{m1} t_e$ termina em t_e e possui $\Upsilon(t_{m1}, \pi_3) \neq \Upsilon(t_{m2}, \pi_3)$. Como um prefixo de π' viola a primeira especificação da definição 3.5, então π' não pode ser uma sequência possível.

Finalmente, considere a sequência $\pi'' = t_e t_{m2} t_{m1} t_e$. Esta sequência possui dois prefixos terminando em t_e , e para ambos a primeira especificação da definição 3.5 é respeitada. Por outro lado, a segunda especificação da definição 3.5 não é respeitada, visto que para o prefixo $\pi_4 = t_e t_{m2}$ temos $\Upsilon(t_{m2}, \pi_4) > \Upsilon(t_{m1}, \pi_4)$. Sendo assim, π'' não poderá ser uma sequência válida.

3.3 Síntese do sistema em cooperação com lugares comuns

Nesta seção apresentaremos as duas soluções propostas para a implementação das coordenações por habilitação e impedimento, forçando no sistema as especificações descritas na seção 3.2. Estas soluções irão envolver a criação de lugares comuns (ver seção 2.2.2) entre os módulos envolvidos na coordenação. Também serão apresentados os algoritmos para a implementação das soluções.

Antes de apresentar as soluções propostas, contudo, precisamos apresentar o conceito de grupo de cooperação. Dado um conjunto de k redes de Petri $\mathcal{N} = \{N_1, \dots, N_k\}$, podemos escolher um par qualquer de redes para estabelecer entre elas uma cooperação por habilitação ou por impedimento. É necessário escolhermos qual módulo será o mestre e qual será o escravo, além de escolher também as transições que serão envolvidas. Matematicamente, para definir uma cooperação por habilitação precisamos escolher as tuplas (N_m, t_m) e (N_e, t_e) , enquanto para definir uma cooperação por impedimento precisamos escolher a trinca (N_m, t_{m1}, t_{m2}) e a tupla (N_e, t_e) , como definido na seção 3.2.

Podemos então repetir este processo um número qualquer de vezes, criando novas cooperações entre pares de módulos que independem das demais cooperações já criadas. Definiremos cada uma destas cooperações como um grupo de cooperação, e agruparemos no mesmo grupo todas as cooperações que compartilhem um módulo mestre N_m usando também as mesmas transições de N_m , ou seja, que tenham a mesma tupla (N_m, t_m) , para a cooperação por habilitação, ou a mesma trinca (N_m, t_{m1}, t_{m2}) para a cooperação por impedimento.

Exemplo 3.3 *Vamos tomar como exemplo as redes de Petri mostradas na figura 3.2, em que $\mathcal{N} = \{N_1, N_2, N_3\}$. Começaremos criando uma cooperação por habi-*

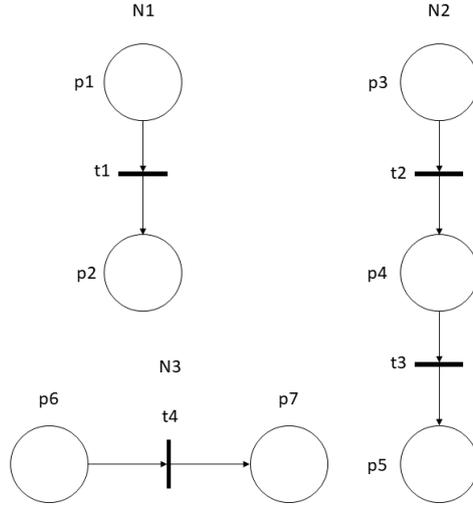


Figura 3.2: Redes de Petri do exemplo 3.3

litação entre as redes N_1 e N_2 , com N_1 de mestre e envolvendo as transições t_1 e t_2 . Este será nosso primeiro grupo de cooperação, que tem como mestre N_1 com a transição t_1 . A seguir, criaremos outra cooperação por habilitação, desta vez entre N_1 e N_3 , com N_1 de mestre e envolvendo as transições t_1 e t_4 . Como esta cooperação possui a mesma tupla (N_1, t_1) para o módulo mestre, ela será também incluída no primeiro grupo de cooperação, que agora será composto por (N_1, t_1) para o mestre e (N_2, t_2) e (N_3, t_4) para os escravos. Cada nova cooperação que compartilhar a mesma tupla (N_1, t_1) como mestre pertencerá ao mesmo grupo. Finalmente, se criarmos uma cooperação entre N_2 e N_3 com N_2 de mestre e envolvendo as transições t_2 e t_4 , precisaremos adicionar esta cooperação a um novo grupo, visto que não temos nenhum grupo já criado que tenha a tupla (N_2, t_2) para o mestre.

Matematicamente, para a cooperação por habilitação um grupo de cooperação G_i será definido como $G_i = (\mathcal{G}_i^{mest}, \mathcal{G}_i^{esc})$, em que a tupla $\mathcal{G}_i^{mest} = (N_m, t_m)$ representa o subgrupo do módulo mestre e o conjunto de tuplas $\mathcal{G}_i^{esc} = \{(N_{i_1}, t_{j_1}), \dots, (N_{i_r}, t_{j_r})\}$ representa o subgrupo dos p módulos escravos que compartilham o mesmo subgrupo \mathcal{G}_i^{mest} , em que $\{i_1, \dots, i_r\} \subset \mathcal{I}_r$, $\mathcal{I}_r = \{1, \dots, k\}$, $N_{i_p} \in \mathcal{N}$, $N_{i_p} \neq N_m$, $j_p \in \{1, \dots, |T_{i_p}|\}$ e $t_{j_p} \in T_{i_p}$.

No sistema mostrado no exemplo 3.3 teremos para o primeiro grupo $G_1 = (\mathcal{G}_1^{mest}, \mathcal{G}_1^{esc})$, em que $\mathcal{G}_1^{mest} = (N_1, t_1)$ e $\mathcal{G}_1^{esc} = \{(N_2, t_2), (N_3, t_4)\}$. Já para o segundo grupo, teremos $G_2 = (\mathcal{G}_2^{mest}, \mathcal{G}_2^{esc})$, em que $\mathcal{G}_2^{mest} = (N_2, t_2)$ e $\mathcal{G}_2^{esc} = \{(N_3, t_4)\}$.

Para a coordenação por impedimento, o grupo de cooperação G_i será também definido como $G_i = (\mathcal{G}_i^{mest}, \mathcal{G}_i^{esc})$. Desta vez, contudo, o subgrupo do módulo mestre será definido como a trinca $\mathcal{G}_i^{mest} = (N_m, t_{m1}, t_{m2})$, enquanto o subconjunto dos módulos escravos é definido da mesma forma que na coordenação por habilitação,

agrupando todos os escravos que compartilhem o mesmo subgrupo \mathcal{G}_i^{mest} .

Com isso, podemos apresentar os algoritmos propostos para a implementação das cooperações definidas na seção 3.2.

3.3.1 Síntese do sistema em cooperação por habilitação utilizando lugares comuns

A seguir será apresentado o algoritmo para a síntese do sistema em cooperação por habilitação usando lugares comuns. O algoritmo terá como objetivo forçar no sistema a especificação (3.1).

Algoritmo 3.1: Sistema em cooperação por habilitação

input :

- k : número de módulos envolvidos;
- g : número de grupos de cooperação;
- $\mathcal{N} = \{N_1, \dots, N_k\}$: um conjunto de redes de Petri tal que $N_j = (P_j, T_j, F_j, W_j)$, para $j \in \{1, \dots, k\}$;
- Grupos: $G_i = (\mathcal{G}_i^{mest}, \mathcal{G}_i^{esc})$, $i = 1, 2, \dots, g$, tal que
 - $\mathcal{G}_i^{mest} = (N_m, t')$: $t' \in T_m$, $m \in \{1, 2, \dots, k\}$;
 - $\mathcal{G}_i^{esc} = \{(N_{i_1}, t_{j_1}), \dots, (N_{i_p}, t_{j_p})\}$ um conjunto com r elementos em que $\{i_1, \dots, i_r\} \subset \mathcal{I}_r$, $\mathcal{I}_r = \{1, \dots, k\}$, $N_{i_p} \in \mathcal{N}$, $N_{i_p} \neq N_m$, $t_{j_p} \in T_{i_p}$ e $j_p \in \{1, \dots, |T_{i_p}|\}$.

output: $\mathcal{N}^* = \{N_1^*, \dots, N_k^*\}$: conjunto \mathcal{N} modificado tal que

$$N_j^* = (P_j^*, T_j, F_j^*, W_j^*), j \in \{1, \dots, k\}.$$

1 $\mathcal{N}^* \leftarrow \mathcal{N}$;

```

2   $i = 1, 2, \dots, g$ 
3   $L \leftarrow \emptyset;$ 
4   $Count \leftarrow 0;$ 
5  for  $w = 1, \dots, p$  do
6       $Count \leftarrow Count + 1;$ 
7      Crie o lugar comum  $p_{i,Count}$ ;
8       $L \leftarrow L \cup \{p_{i,Count}\};$ 
9       $P_{i,w}^* \leftarrow P_{i,w}^* \cup \{p_{i,Count}\};$ 
10      $F_{i,w}^* \leftarrow F_{i,w}^* \cup \{(p_{i,Count}, t_{i,w})\};$ 
11     Defina  $W_{i,w}^* : F_{i,w}^* \rightarrow \mathbb{N}$ , tal que:
        

- $W_{i,w}^*(p_{i,Count}, t_{i,w}) = 1,$
- $W_{i,w}^*(p, t) = W_{i,w}^*(p, t), \forall (p, t) \in F_{i,w}^*$
- $W_{i,w}^*(t, p) = W_{i,w}^*(t, p), \forall (t, p) \in F_{i,w}^*$

 $N_{i,w}^* \leftarrow (P_{i,w}^*, T_{i,w}, F_{i,w}^*, W_{i,w}^*);$ 
12     end
13      $P_m^* \leftarrow P_m^* \cup L$ 
14 end
15
16     foreach  $p_l \in L$  do
17          $F_m^* \leftarrow F_m^* \cup \{(t', p_l)\};$ 
18     end
19     Defina  $W_m^* : F_m^* \rightarrow \mathbb{N}$ , tal que:
        

- $W_m^*(t', p_l) = 1, \forall p_l \in L$
- $W_m^*(t, p) = W_m^*(t, p), \forall (t, p) \in F_m^*$
- $W_m^*(p, t) = W_m^*(p, t), \forall (p, t) \in F_m^*$

 $N_m^* \leftarrow (P_m^*, T_m, F_m^*, W_m^*);$ 
20 end

```

O algoritmo tem como entrada as k redes de Petri dos módulos do sistema e os g grupos de cooperação escolhidos, com seus respectivos subgrupos do modulo mestre e módulos escravos. Na linha 1 atribuímos \mathcal{N} ao conjunto de redes de Petri modificadas \mathcal{N}^* . Na linha 2 executaremos o algoritmo g vezes, uma para cada grupo. Nas linhas 3 e 4 criamos um conjunto L e uma variável $Count$, em que L será usado para salvar a informação de quantos lugares comuns foram criados para cada grupo, enquanto $Count$ será usada para nomear os lugares comuns. Entre as

linhas 6 e 9, criaremos um lugar comum para cada módulo escravo e modificaremos a rede de Petri daquele módulo para incluir o lugar comum, adicionando-o também ao conjunto L . Nas linhas 10 e 11 adicionamos também à rede de cada escravo um arco indo do lugar comum até cada uma das transições daquele módulo envolvidas na cooperação, redefinindo também a função que descreve o peso destes arcos. Na linha 13 adicionamos todos os lugares comuns criados para aquele grupo ao módulo mestre, através do conjunto L . Entre as linhas 15 e 19, criamos também um arco para cada lugar comum conectando a transição do módulo mestre ao lugar comum e atribuímos a este arco peso 1. Ao final do algoritmo, teremos o conjunto \mathcal{N}^* que contém cada rede de Petri modificada para incluir os lugares comuns e seus arcos.

Exemplo 3.4 *Tomando novamente o exemplo 3.3 e o sistema da figura 3.2, aplicaremos o algoritmo 3.1 com as seguintes entradas: $k = 3, g = 2, \mathcal{N} = \{N_1, N_2, N_3\}$, $G_1 = (\mathcal{G}_1^{mest}, \mathcal{G}_1^{esc})$ e $G_2 = (\mathcal{G}_2^{mest}, \mathcal{G}_2^{esc})$, como definidos no exemplo 3.3. Teremos $N_1 = (P_1, T_1, F_1, W_1)$, $N_2 = (P_2, T_2, F_2, W_2)$ e $N_3 = (P_3, T_3, F_3, W_3)$*

As linhas 2 a 19 são executadas duas vezes, uma para cada grupo. Começando pelo grupo 1, após criar o conjunto vazio L e a variável $Count$, executamos as linhas 5 a 12 duas vezes, uma para cada escravo do grupo. Para o escravo (N_2, t_2) criamos o lugar comum $p_{1,1}$ e adicionamos ele à P_2 , além de criar também um arco indo de $p_{1,1}$ para t_2 com peso 1, que por sua vez será adicionado a F_2 . O mesmo procedimento é repetido para o escravo (N_3, t_4) , criando o lugar comum $p_{1,2}$ e modificando P_3, F_3 e W_3 . Neste ponto do algoritmo, o conjunto L será igual a $\{p_{1,1}, p_{1,2}\}$. Como não existem mais escravos para o grupo 1, seguimos para a linha 13. Adicionaremos todos os lugares criados para este grupo no conjunto de lugares do módulo mestre P_1 e criamos também um arco indo de t_1 para cada lugar comum criado. Ao final, temos cada uma das redes modificadas para incluírem os lugares comuns criados para aquele grupo, assim como seus arcos. Repetimos todo este procedimento para os módulos do segundo grupo, modificando mais uma vez todas as redes envolvidas naquele grupo. Para o segundo grupo, como temos apenas um escravo, criamos apenas o lugar comum $p_{2,1}$ que é adicionado nas redes N_2 e N_3 , junto com seus respectivos arcos.

Os lugares comuns criados assim como seus arcos podem ser vistos na figura 3.3. Os lugares pintados de vermelho pertencem ao grupo 1 enquanto os pintados de verde pertencem ao grupo 2.

Ao final da execução, teremos $\mathcal{N}^ = \{N_1^*, N_2^*, N_3^*\}$ em que $N_1^* = (P_1^*, T_1, F_1^*, W_1^*)$, $N_2^* = (P_2^*, T_2, F_2^*, W_2^*)$ e $N_3^* = (P_3^*, T_3, F_3^*, W_3^*)$. Para as redes modificadas, teremos pra o módulo 1 $P_1^* = \{p_1, p_2, p_{1,1}, p_{1,2}\}$, $T_1 = \{t_1\}$, $F_1 = \{(p_1, t_1), (t_1, p_2), (t_1, p_{1,1}), (t_1, p_{1,2})\}$. Para o módulo 2 teremos $P_2^* = \{p_3, p_4, p_5, p_{1,1}, p_{2,1}\}$, $T_2 = \{t_2, t_3\}$, $F_2 =$*

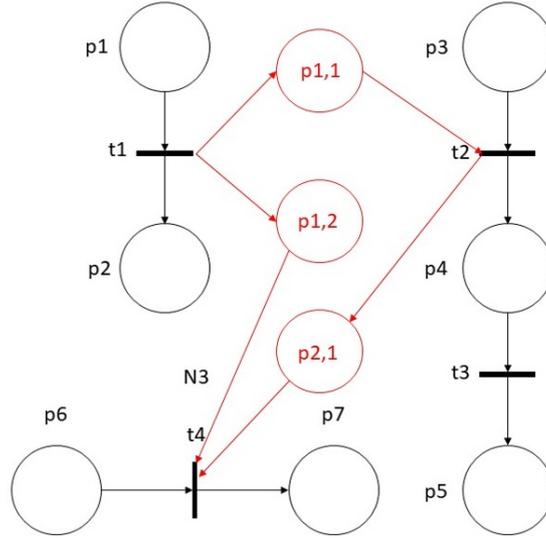


Figura 3.3: Redes de Petri do exemplo 3.4

$\{(p_3, t_2), (t_2, p_4), (p_4, t_3), (t_3, p_5), (p_{1,1}, t_2), (t_2, p_{2,1})\}$. Finalmente, para o módulo 3 teremos $P_3^* = \{p_6, p_7, p_{1,2}, p_{2,1}\}$, $T_3 = \{t_4\}$ e $F_3 = \{(p_6, t_4), (t_4, p_7), (p_{1,2}, t_4), (p_{2,1}, t_4)\}$.

Proposição 3.1 *Sejam $N_e = (P_e, T_e, F_e, W_e)$ e $N_m = (P_m, T_m, F_m, W_m)$ duas redes de Petri com marcações iniciais M_{0e} e M_{0m} , tal que $t_e \in T_e$ e $t_m \in T_m$ sejam um par de transições escolhidas. Se $\mathcal{N}^* = \{N_e^*, N_m^*\}$ for obtido executando o algoritmo 3.1 com entradas $\mathcal{N} = \{N_e, N_m\}$ e $G_1 = ((N_m, t_m), (N_e, t_e))$, então o sistema $N_G = (P_e^* \cup P_m^*, T_e \cup T_m, F_e^* \cup F_m^*, W_g^*)$, com W_g^* definido com os mesmos pesos de W_e^* e W_m^* e marcação inicial $M_{0G} = [M_{0m} M_{0e} 0]$, respeitará a especificação da definição 3.1 para quaisquer M_{0m} e M_{0e} .*

Prova. O lugar comum p_c , obtido através do algoritmo 3.1, sempre possuirá como transição de entrada t_m e como transição de saída t_e . A única forma de adicionar uma ficha à p_c é com o disparo de t_m e a única forma de retirar uma ficha é com o disparo de t_e . Além disso, t_e só poderá disparar quando houver ao menos uma ficha em p_c , visto que na linha 10 do algoritmo 3.1 um arco ligando p_c a t_e é criado. Finalmente, por construção, p_c não possui fichas em sua marcação inicial, o que é representado no último elemento da marcação inicial $M_{0G} = [M_{0m} M_{0e} 0]$ do sistema global. Sendo assim, o número de fichas em p_c em qualquer momento será igual ao número de fichas adicionadas pelo disparo de t_m menos o número de fichas retiradas pelo disparo de t_e , ou seja, será igual a $\Upsilon(t_m, \pi) - \Upsilon(t_e, \pi)$, em que π é uma sequência qualquer de N_G para uma marcação inicial qualquer.

Vamos supor que, para alguma marcação inicial M_{0G} , exista uma sequência $\pi' \in \Pi\{N_g, M_{0G}\}$ que viole a especificação (3.1), ou seja, $\Upsilon(t_e, \pi') > \Upsilon(t_m, \pi')$. Após a ocorrência desta sequência, teremos que o número de fichas em p_c será

igual a $\Upsilon(t_m, \pi') - \Upsilon(t_e, \pi')$, o que resulta em uma quantidade negativa de fichas. Isto significa que em algum momento p_c não possuía fichas e t_e disparou, o que é impossível. Sendo assim, a sequência π' não pode existir em N_G . ■

Proposição 3.2 *Sejam $N_e = (P_e, T_e, F_e, W_e)$ e $N_m = (P_m, T_m, F_m, W_m)$ duas redes de Petri com marcações iniciais M_{0e} e M_{0m} , em que $t_e \in T_e$ e $t_m \in T_m$ sejam um par de transições escolhidas. Seja também $N_G = (P_e \cup P_m, T_e \cup T_m, F_e \cup F_m, W_g)$ o sistema global, com W_g definido com os mesmos pesos de W_e e W_m e marcação inicial $M_{0G} = [M_{0m} M_{0e}]$. Se $N^* = \{N_e^*, N_m^*\}$ for obtido executando o algoritmo 3.1 com entradas $\mathcal{N} = \{N_e, N_m\}$ e $G_1 = ((N_m, t_m), (N_e, t_e))$, então o sistema global $N_G^* = (P_e^* \cup P_m^*, T_e \cup T_m, F_e^* \cup F_m^*, W_g^*)$, com W_g^* definido com os mesmos pesos de W_e^* e W_m^* e marcação inicial $M_{0G}^* = [M_{0m} M_{0e} 0]$, terá a seguinte propriedade: $\Pi(N_g^*, M_{0g}^*) \subseteq \Pi(N_g, M_{0g})$.*

Prova. Queremos provar que $\Pi(N_g^*, M_{0g}^*) \subseteq \Pi(N_g, M_{0g})$, ou seja, não existe nenhuma sequência de disparo π que seja possível em N_g^* e não seja possível em N_g . Isto é, com a aplicação do algoritmo 3.1, que insere um lugar comum nas redes de Petri, não são adicionados novos comportamentos no sistema global, sendo este apenas restringido. Sejam N_m e N_e as redes de Petri dos módulos mestre e escravo antes da aplicação do algoritmo 3.1 e N_m^* e N_e^* as redes obtidas ao final do algoritmo. Para que exista uma sequência π^* em N_g^* que não exista em N_g , é preciso que: (i) exista uma sequência π_m^* em N_m^* que não existe em N_m ou; (ii) exista uma sequência π_e^* em N_e^* que não existe em N_e . Isto sempre será verdade visto que não existem arcos ligando elementos de N_e a elementos de N_m , e nenhum arco com esta característica é criado com a execução do algoritmo 3.1. Sendo assim, os disparos de N_e dependem apenas da marcação de N_e , enquanto os disparos de N_m dependem apenas da marcação de N_m . O mesmo pode ser dito para N_e^* e N_m^* .

- (i) Primeiramente vamos assumir que exista uma sequência π_m^* que seja possível em N_m^* , mas não seja possível em N_m . Podemos escrever π_m^* como $\pi_m^* = \sigma_1 t' \sigma_2$, em que $\sigma_1 \in \Pi(N_m, M_{0m})$, $\sigma_1 \in \Pi(N_m^*, M_{0m}^*)$, $\sigma_1 t' \notin \Pi(N_m, M_{0m})$ e $\sigma_1 t' \in \Pi(N_m^*, M_{0m}^*)$, ou seja, σ_1 é uma sequência possível tanto em N_m quanto em N_m^* , e $\sigma_1 t'$ é uma sequência possível apenas em N_m^* .

Seja M' a marcação alcançada por N_m depois da sequência σ_1 , ou seja $M_{0m}[\sigma_1]M'$ e M'' a marcação alcançada pelos lugares de N_m^* que existem em N_m (todos exceto por p_l) após a sequência σ_1 , ou seja, $M_{0m}[\sigma_1]M''$. Para que $\sigma_1 t'$ seja possível em N_m^* e não seja possível em N_m é necessário que: (a) $M' \neq M''$, ou seja, a rede alterada alcançou uma marcação diferente com a mesma sequência σ_1 ou; (b) $M' = M''$ e t' está habilitada para M' em N_g^* mas

não em N_g . Para que qualquer um dos dois casos seja possível, é necessário que um ou mais arcos da rede sejam alterados ou novos arcos sejam criados envolvendo os lugares e transições já existentes na rede, de forma que uma nova marcação possa ser alcançada ou uma transição passe a estar habilitada. Como por construção, as únicas diferenças entre N_m e N_m^* são o lugar comum p_c que foi criado pelo algoritmo 3.1 e um arco ligando uma transição de N_m a p_c , nenhuma das duas opções seria possível, visto que o único arco criado não altera a marcação de nenhum lugar além de p_c . Portanto, concluímos que $\pi_m^* \notin \Pi(N_m^*, M_{0m}^*)$.

- (ii) De forma semelhante, podemos provar que não é possível existir uma sequência de disparo π_e^* em N_e^* que não existe em N_e . Por construção, as únicas diferenças entre N_e e N_e^* são o lugar comum p_c e um arco ligando uma transição de p_c a N_e . Este arco pode retirar fichas apenas de p_c , o que não afetará as marcações dos demais lugares. Portanto, concluímos que $\pi_e^* \notin \Pi(N_e^*, M_{0e}^*)$.

Finalmente, como nem π_m^* e nem π_e^* poderão existir, π^* não pode existir no sistema global N_g^* sem ser possível também em N_g . ■

Teorema 3.1 *Seja $\mathcal{N} = \{N_1, \dots, N_k\}$ um conjunto com k redes de Petri e sejam $G_i = (\mathcal{G}_i^{mest}, \mathcal{G}_i^{esc})$, $i = 1, \dots, g$, os grupos de cooperação envolvendo os módulos de \mathcal{N} . Se $\mathcal{N}^* = \{N_1^*, \dots, N_k^*\}$ for obtido aplicando o algoritmo 3.1 com entradas k, g, \mathcal{N} e G_i , $i = 1, \dots, g$, então o sistema global $N_G = (\bigcup_{i=1}^k P_i^*, \bigcup_{i=1}^k T_i, \bigcup_{i=1}^k F_i^*, W_g^*)$ respeitará a especificação (3.1) para todos os pares escravo-mestre nos grupos de cooperação.*

Prova. Seja $\mathcal{N} = \{N_1, \dots, N_k\}$ um sistema com k redes de Petri. Vamos inicialmente considerar que temos apenas um grupo de cooperação $G_1 = (\mathcal{G}_1^{mest}, \mathcal{G}_1^{esc})$. A proposição 3.1 nos garante que o sistema global $N_G = (\bigcup_{i=1}^k P_i^*, \bigcup_{i=1}^k T_i, \bigcup_{i=1}^k F_i^*, W_g^*)$, obtido com a aplicação do algoritmo 3.1 e com marcação inicial M_{0G} , respeita a especificação do grupo G_1 .

Suponha agora que aplicamos o algoritmo 3.1 um número x de vezes, escolhendo um novo grupo G_i cada vez, obtendo o sistema global N_G^x e o conjunto \mathcal{N}^x , de forma que N_G^x respeite todas as especificações dos grupos G_i , $i = 1, \dots, x$. Aplicando agora mais uma vez o algoritmo 3.1 para um novo grupo G_{x+1} , obtemos \mathcal{N}^{x+1} . Queremos provar que $N_G^{x+1} = (\bigcup_{i=1}^k P_i^{x+1}, \bigcup_{i=1}^k T_i, \bigcup_{i=1}^k F_i^{x+1}, W_g^{x+1})$ respeitará as especificações dos grupos G_i , $i = 1, \dots, x+1$.

Primeiramente, o sistema \mathcal{N}^{x+1} possuirá um novo lugar comum p_c^{x+1} . Por construção, este lugar sempre possuirá apenas t_m^x como transição de entrada e t_e^x como transição de saída, independente de \mathcal{N}^x . Sendo assim, toda a prova da proposição 3.1 pode ser repetida independente da entrada escolhida para o algoritmo, de forma que

$\Upsilon(t_m^x, \pi) > \Upsilon(t_e^x, \pi)$ valerá para toda e qualquer sequência de \mathcal{N}^{x+1} , respeitando a especificação de G_{x+1}

Note que o sistema \mathcal{N} não possui arcos interligando lugares e transições de redes diferentes. Esta propriedade é mantida em todos os módulos de \mathcal{N}^x , após x aplicações do algoritmo 3.1, visto que este algoritmo não modifica arcos ou transições já existentes na rede, e os únicos arcos criados ligam transições a lugares comuns. Sendo assim, teremos que $\Pi(N_G^{x+1}, M_{0G}^{x+1}) \subseteq \Pi(N_G^x, M_{0G}^x)$ pela proposição 3.2, em que $M_{0G}^{x+1} = [M_{0G}^x 0]$. Como todas as sequências de $\Pi(N_G^x, M_{0G}^x)$ respeitam as especificações de N_G^x , então todas as sequências de $\Pi(N_G^{x+1}, M_{0G}^{x+1})$ respeitarão também estas especificações, além de respeitarem também a nova especificação imposta.

Com isso, provamos que podemos repetir este procedimento um número qualquer de vezes para um par qualquer de mestre-escravo, de forma que cada especificação imposta se mantém respeitada e toda especificação nova é incluída no sistema. ■

3.3.2 Síntese do sistema em cooperação por impedimento utilizando lugares comuns

A seguir será apresentado o algoritmo para a síntese do sistema em cooperação por impedimento usando lugares comuns. O algoritmo terá como objetivo impor no sistema as especificações (3.5) e (3.6). Diferente do algoritmo 3.1, este utilizará a estrutura de redes de Petri estendidas (definição 2.7) em que se considera o arco inibidor.

Algoritmo 3.2: Sistema em cooperação por impedimento

input :

- k : número de módulos envolvidos;
- g : número de grupos de cooperação;
- $\mathcal{N} = \{N_1, \dots, N_k\}$: um conjunto de redes de Petri tal que $N_j = (P_j, T_j, F_j, W_j, I_j)$, para $j \in \{1, \dots, k\}$;
- Grupos: $G_i = (\mathcal{G}_i^{mest}, \mathcal{G}_i^{esc})$, $i = 1, 2, \dots, g$, tal que
 - $\mathcal{G}_i^{mest} = (N_m, t_{m1}, t_{m2})$: $t_{m1}, t_{m2} \in T_m$, $m \in \{1, 2, \dots, k\}$;
 - $\mathcal{G}_i^{esc} = \{(N_{i_1}, t_{j_1}), \dots, (N_{i_p}, t_{j_p})\}$ um conjunto com r elementos em que $\{i_1, \dots, i_r\} \subset \mathcal{I}_r$, $\mathcal{I}_r = \{1, \dots, k\}$, $N_{i_p} \in \mathcal{N}$, $N_{i_p} \neq N_m$, $t_{j_p} \in T_{i_p}$ e $j_p \in \{1, \dots, |T_{i_p}|\}$.

output: $\mathcal{N}^* = \{N_1^*, \dots, N_k^*\}$: conjunto \mathcal{N} modificado tal que

$$N_j^* = (P_j^*, T_j, F_j^*, W_j^*, I_j^*), j \in \{1, \dots, k\}.$$

$\mathcal{N}^* \leftarrow \mathcal{N}$;

```

13  $i = 1, 2, \dots, g$ 
14    $L \leftarrow \emptyset;$ 
15    $Count \leftarrow 0;$ 
16   for  $w = 1, \dots, p$  do
17      $Count \leftarrow Count + 1;$ 
18     Crie o lugar comum  $p_{i,Count}$ ;
19      $L \leftarrow L \cup \{p_{i,Count}\};$ 
20      $P_{i,w}^* \leftarrow P_{i,w}^* \cup \{p_{i,Count}\};$ 
21     Defina  $I_{i,w}^* : (P_{i,w}^*, T_{i,w}) \rightarrow \mathbb{N}$ , tal que:
           •  $I_{i,w}^*(p_{i,Count}, t_{i,w}) = 1,$ 
           •  $I_{i,w}^*(p, t) = I_{i,w}^*(p, t), \forall (p, t) \in F_{i,w}^*$ 
            $N_{i,w}^* \leftarrow (P_{i,w}^*, T_{i,w}, F_{i,w}, W_{i,w}, I_{i,w}^*);$ 
22   end
23    $P_m^* \leftarrow P_m^* \cup L$ 
24 end
25
26   foreach  $p_l \in L$  do
27      $F_m^* \leftarrow F_m^* \cup \{(t_{m1}, p_l)\};$ 
28      $F_m^* \leftarrow F_m^* \cup \{(p_l, t_{m2})\};$ 
29   end
30   Defina  $W_m^* : F_m^* \rightarrow \mathbb{N}$ , tal que:
           •  $W_m^*(t_{m1}, p_l) = 1, \forall p_l \in L$ 
           •  $W_m^*(p_l, t_{m2}) = 1, \forall p_l \in L$ 
           •  $W_m^*(t, p) = W_m^*(t, p), \forall (t, p) \in F_m^*$ 
           •  $W_m^*(p, t) = W_m^*(p, t), \forall (p, t) \in F_m^*$ 
            $N_m^* \leftarrow (P_m^*, T_m, F_m^*, W_m^*, I_m);$ 
31 end

```

Assim como no caso por habilitação, o algoritmo 3.2 tem como entrada as k redes de Petri dos módulos do sistema e os g grupos de cooperação escolhidos, com seus respectivos subgrupos do modulo mestre e módulos escravos. Desta vez contudo, o grupo do mestre deverá possuir duas transições do módulo mestre, isto é $\mathcal{G}_i^{mest} = (N_m, t_{m1}, t_{m2})$. Na linha 1, atribuímos \mathcal{N} ao conjunto de redes de Petri modificadas \mathcal{N}^* . Na linha 2, executaremos o algoritmo g vezes, uma para cada grupo. Nas linhas 3 e 4, criamos um conjunto L e uma variável $Count$, em que L

será usado para salvar a informação de quantos lugares comuns foram criados para cada grupo, enquanto *Count* será usada para nomear os lugares comuns. Entre as linhas 6 e 9, criaremos um lugar comum para cada módulo escravo e modificaremos a rede de Petri daquele módulo para incluir o lugar comum, adicionando-o também ao conjunto L . Nas linhas 10 e 11, redefinimos a função $I_{i,w}$ do módulo adicionando à rede de cada escravo um arco inibidor, indo do lugar comum até cada uma das transições daquele módulo envolvidas na cooperação. Na linha 12, adicionamos todos os lugares comuns criados para aquele grupo ao módulo mestre, através do conjunto L . Entre as linhas 14 e 19, criamos dois arcos para cada lugar comum, um conectando t_{m1} ao lugar comum e outro conectando o lugar comum à t_{m2} , ambos com peso 1. Ao final do algoritmo, teremos o conjunto \mathcal{N}^* que contém cada rede de Petri modificada para incluir os lugares comuns e seus arcos.

Exemplo 3.5 *Vamos tomar como exemplo mais uma vez o sistema da figura 3.2. Desta vez, contudo, queremos impor no sistema uma cooperação por impedimento aplicando o algoritmo 3.2. Escolheremos a rede N_1 como escravo, usando a transição t_1 , e a rede N_2 como mestre, usando as transições t_2 e t_3 . Sendo assim, teremos como entradas do algoritmo os seguintes valores: $k = 3, g = 1, \mathcal{N} = \{N_1, N_2, N_3\}$, $G_1 = (\mathcal{G}_1^{mest}, \mathcal{G}_1^{esc})$, em que $\mathcal{G}_1^{mest} = (N_2, t_2, t_3)$ e $\mathcal{G}_1^{esc} = \{(N_1, t_1)\}$. Além disso, teremos $N_1 = (P_1, T_1, F_1, W_1, I_1)$, $N_2 = (P_2, T_2, F_2, W_2, I_2)$ e $N_3 = (P_3, T_3, F_3, W_3, I_3)$*

No passo 1, definimos $\mathcal{N}^ = \mathcal{N} = \{N_1, N_2, N_3\}$. Como $g = 1$, as linhas 2 a 19 são executadas apenas uma vez. Nos passos 3 e 4 definimos o conjunto $L = \emptyset$ e a variável $Count = 0$. Como o grupo possui apenas um escravo, ou seja, $p = 1$, executamos as linhas 5 à 11 apenas uma vez. Na linha 6, definimos $Count = 1$, na linha 7, criamos o lugar comum $p_{1,1}$ e na linha 8 definimos $L = \{p_{1,1}\}$. Na linha 9 definimos $P_1^* \leftarrow P_1 \cup \{p_{1,1}\}$, e na linha 11 criamos um arco inibidor indo de t_1 para $p_{1,1}$, com peso 1, isto é, . Seguimos então para a linha 12, em que adicionaremos $p_{1,1}$ ao conjunto de lugares do módulo mestre P_2 . Nas linhas 15 a 19 criamos um arco indo de t_2 para $p_{1,1}$ e um arco indo de $p_{1,1}$ para t_3 , ambos com peso 1. Ao final, temos cada uma das redes modificadas para incluírem o lugar comum criado, assim como seus arcos, como pode ser visto na figura 3.4.*

Matematicamente, teremos $\mathcal{N}^ = \{N_1^*, N_2^*, N_3^*\}$ em que $N_1^* = (P_1^*, T_1, F_1, W_1, I_1^*)$, $N_2^* = (P_2^*, T_2, F_2^*, W_2^*, I_2)$ e $N_3^* = (P_3, T_3, F_3, W_3, I_3)$. Note que, como a rede N_3 não foi envolvida no grupo de cooperação, ela não foi alterada. Para as redes modificadas, teremos pra o módulo 1 $P_1^* = \{p_1, p_2, p_{1,1}\}$, $T_1 = \{t_1\}$, $F_1 = \{(p_1, t_1), (t_1, p_2)\}$ e I_1 redefinido para ter um arco inibidor indo de $p_{1,1}$ para t_1 . Por sua vez, para o módulo 2 teremos $P_2^* = \{p_3, p_4, p_5, p_{1,1}\}$, $T_2 = \{t_2, t_3\}$, $F_2 = \{(p_3, t_2), (t_2, p_4), (p_4, t_3), (t_3, p_5), (p_{1,1}, t_3), (t_2, p_{1,1})\}$.*

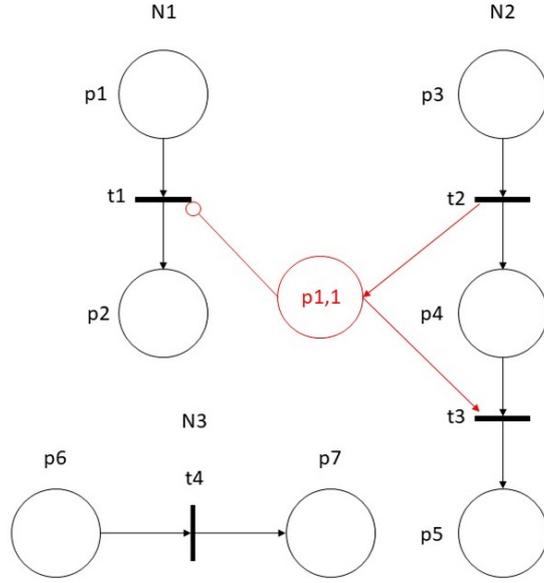


Figura 3.4: Redes de Petri do exemplo 3.5

Proposição 3.3 *Sejam $N_e = (P_e, T_e, F_e, W_e, I_e)$ e $N_m = (P_m, T_m, F_m, W_m, I_m)$ duas redes de Petri estendidas com marcações iniciais M_{0e} e M_{0m} respectivamente. Se $\mathcal{N}^* = \{N_e^*, N_m^*\}$ for obtido executando o algoritmo 3.2 com entradas $\mathcal{N} = \{N_e, N_m\}$ e $G_1 = ((N_m, t_{m1}, t_{m2}), (N_e, t_e))$, então o sistema $N_G = (P_e^* \cup P_m^*, T_e \cup T_m, F_e^* \cup F_m^*, W_g^*, I_g^*)$, com W_g^* definido com os mesmos pesos de W_e^* e W_m^* e marcação inicial $M_{0G} = [M_{0m} M_{0e} 0]$, respeitará as especificações da definição 3.5 para quaisquer M_{0m} e M_{0e} .*

Prova. Note que a segunda especificação da definição 3.5 é muito similar à especificação da definição 3.1, com t_{m2} no lugar de t_e e t_{m1} no lugar de t_m . Além disso, o lugar comum p_c obtido pela aplicação do algoritmo 3.2 sempre possuirá como transição de entrada t_{m1} e como transição de saída t_{m2} , de forma que apenas o disparo de t_{m1} pode adicionar fichas à p_c e apenas o disparo de t_{m2} pode retirá-las. Esta característica se assemelha às características de um lugar comum obtido pela aplicação do algoritmo 3.1, que terá como entrada t_m e como saída t_e . Sendo assim, para provar que a segunda especificação da definição 3.5 será satisfeita, podemos repetir a prova da proposição 20, usando t_{m2} no lugar de t_e e t_{m1} no lugar de t_m .

Para a primeira especificação da definição 3.5, por construção teremos um arco inibidor ligando p_c à transição t_e . Este arco fará com que esta transição só possa disparar quando não houver fichas em p_c . Considere agora que exista uma sequência π que viole a primeira especificação da definição 3.5, ou seja, $\Upsilon(t_{m1}, \pi) \neq \Upsilon(t_{m2}, \pi)$ e $Suffix(\pi) = t_e$. Antes do disparo de t_e , o número de fichas em p_c será igual a $\Upsilon(t_{m2}, \pi) - \Upsilon(t_{m1}, \pi)$, que precisa ser maior que zero visto que não pode existir um número negativo de fichas. Isto significa que t_e teria que disparar enquanto existem

fichas em p_c , o que não é possível devido ao arco inibidor. Sendo assim, π não será uma sequência possível no sistema após a criação do lugar comum, garantindo que a especificação da definição 3.5 seja respeitada. ■

Proposição 3.4 *Sejam $N_e = (P_e, T_e, F_e, W_e, I_e)$ e $N_m = (P_m, T_m, F_m, W_m, I_m)$ duas redes de Petri estendidas com marcações iniciais M_{0e} e M_{0m} , em que $t_e \in T_e$, $t_{m1}, t_{m2} \in T_m$ sejam três transições escolhidas. Seja também $N_G = (P_e \cup P_m, T_e \cup T_m, F_e \cup F_m, W_g, I_g)$ o sistema global, com W_g definido com os mesmos pesos de W_e e W_m , I_g definido com os mesmos pesos de I_e e I_m e marcação inicial $M_{0G} = [M_{0m} M_{0e}]$. Se $\mathcal{N}^* = \{N_e^*, N_m^*\}$ for obtido executando o algoritmo 3.2 com entradas $\mathcal{N} = \{N_e, N_m\}$ e $G_1 = ((N_m, t_{m1}, t_{m2}), (N_e, t_e))$, então o sistema global $N_G^* = (P_e^* \cup P_m^*, T_e \cup T_m, F_e^* \cup F_m^*, W_g^*, I_g^*)$, com W_g^* definido com os mesmos pesos de W_e^* e W_m^* , I_g^* definido com os mesmos pesos de I_e^* e I_m^* e marcação inicial $M_{0G}^* = [M_{0m} M_{0e} 0]$, terá a seguinte propriedade: $\Pi(N_G^*, M_{0G}^*) \subseteq \Pi(N_G, M_{0G})$.*

Prova. De forma semelhante a proposição 3.2, queremos provar que $\Pi(N_G^*, M_{0G}^*) \subseteq \Pi(N_G, M_{0G})$, ou seja, não existe nenhuma sequência de disparo π que seja possível em N_G^* e não seja possível em N_G . Sejam N_m e N_e as redes de Petri dos módulos mestre e escravo antes da aplicação do algoritmo 3.2 e N_m^* e N_e^* as redes obtidas ao final do algoritmo. Mais uma vez, para que exista uma sequência π^* em N_G^* que não exista em N_G , é preciso que: (i) exista uma sequência π_m^* em N_m^* que não existe em N_m ou; (ii) exista uma sequência π_e^* em N_e^* que não existe em N_e . Mais uma vez, isto sempre será verdade visto que não existem arcos ligando elementos de N_e a elementos de N_m , e nenhum arco com esta característica é criado com a execução do algoritmo 3.2. Sendo assim, os disparos de N_e dependem apenas da marcação de N_e , enquanto os disparos de N_m dependem apenas da marcação de N_m . O mesmo pode ser dito para N_e^* e N_m^* .

- (i) Primeiramente vamos assumir que exista uma sequência π_m^* que seja possível em N_m^* , mas não seja possível em N_m . De forma semelhante à prova da proposição 3.2, podemos escrever π_m^* como $\pi_m^* = \sigma_1 t' \sigma_2$, em que $\sigma_1 \in \Pi(N_m, M_{0m})$, $\sigma_1 \in \Pi(N_m^*, M_{0m}^*)$, $\sigma_1 t' \notin \Pi(N_m, M_{0m})$ e $\sigma_1 t' \in \Pi(N_m^*, M_{0m}^*)$, ou seja, σ_1 é uma sequência possível tanto em N_m quanto em N_m^* , e $\sigma_1 t'$ é uma sequência possível apenas em N_m^* .

Seja M' a marcação alcançada por N_m depois da sequência σ_1 , ou seja $M_{0m}[\sigma_1]M'$ e M'' a marcação alcançada pelos lugares de N_m^* que existem em N_m (todos exceto por p_l) após a sequência σ_1 , ou seja, $M_{0m}[\sigma_1]M''$. Mais uma vez, para que $\sigma_1 t'$ seja possível em N_m^* e não seja possível em N_m é necessário que: (a) $M' \neq M''$, ou seja, a rede alterada alcançou uma marcação diferente

com a mesma sequência σ_1 ou; (b) $M' = M''$ e t' está habilitada para M' em N_g^* mas não em N_g . Como, por construção, a execução do algoritmo 3.2 não altera arcos já existentes e nem cria arcos adicionando ou retirando fichas de lugares já existentes, nenhum dos dois casos mencionados acima será possível de acontecer. Portanto, concluímos que $\pi_m^* \notin \Pi(N_m^*, M_{0m}^*)$.

- (ii) Para o caso do módulo escravo, note que o único arco adicionado à rede é um arco inibidor. Este arco não é suficiente para alterar uma marcação, visto que ele não remove nem adiciona fichas. Sendo assim, não é possível existir uma sequência de disparo π_e^* em N_e^* que não exista também em N_e , visto que novas marcações não serão alcançadas e nenhuma transição foi alterada. Portanto, concluímos que $\pi_e^* \notin \Pi(N_e^*, M_{0e}^*)$.

Finalmente, como nem π_m^* e nem π_e^* poderão existir, então π^* não pode existir no sistema global N_g^* sem já existir em N_g . ■

Teorema 3.2 *Seja $\mathcal{N} = \{N_1, \dots, N_k\}$ um conjunto com k redes de Petri Extendidas e sejam $G_i = (\mathcal{G}_i^{mest}, \mathcal{G}_i^{esc})$, $i = 1, \dots, g$, os grupos de cooperação envolvendo os módulos de \mathcal{N} . Se $\mathcal{N}^* = \{N_1^*, \dots, N_k^*\}$ for obtido aplicando o algoritmo 3.2 com entradas k, g, \mathcal{N} e G_i , $i = 1, \dots, g$, então o sistema global $N_G = (\bigcup_{i=1}^k P_i^*, \bigcup_{i=1}^k T_i, \bigcup_{i=1}^k F_i^*, W_g^*, I_g^*)$ respeitará as especificações da definição (3.5) para todos os pares escravo-mestre nos grupos de cooperação.*

Prova. Seja $\mathcal{N} = \{N_1, \dots, N_k\}$ um sistema com k redes de Petri. Vamos inicialmente considerar que temos apenas um grupo de cooperação $G_1 = (\mathcal{G}_1^{mest}, \mathcal{G}_1^{esc})$. A proposição 3.3 nos garante que o sistema global $N_G = (\bigcup_{i=1}^k P_i^*, \bigcup_{i=1}^k T_i, \bigcup_{i=1}^k F_i^*, W_g^*, I_g^*)$, obtido com a aplicação do algoritmo 3.2 e com marcação inicial M_{0G} , respeita a especificação do grupo G_1 .

Suponha agora que aplicamos o algoritmo 3.2 um número x de vezes, escolhendo um novo grupo G_i cada vez, obtendo o sistema global N_G^x e o conjunto \mathcal{N}^x , de forma que N_G^x respeite todas as especificações dos grupos G_i , $i = 1, \dots, x$. Aplicando agora mais uma vez o algoritmo 3.2 para um novo grupo G_{x+1} , obtemos \mathcal{N}^{x+1} . Queremos provar que $N_G^{x+1} = (\bigcup_{i=1}^k P_i^{x+1}, \bigcup_{i=1}^k T_i, \bigcup_{i=1}^k F_i^{x+1}, W_g^{x+1}, I_g^{x+1})$ respeitará as especificações dos grupos G_i , $i = 1, \dots, x+1$.

Primeiramente, o sistema \mathcal{N}^{x+1} possuirá um novo lugar comum p_c^{x+1} . Por construção, este lugar sempre possuirá como transições de entrada e saída duas transições do módulo mestre e terá um arco inibidor conectado a uma transição do módulo escravo, independente de \mathcal{N}^x . Podemos então repetir a prova da proposição 3.3 independente da entrada escolhida para o algoritmo, de forma que as duas especificações da definição (3.5) serão respeitadas para toda e qualquer sequência de \mathcal{N}^{x+1} .

A seguir, assim como na prova do teorema 3.1, o sistema \mathcal{N} não possui arcos interligando lugares e transições de redes diferentes. Esta propriedade também é mantida em todos os módulos de \mathcal{N}^x , após x aplicações do algoritmo 3.2, visto que este algoritmo não modifica arcos ou transições já existentes na rede, e os únicos arcos criados ligam transições a lugares comuns. Como estes arcos não existem no sistema de entrada do algoritmo 3.2, podemos replicar a prova da proposição 3.4 para concluir que $\Pi(N_G^{x+1}, M_{0G}^{x+1}) \subseteq \Pi(N_G^x, M_{0G}^x)$, em que $M_{0G}^{x+1} = [M_{0G}^x 0]$. Como todas as sequências de $\Pi(N_G^x, M_{0G}^x)$ respeitam as especificações de N_G^x , então todas as sequências de $\Pi(N_G^{x+1}, M_{0G}^{x+1})$ respeitarão também estas especificações, além de respeitarem também a nova especificação imposta.

Com isso, provamos que podemos repetir este procedimento um número qualquer de vezes para um par qualquer de mestre-escravo, de forma que cada especificação imposta se mantém respeitada e toda especificação nova é incluída no sistema. ■

3.4 Análise de bloqueio do sistema em cooperação com lugares comuns

Nas seções anteriores foram apresentadas duas estratégias para a criação de um sistema em cooperação envolvendo diversos módulos. É possível, contudo, que a composição deste sistema em cooperação resulte em bloqueios não desejados na dinâmica do sistema global. Nesta seção iremos analisar os bloqueios no sistema em cooperação, procurando identificar os casos em que um bloqueio pode acontecer, assim como suas causas.

Seguindo a hipótese H3 da arquitetura considerada, iremos considerar que inicialmente as redes de Petri dos módulos individuais não possuem nenhum bloqueio. Isso significa dizer que, para suas respectivas marcações iniciais, não existe nenhuma sequência de disparos que resulta em uma marcação em que nenhuma transição pode disparar.

Com a implementação de um sistema em cooperação, é possível que bloqueios previamente inexistentes apareçam na dinâmica destas redes, como pode ser visto no exemplo 3.6.

Exemplo 3.6 *Na figura 3.5 podemos ver um caso em que originalmente tínhamos dois módulos, $N_1 = \{P_1, T_1, F_1, W_1\}$ e $N_2 = \{P_2, T_2, F_2, W_2\}$ em que $P_1 = \{p_1, p_2\}$, $T_1 = \{t_1, t_2\}$, $F_1 = \{(p_1, t_2), (p_2, t_1), (t_2, p_2), (t_1, p_1)\}$, $P_2 = \{p_3, p_4\}$, $T_2 = \{t_3, t_4\}$, $F_2 = \{(p_3, t_4), (p_4, t_3), (t_4, p_4), (t_3, p_3)\}$ e W_1 e W_2 definidos com peso igual a 1 para todos os arcos definidos. Note que, sem os lugares comuns p_{c1} e p_{c2} , ambas as redes são livres de bloqueio para as marcações iniciais $M_{01} = M_{02} = [1 \ 0]$, visto que as sequências $t_1 t_2$ e $t_4 t_3$ podem se repetir indefinidamente.*

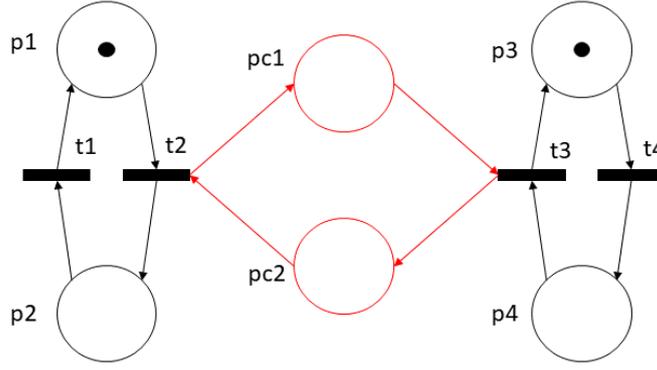


Figura 3.5: Redes de Petri do Exemplo 3.6: Módulos N_1 e N_2 em preto e lugares comuns p_{c1} e p_{c2} em vermelho.

Com a adição dos lugares comuns p_{c1} e p_{c2} , a transição t_2 não poderá disparar enquanto p_{c2} não possuir fichas, enquanto a transição t_3 não poderá disparar enquanto p_{c1} não possuir fichas. Além disso, a única forma de adicionar fichas a p_{c1} é com o disparo de t_2 , enquanto a única forma de adicionar fichas a p_{c2} é com o disparo de t_3 . Desta forma, o sistema global se encontra em um bloqueio, visto que nenhuma transição pode disparar.

O estudo de bloqueios em redes de Petri é bastante explorado na literatura, e é majoritariamente feito a partir da análise dos sifões da rede, como apresentado na subseção 2.2.1. Após a síntese do sistema em cooperação por habilitação, vamos analisar suas consequências do ponto de vista do bloqueio, inicialmente para o caso mais simples em que apenas um lugar comum foi criado.

Teorema 3.3 (*Sifões mínimos com a criação de um lugar comum*) Dadas duas redes de Petri N_1 e N_2 com conjuntos de sifões mínimos \mathcal{S}_{m1} e \mathcal{S}_{m2} . A criação de um lugar comum p_c conectando N_1 (mestre) à N_2 (escravo) representando uma cooperação por habilitação não alterará o conjunto de sifões mínimos do sistema, ou seja, o conjunto de sifões mínimos da nova rede contendo N_1 , N_2 e p_c será igual a $\mathcal{S}_{m1} \cup \mathcal{S}_{m2}$.

Prova 3.3.1 Definindo $N_1 = (P_1, T_1, F_1, W_1)$ como módulo mestre, com conjunto de sifões mínimos \mathcal{S}_{m1} , $N_2 = (P_2, T_2, F_2, W_2)$ como módulo escravo com conjunto de sifões mínimos \mathcal{S}_{m2} e $N_f = (P_f, T_f, F_f, W_f)$ como a rede total após a adição do lugar comum p_c , dividiremos a prova em duas partes. Primeiro, vamos assumir que depois da adição do lugar comum, um sifão mínimo S_m que contém lugares de P_1 e P_2 mas não contém p_c foi criado. Note que após a adição de lugares comuns, seguindo as equações 2.1 e 2.2 teremos:

$$\nexists p \in P_1 \setminus \{p_c\}, \nexists t \in T_2 \Rightarrow (p, t) \in F_f \vee (t, p) \in F_f \quad (3.7)$$

$$\nexists p \in P_2 \setminus \{p_c\}, \nexists t \in T_1 \Rightarrow (p, t) \in F_f \vee (t, p) \in F_f \quad (3.8)$$

Ou seja, não existem arcos conectando lugares de P_1 à transições de T_2 nem conectando lugares de P_2 à transições de T_1 .

Podemos separar os lugares de S_m em dois conjuntos $S_m = S_{m1} \cup S_{m2}$ em que S_{m1} e S_{m2} são subconjuntos contendo apenas lugares de P_1 e P_2 respectivamente. Consequentemente, teremos $S_m \bullet = S_{m1} \bullet \cup S_{m2} \bullet$ e $\bullet S_m = \bullet S_{m1} \cup \bullet S_{m2}$. Como $\bullet S_m \subseteq S_m \bullet$ é sempre verdade, visto que S_m é um sifão, então $(\bullet S_{m1} \cup \bullet S_{m2}) \subseteq (S_{m1} \bullet \cup S_{m2} \bullet)$. Note que 3.7 e 3.8 nos garantem que $S_{m1} \bullet$ e $\bullet S_{m1}$ conterão apenas transições de T_1 . Podemos concluir então que $\bullet S_{m1} \subseteq S_{m1} \bullet$ e $\bullet S_{m2} \subseteq S_{m2} \bullet$, ou seja, S_{m1} e S_{m2} são sifões, o que implica que S_m não é mínimo.

A seguir, vamos assumir que depois da adição do lugar comum p_c , um sifão mínimo $S_m : p_c \in S_m$ foi criado. Por definição temos que $\bullet S_m \subseteq S_m \bullet$. Por construção, o lugar comum terá como entrada uma transição t_i de T_1 , ou seja, $\exists t_i \in T_1 \Rightarrow (t_i, p_c) \in F_f$. Se $p_c \in S_m$, então certamente $t_i \in \bullet S_m$. Para S_m ser um sifão, deve ser verdade que $t_i \in S_m \bullet$. Como as equações 3.7 e 3.8 permanecem verdadeiras para este caso, então teremos que $\exists p \in P_1, t_i \in T_1 \Rightarrow (p, t_i) \in F_f \wedge (t_i, p_c) \in F_f$. Isso implica que $\exists p \in P_1 \Rightarrow p \in S_m$. Em outras palavras, se p_c faz parte de S_m , então necessariamente deve existir um lugar de N_1 que também faça parte de S_m . Se chamarmos de S_{m1} o conjunto de lugares de S_m que pertencem a P_1 , de maneira similar à anterior, podemos concluir que: $S_{m1} \neq \emptyset$ e $\bullet S_{m1} \subseteq S_{m1} \bullet$. Sendo assim, existe um sifão $S_{m1} \subseteq S_m$ com menor número de lugares que S_m , o que resulta em uma contradição com a afirmação inicial que S_m era mínimo. Com isso, concluímos que não pode existir um sifão mínimo contendo um lugar comum.

Finalmente, como nenhuma transição foi adicionada ou removida e todos os arcos criados envolvem o lugar comum, então os sifões mínimos previamente existentes nas redes N_1 e N_2 não serão modificados.

Podemos concluir então que o conjunto \mathcal{S}_m de todos os sifões mínimos de N_f será igual a $\mathcal{S}_{m1} \cup \mathcal{S}_{m2}$.

Corolário 3.1 *O teorema 3.3 será válido para um caso em que mais de uma co-operação por habilitação foi implementada, contanto que todos os lugares comuns criados possuam o mesmo mestre e o mesmo escravo.*

O Corolário 3.1 pode ser provado com a mesma prova do teorema 3.3, visto que todas as considerações feitas se mantêm para o caso em que múltiplos lugares comuns foram criados com o mesmo mestre e escravo.

Uma consequência direta do teorema 3.3 é o fato de que se o conjunto de sifões mínimos dos módulos iniciais é controlável, após a adição do lugar comum a rede global será livre de bloqueios, visto que todos os seus sifões serão controláveis (vide 2.1).

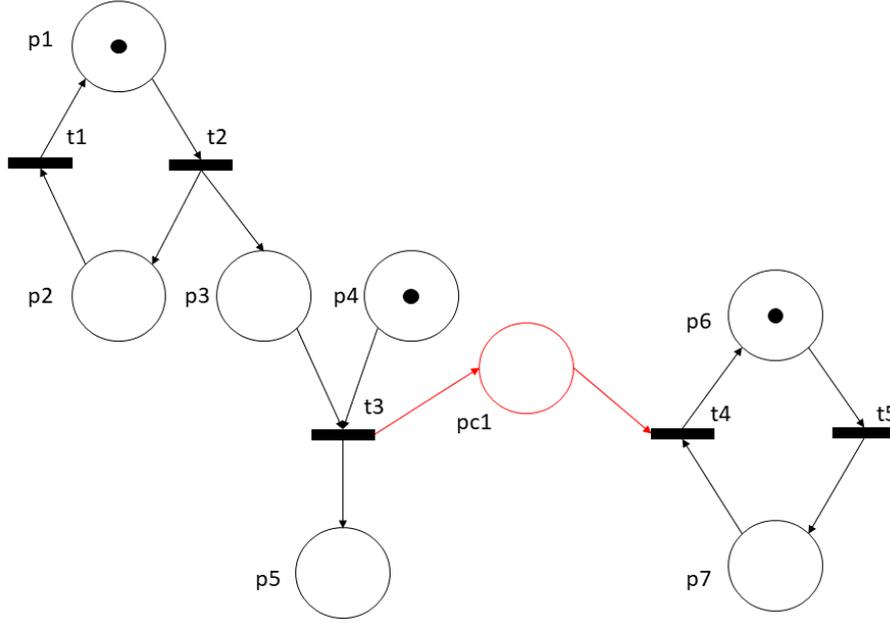


Figura 3.6: Redes de Petri do Exemplo 3.7: Módulos N_1 e N_2 em preto e lugar comum p_{c1} em vermelho.

Contudo, o fato de as redes dos módulos serem inicialmente livres de bloqueio não implica que todos os sífões mínimos da rede são controláveis, visto que uma rede pode possuir um sífão mínimo não controlável e continuar sendo livre de bloqueios. Como as transições de um sífão nunca irão disparar a partir do momento que o sífão fica sem fichas, caso a cooperação por habilitação seja feita envolvendo um sífão não controlável do módulo mestre, é possível que criemos um bloqueio local sem necessariamente criar nenhum sífão mínimo novo, como pode ser visto no exemplo 3.7.

Exemplo 3.7 As redes da figura 3.6 são definidas como $N_1 = \{P_1, T_1, F_1, W_1\}$ e $N_2 = \{P_2, T_2, F_2, W_2\}$ em que $P_1 = \{p_1, p_2, p_3, p_4, p_5\}$, $T_1 = \{t_1, t_2, t_3\}$, $F_1 = \{(p_1, t_2), (p_2, t_1), (t_2, p_2), (t_1, p_1), (t_2, p_3), (p_4, t_3), (t_3, p_5)\}$, $P_2 = \{p_6, p_7\}$, $T_2 = \{t_4, t_5\}$, $F_2 = \{(p_6, t_5), (p_7, t_4), (t_5, p_7), (t_4, p_6)\}$ e W_1 e W_2 definidos com peso igual a 1 para todos os arcos definidos.

Note que $S_1 = \{p_4\}$ é um sífão, visto que $S_1 \bullet = \{t_3\}$, $\bullet S_1 = \{\}$ e $\bullet S_1 \subset S_1 \bullet$. Além disso, com o disparo de t_3 todas as fichas de S_1 são removidas, o que significa que S_1 é um sífão não-controlável. Note, contudo, que apesar da existência de um sífão não controlável em N_1 , a rede ainda é livre de bloqueios, visto que a sequência $t_2 t_1$ pode se repetir indefinidamente.

Quando inserimos na rede o lugar comum p_{c1} , contudo, nos deparamos com um problema. A rede N_1 consegue continuar sua dinâmica sem problemas, mas a rede N_2 , que previamente não possuía bloqueios, se encontrará em bloqueio depois da sequência $t_3 t_4 t_5$. Uma vez que t_3 nunca mais poderá disparar novamente, visto que

não existem mais fichas em p_4 , p_{c1} não receberá mais fichas, impossibilitando t_4 de disparar novamente.

A seguir, vamos analisar o caso em que duas cooperações por habilitação são implementadas entre um par de redes N_1 e N_2 , em que N_1 opera como o módulo mestre da primeira cooperação e como módulo escravo da segunda.

Exemplo 3.8 Voltando para a figura 3.5 do exemplo 3.6, em que temos as redes N_1 e N_2 . Note que o conjunto de lugares $S_1 = \{p_{c1}, p_{c2}\}$ compõe um sifão, visto que $\bullet S_1 = S_1 \bullet = \{t_2, t_3\}$. Como, para a marcação inicial das redes, todos os lugares de S_1 se encontram sem fichas, S_1 é um sifão não-controlável. Sendo assim, t_2 e t_3 nunca poderão disparar, visto que estes lugares nunca receberão fichas.

Com isso, podemos apresentar um teorema para análise de bloqueios em sistemas em cooperação baseado em sifões contendo lugares comuns.

Teorema 3.4 (Bloqueio com lugares comuns) Após a síntese do sistema em cooperação, se não existe nenhum sifão não-controlável que contenha algum lugar comum, o sistema em cooperação, assim como seus módulos, serão livres de bloqueio.

Prova 3.4.1 Vamos assumir que, após a síntese do sistema em cooperação, exista uma sequência de disparos de transições π que leva o sistema a um bloqueio. Além disso, vamos assumir que não existem sifões não-controláveis contendo lugares comuns após a síntese do sistema em cooperação.

Após a sequência de disparos π , de acordo com o teorema 2.1, o conjunto de lugares sem fichas irá compor um sifão S_1 . Como o S_1 não possui fichas, ele é um sifão não-controlável. Por hipótese, S_1 não conterá nenhum lugar comum. Como não existem arcos ligando lugares de uma rede a transições da outra, e vice-versa, então sempre podemos dividir S_1 em outros dois sifões S_1^1 e S_1^2 , em que cada subconjunto contém apenas lugares de uma das duas redes.

Finalmente, ambos S_1^1 e S_1^2 serão sifões não controláveis, visto que todos os seus lugares se encontram sem fichas. Como a adição dos lugares comuns não altera arcos já existentes e nem adiciona transições, para que S_1^1 e S_1^2 existam no sistema em cooperação eles deveriam também existir nos módulos originais, e se eles são não-controláveis no sistema em cooperação, eles serão não-controláveis nos módulos individuais também. Sendo assim, se algum módulo se encontra em bloqueio por conta de S_1^1 ou S_1^2 , isso implica que originalmente os módulos não eram livres de bloqueio, visto que a adição dos lugares comuns não altera os sifões já existentes ou sua controlabilidade. Como chegamos a uma contradição, concluímos que, se não existe nenhum sifão não-controlável que contenha algum lugar comum, o sistema em cooperação, assim como seus módulos, serão livres de bloqueio.

A tarefa de identificar todos os sifões não controláveis de um sistema, contudo, não é trivial. Metodologias para a busca de sifões e para a determinação da controlabilidade dos mesmos foram exploradas em [15] e [16], e possuem complexidade computacional na ordem de $O(2^n)$. Apesar de neste trabalho precisarmos identificar apenas os sifões contendo lugares comuns, esta restrição não altera a ordem da complexidade dos algoritmos de busca, visto que limitar a busca para incluir um lugar específico teria complexidade $O(2^{(n-1)}) = O(2^n)$.

Por fim, observe que na seção 3.1.2 definimos que as especificações locais seriam respeitadas se a equação 3.1 fosse satisfeita. Contudo, nas provas dos teoremas 3.1 e 3.2 concluímos que com a aplicação dos algoritmos 3.1 e 3.2, teremos $\Pi(N_G^{x+1}, M_{0G}^{x+1}) \subseteq \Pi(N_G^x, M_{0G}^x)$. Em palavras, o conjunto de sequências do sistema em cooperação deveria ser igual ao conjunto de sequências do sistema original, mas pode ser que ele acabe sendo um subconjunto com algumas sequências faltando. Esta diferença ocorrerá quando um bloqueio for criado no sistema, de forma que algumas sequências não poderão mais ocorrer e os conjuntos antes e depois da cooperação não serão iguais. A igualdade será atingida quando o sistema for livre de bloqueios.

Capítulo 4

Aplicação prática

Neste capítulo, apresentaremos as duas cooperações propostas no capítulo 3, aplicadas em uma planta mecatrônica real, com o objetivo de estabelecer entre os módulos uma cooperação livre de bloqueios.

Na seção 4.1, apresentaremos a planta mecatrônica utilizada, assim como seus módulos individuais, e na seção 4.2, apresentaremos as cooperações implementadas, assim como a estratégia utilizada para implementá-las na prática.

4.1 Planta mecatrônica

A planta mecatrônica utilizada neste trabalho pode ser vista na figura 4.1. Ela consiste em uma planta pneumática com diversos atuadores e sensores cujo principal objetivo é a montagem de cubos, assim como seu armazenamento. Estes cubos, por sua vez, são constituídos por duas peças, que podem ser de metal ou plástico e podem ter coloração branca ou preta, como pode ser visto na figura 4.2. A planta se divide em cinco módulos que operam de forma independente, cada um realizando uma etapa do processo de montagem e armazenamento de cubos. O fluxograma da planta pode ser visto na figura 4.3.

4.1.1 Módulo 1

Este módulo é composto por duas unidades de armazenamento de peças, uma esteira e um grupo de sensores ao longo da esteira, como pode ser visto na figura 4.4.

O processo se inicia com a primeira unidade de armazenamento, que conterà peças de metal que podem estar com suas cunhas viradas para cima ou para baixo. Através de um atuador pneumático, a primeira unidade de armazenamento empurra uma peça na direção da esteira, que por sua vez levará esta peça até um conjunto de sensores, dentre os quais temos: um sensor capacitivo responsável por detectar passagem de qualquer peça; um sensor indutivo para detectar se a peça é metálica

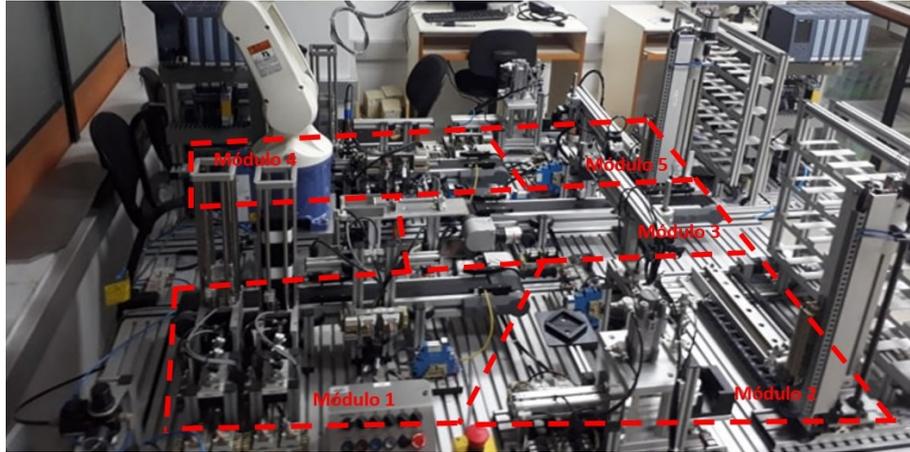


Figura 4.1: Planta mecatrônica disponibilizada no Laboratório de Controle e Automação (LCA) da UFRJ

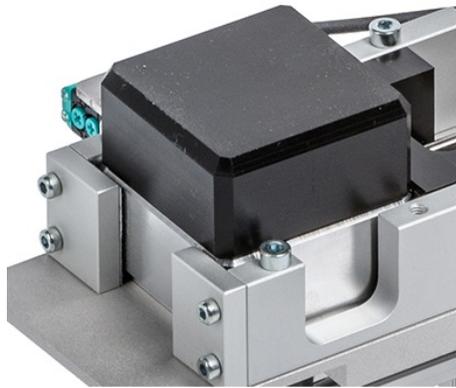


Figura 4.2: Peças plásticas que são montadas na planta

ou plástica; um sensor óptico responsável por detectar se a peça tem coloração branca ou preta; e um sensor de profundidade para saber se a peça está virada para cima ou para baixo. A verificação da posição é importante, visto que nenhum módulo consegue rotacionar as peças, e elas precisam estar na posição correta para a confecção do cubo, de forma que primeira peça do cubo deve ser metálica e estar virada para cima enquanto a segunda deve ser plástica e estar virada para baixo. Após passar por todos estes sensores, a peça chega ao final da esteira onde existe um último sensor de presença óptico. Nesta etapa, um relé de dois estados também é acionado pelo controlador do módulo, indicando se a peça no final da esteira possui os atributos de tipo, cor e profundidade corretos ou não. Neste trabalho, configuramos o sistema de forma que a primeira peça deva ser de metal e virada para cima. O sistema se manterá em estado de espera até que a peça seja removida da esteira, o que é detectado pelo mesmo sensor de presença ao final da mesma. Quando a peça é retirada, se ela não possuía todos os atributos corretos (metal e virada para cima), uma nova peça será inserida na esteira pela primeira unidade de

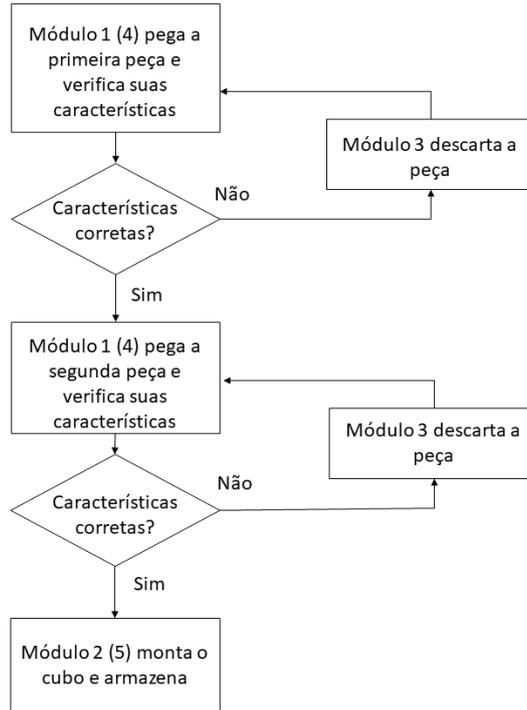


Figura 4.3: Fluxograma da planta mecatrônica

armazenamento, repetindo o processo anterior. Se, contudo, a peça possuía todos os atributos necessários, uma nova peça será inserida na esteira pela segunda unidade de armazenamento, que por sua vez conterá peças de plástico da cor branca ou preta e viradas para cima ou para baixo. Nesta segunda etapa, configuramos o sistema para trabalhar com peças plásticas pretas viradas para baixo. A peça seguirá o mesmo percurso da primeira parte, terminando no final da esteira. Se esta peça possuir os atributos necessários, o sistema volta para o estado inicial e o processo inteiro se repete, começando novamente pela primeira unidade de armazenamento. Se, contudo, a peça não possuir os atributos necessários, uma nova peça é inserida na esteira pela segunda unidade de armazenamento.

4.1.2 Módulo 2

Este módulo é composto por um braço mecânico, uma unidade de prensa de cubos e uma unidade de estoque, como pode ser visto na figura 4.5. O braço mecânico é posicionado de forma que ele consiga acessar as unidades de prensa e estoque, assim como os módulos 1 e 3.

Inicialmente, o braço deve ser calibrado usando um conjunto de sensores internos para garantir que ele se encontre na sua posição inicial. Uma vez calibrado, o braço retira uma peça da esteira do módulo 1 e leva a mesma até o módulo 3 (caso ela precise ser descartada) ou até a unidade de prensa (caso ela tenha os atributos

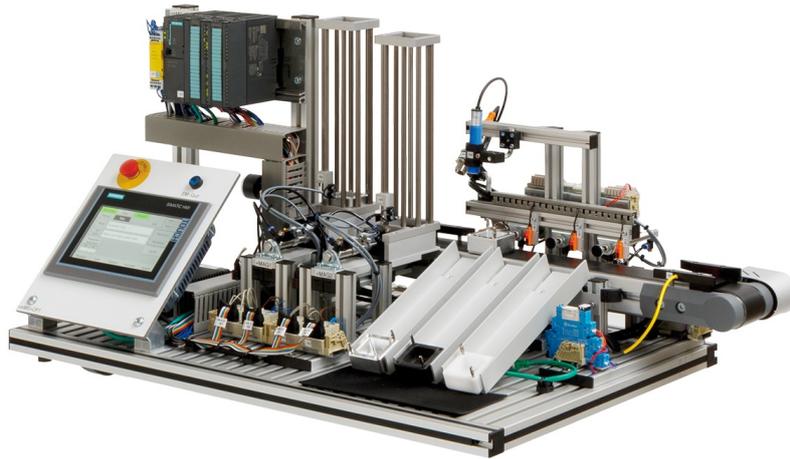


Figura 4.4: Módulo 1



Figura 4.5: Módulo 2

necessários para a montagem do cubo). O braço repetirá este processo até que duas peças tenham sido levadas para a prensa, onde as duas peças serão unidas e o cubo será montado. Como mencionado anteriormente, para que a confecção do cubo nesta etapa seja feita corretamente, é necessário que as duas peças estejam nas posições corretas, a primeira sendo de metal e virada para cima e segunda sendo de plástico e preta e virada para baixo. Uma vez que o cubo tenha sido montado, o braço o retira da prensa e o leva até a unidade de estoque, que por sua vez finalizará o processo levando o cubo até uma posição vaga do estoque. A escolha da posição é feita com base na memória do sistema, considerando que no início do processo, todas as posições estão livres.

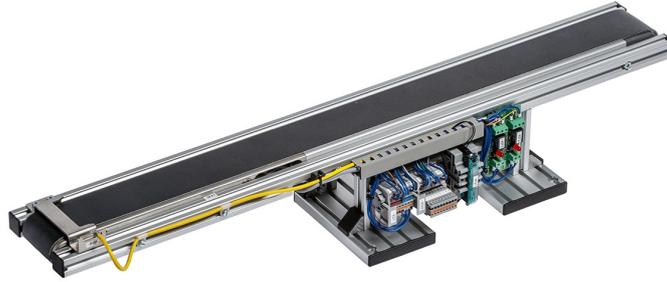


Figura 4.6: Módulo 3

4.1.3 Módulo 3

Este módulo é composto por uma esteira, como pode ser visto na figura 4.6. A esteira se inicia parada e assim que uma peça é disponibilizada pelos módulos 2 ou 5 ela é acionada para descartar aquela peça. Ao final da esteira, um sensor indica quando a peça foi descartada, fazendo a esteira parar novamente.

Uma característica importante do módulo 3 é que as peças serão disponibilizadas pelos módulos 2 e 5 através de braços mecânicos. Por conta de uma limitação espacial, se um destes módulos tentar levar uma peça para o módulo 3 enquanto o outro já estiver levando outra peça, os braços mecânicos colidirão entre si e o sistema pode ser danificado. Desta forma, o sistema em cooperação deve impedir que colisões deste tipo aconteçam.

4.1.4 Módulos 4 e 5

O quarto módulo é idêntico ao primeiro, contendo também duas unidades de armazenamento de peças, uma esteira e diversos sensores ao longo da mesma. Diferente do primeiro módulo, contudo, neste módulo iremos montar cubos compostos por uma peça metálica e uma peça plástica branca, descartando todas as peças que contenham o tipo, cor ou posição não desejadas.

De forma semelhante, o quinto módulo é idêntico ao segundo, contendo um braço robótico, uma prensa e uma unidade de estoque. A única diferença entre estes dois módulos é a direção pela qual o braço mecânico acessa o módulo 3, de forma que o módulo 5 acessa pela direita enquanto o módulo 2 acessa pela esquerda. Como mencionado anteriormente, devido a uma limitação espacial, os dois braços não podem tentar acessar o módulo 3 simultaneamente, visto que este cenário resultaria em uma colisão.

4.2 Modelagem dos módulos

Originalmente, cada um dos módulos foi modelado para operar sozinho, sem qualquer comunicação com os demais módulos. Começando pelo módulo 1, na figura 4.7 podemos ver uma abstração de sua rede de Petri, em que cada lugar representa uma ação e cada transição representa uma condição, como descrito na tabela 4.1.

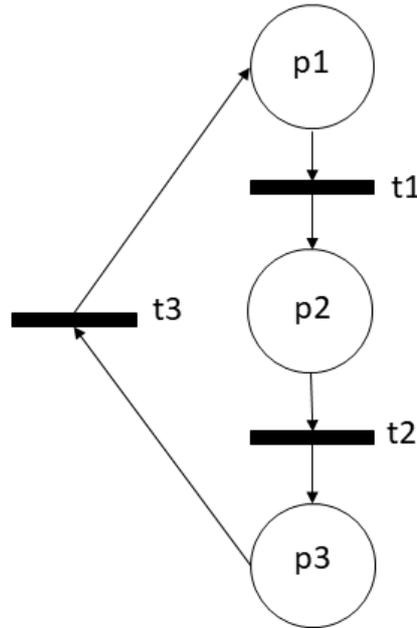


Figura 4.7: Rede de Petri do módulo 1

Tabela 4.1: Condições e ações da rede de Petri do módulo 1

Módulo 1			
Lugar	Ação	Transição	Condição
p_1	Coloca peça na esteira	t_1	Peça colocada
p_2	Verifica as características da peça	t_2	Fim da esteira
p_3	Estado de espera	t_3	Peça retirada da esteira

Nesta representação simplificada, iniciamos o sistema com apenas uma ficha em p_1 , que representa uma peça sendo colocada na esteira por alguma das duas unidades de armazenamento de peças. Após a peça se encontrar na esteira, a ficha passa para p_2 e o processo de verificar os atributos da peça é iniciado. Ao final deste processo, a ficha passa para p_3 e o sistema entra em estado de espera, aguardando até que a peça seja removida. Por fim, quando a peça é removida, a ficha volta para p_1 onde uma nova peça será disponibilizada na esteira.

A rede de Petri simplificada do módulo 2 pode ser vista na figura 4.8 com suas respectivas ações e condições descritas na tabela 4.2.

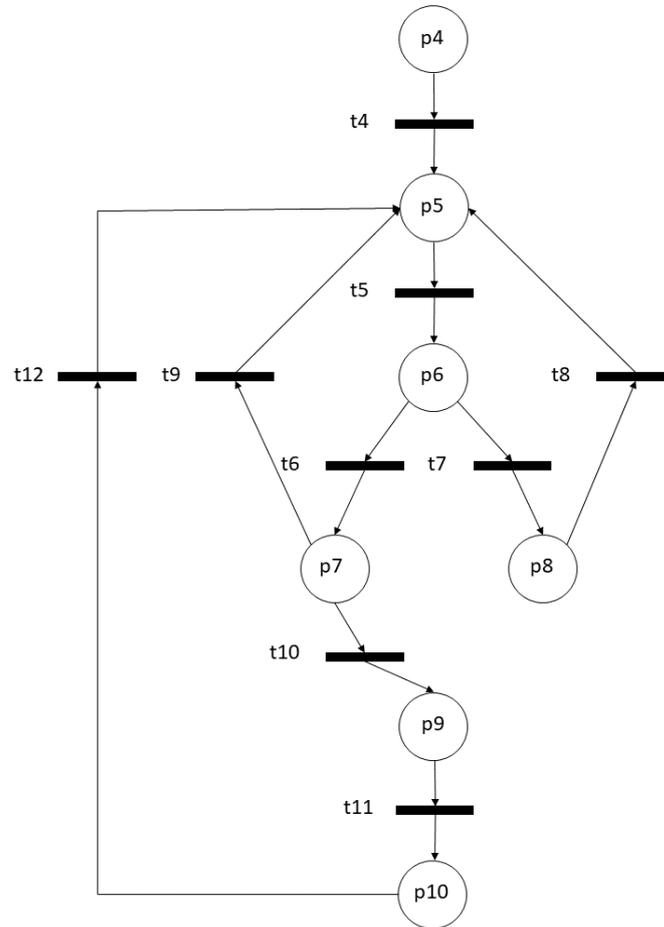


Figura 4.8: Rede de Petri do módulo 2

O sistema se inicia com uma ficha em p_4 , com a calibração da unidade de estoque e da prensa. A ficha passa então para p_5 representando a calibração do braço mecânico, levando-o para sua posição inicial. Seguindo para p_6 , o braço retira uma peça da esteira do módulo 1 e verifica o sensor que indica se a peça deve ser descartada ou usada na confecção do cubo. No primeiro caso, teremos o disparo de t_7 levando a ficha para p_8 , simbolizando a peça sendo levada até o módulo 3 para ser descartada. Caso, contudo, a peça esteja marcada para a confecção, teremos o disparo de t_6 levando a ficha para p_7 , simbolizando a peça sendo levada até a prensa. Neste momento, caso esta seja a primeira peça a ser disponibilizada na prensa, teremos o disparo de t_9 , levando a ficha de volta para p_5 onde o braço será calibrado novamente para pegar outra peça. Se, no entanto, já possuímos duas peças na prensa, teremos o disparo de t_{10} levando a ficha para p_9 , onde a confecção do cubo será feita pela prensa, e em seguida para p_{10} , onde o cubo será armazenado na unidade de estoque. Ao fim deste processo, a ficha volta para p_4 para dar início a um novo ciclo.

Tabela 4.2: Condições e ações da rede de Petri do módulo 2

Módulo 2			
Lugar	Ação	Transição	Condição
p_4	Calibração do estoque e prensa	t_4	Fim da calibração
p_5	Calibração do braço	t_5	Fim da calibração
p_6	Pega peça do módulo 1	t_6	Peça marcada para confecção do cubo
p_7	Leva peça até a prensa	t_7	Peça marcada para descarte
p_8	Leva peça até o módulo 3	t_8	Peça descartada
p_9	Prensa o cubo	t_9	Primeira peça na prensa
p_{10}	Armazena o cubo	t_{10}	Segunda peça na prensa
-	-	t_{11}	Fim da prensa
-	-	t_{12}	Fim do armazenamento

A seguir, podemos ver na figura 4.9 a rede de Petri simplificada do módulo 3, com suas respectivas ações e condições descritas na tabela 4.3. Neste módulo, iniciamos o sistema com uma ficha em p_{11} , simbolizando que o sistema se encontra em estado de espera, sem realizar qualquer ação. O disparo de t_{13} ou de t_{14} representa a chegada de uma peça para descarte, o que leva a ficha para p_{12} iniciando o processo de descarte da peça. Ao final deste processo, a ficha volta para p_{11} , voltando o sistema para seu estado de espera.

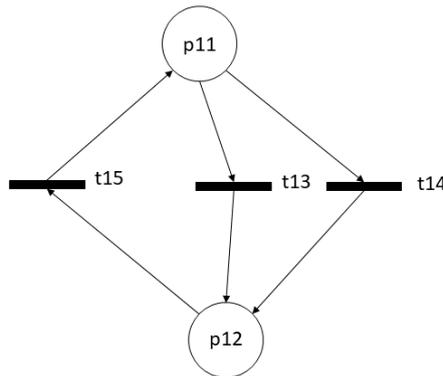


Figura 4.9: Rede de Petri do módulo 3

Finalmente, podemos ver as redes simplificadas dos módulos 4 e 5 nas figuras 4.10 e 4.11, com as tabelas de ações e condições 4.4 e 4.5 respectivamente. A dinâmica do módulo 4 é idêntica a dinâmica do módulo 1, enquanto a dinâmica do módulo 5 é idêntica a dinâmica do módulo 2.

Tabela 4.3: Condições e ações da rede de Petri do módulo 3

Módulo 3			
Lugar	Ação	Transição	Condição
p_{11}	Estado de espera	t_{13}	Peça vinda da esquerda (módulo 3)
p_{12}	Esteira ligada para descarte	t_{14}	Peça vinda da direita (módulo 5)
-	-	t_{15}	Fim do descarte

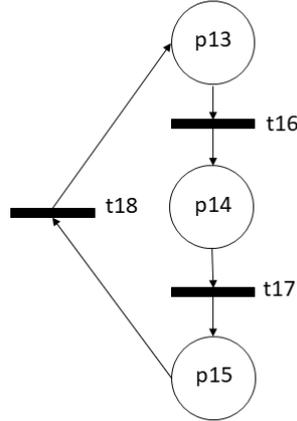


Figura 4.10: Rede de Petri do módulo 4

4.3 Cooperações implementadas

O objetivo deste trabalho é implementar entre os módulos cooperações que resultem na operação em conjunto do sistema global, respeitando as seguintes especificações: (I) o módulo 2 não pode tentar mover uma peça antes que ela tenha sido disponibilizada pelo módulo 1; (II) o módulo 5 não pode tentar mover uma peça antes que ela tenha sido disponibilizada pelo módulo 4; (III) o módulo 3 não pode tentar descartar uma peça vinda do módulo 2 sem que este tenha disponibilizado tal peça; (IV) o módulo 3 não pode tentar descartar uma peça vinda do módulo 5 sem que este tenha disponibilizado tal peça; (V) o módulo 5 não pode tentar levar uma peça para o módulo 3 enquanto o módulo 2 estiver fazendo o mesmo; (VI) o módulo 2 não pode tentar levar uma peça para o módulo 3 enquanto o módulo 5 estiver fazendo o mesmo;

Inicialmente, teremos as redes de Petri individuais de cada um dos módulos, como mostrado nas figuras 4.7, 4.8, 4.9, 4.10 e 4.11 da seção 4.2. Matematicamente, teremos para o módulo 1, a rede $N_1 = \{P_1, T_1, F_1, W_1, I_1\}$, com o conjunto de lugares $P_1 = \{p_1, p_2, p_3\}$, o conjunto de transições $T_1 = \{t_1, t_2, t_3\}$ e o conjunto de arcos $F_1 = \{(p_1, t_1), (t_1, p_2), (p_2, t_2), (t_2, p_3), (p_3, t_3), (t_3, p_1)\}$. Para o módulo 2, teremos a rede $N_2 = \{P_2, T_2, F_2, W_2, I_2\}$, o con-

Tabela 4.4: Condições e ações da rede de Petri do módulo 4

módulo 4			
Lugar	Ação	Transição	Condição
p_{13}	Coloca peça na esteira	t_{16}	Peça colocada
p_{14}	Verifica as características da peça	t_{17}	Fim da esteira
p_{15}	Estado de espera	t_{18}	Peça retirada da esteira

Tabela 4.5: Condições e ações da rede de Petri do módulo 5

módulo 5			
Lugar	Ação	Transição	Condição
p_{16}	Calibração do estoque e prensa	t_{19}	Fim da calibração
p_{17}	Calibração do braço	t_{20}	Fim da calibração
p_{18}	Pega peça do módulo 1	t_{21}	Peça marcada para confecção do cubo
p_{19}	Leva peça até a prensa	t_{22}	Peça marcada para descarte
p_{20}	Leva peça até o módulo 3	t_{23}	Peça descartada
p_{21}	Prensa o cubo	t_{24}	Primeira peça na prensa
p_{22}	Armazena o cubo	t_{25}	Segunda peça na prensa
-	-	t_{26}	Fim da prensa
-	-	t_{27}	Fim do armazenamento

junto de lugares $P_2 = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}\}$, o conjunto de transições $T_2 = \{t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}\}$ e o conjunto de arcos $F_2 = \{(p_4, t_4), (t_4, p_5), (p_5, t_5), (t_5, p_6), (p_6, t_6), (t_6, p_7), (p_7, t_7), (t_7, p_8), (p_8, t_8), (t_8, p_9), (p_9, t_9), (t_9, p_{10}), (p_{10}, t_{10}), (t_{10}, p_1), (p_1, t_1), (t_1, p_2), (p_2, t_2), (t_2, p_3), (p_3, t_3), (t_3, p_4)\}$. Para o módulo 3, teremos a rede $N_3 = \{P_3, T_3, F_3, W_3, I_3\}$, com conjunto de lugares $P_3 = \{p_{11}, p_{12}\}$, conjunto de transições $T_3 = \{t_{13}, t_{14}, t_{15}\}$ e conjunto de arcos $F_3 = \{(p_{11}, t_{13}), (t_{13}, p_{12}), (p_{12}, t_{14}), (t_{14}, p_{11}), (p_{11}, t_{15}), (t_{15}, p_{12})\}$. Para o módulo 4, teremos a rede $N_4 = \{P_4, T_4, F_4, W_4, I_4\}$, o conjunto de lugares $P_4 = \{p_{13}, p_{14}, p_{15}\}$, o conjunto de transições $T_4 = \{t_{16}, t_{17}, t_{18}\}$ e o conjunto de arcos $F_4 = \{(p_{13}, t_{16}), (t_{16}, p_{14}), (p_{14}, t_{17}), (t_{17}, p_{15}), (p_{15}, t_{18}), (t_{18}, p_{13})\}$. Finalmente, para o módulo 5, teremos a rede $N_5 = \{P_5, T_5, F_5, W_5, I_5\}$, o conjunto de lugares $P_5 = \{p_{16}, p_{17}, p_{18}, p_{19}, p_{20}, p_{21}, p_{22}\}$, o conjunto de transições $T_5 = \{t_{19}, t_{20}, t_{21}, t_{22}, t_{23}, t_{24}, t_{25}, t_{26}\}$ e o conjunto de arcos $F_5 = \{(p_{16}, t_{19}), (t_{19}, p_{17}), (p_{17}, t_{20}), (t_{20}, p_{18}), (p_{18}, t_{21}), (t_{21}, p_{19}), (p_{19}, t_{22}), (t_{22}, p_{20}), (p_{20}, t_{23}), (p_{20}, t_{24}), (t_{23}, p_{17}), (t_{24}, p_{17}), (p_{19}, t_{25}), (t_{25}, p_{21}), (p_{21}, t_{26}), (t_{26}, p_{22}), (p_{22}, t_{27}), (t_{27}, p_{17})\}$.

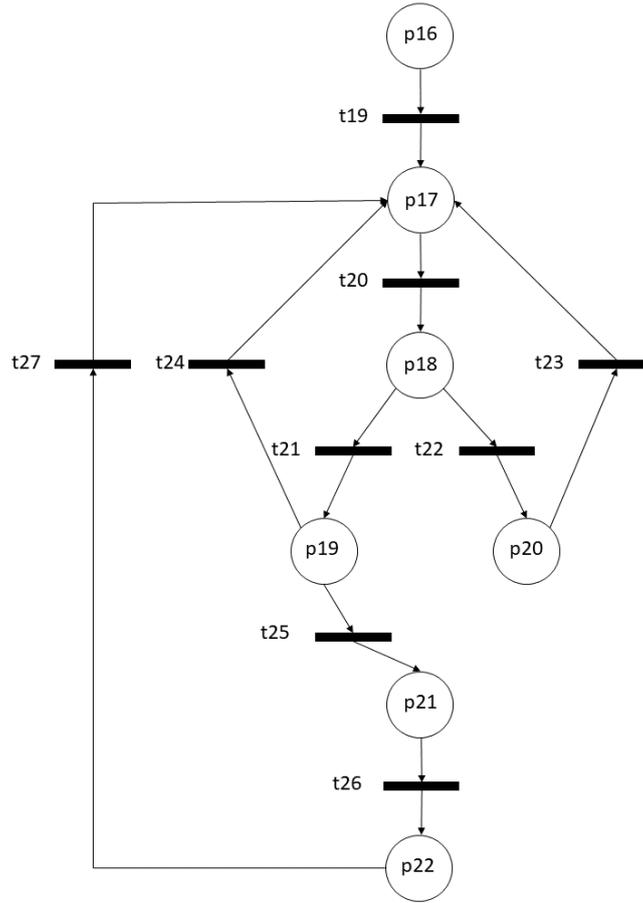


Figura 4.11: Rede de Petri do módulo 5

Note que a função W de cada uma das redes foi definida de forma que todos os arcos possuem peso 1. Além disso, como nenhuma rede possui arcos inibidores inicialmente, os conjuntos I de cada uma das redes serão iguais ao conjunto vazio.

Para implementar a especificação (I) optamos por usar uma cooperação por habilitação. Primeiramente, criamos o grupo de cooperação G_1 em que o módulo 1 opera como mestre e o módulo 2 como escravo, escolhendo a transição t_2 do módulo 1 e a transição t_5 do módulo 2 para participarem da cooperação. Escolhemos a transição t_2 pois seu disparo representa a peça chegando no final da esteira, momento no qual ela já pode ser movida pelo módulo 2, e escolhemos a transição t_5 pois ela representa o fim da calibração do braço, de forma que seu disparo inicia a sequência de pegar uma peça do módulo 1. Desta forma, seguindo a equação 3.1, a especificação que define a cooperação por habilitação para este primeiro grupo será: $\forall \pi \in \Pi(N_g, M_{0g}), \Upsilon(t_5, \pi) \leq \Upsilon(t_2, \pi)$, em que N_g representa a rede do sistema global após a cooperação e M_{0g} sua marcação inicial. Para obter o sistema em cooperação, executaremos o algoritmo 3.1 com entradas: número de módulos envolvidos na cooperação $k = 2$ (módulos 1 e 2), número de grupos de cooperação $g = 1$ (Grupo G_1), conjunto de redes de Petri dos módulos envolvidos $\mathcal{N} = \{N_1, N_2\}$

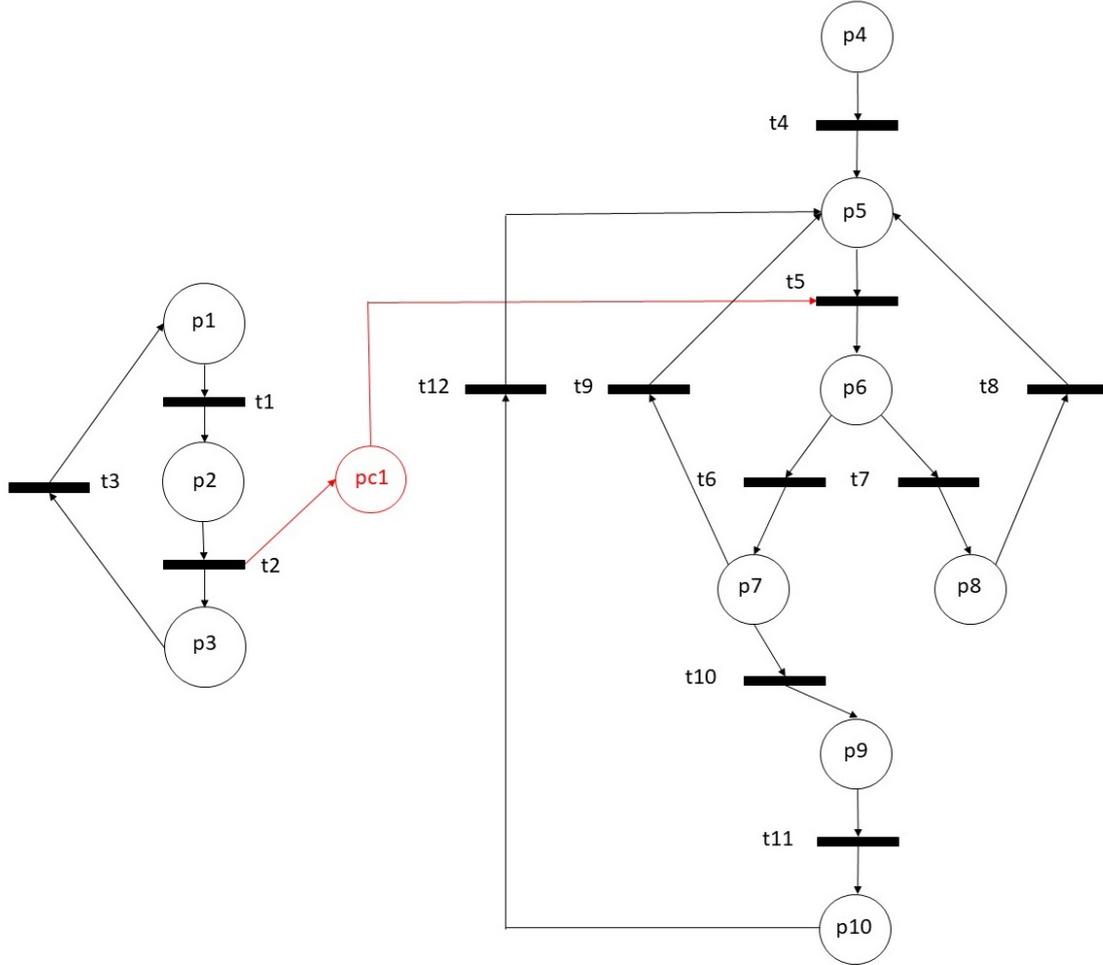


Figura 4.12: Representação da cooperação entre os módulos 1 e 2, respectivamente

e $G_1 = (\mathcal{G}_1^{mest}, \mathcal{G}_1^{esc})$, em que $\mathcal{G}_1^{mest} = (N_1, t_2)$ e $\mathcal{G}_1^{esc} = \{(N_2, t_5)\}$. O algoritmo resultará na criação do lugar comum $p_{1,1}$ e terá como saída o conjunto de redes modificadas $\mathcal{N}^* = \{N_1^*, N_2^*\}$. Para facilitar a identificação dos lugares comuns na rede global, renomearemos o lugar $p_{1,1}$ para p_{c1} . A rede global resultante da aplicação do algoritmo pode ser vista na figura 4.12.

Com o lugar comum p_{c1} criado, o módulo 2 inicia sua dinâmica normalmente com a ficha em p_4 e segue até que a ficha chegue em p_5 . Neste momento, o módulo 2 irá aguardar pelo disparo de t_5 , que só será possível caso exista uma ficha em p_{c1} , o que por sua vez não ocorrerá até que a transição t_2 do módulo 1 dispare. Desta forma, o módulo 2 não iniciará a sequência para pegar uma peça da esteira, representada por p_6 , enquanto o módulo 1 não disponibilizar tal peça, representado pelo disparo de t_2 . É importante notar, contudo, que a dinâmica do módulo 1 não sofre qualquer alteração com a implementação desta cooperação por habilitação.

Como a especificação (II) é muito semelhante à especificação (I), a mesma lógica foi usada para implementar entre os módulos 4 e 5 mais uma cooperação por

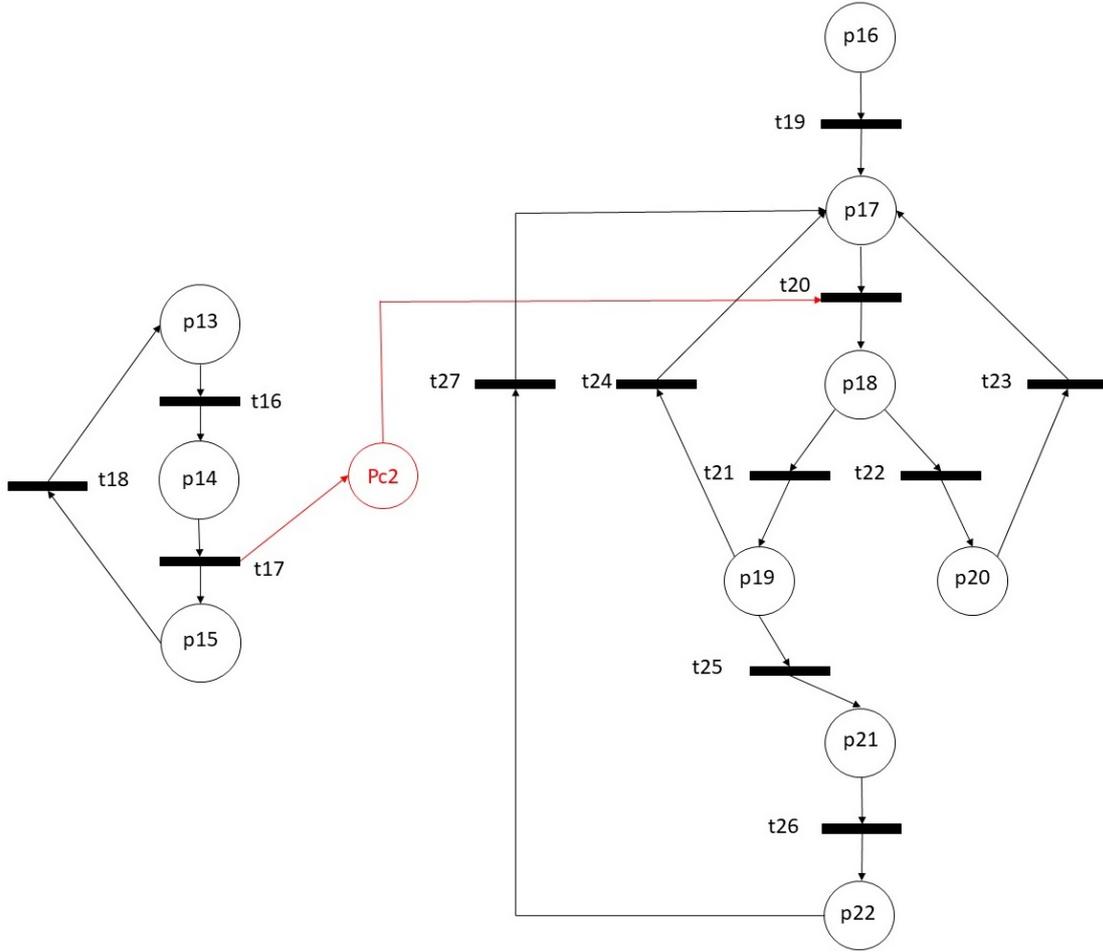


Figura 4.13: Representação da cooperação entre os módulos 4 e 5, respectivamente

habilitação, criando o grupo de cooperação G_2 . Neste grupo teremos o módulo 4 de mestre e o 5 de escravo, utilizando as transições t_{17} do módulo 4 e t_{20} do módulo 5, que são equivalentes às transições escolhidas para a especificação (I). A especificação que define a cooperação por habilitação para G_2 será: $\forall \pi \in \Pi(N_g, M_{0g}), \Upsilon(t_{20}, \pi) \leq \Upsilon(t_{17}, \pi)$.

Para obter o novo sistema em cooperação, executaremos o algoritmo 3.1 novamente, com entradas: número de módulos envolvidos na cooperação $k = 2$ (módulos 4 e 5), número de grupos de cooperação $g = 1$ (Grupo G_2), conjunto de redes de Petri dos módulos envolvidos $\mathcal{N} = \{N_4, N_5\}$ e $G_2 = (\mathcal{G}_2^{mest}, \mathcal{G}_2^{esc})$, em que $\mathcal{G}_2^{mest} = (N_4, t_{17})$ e $\mathcal{G}_2^{esc} = \{(N_5, t_{17})\}$. O algoritmo resultará na criação do lugar comum $p_{1,1}$, que será renomeado para p_{c2} , e terá como saída o conjunto de redes modificadas $\mathcal{N}^* = \{N_4^*, N_5^*\}$. A rede global resultante da aplicação do algoritmo pode ser vista na figura 4.13.

Assim como para a especificação (I), o lugar comum p_{c2} garantirá que o módulo 5 só possa iniciar sua sequência para pegar uma peça da esteira quando o módulo 4 disparar a transição t_{17} , colocando uma ficha em p_{c2} .

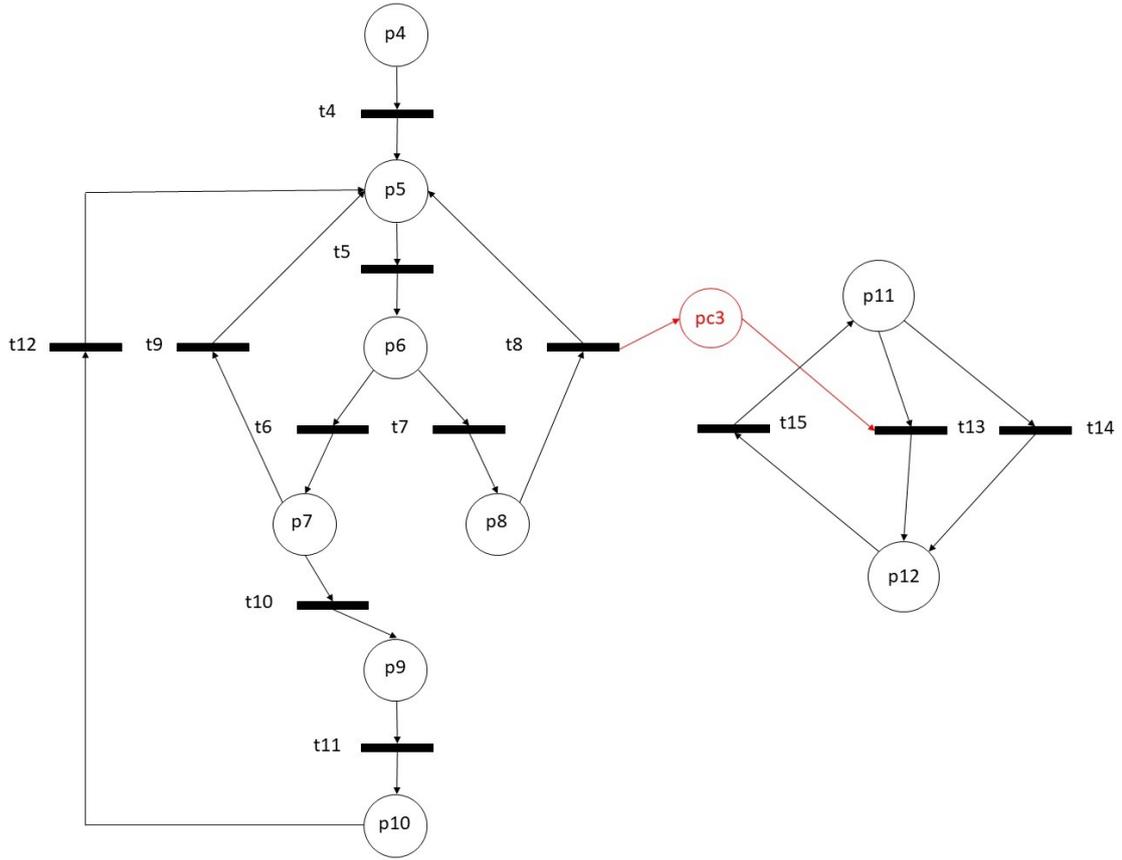


Figura 4.14: Representação da cooperação por habilitação entre os módulos 2 e 3, respectivamente

Seguindo para a especificação (III), optamos por usar mais uma cooperação por habilitação. Para garantir que o módulo 3 pegue uma peça apenas após ela ser disponibilizada pelo módulo 2, um grupo de cooperação G_3 foi criado, com o módulo 2 de mestre e o módulo 3 de escravo, envolvendo as transições t_8 do módulo 2 e t_{13} do módulo 3. A transição t_8 representa o momento em que uma peça foi descartada pelo módulo 2, enquanto a transição t_{13} representa o momento em que uma peça chega no módulo 3 vinda do módulo 2. Com isso, teremos a especificação que define a cooperação por habilitação definida como: $\forall \pi \in \Pi(N_g, M_{0g}), \Upsilon(t_{13}, \pi) \leq \Upsilon(t_8, \pi)$.

Mais uma vez, executamos o algoritmo 3.1 para obter o sistema em cooperação, com entradas: $k = 2$, $g = 1$, $\mathcal{N} = \{N_2^*, N_3\}$ e $G_3 = (\mathcal{G}_3^{mest}, \mathcal{G}_3^{esc})$, em que $\mathcal{G}_3^{mest} = (N_2^*, t_8)$ e $\mathcal{G}_3^{esc} = \{(N_3, t_{13})\}$. Note que, desta vez, a rede do módulo 2 N_2^* já foi previamente modificada, e já possui um lugar comum. O algoritmo resultará na criação do lugar comum $p_{1,1}$, que será renomeado para p_{c3} , e terá como saída o conjunto de redes modificadas $\mathcal{N}^* = \{N_2^{**}, N_3^*\}$. A rede global resultante da aplicação do algoritmo pode ser vista na figura 4.14.

Desta forma, t_{13} será impossibilitada de disparar enquanto não existir uma ficha em p_{c3} , o que só ocorrerá após o disparo de t_8 . Sendo assim, apenas quando uma

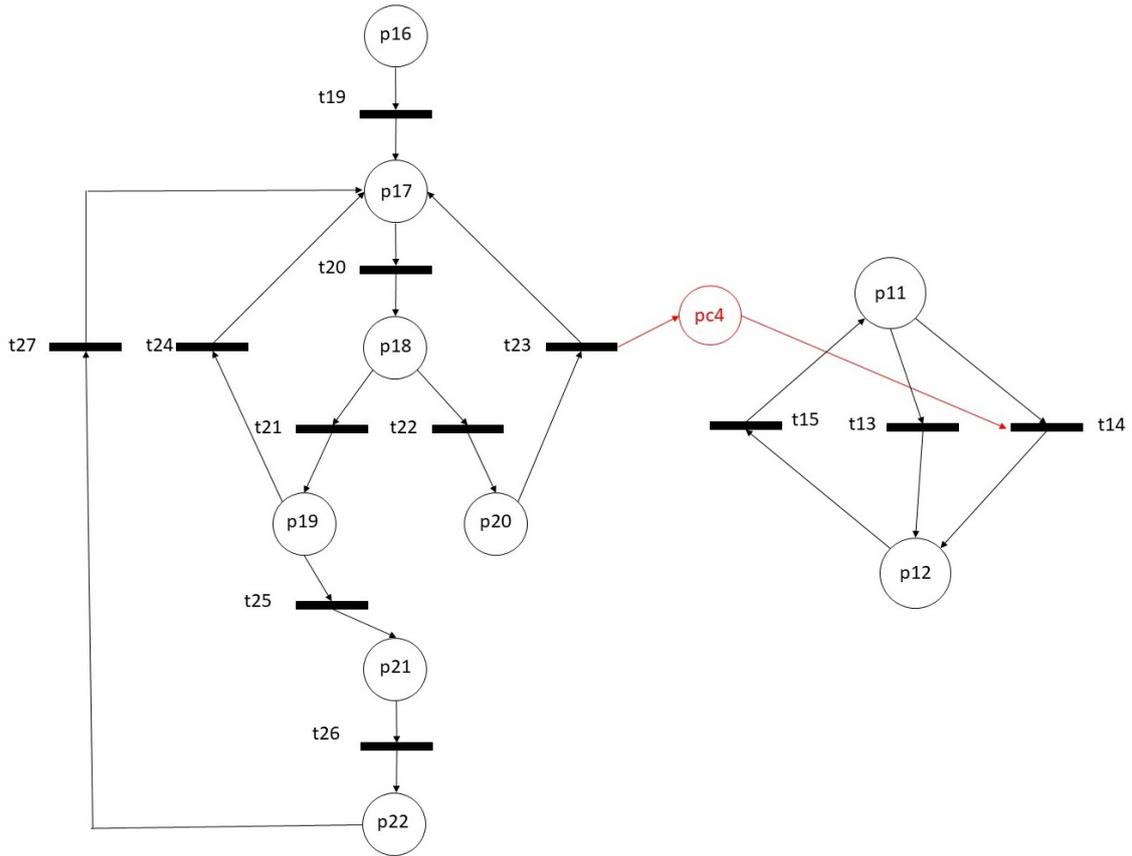


Figura 4.15: Representação da cooperação por habilitação entre os módulos 5 e 3, respectivamente

peça for disponibilizada pelo módulo 2 que t_{13} poderá disparar, realizando o descarte de tal peça. Finalmente, note que a existência de p_{c1} antes da aplicação do algoritmo não influenciou na aplicação do mesmo, ou em seu resultado.

Finalmente, criamos uma última cooperação por habilitação para a especificação (IV), visto que ela é muito semelhante à especificação (III). O grupo G_4 foi criado com o módulo 5 de mestre e o 3 de escravo, desta vez envolvendo as transições t_{23} do módulo 5 e t_{14} do módulo 3, com o objetivo de habilitar a passagem de peça do módulo 5 para o 3. Estas transições são equivalentes às transições escolhidas na especificação (III). A especificação que define a cooperação por habilitação será definida como: $\forall \pi \in \Pi(N_g, M_{0g}), \Upsilon(t_{14}, \pi) \leq \Upsilon(t_{23}, \pi)$.

Para obter o novo sistema em cooperação, aplicaremos o algoritmo 3.1 uma última vez com entradas: $k = 2$, $g = 1$, $\mathcal{N} = \{N_3^*, N_5^*\}$ e $G_4 = (\mathcal{G}_4^{mest}, \mathcal{G}_4^{esc})$, em que $\mathcal{G}_4^{mest} = (N_5^*, t_{23})$ e $\mathcal{G}_4^{esc} = \{(N_3^*, t_{14})\}$. Observe que desta vez, ambos os módulos envolvidos já foram previamente modificados. O algoritmo resultará na criação do lugar comum $p_{1,1}$, que será renomeado para p_{c4} , e terá como saída o conjunto de redes modificadas $\mathcal{N}^* = \{N_3^{**}, N_5^{**}\}$. A rede global resultante da aplicação do algoritmo pode ser vista na figura 4.15.

Assim como na especificação (III), o lugar comum p_{c4} garantirá que o módulo 3 não conseguirá iniciar sua sequência de descarte de peças do módulo 5 enquanto este não tiver disparado t_{23} , simbolizando que uma peça está pronta para o descarte.

Por fim, temos as especificações (V) e (VI), que tem como objetivo impedir que os módulos 2 e 5 tentem disponibilizar uma peça no módulo 3 simultaneamente. Para ambos os casos, optamos por implementar uma cooperação por impedimento, criando mais dois grupos G_5 e G_6 .

Começando pela especificação (V), criamos o grupo G_5 composto pelo módulo 2 de mestre, usando as transições t_7 e t_8 , e pelo módulo 5 de escravo, usando a transição t_{22} . A transição t_7 representa o início da sequência de descarte do módulo 2, enquanto a transição t_8 representa o final desta mesma sequência. A transição t_{22} , por outro lado, representa o início da sequência de descarte do módulo 5, que não pode ocorrer enquanto o módulo 2 estiver em sua sequência de descarte. Desta vez, as equações que definirão a cooperação por impedimento de G_5 serão definidas como: $\forall \pi \in \Pi(N_g, M_{0g}), \text{Sufix}(\pi) \neq t_{22} \vee \Upsilon(t_7, \pi) = \Upsilon(t_8, \pi)$ e $\forall \pi \in \Pi(N_g, M_{0g}), \Upsilon(t_8, \pi) \leq \Upsilon(t_7, \pi)$, como nas equações 3.5 e 3.6.

Para obter o sistema em cooperação, executamos o algoritmo 3.2, com entradas: número de módulos envolvidos na cooperação $k = 2$ (módulos 2 e 5), número de grupos de cooperação $g = 1$ (Grupo G_5), conjunto de redes de Petri dos módulos envolvidos $\mathcal{N} = \{N_2^{**}, N_5^{**}\}$ e $G_5 = (\mathcal{G}_5^{mest}, \mathcal{G}_5^{esc})$, em que $\mathcal{G}_5^{mest} = (N_2^{**}, t_7, t_8)$ e $\mathcal{G}_5^{esc} = \{(N_5^{**}, t_{22})\}$. O lugar comum $p_{1,1}$ será criado com a execução do algoritmo, que será então renomeado para p_{c5} . Teremos como saída o conjunto de redes modificadas $\mathcal{N}^* = \{N_2^{***}, N_5^{***}\}$. A rede global resultante da aplicação do algoritmo pode ser vista na figura 4.16.

Enquanto o lugar comum p_{c5} estiver sem fichas, o módulo 5 não perceberá qualquer alteração em sua dinâmica. A partir do momento que este lugar receber uma ficha, contudo, o módulo 5 será impossibilitado de disparar t_{22} , o que significa que ele não conseguirá levar uma peça para o módulo 3 (representado pelo lugar p_{20}). Note que o lugar comum receberá uma ficha quando a transição t_7 do módulo 2 disparar, ou seja, quando o módulo 2 começar sua sequência para levar uma peça até o módulo 3. Apenas quando essa sequência terminar, simbolizado pelo disparo de t_8 , que a ficha será removida de p_{c5} e t_{22} poderá disparar novamente, permitindo que o módulo 5 leve uma peça para o módulo 3.

Terminando com a especificação (VI), uma última cooperação por impedimento foi criada, representada pelo grupo G_6 . Este grupo terá como mestre o módulo 5, usando as transições t_{22} e t_{23} , e como escravo o módulo 2, com a transição t_7 . Da mesma forma que na especificação (V), t_{22} e t_{23} representam o início e fim da sequência de descarte do módulo 5, enquanto t_7 representa o início da sequência de descarte do módulo 2. Seguindo as equações 3.5 e 3.6, teremos as

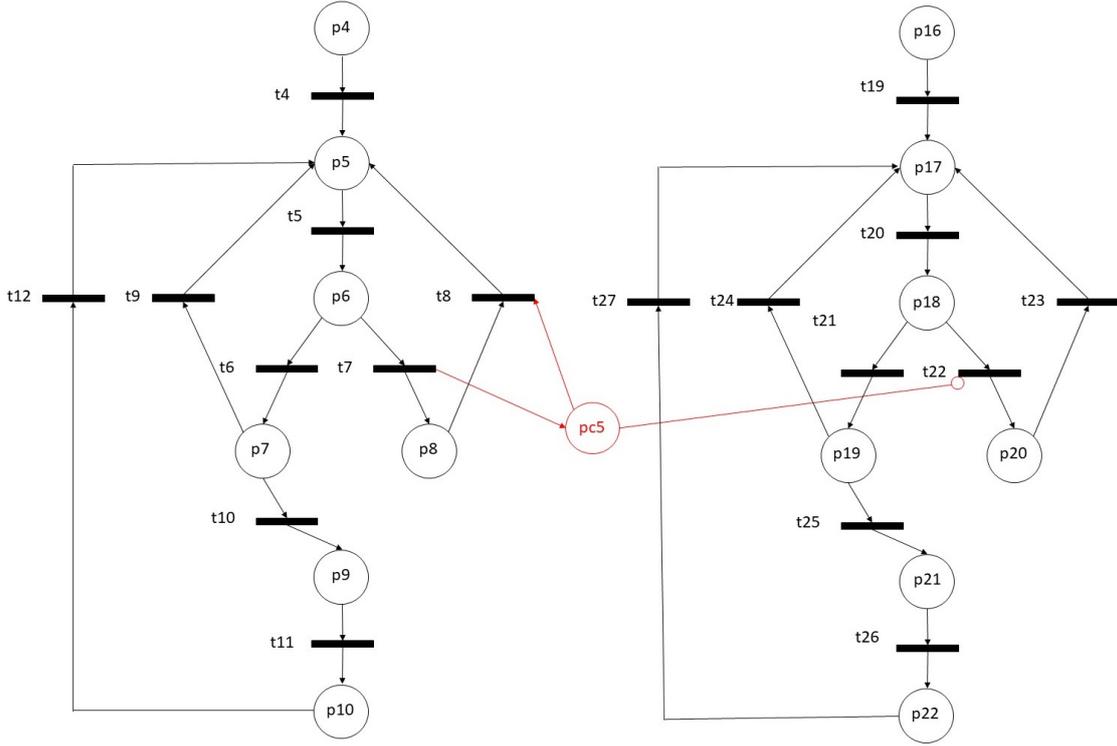


Figura 4.16: Representação da cooperação por impedimento entre os módulos 2 e 5, respectivamente

especificações que definem esta cooperação por impedimento definidas como: $\forall \pi \in \Pi(N_g, M_{0g}), Suffix(\pi) \neq t_7 \vee \Upsilon(t_{22}, \pi) = \Upsilon(t_{23}, \pi)$ e $\forall \pi \in \Pi(N_g, M_{0g}), \Upsilon(t_{23}, \pi) \leq \Upsilon(t_{22}, \pi)$.

Executamos o algoritmo 3.2 para obter o sistema em cooperação, com entradas: número de módulos envolvidos na cooperação $k = 2$ (módulos 4 e 5), número de grupos de cooperação $g = 1$ (Grupo G_6), conjunto de redes de Petri dos módulos envolvidos $\mathcal{N} = \{N_2^{***}, N_5^{***}\}$ e $G_6 = (\mathcal{G}_6^{mest}, \mathcal{G}_6^{esc})$, em que $\mathcal{G}_6^{mest} = (N_5^{***}, t_{22}, t_{23})$ e $\mathcal{G}_6^{esc} = \{(N_2^{***}, t_7)\}$. O lugar comum $p_{1,1}$ será criado com a execução do algoritmo, e renomeado para p_{c6} . A saída do algoritmo será o conjunto de redes modificadas $\mathcal{N}^* = \{N_2^{****}, N_5^{****}\}$. A rede global resultante da aplicação do algoritmo pode ser vista na figura 4.17.

A implementação das especificações (V) e (VI) simultaneamente fará com que exista na planta uma prioridade por ordem de chegada. Caso ambos os módulos 2 e 5 tentem levar uma peça para o módulo 3, o que tiver tentado primeiro impedirá o outro de seguir com sua dinâmica até que o descarte seja concluído. É importante ressaltar que o disparo de t_7 e t_{22} no mesmo instante resultaria em uma quebra de especificação, visto que os lugares p_{c5} e p_{c6} receberiam uma ficha simultaneamente e não cumpririam sua tarefa de impedir o disparo das transições t_{22} e t_7 , respectivamente. Note, contudo, que este cenário de disparo simultâneo não pode ocorrer visto

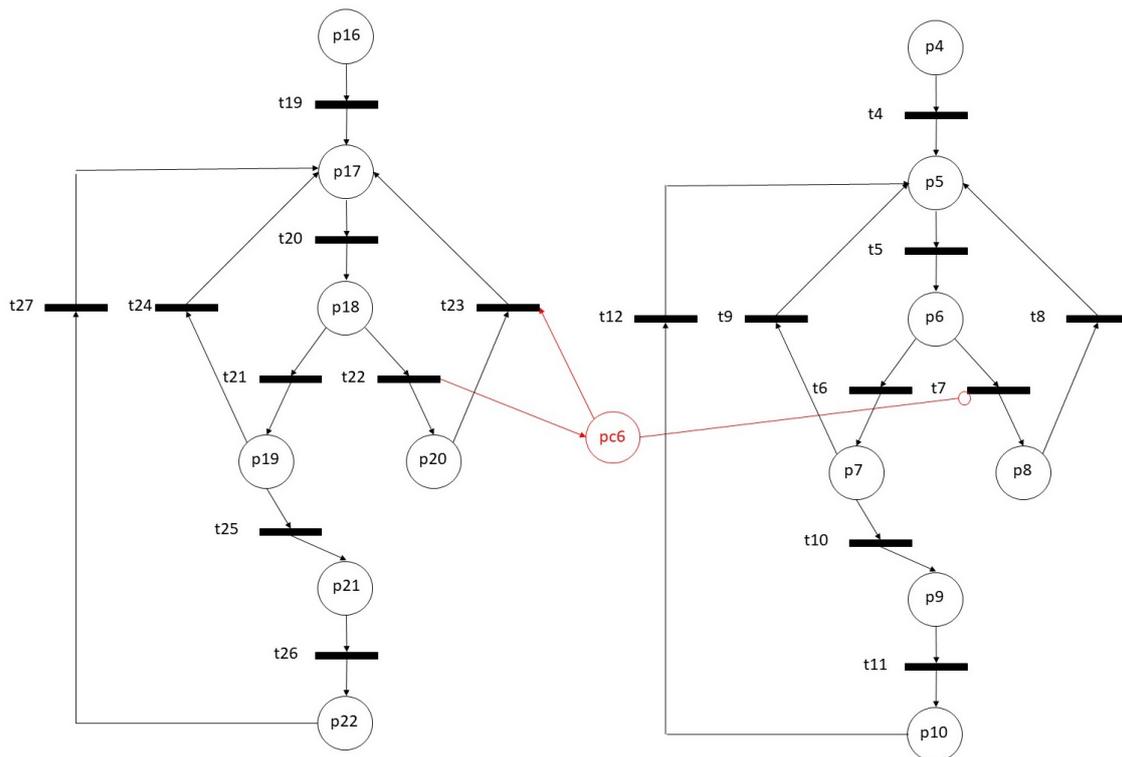


Figura 4.17: Representação da cooperação por impedimento entre os módulos 5 e 2, respectivamente

que estamos considerando sequências de disparo de transições, que por definição não podem conter transições disparando no mesmo instante.

O sistema em cooperação completo pode ser visto na figura 4.18. Para implementar na planta as cooperações criadas, um canal de comunicação foi estabelecido para cada cooperação, conectando o mestre daquela cooperação ao escravo. Para as cooperações por habilitação, um simples canal de comunicação unidirecional, levando informação do mestre para o escravo, é suficiente, visto que o módulo mestre não sofre qualquer alteração com a inclusão da cooperação. Para a cooperação por impedimento, contudo, é necessário além do canal unidirecional a inclusão de um contador, visto que o módulo mestre precisa da informação de quantas fichas existem no lugar comum para poder disparar suas transições.

Note que os algoritmos 3.1 e 3.2 aceitam mais de um grupo de cooperação simultaneamente em suas entradas. Sendo assim, ao invés de aplicar o algoritmo 3.1 quatro vezes, cada vez com um grupo de cooperação como entrada, poderíamos ter aplicado ele apenas uma vez com todos os quatro grupos de cooperação como entrada, obtendo o mesmo resultado. Da mesma forma, ao invés de aplicar o algoritmo 3.2 duas vezes, poderíamos ter aplicado apenas uma vez, com os dois grupos de cooperação como entrada. Neste trabalho, optamos por limitar a aplicação dos algoritmos a apenas um grupo na entrada para facilitar o acompanhamento da criação

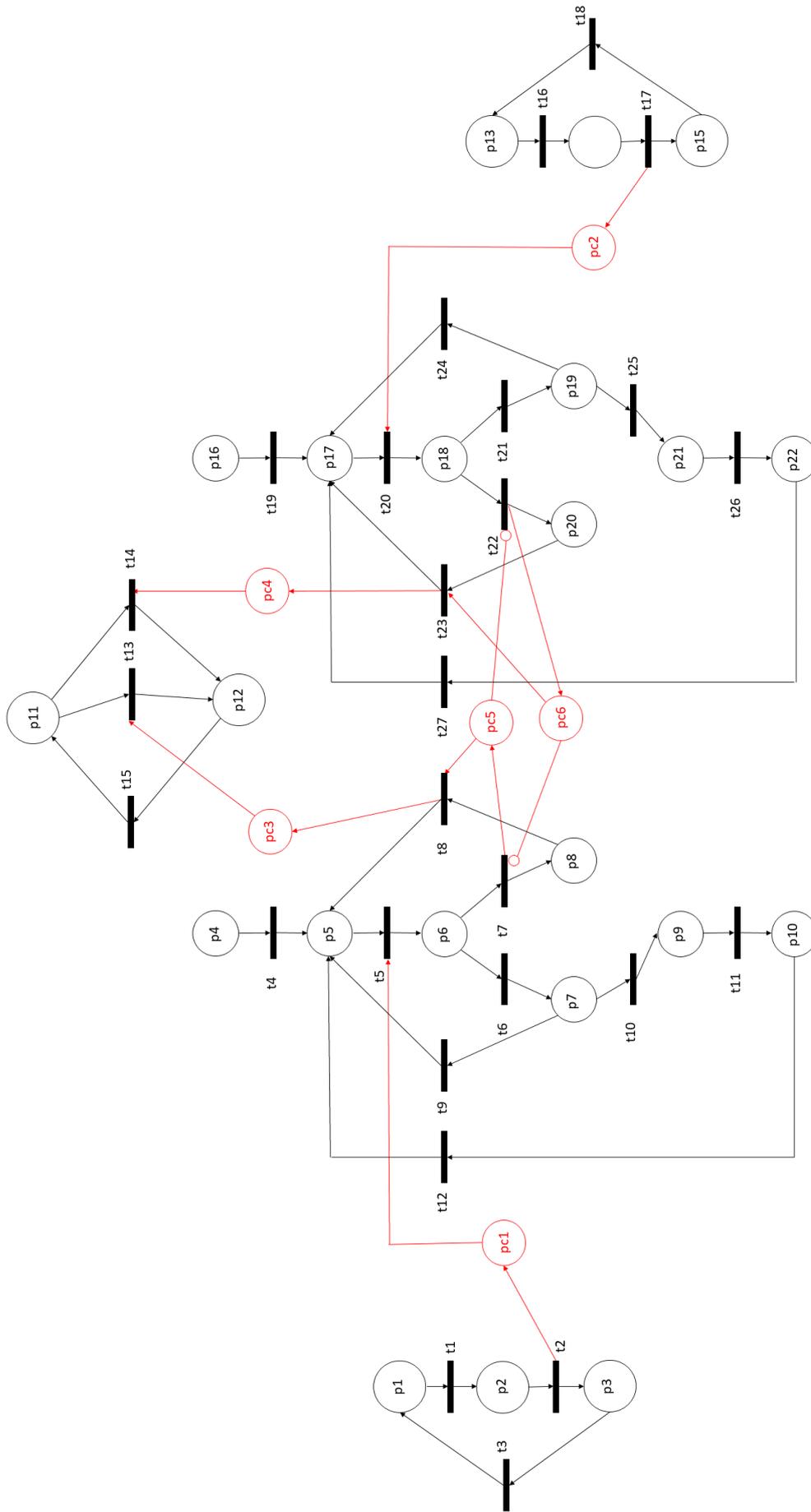


Figura 4.18: Sistema em cooperação completo

dos lugares comuns.

Com as cooperações implementadas, podemos analisar a rede global para identificar se algum bloqueio foi criado. De acordo com o teorema 3.3, o conjunto de sífões mínimos dos módulos 1 e 2 não é alterado com a implementação da cooperação por habilitação do grupo 1. De forma semelhante, os conjuntos de sífões mínimos dos módulos 1, 2, 3, 4 e 5 não é alterado com a implementação das cooperações por habilitação dos grupos 2, 3 e 4. Finalmente, de acordo com o teorema 3.4, o sistema global será livre de bloqueio, visto que ao realizar uma busca por sífões não-controláveis, não encontramos nenhum contendo algum lugar comum.

Por fim, note que as alterações nas redes originais são simples, consistindo apenas na criação de alguns lugares comuns e alguns arcos, sem alterar lugares, transições ou arcos previamente existentes na rede. Por este motivo, a implementação desta metodologia em um sistema real é rápida, de forma que cada lugar comum pode ser representado por um canal de comunicação interligando o módulo mestre ao escravo.

Capítulo 5

Conclusão e trabalhos futuros

Neste trabalho foi abordado o problema de cooperação de sistemas a eventos discretos conectados em rede. Neste problema, temos inicialmente um conjunto de módulos que operam sozinhos respeitando suas especificações locais. Queremos então fazer com que estes módulos trabalhem em conjunto respeitando uma determinada especificação global, sem deixar de desempenhar suas funções locais.

Foram definidos dois tipos de cooperação diferentes, um por habilitação e um por impedimento, assim como os algoritmos para a implementação destas cooperações em redes de Petri. Foi feita também uma análise em cima dos casos de bloqueio no sistema global resultante, concluindo que caso não existam sifões não-controláveis na rede final contendo algum lugar comum, o sistema será livre de bloqueios. Finalmente, foi apresentada uma implementação prática das cooperações apresentadas usando uma planta mecatrônica de montagem de cubos.

A principal vantagem das metodologias aqui desenvolvidas quando comparadas com outras metodologias na literatura está na flexibilidade e simplicidade, resultando em implementações rápidas sem um grande crescimento do sistema global. Como trabalhos futuros, planejamos estender a abordagem desenvolvida por [26] para lidar com a existência de atrasos de comunicação e perda de pacotes nas redes envolvidas. Além disso, pretendemos também expandir a análise das redes de Petri após a obtenção do sistema em cooperação, verificando propriedades como conservabilidade, estabilidade e se as redes continuam seguras.

Referências Bibliográficas

- [1] GILCHRIST, A. *Industry 4.0: the industrial internet of things*. Springer, 2016.
- [2] GURJANOV, A., ZAKOLDAEV, D., SHUKALOV, A., et al. “The description of the Industry 4.0 cyber and physical production division based on Petri net models”, v. 1050, n. 1, pp. 012032, 2018.
- [3] LATORRE-BIEL, J.-I., FAULÍN, J., JUAN, A. A., et al. “Petri net model of a smart factory in the frame of industry 4.0”, *IFAC-PapersOnLine*, v. 51, n. 2, pp. 266–271, 2018.
- [4] LIU, G., ZHOU, M., JIANG, C. “Petri net models and collaborativeness for parallel processes with resource sharing and message passing”, *ACM Transactions on Embedded Computing Systems (TECS)*, v. 16, n. 4, pp. 1–20, 2017.
- [5] ZGORZELSKI, M., LUNZE, J. “A method for the synchronisation of networked discrete-event systems”. In: *2016 13th International Workshop on Discrete Event Systems (WODES)*, pp. 444–451, Xi’an, China, 2016. IEEE.
- [6] MURATA, T. “Petri nets: Properties, analysis and applications”, *Proceedings of the IEEE*, v. 77, n. 4, pp. 541–580, 1989.
- [7] NAKAMOTO, F. Y., MIYAGI, P. E., SANTOS FILHO, D. J. D. “Geração automática da solução de controle para alocação de recursos utilizando redes de Petri”. 2009.
- [8] GARCIA MELO, J. I., JUNQUEIRA, F., EIGI MIYAGI, P. “Towards modular and coordinated manufacturing systems oriented to services”, *Dyna*, v. 77, n. 163, pp. 201–210, 2010.
- [9] HASEGAWA, K., MIYAGI, P. E., SANTOS FILHO, D. J., et al. “On resource arc for Petri net modelling of complex resource sharing system”, *Journal of Intelligent and Robotic Systems*, v. 26, n. 3, pp. 423–437, 1999.
- [10] GIUA, A. *Petri nets as discrete event models for supervisory control*. Tese de Doutorado, Rensselaer Polytechnic Institute, Troy, New York, 1992.

- [11] RUDIE, K., WONHAM, W. M. “Think globally, act locally: Decentralized supervisory control”. In: *1991 American Control Conference*, pp. 898–903. IEEE, 1991.
- [12] BARRETT, G., LAFORTUNE, S. “Decentralized supervisory control with communicating controllers”, *IEEE Transactions on Automatic Control*, v. 45, n. 9, pp. 1620–1638, 2000.
- [13] DE QUEIROZ, M. H., CURY, J. E. “Modular supervisory control of large scale discrete event systems”. In: *Discrete Event Systems*, Springer, pp. 103–110, 2000.
- [14] GONZALEZ, A. G., ALVES, M. V., VIANA, G. S., et al. “Supervisory control-based navigation architecture: a new framework for autonomous robots in industry 4.0 environments”, *IEEE Transactions on Industrial Informatics*, v. 14, n. 4, pp. 1732–1743, 2017.
- [15] CHU, F., XIE, X.-L. “Deadlock analysis of Petri nets using siphons and mathematical programming”, *IEEE Transactions on Robotics and Automation*, v. 13, n. 6, pp. 793–804, 1997.
- [16] HUANG, Y., JENG, M., XIE, X., et al. “Deadlock prevention policy based on Petri nets and siphons”, *International Journal of Production Research*, v. 39, n. 2, pp. 283–305, 2001.
- [17] VILLANI, E., MIYAGI, P. E., VALETTE, R. “Petri nets and Object-Oriented Approach for the Analysis of Hybrid Systems”, *XIV C. Brasileiro Automatica*, 2002.
- [18] DAVID, R., ALLA, H. *Discrete, continuous, and hybrid Petri nets*, v. 1. Springer, 2010.
- [19] GIUA, A., SILVA, M. “Modeling, analysis and control of discrete event systems: a Petri net perspective”, *IFAC-PapersOnLine*, v. 50, n. 1, pp. 1772–1783, 2017.
- [20] LI, Z., ZHOU, M. *Deadlock resolution in automated manufacturing systems: a novel Petri net approach*. Springer Science & Business Media, 2009.
- [21] GENC, S., LAFORTUNE, S. “Distributed diagnosis of place-bordered Petri nets”, *IEEE Transactions on Automation science and Engineering*, v. 4, n. 2, pp. 206–219, 2007.
- [22] IORDACHE, M., ANTSAKLIS, P. J. *Supervisory control of concurrent systems: a Petri net structural approach*. Springer Science & Business Media, 2007.

- [23] LIN, F., WONHAM, W. M. “Decentralized control and coordination of discrete-event systems with partial observation”, *IEEE Transactions on automatic control*, v. 35, n. 12, pp. 1330–1337, 1990.
- [24] LEDUC, R. J. *Hierarchical interface-based supervisory control*. PhD thesis, Department of Electrical and Computer Engineering, University of . . . , 2002.
- [25] LEDUC, R. J., LAWFORD, M., WONHAM, W. M. “Hierarchical interface-based supervisory control-part II: parallel case”, *IEEE Transactions on Automatic Control*, v. 50, n. 9, pp. 1336–1348, 2005.
- [26] VIANA, G., ALVES, M. V. S., BASILIO, J. C. “Codiagnosability of networked discrete event systems with timing structure”, *IEEE Transactions on Automatic Control*, 2021.