
**Uma Metodologia baseada na Lógica Linear
para Análise de Processos de *Workflow*
Interorganizacionais**

Lígia Maria Soares Passos



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uberlândia
2016

Lígia Maria Soares Passos

**Uma Metodologia baseada na Lógica Linear
para Análise de Processos de *Workflow*
Interorganizacionais**

Tese de doutorado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Engenharia de Software

Orientador: Prof. Dr. Stéphane Julia

Uberlândia

2016

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

P289m Passos, Lígia Maria Soares, 1984-
2016 Uma metodologia baseada na Lógica Linear para análise de
processos de workflow interorganizacionais / Lígia Maria Soares Passos.
- 2016.
182 f. : il.

Orientador: Stéphane Julia.
Tese (doutorado) - Universidade Federal de Uberlândia, Programa
de Pós-Graduação em Ciência da Computação.
Inclui bibliografia.

1. Computação - Teses. 2. Redes de Petri - Teses. 3. Lógica - Teses.
4. Fluxo de trabalho - Teses. I. Julia, Stéphane. II. Universidade Federal
de Uberlândia. Programa de Pós-Graduação em Ciência da Computação.
III. Título.

CDU: 681.3

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Computação a aceitação da Tese intitulada "**Uma Metodologia baseada na Lógica Linear para Verificação de Processos de *Workflow* Interorganizacionais**" defendida por **Lígia Maria Soares Passos** como parte dos requisitos exigidos para a obtenção do título de **Doutor em Ciência da Computação**.

Uberlândia/MG, 22 de fevereiro de 2016.

Orientador: _____
Prof. Dr. Stéphane Julia
Universidade Federal de Uberlândia

Banca Examinadora:

Prof. Dr. Pedro Frosi Rosa
Universidade Federal de Uberlândia

Prof. Dr. Carlos Roberto Lopes
Universidade Federal de Uberlândia

Prof. Dr. Paulo Eigi Miyagi
Universidade de São Paulo

Prof. Dra. Emilia Villani
Instituto Tecnológico de Aeronáutica

*Dedico esta pesquisa ao meu esposo,
aos meus pais,
aos meus irmãos
e à pequena Ana Clara.*

Agradecimentos

Agradeço primeiramente a Deus, por me conceder mais esta oportunidade de vida e por permitir o reencontro com pessoas especiais, que sempre deixam a minha caminhada mais agradável.

Agradeço aos meus pais, Milton e Cidinha, por me darem a vida e por me ensinarem valores essenciais como a honestidade, a humildade, a caridade e o amor. Obrigada por me mostrarem o verdadeiro sentido da vida e por serem o porto seguro para onde sempre volto.

Agradeço às minhas irmãs Aline e Giselle e ao meu irmão Leonardo pela amizade e carinho contínuos. Apesar da distância física que atualmente nos separa no cotidiano, estaremos sempre unidos pelo elo invisível que é o amor fraterno.

Agradeço à minha sobrinha Ana Clara, por todos os momentos de felicidade e alegria proporcionados, apesar da sua tão recente chegada.

Agradeço ao meu esposo, Luís Fernando, por seu amor e companheirismo, sua compreensão e paciência. Obrigada por sempre me motivar a seguir em frente, apesar das dificuldades e obstáculos encontrados pelos diversos caminhos que já percorremos juntos.

Agradeço aos meus familiares por todo apoio e torcida. Especialmente agradeço à minha querida avó Tianinha, por sua extrema bondade, por seu exemplo de força, fé e humildade.

Agradeço às minhas amigas Raquel Lafetá, Camila Molina e Sumaya Gonçalves pela contínua, sincera e verdadeira amizade.

Agradeço ao meu orientador, Prof. Dr. Stéphane Julia, pelos valiosos ensinamentos e pelas oportunidades de estudo, fundamentais para a minha formação acadêmica.

Agradeço a todos os professores da Faculdade de Computação da UFU que diretamente contribuíram para meu amadurecimento acadêmico, em especial à Profa. Dra. Sandra Aparecida de Amo e à Profa. Dra. Marcia Aparecida Fernandes.

Por fim, agradeço aos amigos Erisvaldo Fialho e Mariana Prudente por todo o auxílio prestado durante estes longos anos em que fui aluna do Programa de Pós-Graduação em Ciência da Computação da FACOM/UFU.

“Não há tanto mérito em que domines essa ou aquela ciência, e sim em que lhe utilizes os recursos, a fim de ajudar os companheiros da Humanidade a se desvencilharem da insipiência e da ignorância.”
(Emmanuel)

Resumo

Este trabalho formaliza quatro métodos baseados na Lógica Linear para verificação de processos de *workflow* interorganizacionais modelados por Workflow nets interorganizacionais, que são redes de Petri que modelam tais processos. O primeiro método está relacionado com a verificação do critério de correção *Soundness* para processos de *workflow* interorganizacionais. O método é baseado na construção e análise de árvores de prova da Lógica Linear que representam tanto os processos locais quanto o processo global. O segundo e terceiro métodos estão relacionados, respectivamente, com a verificação dos critérios de correção *Relaxed Soundness* e *Weak Soundness* para processos de *workflow* interorganizacionais, e são obtidos através da análise de árvores de prova da Lógica Linear reutilizadas, construídas para a prova do critério de correção *Soundness*. Já o quarto método tem por objetivo a detecção dos cenários livres de *deadlock* em processos de *workflow* interorganizacionais e é baseado na construção e análise de árvores de prova da Lógica Linear que consideram, inicialmente, os processos locais e as comunicações entre estes e, posteriormente, os cenários candidatos.

Um estudo de caso é realizado no contexto da verificação de composições de serviços Web, uma vez que há uma relação estreita entre a modelagem de um processo de *workflow* interorganizacional e uma composição de serviços Web. Assim, os quatro métodos propostos no contexto dos processos de *workflow* interorganizacionais são aplicados a uma composição de serviços Web. A avaliação dos resultados mostra que o reuso de árvores de prova da Lógica Linear construídas inicialmente para a prova do critério de correção *Soundness* de fato ocorre no contexto da verificação dos critérios de correção *Relaxed Soundness* e *Weak Soundness*. Além disso, a avaliação mostra como os sequentes da Lógica Linear e suas árvores de prova explicitam as possibilidades de colaboração existentes em uma composição de serviços Web. Uma avaliação que leva em conta o número de árvores de prova da Lógica Linear construídas mostra que este número pode ser significativamente reduzido no método para detecção de cenários livres de *deadlock*.

Uma abordagem para planejamento de recursos, baseada no cálculo de datas simbólicas, que considera dados extraídos de árvores de prova da Lógica Linear, é apresentada

e validada através de simulações realizadas no simulador CPN Tools. Duas abordagens para a monitoração dos cenários livres de *deadlock* são introduzidas e mostram como dados obtidos nas árvores de prova da Lógica Linear podem ser utilizados para guiar a execução de tais cenários.

Palavras-chave: Rede de Petri, Lógica Linear, *Workflow*, WorkFlow net Interorganizacional, Verificação, Monitoração, *Soundness*, *Relaxed Soundness*, *Weak Soundness*, Bloqueio Mortal, Serviço Web.

Abstract

This work formalizes four methods based on Linear Logic for the verification of inter-organizational workflow processes modelled by Interorganizational Workflow nets, which are Petri nets that model such processes. The first method is related to the verification of the Soundness criteria for interorganizational workflow processes. The method is based on the construction and analysis of Linear Logic proof trees, which represent the local processes as much as they do the global processes. The second and third methods are related, respectively to Soundness criteria verification, Relaxed Soundness and Weak Soundness for the interorganizational workflow processes. These are obtained through the analysis of reutilized Linear Logic proof trees that have been constructed for the verification of the Soundness criteria. However, the fourth method has the objective of detecting the deadlock free scenarios in interorganizational workflow and is based on the construction and analysis of Linear Logic proof trees, which initially takes into consideration the local processes and communication between such, and thereafter the candidate scenarios. A case study is carried out in the context of a Web services composition check, since there is a close correlation between the modelling of the interorganizational workflow process and a Web services composition. Therefore, the four methods proposed in the interorganizational workflow process context, are applied to a Web services composition. The evaluation of the obtained results shows that the reutilization of Linear Logic proof trees initially constructed for verifying the Soundness criteria, in fact occurs in the context of verifying the Relaxed Soundness and Weak Soundness criteria. In addition, the evaluation shows how the Linear Logic sequents and their proof trees explicitly show the possibilities for existing collaborations in a Web service composition. An evaluation that takes into account the number of constructed linear logic proof trees shows that this number can be significantly reduced in the deadlock-freeness scenarios detection method. An approach for resource planning based on the symbolic date calculation, which considers data extracted from Linear Logic proof trees is presented and validated through simulations performed on the CPN tools simulator. Two approaches for the monitoring of deadlock-freeness scenarios are introduced and show how data obtained from the Linear Logic proof

trees can be used to guide the execution of such scenarios.

Keywords: Petri net, Linear Logic, Workflow, Interorganizational WorkFlow net, Verification, Monitoring, Soundness, Relaxed Soundness, Weak Soundness, Deadlock, Web Service.

Lista de ilustrações

Figura 1 – Exemplos de sensibilização e disparo de transição em uma rede de Petri.	36
Figura 2 – Modelos em rede de Petri colorida (MURATA, 1989).	39
Figura 3 – Um exemplo de rede de Petri t-temporal.	40
Figura 4 – WorkFlow net para o processo de tratamento de reclamações (AALST; HEE, 2004).	41
Figura 5 – Substituição de um roteiro iterativo por uma tarefa global (PASSOS; JULIA, 2009).	42
Figura 6 – Um exemplo de uma WorkFlow net Interorganizacional (AALST, 1998b).	47
Figura 7 – A $U(\text{IOWF-net})$ para a IOWF-net mostrada na Figura 6.	48
Figura 8 – Redes de Petri para a exemplificação da tradução de redes de Petri em fórmulas da Lógica Linear.	50
Figura 9 – Rede de Petri para exemplificação da construção de uma árvore de prova canônica da Lógica Linear.	51
Figura 10 – Grafo de precedência para a árvore de prova rotulada para o sequente $P1, t_1, t_2, t_3, t_4, t_5 \vdash P7$.	54
Figura 11 – Cenários para a Local WorkFlow net (LWF-net) PC .	60
Figura 12 – Situação de <i>deadlock</i> para o processo de <i>workflow</i> interorganizacional modelado pela Interorganizacional WorkFlow net (IOWF-net) da Figura 6.	69
Figura 13 – Método para detecção de cenários livres de <i>deadlock</i> em processos de <i>workflow</i> interorganizacionais.	78
Figura 14 – IOWF-net com lugares de comunicação Input Communication Place (ICP)s e Output Communication Place (OCP)s em destaque.	81
Figura 15 – Cenários Sc_1A e Sc_3PC , que compõem o cenário candidato $Sc_1A \times Sc_3PC$.	84
Figura 16 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_1A \times Sc_3PC$.	86

Figura 17 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_2A \times Sc_2PC$	87
Figura 18 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_3A \times Sc_1PC$	87
Figura 19 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_4A \times Sc_1PC$	88
Figura 20 – Serviços Web para compra virtual de bilhetes. (Adaptado de Xiong, Fan e Zhou (2010)).	91
Figura 21 – Módulos de <i>workflow</i> <i>Customer</i> (C1) e <i>Ticket</i> (T1).	91
Figura 22 – O Sistema Composto $C1 \oplus T1$	92
Figura 23 – Grafo de precedência para o serviço $Sc_1C1 \times Sc_2T1$	104
Figura 24 – Grafo de precedência para o serviço $Sc_2C1 \times Sc_1T1$	104
Figura 25 – O módulo $\overline{C1}$ implementado utilizando o simulador CPN Tools e o respectivo grafo das marcações acessíveis.	107
Figura 26 – O módulo $\overline{T1}$ implementado utilizando o simulador CPN Tools e o respectivo grafo das marcações acessíveis.	107
Figura 27 – O sistema composto $\overline{C1} \oplus \overline{T1}$ implementado utilizando o simulador CPN Tools.	108
Figura 28 – Grafo das marcações acessíveis para o sistema composto $\overline{C1} \oplus \overline{T1}$, gerado pelo simulador CPN Tools.	108
Figura 29 – Marcação referente ao nó 7 do grafo das marcações acessíveis apresentado na Figura 28.	109
Figura 30 – Marcação referente ao nó 7 do grafo das marcações acessíveis apresentado na Figura 28.	109
Figura 31 – Unfolded Interorganizational WorkFlow net ($U(IOWF-net)$) da Figura 7 com tempo associado às transições.	114
Figura 32 – Implementação no CPN Tools do modelo apresentado na Figura 31.	116
Figura 33 – Relatório de replicações de simulação que considera a monitoração do lugar o	116
Figura 34 – Implementação do cenário $Sc_1A \times Sc_3PC$ no CPN Tools.	117
Figura 35 – Relatório de replicações de simulação que considera a monitoração do lugar o no cenário $Sc_1A \times Sc_3PC$	117
Figura 36 – Implementação do cenário $Sc_2A \times Sc_2PC$ no CPN Tools.	118
Figura 37 – Relatório de replicações de simulação que considera a monitoração do lugar o no cenário $Sc_2A \times Sc_2PC$	118
Figura 38 – Implementação do cenário $Sc_3A \times Sc_1PC$ no CPN Tools.	119
Figura 39 – Relatório de replicações de simulação que considera a monitoração do lugar o no cenário $Sc_3A \times Sc_1PC$	119
Figura 40 – Implementação do cenário $Sc_4A \times Sc_1PC$ no CPN Tools.	120

Figura 41 – Relatório de replicações de simulação que considera a monitoração do lugar o no cenário $Sc_4A \times Sc_1PC$	120
Figura 42 – Implementação cenário $Sc_1A \times Sc_3PC$ com transições auxiliares no CPN Tools.	124
Figura 43 – Implementação cenário $Sc_2A \times Sc_2PC$ com transições auxiliares no CPN Tools.	125
Figura 44 – Implementação cenário $Sc_3A \times Sc_1PC$ com transições auxiliares no CPN Tools.	126
Figura 45 – Implementação cenário $Sc_4A \times Sc_1PC$ com transições auxiliares no CPN Tools.	126
Figura 46 – Grafo de precedência para o cenário $Sc_1C1 \times Sc_2P1$	128
Figura 47 – Grafo de precedência para o cenário $Sc_1C1 \times Sc_2P1$	128
Figura 48 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_1A \times Sc_3PC$	129
Figura 49 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_3A \times Sc_1PC$	129
Figura 50 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_4A \times Sc_1PC$	129
Figura 51 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_2A \times Sc_2PC$	130
Figura 52 – Sistema composto $C1 \oplus T1$ com guardas associadas às transições. . . .	131
Figura 53 – Sistema composto $\overline{C1} \oplus \overline{T1}$ com guardas associadas às transições, implementado no CPN Tools.	132
Figura 54 – Grafo das marcações acessíveis para o sistema composto $\overline{C1} \oplus \overline{T1}$ com guardas associadas às transições, gerado pelo CPN Tools.	133
Figura 55 – $U(\text{IOWF-net})$ da Figura 7 com guardas associadas às transições, implementada no CPN Tools.	134
Figura 56 – Situação de <i>deadlock</i> para a $U(\text{IOWF-net})$ da Figura 55.	134

Lista de tabelas

Tabela 1 – Comparação entre os diferentes tipos de <i>Soundness</i>	44
Tabela 2 – Datas simbólicas de produção e consumo dos átomos da rede de Petri t-temporal da Figura 3.	55
Tabela 3 – Intervalos de datas simbólicas de produção e consumo dos átomos da rede de Petri t-temporal da Figura 3.	55
Tabela 4 – Intervalos de datas numéricas de produção e consumo dos átomos da rede de Petri t-temporal da Figura 3.	56
Tabela 5 – Relação entre transições e cenários	73
Tabela 6 – Comparação entre os diferentes tipos de <i>Soundness</i> e o resultado do Método para Detecção de Cenários Livres de <i>Deadlock</i> em processos de <i>workflow</i> interorganizacionais	77
Tabela 7 – Relação entre transições e cenários do sistema composto $C1 \oplus T1$. . .	100
Tabela 8 – Datas de produção do átomo o nos cenários livres de <i>deadlock</i>	113
Tabela 9 – Intervalos de datas simbólicas, $[D_{Pmin}, D_{Pmax}]$, para a finalização dos cenários livres de <i>deadlock</i>	115
Tabela 10 – Intervalos de datas numéricas para a finalização dos cenários livres de <i>deadlock</i>	115
Tabela 11 – Comparação entre as datas numéricas e simuladas para finalização dos cenários livres de <i>deadlock</i>	121
Tabela 12 – Intervalos de datas simbólicas para execução de tarefas do tipo usuário dos cenários $Sc_1A \times Sc_3PC$, $Sc_2A \times Sc_2PC$, $Sc_3A \times Sc_1PC$ e $Sc_4A \times Sc_1PC$	122
Tabela 13 – Intervalos de datas numéricas para execução de tarefas do tipo usuário dos cenários $Sc_1A \times Sc_3PC$, $Sc_2A \times Sc_2PC$, $Sc_3A \times Sc_1PC$ e $Sc_4A \times Sc_1PC$	123
Tabela 14 – Intervalos de datas simuladas para execução de tarefas do tipo usuário do cenário $Sc_1A \times Sc_3PC$	124

Tabela 15 – Intervalos de datas simuladas para execução de tarefas do tipo usuário do cenário $Sc_2A \times Sc_2PC$	124
Tabela 16 – Intervalos de datas simuladas para execução de tarefas do tipo usuário do cenário $Sc_3A \times Sc_1PC$	125
Tabela 17 – Intervalos de datas simuladas para execução de tarefas do tipo usuário do cenário $Sc_4A \times Sc_1PC$	127
Tabela 18 – Comparação entre os intervalos de datas numéricas e simuladas para execução de tarefas do tipo usuário dos cenários livres de <i>deadlock</i> . . .	127

Lista de siglas

ATL Atlas Transformation Language

BPM Business Process Management

BPMN Business Process Model and Notation

IOWF-net Interorganizational WorkFlow net

ICP Input Communication Place

LWF-net Local WorkFlow net

OCP Output Communication Place

SOA Service Oriented Architecture

UML Unified Modeling Language

U(**IOWF-net**) Unfolded Interorganizational WorkFlow net

YAWL Yet Another Workflow Language

Sumário

1	Introdução	25
1.1	Motivação	28
1.2	Objetivos da Pesquisa	30
1.3	Hipóteses	32
1.4	Contribuições	33
1.5	Organização da Tese	33
2	Fundamentação Teórica	35
2.1	Fundamentos Teóricos	35
2.1.1	Rede de Petri	35
2.1.2	WorkFlow net	40
2.1.3	WorkFlow net Interorganizacional	45
2.1.4	Rede de Petri e Lógica Linear	47
2.2	Trabalhos Relacionados	56
3	Métodos Baseados na Lógica Linear para Verificação de Processos de Workflow Interorganizacionais	59
3.1	Verificação do Critério de Correção <i>Soundness</i> para Processos de <i>Workflow</i> Interorganizacionais	61
3.2	Verificação do Critério de Correção <i>Relaxed Soundness</i> para Processos de <i>Workflow</i> Interorganizacionais	70
3.3	Verificação do Critério de Correção <i>Weak Soundness</i> para Processos de <i>Workflow</i> Interorganizacionais	74
3.4	Detecção de Cenários Livres de <i>Deadlock</i> em Processos de <i>Workflow</i> Inte- rorganizacionais	77
4	Estudo de Caso	89
4.1	Módulo de <i>Workflow</i> , Sistema Composto e WorkFlow net Interorganizacional	90
4.2	Verificação de Composição de Serviços Web utilizando Lógica Linear	93
4.2.1	Verificação do critério de correção <i>Soundness</i> para o sistema com- posto $C1 \oplus T1$	93

4.2.2	Verificação do critério de correção <i>Relaxed Soundness</i> para o sistema composto $C1 \oplus T1$	98
4.2.3	Verificação do critério de correção <i>Weak Soundness</i> para o sistema composto $C1 \oplus T1$	100
4.2.4	Detecção de cenários livres de <i>deadlock</i> no sistema composto $C1 \oplus T1$	101
4.3	Avaliação dos Resultados	104
5	Planejamento de Recursos e Monitoração de Cenários Livres de <i>Deadlock</i> utilizando Árvores de Prova da Lógica Linear	111
5.1	Planejamento de Recursos: uma análise quantitativa	111
5.2	Árvores de Prova da Lógica Linear e a Monitoração de Cenários Livres de <i>Deadlock</i>	127
6	Conclusão	137
6.1	Principais Contribuições	137
6.2	Trabalhos Futuros	140
6.3	Contribuições em Produção Bibliográfica	141
	Referências	143

Apêndices 149

APÊNDICE A	Árvores de Prova da Lógica Linear	151
A.1	Árvores de Prova da Lógica Linear para Aplicação dos Métodos para Verificação dos Critérios de Correção <i>Soundness</i> , <i>Relaxed Soundness</i> e <i>Weak Soundness</i>	151
A.2	Árvores de Prova da Lógica Linear para Aplicação do Método para Detecção de Cenários Livres de <i>Deadlock</i>	158
A.2.1	Árvores de Prova da Lógica Linear Rotuladas	165
A.3	Árvores de Prova da Lógica Linear para o Estudo de Caso	169
A.3.1	Árvores de Prova da Lógica Linear Rotuladas para o Estudo de Caso	172
A.4	Árvores de prova da Lógica Linear com Cálculo de Datas para o Planejamento de Recursos	174
A.5	Árvores de Prova da Lógica Linear com Cálculo de Datas para Monitoração de Cenários Livres de <i>Deadlock</i>	180

Introdução

O principal objetivo dos sistemas de gerenciamento de *workflow* é suportar a definição, a execução, o registro e o controle de processos de *workflow* (AALST, 1998a). De acordo com Aalst e Hee (2004), os processos de *workflow* representam as sequências de atividades que devem ser executadas em uma organização para tratar casos específicos e alcançar uma meta bem definida.

De acordo com Lomazova (2010), *workflow* é a automação de um processo de negócio, onde negócio é entendido de forma ampla, incluindo o ambiente organizacional e sistemas computacionais, sendo a modelagem de processos de *workflow* a primeira e mais importante fase no ciclo de vida de gerenciamento de *workflow*.

De acordo com Aalst et al. (2010), a área de Gerenciamento de Processos de Negócio, do inglês *Business Process Management (BPM)*, tem recebido atenção considerável nos últimos anos em razão de seu potencial em aumentar significativamente a produtividade das organizações e economizar gastos. Além disso, os autores também destacam que o conceito de processo é fundamental e serve como um ponto de partida para o entendimento de como o negócio opera e quais oportunidades existem para coordenar suas atividades constituintes.

Várias linguagens de modelagem podem ser utilizadas no contexto da modelagem de processos de *workflow* como, por exemplo, *Business Process Model and Notation (BPMN)* (OMG, 2011a), *Yet Another Workflow Language (YAWL)* (AALST et al., 2010; AALST; HOFSTEDE, 2005), rede de Petri (MURATA, 1989) e, até mesmo, Diagramas de Atividade da *Unified Modeling Language (UML)* (OMG, 2011b). Um dos problemas encontrados quando utiliza-se modelos semiformais ou informais, como é o caso de BPMN, YAWL e UML, é a verificação dos processos de *workflow* modelados utilizando estas linguagens.

De acordo com Valette e Cardoso (1997), rede de Petri (MURATA, 1989) é uma ferramenta gráfica e pode ser usado para gerar um modelo formal que pode ser utilizado para a representação de diversos tipos de sistemas a eventos discretos. O fato de ser formal permite, entre outras características, a verificação formal de certas “boas” propriedades do modelo. Assim, Aalst (1996b) apresenta, inicialmente, três motivos para a utilização da

rede de Petri no contexto dos sistemas de gerenciamento de *workflow*. Na sequência, Aalst (1996a) define formalmente a WorkFlow net, que é uma rede de Petri que modela processos de *workflow* simples. A WorkFlow net tornou-se referência na área de modelagem e análise formais de processos de *workflow* e vários trabalhos passaram a considerá-la em suas abordagens. Aalst (1998a) reforça os motivos apresentados em seu trabalho anterior, contabilizando seis motivos para utilizar a rede de Petri na modelagem de processos de *workflow*: semântica formal, natureza gráfica, expressividade, variedade de propriedades já demonstradas, disponibilidade de várias técnicas de análise e, finalmente, o fato de ser uma ferramenta independente, ou seja, não proprietária.

Segundo Aalst (1998a), a correção, eficácia e eficiência de processos de *workflow* são vitais para as organizações e uma definição de processo de *workflow* que contém erros pode levar a clientes irritados, alto volume de pedidos acumulados, reivindicações de danos ou perdas, isto é, a uma baixa qualidade dos serviços prestados. Por isso, é importante verificar os processos de *workflow* antes de colocá-los em operação.

Aalst (1998a) define três tipos básicos de análise de processos de *workflow*: validação, verificação e análise de desempenho. A verificação preocupa-se com a correção de um processo de *workflow*. Assim, no contexto da WorkFlow net, é definida a propriedade *Soundness* (AALST, 1996a; AALST, 1997). Tal propriedade é considerada como o principal critério de correção para os modelos em WorkFlow net.

É importante verificar a propriedade *Soundness* para um processo de *workflow* pois, como mostra Aalst (1998a), tendo-se um processo de *workflow Sound*, haverá a garantia de que: se uma instância de processo começou a ser tratada, o tratamento desta será finalizado; quando este tratamento for finalizado, não poderá haver nenhuma pendência não tratada no processo para tal instância; e, finalmente, não haverá tarefas que não serão executadas para nenhuma instância do processo de *workflow*, ou seja, haverá ausência de tarefas mortas no processo verificado. Vários trabalhos foram desenvolvidos considerando a verificação de tal propriedade como, por exemplo, as abordagens apresentadas por Aalst (1997), Aalst (1999), Passos e Julia (2009) e, mais recentemente, Aalst et al. (2011) que apresentam várias variantes do critério *Soundness*, como *Relaxed Soundness* e *Weak Soundness*, definidas para os modelos em WorkFlow net.

Considerando a importância do critério de correção *Soundness*, Fahland et al. (2011) verificaram 735 processos de *workflow* industriais, de serviços financeiros, telecomunicações e outros domínios. Os autores mostram que apenas 46% dos modelos de processos verificados são, de fato, *Sound*. Ou seja, a maioria dos processos de *workflow* analisados não satisfizeram o critério de correção *Soundness*. O trabalho desenvolvido por Fahland et al. (2011) permite concluir que, apesar da importância do critério de correção *Soundness*, na prática, na maioria das vezes, os processos de *workflow* não satisfazem tal critério.

É importante destacar que os processos de *workflow* modelados em WorkFlow net consideram o envolvimento de uma única organização nos processos modelados. No en-

tanto, atualmente, há situações em que várias organizações operam em conjunto para que um objetivo comum de negócio seja atingido. De acordo com Captarencu (2012), os processos de *workflow* interorganizacionais representam os processos de *workflow* que envolvem vários processos de *workflow* pertencentes a diferentes organizações, que precisam ter suas ações coordenadas para atingirem uma meta comum. Para Lim, Kerschbaum e Wang (2012), os sistemas de *workflow* interorganizacionais desempenham um papel fundamental em parcerias de negócio. Além disso, os autores destacam que, atualmente, formar uma aliança com parceiros de negócios adequados é uma estratégia comum para as empresas se manterem competitivas, oferecendo uma gama mais ampla de produtos e serviços aos seus clientes.

De acordo com Aalst (1998b), um processo de *workflow* interorganizacional é essencialmente um conjunto de processos de *workflow* fracamente acoplados onde, tipicamente, existem n parceiros de negócio, isto é, processos de *workflow* locais, envolvidos em um processo de *workflow* global. Ainda segundo Aalst (1998b), estes processos de *workflow* locais precisam se comunicar para a correta execução dos casos tratados pelo sistema de gerenciamento de *workflow* global. O processo de *workflow* global consiste, assim, de processos de *workflow* locais interagindo entre si (AALST, 1998b).

Vários estudos já consideraram a análise do critério de correção *Soundness* para os processos de *workflow* interorganizacionais. Em Aalst (1998b), Aalst (2000), Yamaguchi et al. (2007) e Sun e Du (2008), por exemplo, as abordagens propostas são baseadas na construção e análise de grafos das marcações acessíveis. O problema com esse tipo de abordagem é que a construção do grafo das marcações acessíveis pode levar a uma explosão do espaço de estados. Como mostra Aalst (1996a), decidir *Soundness* considerando as propriedades da vivacidade e limitabilidade pode ser intratável por este ser um problema EXPSPACE-difícil (procedimento determinístico e espaço exponencial). Além disso, essas propriedades são definidas em um contexto de modelos em redes de Petri com um comportamento cíclico, que não é o caso dos modelos em redes de Petri de processos de *workflow* interorganizacionais.

Uma possibilidade para contornar o problema da explosão do espaço de estados é utilizar a Lógica Linear, proposta por Girard (1987). Neste contexto, alguns trabalhos mostraram a ligação entre a teoria da rede de Petri e a Lógica Linear como, por exemplo, os trabalhos apresentados por Girault, Pradin-Chézalviel e Valette (1997), Pradin-Chézalviel, Valette e Kunzle (1999) e Riviere, Pradin-Chézalviel e Valette (2001), uma vez que existe uma tradução quase direta entre a estrutura de uma rede de Petri e um sequente da Lógica Linear. Através da prova de um sequente linear, pode-se provar características importantes, como a alcançabilidade em um modelo em rede de Petri, sem considerar a construção de um grafo das marcações acessíveis, como mostram Girault, Pradin-Chézalviel e Valette (1997).

No contexto dos processos de *workflow* simples, Passos e Julia (2009) mostram como a

Lógica Linear pode ser utilizada para provar o critério de correção *Soundness* para modelos em *WorkFlow nets*, sem que seja necessária a construção de um grafo das marcações acessíveis, considerando diretamente a própria estrutura do modelo em *WorkFlow net*, em vez de considerar seu autômato correspondente. No que se diz respeito à estrutura destes modelos, é importante destacar que as árvores de prova da Lógica Linear mantém a estrutura acíclica dos modelos em *WorkFlow net*, diferentemente das abordagens clássicas, que transformam estas em modelos cíclicos. Neste contexto, Passos e Julia (2009) mostram diretamente o relacionamento entre a Lógica Linear e a prova do critério de correção *Soundness*, definido para os modelos em *WorkFlow net*.

Desta forma, considerando a importância da verificação dos processos de *workflow* interorganizacionais modelados por rede de Petri e a possibilidade do uso da Lógica Linear para a verificação de tais processos, é que se delimita o contexto deste trabalho.

1.1 Motivação

Considerando os aspectos anteriormente apresentados, relacionados principalmente à verificação dos processos de *workflow* interorganizacionais e ao uso da Lógica Linear como ferramenta para tal verificação, é importante destacar que a abordagem apresentada por Passos e Julia (2009) utiliza a Lógica Linear para a verificação de *Soundness* no contexto dos processos de *workflow* envolvidos em uma única organização, ou seja, os processos de *workflow* organizacionais simples, modelados em *WorkFlow net*. Assim, a abordagem apresentada por Passos e Julia (2009) não considera a comunicação entre processos (organizações) e, conseqüentemente, não considera a verificação do critério de correção *Soundness* para os processos de *workflow* interorganizacionais.

Um aspecto relevante do critério de correção *Soundness*, no contexto dos processos de *workflow* interorganizacionais, é que este considera a verificação individual dos processos de *workflow* locais e o processo de *workflow* global (AALST, 1998b). Neste caso, se os processos de *workflow* locais não satisfazem o critério de correção *Soundness*, tais processos precisam ser reprojitados, remodelados e, na sequência, uma nova verificação dos processos remodelados precisa ser realizada. Este processo de modelagem e verificação é repetido até que os processos de *workflow* locais satisfaçam o critério de correção *Soundness*. Obtendo-se processos de *workflow* locais que satisfazem o critério de correção *Soundness*, o processo global deve ser analisado. Assim, uma nova verificação de *Soundness* deve ser realizada, considerando então o processo de *workflow* global. Se o processo de *workflow* global analisado não satisfizer o critério de correção *Soundness*, este precisa ser reprojitado, remodelado e novamente verificado até tal critério ser satisfeito.

Assim, além desta verificação ser extremamente complexa e demorada, o principal problema encontrado é a possível necessidade de modificação dos processos locais e global do processo de *workflow* interorganizacional. Principalmente porque, tratando-se de

processos interorganizacionais, os processos locais pertencem geralmente a organizações distintas. E esses processos locais, por exemplo, podem ser utilizados para colaboração com mais de uma organização parceira de negócio. Desta forma, uma alteração em um dado processo local poderá solucionar problemas relacionados ao critério de correção *Soundness* com uma organização parceira mas, pode gerar problemas com outra organização parceira de negócio. Além disso, os processos de *workflow* são geralmente ativos importantes das organizações, que nem sempre estão dispostas a modificá-los.

A situação ideal é aquela em que os processos de *workflow* interorganizacionais satisfazem o critério de correção *Soundness*. No entanto, conforme mostram os resultados apresentados por Fahland et al. (2011), esta não é a situação real dos processos de *workflow* simples. Como um processo de *workflow* interorganizacional é essencialmente um conjunto de processos de *workflow* locais envolvidos em um processo de *workflow* global, como definido por Aalst (1998b), e uma ampla porcentagem desses processos de *workflow* locais não satisfaz o critério de correção *Soundness*, o processo de *workflow* interorganizacional que considerar um desses processos locais não *Sound* também será não *Sound*, considerando a definição de *Soundness* para processos de *workflow* interorganizacionais proposta por Aalst (1998b). Assim, pode-se concluir que uma ampla gama de processos de *workflow* interorganizacionais pode não satisfazer o critério de correção *Soundness*.

No contexto dos processos de *workflow* simples, Dehnert e Rittgen (2001) propuseram relaxar o critério de correção *Soundness*. O novo critério definido pelos autores é o critério *Relaxed Soundness*. A ideia por trás deste critério é que o comportamento do sistema é correto se cada tarefa do processo é considerada em pelo menos uma execução que finaliza corretamente (DEHNERT; RITTGEN, 2001). Na abordagem apresentada por Siegeris e Zimmermann (2006), várias propostas de composição de modelos de *workflow* são apresentadas e os autores investigam a capacidade destes mecanismos de composição preservarem o critério de correção *Relaxed Soundness*. No entanto, para preservar o critério de correção *Relaxed Soundness*, os processos de *workflow* considerados na composição devem satisfazer inicialmente o critério *Relaxed Soundness*. Logo, se este não é o caso dos processos locais, tais processos de *workflow* precisam ser reprojitados, remodelados e verificados até que satisfaçam tal critério de correção, para depois serem considerados em uma composição. Ou seja, mais uma vez os problemas relacionados às modificações em processos de *workflow* locais anteriormente discutidos serão encontrados.

Uma outra variante do critério de correção *Soundness* é proposta por Martens (2003). O autor propõe o enfraquecimento do critério de correção *Soundness*. O novo critério definido pelo autor é o critério de correção *Weak Soundness*. A ideia por trás deste critério é que o comportamento do sistema é correto quando há ausência de *deadlock* no modelo e a correta finalização do processo é garantida, mesmo nos casos em que o processo analisado apresentar tarefas mortas. Se o modelo analisado apresentar situação de *deadlock*, por exemplo, os processos envolvidos devem ser reprojitados e remodelados

até que sua verificação satisfaça o critério de correção *Weak Soundness*.

É importante observar que o fato de cada tarefa do processo de *workflow* interorganizacional ser considerada pelo menos uma vez não garante que todas as possíveis necessidades de negócio estarão contempladas. Pode haver uma dada tarefa do processo que é contemplada uma única vez, satisfazendo o critério *Relaxed Soundness*, mas esta única contemplação pode não satisfazer as necessidades de negócio das organizações envolvidas, por exemplo. Além disso, um processo de *workflow* interorganizacional pode prover todas as necessidades comuns de negócio das organizações envolvidas, sem que alguma dada tarefa do processo global seja considerada. Além disso, nem sempre será possível obter um processo de *workflow* interorganizacional completamente livre de *deadlocks*. Desta forma, é interessante considerar, ainda no contexto da verificação de processos de *workflow* interorganizacionais não *Sound*, a detecção de todos dos cenários livres de bloqueio mortal (*deadlock*) em um processo de *workflow* interorganizacional. A obtenção de tais cenários livres de *deadlock* permite às organizações envolvidas analisar se tais cenários livres de bloqueio podem ou não satisfazer suas necessidades de negócio. Se este for o caso, somente tais cenários livres de *deadlock* deverão ser considerados na execução do processo de *workflow* interorganizacional evitando, assim, os cenários que eventualmente poderiam levar o sistema a uma situação de *deadlock*.

1.2 Objetivos da Pesquisa

Refletindo sobre as questões apresentadas na seção 1.1, inicialmente, considera-se a formalização do método para verificação do critério de correção *Soundness* no contexto dos processos de *workflow* interorganizacionais, utilizando a Lógica Linear. Tal método permitirá a verificação de *Soundness* para processos de *workflow* interorganizacionais sem considerar a construção de um grafo das marcações acessíveis, mas considerando a própria estrutura acíclica do modelo analisado.

Além disso, e considerando ainda os aspectos discutidos na seção 1.1, é importante dar atenção aos processos de *workflow* interorganizacionais que não satisfazem o critério de correção *Soundness*. Nestes casos, é de grande interesse verificar os critérios de correção clássicos, como *Relaxed Soundness* e *Weak Soundness* para os processos de *workflow* interorganizacionais não *Sound*, utilizando a Lógica Linear. O uso da Lógica Linear, em ambos os casos, permitirá ainda o reuso de árvores de prova construídas para a verificação do critério de correção *Soundness* para o processo de *workflow* interorganizacional considerado.

Levando-se em conta os critérios de correção *Relaxed Soundness* e *Weak Soundness*, formaliza-se aqui um método que identifica todos os cenários livres de *deadlock*, permitindo, assim, que as organizações envolvidas possam analisar tais cenários e verificar se os mesmos satisfazem suas necessidades de negócio. Utilizando este método, as organizações

envolvidas no processo de *workflow* interorganizacional poderão verificar se seus principais serviços de negócio podem ser realizados por completo, evitando as situações de *deadlock* mesmo quando elas ocorrerem no modelo global, sem a necessidade de revisão de seus processos de *workflow* locais e global para satisfazer um dado critério de correção.

Uma vez que existe uma ligação estreita entre a modelagem e verificação de processos de *workflow* interorganizacionais e a modelagem e verificação de composições de serviços Web (do inglês *Web Services*), como mostram Martens (2003), Martens (2005), Xiong, Fan e Zhou (2010), Klai, Ochi e Tata (2013), por exemplo, é também objetivo desta pesquisa aplicar os métodos de verificação propostos no contexto dos processos de *workflow* interorganizacionais a uma composição de serviços Web, como estudo de caso, para ilustrar a aplicação dos métodos formalizados nesta pesquisa.

Além disso, é objetivo desta pesquisa mostrar como os resultados fornecidos no contexto da análise qualitativa (verificação das propriedades relacionadas à noção do critério de correção *Soundness*) poderão ser diretamente explorados num tipo de análise quantitativa cujo propósito é o planejamento dos recursos envolvidos na execução das tarefas do processo de *workflow* interorganizacional através da definição de intervalos de datas simbólicas que servirão à localização no tempo da disponibilidade dos recursos. É importante também mostrar como os resultados das diversas análises (qualitativa e quantitativa) serão explorados para especificar um tipo de controle supervisorio para assegurar os cenários livres de *deadlock* em tempo de execução, isto é, que contorna as situações de *deadlock* quando elas existem para que os processos possam ser devidamente realizados.

Do exposto, os objetivos de pesquisa podem ser estruturados da seguinte maneira:

- a) formalizar um método para verificação do critério de correção *Soundness* para processos de *workflow* interorganizacionais, sem considerar a construção e análise de grafos das marcações acessíveis correspondentes, considerando diretamente a estrutura acíclica do modelo utilizado para análise do processo;
- b) formalizar um método para verificação do critério de correção *Relaxed Soundness* para processos de *workflow* interorganizacionais não *Sound*, baseado em reuso, sem que seja necessário verificar e/ou modificar obrigatoriamente os processos de *workflow* locais para torná-los *Relaxed Sound*, considerando a estrutura acíclica do modelo utilizado para a análise do processo;
- c) formalizar um método para verificação do critério de correção *Weak Soundness* para processos de *workflow* interorganizacionais não *Sound*, baseado em reuso, considerando a estrutura acíclica do modelo utilizado para a análise do processo;
- d) formalizar um método para verificação de processos de *workflow* interorganizacionais não *Sound* que permita determinar todos os cenários livres de bloqueio (*deadlock*) no processo analisado;

- e) realizar um estudo de caso ilustrativo, aplicando os métodos formalizados no contexto dos processos de *workflow* interorganizacionais ao contexto da composição de serviços Web, obtendo assim a verificação de composições de serviços Web;
- f) mostrar como é realizado o planejamento de recursos para o tratamento de tarefas, que deve definir intervalos de datas simbólicas nas quais os recursos devem permanecer disponíveis e apresentar uma visão geral sobre a monitoração da execução dos cenários livres de *deadlock*.

1.3 Hipóteses

Considerando os objetivos da pesquisa apresentados na seção 1.2, são consideradas as seguintes hipóteses:

- a) o uso da Lógica Linear, mais especificamente a construção e análise de árvores de prova da Lógica Linear, permite verificar o critério de correção *Soundness* para processos de *workflow* interorganizacionais, sem considerar a construção e análise de grafos das marcações acessíveis correspondentes, considerando diretamente a estrutura acíclica do modelo utilizado para análise do processo;
- b) o reuso de árvores de prova da Lógica Linear construídas para a verificação do critério de correção *Soundness* para um processo de *workflow* interorganizacional e um método de análise para tais árvores de prova reutilizadas, permite a verificação do critério de correção *Relaxed Soundness* para processos de *workflow* interorganizacionais não *Sound*, sem que seja necessária a verificação do critério de correção *Relaxed Soundness* para os processos de *workflow* locais de forma singular e considerando diretamente a estrutura acíclica do modelo utilizado para análise do processo;
- c) o reuso de árvores de prova da Lógica Linear construídas para a verificação do critério de correção *Soundness* para um processo de *workflow* interorganizacional e um método de análise para tais árvores de prova reutilizadas, permite a verificação do critério de correção *Weak Soundness* para processos de *workflow* interorganizacionais não *Sound*, considerando diretamente a estrutura acíclica do modelo utilizado para análise do processo;
- d) o uso da Lógica Linear, mais especificamente a construção e análise de árvores de prova da Lógica Linear, permite determinar todos os cenários livres de *deadlock* em um processo de *workflow* interorganizacional;
- e) um estudo de caso pode ser realizado, no contexto da verificação de composições de serviços Web modelados por rede de Petri, considerando as hipóteses a), b), c) e d), permitindo mostrar a utilização da Lógica Linear para a verificação de tais composições;

- f) dados derivados das árvores da Lógica Linear permitem realizar um planejamento de recursos para o tratamento de tarefas, de forma que sejam determinados intervalos de datas simbólicas nas quais os recursos devem permanecer disponíveis. Além disso, tais dados derivados das árvores de prova podem ser utilizados em estratégias de monitoração dos cenários livres de *deadlock*, de forma que o sistema global evite as situações de bloqueio.

1.4 Contribuições

Considerando os objetivos da pesquisa e as hipóteses apresentadas, respectivamente, nas seções 1.2 e 1.3, são contribuições desta pesquisa:

- a) um método para verificação do critério de correção *Soundness* para processos de *workflow* interorganizacionais, baseado na construção e análise de árvores de prova da Lógica Linear;
- b) um método para verificação do critério de correção *Relaxed Soundness* para processos de *workflow* interorganizacionais não *Sound*, baseado no reuso de árvores de prova da Lógica Linear construídas para a verificação do critério de correção *Soundness* para processos de *workflow* interorganizacionais e na análise de tais árvores de prova reutilizadas;
- c) um método para verificação do critério de correção *Weak Soundness* para processos de *workflow* interorganizacionais não *Sound*, baseado no reuso de árvores de prova da Lógica Linear construídas para a verificação do critério de correção *Soundness* para processos de *workflow* interorganizacionais e na análise de tais árvores de prova reutilizadas;
- d) um método para verificação de processos de *workflow* interorganizacionais não *Sound*, baseado na construção e análise de árvores de prova da Lógica Linear, que determina todos os cenários livres de bloqueio (*deadlock*) no processo analisado;
- e) um estudo de caso ilustrativo que mostra como a verificação de composições de serviços Web deve ser realizada, utilizando os métodos propostos nesta pesquisa.
- f) um método para o planejamento de recursos para o tratamento de tarefas, de forma que sejam determinados intervalos de datas simbólicas nas quais os recursos devem permanecer disponíveis e uma visão geral sobre como os dados derivados das árvores de prova podem ser utilizados para a monitoração dos cenários livres de *deadlock*.

1.5 Organização da Tese

A presente tese encontra-se organizada em seis capítulos, da maneira como se segue.

No Capítulo 2 é apresentada a fundamentação teórica necessária para a compreensão dos métodos formalizados nesta pesquisa. Os fundamentos teóricos são apresentados na seção 2.1 e a seção 2.2 apresenta os trabalhos relacionados. Os fundamentos teóricos, relacionados à Rede de Petri, à WorkFlow net, à WorkFlow net Interorganizacional e que mostram a relação entre rede de Petri e Lógica Linear, são apresentados nas subseções 2.1.1, 2.1.2, 2.1.3 e 2.1.4, respectivamente.

No Capítulo 3 são apresentados os métodos baseados na Lógica Linear para verificação de processos de *workflow* interorganizacionais. O método para verificação do critério de correção *Soundness* para processos de *workflow* interorganizacionais é formalizado na seção 3.1. O método para verificação do critério de correção *Relaxed Soundness* para processos de *workflow* interorganizacionais é formalizado na seção 3.2. Já o método para verificação do critério de correção *Weak Soundness* para processos de *workflow* interorganizacionais é formalizado na seção 3.3. A seção 3.4 formaliza o método para detecção de cenários livres de *deadlock* em processos de *workflow* interorganizacionais.

O Capítulo 4 apresenta o estudo de caso ilustrativo, que mostra a verificação de composições de serviços Web, considerando os métodos baseados na Lógica Linear, apresentados no Capítulo 3. A seção 4.1 apresenta a relação direta entre a modelagem de serviços Web e a modelagem de processos de *workflow* interorganizacionais. Já a seção 4.2 apresenta a verificação de uma composição de serviços Web no que se diz respeito aos critérios de correção *Soundness*, *Relaxed Soundness* e *Weak Soundness*, respectivamente, nas subseções 4.2.1, 4.2.2 e 4.2.3. Além disso, na subseção 4.2.4, é apresentada a aplicação do método para detecção de cenários livres de *deadlock* a uma composição de serviços Web. A seção 4.3 finaliza este capítulo, apresentando uma avaliação dos resultados obtidos no estudo de caso.

O planejamento de recursos e a monitoração de cenários livres de *deadlock* utilizando árvores de prova da Lógica Linear são apresentados no Capítulo 5. Na seção 5.1 é apresentado o planejamento de recursos, ou seja, a análise quantitativa para processos de *workflow* interorganizacionais e composições de serviços Web. Já na seção 5.2, é apresentada uma visão geral sobre maneiras de monitoração de cenários livres de *deadlock*.

No Capítulo 6 a conclusão desta pesquisa é apresentada. A seção 6.1 apresenta as principais contribuições, os trabalhos futuros são apresentados na seção 6.2 e, finalmente, na seção 6.3 são apresentadas as contribuições na forma de produção bibliográfica derivadas da presente pesquisa.

Fundamentação Teórica

Neste capítulo é apresentada a fundamentação teórica desta pesquisa. Os conceitos teóricos relacionados à rede de Petri e à Lógica Linear são apresentados na seção 2.1. A seção 2.2 apresenta os trabalhos relacionados.

2.1 Fundamentos Teóricos

Esta seção apresenta os fundamentos teóricos necessários para compreender os métodos formalizados no Capítulo 3. A subseção 2.1.1 apresenta as terminologias e notações básicas relacionadas à teoria da rede de Petri, que são essenciais para o entendimento da WorkFlow net e da WorkFlow net interorganizacional, que são apresentadas nas subseções 2.1.2 e 2.1.3, respectivamente. A subseção 2.1.4 apresenta os conceitos básicos da relação entre a rede de Petri e a Lógica Linear, que são utilizados nos métodos apresentados no Capítulo 3.

2.1.1 Redes de Petri

A teoria inicial da rede de Petri foi proposta por Carl Adam Petri em sua tese de doutorado ((PETRI, 1962)). Desde então, o estudo e uso da rede de Petri tem crescido consideravelmente. Murata (1989) apresenta uma revisão histórica e bibliográfica extensa da rede de Petri.

De acordo com Murata (1989), a rede de Petri clássica é um grafo bipartido direcionado com dois tipos de nós denominados lugares e transições, conectados via arcos direcionados. As conexões entre dois nós do mesmo tipo não é permitida (MURATA, 1989). Os lugares são representados por círculos e as transições por retângulos ou barras. Murata (1989) apresenta a seguinte definição para uma rede de Petri.

Definição 1. (Rede de Petri)

Uma rede de Petri clássica é uma tripla $PN = \{P, T, F\}$, onde:

- a) P é um conjunto finito de lugares;

- b) T é um conjunto finito de transições, onde $(P \cap T = \emptyset)$;
 c) $F \subseteq (P \times T) \cup (T \times P)$ é um conjunto de arcos (relação de fluxo).

Um lugar p é denominado lugar de entrada de uma transição t se, e somente se, existe um arco direcionado de p para t . Por exemplo, considerando a rede de Petri da Figura 1(a), os lugares $P1$ e $P2$ são lugares de entrada da transição $t1$.

Um lugar p é chamado de lugar de saída de uma transição t se, e somente se, existe um arco direcionado de t para p . Por exemplo, considerando a rede de Petri da Figura 1(a), o lugar $P3$ é um lugar de saída da transição $t1$.

Um lugar p contém em um dado momento zero ou mais fichas (*tokens*), que são representadas por pontos pretos. Por exemplo, considerando a rede de Petri da Figura 1(a), o lugar $P1$ contém uma ficha e o lugar $P2$ contém zero fichas.

A marcação de uma rede de Petri refere-se à distribuição de fichas nos lugares, sendo que o número de fichas pode mudar durante a execução da rede. As transições são os componentes ativos em uma rede de Petri: elas mudam a marcação da rede de acordo com as seguintes regras de disparo (MURATA, 1989):

- a) uma transição t é dita sensibilizada se, e somente se, cada lugar de entrada p de t contém pelo menos uma ficha. Por exemplo, a transição $t1$ da rede de Petri da Figura 1(a) não está sensibilizada. Já a transição $t1$ da rede de Petri da Figura 1(b) está sensibilizada, pois há uma ficha em cada lugar de entrada desta transição;
- b) uma transição sensibilizada pode disparar. Se a transição t disparar, então t consome uma ficha de cada lugar de entrada p de t e produz uma ficha em cada lugar de saída p de t . As Figuras 1(b) e 1(c) mostram um exemplo de disparo de transição. Neste caso, $t1$ consome uma ficha de cada lugar de entrada ($P1$ e $P2$) e produz uma ficha em cada lugar de saída ($P3$), como pode-se verificar na Figura 1(c).

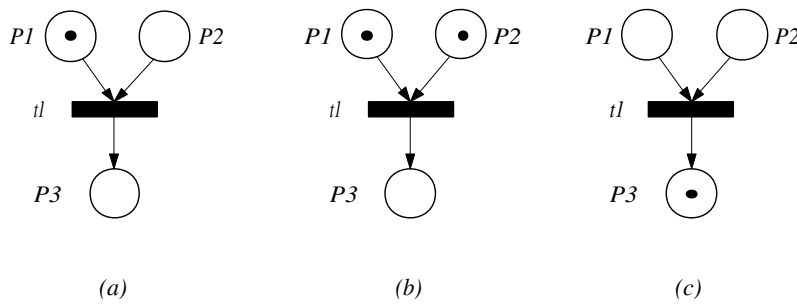


Figura 1 – Exemplos de sensibilização e disparo de transição em uma rede de Petri.

Murata (1989) utiliza $\bullet t$ para denotar o conjunto de lugares de entrada para uma transição t . As notações $t\bullet$, $\bullet p$ e $p\bullet$ possuem significados similares. Por exemplo, $p\bullet$ é o conjunto de transições que compartilham p como lugar de entrada. Considerando a Figura 1(a), tem-se: $\bullet P1 = \emptyset$, $P1\bullet = \{t1\}$, $\bullet P2 = \emptyset$, $P2\bullet = \{t1\}$, $\bullet P3 = t1$, $P3\bullet = \emptyset$, $\bullet t1 = \{P1, P2\}$ e $t1\bullet = \{P3\}$.

Um estado de uma rede de Petri, denominado marcação, é uma distribuição de fichas sobre os lugares. Uma marcação é denotada por M , um vetor de tamanho m , onde m é o número total de lugares. O p -ésimo componente de M , denotado por $M(p)$ é o número de fichas no lugar p (MURATA, 1989). Aalst (1998a) representa uma marcação da seguinte maneira: $1p1 + 2p2 + 1p3 + 0p4$ é o estado com uma ficha em $p1$, duas fichas em $p2$, uma ficha em $p3$ e nenhuma ficha em $p4$. Esta mesma marcação pode ser equivalentemente representada por $p1 + 2p2 + p3$.

Assim, dada uma rede de Petri PN e uma marcação M_1 , Murata (1989) apresenta as seguintes notações:

- a) $M_1 \xrightarrow{t} M_2$: a transição t é sensibilizada na marcação M_1 e o disparo de t resulta na marcação M_2 ;
- b) $M_1 \rightarrow M_2$: há uma transição t tal que $M_1 \xrightarrow{t} M_2$;
- c) $M_1 \xrightarrow{\sigma} M_n$: a sequência de disparo $\sigma = t_1 t_2 t_3 \dots t_{n-1}$ leva da marcação M_1 para a marcação M_n , isto é, $M_1 \xrightarrow{t_1} M_2 \xrightarrow{t_2} \dots \xrightarrow{t_{n-1}} M_n$.

Uma marcação M_n é alcançável a partir de M_1 (notação $M_1 \xrightarrow{*} M_n$) se, e somente, se existe uma sequência de disparo $\sigma = t_1 t_2 t_3 \dots t_{n-1}$ tal que $M_1 \xrightarrow{\sigma} M_n$ (MURATA, 1989). Além disso, a notação (PN, M) é utilizada para denotar uma rede de Petri PN com uma marcação inicial M . Uma marcação M' é alcançável a partir de (PN, M) se, e somente se, $M \xrightarrow{*} M'$ (MURATA, 1989).

De acordo com Aalst e Hee (2004), é importante considerar no contexto do gerenciamento de *workflow* redes de Petri de alto nível, que são extensões da rede de Petri clássica. Os autores destacam que as extensões mais importantes são as extensões de cor, tempo e de hierarquia. Na presente pesquisa são utilizadas somente as extensões de cor e tempo. Assim, na sequência são apresentadas noções básicas sobre a rede de Petri colorida e a rede de Petri t-temporal.

A rede de Petri colorida foi definida por Jensen (1981). Uma rede de Petri colorida, de acordo com Aalst e Stahl (2011), é uma rede de Petri onde cada lugar possui um tipo, e cada ficha possui um valor (isto é, uma cor) em conformidade com o tipo do lugar. De acordo com Valette e Cardoso (1997), as cores são adicionadas às fichas com o objetivo de diferenciá-las. Um arco em uma rede de Petri colorida pode possuir uma inscrição de arco, que é uma expressão com algumas variáveis que avaliam a um *multiset* – um *multiset* se assemelha a um conjunto comum, mas o mesmo elemento pode aparecer múltiplas vezes (AALST; STAHL, 2011). A cada lugar se associa o conjunto de cores das fichas que podem pertencer a este lugar e a cada transição se associa um conjunto de cores que corresponde às diferentes maneiras de disparar uma transição (VALETTE; CARDOSO, 1997). Além disso, uma transição pode possuir uma guarda, isto é, uma expressão booleana que condiciona o disparo da mesma (AALST; STAHL, 2011).

A definição formal de rede de Petri Colorida, de acordo com Valette e Cardoso (1997), é apresentada na sequência.

Definição 2. (Rede de Petri Colorida) Uma rede de Petri colorida N_c é uma sêxtupla $(P, T, C_{or}, C_{sc}, W, M_0)$ onde:

- a) P é um conjunto finito de lugares;
- b) T é um conjunto finito de transições;
- c) C_{or} é um conjunto finito de cores;
- d) C_{sc} é a função do subconjunto de cores que a cada lugar e a cada transição associa um sub-conjunto de C_{or} (as cores possíveis para este lugar ou esta transição): $C_{sc} : P \cup T \rightarrow \mathcal{P}(C_{or})$;
- e) W é a função de incidência; cada elemento $W(p, t)$ de W é também uma função: $W(p, t) : C_{sc}(t) \times C_{sc}(p) \rightarrow \mathbb{N}$;
- f) M_0 é a marcação inicial que associa, para cada lugar e para cada cor possível neste lugar, um número de fichas: $M_0(p) : C_{sc}(p) \rightarrow \mathbb{N}$.

A Figura 2 mostra modelos em rede de Petri colorida apresentados por Murata (1989). Os modelos consistem em uma transição t e quatro lugares: os lugares $P1$ e $P2$ são lugares de entrada e os lugares $P3$ e $P4$ são lugares de saída da transição t (MURATA, 1989). Os quatro arcos são rotulados por $2x$, $< x, y > + < y, z >$, $< x, z >$ e e , indicando quantas e quais tipos de fichas “coloridas” serão removidas ou adicionadas aos lugares (MURATA, 1989). De acordo com Murata (1989), quando a transição t dispara, tem-se que $P1$ perde duas fichas da mesma cor, x ; $P2$ perde duas fichas de cores diferentes, $< x, y >$ e $< y, z >$; $P3$ recebe uma ficha da cor $< x, z >$; e $P4$ recebe uma ficha da cor e (uma constante). A marcação inicial consiste de quatro fichas coloridas, dois a 's e dois d 's, em $P1$; $P2$ tem três fichas coloridas (pares ordenados), $< a, b >$, $< b, c >$ e $< d, a >$; $P3$ e $P4$ não possuem fichas inicialmente (MURATA, 1989). Neste exemplo, variáveis são denotadas por x, y, z, \dots e constantes por a, b, c, d, \dots ; para cada transição, uma variável do mesmo tipo aparecendo nos arcos de entrada e saída denotam a mesma variável e uma constante de mesmo símbolo é a mesma em toda a rede (MURATA, 1989). Na sequência são descritas as noções de sensibilização e disparo de transição apresentadas em (MURATA, 1989). Uma transição t é dita sensibilizada se há fichas suficientes das cores “corretas” em cada lugar de entrada de t . O termo cores “corretas” significa a existência de substituições consistentes de constantes em variáveis, as quais são consistentes com a rotulação dos arcos e outras restrições adicionais. Por exemplo, a transição t da Figura 2(a) está sensibilizada uma vez que há fichas suficientes nos seus lugares de entrada, com duas substituições consistentes: $\{a|x, b|y, c|z\}$ e $\{d|x, a|y, b|z\}$. Assim, há duas diferentes maneiras de disparar t , considerando tais substituições. As redes mostradas nas Figura 2(b) e (c) mostram a marcação após o disparo da transição t com as substituições $\{a|x, b|y, c|z\}$ e $\{d|x, a|y, b|z\}$, respectivamente (MURATA, 1989).

Em uma rede de Petri t-temporal (MERLIN, 1974), o tempo é representado por um intervalo $[\theta_{min}, \theta_{max}]$ associado a cada transição, de forma que este intervalo de tempo

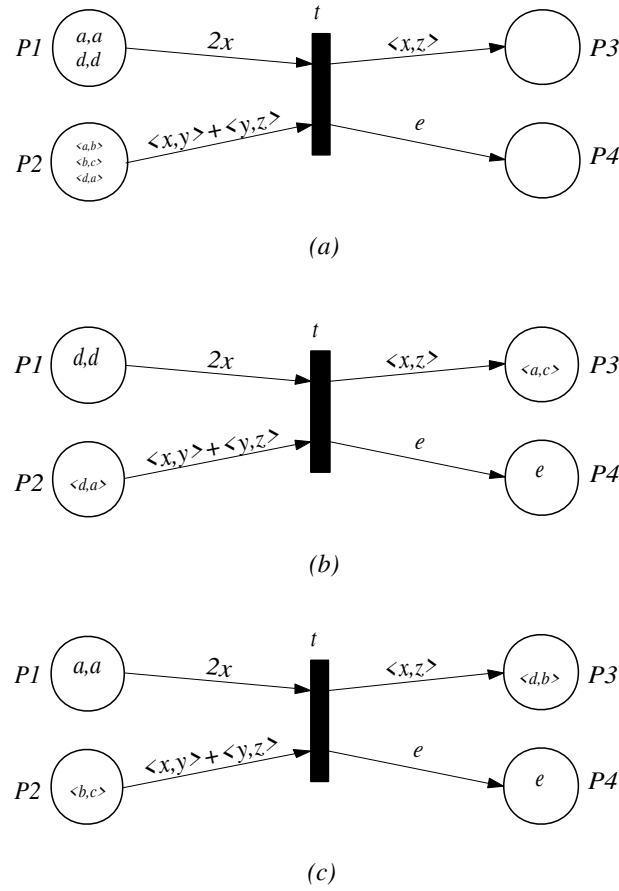


Figura 2 – Modelos em rede de Petri colorida (MURATA, 1989).

corresponde a uma duração de sensibilização. De acordo com Valette e Cardoso (1997), a duração de sensibilização deve ser maior que θ_{min} e menor que θ_{max} . O disparo de uma transição é instantâneo, mas a transição deve estar sensibilizada durante o intervalo de tempo dado (VALETTE; CARDOSO, 1997). A definição da rede de Petri t-temporal, considerada nesta pesquisa e adaptada a partir da definição apresentada por Valette e Cardoso (1997), é mostrada na sequência.

Definição 3. (Rede de Petri t-temporal) Uma rede de Petri t-temporal N é uma quádrupla (P, T, F, I) tal que:

a) (P, T, F) é uma rede de Petri, onde:

- P é um conjunto finito de lugares;
- T é um conjunto finito de transições; e
- $F \subseteq (P \times T) \cup (T \times P)$ é um conjunto de arcos (relação de fluxo).

b) I é uma aplicação que associa a cada transição $t \in T$ um intervalo de sensibilização $I(t) = [\theta_{min}(t), \theta_{max}(t)]$, onde $\theta_{min}(t)$ representa o tempo mínimo de disparo da transição e $\theta_{max}(t)$ representa o tempo máximo de disparo da transição t .

A Figura 3 mostra uma rede de Petri t-temporal. O intervalo $[8, 12]$ associado à transição $t4$ indica que esta transição irá disparar depois de pelo menos oito unidades de tempo após ter sido sensibilizada e no máximo doze unidades de tempo após a sua sensibilização.

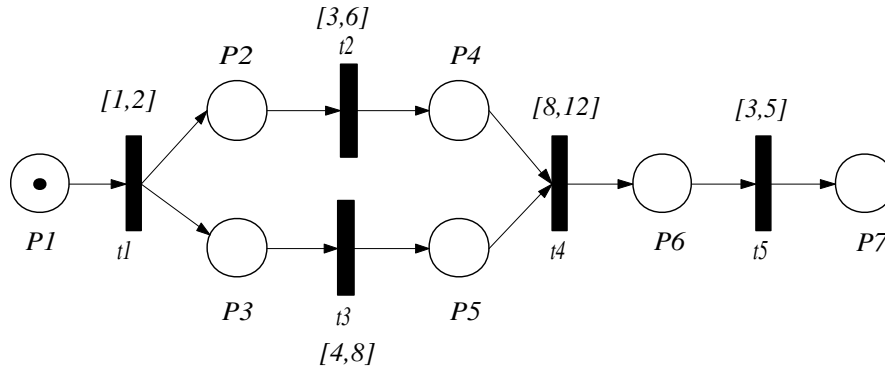


Figura 3 – Um exemplo de rede de Petri t-temporal.

2.1.2 WorkFlow net

Um processo de *workflow* define quais tarefas de um processo de negócio precisam ser executadas e em qual ordem a execução deve ocorrer. Além disso, de acordo com Aalst (1998a), processos de *workflow* são baseados em casos, isto é, cada parte do trabalho é executada para um caso específico. Modelar um processo de *workflow* em termos de uma rede de Petri é bem direto: transições são componentes ativos e modelam as tarefas, lugares são componentes passivos e modelam as condições (pré e pós), e as fichas modelam os casos (AALST, 1998a; AALST; HEE, 2004). Assim, uma rede de Petri que modela um processo de *workflow* é denominada WorkFlow net (AALST, 1998a). Uma WorkFlow net satisfaz as seguintes propriedades (AALST, 1998a):

- tem apenas um lugar de início denominado i e apenas um lugar de término, denominado o . Estes dois lugares são tratados como lugares especiais, onde o lugar i tem apenas arcos de saída e o lugar o possui apenas lugares de entrada;
- uma ficha no lugar i representa um caso que precisa ser tratado e uma ficha no lugar o representa um caso já tratado;
- toda tarefa t (transição) e condição p (lugar) deve estar em um caminho entre o lugar i e o lugar o .

Na sequência, a definição formal de uma WorkFlow net, proposta por Aalst (1998a), é apresentada.

Definição 4. (WorkFlow net) Uma rede de Petri $PN = \{P, T, F\}$ é uma WorkFlow net se, e somente se:

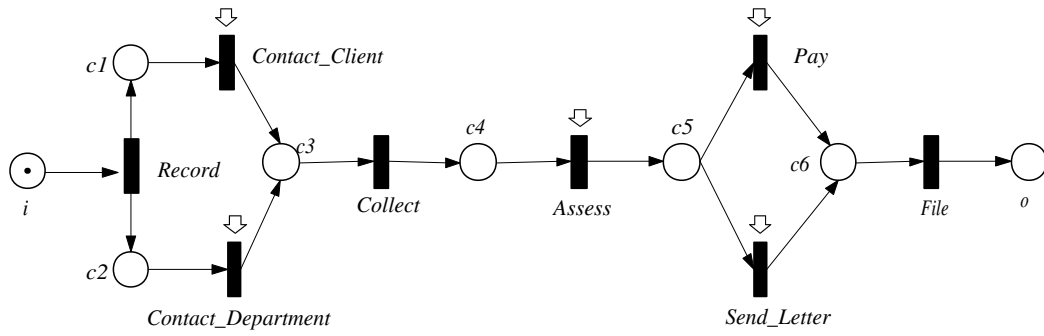


Figura 4 – WorkFlow net para o processo de tratamento de reclamações (AALST; HEE, 2004).

- PN tem dois lugares especiais: i e o . O lugar i é um lugar de início: $\bullet i = \emptyset$. O lugar o é um lugar de término: $o \bullet = \emptyset$;
- todo nó está em um caminho entre os lugares i e o .

Para ilustrar o mapeamento de processos de *workflow* em WorkFlow net, considera-se o processo de tratamento de reclamações apresentado em Aalst e Hee (2004):

An incoming complaint first is recorded. Then the client who has complained and the department affected by the complaint are contacted. The client is approached for more information. The department is informed of the complaint and may be asked for its initial reaction. These two tasks may be performed in parallel – that is, simultaneously or in any order. After this, the data are gathered and a decision is taken. Depending upon the decision, either a compensation payment is made or a letter is sent. Finally, the complaint is filed (AALST; HEE, 2004).

A Figura 4 mostra uma WorkFlow net, com erro de modelagem, para este processo.

De acordo com Aalst e Hee (2004), as seguintes construções básicas para o roteamento de tarefas devem ser consideradas:

- sequencial: a forma mais simples de execução de tarefas, onde uma tarefa é executada após a outra, havendo, claramente, dependência entre elas;
- paralela: mais de uma tarefa pode ser executada simultaneamente, ou em qualquer ordem;
- condicional (ou rota seletiva): quando há uma escolha entre duas ou mais tarefas;
- iterativa: quando é necessário executar uma mesma tarefa (ou conjunto de tarefas) múltiplas vezes.

Considerando o processo de tratamento de reclamações mostrado na Figura 4, as tarefas *Contact_Client* e *Contact_Department* são um exemplo de roteamento paralelo. As tarefas *Collect* e *Assess* são um exemplo de roteamento sequencial. Já as tarefas *Pay* e *Send_Letter* são um exemplo de roteamento condicional. Conforme apresentado

por Passos e Julia (2009), um roteamento iterativo pode ser substituído por uma tarefa global, que considera a definição dos blocos bem formados, apresentada por Valette (1979). Assim, uma rota iterativa será substituída por uma tarefa global, como mostra a Figura 5.

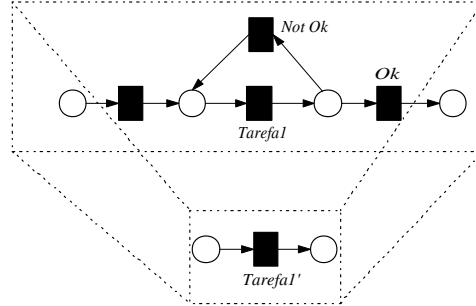


Figura 5 – Substituição de um roteiro iterativo por uma tarefa global (PASSOS; JULIA, 2009).

Além dos roteamentos, Aalst e Hee (2004) definem a noção de acionamento, sendo um acionamento uma condição externa que guia a execução de uma tarefa sensibilizada. De acordo com Aalst e Hee (2004), há quatro tipos distintos de acionamento: usuário, mensagem, tempo e automático. No entanto, somente nos acionamentos do tipo *usuário* (este tipo de acionamento é identificado através do símbolo \square associado às transições), onde uma tarefa é acionada por um recurso (humano, uma máquina, uma equipe etc.), há a necessidade de alocação de recurso para o tratamento da tarefa. Nos demais tipos de acionamento, não há alocação de recursos associada. Na Figura 4, as tarefas *Contact_Client*, *Contact_Department*, *Assess*, *Pay* e *Send_Letter* são tarefas do tipo usuário.

Soundness é o principal critério de correção definido para a WorkFlow net. Uma WorkFlow net é *Sound* se, e somente se, os três requisitos a seguir são satisfeitos (AALST; HEE, 2004):

- para cada ficha colocada no lugar de início i , uma (e apenas uma) aparece no lugar de término o ;
- quando uma ficha aparece no lugar o , todos os outros lugares estão vazios, considerando o caso em questão;
- considerando uma tarefa associada a uma transição, é sempre possível evoluir da marcação inicial até uma marcação que sensibiliza tal transição, ou seja, não deve haver nenhuma transição morta na WorkFlow net.

A definição formal do critério de correção *Soundness* no contexto da WorkFlow net, proposto por Aalst (1998a), é apresentada na sequência.

Definição 5. (*Soundness*) Um processo de *workflow* modelado por uma WorkFlow net $PN = (P, T, F)$ é *Sound* se, e somente se:

- para cada marcação M alcançável a partir da marcação i , existe uma sequência de disparo que leva da marcação M para a marcação o . Formalmente:

$$\forall_M(i \xrightarrow{*} M) \Rightarrow (M \xrightarrow{*} o);$$

- b) a marcação o é o único estado alcançável a partir da marcação i com exatamente uma ficha no lugar o . Formalmente:

$$\forall_M(i \xrightarrow{*} M \wedge M \geq o) \Rightarrow (M = o);$$

- c) não há nenhuma transição morta em (PN, i) . Formalmente:

$$\forall_{t \in T} \exists_{M, M'} i \xrightarrow{*} M \xrightarrow{t} M'.$$

A WorkFlow net mostrada na Figura 4 não é *Sound*, pois considerando por exemplo a sequência de disparo $\sigma_1 = \text{Record}, \text{Contact_Client}, \text{Contact_Department}, \text{Collect}, \text{Assess}, \text{Pay}, \text{File}$, quando uma ficha atinge o lugar de término o , ainda há uma ficha remanescente no lugar $c3$.

Dehnert e Rittgen (2001) propõem o relaxamento do critério de correção *Soundness*, definindo um novo critério denominado *Relaxed Soundness*. Os autores argumentam que esse novo critério é mais próximo da intuição do modelador. De acordo com Dehnert e Rittgen (2001), o critério *Relaxed Soundness* pretende representar uma visão mais pragmática no contexto da correção, sendo mais fraco, em um sentido formal, que o critério de correção *Soundness*. Para Dehnert e Rittgen (2001), o critério de correção *Relaxed Soundness* significa que existe um número suficiente de execuções do processo de *workflow* que finalizam corretamente, isto é, sem fichas remanescentes. Neste contexto, suficiente significa, de acordo com Dehnert e Rittgen (2001), que cada transição do processo de *workflow* é disparada pelo menos uma vez, quando tomado em conta o conjunto de sequências de disparo que finalizam corretamente. O critério de correção *Relaxed Soundness*, proposto por Dehnert e Rittgen (2001), é apresentado na sequência.

Definição 6. (*Relaxed Soundness*) Um processo de *workflow* modelado por uma WorkFlow net $PN = (P, T, F)$ é *Relaxed Sound* se, e somente se, cada transição está em uma sequência de disparo que inicia na marcação i e finaliza na marcação o .

Formalmente:

$$\forall t \in T : \exists M, M' : (i \xrightarrow{*} M \xrightarrow{t} M' \xrightarrow{*} o).$$

A WorkFlow net mostrada na Figura 4 não é *Relaxed Sound*, pois, considerando todas as transições $t \in T$, não existe sequência de disparo que inicia na marcação i e finaliza na marcação o , que atenda a condição acima. Por exemplo, considerando as sequências de disparo $\sigma_1 = \text{Record}, \text{Contact_Client}, \text{Contact_Department}, \text{Collect}, \text{Assess}, \text{Pay}, \text{File}$ e $\sigma_2 = \text{Record}, \text{Contact_Client}, \text{Contact_Department}, \text{Collect}, \text{Assess}, \text{Send_Letter}, \text{File}$, quando o lugar o é alcançado, há ainda uma ficha no lugar $c3$, para ambas as sequências.

Martens (2003) propõe uma outra variante do critério de correção *Soundness*. O novo critério definido é denominado *Weak Soundness*. A ideia por trás do critério *Weak Soundness* é que um processo pode ser considerado mesmo quando este contempla tarefas mortas. Segundo Martens (2003), o fato de não haver tarefas mortas em um processo de *workflow* é razoável quando um processo é modelado do zero. No entanto, quando um processo decorre da composição de outros processos, é razoável considerar o sistema viável, apesar de nem todas as funcionalidades de um componente específico serem consideradas no sistema composto (MARTENS, 2005). O critério de correção *Weak Soundness*, proposto por (MARTENS, 2003), é apresentado na sequência.

Definição 7. (*Weak Soundness*) Um processo de *workflow* modelado por uma Work-Flow net $PN = (P, T, F)$ é *Weak Sound* se:

- a) para cada marcação M alcançável: a marcação final o é alcançável. Formalmente:

$$\forall_M (i \xrightarrow{*} M) \Rightarrow (M \xrightarrow{*} o);$$

- b) Para cada marcação M alcançável, com $M \geq o$: $M = o$.

Formalmente:

$$\forall_M (i \xrightarrow{*} M \wedge M \geq o) \Rightarrow (M = o).$$

A WorkFlow net mostrada na Figura 4 não é *Weak Sound*, pois não existe sequência de disparo que inicia na marcação i e finaliza na marcação o . Por exemplo, considerando as sequências de disparo $\sigma_1 = \text{Record, Contact_Client, Contact_Department, Collect, Assess, Pay, File}$ e $\sigma_2 = \text{Record, Contact_Client, Contact_Department, Collect, Assess, Send_Letter, File}$, tem-se que quando o lugar o é alcançado, há ainda uma ficha no lugar $c3$, para ambas as sequências.

Puhlmann (2007) apresenta uma comparação estruturada entre os critérios de correção *Soundness*, *Relaxed Soundness* e *Weak Soundness*, no que diz respeito à correta finalização do processo, à ausência de *deadlock* no processo e à participação de todas as atividades do processo, ou seja, a ausência de tarefas mortas. A Tabela 1 mostra esta comparação.

Tabela 1 – Comparação entre os diferentes tipos de *Soundness* (adaptado de (PUHLMANN, 2007)).

Característica	<i>Soundness</i>	<i>Relaxed Soundness</i>	<i>Weak Soundness</i>
Correta Finalização	Sim	Sim	Sim
Ausência de <i>Deadlock</i>	Sim	Não	Sim
Participação de todas as tarefas	Sim	Sim	Não

2.1.3 WorkFlow net Interorganizacional

Uma WorkFlow net Interorganizacional, do inglês *IOWF-net*, é uma rede de Petri que modela um processo de *workflow* interorganizacional. Um processo de *workflow* interorganizacional é um processo de *workflow* global que tem n parceiros de negócio envolvidos (AALST, 1998b). Cada parceiro tem seu próprio processo de *workflow* local. Assim, um processo de *workflow* interorganizacional é composto por pelo menos dois processos de *workflow* locais. Desta forma, de acordo com Aalst (1998b), uma IOWF-net é composta por, pelo menos, dois modelos, isto é, WorkFlow nets locais, do inglês *LWF-nets*. Em Aalst (1998b), o processo de *workflow* global consiste de processos de *workflow* locais e uma estrutura de interação: comunicação síncrona e assíncrona, de forma que a comunicação síncrona corresponde à fusão de um número de transições. Na presente pesquisa tais comunicações síncronas não são consideradas, uma vez que considera-se que cada parceiro envolvido controla seu próprio processo. Assim, somente as comunicações assíncronas, que correspondem a troca de mensagens entre os processos de *workflow* locais, serão consideradas. Na sequência, a WorkFlow net interorganizacional considerada neste trabalho é definida. É relevante destacar que está é uma adaptação da definição apresentada por Aalst (1998b).

Definição 8. (IOWF-net) Uma WorkFlow net Interorganizacional (IOWF-net) é uma tupla $IOWF - net = \{PN_1, PN_2, \dots, PN_n, P_{AC}, AC\}$, onde:

- a) $n \in \mathbb{N}$ é o número de LWF-nets;
- b) para cada $k \in \{1, \dots, n\}$: PN_k é uma WorkFlow net com lugar de início i_k e lugar de término o_k ;
- c) para cada $k, l \in \{1, \dots, n\}$: se $k \neq l$, então $(P_k \cup T_k) \cap (P_l \cup T_l) = \emptyset$;
- d) $T^* = \bigcup_{k \in \{1, \dots, n\}} T_k$, $P^* = \bigcup_{k \in \{1, \dots, n\}} P_k$, $F^* = \bigcup_{k \in \{1, \dots, n\}} F_k$ (relações entre os elementos das LWF-nets);
- e) P_{AC} é o conjunto de elementos de comunicação assíncrona (lugares de comunicação);
- f) $AC \subseteq P_{AC} \times \mathbb{P}(T^*) \times \mathbb{P}(T^*)$ é a relação de comunicação assíncrona¹.

Cada elemento de comunicação assíncrona corresponde a um lugar em P_{AC} e a relação AC especifica um conjunto de transições de entrada e um conjunto de transições de saída para cada elemento de comunicação assíncrona (AALST, 1998b). Assim, considerando uma transição t , um lugar de comunicação assíncrona p pode ser visto como um lugar de comunicação de entrada (do inglês *ICP*) ou um lugar de comunicação de saída (do inglês *OCP*) da transição t .

Para ilustrar os conceitos definidos acima, considere o processo de *workflow* interorganizacional apresentado em Aalst (1998b), que modela um processo que precede a apresentação de um artigo em uma conferência:

¹ $\mathbb{P}(T^*)$ é o conjunto de todos os subconjuntos não vazios de T^* .

This workflow has two loosely coupled workflow processes: (1) the process of an author preparing, submitting and revising a paper, and (2) the process of evaluating and monitoring submissions by the program committee. In this case there are two ‘organizations’ involved in the interorganizational workflow: the author (AU) and the program committee (PC). The author sends a draft version of the paper to the program committee. The program committee acknowledges the receipt and evaluates the submission. The paper is accepted or rejected by the program committee. In both cases the author is notified. If the paper is rejected, the workflow terminates, otherwise the author can start preparing the final version. After completing the final version, a copy is sent to the program committee and the program committee acknowledges the receipt of the final version. If the final version is not received by the program committee by a specified due date, the author is notified that the paper is considered to be too late. A paper which is too late will not be published in the proceedings (AALST, 1998b).

A Figura 6 mostra a IOWF-net que modela este processo. Esta IOWF-net possui duas LWF-nets: *Author* e *PC*. Cada uma destas LWF-nets possui um lugar de início e um lugar de término. No caso da LWF-net *Author*, o lugar de início é *start_flow_author* e o lugar de término é *end_flow_author*. Na LWF-net *PC*, o lugar de início e término são *start_flow_PC* e *end_flow_PC*, respectivamente. Os lugares *draft*, *ack_draft*, *accept*, *reject*, *too_late*, *final_version* e *ack_final* são os lugares de comunicação. O lugar de comunicação assíncrona *draft* é o ICP da transição *receive_draft* e o OCP da transição *send_draft*.

No contexto dos processos de *workflow* interorganizacionais, também é desejável que o critério de correção *Soundness* seja verificado. Conforme mostra Aalst (1998b), o fato de cada LWF-net ser *Sound* não garante que o modelo global será *Sound*. Também é possível ter um processo de *workflow* interorganizacional globalmente *Sound* mas não localmente *Sound*, como mostra Aalst (1998b). Com o objetivo de solucionar estes problemas, Aalst (1998b) define que a noção global do critério de correção *Soundness* é baseada na representação de uma IOWF-net através de uma WorkFlow net. Assim, Aalst (1998b) define a *unfolded* net, isto é, a $U(IOWF-net)$. Em uma $U(IOWF-net)$, todas as LWF-nets são inseridas em um processo de *workflow* simples, considerando uma transição de início t_i e uma transição de término t_o (AALST, 1998b). Um lugar de início global i e um lugar de término global o precisam ser adicionados para respeitar a estrutura básica de uma WorkFlow net simples e os elementos de comunicação assíncrona são mapeados em lugares ordinários (AALST, 1998b). A $U(IOWF-net)$ para a IOWF-net da Figura 6 é mostrada na Figura 7.

O critério de correção *Soundness* no contexto das IOWF-nets é dado pela Definição 9.

Definição 9. (*Soundness* para IOWF-nets) Uma WorkFlow net Interorganizacional (IOWF-net) é *Sound* se, e somente se, for localmente *Sound* e globalmente *Sound*. Uma IOWF-net é localmente *Sound* se, e somente se, cada uma das suas WorkFlow nets locais

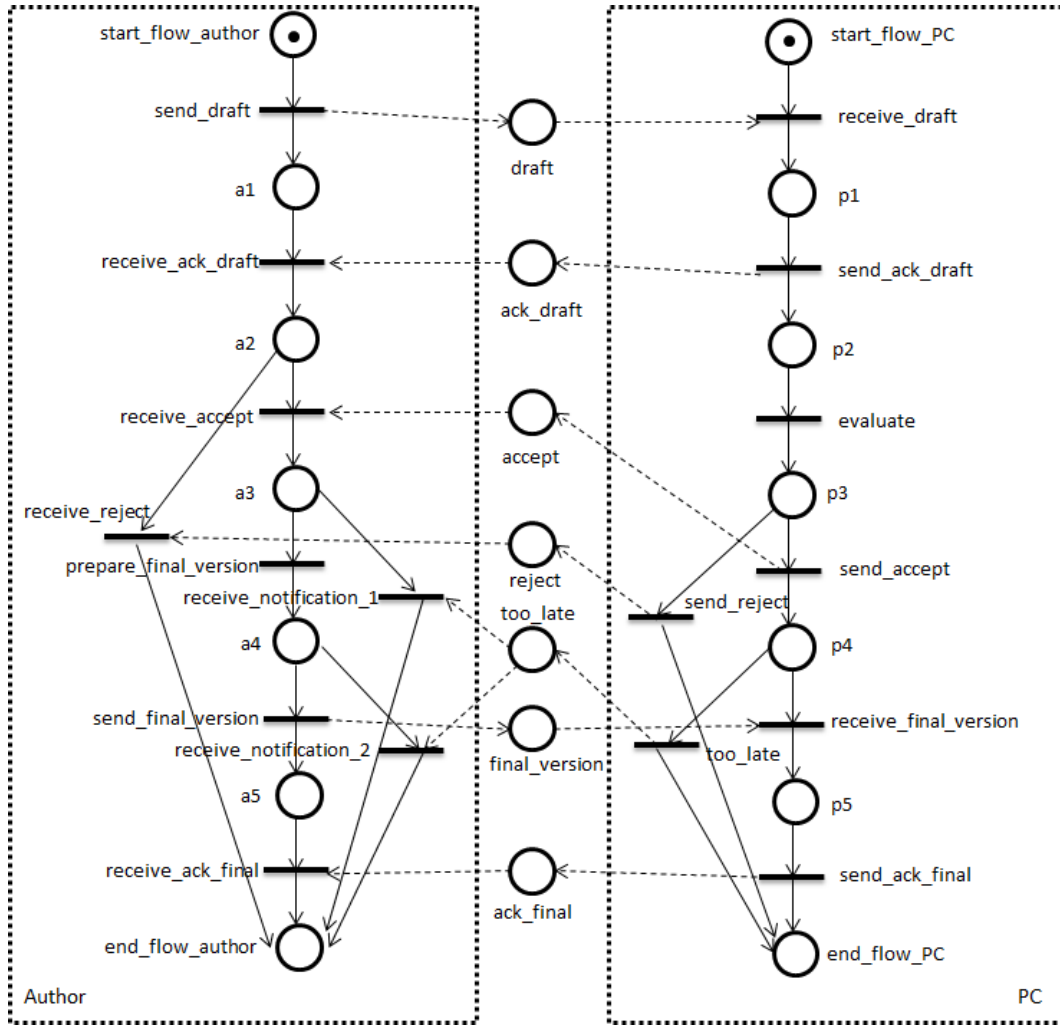


Figura 6 – Um exemplo de uma WorkFlow net Interorganizacional (AALST, 1998b).

(LWF-nets) PN_k é *Sound*. Uma IOWF-net é globalmente *Sound* se, e somente se, a $U(IOWF-net)$ é *Sound*.

Considerando a Figura 6, as LWF-nets *Author* e *PC* são ambas *Sound*. Entretanto, a $U(IOWF-net)$ não é *Sound*. Portanto, a IOWF-net não satisfaz o critério de correção *Soundness*. É fácil observar que se a transição *too_late* da LWF-net *PC* é disparada e as transições *prepare_final_version* e *send_final_version* da LWF-net *Author* são disparadas, as mensagens *too_late* e *final_version* se entrelaçam, levando a um estado de *deadlock* com uma ficha no lugar *a5* e as duas mensagens que nunca serão recebidas (uma ficha no lugar *too_late* e uma ficha no lugar *final_version*) (AALST, 1998b).

2.1.4 Rede de Petri e Lógica Linear

A Lógica Linear foi proposta por Girard (1987). Na Lógica Linear, as proposições são consideradas como recursos, representados por átomos, que são consumidos ou produzidos

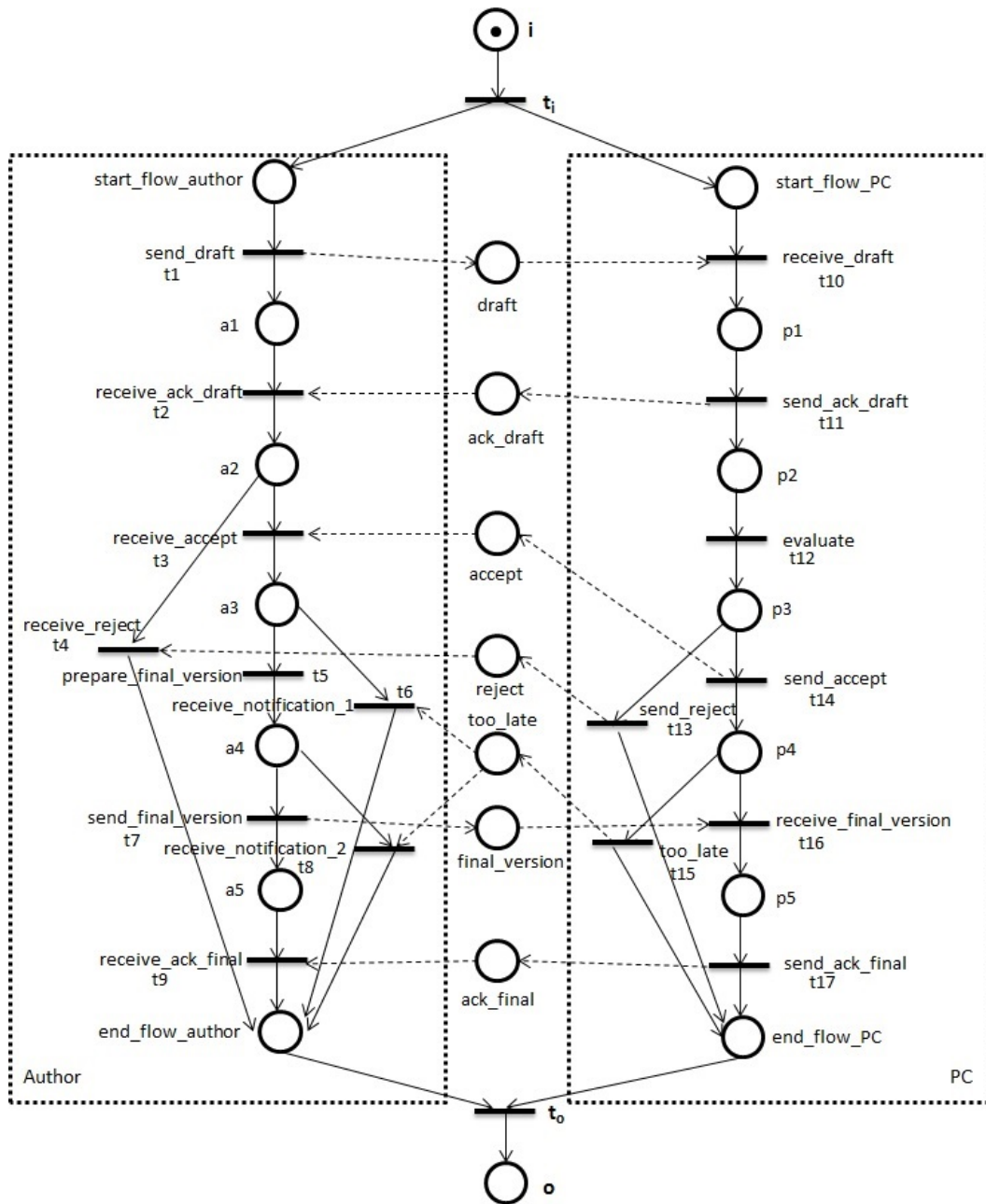


Figura 7 – A $U(IOWF-net)$ para a IOWF-net mostrada na Figura 6.

a cada mudança de estado (PRADIN-CHÉZALVIEL; VALETTE; KUNZLE, 1999). Já na Lógica Clássica, as proposições podem tomar somente valores *verdadeiro* ou *falso*.

A Lógica Linear introduz novos conectivos, como os conectivos “*par*”(\wp), “*times*”(\otimes), “*with*”($\&$), “*plus*”(\oplus), *implicação linear*(\multimap), “*of course*”(!) e “*why not*”(?) (GIRARD, 1987).

No contexto deste trabalho, apenas dois conectivos da Lógica Linear são considerados:

- a) o conectivo *times*, denotado pelo símbolo \otimes , representa a disponibilidade simultânea de recursos. Por exemplo, $A \otimes B$ representa a disponibilidade simultânea dos recursos A e B (PRADIN-CHÉZALVIEL; VALETTE; KUNZLE, 1999);
- b) o conectivo *implicação linear*, denotado pelo símbolo \multimap , representa uma mudança de estado. Por exemplo, $A \multimap B$ denota que consumindo A, B é produzido. Deve-se notar que após a produção de B, A não estará mais disponível (PRADIN-CHÉZALVIEL; VALETTE; KUNZLE, 1999).

Apesar de modelos em rede de Petri poderem ser reescritos utilizando Lógica Linear, as duas teorias não são equivalentes, sendo que cada uma tem suas vantagens específicas (CHAMPAGNAT; PRADIN-CHÉZALVIEL; VALETTE, 2000). A tradução de um modelo em rede de Petri em fórmulas da Lógica Linear é apresentada por Pradin-Chézalviel, Valette e Kunzle (1999) e dada pelas formalizações mostradas na sequência.

Uma marcação M é um monômio em \otimes , ou seja, uma marcação é representada por $M = A_1 \otimes A_2 \otimes \dots \otimes A_k$ onde A_i são nomes de lugares: se algum lugar A_j contém várias fichas (m , por exemplo), m instâncias do átomo A_j devem aparecer no monômio. Por exemplo, considerando as redes de Petri das Figuras 8(a), 8(c) e 8(d), tem-se a marcação inicial $M = P1$. Já a marcação inicial da rede de Petri da Figura 8(b) é dada por $M = P1 \otimes P2$.

Uma transição é uma expressão da forma $M_1 \multimap M_2$ onde M_1 e M_2 são marcações. Por exemplo, considerando a transição $t1$ da rede de Petri mostrada na Figura 8(a), $t1 = P1 \multimap P2$. Para a transição $t1$ da rede de Petri da Figura 8(b), $t1 = P1 \otimes P2 \multimap P3$. Já para a transição $t1$ da rede de Petri da Figura 8(c), $t1 = P1 \multimap P2 \otimes P3$. Considerando a rede de Petri da Figura 8(d), $t1 = P1 \multimap P2$ e $t2 = P1 \multimap P3$.

Um sequente linear $M, t_k \vdash M'$ representa um cenário onde M e M' são, respectivamente, a marcação inicial e final e t_k é uma lista de transições não ordenadas. Por exemplo, considerando a rede de Petri da Figura 8(a) e sua marcação final $P2$, tem-se o seguinte sequente linear $P1, P1 \multimap P2 \vdash P2$. Para a rede de Petri da Figura 8(b), considerando que sua marcação final seja $P3$, tem-se o sequente $P1 \otimes P2, P1 \otimes P2 \multimap P3 \vdash P3$. Considerando a rede de Petri da Figura 8(c) e sua marcação final $P2 \otimes P3$, tem-se o sequente $P1, P1 \multimap P2 \otimes P3 \vdash P2 \otimes P3$. Para a rede de Petri da Figura 8(d), há dois possíveis cenários: o primeiro cenário, sc_1 , considera o disparo da transição $t1$; o segundo cenário, sc_2 , considera o disparo da transição $t2$. Para o cenário sc_1 tem-se o sequente linear $P1, P1 \multimap P2 \vdash P2$ e, para o cenário sc_2 , o sequente linear é dado por

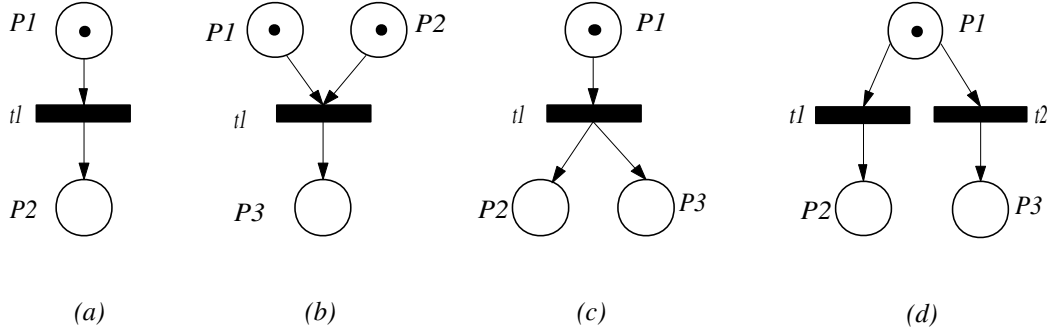


Figura 8 – Redes de Petri para a exemplificação da tradução de redes de Petri em fórmulas da Lógica Linear.

$P1, P1 \multimap P3 \vdash P3$.

É importante destacar que o sistema de dedução linear é similar ao sistema de dedução de Gentzen, proposto em 1934 (GOCHET; GRIBOMONT, 1990). Um seqüente linear tem a forma $\Gamma \vdash \Delta$, onde Γ e Δ são conjuntos finitos de fórmulas, isto é, $\Gamma = \Gamma_1, \Gamma_2, \dots, \Gamma_n$ e $\Delta = \Delta_1, \Delta_2, \dots, \Delta_n$. O símbolo Γ é o antecedente da fórmula e o símbolo Δ o conseqüente.

De acordo com Girault, Pradin-Chézalviel e Valette (1997), um seqüente pode ser provado através de aplicações sucessivas de regras do cálculo de seqüentes e há equivalência entre a prova de seqüentes da Lógica Linear e o problema de alcançabilidade em uma rede de Petri.

Uma árvore de prova da Lógica Linear é construída para provar se um dado seqüente linear é ou não sintaticamente correto (DIAZ, 2010). A árvore de prova é lida de baixo para cima (bottom-up) e termina quando todas as folhas forem seqüentes identidade (seqüentes do tipo $A \vdash A$), nos casos em que o seqüente for sintaticamente correto (DIAZ, 2010).

O presente trabalho considera apenas algumas regras da Lógica Linear. Assim, somente estas regras serão explicadas e terão suas aplicações exemplificadas. As demais regras do cálculo de seqüentes da Lógica Linear podem ser encontradas em Girard (1987) e em Girard (1995). Inicialmente, considere que F , G e H são fórmulas e que Γ e Δ são blocos de fórmulas da Lógica Linear. As regras a seguir, apresentadas em Riviere, Pradin-Chézalviel e Valette (2001), são as regras utilizadas nos métodos propostos nesta pesquisa:

- a) a regra \multimap_L , dada por $\frac{\Gamma \vdash F \quad \Delta, G \vdash H}{\Gamma, \Delta, F \multimap G \vdash H} \multimap_L$, expressa o disparo de uma transição e gera dois seqüentes, tal que o seqüente à direita representa o subsequente restante a ser provado e o seqüente à esquerda representa as fichas consumidas pelo disparo da transição. Por exemplo, considerando o disparo da transição $t_1 = P1 \multimap P2 \otimes P3$ da rede de Petri mostrada na Figura 9, dois seqüentes são gerados: $P1 \vdash P1$, representando as fichas consumidas por esse disparo, e o subsequente remanescente, do qual a marcação $P2 \otimes P3$ fará parte, ou seja: $\frac{P1 \vdash P1 \quad P2 \otimes P3, P2 \multimap P4, t_3, t_4, t_5 \vdash P7}{P1, P1 \multimap P2 \otimes P3, t_2, t_3, t_4, t_5 \vdash P7} \multimap_L$;

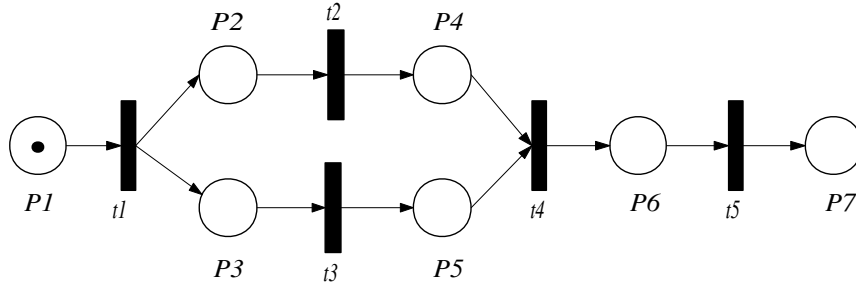


Figura 9 – Rede de Petri para exemplificação da construção de uma árvore de prova canônica da Lógica Linear.

- b) a regra \otimes_L , dada por $\frac{\Gamma, F, G \vdash H}{\Gamma, F \otimes G \vdash H} \otimes_L$, transforma uma marcação em uma lista de átomos. Por exemplo, a marcação $P2 \otimes P3$ gerada pelo disparo da transição $t_1 = P1 \multimap P2 \otimes P3$ da rede de Petri mostrada na Figura 9 utilizará a regra \otimes_L para ser transformada em uma lista de átomos $P2, P3$, ou seja: $\frac{P2, P3, t_2, t_3, t_4, t_5 \vdash P7}{P2 \otimes P3, t_2, t_3, t_4, t_5 \vdash P7} \otimes_L$;
- c) a regra \otimes_R , dada por $\frac{\Gamma \vdash F \quad \Delta \vdash G}{\Delta, \Gamma \vdash F \otimes G} \otimes_R$, transforma um sequente do tipo $A, B \vdash A \otimes B$ em dois sequentes identidade $A \vdash A$ e $B \vdash B$. Por exemplo, considerando o disparo da transição $t_4 = P4 \otimes P5 \multimap P6$ da rede de Petri da Figura 9, o sequente que representa as fichas consumidas pelo disparo desta transição, $P4, P5 \vdash P4 \otimes P5$, deverá ser provado utilizando a regra \otimes_R , ou seja: $\frac{P4 \vdash P4 \quad P5 \vdash P5}{P4, P5 \vdash P4 \otimes P5} \otimes_R$.

Para a exemplificação da construção de uma árvore de prova canônica da Lógica Linear, considere a rede de Petri da Figura 9. Transformando as transições desta rede de Petri em fórmulas da Lógica Linear, tem-se:

$$\begin{aligned} t_1 &= P1 \multimap P2 \otimes P3, \\ t_2 &= P2 \multimap P4, \\ t_3 &= P3 \multimap P5, \\ t_4 &= P4 \otimes P5 \multimap P6 \text{ e} \\ t_5 &= P6 \multimap P7. \end{aligned}$$

A marcação inicial desta rede de Petri é $M = P1$. Assim, o sequente linear a ser provado é dado por $P1, t_1, t_2, t_3, t_4, t_5 \vdash P7$, onde $P1$ e $P7$ são, respectivamente, a marcação inicial e final da rede de Petri da Figura 9. Aplicando as regras do cálculo de sequentes a este sequente linear, é possível provar se o mesmo é ou não um sequente sintaticamente válido. A árvore de prova é mostrada na sequência.

$$\begin{array}{c}
\frac{P6 \vdash P6 \quad P7 \vdash P7}{\vdash \multimap_L} \\
\frac{\frac{P4 \vdash P4 \quad P5 \vdash P5}{P4, P5 \vdash P4 \otimes P5} \otimes_R \quad P6, P6 \multimap P7 \vdash P7}{\vdash \multimap_L} \\
\frac{P3 \vdash P3 \quad P4, P5, P4 \otimes P5 \multimap P6, t_5 \vdash P7}{\vdash \multimap_L} \\
\frac{P2 \vdash P2 \quad P3, P4, P3 \multimap P5, t_4, t_5 \vdash P7}{\vdash \multimap_L} \\
\frac{P2, P3, P2 \multimap P4, t_3, t_4, t_5 \vdash P7}{\vdash \otimes_L} \\
\frac{P1 \vdash P1 \quad P2 \otimes P3, P2 \multimap P4, t_3, t_4, t_5 \vdash P7}{\vdash \multimap_L} \\
P1, P1 \multimap P2 \otimes P3, t_2, t_3, t_4, t_5 \vdash P7
\end{array}$$

Uma vez que uma árvore de prova da Lógica Linear é construída de forma *bottom-up*, a raiz da árvore de prova corresponde à primeira linha da árvore de prova, neste caso $P1, P1 \multimap P2 \otimes P3, t_2, t_3, t_4, t_5 \vdash P7$, que é o sequente linear a ser provado. A primeira regra a ser aplicada é a regra \multimap_L . Como mencionado anteriormente, a regra \multimap_L expressa o disparo de uma transição, neste caso a transição t_1 , e gera dois sequentes, tais que o sequente à esquerda, i.e. $P1 \vdash P1$, representa as fichas consumidas pelo disparo da transição t_1 e o sequente à direita, i.e. $P2 \otimes P3, P2 \multimap P4, t_3, t_4, t_5 \vdash P7$ representa o subsequente restante a ser provado. O sequente identidade, $P1 \vdash P1$, à esquerda é um nó folha da árvore de prova. Deve-se notar que os átomos $P2 \otimes P3$ no sequente à direita representam os átomos produzidos pelo disparo da transição t_1 . A regra \otimes_L deve ser então aplicada para transformar tal marcação em uma lista de átomos. Assim, após a aplicação da regra \otimes_L , tem-se o sequente $P2, P3, P2 \multimap P4, t_3, t_4, t_5 \vdash P7$. Para provar tal sequente, mais uma vez a regra \multimap_L deve ser aplicada. A aplicação desta regra representa o disparo da transição t_2 . Novamente são gerados dois sequentes, tais que o sequente à esquerda é o nó folha $P2 \vdash P2$, que representa o átomo consumido pelo disparo de t_2 e o sequente à direita é dado por $P3, P4, P3 \multimap P5, t_4, t_5 \vdash P7$, onde: $P3$ é átomo disponível no sequente anterior e não utilizado no disparo de t_2 ; $P4$ é o átomo produzido pelo disparo de t_2 ; t_3, t_4 e t_5 representam as fórmulas para as referidas transições e, finalmente, $P7$ representa a marcação a ser atendida. Para a prova de $P3, P4, P3 \multimap P5, t_4, t_5 \vdash P7$, mais uma vez a regra \multimap_L deve ser aplicada. A aplicação desta regra, desta vez, representa o disparo da transição t_3 . Novamente, dois sequentes são gerados: $P3 \vdash P3$ e $P4, P5, P4 \otimes P5 \multimap P6, t_5 \vdash P7$. O primeiro sequente corresponde ao átomo consumido para disparo de t_3 e o segundo corresponde ao subsequente restante a ser provado, onde $P5$ representa o átomo produzido pelo disparo de t_3 . Mais uma vez a regra \multimap_L deve ser aplicada. Desta vez, a aplicação desta regra representa o disparo de t_4 . Note que, a fórmula para a transição t_4 é $P4 \otimes P5 \multimap P6$, ou seja, deve-se consumir os átomos $P4$ e $P5$ para produzir o átomo $P6$. Assim, após a aplicação da regra \multimap_L , tem-se dois sequentes a serem provados: $P4, P5 \vdash P4 \otimes P5$ e $P6, P6 \multimap P7 \vdash P7$. O sequente $P4, P5 \vdash P4 \otimes P5$ representa os átomos consumidos no disparo de t_4 e para provar tal sequente, deve-se utilizar a regra \otimes_R . A regra \otimes_R transforma o sequente

$P4, P5 \vdash P4 \otimes P5$ em dois sequentes identidade e, consequentemente, nós fólha da árvore de prova: $P4 \vdash P4$ e $P5 \vdash P5$. Deve-se provar então o sequeute $P6, P6 \multimap P7 \vdash P7$. Para provar tal sequeute, deve-se aplicar a regra \multimap_L que, nesse caso, representa o disparo da transição t_5 . A aplicação da regra \multimap_L gera dois sequentes identidade e nós folha da árvore de prova: $P6 \vdash P6$ e $P7 \vdash P7$. Como todos os nós folha da árvore de prova apresentada e detalhada são sequentes identidade, tem-se que o sequeute linear $P1, t_1, t_2, t_3, t_4, t_5 \vdash P7$ é sintaticamente correto.

Uma vez que as árvores de prova da Lógica Linear não possuem fácil leitura, elas podem ser transformadas em um grafo de precedência associado, como é mostrado em Diaz (2010). O grafo de precedência apresenta informações que concernem as relações de causalidade e a precedência entre eventos do sequeute representado. Em tais grafos, as situações de entrelaçamento, comuns nas abordagens baseadas em grafos da alcançabilidade quando roteamentos paralelos são considerados, não são representadas. O grafo de precedência é então obtido através da rotulação das árvores de prova originais. Como é apresentado em Diaz (2010), para rotular uma árvore de prova, cada vez que a regra \multimap_L é aplicada, a transição t_i correspondente rotula a aplicação da regra, bem como os átomos produzidos e consumidos pela mesma. Além disso, os eventos iniciais devem ser rotulados por i_i e os eventos finais são rotulados por f_i . Uma vez que a rotulação foi realizada, cada sequeute identidade representa a associação de duas visões do mesmo átomo: a parte da esquerda é rotulada pelo evento que o produziu e a parte da direita é rotulada pelo evento que o consumiu (DIAZ, 2010). Os rótulos são mostrados acima dos átomos e abaixo das regras \multimap_L .

A árvore de prova rotulada para o sequeute $P1, t_1, t_2, t_3, t_4, t_5 \vdash P7$ é apresentada na sequência.

$$\begin{array}{c}
\frac{\frac{t_4 \quad t_5}{P6 \vdash P6} \quad \frac{t_5 \quad f_i}{P7 \vdash P7}}{\multimap_L} t_5 \\
\frac{\frac{\frac{t_2 \quad t_4}{P4 \vdash P4} \quad \frac{t_3 \quad t_4}{P5 \vdash P5}}{P4, P5 \vdash P4 \otimes P5} \otimes_R \quad \frac{t_4}{P6, P6 \multimap P7 \vdash P7}}{\multimap_L} t_4 \\
\frac{\frac{t_1 \quad t_3}{P3 \vdash P3} \quad \frac{t_2 \quad t_3}{P4, P5, P4 \otimes P5 \multimap P6, t_5 \vdash P7}}{\multimap_L} t_3 \\
\frac{\frac{t_1 \quad t_2}{P2 \vdash P2} \quad \frac{t_1 \quad t_2}{P3, P4, P3 \multimap P5, t_4, t_5 \vdash P7}}{\multimap_L} t_2 \\
\frac{\frac{t_1 \quad t_1}{P2, P3, P2 \multimap P4, t_3, t_4, t_5 \vdash P7} \otimes_L \quad \frac{i_i \quad t_1}{P1 \vdash P1} \quad \frac{t_1 \quad t_1}{P2 \otimes P3, P2 \multimap P4, t_3, t_4, t_5 \vdash P7}}{\multimap_L} t_1 \\
\frac{i_i}{P1, P1 \multimap P2 \otimes P3, t_2, t_3, t_4, t_5 \vdash P7}
\end{array}$$

Nos grafos de precedência, os nós são eventos (disparo de transições, eventos iniciais e eventos finais) e os arcos são sequentes identidade (relação entre o evento que produziu o átomo e aquele que o consumiu) (DIAZ, 2010).

O grafo de precedência associado à árvore de prova rotulada para o seqüente $P1, t_1, t_2, t_3, t_4, t_5 \vdash P7$ é apresentado na Figura 10.

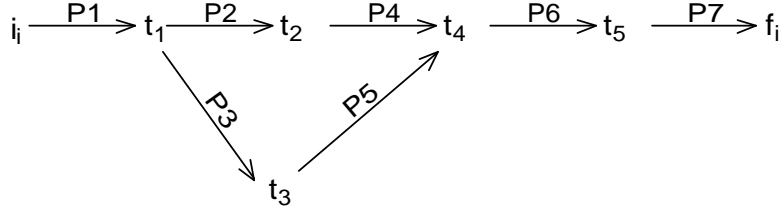


Figura 10 – Grafo de precedência para a árvore de prova rotulada para o seqüente $P1, t_1, t_2, t_3, t_4, t_5 \vdash P7$.

No contexto das redes de Petri t-temporais, em uma árvore de prova da Lógica Linear, cada disparo de transição pode gerar uma data simbólica associada a cada átomo (token), como mostram Riviere, Pradin-Chézalviel e Valette (2001).

No presente trabalho, D_i denota uma data e d_i uma duração da sensibilização associada a um disparo de uma transição (t_i). Um par (D_p, D_c) será associado a cada átomo da árvore de prova, tal que D_p e D_c representam, respectivamente, as datas de produção e consumo de um átomo. O cálculo de datas em árvores de prova canônica é dado pelos seguintes passos (RIVIERE; PRADIN-CHÉZALVIEL; VALETTE, 2001):

- determinar uma data de produção D_i para todas as fichas da marcação inicial;
- para cada instância da regra \multimap_L , calcule a data de disparo desta transição: isto é igual ao maior valor de data de produção dos átomos consumidos por esta transição, acrescido pela duração da sensibilização d_i associada à transição considerada;
- atualizar as datas de todos os átomos que foram consumidos e produzidos.

Por exemplo, considerando a rede de Petri t-temporal da Figura 3, as fórmulas da Lógica Linear para cada uma de suas transições (definidas anteriormente), sua marcação inicial $P1$ e que $Seq = D_1 + d_1 + \max(d_2, d_3) + d_4$, tem-se a seguinte árvore de prova com cálculo de datas:

$$\begin{array}{c}
 \frac{P6(Seq, Seq+d_5) \vdash P6 \quad P7(Seq+d_5, \cdot) \vdash P7}{\multimap_L} \\
 \frac{\frac{P4(D_1+d_1+d_2, Seq) \vdash P4 \quad P5(D_1+d_1+d_3, Seq) \vdash P5}{P4(D_1+d_1+d_2, Seq), P5(D_1+d_1+d_3, Seq) \vdash P4 \otimes P5} \otimes_R \quad P6(Seq, \cdot), P6 \multimap P7 \vdash P7}{\multimap_L} \\
 \frac{P3(D_1+d_1, D_1+d_1+d_3) \vdash P3 \quad P4(D_1+d_1+d_2, \cdot), P5(D_1+d_1+d_3, \cdot), P4 \otimes P5 \multimap P6, t_5 \vdash P7}{\multimap_L} \\
 \frac{P2(D_1+d_1, D_1+d_1+d_2) \vdash P2 \quad P3(D_1+d_1, \cdot), P4(D_1+d_1+d_2, \cdot), P3 \multimap P5, t_4, t_5 \vdash P7}{\multimap_L} \\
 \frac{P2(D_1+d_1, \cdot), P3(D_1+d_1, \cdot), P2 \multimap P4, t_3, t_4, t_5 \vdash P7}{\otimes_L} \\
 \frac{P1(D_1, D_1+d_1) \vdash P1 \quad P2(D_1+d_1, \cdot) \otimes P3(D_1+d_1, \cdot), P2 \multimap P4, t_3, t_4, t_5 \vdash P7}{\multimap_L} \\
 P1(D_1, \cdot), P1 \multimap P2 \otimes P3, t_2, t_3, t_4, t_5 \vdash P7
 \end{array}$$

A Tabela 2 mostra as datas simbólicas de produção e consumo de cada átomo da rede de Petri t-temporal da Figura 3.

Tabela 2 – Datas simbólicas de produção e consumo dos átomos da rede de Petri t-temporal da Figura 3.

Átomo	Data de Produção	Data de Consumo
$P1$	D_1	D_1+d_1
$P2$	D_1+d_1	$D_1+d_1+d_2$
$P3$	D_1+d_1	$D_1+d_1+d_3$
$P4$	$D_1+d_1+d_2$	$D_1+d_1+max(d_2,d_3)+d_4$
$P5$	$D_1+d_1+d_3$	$D_1+d_1+max(d_2,d_3)+d_4$
$P6$	$D_1+d_1+max(d_2,d_3)+d_4$	$D_1+d_1+max(d_2,d_3)+d_4+d_5$
$P7$	$D_1+d_1+max(d_2,d_3)+d_4+d_5$	<i>desconhecido</i>

Em um modelo de rede de Petri t-temporal, toda duração de sensibilização d_i de uma transição t_i tem um valor que pertence a um intervalo de tempo $\Delta_i = [\delta_{imin}, \delta_{imax}]$. Logo, uma vez que as datas simbólicas computadas dependem de d_i , seus domínios também serão em função de intervalos de tempo. A Tabela 3 mostra os intervalos de datas simbólicas de produção e consumo de cada átomo da rede de Petri t-temporal da Figura 3.

Tabela 3 – Intervalos de datas simbólicas de produção e consumo dos átomos da rede de Petri t-temporal da Figura 3.

Átomo	Data de Produção	Data de Consumo
$P1$	D_1	$[D_1+d_{1min}, D_1+d_{1max}]$
$P2$	$[D_1+d_{1min}, D_1+d_{1max}]$	$[D_1+d_{1min}+d_{2min}, D_1+d_{1max}+d_{2max}]$
$P3$	$[D_1+d_{1min}, D_1+d_{1max}]$	$[D_1+d_{1min}+d_{3min}, D_1+d_{1max}+d_{3max}]$
$P4$	$[D_1+d_{1min}+d_{2min}, D_1+d_{1max}+d_{2max}]$	$[D_1+d_{1min}+max(d_{2min}, d_{3min})+d_{4min}, D_1+d_{1max}+max(d_{2max}, d_{3max})+d_{4max}]$
$P5$	$[D_1+d_{1min}+d_{3min}, D_1+d_{1max}+d_{3max}]$	$[D_1+d_{1min}+max(d_{2min}, d_{3min})+d_{4min}, D_1+d_{1max}+max(d_{2max}, d_{3max})+d_{4max}]$
$P6$	$[D_1+d_{1min}+max(d_{2min}, d_{3min})+d_{4min}, D_1+d_{1max}+max(d_{2max}, d_{3max})+d_{4max}]$	$[D_1+d_{1min}+max(d_{2min}, d_{3min})+d_{4min}+d_{5min}, D_1+d_{1max}+max(d_{2max}, d_{3max})+d_{4max}+d_{5max}]$
$P7$	$[D_1+d_{1min}+max(d_{2min}, d_{3min})+d_{4min}+d_{5min}, D_1+d_{1max}+max(d_{2max}, d_{3max})+d_{4max}+d_{5max}]$	<i>desconhecido</i>

Considerando os intervalos de datas numéricas, apresentados na Figura 3, e os intervalos de datas simbólicas, apresentados na Tabela 3, pode-se substituir as datas simbólicas pelas numéricas, obtendo datas numéricas para de produção e consumo de cada átomo da rede de Petri t-temporal da Figura 3. O resultado obtido, considerando que $D_1 = 0$, é apresentado na Tabela 4.

Para finalizar, deve-se destacar a importância da derivação de datas simbólicas em vez de datas numéricas: as datas simbólicas podem ser utilizadas para o cômputo de quaisquer datas numéricas que venham a ser associadas às transições. Logo, uma vez calculadas, estas podem ser reutilizadas quando os valores numéricos associados às transições são alterados.

Tabela 4 – Intervalos de datas numéricas de produção e consumo dos átomos da rede de Petri t-temporal da Figura 3.

Átomo	Data de Produção	Data de Consumo
$P1$	0	[1,2]
$P2$	[1,2]	[4,8]
$P3$	[1,2]	[5,10]
$P4$	[4,8]	[13,22]
$P5$	[5,10]	[13,22]
$P6$	[13,22]	[16,27]
$P7$	[16,27]	<i>desconhecido</i>

2.2 Trabalhos Relacionados

Um dos principais trabalhos correlatos a esta pesquisa é apresentado por Aalst (1998b). Isso porque o autor define a WorkFlow net Interorganizacional, que são usadas para modelar os processos de *workflow* interorganizacionais, considerados nesta pesquisa. Além disso, o autor define o critério de correção *Soundness* no contexto dos processos de *workflow* interorganizacionais. Este trabalho também apresenta a primeira abordagem para verificação de processos de *workflow* interorganizacionais. O método proposto para verificação do critério de correção *Soundness* considera a prova de limitabilidade e vivacidade para $n + 1$ WorkFlow nets. Ou seja, n WorkFlow nets locais (LWF-net) mais a $U(IOWF-net)$. As abordagens apresentadas em Aalst (1998b), Aalst (2000), Yamaguchi et al. (2007) e Sun e Du (2008) para verificação do critério de correção *Soundness* para processos de *workflow* interorganizacionais também são baseadas na prova de vivacidade e limitabilidade para a rede de Petri considerada. O problema é que, para decidir *Soundness* considerando a prova de vivacidade e limitabilidade, é necessário construir um grafo das marcações acessíveis que pode apresentar uma explosão no espaço de estados. Como citado anteriormente e mostrado por Aalst (1996a), decidir *Soundness* considerando a verificação da limitabilidade e vivacidade é EXPSPACE-difícil.

No contexto dos processos de *workflow* organizacionais simples, Dehnert e Rittgen (2001) propõem relaxar o critério de correção *Soundness*. Os autores definem o critério denominado *Relaxed Soundness* e argumentam que a noção de *Relaxed Soundness* é mais próxima da intuição do modelador, uma vez que não o força a pensar sobre todas as possibilidades de execução. Um processo de *workflow* é *Relaxed Sound* se possui execuções suficientes que terminam corretamente. A noção de suficiência está ligada ao fato de cada transição da rede de Petri que modela o processo de *workflow* ser disparada em pelo menos uma execução que finaliza corretamente. Siegeris e Zimmermann (2006) apresentam várias possibilidades para a composição de processos de *workflow*. Os autores investigam a habilidade dessas composições preservarem o critério *Relaxed Soundness*, quando os processos considerados na composição satisfazem tal critério de correção. Assim, os processos de *workflow* a serem compostos precisam ser previamente verificados em relação ao

critério *Relaxed Soundness* e caso não satisfaçam este critério, devem ser modificados até que o critério seja satisfeito. Além disso, o critério *Relaxed Soundness* não foi formalizado no contexto dos processos de *workflow* interorganizacionais.

Ainda de acordo com Dehnert e Rittgen (2001), não há nenhuma propriedade estrutural, como limitabilidade ou vivacidade, a partir da qual o critério *Relaxed Soundness* possa ser derivado. Conforme apresentado em Dehnert (2003), para a prova do critério *Relaxed Soundness* é necessário verificar, além da ausência de tarefas mortas, os invariantes de transição. Assim, quando considera-se uma abordagem baseada na construção e análise de grafos das marcações acessíveis para a prova de *Soundness*, como a abordagem apresentada por Aalst (1998b), se o modelo analisado não é *Sound*, é necessário reanalisar e reexplorar todo o modelo para verificar se o mesmo satisfaz o critério de correção *Relaxed Soundness*.

O critério de correção *Weak Soundness*, que representa o enfraquecimento do critério de correção *Soundness*, é proposto por Martens (2003). Nas abordagens apresentadas por Martens (2003) e Martens (2005), a verificação do critério *Weak Soundness* é baseada na construção e análise de grafos de comunicação (*c-graph*) e grafos de usabilidade (*u-graph*). No entanto, é importante destacar que, apesar da abordagem apresentada em Martens (2003) e Martens (2005) não ser diretamente obtida a partir da análise de um grafo das marcações acessíveis, é possível decidir *Weak Soundness* considerando a análise de um grafo das marcações acessíveis construído para verificação do critério de correção *Soundness*, uma vez que é possível verificar em um grafo das marcações acessíveis a presença ou ausência de um *deadlock*.

Passos e Julia (2009) apresentam uma abordagem baseada na Lógica Linear para verificação do critério de correção *Soundness* no contexto das WorkFlow nets, ou seja, dos processos de *workflow* que consideram apenas uma organização envolvida. Logo, a abordagem apresentada não considera a comunicação entre processos (ou organizações). E, por consequência, a verificação de *Soundness* para os processos de *workflow* interorganizacionais não é atingida. O método proposto por Passos e Julia (2009) é baseado na construção e análise de árvores de prova da Lógica Linear construídas para os cenários de uma WorkFlow net. É importante destacar a natureza de um cenário: um cenário é um conjunto de disparo de transições provido de uma ordem parcial, que passa de uma marcação inicial a uma marcação final. O meta-conector “,” de um sequente linear é comutativo. No caso de um grafo das marcações acessíveis, um cenário será um conjunto de transições completamente ordenadas. Consequentemente, quando há paralelismos em uma WorkFlow net, um cenário dado por um sequente da Lógica Linear corresponde a um conjunto de cenários em um grafo das marcações acessíveis, uma vez que o paralelismo em tais grafos é representado por intercalação. A complexidade adicional que uma estrutura paralela em uma rede de Petri produz no grafo das marcações acessíveis desaparecerá nas árvores de prova do cálculo de sequentes da Lógica Linear e a formalização do método

proposto considera a estrutura real de um processo de *workflow*, dado por uma WorkFlow net. Além disso, Passos e Julia (2009) mostram que o tempo de construção de uma árvore da Lógica Linear que representa um cenário de uma WorkFlow net é linear em relação ao número de transições da WorkFlow net analisada.

Passos et al. (2013) propõem a transformação de um diagrama de atividades da UML, que modela um processo de *workflow*, em uma WorkFlow net. A transformação é automatizada utilizando a linguagem de transformação *Atlas Transformation Language (ATL)*, que é reconhecida no contexto do desenvolvimento dirigido a modelos. Esta transformação possibilita a verificação do critério de correção *Soundness* utilizando a construção e análise de árvores de prova da Lógica Linear. As árvores de prova construídas também auxiliam na busca pelas possíveis causas do modelo não ser *Sound*. Um estudo de caso ilustrativo é apresentado para demonstrar a abordagem. Os resultados da abordagem podem motivar as companhias de *software* a explorar o rigor das verificações formais, utilizando a rede de Petri e a Lógica Linear, sem sacrificar as práticas comuns dos desenvolvedores, que podem continuar utilizando a notação UML integrada com ferramentas industriais, como é o caso da ATL.

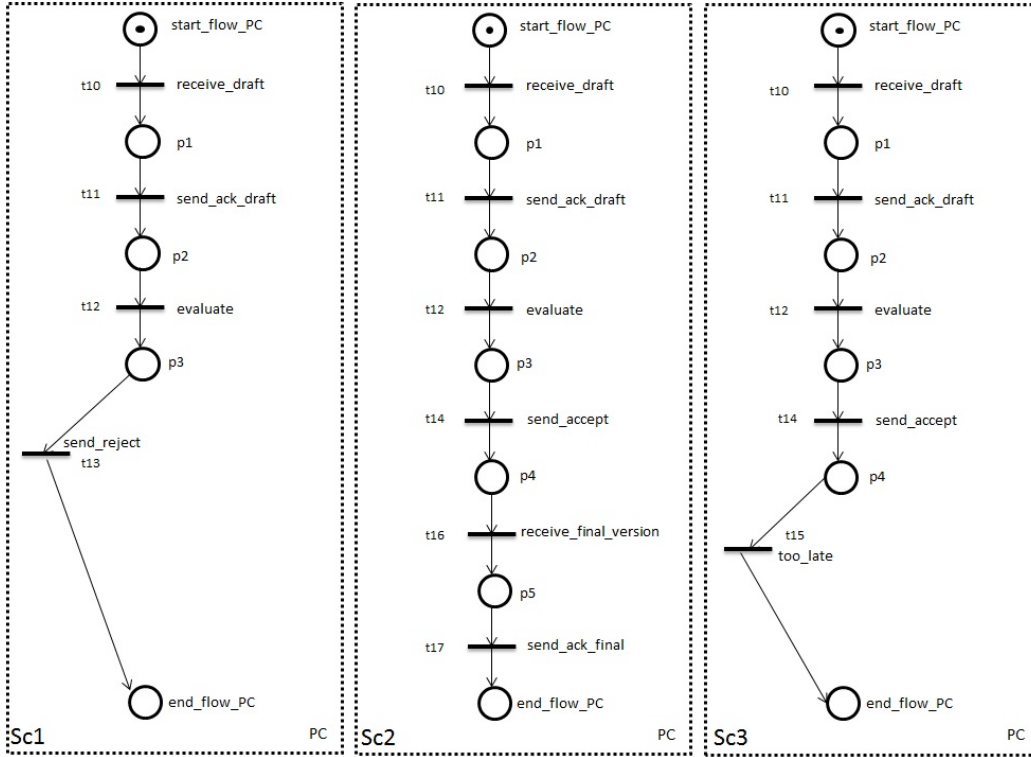
Ainda no contexto da análise de processos de *workflow* utilizando Lógica Linear, um artigo intitulado “*Linear Logic as a Tool for Qualitative and Quantitative Analysis of Workflow Processes*” (PASSOS; JULIA, 2015a), foi aceito para publicação no *International Journal on Artificial Intelligence Tools*. O artigo apresenta dois métodos: um para análise qualitativa e outro para análise quantitativa de modelos em WorkFlow nets. Estes métodos são baseados na construção e análise de árvores de prova da Lógica Linear. A análise qualitativa está relacionada com a prova de *Soundness* para a WorkFlow net. A análise quantitativa está relacionada com o planejamento de recursos para cada tarefa do processo de *workflow* analisado, isto é, para as atividades que possuem acionamento do tipo usuário. A análise quantitativa é baseada na computação de intervalos de datas simbólicas para a execução das tarefas, de forma que um planejamento dos recursos envolvidos possa ser realizado. Em particular, tais intervalos de datas simbólicas são computados reutilizando os resultados da análise qualitativa, ou seja, das árvores de prova da Lógica Linear utilizadas para a prova do critério de correção *Soundness*. Esta é uma das vantagens encontradas ao se utilizar a Lógica Linear para a verificação de processos de *workflow*, uma vez que, quando considera-se abordagens tradicionais baseadas na construção e análise de grafos das marcações acessíveis, tal planejamento de recursos não é diretamente obtido. Além disso, este trabalho apresenta também resultados de simulação que confirmam os dados obtidos com o cálculo de datas simbólicas. Os métodos para análise qualitativa e quantitativa apresentadas neste artigo são fruto dos trabalhos desenvolvidos anteriormente pelos autores.

Métodos Baseados na Lógica Linear para Verificação de Processos de *Workflow* Interorganizacionais

Este capítulo apresenta quatro métodos baseados na Lógica Linear para verificação de processos de *workflow* interorganizacionais formalizados nesta pesquisa. O método para verificação do critério de correção *Soundness* para processos de *workflow* interorganizacionais é apresentado na seção 3.1. A seção 3.2 apresenta o método para verificação do critério de correção *Relaxed Soundness* para processos de *workflow* interorganizacionais. Já a seção 3.3 apresenta a formalização do método para verificação do critério de correção *Weak Soundness* para processos de *workflow* interorganizacionais. Finalmente, a seção 3.4 formaliza o método para detecção de cenários livres de *deadlock* em processos de *workflow* interorganizacionais.

Uma noção comum aos quatro métodos é o conceito de cenário. Um cenário, no contexto das LWF-nets e da $U(\text{IOWF-net})$, corresponde a uma rota bem definida em uma LWF-nets ou $U(\text{IOWF-net})$. Assim, se o modelo analisado tem mais de uma rota (lugar com dois ou mais arcos de saída), é necessário considerar um sequente linear diferente para cada cenário existente. Por exemplo, considerando a LWF-net PC apresentada na Figura 6, há três cenários diferentes: o primeiro cenário, Sc_1 , onde a tarefa *send_reject* é executada (disparo da transição *send_reject*); o segundo cenário, Sc_2 , onde a tarefa *receive_final_version* é realizada (disparo da transição *receive_final_version*); e o terceiro cenário, Sc_3 , onde a tarefa *too_late* é executada. A Figura 11 mostra os cenários para a LWF-net PC .

É importante enfatizar novamente a natureza de um cenário que, nesta pesquisa, é um conjunto de disparo de transições provido de uma ordem parcial, que passa de uma marcação inicial a uma marcação final. O meta-conector “,” de um sequente linear é comutativo. Já no caso de um grafo das marcações acessíveis, um cenário é um conjunto de transições completamente ordenadas. Consequentemente, quando existe paralelismo


 Figura 11 – Cenários para a LWF-net PC .

em uma LWF-net (ou $U(IOWF-net)$), um cenário dado por um sequente da Lógica Linear corresponde a um conjunto de cenários em um grafo das marcações acessíveis, uma vez que o paralelismo em tais grafos é representado por intercalação. Em particular, se dois eventos A e B podem ocorrer paralelamente, o grafo das marcações irá produzir duas ordens possíveis: A seguido de B e B seguido de A , mesmo se não houver nenhuma relação de causalidade entre tais eventos. A complexidade adicional que uma estrutura paralela em uma rede de Petri produz no grafo das marcações acessíveis desaparecerá nas árvores de prova do cálculo de sequentes da Lógica Linear e a formalização dos métodos propostos nesta pesquisa irão considerar então a estrutura real de um processo de *workflow* interorganizacional dado pela $IOWF-net$, que é uma rede de Petri acíclica. No pior caso, a complexidade dos métodos propostos pode eventualmente se tornar exponencial, em função do número de roteiros condicionais (ou seletivos) existentes no modelo analisado. No entanto, os roteiros paralelos nos processos de *workflow* interorganizacionais não introduzem complexidade adicional, como acontece em abordagens clássicas que são baseadas na construção de grafos das marcações acessíveis.

Além disso, para a exemplificação dos quatro métodos, é utilizado o processo de *workflow* interorganizacional modelado através da $IOWF-net$ mostrada na Figura 6. Considerando a representação das transições por expressões da Lógica Linear, apresentadas na subseção 2.1.4 e a $U(IOWF-net)$ da Figura 7, as transições da $U(IOWF-net)$ são representadas por:

$$t_i = i \multimap start_flow_author \otimes start_flow_PC,$$

$$\begin{aligned}
t_1 &= \text{send_draft} = \text{start_flow_author} \multimap a1 \otimes \text{draft}, \\
t_2 &= \text{receive_ack_draft} = a1 \otimes \text{ack_draft} \multimap a2, \\
t_3 &= \text{receive_accept} = a2 \otimes \text{accept} \multimap a3, \\
t_4 &= \text{receive_reject} = a2 \otimes \text{reject} \multimap \text{end_flow_author}, \\
t_5 &= \text{prepare_final_version} = a3 \multimap a4, \\
t_6 &= \text{receive_notification_1} = a3 \otimes \text{too_late} \multimap \text{end_flow_author}, \\
t_7 &= \text{send_final_version} = a4 \multimap a5 \otimes \text{final_version}, \\
t_8 &= \text{receive_notification_2} = a4 \otimes \text{too_late} \multimap \text{end_flow_author}, \\
t_9 &= \text{receive_ack_final} = a5 \otimes \text{ack_final} \multimap \text{end_flow_author}, \\
t_{10} &= \text{receive_draft} = \text{start_flow_PC} \otimes \text{draft} \multimap p1, \\
t_{11} &= \text{send_ack_draft} = p1 \multimap \text{ack_draft} \otimes p2, \\
t_{12} &= \text{evaluate} = p2 \multimap p3, \\
t_{13} &= \text{send_reject} = p3 \multimap \text{reject} \otimes \text{end_flow_PC}, \\
t_{14} &= \text{send_accept} = p3 \multimap \text{accept} \otimes p4, \\
t_{15} &= \text{too_late} = p4 \multimap \text{too_late} \otimes \text{end_flow_PC}, \\
t_{16} &= \text{receive_final_version} = \text{final_version} \otimes p4 \multimap p5, \\
t_{17} &= \text{send_ack_final} = p5 \multimap \text{ack_final} \otimes \text{end_flow_PC}, \\
t_o &= \text{end_flow_author} \otimes \text{end_flow_PC} \multimap o.
\end{aligned}$$

Além disso, os átomos nas árvores de prova poderão ser representados apenas pelas letras iniciais de seus nomes completos. Por exemplo, *too_late* e *final_version* poderão ser representados por *tl* e *fv*, respectivamente. Além disso, um conjunto de transições t_i, t_j, t_k and t_l pode ser representado por $t_{i,j,k,l}$. Estas reduções podem ocorrer devido ao tamanho e legibilidade das árvores de prova apresentadas.

Considerando tais informações, os métodos baseados na Lógica Linear para a verificação de processos de *workflow* interorganizacionais são apresentados na sequência.

3.1 Verificação do Critério de Correção *Soundness* para Processos de *Workflow* Interorganizacionais

Nesta seção é formalizado o método para verificação do critério de correção *Soundness* no contexto dos processos de *workflow* interorganizacionais. Este método considera a definição da WorkFlow net Interorganizacional (IOWF-net) apresentada na subseção 2.1.3. O método apresentado permite a verificação do critério *Soundness* sem considerar a construção de um grafo das marcações acessíveis, como acontece nas abordagens clássicas, considerando a própria estrutura do modelo em IOWF-net que é acíclico.

Considerando a IOWF-net e o critério de correção *Soundness*, é necessário provar *Soundness* para cada WorkFlow net local (LWF-net), desconsiderando os lugares de comunicação existentes. Se pelo menos um dos modelos em LWF-net analisados não satisfizer o critério de correção *Soundness*, a IOWF-net também não satisfará tal critério,

de acordo com a definição de *Soundness* para IOWF-net apresentada na subseção 2.1.3. Além disso, se todos os modelos em LWF-net são *Sound*, é então necessário provar que a $U(\text{IOWF-net})$ satisfaz o critério de correção *Soundness*, de acordo com a definição de *Soundness* para IOWF-net apresentada na subseção 2.1.3, uma vez que podem ocorrer erros de sincronização quando considerada a composição dos modelos em LWF-net. Assim, se a $U(\text{IOWF-net})$ não é *Sound*, a IOWF-net também não será *Sound*. Somente nos casos em que os modelos em LWF-net e a $U(\text{IOWF-net})$ satisfizerem o critério de correção *Soundness* é que a IOWF-net será *Sound*.

Para provar o critério de correção *Soundness* para os modelos em IOWF-net utilizando Lógica Linear, inicialmente, é necessário representar cada LWF-net que compõe a IOWF-net e a $U(\text{IOWF-net})$ através de fórmulas da Lógica Linear. Assim, cada LWF-net e a $U(\text{IOWF-net})$ são representadas por um ou mais sequentes da Lógica Linear. Cada sequente linear considera a marcação inicial e final da LWF-net ou da $U(\text{IOWF-net})$ analisada e uma lista não ordenada das transições envolvidas no cenário.

Após a definição dos sequentes da Lógica Linear que representam os cenários existentes de cada LWF-net e da $U(\text{IOWF-net})$, tais sequentes devem ser provados. Para provar os sequentes, árvores de prova da Lógica Linear são então construídas.

Após a construção das árvores de prova, cada cenário de cada LWF-net PN_k que compõe a IOWF-net e cada cenário da $U(\text{IOWF-net})$ deve ser analisado seguindo os seguintes passos:

1. para cada árvore de prova de cada LWF-net relativa a PN_k (e para cada árvore de prova da $U(\text{IOWF-net})$):
 - a) se apenas um átomo o , que corresponde a uma ficha no lugar de término da LWF-net (ou da $U(\text{IOWF-net})$) analisada foi produzido na árvore de prova (este fato é representado na árvore de prova pelo sequente identidade $o \vdash o$), então o primeiro requisito para a prova de *Soundness* é verificado, isto é, apenas uma ficha aparece no lugar de término;
 - b) se não há nenhum átomo disponível para consumo no último sequente da árvore de prova, então significa que todos os lugares da LWF-net (ou da $U(\text{IOWF-net})$) analisada estão vazios quando a execução do processo é finalizada, isto é, o segundo requisito para a prova de *Soundness* é verificado;
 - c) não existe nenhuma fórmula de transição disponível no último sequente da árvore de prova a ser considerada;
2. considerando todos os cenários Sc_1, Sc_2, \dots, Sc_n para a LWF-net relativa a PN_k (ou $U(\text{IOWF-net})$) analisada, cada transição $t \in T_k$, (ou $T^* \cup \{t_i, t_o\}$, no caso da $U(\text{IOWF-net})$) precisa aparecer, em pelo menos, em uma árvore de prova. Isto prova que todas as transições serão disparadas (ou seja, não existe transições mortas). Este

fato e a condição c) implicam na verificação do terceiro requisito para a prova de *Soundness*.

Se as condições 1 e 2 acima são satisfeitas, considerando cada LWF-net que compõe a IOWF-net e a $U(\text{IOWF-net})$, a IOWF-net analisada é *Sound*.

É importante destacar que o método proposto deve ser aplicado a cada LWF-net que compõe a IOWF-net. Se todas as LWF-nets são *Sound*, então é necessário aplicar o método apresentado aos cenários da $U(\text{IOWF-net})$, para verificar se esta também é *Sound*. Se as condições 1 e 2 são então satisfeitas considerando as LWF-nets e a $U(\text{IOWF-net})$, a IOWF-net é *Sound*.

Para ilustrar o método proposto, a IOWF-net mostrada na Figura 6 é considerada. Neste caso, é necessário inicialmente provar *Soundness* para as LWF-nets *Author* e *PC*. Para provar *Soundness* para a LWF-net *Author* é necessário provar os quatro sequentes da Lógica Linear mostrados na sequência, cada um correspondendo a um cenário da LWF-net *Author*:

$$\begin{aligned} Sc_1A &= sfa, t_1, t_2, t_4 \vdash efa, \\ Sc_2A &= sfa, t_1, t_2, t_3, t_5, t_7, t_9 \vdash efa, \\ Sc_3A &= sfa, t_1, t_2, t_3, t_6 \vdash efa, \\ Sc_4A &= sfa, t_1, t_2, t_3, t_5, t_8 \vdash efa. \end{aligned}$$

Para provar *Soundness* para a LWF-net *PC*, é necessário provar três sequentes da Lógica Linear, cada um correspondendo a um cenário da LWF-net *PC*. Estes cenários são:

$$\begin{aligned} Sc_1PC &= sfpc, t_{10}, t_{11}, t_{12}, t_{14}, t_{15} \vdash efpc, \\ Sc_2PC &= sfpc, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17} \vdash efpc, \\ Sc_3PC &= sfpc, t_{10}, t_{11}, t_{12}, t_{13} \vdash efpc. \end{aligned}$$

Após a definição dos sequentes da Lógica Linear de cada LWF-net, estes devem ser provados através da construção de árvores de prova da Lógica Linear. Deve-se observar que os lugares de comunicação não devem ser considerados.

As árvores de prova completas para os cenários Sc_1A , Sc_2A , Sc_3A , Sc_4A , Sc_1PC , Sc_2PC e Sc_3PC são apresentadas no Apêndice A.1. Na sequência são apresentados somente os resumos destas árvores de prova. Tais resumos apresentam apenas o primeiro e último sequentes das árvores de prova completas.

O resumo da árvore de prova para o cenário Sc_1A é apresentado na sequência.

$$\frac{\frac{efa \vdash efa \multimap_L}{\vdots}}{sfa, sfa \multimap a1, a1 \multimap a2, a2 \multimap efa \vdash efa}$$

O resumo da árvore de prova para o cenário Sc_2A é o seguinte:

$$\frac{\frac{efa \vdash efa}{\multimap_L} \quad \vdots}{sfa, sfa \multimap a1, a1 \multimap a2, a2 \multimap a3, a3 \multimap a4, a4 \multimap a5, a5 \multimap efa \vdash efa}$$

O resumo da árvore de prova para o cenário Sc_3A é dado por:

$$\frac{\frac{efa \vdash efa}{\multimap_L} \quad \vdots}{sfa, sfa \multimap a1, a1 \multimap a2, a2 \multimap a3, a3 \multimap efa \vdash efa}$$

O resumo da árvore de prova para o cenário Sc_4A é apresentado na sequência.

$$\frac{\frac{efa \vdash efa}{\multimap_L} \quad \vdots}{sfa, sfa \multimap a1, a1 \multimap a2, a2 \multimap a3, a3 \multimap a4, a4 \multimap efa \vdash efa}$$

Considerando as árvores de prova da Lógica Linear apresentadas para os cenários Sc_1A , Sc_2A , Sc_3A e Sc_4A da LWF-net *Author*, é possível verificar que as condições 1 e 2 são satisfeitas, ou seja:

1. em cada árvore de prova da LWF-net *Author*:
 - a) apenas um átomo *efa* que corresponde ao átomo *o* na definição de WorkFlow net mostrada na subseção 2.1.2, que indica a presença de uma ficha no lugar de término da LWF-net *Author*, foi produzido na árvore de prova (este fato é representado nas árvores de prova pelo sequente identidade $efa \vdash efa$);
 - b) não há nenhum átomo disponível para consumo no último sequente de cada árvore de prova construída; isso significa que todos os lugares da LWF-net *Author* estão vazios nos cenários analisados;
 - c) não há nenhuma fórmula de transição disponível no último sequente das árvores de prova construídas; isso significa que todas as transições do sequente inicial foram disparadas;
2. considerando todos os cenários Sc_1A , Sc_2A , Sc_3A e Sc_4A para a LWF-net *Author*, cada transição $t \in T$ da LWF-net *Author* aparece pelo menos uma vez em uma das árvores de prova.

Como as condições 1 e 2 são satisfeitas, tem-se que a LWF-net *Author* satisfaz o critério de correção *Soundness*.

Na sequência, são mostrados os resumos das árvores de prova construídas para os cenários Sc_1PC , Sc_2PC e Sc_3PC da LWF-net PC . O resumo da árvore de prova para o cenário Sc_1PC é dado por:

$$\frac{\frac{efpc \vdash efpc}{\vdash} \multimap_L}{sfpc, sfpc \multimap p1, p1 \multimap p2, p2 \multimap p3, p3 \multimap p4, p4 \multimap efpc \vdash efpc}$$

O resumo da árvore de prova para o cenário Sc_2PC é dado por:

$$\frac{\frac{efpc \vdash efpc}{\vdash} \multimap_L}{sfpc, sfpc \multimap p1, p1 \multimap p2, p2 \multimap p3, p3 \multimap p4, p4 \multimap p5, p5 \multimap efpc \vdash efpc}$$

O resumo da árvore de prova para o cenário Sc_3PC é o seguinte:

$$\frac{\frac{efpc \vdash efpc}{\vdash} \multimap_L}{sfpc, sfpc \multimap p1, p1 \multimap p2, p2 \multimap p3, p3 \multimap efpc \vdash efpc}$$

Considerando as árvores de prova da Lógica Linear construídas para os cenários Sc_1PC , Sc_2PC e Sc_3PC da LWF-net PC , é possível verificar que as condições 1 e 2 são satisfeitas, ou seja:

1. em cada árvore de prova da LWF-net PC :
 - a) apenas um átomo $efpc$ que corresponde ao átomo o na definição de Workflow net mostrada na subseção 2.1.2, que indica a presença de uma ficha no lugar de término da LWF-net PC , foi produzido na árvore de prova (este fato é representado nas árvores de prova pelo seguinte identidade $efpc \vdash efpc$);
 - b) não há nenhum átomo disponível para consumo no último sequente de cada árvore de prova construída; isso significa que todos os lugares da LWF-net PC estão vazios nos cenários analisados;

- c) não há nenhuma fórmula de transição disponível no último sequente das árvores de prova construídas; isso significa que todas as transições do sequente inicial foram disparadas;
2. considerando todos os cenários Sc_1PC , Sc_2PC e Sc_3PC para a LWF-net PC , cada transição $t \in T$ da LWF-net PC aparece pelo menos uma vez em uma das árvores de prova.

Como as condições 1 e 2 são satisfeitas, tem-se que a LWF-net PC satisfaz o critério de correção *Soundness*.

Uma vez que as LWF-nets são *Sound*, é necessário provar o critério de correção *Soundness* para a $U(\text{IOWF-net})$, mostrada na Figura 7. Para tanto, é necessário definir e provar cinco sequentes da Lógica Linear, sendo que cada sequente representa um cenário na $U(\text{IOWF-net})$. Os cenários da $U(\text{IOWF-net})$ mostrada na Figura 7 são dados pelos seguintes sequentes da Lógica Linear:

$$\begin{aligned} Sc_1 &= i, t_i, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o, \\ Sc_2 &= i, t_i, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o, \\ Sc_3 &= i, t_i, t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o, \\ Sc_4 &= i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16,17,o} \vdash o, \\ Sc_5 &= i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o. \end{aligned}$$

O primeiro cenário, Sc_1 , é o cenário onde a tarefa *send_reject* será executada (disparo da transição *send_reject*). O segundo cenário, Sc_2 , é o cenário no qual as tarefas *too_late* e *receive_notification_1* serão executadas (disparo das transições *too_late* e *receive_notification_1*). Já o terceiro cenário, Sc_3 , corresponde ao cenário onde as tarefas *too_late* e *receive_notification_2* são realizadas (disparo das transições *too_late* e *receive_notification_2*). As tarefas *send_final_version* e *receive_final_version* (disparo das transições *send_final_version* e *receive_final_version*) são consideradas no cenário Sc_4 . Finalmente, o quinto cenário, Sc_5 , é o cenário onde as tarefas *too_late* e *send_final_version* são executadas (disparo das transições *too_late* e *send_final_version*).

Após a definição dos sequentes da Lógica Linear que representam cada cenário da $U(\text{IOWF-net})$, estes sequentes precisam ser provados. Para prová-los, árvores de prova da Lógica Linear serão construídas.

As árvores de prova completas construídas para a prova de cada um destes sequentes são apresentadas no Apêndice A.1. Na sequência são apresentados os resumos das árvores de prova para os cenários Sc_1 , Sc_2 , Sc_3 , Sc_4 e a árvore de prova completa para o cenário Sc_5 .

O resumo da árvore de prova para o cenário Sc_1 é como se segue:

$$\frac{\frac{o \vdash o}{\vdash o} \multimap_L}{\vdash o} \vdots$$

$$\frac{}{i, t_i, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o}$$

O resumo da árvore de prova para o cenário Sc_2 é dado por:

$$\frac{\frac{o \vdash o}{\vdash o} \multimap_L}{\vdash o} \vdots$$

$$\frac{}{i, t_i, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o}$$

O resumo da árvore de prova para o cenário Sc_3 é a mostrado na sequência.

$$\frac{\frac{o \vdash o}{\vdash o} \multimap_L}{\vdash o} \vdots$$

$$\frac{}{i, t_i, t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o}$$

O resumo da árvore de prova para o cenário Sc_4 é como se segue:

$$\frac{\frac{o \vdash o}{\vdash o} \multimap_L}{\vdash o} \vdots$$

$$\frac{}{i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, t_o \vdash o}$$

E, finalmente, a árvore de prova para o cenário Sc_5 é dada por:

$$\begin{array}{c}
\frac{}{tl,efpc,a5,fv,t9,to \vdash o} \otimes_L \\
\frac{a4 \vdash a4 \quad tl,efpc,a5 \otimes fv,t9,to \vdash o}{\vdash} \multimap_L \\
\frac{a3 \vdash a3 \quad tl,efpc,a4,a4 \multimap a5 \otimes fv,t9,to \vdash o}{\vdash} \multimap_L \\
\frac{}{a3,tl,efpc,a3 \multimap a4,t7,t9,to \vdash o} \otimes_L \\
\frac{p4 \vdash p4 \quad a3,tl \otimes efpc,t5,t7,t9,to \vdash o}{\vdash} \multimap_L \\
\frac{\frac{a2 \vdash a2}{a2,a \vdash a2 \otimes a} \quad \frac{a \vdash a}{\vdash} \otimes_R \quad p4,a3,t5,t7,t9,p4 \multimap tl \otimes efpc,to \vdash o}{\vdash} \multimap_L \\
\frac{}{a2,a,p4,a2 \otimes a \multimap a3,t5,t7,t9,t15,to \vdash o} \otimes_L \\
\frac{p3 \vdash p3 \quad a2,a \otimes p4,t3,t5,t7,t9,t15,to \vdash o}{\vdash} \multimap_L \\
\frac{p2 \vdash p2 \quad a2,p3,t3,t5,t7,t9,p3 \multimap a \otimes p4,t15,to \vdash o}{\vdash} \multimap_L \\
\frac{\frac{a1 \vdash a1}{a1,ad \vdash a1 \otimes ad} \quad \frac{ad \vdash ad}{\vdash} \otimes_R \quad p2,a2,t3,t5,t7,t9,p2 \multimap p3,t14,t15,to \vdash o}{\vdash} \multimap_L \\
\frac{}{a1,ad,p2,a1 \otimes ad \multimap a2,t3,t5,t7,t9,t12,t14,t15,to \vdash o} \otimes_L \\
\frac{p1 \vdash p1 \quad a1,ad \otimes p2,t2,t3,t5,t7,t9,t12,t14,t15,to \vdash o}{\vdash} \multimap_L \\
\frac{\frac{sf p \vdash sf p}{a1,ad \vdash sf p \otimes d} \quad \frac{d \vdash d}{\vdash} \otimes_R \quad a1,p1,t2,3,5,7,9,p1 \multimap ad \otimes p2,t12,t14,t15,to \vdash o}{\vdash} \multimap_L \\
\frac{}{sf p,a1,d,t2,t3,t5,t7,t9,stp \otimes d \multimap p1,t11,t12,t14,t15,to \vdash o} \otimes_L \\
\frac{sfa \vdash sfa \quad sf p,a1 \otimes d,t2,t3,t5,t7,t9,t10,t11,t12,t14,t15,to \vdash o}{\vdash} \multimap_L \\
\frac{}{sfa,sfp,sfa \multimap a1 \otimes d,t2,t3,t5,t7,t9,t10,t11,t12,t14,t15,to \vdash o} \otimes_L \\
\frac{i \vdash i \quad sfa \otimes sf p,t1,t2,t3,t5,t7,t9,t10,t11,t12,t14,t15,to \vdash o}{\vdash} \multimap_L \\
i,t_i,t_1,t_2,t_3,t_5,t_7,t_9,t_{10},t_{11},t_{12},t_{14},t_{15},to \vdash o
\end{array}$$

Considerando as árvores de prova da Lógica Linear apresentadas para os cenários Sc_1 , Sc_2 , Sc_3 , Sc_4 e a árvore de prova completa para o cenário Sc_5 da $U(\text{IOWF-net})$ mostrada na Figura 7, é possível verificar que a condição 1 não é satisfeita, uma vez que na árvore de prova para o cenário Sc_5 da $U(\text{IOWF-net})$:

- nenhum átomo o foi produzido na árvore de prova para este cenário; ou seja, o seguinte identidade $o \vdash o$ não aparece na árvore de prova; isso significa que nenhuma ficha foi produzida no lugar de término deste cenário, ou seja, para este cenário, o processo de *workflow* interorganizacional não finaliza;
- há quatro átomos disponíveis para consumo no último sequente da árvore de prova construída para o cenário Sc_5 : *too_late*, *end_flow_pc*, *a5* e o átomo *final_version*; isso significa que existe uma ficha em cada um destes lugares;
- há duas fórmulas de transição disponíveis, que não foram disparadas, no último sequente das árvore de prova: t_9 e t_o ; isso significa que as transições t_9 e t_o não foram disparadas neste cenário.

Como a condição 1 não é satisfeita, tem-se que a $U(\text{IOWF-net})$ não satisfaz o critério de correção *Soundness*. Assim, conclui-se que, apesar das LWF-nets *Author* e *PC*

satisfazerem o critério de correção *Soundness*, a IOWF-net mostrada na Figura 6 não é *Sound*.

É importante observar que o cenário Sc_5 é o cenário que leva o processo de *workflow* interorganizacional, modelado pela IOWF-net mostrada na Figura 6, a uma situação de *deadlock*. A Figura 12 mostra a marcação da situação de bloqueio, isto é, as fichas nos lugares *too_late*, *end_flow_pc*, *a5* e *final_version*, que representam exatamente os átomos remanescentes no último sequente da árvore de prova para o cenário Sc_5 .

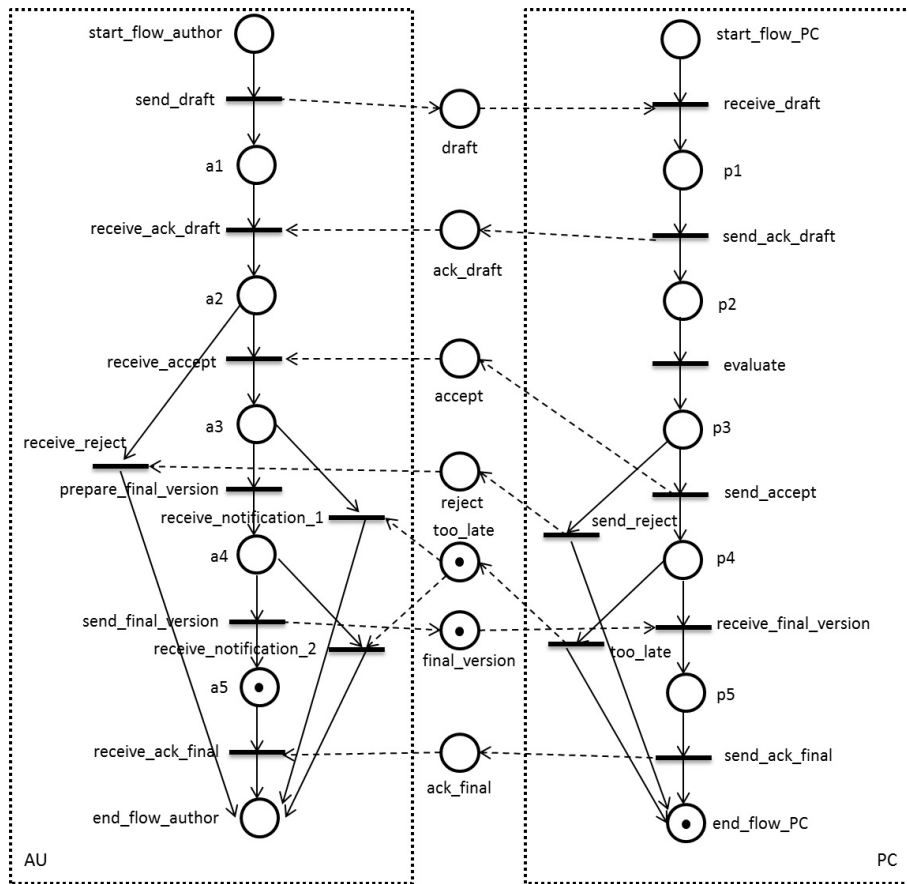


Figura 12 – Situação de *deadlock* para o processo de *workflow* interorganizacional modelado pela IOWF-net da Figura 6.

Além disso, o método para verificação do critério de correção *Soundness* apresentado nesta seção pode ser interrompido quando um dos cenários não satisfizer as condições apresentadas, uma vez que isso implicará no fato do processo analisado não ser *Sound*. Assim, nem sempre é necessário construir as árvores de prova para todos os possíveis cenários para decidir que um dado processo de *workflow* interorganizacional não satisfaz o critério de correção *Soundness*. Desta forma, o uso da Lógica Linear permite uma análise parcial do modelo, o que não é possível quando uma abordagem baseada na construção e análise de grafos das marcações acessíveis é utilizada.

3.2 Verificação do Critério de Correção *Relaxed Soundness* para Processos de *Workflow* Interorganizacionais

Como exposto anteriormente, o critério correção *Relaxed Soundness* foi definido apenas no contexto da Workflow net e não no contexto da IOWF-net. Entretanto, este critério também é importante no contexto da IOWF-net, especialmente nos casos em que o critério de correção *Soundness* não é satisfeito para uma dada IOWF-net. Conforme apresentado na Tabela 1, o critério de correção *Relaxed Soundness* implica na possibilidade de finalização do processo e na participação de todas as atividades envolvidas no processo analisado. Assim, do ponto de vista interorganizacional, a garantia obtida com a prova deste critério é, além da possibilidade do processo finalizar, o fato de cada tarefa do processo de *workflow* interorganizacional ser reallizada pelo menos em um dos cenários existentes no sistema. Assim, com a prova deste critério, as organizações podem identificar as atividades que não são contempladas em nenhuma sequência que finaliza corretamente. Essa informação pode ser útil para as organizações envolvidas considerando os dois aspectos a seguir: a tarefa pode ser essencial e, assim, um dos processos locais deve ser modificado até que a mesma seja contemplada; ou a tarefa não é essencial, podendo ser removida do processo. Pode ainda existir o fato de um recurso estar associado a uma tarefa que nunca será executada. Neste caso, o recurso pode ficar ocioso, já que está associado a uma tarefa que nunca será executada.

Para a formalização do método, deve-se observar que como a $U(\text{IOWF-net})$ de uma IOWF-net tem a mesma estrutura de uma Workflow net, é possível verificar o critério de correção *Relaxed Soundness* para uma dada IOWF-net, considerando a análise da $U(\text{IOWF-net})$ correspondente. Assim, uma IOWF-net é *Relaxed Sound* quando a $U(\text{IOWF-net})$ da IOWF-net analisada satisfaz o critério de correção *Relaxed Soundness*.

Assim, inicialmente, a $U(\text{IOWF-net})$ deve ser representada através de fórmulas da Lógica Linear. A $U(\text{IOWF-net})$ pode ser representada por um ou mais sequentes da Lógica Linear, cada um representando um cenário possível da $U(\text{IOWF-net})$. Desta forma, para verificar o critério *Relaxed Soundness* para as IOWF-nets, é necessário definir e provar sequentes da Lógica Linear, que representam cada cenário da $U(\text{IOWF-net})$ correspondente.

Neste método, cada um destes cenários é então representado por um sequente linear específico, que considera a marcação inicial e final da $U(\text{IOWF-net})$ e uma lista não ordenada das transições envolvidas em cada cenário. Além disso, cada sequente linear tem apenas um átomo que representa a marcação inicial da $U(\text{IOWF-net})$.

É importante observar que o método proposto para verificar o critério de correção *Soundness* para IOWF-net, apresentado na seção 3.1 considera a construção e análise

de árvores de prova da Lógica Linear que representam os cenários das LWF-nets e da $U(\text{IOWF-net})$ correspondente. O método proposto nesta seção para verificação do critério *Relaxed Soundness* para IOWF-net considera a análise de árvores de prova da Lógica Linear que representam cada cenário da $U(\text{IOWF-net})$ correspondente, isto é, a análise de um subconjunto dos cenários considerados na verificação do critério de correção *Soundness*. Assim, quando uma abordagem baseada na Lógica Linear para verificar o critério de correção *Soundness* para IOWF-nets, como a apresentada na seção 3.1, é considerada, se a IOWF-net analisada não é *Sound*, o subconjunto de sequentes da Lógica Linear que representam os cenários da $U(\text{IOWF-net})$ pode ser reutilizado no contexto desta abordagem, executando apenas os passos 1 e 2 de análise das árvores de prova construídas para cada cenário da $U(\text{IOWF-net})$ para decidir o critério de correção *Relaxed Soundness* para a IOWF-net analisada. É importante destacar ainda que as condições 1 e 2 apresentadas na sequência apenas relaxam as condições 1 e 2 apresentadas no método para verificação do critério de correção *Soundness* na seção 3.1, uma vez que o critério *Relaxed Soundness* simplesmente relaxa o critério de correção *Soundness*.

Assim, considerando as árvores de prova para os sequentes da Lógica Linear que representam cada cenário da $U(\text{IOWF-net})$ analisada, tais árvores de prova devem ser analisadas respeitando os seguintes passos:

1. para cada árvore de prova que representa um cenário da $U(\text{IOWF-net})$:
 - a) se apenas um átomo o que corresponde a uma ficha no lugar de término da $U(\text{IOWF-net})$ foi produzido na árvore de prova (isto é representado na árvore de prova pelo sequeute identidade $o \vdash o$), então o cenário analisado pode finalizar corretamente;
 - b) se não há nenhum átomo disponível para consumo no último sequeute da árvore de prova então significa que todos os lugares da $U(\text{IOWF-net})$ estão vazios e, então, a execução finaliza sem fichas remanescentes;
2. considerando as transições $t \in T^* \cup \{t_i, t_o\}$ da $U(\text{IOWF-net})$ analisada, cada transição t deve aparecer em pelo menos um cenário Sc_i que satisfaz o passo 1. Isto prova que cada transição t da IOWF-net está em pelo menos uma sequência de disparo que inicia na marcação inicial, i , e finaliza na marcação final, o , da $U(\text{IOWF-net})$.

Se as condições 1 e 2 acima são satisfeitas, a $U(\text{IOWF-net})$ é *Relaxed Sound* e, consequentemente, a IOWF-net analisada satisfaz o critério de correção *Relaxed Soundness*.

É relevante salientar que o método proposto para a verificação do critério de correção *Relaxed Soundness* formalizado nesta seção não considera a verificação individual dos processos de *workflow* locais envolvidos no processo de *workflow* interorganizacional, isto é, não considera a verificação individual das LWF-nets que compõem a IOWF-net. Em vez disso, o método proposto considera a verificação do modelo global, representado pela

$U(\text{IOWF-net})$. Além disso, o método considera ainda a estrutura acíclica da IOWF-net analisada.

Para ilustrar o método apresentado, a IOWF-net mostrada na Figura 6 é considerada. Para provar o critério *Relaxed Soundness* para esta IOWF-net, a $U(\text{IOWF-net})$ correspondente mostrada na Figura 7 é considerada. Para esta rede, é necessário provar cinco sequentes da Lógica Linear, cada um representando um cenário diferente.

O primeiro cenário, Sc_1 , é o cenário onde a tarefa *send_reject* será executada (disparo da transição *send_reject*). O segundo cenário, Sc_2 , é o cenário no qual as tarefas *too_late* e *receive_notification_1* serão executadas (disparo das transições *too_late* e *receive_notification_1*). Já o terceiro cenário, Sc_3 , corresponde ao cenário onde as tarefas *too_late* e *receive_notification_2* são realizadas (disparo das transições *too_late* e *receive_notification_2*). As tarefas *send_final_version* e *receive_final_version* (disparo das transições *send_final_version* e *receive_final_version*) são consideradas no cenário Sc_4 . Finalmente, o quinto cenário, Sc_5 , é o cenário onde as tarefas *too_late* e *send_final_version* são executadas (disparo das transições *too_late* e *send_final_version*). Estes cenários são dados pelos sequentes da Lógica Linear a seguir:

$$\begin{aligned} Sc_1 &= i, t_i, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o, \\ Sc_2 &= i, t_i, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o, \\ Sc_3 &= i, t_i, t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o, \\ Sc_4 &= i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16,17,o} \vdash o, \\ Sc_5 &= i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o. \end{aligned}$$

Mais uma vez é relevante notar que estes cenários são os mesmos cenários considerados na exemplificação do método para verificação do critério de correção *Soundness* para processos de *workflow* interorganizacionais, formalizado na seção 3.1. Assim, as árvores de prova construídas para provar tais sequentes, apresentadas no Apêndice A.1, podem ser consideradas, ou seja, reutilizadas.

O próximo passo, de acordo com o método para verificação do critério de correção *Relaxed Soundness* proposto, é analisar as árvores de prova produzidas para cada cenário Sc_i da $U(\text{IOWF-net})$ mostrada na Figura 7. Assim, considerando as árvores de prova para os cenários Sc_1 , Sc_2 , Sc_3 , Sc_4 e Sc_5 , mostradas no Apêndice seção A.1, é necessário verificar para quais cenários Sc_i a condição 1 é satisfeita e se a condição 2 é satisfeita considerando os cenários que satisfazem a condição 1.

Facilmente nota-se que o último sequente nas árvores de prova dos cenários Sc_1 , Sc_2 , Sc_3 e Sc_4 é $o \vdash o$. Desta maneira, as condições a) e b) do passo 1 são satisfeitas, isto é, apenas um átomo o foi produzido nestas árvores de prova e, como o último sequente é um sequente identidade, não há nenhum átomo disponível para consumo nestes cenários, ou seja, a execução destes cenários finaliza sem fichas remanescentes.

O último sequente do cenário Sc_5 é $tl, efpc, a5, fv, t_9, t_o \vdash o$. Nota-se, assim, que nenhum átomo o foi produzido nesse cenário. Logo, a condição a) do passo 1 não é verificada

para este cenário. Este sequente também possui quatro átomos disponíveis para consumo tl , $efpc$, $a5$ e fv , isto é, too_late , end_flow_PC , $a5$ e $final_version$. Consequentemente, este cenário também não satisfaz a condição b) do passo 1 de verificação.

Desta maneira, na segunda parte do processo de verificação, (passo 2), apenas os cenários Sc_1 , Sc_2 , Sc_3 e Sc_4 são considerados. Assim, é preciso verificar se cada transição $t \in T^* \cup \{t_i, t_o\}$ da $U(IOWF-net)$ aparece em pelo menos um destes cenários. A Tabela 5 mostra, para cada transição $t \in T^* \cup \{t_i, t_o\}$ da $U(IOWF-net)$, os cenários nos quais a transição t aparece, considerando somente os cenários Sc_1 , Sc_2 , Sc_3 e Sc_4 .

Tabela 5 – Relação entre transições e cenários.

Transição	Cenários
t_i	Sc_1, Sc_2, Sc_3, Sc_4
t_1	Sc_1, Sc_2, Sc_3, Sc_4
t_2	Sc_1, Sc_2, Sc_3, Sc_4
t_3	Sc_2, Sc_3, Sc_4
t_4	Sc_1
t_5	Sc_3, Sc_4
t_6	Sc_2
t_7	Sc_4
t_8	Sc_3
t_9	Sc_4
t_{10}	Sc_1, Sc_2, Sc_3, Sc_4
t_{11}	Sc_1, Sc_2, Sc_3, Sc_4
t_{12}	Sc_1, Sc_2, Sc_3, Sc_4
t_{13}	Sc_1
t_{14}	Sc_2, Sc_3, Sc_4
t_{15}	Sc_2, Sc_3
t_{16}	Sc_4
t_{17}	Sc_4
t_o	Sc_1, Sc_2, Sc_3, Sc_4

A Tabela 5 mostra que cada transição $t \in T^* \cup \{t_i, t_o\}$ aparece em pelo menos um cenário. Logo, a condição 2 também é satisfeita e a $U(IOWF-net)$ mostrada na Figura 7 é *Relaxed Sound*. Consequentemente, a IOWF-net mostrada na Figura 6 satisfaz o critério de correção *Relaxed Soundness*. Como consequência da prova do critério de correção *Relaxed Soundness*, tem-se que cada atividade do processo de *workflow* interorganizacional modelado pela IOWF-net mostrada na Figura 6 é contemplada em pelo menos um cenário que finaliza corretamente. Logo, têm-se a garantia da participação de todas as atividades no processo de *workflow* analisado.

Para finalizar, observa-se que, de fato, a reutilização dos cenários e árvores de prova da Lógica Linear ocorre. Além disso, é relevante salientar que o método proposto considerou a análise da $U(IOWF-net)$, desconsiderando a análise individual dos processos de *workflow* locais, isto é, das LWF-nets *Author* e *PC*.

3.3 Verificação do Critério de Correção *Weak Soundness* para Processos de *Workflow* Interorganizacionais

Uma vez que o critério de correção *Weak Soundness* implica na garantia da finalização do processo de *workflow* e na ausência de *deadlock* no mesmo, este critério também é relevante no contexto das IOWF-nets, especialmente nos casos em que o critério de correção *Soundness* não é satisfeito para uma dada IOWF-net. Como a $U(\text{IOWF-net})$ de uma IOWF-net tem a mesma estrutura de uma Workflow net, é possível verificar o critério de correção *Weak Soundness* para uma dada IOWF-net, considerando a análise da $U(\text{IOWF-net})$ correspondente. Assim, uma IOWF-net é *Weak Sound* quando a $U(\text{IOWF-net})$ da IOWF-net analisada satisfaz o critério de correção *Weak Soundness*.

Inicialmente, a $U(\text{IOWF-net})$ deve então ser representada através de fórmulas da Lógica Linear. A $U(\text{IOWF-net})$ pode ser representada por um ou mais sequentes da Lógica Linear, cada um representando um cenário possível da $U(\text{IOWF-net})$. Desta forma, para verificar o critério *Weak Soundness* para as IOWF-nets, é necessário definir e provar sequentes da Lógica Linear que representam os cenários da $U(\text{IOWF-net})$ correspondente.

Neste método, cada um destes cenários é então representado por um sequente linear específico, que considera a marcação inicial e final da $U(\text{IOWF-net})$ assim como uma lista não ordenada das transições envolvidas em cada cenário. Além disso, cada sequente linear tem apenas um átomo que representa a marcação inicial da $U(\text{IOWF-net})$.

Deve-se observar que o método proposto para verificar o critério de correção *Soundness* para IOWF-net, apresentado na seção 3.1, considera a construção e análise de árvores de prova da Lógica Linear que representam os cenários das LWF-nets e da $U(\text{IOWF-net})$ correspondente. O método proposto nesta seção para verificação do critério *Weak Soundness* para IOWF-nets considera a análise de árvores de prova da Lógica Linear que representam os cenários da $U(\text{IOWF-net})$ correspondente, isto é, a análise de um subconjunto dos cenários considerados na verificação de *Soundness*. Assim, quando uma abordagem baseada na Lógica Linear para verificar o critério de correção *Soundness* para IOWF-net, como a apresentada na seção 3.1, é considerada, se a IOWF-net analisada não é *Sound*, o subconjunto de sequentes da Lógica Linear que representam os cenários da $U(\text{IOWF-net})$ pode ser reutilizado no contexto desta abordagem, executando apenas o passo 1 de análise das árvores de prova construídas para cada cenário da $U(\text{IOWF-net})$, para decidir o critério de correção *Weak Soundness* para a IOWF-net analisada. É importante destacar ainda que as condições 1 e 2 apresentadas no método para verificação do critério de correção *Soundness* na seção 3.1 são enfraquecidas, uma vez que o critério *Weak Soundness* simplesmente desconsidera a verificação das condições 1.c) e 2 que estão relacionadas à verificação da ausência de tarefas mortas no modelo.

Assim, considerando as árvores de prova para os sequentes da Lógica Linear que representam cada cenário da $U(\text{IOWF-net})$ analisada, tais árvores de prova devem ser analisadas respeitando o seguinte passo:

1. para cada árvore de prova que representa um cenário da $U(\text{IOWF-net})$:
 - a) se apenas um átomo o , que corresponde a uma ficha no lugar de término da $U(\text{IOWF-net})$ analisada foi produzido na árvore de prova (este fato é representado na árvore de prova pelo sequente identidade $o \vdash o$) então o primeiro requisito para a prova de *Weak Soundness* é verificado, isto é, apenas uma ficha aparece no lugar o ;
 - b) se não há nenhum átomo disponível para consumo no último sequente da árvore de prova então significa que todos os lugares da $U(\text{IOWF-net})$ analisada estão vazios, isto é, o segundo requisito para a prova de *Weak Soundness* é verificado.

Se a condição 1 acima é satisfeita, a $U(\text{IOWF-net})$ é *Weak Sound* e, conseqüentemente, a IOWF-net analisada satisfaz o critério de correção *Weak Soundness*.

É relevante salientar que o método proposto para a verificação do critério de correção *Weak Soundness* proposto nesta seção não considera a verificação individual dos processos de *workflow* locais envolvidos no processo de *workflow* interorganizacional, isto é, não considera a verificação individual das LWF-nets que compõem a IOWF-net. Em vez disso, o método proposto considera a verificação do modelo global, representado pela $U(\text{IOWF-net})$. Além disso, o método considera a estrutura acíclica do modelo analisado.

Para ilustrar o método proposto, a IOWF-net mostrada na Figura 6 é considerada. Para provar o critério *Weak Soundness* para esta IOWF-net, a $U(\text{IOWF-net})$ correspondente, mostrada na Figura 7, é considerada. Para esta rede, é necessário provar cinco sequentes da Lógica Linear, cada um representando um cenário diferente.

O primeiro cenário, Sc_1 , é o cenário onde a tarefa *send_reject* será executada (disparo da transição *send_reject*). O segundo cenário, Sc_2 , é o cenário no qual as tarefas *too_late* e *receive_notification_1* serão executadas (disparo das transições *too_late* e *receive_notification_1*). Já o terceiro cenário, Sc_3 , corresponde ao cenário onde as tarefas *too_late* e *receive_notification_2* são realizadas (disparo das transições *too_late* e *receive_notification_2*). As tarefas *send_final_version* e *receive_final_version* (disparo das transições *send_final_version* e *receive_final_version*) são consideradas no cenário Sc_4 . Finalmente, o quinto cenário, Sc_5 , é o cenário onde as tarefas *too_late* e *send_final_version* são executadas (disparo das transições *too_late* e *send_final_version*). Estes cenários são dados pelos sequentes da Lógica Linear a seguir:

$$Sc_1 = i, t_i, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o,$$

$$Sc_2 = i, t_i, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o,$$

$$Sc_3 = i, t_i, t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o,$$

$$Sc_4 = i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16,17,o} \vdash o,$$

$$Sc_5 = i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o.$$

Mais uma vez, nota-se que estes cenários são os mesmos cenários considerados na exemplificação do método para verificação do critério de correção *Soundness* para processos de *workflow* interorganizacionais formalizado na seção 3.1. Assim, as árvores de prova construídas para provar tais sequentes, apresentadas no Apêndice A.1, podem ser consideradas, ou seja, reutilizadas.

O próximo passo, de acordo com o método para verificação do critério de correção *Weak Soundness* apresentado, é analisar as árvores de prova produzidas para cada cenário Sc_i da $U(IOWF\text{-}net)$ mostrada na Figura 7. Assim, deve-se analisar as árvores de prova para os cenários Sc_1 , Sc_2 , Sc_3 , Sc_4 e Sc_5 , mostradas no Apêndice A.1, de forma a verificar se a condição 1 é satisfeita.

Procedendo com esta análise, nota-se que o último sequente nas árvores de prova dos cenários Sc_1 , Sc_2 , Sc_3 e Sc_4 é $o \vdash o$. Desta maneira, as condições a) e b) do passo 1 são satisfeitas, isto é, apenas um átomo o foi produzido nestas árvores de prova e, como o último sequente é um sequente identidade, não há nenhum átomo disponível para consumo nestes cenários.

No entanto, o mesmo não ocorre para o cenário Sc_5 , onde a condição 1 não é satisfeita, uma vez que na árvore de prova para o cenário Sc_5 da $U(IOWF\text{-}net)$:

- a) nenhum átomo o foi produzido na árvore de prova, ou seja, o sequente identidade $o \vdash o$ não aparece na árvore de prova. Isso significa que nenhuma ficha foi produzida no lugar de término deste cenário, ou seja, para este cenário, o processo de *workflow* interorganizacional não finaliza;
- b) há quatro átomos disponíveis para consumo no último sequente da árvore de prova construída para o cenário Sc_5 : *too_late*, *end_flow_pc*, *a5* e o átomo *final_version*.

Como a condição 1 não é satisfeita para o cenário Sc_5 , tem-se que a $U(IOWF\text{-}net)$ da Figura 7 não satisfaz o critério de correção *Weak Soundness* e, conseqüentemente, a $IOWF\text{-}net$ mostrada na Figura 6 não é *Weak Sound*, ou seja, o sistema representado por este processo de *workflow* interorganizacional não é livre de *deadlock*.

Para finalizar, observa-se que, de fato, a reutilização dos cenários e árvores de prova da Lógica Linear ocorre no método apresentando nesta seção quando, *a priori*, uma abordagem baseada na Lógica Linear é considerada para verificação do critério de correção *Soundness* para processos de *workflow* interorganizacionais.

3.4 Detecção de Cenários Livres de *Deadlock* em Processos de *Workflow* Interorganizacionais

O método apresentado nessa seção tem por objetivo identificar todos os cenários livres de *deadlock* em um processo de *workflow* interorganizacional, permitindo assim que as organizações envolvidas neste processo interorganizacional possam analisar tais cenários e verificar se os mesmos satisfazem suas necessidades de negócio. Utilizando este método, as organizações envolvidas no processo de *workflow* interorganizacional podem verificar se seus principais serviços de negócio finalizam corretamente, evitando as situações de *deadlock*, sem a necessidade de revisar seus processos de *workflow* locais ou globais, com o objetivo de obter um conjunto de processos livre de *deadlock*.

Diferentemente do que acontece com os critérios de correção *Soundness* e *Weak Soundness*, o método apresentado nesta seção não garante a ausência total de *deadlock* no processo de *workflow* interorganizacional analisado. Além disso, diferentemente do que acontece com os critérios de correção *Soundness* e *Relaxed Soundness*, não há garantia da participação de todas as atividades envolvidas no processo analisado. No entanto, este método identifica todos os cenários que são livres de *deadlock* em um processo de *workflow* interorganizacional. Se estes cenários identificados satisfazem as necessidades de negócio entre as organizações envolvidas no processo de *workflow* interorganizacional e se somente estes cenários são seguidos, o sistema global satisfará as necessidades de negócio estabelecidas sem ser levado a situações de bloqueio. Ou seja, o presente método não se preocupa com a ausência total de *deadlock* no processo global, nem mesmo com a participação de todas as atividades do processo global. Entretanto, o método identifica as possibilidades de comunicação entre os processos de *workflow* locais que não levam o processo global a situações de *deadlock*. A Tabela 6 apresenta a comparação entre os diferentes tipos de *Soundness* e o resultado do método para detecção de cenários livres de *deadlock* apresentado nesta seção.

Tabela 6 – Comparação entre os diferentes tipos de *Soundness* e o resultado do Método para Detecção de Cenários Livres de *Deadlock* em Processos de *Workflow* Interorganizacionais.

Característica	<i>Soundness</i>	<i>Relaxed Soundness</i>	<i>Weak Soundness</i>	Cenários Livres de <i>Deadlock</i>
Correta Finalização	Sim	Sim	Sim	Sim
Ausência de <i>Deadlock</i>	Sim	Não	Sim	Não
Participação de todas as atividades	Sim	Sim	Não	Não

Este método é baseado na prova de sequentes da Lógica Linear que consideram os lugares de comunicação assíncrona – lugares de comunicação de entrada (ICPs) e lugares de comunicação de saída (OCPs) – dos cenários das LWF-nets que compõem a IOWF-net analisada. Os átomos do tipo ICP representam as informações necessárias para a execução de um dado cenário e os átomos do tipo OCP representarão as informações produzidas e

não consumidas por tal cenário. A Figura 13 apresenta uma visão geral do método para detecção de cenários livres de *deadlock* em processos de *workflow* interorganizacionais, formalizado na sequência.

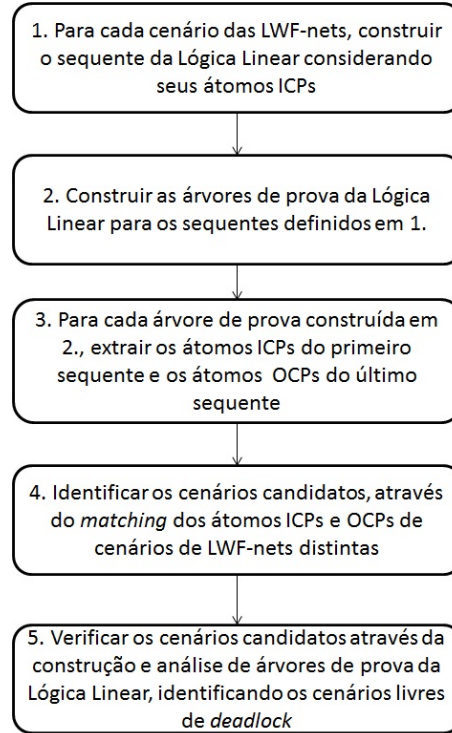


Figura 13 – Método para detecção de cenários livres de *deadlock* em processos de *workflow* interorganizacionais.

Assim, considerando uma IOWF-net que modela um processo de *workflow* interorganizacional, inicialmente é necessário definir e analisar todos os cenários das LWF-nets que compõem a IOWF-net, considerando os lugares de comunicação assíncrona existentes na IOWF-net. É importante recordar que o método apresentado para verificação do critério de correção *Soundness* desconsidera os lugares de comunicação assíncrona. No presente método, cada um destes cenários é então inicialmente representado por um sequente da Lógica Linear que considera as marcações inicial e final da LWF-net analisada, uma lista não ordenada das transições envolvidas no cenário e os átomos que representam os lugares de comunicação de entrada (ICPs) das transições do cenário.

Após a definição dos sequentes da Lógica Linear que representam os cenários de cada LWF-net, estes sequentes precisam ser provados. Para prová-los, árvores de prova da Lógica Linear são então construídas.

Considerando as árvores de prova da Lógica Linear construídas, deve-se destacar dois sequentes principais: o primeiro e o último, lembrando sempre que as árvores de prova são construídas de forma *bottom-up*. O primeiro sequente linear da árvore de prova possui os átomos do tipo ICP necessários para a correta execução do cenário que o sequente

representa. Já o último sequente linear da árvore de prova mostrará os átomos do tipo OCP produzidos que não foram consumidos por tal cenário.

Inicialmente, se somente dois processos de *workflow* locais, ou seja somente duas LWF-nets, PN_1 e PN_2 , compõem o processo de *workflow* interorganizacional a ser analisado, isto é, a IOWF-net, deve-se então selecionar todos os pares de cenários Sc_i e Sc_j , tais que Sc_i é cenário de PN_1 e Sc_j é cenário de PN_2 , onde os átomos ICPs de Sc_i são iguais aos átomos OCPs de Sc_j e os átomos ICPs do cenário Sc_j são iguais aos átomos OCPs do cenário Sc_i . Ou seja, as mensagens produzidas por um cenário devem corresponder exatamente às mensagens consumidas pelo outro cenário e vice-versa. A composição destes cenários, isto é, o cenário $Sc_i \times Sc_j$, é um cenário candidato, que pode eventualmente ser um cenário livre de *deadlock*.

É importante destacar que esta seleção significa, no contexto dos processos de *workflow* interorganizacionais, que todas as informações (via canais de comunicação) enviadas pelos processos de *workflow* locais são recebidas e utilizadas por outros processos locais que, em conjunto, compõem o processo global. Ou seja, nenhuma mensagem de comunicação pode ser produzida e não consumida pelo conjunto de cenários selecionado, o que, de fato, deve ocorrer em um ambiente interorganizacional.

Se o processo de *workflow* interorganizacional a ser analisado possui três processos de *workflow* locais, isto é, a IOWF-net possui três LWF-nets, PN_1 , PN_2 e PN_3 , deve-se então selecionar todos os cenários Sc_i , Sc_j e Sc_k , onde Sc_i é cenário da LWF-net PN_1 , Sc_j é cenário da LWF-net PN_2 , Sc_k é cenário da LWF-net PN_3 e i, j, k são índices que representam os cenários das respectivas LWF-nets tais que $ICP(Sc_j) = OCP(Sc_i) \cup OCP(Sc_k)$ e $OCP(Sc_j) = ICP(Sc_i) \cup ICP(Sc_k)$. Desta forma, os átomos ICPs do cenário Sc_j , ou seja, as mensagens que são recebidas do meio externo e processadas pelo cenário Sc_j , devem ser iguais à união dos átomos OCPs dos cenários Sc_i e Sc_k , ou seja, devem corresponder às mensagens produzidas pelos cenários Sc_i e Sc_k . Da mesma forma, os átomos OCPs do cenário Sc_j , ou seja, as mensagens que são produzidas por este cenário e enviadas ao meio externo, devem ser iguais à união dos átomos ICPs dos cenários Sc_i e Sc_k , ou seja, devem corresponder às mensagens necessárias para a execução dos cenários Sc_i e Sc_k . A composição destes cenários, isto é, o cenário $Sc_i \times Sc_j \times Sc_k$, é um cenário candidato, que pode eventualmente ser um cenário livre de *deadlock*.

De forma generalizada, deve-se então selecionar todos os cenários Sc_{i1} , Sc_{j2} , Sc_{k3} , ..., Sc_{mn} onde n é o número de LWF-nets da IOWF-net analisada e os índices i, j, k, \dots, m correspondem aos cenários das LWF-nets tais que $ICP(Sc_{i1}) \cup ICP(Sc_{j2}) \cup ICP(Sc_{k3}) \cup \dots \cup ICP(Sc_{mn}) = OCP(Sc_{i1}) \cup OCP(Sc_{j2}) \cup OCP(Sc_{k3}) \cup \dots \cup OCP(Sc_{mn})$, ou seja, o conjunto obtido pela união de todos os átomos ICPs dos cenários envolvidos é igual ao conjunto obtido pela união de todos os átomos OCPs destes cenários. A composição destes cenários, isto é, o cenário $Sc_{i1} \times Sc_{j2} \times Sc_{k3} \times \dots \times Sc_{mn}$, é um cenário candidato, que pode eventualmente ser um cenário livre de *deadlock*.

Para verificar se um cenário candidato genérico $Sc_i1 \times Sc_j2 \times Sc_k3 \times \dots \times Sc_mn$ é de fato um cenário livre de *deadlock*, é necessário provar um sequente linear que considera as listas de transições dos cenários Sc_i1 , Sc_j2 , Sc_k3 , ..., Sc_mn , as transições t_i e t_o e os lugares de entrada e saída globais, respectivamente i e o , da $U(\text{IOWF-net})$. Assim, o sequente linear a ser provado é dado por: $i, t_i, \alpha_i1, \alpha_j2, \alpha_k3, \dots, \alpha_mn, t_o \vdash o$, onde α_i1 é a lista de transições do cenário Sc_i1 , α_j2 é a lista de transições do cenário Sc_j2 , α_k3 é a lista de transições do cenário Sc_k3 e, conseqüentemente, α_mn é a lista de transições do cenário Sc_mn . É importante notar que a lista de transições no sequente linear é não ordenada.

Após a construção das árvores de prova da Lógica Linear, que consideram as provas dos cenários candidatos, estas árvores de prova devem ser analisadas seguindo os seguintes passos:

- a) se apenas um átomo o , que corresponde a um átomo no lugar de término da $U(\text{IOWF-net})$, foi produzido na árvore de prova (isto é representado na árvore de prova pelo sequente identidade $o \vdash o$), significa que o cenário candidato pode finalizar corretamente;
- b) se não há nenhum átomo disponível para consumo no último sequente linear da árvore de prova, significa que quando a execução do cenário finaliza, não há nenhuma tarefa remanescente a ser tratada.

Se as condições a) e b) acima são satisfeitas, o cenário candidato analisado é um cenário livre de *deadlock*. Além disso, o cenário finaliza sem deixar tarefas remanescentes a serem tratadas. É importante observar que as condições a) e b) não consideram a verificação das transições da IOWF-net e, conseqüentemente, das tarefas envolvidas no processo de *workflow* interorganizacional, como acontece com os métodos para verificação dos critérios de correção *Soundness* e *Relaxed Soundness* apresentados nas seções anteriores. Além disso, a verificação acontece somente para os cenários candidatos, ou seja, nada se sabe a respeito dos cenários não candidatos. Assim, os cenários não candidatos podem levar o sistema a situações de *deadlock*, por exemplo. Desta forma, diferentemente do que acontece com a verificação de *Weak Soundness*, não há garantia de ausência de *deadlock* no modelo global analisado.

Para exemplificação dos lugares de comunicação ICP e OCP, a Figura 14 apresenta uma visão que destaca os lugares de comunicação da IOWF-net apresentada na Figura 6. Os lugares do tipo ICP são representados pelos lugares nas fronteiras, preenchidos por pontilhados. Já os lugares de comunicação de saída (OCPs) são os lugares de fronteira preenchidos por tracejados verticais. É importante observar que, na verdade, não há a duplicação dos lugares de comunicação, sendo que a figura apresentada tem por objetivo simplesmente tornar mais clara a apresentação do exemplo. Assim, um cenário da LWF-net *PC* pode ser representado, por exemplo, por *draft, start_flow_PC, receive_draft, send_ack_draft, evaluate, send_reject* \vdash *end_flow_PC*. O átomo *draft* neste sequente linear representa o ICP da transição *receive_draft* (a única transição deste cenário

rio que possui um átomo do tipo ICP). O átomo *start_flow_PC* representa a marcação inicial da LWF-net *PC*, *receive_draft*, *send_ack_draft*, *evaluate* e *send_reject* representam as fórmulas das transições envolvidas nesse cenário, e *end_flow_PC* representa a marcação final a ser alcançada pela LFW-net *PC*.

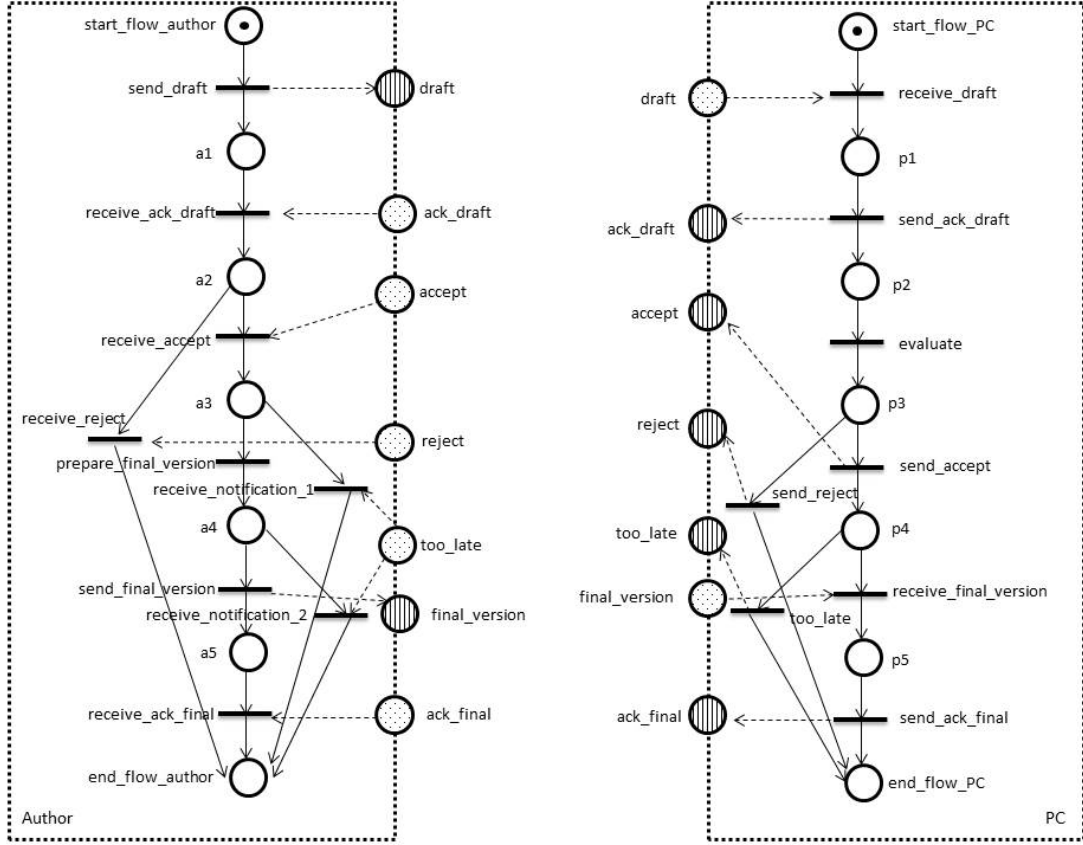


Figura 14 – IOWF-net com lugares de comunicação ICPs e OCPs em destaque.

Para exemplificação do método apresentado, considera-se a verificação do processo de *workflow* interorganizacional mostrado na Figura 6. Considerando a LWF-net *Author*, quatro diferentes cenários e, conseqüentemente, sequentes da Lógica Linear são definidos:

$$Sc_1A = ad, r, sfa, t_1, t_2, t_4 \vdash efa,$$

$$Sc_2A = ad, a, af, sfa, t_1, t_2, t_3, t_5, t_7, t_9 \vdash efa,$$

$$Sc_3A = ad, a, tl, sfa, t_1, t_2, t_3, t_6 \vdash efa,$$

$$Sc_4A = ad, a, tl, sfa, t_1, t_2, t_3, t_5, t_8 \vdash efa.$$

Já para a LWF-net *PC* mostrada na Figura 6, existem três cenários diferentes:

$$Sc_1PC = d, sfpc, t_{10}, t_{11}, t_{12}, t_{14}, t_{15} \vdash efpc,$$

$$Sc_2PC = d, fv, sfpc, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17} \vdash efpc,$$

$$Sc_3PC = d, sfpc, t_{10}, t_{11}, t_{12}, t_{13} \vdash efpc.$$

Após a definição dos sequentes da Lógica Linear que representam cada cenário das LWF-nets *Author* e *PC*, árvores de prova da Lógica Linear devem ser construídas para prová-los. Tais árvores de prova são apresentadas no Apêndice A.2. Somente para exemplificação, a árvore de prova para o cenário Sc_1A é apresentada na sequência.

$$\begin{array}{c}
\frac{\frac{a2 \vdash a2}{a2, r \vdash a2 \otimes r} \otimes_R \quad d, efa \vdash efa}{\quad} \multimap_L \\
\frac{\frac{a1 \vdash a1}{a1, ad \vdash a1 \otimes ad} \otimes_R \quad r, d, a2, a2 \otimes r \multimap efa \vdash efa}{\quad} \multimap_L \\
\frac{\quad}{ad, r, a1, d, a1 \otimes ad \multimap a2, t_4 \vdash efa} \otimes_L \\
\frac{sfa \vdash sfa \quad ad, r, a1 \otimes d, t_2, t_4 \vdash efa}{\quad} \multimap_L \\
ad, r, sfa, sfa \multimap a1 \otimes d, t_2, t_4 \vdash efa
\end{array}$$

Considerando que as árvores de prova da Lógica Linear são construídas de forma *bottom-up*, o primeiro sequente da árvore de prova para o cenário Sc_1A é dado por $ad, r, sfa, sfa \multimap a1 \otimes d, t_2, t_4 \vdash efa$ e o último sequente desta árvore de prova é $d, efa \vdash efa$. Os átomos $ad = ack_draft$ e $r = reject$ no primeiro sequente correspondem aos átomos do tipo ICP para o cenário Sc_1A . Isso significa que, para funcionar corretamente, este cenário deve consumir estes dois átomos. O átomo $d = draft$ no último sequente corresponde ao átomo produzido pelo cenário Sc_1A , átomo este que não foi consumido por nenhuma fórmula de transição neste cenário, isto é, um átomo remanescente na árvore de prova, que corresponde ao átomo do tipo OCP para este cenário.

Para a árvore de prova para o cenário Sc_2A , apresentada no Apêndice A.2, o primeiro sequente é dado por $ad, a, af, sfa, sfa \multimap a1 \otimes d, t_2, t_3, t_5, t_7, t_9 \vdash efa$ e o último sequente é $d, fv, efa \vdash efa$. Os átomos $ad = ack_draft$, $a = accept$ e $af = ack_final$ no primeiro sequente correspondem aos átomos do tipo ICP para o cenário Sc_2A . Isso significa que, para funcionar corretamente, este cenário deve consumir estes três átomos. Os átomos $d = draft$ e $fv = final_version$ no último sequente correspondem aos átomos produzidos pelo cenário Sc_2A , átomos estes que não foram consumidos por nenhuma fórmula de transição neste cenário, isto é, átomos remanescentes na árvore de prova, que correspondem aos átomos do tipo OCP para este cenário.

Já para a árvore de prova para o cenário Sc_3A , apresentada no Apêndice A.2, tem-se que o primeiro sequente é dado por $ad, a, tl, sfa, sfa \multimap a1 \otimes d, t_2, t_3, t_6 \vdash efa$ e o último sequente é $d, fv, efa \vdash efa$. Os átomos $ad = ack_draft$, $a = accept$ e $tl = too_late$ no primeiro sequente correspondem aos ICPs para o cenário Sc_3A . Isso significa que, para funcionar corretamente, este cenário deve consumir estes três átomos. Os átomos $d = draft$ e $fv = final_version$ no último sequente correspondem aos átomos produzidos pelo cenário Sc_3A e que não foram consumidos por nenhuma fórmula de transição neste cenário, isto é, estes átomos são átomos remanescentes na árvore de prova e correspondem aos átomos OCPs para este cenário.

No que se diz respeito à árvore de prova para o cenário Sc_4A , apresentada no Apêndice A.2, tem-se que o primeiro sequente da árvore de prova é dado por $ad, a, tl, sfa, sfa \multimap a1 \otimes d, t_2, t_3, t_5, t_8 \vdash efa$ e o último sequente é dado por $d, efa \vdash efa$. Os átomos $ad = ack_draft$, $a = accept$ e $tl = too_late$ no primeiro sequente correspondem aos ICPs para o cenário Sc_4A . Isso significa que, para funcionar corretamente, este cenário

deve consumir estes três átomos. O átomo $d = draft$ no último sequente corresponde ao átomo produzido pelo cenário Sc_4A , que não foi consumido por nenhuma fórmula de transição neste cenário, isto é, trata-se de um átomo remanescente na árvore de prova, que corresponde ao átomo OCP para este cenário.

Considerando a árvore de prova para o cenário Sc_1PC , apresentada no Apêndice A.2, tem-se que o primeiro sequente desta árvore de prova é dado por $d, sfpc, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{14}, t_{15} \vdash efpc$ e o último sequente é dado por $ad, a, tl, efpc \vdash efpc$. O átomo $d = draft$ no primeiro sequente corresponde ao ICP para o cenário Sc_1PC . Isso significa que, para funcionar corretamente, este cenário deve consumir este átomo. Os átomos $ad = ack_draft$, $a = accept$ e $tl = too_late$, no último sequente, correspondem aos átomos produzidos pelo cenário Sc_1PC . Estes átomos não foram consumidos por nenhuma fórmula de transição neste cenário, isto é, são átomos remanescentes na árvore de prova, que correspondem aos átomos OCPs para este cenário.

Para a árvore de prova para o cenário Sc_2PC , apresentada no Apêndice A.2, tem-se que o primeiro sequente é $d, fv, sfpc, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{14}, t_{16}, t_{17} \vdash efpc$ e o último sequente é dado por $ad, a, af, efpc \vdash efpc$. Os átomos $d = draft$ e $fv = final_version$ no primeiro sequente correspondem aos ICPs para o cenário Sc_2PC . Isso significa que, para funcionar corretamente, este cenário deve consumir estes dois átomos. Os átomos $ad = ack_draft$, $a = accept$ e $af = ack_final$, no último sequente, correspondem aos átomos produzidos pelo cenário Sc_2PC . Estes átomos não foram consumidos por nenhuma fórmula de transição neste cenário, isto é, são átomos remanescentes na árvore de prova e correspondem aos átomos OCPs para este cenário.

Finalmente, considerando a árvore de prova para o cenário Sc_3PC , apresentada no Apêndice A.2, tem-se que o primeiro sequente desta árvore de prova é dado por $d, sfpc, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{13} \vdash efpc$ e o último sequente desta árvore de prova é dado por $ad, r, efpc \vdash efpc$. O átomo $d = draft$ no primeiro sequente corresponde ao átomo do tipo ICP para o cenário Sc_3PC . Isso significa que, para funcionar corretamente, este cenário deve consumir este átomo. Os átomos $ad = ack_draft$ e $r = reject$ no último sequente correspondem aos átomos produzidos pelo cenário Sc_3PC . Estes átomos não foram consumidos por nenhuma fórmula de transição neste cenário, isto é, são átomos remanescentes na árvore de prova e correspondem aos átomos do tipo OCP para este cenário.

Considerando as árvores de prova apresentadas no Apêndice A.2 e as considerações apresentadas em relação às árvores de prova, nota-se que os átomos do tipo ICP para os cenários Sc_3A e Sc_4A , isto é $ad = ack_draft$, $a = accept$ e $tl = too_late$ são iguais aos átomos do tipo OCP do cenário Sc_1PC . Ocorre também que o átomo do tipo ICP para o cenário Sc_1PC , $d = draft$, é igual ao átomo do tipo OCP dos cenários Sc_3A e Sc_4A . Ou seja, $IPC(Sc_3A) \cup IPC(Sc_1PC) = \{ad, a, tl, d\}$ e $OPC(Sc_3A) \cup OPC(Sc_1PC) = \{d, ad, a, tl\}$. Como $\{ad, a, tl, d\} = \{d, ad, a, tl\}$, tem-se que o cenário $Sc_3A \times Sc_1PC$ é um

cenário candidato a ser livre de *deadlock*. Além disso, como $ICP(S_{c_4A}) \cup ICP(S_{c_1PC}) = \{ad, a, tl, d\} = OCP(S_{c_4A}) \cup OCP(S_{c_1PC}) = \{d, ad, a, tl, \}$, tem-se que $S_{c_4A} \times S_{c_1PC}$ também é cenário candidato a ser livre de *deadlock*.

O mesmo ocorre com os cenários S_{c_2A} e S_{c_2PC} . Os ICPs para o cenário S_{c_2A} são $ad = ack_draft$, $a = accept$, e $af = ack_final$. Os OCPs para o cenário S_{c_2PC} também são ad , a e af . Os ICPs para o cenário S_{c_2PC} são exatamente os mesmos que os OCPs do cenário S_{c_2A} : $d = draft$ e $fv = final_version$. Logo, $ICP(S_{c_2A}) \cup ICP(S_{c_2PC}) = \{ad, a, af, d, fv\} = OCP(S_{c_2A}) \cup OCP(S_{c_2PC}) = \{d, fv, ad, a, af\}$. Assim, o cenário $S_{c_2A} \times S_{c_2PC}$ é um cenário candidato a ser livre de *deadlock*.

A mesma situação pode ser facilmente verificada para os cenários S_{c_1A} e S_{c_3PC} . Os ICPs para o cenário S_{c_1A} são $ad = ack_draft$ e $r = reject$, que são os OCPs do cenário S_{c_3PC} . O ICP do cenário S_{c_3PC} é $d = draft$, que é o OCP do cenário S_{c_1A} . Desta forma, $ICP(S_{c_1A}) \cup ICP(S_{c_3PC}) = \{ad, r, d\} = OCP(S_{c_1A}) \cup OCP(S_{c_3PC}) = \{a, ad, r\}$. Logo, o cenário $S_{c_1A} \times S_{c_3PC}$ também é um cenário candidato a ser livre de *deadlock*. A Figura 15 mostra à esquerda o cenário S_{c_1A} e à direita o cenário S_{c_3PC} , cuja composição resulta no cenário candidato $S_{c_1A} \times S_{c_3PC}$, onde é possível verificar as questões referentes aos lugares de comunicação de entrada e de saída discutidas.

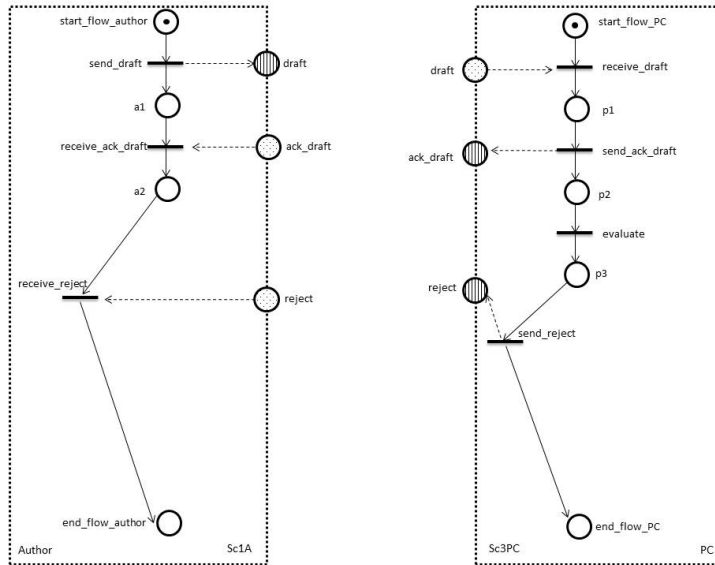


Figura 15 – Cenários S_{c_1A} e S_{c_3PC} , que compõem o cenário candidato $S_{c_1A} \times S_{c_3PC}$.

Sumarizando, há quatro cenários candidados: $S_{c_3A} \times S_{c_1PC}$, $S_{c_4A} \times S_{c_1PC}$, $S_{c_2A} \times S_{c_2PC}$ e $S_{c_1A} \times S_{c_3PC}$.

Para verificar se os cenários candidados acima são, de fato, cenários livres de *deadlock*, é necessário provar os seguintes sequentes da Lógica Linear:

$$S_{c_3A} \times S_{c_1PC} = i, t_i, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o,$$

$$S_{c_4A} \times S_{c_1PC} = i, t_i, t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o,$$

$$S_{c_2A} \times S_{c_2PC} = i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, t_o \vdash o,$$

$$Sc_1A \times Sc_3PC = i, t_i, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o.$$

As árvores de prova construídas para provar estes sequentes são apresentadas no Apêndice A.2.

Novamente, considerando que as árvores de prova da Lógica Linear são construídas de forma *bottom-up*, o primeiro sequente da árvore de prova para o cenário $Sc_3A \times Sc_1PC$, apresentada no Apêndice A.2, é dado por $i, t_i, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o$ e o último sequente desta árvore de prova é o sequente identidade $o \vdash o$.

Considerando a árvore de prova para o cenário $Sc_4A \times Sc_1PC$, apresentada no Apêndice A.2, tem-se que o primeiro sequente é dado por $i, t_i, t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o$ e o último sequente desta árvore de prova é o sequente identidade $o \vdash o$.

Já para a árvore de prova para o cenário $Sc_2A \times Sc_2PC$, tem-se que o primeiro sequente da árvore de prova para este cenário é dado por $i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, t_o \vdash o$ e o último sequente desta árvore de prova é o sequente identidade $o \vdash o$.

Finalmente, considerando a árvore de prova para o cenário $Sc_1A \times Sc_3PC$, tem-se que o primeiro sequente da árvore de prova para este cenário é dado por $i, t_i, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o$ e o último sequente desta árvore de prova é o sequente identidade $o \vdash o$.

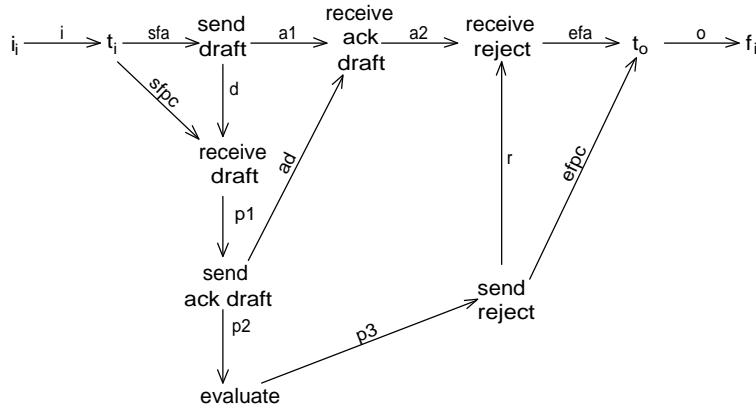
Seguindo os passos para verificar se estes cenários candidatos são de fato livres de *deadlock*, as árvores de prova devem ser analisadas com o objetivo de mostrar se condições a) e b) são satisfeitas. É fácil notar que o último sequente linear de todas as árvores de prova é $o \vdash o$. Assim, apenas um átomo o foi produzido em cada árvore de prova, o que significa que os cenários analisados podem finalizar corretamente, ou seja, a condição a) é satisfeita. Além disso, como o último sequente das árvores de prova é somente $o \vdash o$, não há nenhum átomo disponível para consumo, já que se trata de um sequente identidade. Logo, a condição b) também é satisfeita para todos os cenários candidatos. Desta forma, os cenários $Sc_3A \times Sc_1PC$, $Sc_4A \times Sc_1PC$, $Sc_2A \times Sc_2PC$ e $Sc_1A \times Sc_3PC$ são, de fato, cenários livres de *deadlock*.

É importante observar que estes cenários livres de *deadlock* podem prover todas as necessidades de negócio existentes entre um autor e o comitê de programa. A visualização destes cenários fica mais clara quando considera-se o grafo de precedência associado aos mesmos. Para a construção de tais grafos, deve-se considerar as árvores de prova da Lógica Linear rotuladas, apresentadas no Apêndice A.2.1.

Assim, considerando o processo de *workflow* interorganizacional modelado pela IOWF-net da Figura 6, o cenário $Sc_1A \times Sc_3PC$ é aquele no qual um artigo é rejeitado. Para exemplificação, a árvore de prova rotulada para este cenário é apresentada na sequência. O grafo de precedência associado ao cenário $Sc_1A \times Sc_3PC$ é apresentado na Figura 16.

A árvore de prova rotulada para o cenário $Sc_1A \times Sc_3PC$ é dada por:

$$\begin{array}{c}
 \frac{\frac{t_4 \quad t_o \quad t_{13} \quad t_o}{efa \vdash efa \quad efpc \vdash efpc} \otimes_R \quad \frac{t_o \quad f_i}{o \vdash o} \multimap_L}{\frac{t_4 \quad t_{13} \quad t_o \quad t_o}{efa, efpc \vdash efa \otimes efpc} \multimap_L} \\
 \\
 \frac{\frac{t_2 \quad t_4 \quad t_{13} \quad t_4}{a2 \vdash a2 \quad r \vdash r} \otimes_R \quad \frac{t_{13} \quad t_4}{efpc, efa, efa \otimes efpc \multimap o \vdash o} \multimap_L}{\frac{t_2 \quad t_{13} \quad t_4 \quad t_4}{a2, r \vdash a2 \otimes r} \multimap_L} \\
 \\
 \frac{\frac{t_1 \quad t_2 \quad t_{11} \quad t_2}{a1 \vdash a1 \quad ad \vdash ad} \otimes_R \quad \frac{t_{13} \quad t_{13} \quad t_2}{r, efpc, a2, a2 \otimes r \multimap efa, t_o \vdash o} \multimap_L}{\frac{t_1 \quad t_{11} \quad t_2 \quad t_2}{a1, ad \vdash a1 \otimes ad} \multimap_L} \\
 \\
 \frac{t_1 \quad t_{11} \quad t_{13} \quad t_{13}}{a1, ad, r, efpc, a1 \otimes ad \multimap a2, t_4, o \vdash o} \otimes_L \\
 \\
 \frac{t_{12} \quad t_{13} \quad t_1 \quad t_{11} \quad t_{13} \quad t_{13}}{p3 \vdash p3 \quad a1, ad, r \otimes efpc, t_2, 4, o \vdash o} \multimap_L \\
 \\
 \frac{t_{11} \quad t_{12} \quad t_1 \quad t_{11} \quad t_{12}}{p2 \vdash p2 \quad a1, ad, p3, p3 \multimap r \otimes efpc, t_2, 4, o \vdash o} \multimap_L \\
 \\
 \frac{t_1 \quad t_{11} \quad t_{11}}{a1, ad, p2, p2 \multimap p3, t_2, 4, 13, o \vdash o} \otimes_L \\
 \\
 \frac{t_{10} \quad t_{11} \quad t_1 \quad t_{11} \quad t_{11}}{p1 \vdash p1 \quad a1, ad \otimes p2, t_2, 4, 12, 13, o \vdash o} \multimap_L \\
 \\
 \frac{\frac{t_i \quad t_{10} \quad t_1 \quad t_{10}}{sfpc \vdash sfpc \quad d \vdash d} \otimes_R \quad \frac{t_1 \quad t_{10}}{a1, p1, p1 \multimap ad \otimes p2, t_2, 4, 12, 13, o \vdash o} \multimap_L}{\frac{t_i \quad t_1 \quad t_{10} \quad t_{10}}{sfpc, d \vdash sfpc \otimes d} \multimap_L} \\
 \\
 \frac{t_i \quad t_1 \quad t_1}{sfpc, a1, d, sfpc \otimes d \multimap p1, t_2, 4, 11, 12, 13, o \vdash o} \otimes_L \\
 \\
 \frac{t_i \quad t_1 \quad t_1}{sfa \vdash sfa \quad sfpc, a1 \otimes d, t_2, 4, 10, 11, 12, 13, o \vdash o} \multimap_L \\
 \\
 \frac{t_i \quad t_i}{sfa, sfpc, sfa \multimap a1 \otimes d, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o} \otimes_L \\
 \\
 \frac{i_i \quad t_i \quad t_i}{i \vdash i \quad sfa \otimes sfpc, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o} \multimap_L \\
 \\
 i_i \quad i, i \multimap sfa \otimes sfpc, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o
 \end{array}$$


 Figura 16 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_1A \times Sc_3PC$.

O cenário $Sc_2A \times Sc_2PC$ é o cenário no qual o artigo é aceito, a versão final é recebida pelo comitê de programa e o artigo publicado. O grafo de precedência associado a este cenário é apresentado na Figura 17.

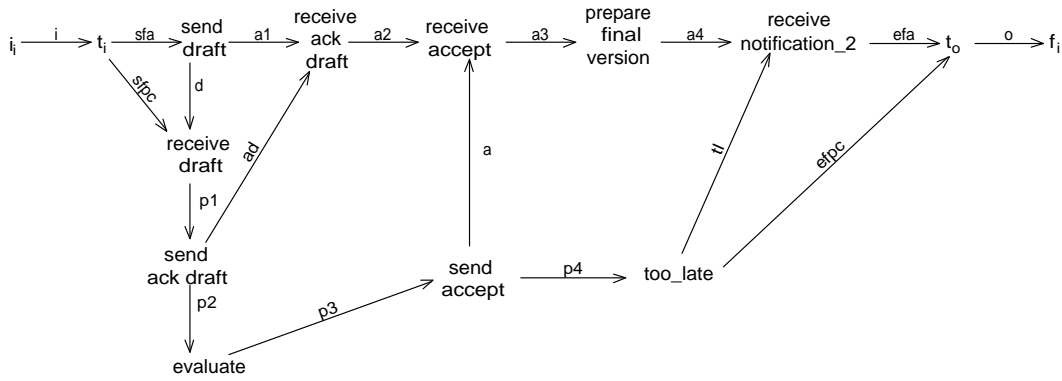


Figura 19 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_4A \times Sc_1PC$.

Neste caso, estes cenários livres de *deadlock* podem prover todas as necessidades de negócio entre o autor e o comitê de programa.

Embora a IOWF-net mostrada na Figura 6 não satisfaça o critério de correção *Soundness* e possa levar o sistema a uma situação de *deadlock*, como mostrado na seção 3.1, a abordagem apresentada na presente seção assegura que os cenários livres de *deadlock* garantem as necessidades do negócio, evitando a situação de bloqueio. É importante observar que o cenário $Sc_2A \times Sc_1PC = i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o$, que leva o processo a uma situação de *deadlock*, conforme mostrado na seção 3.1, não é considerado por esta abordagem como um cenário candidato. Desta forma, haverá uma redução no número de árvores de prova a serem construídas, em relação ao número de árvores de prova construídas para a prova do critério de correção *Soundness*. Este fato é detalhadamente abordado na avaliação dos resultados do estudo de caso, apresentado no próximo capítulo.

No que se refere ao reaproveitamento das árvores de prova da Lógica Linear, é possível, se desejável, adaptar os sequentes da Lógica Linear, a partir do método para verificação do critério de correção *Soundness*, de forma que as árvores de prova construídas para tais sequentes, no contexto da prova do critério *Soundness*, possam ser reutilizadas tanto no contexto dos métodos para verificação dos critérios *Relaxed Soundness* e *Weak Soundness* quanto no contexto do método para detecção de cenários livres de *deadlock*. Isso é possível uma vez que tais sequentes diferem-se somente por considerar ou não os lugares de comunicação assíncrona existentes na IOWF-net a ser analisada. Esta adaptação é apresentada e exemplificada na subseção 4.2.1, que trata da verificação do critério de correção *Soundness* para a composição de um serviço Web.

Estudo de Caso

A arquitetura orientada a serviços, do inglês *Service Oriented Architecture (SOA)*, evoluiu para se tornar uma tecnologia promissora para a integração de componentes de software distintos, denominados serviços Web, através da utilização de protocolos da Internet (KLAI; OCHI; TATA, 2013; XIONG; FAN; ZHOU, 2010). De acordo com Klai, Ochi e Tata (2013), os serviços Web estão disponíveis em um ambiente distribuído, a Internet, e as organizações tentam prover seus próprios serviços através de tarefas complexas que são resolvidas através da combinação ou composição de serviços Web diversos.

Vários trabalhos, como os apresentados por Klai, Ochi e Tata (2013), Xiong, Fan e Zhou (2010) e Martens (2005), já consideraram as redes de Petri como uma ferramenta apropriada para a modelagem e análise de composição de serviços Web. Segundo Xiong, Fan e Zhou (2010), a compatibilidade semântica (ou comportamental) para a composição de serviços Web é proeminente ao longo destes trabalhos, sendo considerado um dos tópicos mais relevantes. Os autores destacam que o comportamento dos serviços Web devem ser levados em consideração quando da análise da compatibilidade dos serviços Web.

Na abordagem apresentada por Klai, Ochi e Tata (2013), a análise da compatibilidade de composição de serviços Web é baseada na verificação de *Soundness*, *Weak Soundness*, *Relaxed Soundness* e *Easy Soundness*. Em Xiong, Fan e Zhou (2010), a análise da compatibilidade de composição de serviços Web considera a prova do critério de correção *Soundness*. Os autores analisam o modelo composto a fim de verificar a compatibilidade que é assegurada quando o modelo composto contém um sifão mínimo não vazio (LI; ZHOU, 2006). Os autores resolvem o problema da incompatibilidade no modelo composto remodelando-o para satisfazer a compatibilidade, impondo restrições e adicionando canais de informação adicionais (lugares e relações de fluxo) ao modelo composto. Já em Martens (2005), a análise da compatibilidade de composição de serviços Web considera a prova do critério de correção *Weak Soundness*. Em particular, os serviços nos quais uma decisão interna é efetuada e não comunicada propriamente ao ambiente são classificados como módulos não usáveis e não devem ser considerados em nenhuma composição de

serviços Web (MARTENS, 2005).

Assim, considerando os métodos baseados na Lógica Linear para verificação de processos de *workflow* interorganizacionais apresentados no Capítulo 3, um estudo de caso que considera a verificação de composição de serviços Web, utilizando os métodos propostos nesta pesquisa, é apresentado.

Inicialmente, a relação direta entre a modelagem de processos de *workflow* e a modelagem de serviços Web é apresentada na seção 4.1. Na seção 4.2, uma composição de serviços Web é verificada no que se diz respeito aos critérios de correção *Soundness*, *Relaxed Soundness* e *Weak Soundness*. Além disso, é aplicado o método para detecção de cenários livres de *deadlock* à composição de serviços Web analisada. Finalmente, uma avaliação dos resultados obtidos é realizada na seção 4.3.

4.1 Módulo de *Workflow*, Sistema Composto e WorkFlow net Interorganizacional

Um serviço Web consiste, de acordo com Martens (2005), de estruturas internas que realizam processos locais e de uma interface que permite a comunicação com outros serviços Web. Assim, um serviço Web pode ser modelado através de uma WorkFlow net suplementada de uma interface, isto é, um conjunto de lugares representando canais de mensagens dirigidas (MARTENS, 2005). Para Martens (2005), tal modelo é denominado módulo de *workflow*. Na sequência, a definição formal de um módulo de *workflow* é apresentada.

Definição 10. (Módulo) Uma rede de Petri $M = (P, T, F)$ é denominada módulo de *workflow* (ou somente módulo) se, e somente se (MARTENS, 2005):

- a) o conjunto de lugares é dividido em três conjuntos disjuntos: lugares internos P^N , lugares de *input* P^I e lugares de *output* P^O ;
- b) A relação de fluxo é dividida entre fluxo interno $F^N \subseteq (P^N \times T) \cup (T \times P^N)$ e fluxo de comunicação $F^C \subseteq (P^I \times T) \cup (T \times P^O)$;
- c) A rede $N(M) = (P^N, T, F^N)$ é uma WorkFlow net.
- d) Nenhuma transição é conectada a um lugar de *input* e a um lugar de *output*.

Para exemplificar os conceitos definidos acima, considere os serviços Web para compra virtual de bilhetes mostrados na Figura 20, adaptados de Xiong, Fan e Zhou (2010). A formalização destes serviços é mostrada na Figura 21, representada pelos módulos C1 (*Customer*) e T1 (*Ticket*), apresentados em Martens (2005) e Xiong, Fan e Zhou (2010). O módulo C1 representa um cliente (*customer*) e o módulo T1 modela um serviço de bilheteria (*ticket service*). Os lugares internos de C1 são os lugares q_0 , q_1 e q_2 . O lugar *Ticket* é o lugar de *input* de C1 e os lugares de *output* do módulo C1 são *VISA* e *eCash*.

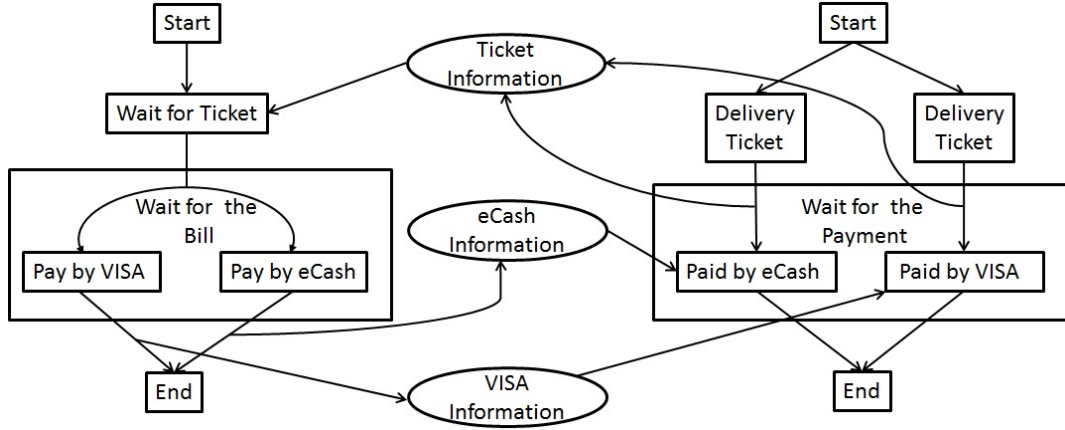


Figura 20 – Serviços Web para compra virtual de bilhetes. (Adaptado de Xiong, Fan e Zhou (2010)).

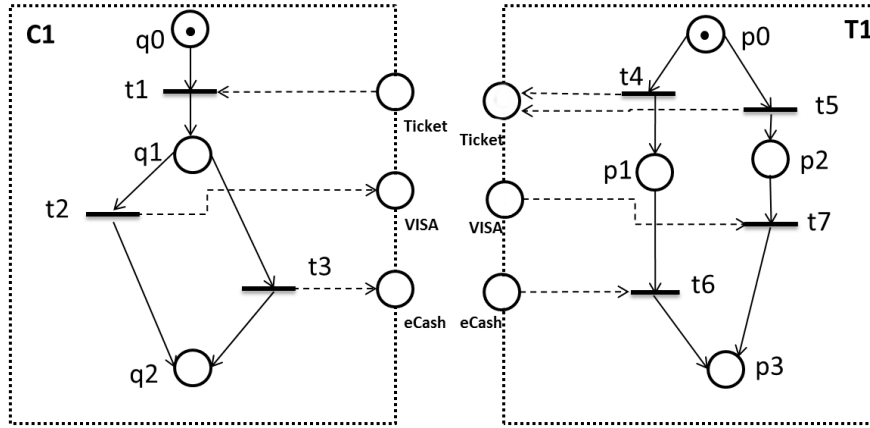


Figura 21 – Módulos de *workflow* Customer (C1) e Ticket (T1).

Considerando o módulo T1, tem-se que $p0$, $p1$, $p2$ e $p3$ são lugares internos, *VISA* e *eCash* são os lugares de *input* e *Ticket* é o único lugar de *output*.

De acordo com Martens (2005), dois módulos são classificados como sintaticamente compatíveis se os processos internos são disjuntos e cada lugar comum é um lugar de *output* de um módulo e um lugar de *input* do outro. Assim, considerando os módulos C1 and T1 apresentados na Figura 21, é fácil notar que os dois módulos são sintaticamente compatíveis. Na abordagem apresentada por Martens (2005), quando dois módulos são compostos, seus lugares em comum são fundidos e os lugares de *input* e *output* tornam-se a nova interface. De acordo com Martens (2005), para alcançar um módulo de *workflow* sintaticamente correto, é necessário adicionar novos componentes de inicialização e finalização ao modelo. Neste contexto, Martens (2005) definiu o conceito de sistema composto. A definição formal de sistema composto é apresentada na sequência.

Definição 11. (Sistema Composto)

Sejam $A = (P_a, T_a, F_a)$ e $B = (P_b, T_b, F_b)$ dois módulos sintaticamente compatíveis.

Sejam $i, o \notin (P_a \cup P_b)$ dois novos lugares e $t_i, t_o \notin (T_a \cup T_b)$ duas novas transições.

O Sistema composto $A \oplus B$ é dado por (P_s, T_s, F_s) , tal que: $P_s = P_a \cup P_b \cup \{i, o\}$, $T_s = T_a \cup T_b \cup \{t_i, t_o\}$ e $F_s = F_a \cup F_b \cup \{(i, t_i), (t_i, \alpha_a), (t_i, \alpha_b), (\omega_a, t_o), (\omega_b, t_o), (t_o, o)\}$.

A Figura 22 mostra o sistema composto $C1 \oplus T1$. Note que o serviço T1 resolve um conflito interno e envia o *Ticket* para o módulo C1. Depois disso, o módulo T1 está no estado $p1$, esperando por *eCash*, ou no estado $p2$, esperando por *VISA*. O módulo C1 recebe o *Ticket* e tem uma escolha entre dois tipos de pagamento: VISA ou eCash. Pode-se observar que o módulo C1 não conhece o estado interno do módulo T1, isto é, $p1$ ou $p2$. De acordo com Martens (2005), quando uma decisão interna é tomada e não propriamente comunicada a um serviço Web externo, um problema bem conhecido na literatura ocorre: o problema da escolha não local (BEN-ABDALLAH; LEUE, 1997) e, nestes casos, segundo o autor, tais módulos não deveriam ser considerados em composições de serviços Web.

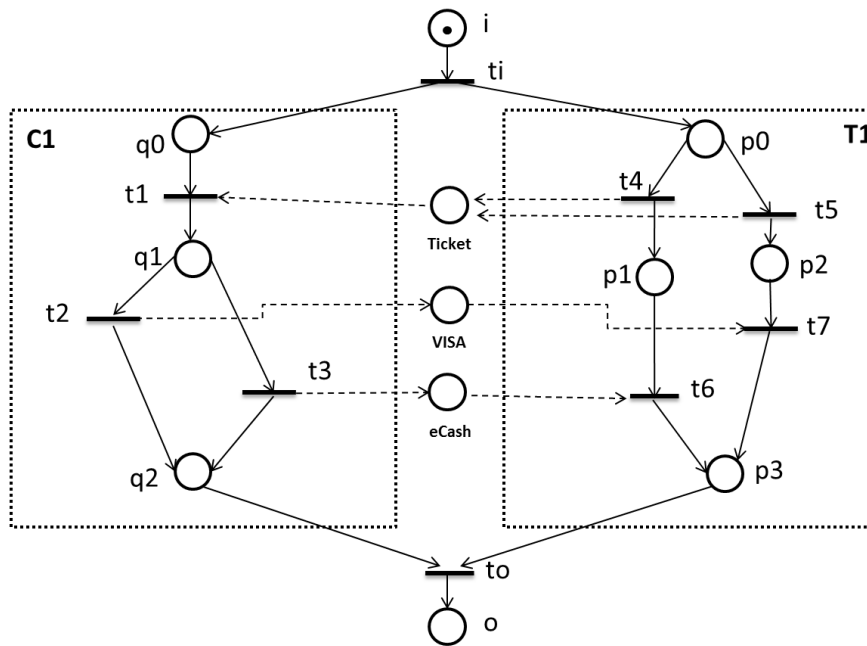


Figura 22 – O Sistema Composto $C1 \oplus T1$.

É importante dar atenção às definições de módulo de *workflow* e de sistema composto apresentadas por Martens (2005) e a definição de IOWF-net, apresentada por Aalst (1998b). Comparando a definição de um módulo de *workflow* e a definição de uma IOWF-net, observa-se que um módulo corresponde a uma WorkFlow net Local (LWF-net) adicionada de sua estrutura de comunicação (lugares e fluxos de comunicação). Deve-se notar ainda que a definição de sistema composto é equivalente à definição de uma $U(\text{IOWF-net})$ que considera apenas duas LWF-nets. Desta forma, os métodos apresentados na presente pesquisa, para verificação de processos de *workflow* interorganizacionais modelados por IOWF-net podem ser diretamente aplicados à verificação de composições

de serviços Web. Além disso, a composição de mais de dois módulos de *workflow* é realizada por aplicação recorrente da composição em pares (MARTENS, 2005).

Assim, a seção 4.2 apresenta a aplicação dos métodos propostos no Capítulo 3 à verificação de composições de serviços Web.

4.2 Verificação de Composição de Serviços Web utilizando Lógica Linear

A presente seção apresenta a aplicação dos métodos para verificação de *Soundness*, *Relaxed Soundness* e *Weak Soundness* e do método para detecção de cenários livres de *deadlock*, propostos no Capítulo 3, ao contexto da verificação de composições de serviços Web.

Assim, inicialmente, é necessário definir o modelo composto sobre o qual os métodos de verificação serão aplicados. O modelo escolhido para a verificação é o sistema composto apresentado na Figura 22. Um dos motivos da escolha deste modelo ocorre em virtude do número de trabalhos que consideram este modelo para análise, principalmente devido às suas características inerentes. A escolha deste modelo pode ser vista em Martens (2003), Martens (2005) e, mais recentemente, em Xiong, Fan e Zhou (2010). Outro aspecto que motivou a utilização deste sistema composto (a verificação da composição dos serviços $C1$ e $T1$) é que Martens (2005) destaca que o serviço Web $T1$ não deve ser utilizado em nenhuma composição de serviços Web em função do problema da escolha não local. Assim, a utilização deste serviço Web em alguma composição deve ser descartada. Já Xiong, Fan e Zhou (2010) propõem reprojeter o sistema composto, adicionando características que tornem o sistema composto seguro, de forma que a situação de *deadlock* existente no modelo seja excluída.

As subseções 4.2.1, 4.2.2 e 4.2.3 apresentam, respectivamente, a verificação dos critérios de correção *Soundness*, *Relaxed Soundness* e *Weak Soundness* para o sistema composto $C1 \oplus T1$. Já a subseção 4.2.4 apresenta a aplicação do método para detecção de cenários livres de *deadlock* ao sistema composto $C1 \oplus T1$.

4.2.1 Verificação do critério de correção *Soundness* para o sistema composto $C1 \oplus T1$

Para verificar o critério de correção *Soundness* para o sistema composto $C1 \oplus T1$, apresentado na Figura 22, é necessário, inicialmente, representar cada cenário de cada um dos módulos de *workflow*, isto é, $C1$ e $T1$, através de um sequente da Lógica Linear.

Para o módulo de *workflow* $C1$, mostrado na Figura 21, há dois diferentes cenários a serem considerados: o primeiro cenário, Sc_1C1 , onde a tarefa $t2$ é executada e o segundo cenário, Sc_2C1 , onde a execução da tarefa $t3$ é considerada. Considerando o módulo $T1$,

existem dois cenários distintos: o primeiro cenário, Sc_1T1 , onde as tarefas $t4$ e $t6$ serão executadas e o segundo cenário, Sc_2T1 , onde as tarefas $t5$ e $t7$ serão realizadas.

Conforme mencionado no método para Detecção de Cenários Livres de *Deadlock* em Processos de *Workflow* Interorganizacionais, apresentado na seção 3.4, os sequentes da Lógica Linear, que representam os cenários, podem ser adaptados, a partir do método para verificação do critério de correção *Soundness*, de forma que as árvores de prova construídas para prová-los possam ser reutilizadas tanto nos métodos para verificação dos critérios de correção *Relaxed Soundness* e *Weak Soundness*, quanto no método para detecção de cenários livres de *deadlock*. Desta forma, no contexto deste estudo de caso, é ilustrado como os sequentes da Lógica Linear devem ser adaptados. É importante observar que no método para Verificação do Critério de Correção *Soundness* para Processos de *Workflow* Interorganizacionais, apresentado na seção 3.1, no que se diz respeito à verificação dos processos locais, os lugares de comunicação existentes são desconsiderados. Na adaptação dos sequentes, para viabilizar o reuso em todas as abordagens de verificação, tais lugares de comunicação devem ser então considerados nos sequentes e, conseqüentemente, nas árvores de prova construídas para prová-los. Assim, cada um dos cenários das LWF-nets (ou módulo de *workflow*, no caso dos serviços Web) é então representado por um sequente da Lógica Linear que considera as marcações inicial e final da LWF-net (ou módulo de *workflow*) analisada, uma lista não ordenada das transições envolvidas no cenário, e os lugares de comunicação de entrada (ICP, no caso das LWF-nets e um subconjunto de P^I no caso dos módulos de *workflow*) das transições envolvidas no cenário. Considerando a adaptação apresentada na análise das árvores de prova, é então necessário desconsiderar os átomos que representam os lugares de comunicação. Por exemplo, de acordo com o método para Verificação do Critério de Correção *Soundness* para Processos de *Workflow* Interorganizacionais, é necessário verificar se apenas um átomo o , que corresponde a uma ficha no lugar de término da LWF-net, é produzido na árvore de prova. Neste ponto, destaca-se que este fato é representado na árvore de prova pelo sequente identidade $o \vdash o$. Com a adaptação dos sequentes, possivelmente, haverá no último sequente átomos disponíveis que correspondem aos lugares de comunicação agora considerados. A presença de tais átomos deve ser então desconsiderada nas condições a serem verificadas na análise das árvores de prova.

Para facilitar a compreensão da análise dos sequentes adaptados, os passos de análise das árvores de prova que consideram tais sequentes adaptados das LWF-nets, no caso das IOWF-nets, e dos módulos de *workflow*, no caso dos serviços Web, são reescritos na sequência. Considere que ψ é um conjunto de átomos que representam lugares de comunicação da rede de Petri analisada. No caso das LWF-nets, $\psi \subseteq P_{AC}$ e, no caso dos serviços Web, $\psi \subseteq P^O$. Assim, deve-se verificar:

1. para cada árvore de prova de cada LWF-net (ou, no caso dos serviços Web, módulo de *workflow*):

- a) se apenas um átomo o , que corresponde a uma ficha no lugar de término da LWF-net (ou do módulo de *workflow*) analisada foi produzido na árvore de prova (este fato é representado na árvore de prova pelo sequente $\psi, o \vdash o$), então o primeiro requisito para a prova de *Soundness* é verificado, isto é, apenas uma ficha aparece no lugar o ;
 - b) se não há nenhum átomo disponível para consumo no último sequente da árvore de prova, exceto os átomos de ψ , então significa que todos os lugares da LWF-net (ou do módulo de *workflow*) analisada estão vazios, isto é, o segundo requisito para a prova de *Soundness* é verificado;
 - c) não há nenhuma fórmula de transição disponível no último sequente da árvore de prova a ser considerada;
2. considerando todos os cenários Sc_1, Sc_2, \dots, Sc_n para a LWF-net (ou módulo de *workflow*) analisada, cada transição $t \in T$ precisa aparecer, pelo menos, em uma árvore de prova. Isto prova que todas as transições podem ser disparadas (ou seja, não há transições mortas). Este fato e a condição c) implicam na verificação do terceiro requisito para a prova de *Soundness*.

Se as condições 1 e 2 acima são satisfeitas, a LWF-net (ou módulo de *workflow*) analisada é *Sound*.

É importante observar que somente o método para análise das LWF-nets (ou módulos de *workflow*) deve ser adaptado. O método para análise da $U(IOWF\text{-}net)$ (ou do sistema composto) continua o mesmo apresentado no método para Verificação do Critério de Correção *Soundness* para Processos de *Workflow* Interorganizacionais, na seção 3.1.

Assim, considerando o módulo de *workflow* $C1$, apresentado na Figura 22, dois cenários diferentes (e, conseqüentemente, sequentes lineares) são definidos:

$$Sc_1C1 = q0, ticket, t_1, t_2 \vdash q2,$$

$$Sc_2C1 = q0, ticket, t_1, t_3 \vdash q2.$$

Para o módulo de *workflow* $T1$, mostrado na Figura 22, dois cenários são definidos:

$$Sc_1T1 = p0, eCash, t_4, t_6 \vdash p3,$$

$$Sc_2T2 = p0, VISA, t_5, t_7 \vdash p3.$$

É importante observar que os átomos *ticket*, *VISA* e *eCash* que aparecem nos sequentes definidos para o exemplo representam os lugares de *input* (um subconjunto de P^I) envolvidos nos cenários descritos. Por exemplo, o cenário Sc_1C1 é dado por $q0, ticket, t_1, t_2 \vdash q2$, onde $q0$ e $q2$ são, respectivamente, as marcações inicial e final do módulo $C1$, *ticket* representa o lugar de *input* (ou seja, um subconjunto de P^I) deste cenário, e t_1, t_2 representa uma lista não ordenada das transições envolvidas no cenário Sc_1C1 .

Considerando o sistema composto apresentado na Figura 22, cada transição é representada por:

$$\begin{aligned}
t_i &= i \multimap q0 \otimes p0, \\
t_1 &= q0 \otimes \text{ticket} \multimap q1, \\
t_2 &= q1 \multimap \text{VISA} \otimes q2, \\
t_3 &= q1 \multimap \text{eCash} \otimes q2, \\
t_4 &= p0 \multimap \text{ticket} \otimes p1, \\
t_5 &= p0 \multimap \text{ticket} \otimes p2, \\
t_6 &= p1 \otimes \text{eCash} \multimap p3, \\
t_7 &= p2 \otimes \text{VISA} \multimap p3, \\
t_o &= q2 \otimes p3 \multimap o.
\end{aligned}$$

Após a definição dos sequentes da Lógica Linear que representam cada módulo de *workflow*, é necessário prová-los. Para tanto, árvores de prova da Lógica Linear serão construídas. As árvores de prova para os cenários Sc_1C1 , Sc_2C1 , Sc_1T1 e Sc_2T1 são apresentadas no Apêndice A.3. Considerando tais árvores de prova, pode-se aplicar o método de verificação adaptado apresentado nesta seção para verificar se estes módulos satisfazem o critério de correção *Soundness*. Inicialmente é considerada a verificação do módulo de *workflow* $C1$, ou seja, as árvores de prova para os cenários Sc_1C1 e Sc_2C1 devem ser analisadas. Na sequência, mostra-se que as condições 1 e 2 são verificadas para os cenários Sc_1C1 e Sc_2C1 , isto é:

1. em cada árvore de prova do módulo de *workflow* $C1$:
 - a) apenas um átomo $q2$, que corresponde ao átomo o na definição de módulo de *workflow*, que indica a presença de uma ficha no lugar de término do módulo $C1$, foi produzido na árvore de prova (este fato é representado nas árvores de prova pelos sequentes $\text{VISA}, q2 \vdash q2$ e $\text{eCash}, q2 \vdash q2$, onde $\{\text{VISA}, \text{eCash}\} \subset P^O$);
 - b) não há nenhum átomo disponível para consumo no último sequeute de cada árvore de prova construída, exceto os átomos $\{\text{VISA}, \text{eCash}\} \subset P^O$; isso significa que todos os lugares do módulo de *workflow* $C1$ estão vazios nos cenários analisados;
 - c) não há nenhuma fórmula de transição disponível no último sequeute das árvores de prova construídas;
2. considerando os cenários Sc_1C1 e Sc_2C1 , para o módulo de *workflow* $C1$, cada transição $t \in T_{C1}$ aparece, pelo menos, em uma árvore de prova.

Como as condições 1 e 2 são satisfeitas, tem-se que o módulo $C1$ satisfaz o critério de correção *Soundness*.

Uma vez que o módulo $C1$ satisfaz o critério de correção *Soundness*, é necessário verificar se o módulo $T1$ também satisfaz tal critério de correção. Neste caso, as árvores

de prova para os cenários Sc_1T1 e Sc_2T1 devem ser analisadas. A análise de tais árvores de prova mostram que as condições 1 e 2 são verificadas, isto é:

1. em cada árvore de prova do módulo de *workflow* $T1$:
 - a) apenas um átomo $p3$, que corresponde ao átomo o na definição de módulo de *workflow*, que indica a presença de uma ficha no lugar de término do módulo $T1$, foi produzido na árvore de prova (este fato é representado nas árvores de prova pelo sequente $ticket, p3 \vdash p3$, onde $\{ticket\} \subset P^O$);
 - b) não há nenhum átomo disponível para consumo no último sequente de cada árvore de prova construída, exceto o átomo $ticket$, onde $\{ticket\} \subset P^O$; isso significa que todos os lugares do módulo de *workflow* $C1$ estão vazios nos cenários analisados;
 - c) não há nenhuma fórmula de transição disponível no último sequente das árvores de prova construídas;
2. considerando todos os cenários Sc_1T1 e Sc_2T1 , para o módulo de *workflow* $T1$, cada transição $t \in T_{T1}$ aparece, pelo menos, em uma árvore de prova.

Como as condições 1 e 2 são satisfeitas, tem-se que o módulo $T1$ satisfaz o critério de correção *Soundness*.

Uma vez que os módulos de *workflow* $C1$ e $T1$ satisfazem o critério de correção *Soundness*, é necessário verificar este critério de correção para o sistema composto $C1 \oplus T1$, mostrado na Figura 22. Para tanto, é necessário definir e provar quatro sequentes da Lógica Linear, sendo que cada sequente representa um cenário do sistema composto $C1 \oplus T1$. Assim, os cenários do sistema composto, mostrado na Figura 22, são dados pelos seguintes sequentes da Lógica Linear:

$$Sc_1C1 \oplus T1 = i, t_i, t_1, t_3, t_4, t_6, t_o \vdash o,$$

$$Sc_2C1 \oplus T1 = i, t_i, t_1, t_2, t_5, t_7, t_o \vdash o,$$

$$Sc_3C1 \oplus T1 = i, t_i, t_1, t_2, t_4, t_6, t_o \vdash o,$$

$$Sc_4C1 \oplus T1 = i, t_i, t_1, t_3, t_5, t_7, t_o \vdash o.$$

O primeiro cenário, $Sc_1C1 \oplus T1$, é o cenário onde a tarefa $t4$ é executada (disparo da transição $t4$ do módulo $T1$) e o *ticket* é enviado para o módulo $C1$. O módulo $C1$ escolhe o modo de pagamento *eCash* (disparo da transição $t3$) e finaliza. O módulo $T1$ recebe a forma de pagamento (*eCash*) e finaliza. Finalmente, o sistema composto $C1 \oplus T1$ finaliza sua execução.

O segundo cenário, $Sc_2C1 \oplus T1$, é o cenário onde a tarefa $t5$ é executada (disparo da transição $t5$ do módulo $T1$) e o *ticket* é enviado para o módulo $C1$. O módulo $C1$ escolhe o modo de pagamento *VISA* (disparo da transição $t2$) e finaliza. O módulo $T1$ recebe a forma de pagamento (*VISA*) e finaliza. Finalmente, o sistema composto $C1 \oplus T1$ finaliza sua execução.

O terceiro cenário, $Sc_3C1 \oplus T1$, é o cenário onde a tarefa t_4 é executada (disparo da transição t_4 do módulo $T1$) e o *ticket* é enviado para o módulo $C1$. O módulo $C1$ escolhe a forma de pagamento *VISA* (disparo da transição t_2) e finaliza. O sistema composto $C1 \oplus T1$ encontra-se em situação de bloqueio (*deadlock*) uma vez que o módulo $T1$ está parado em $p1$, esperando pelo pagamento via *eCash*.

O quarto cenário, $Sc_4C1 \oplus T1$, é o cenário onde a tarefa t_5 é executada (disparo da transição t_5 do módulo $T1$) e o *ticket* é enviado para o módulo $C1$. O módulo $C1$ escolhe a forma de pagamento *eCash* (disparo da transição t_3) e finaliza. O sistema composto $C1 \oplus T1$ encontra-se em situação de bloqueio (*deadlock*) uma vez que o módulo $T1$ está parado em $p2$, esperando pelo pagamento via cartão *VISA*.

Após a definição dos sequentes da Lógica Linear que representam cada cenário do sistema composto $C1 \oplus T1$, estes sequentes precisam ser provados. Para prová-los, árvores de prova da Lógica Linear devem ser construídas. O Apêndice A.3 apresenta as árvores de prova para cada um destes cenários.

Considerando as árvores de prova para os cenários $Sc_1C1 \oplus T1$, $Sc_2C1 \oplus T1$, $Sc_3C1 \oplus T1$ e $Sc_4C1 \oplus T1$, apresentadas no apêndice A.3, verifica-se que nas árvores de prova para os cenários $Sc_3C1 \oplus T1$ e $Sc_4C1 \oplus T1$ do sistema composto $C1 \oplus T1$:

- a) nenhum átomo o foi produzido na árvore de prova para estes cenários. Ou seja, o sequente identidade $o \vdash o$ não aparece nas árvores de prova. Isso significa que nenhuma ficha foi produzida no lugar de término destes cenários, ou seja, para estes cenários, o sistema composto não finaliza;
- b) há três átomos disponíveis para consumo no último sequente da árvore de prova construída para o cenário $Sc_3C1 \oplus T1$: $p1$, *VISA* e o átomo $q2$. O cenário $Sc_4C1 \oplus T1$ também possui três átomos disponíveis para consumo: $p2$, *eCash* e $q2$;
- c) considerando o cenário $Sc_3C1 \oplus T1$, há duas fórmulas de transição disponíveis (que não foram disparadas) no último sequente das árvore de prova: t_6 e t_o . Já para o cenário $Sc_4C1 \oplus T1$, tem-se t_7 e t_o .

Como a condição 1 não é satisfeita, tem-se que o sistema composto $C1 \oplus T1$ não satisfaz o critério de correção *Soundness*. Assim, conclui-se que, apesar dos módulos de *workflow* $C1$ e $T1$ satisfazerem o critério de correção *Soundness*, o sistema composto mostrado na Figura 22 não é *Sound*.

É importante observar que os cenários $Sc_3C1 \oplus T1$ e $Sc_4C1 \oplus T1$ são os cenários que levam o sistema composto $C1 \oplus T1$, mostrado na Figura 22, a situações de *deadlock*.

4.2.2 Verificação do critério de correção *Relaxed Soundness* para o sistema composto $C1 \oplus T1$

Para verificar o critério de correção *Relaxed Soundness* para o sistema composto $C1 \oplus T1$, apresentado na Figura 22, é necessário, inicialmente, representar cada cenário deste

sistema composto através de um sequente da Lógica Linear.

Para tanto, conforme mostra a seção 4.2.1, é necessário definir e provar quatro sequentes da Lógica Linear, sendo que cada sequente representa um cenário do sistema composto $C1 \oplus T1$. Assim, os cenários do sistema composto, mostrado na Figura 22, são dados pelos seguintes sequentes da Lógica Linear:

$$Sc_1C1 \oplus T1 = i, t_i, t_1, t_3, t_4, t_6, t_o \vdash o,$$

$$Sc_2C1 \oplus T1 = i, t_i, t_1, t_2, t_5, t_7, t_o \vdash o,$$

$$Sc_3C1 \oplus T1 = i, t_i, t_1, t_2, t_4, t_6, t_o \vdash o,$$

$$Sc_4C1 \oplus T1 = i, t_i, t_1, t_3, t_5, t_7, t_o \vdash o.$$

Após a definição dos sequentes da Lógica Linear que representam cada cenário do sistema composto $C1 \oplus T1$, estes sequentes precisam ser provados. Para prová-los, árvores de prova da Lógica Linear devem ser construídas. O Apêndice A.3 apresenta as árvores de prova para cada um destes cenários. Observa-se que as árvores de prova para estes cenários são as mesmas construídas no contexto da verificação do critério de correção *Soundness*, apresentada na subseção 4.2.1. Portanto, tais árvores de prova são simplesmente reutilizadas para a verificação do critério de correção *Relaxed Soundness*.

O próximo passo, de acordo com o método para verificação do critério de correção *Relaxed Soundness* proposto na seção 3.2, é analisar as árvores de prova produzidas para cada cenário Sc_i do sistema composto mostrado na Figura 22. A análise das árvores de prova deve considerar exatamente a verificação das condições 1 e 2, especificadas na seção 3.2, ou seja, nenhuma adaptação ao método apresentado é necessária.

Assim, considerando as árvores de prova para os cenários $Sc_1C1 \oplus T1$, $Sc_2C1 \oplus T1$, $Sc_3C1 \oplus T1$ e $Sc_4C1 \oplus T1$, mostradas no Apêndice A.3, é necessário verificar para quais cenários Sc_i a condição 1 é satisfeita e se a condição 2 é satisfeita considerando os cenários que satisfazem a condição 1.

É fácil notar que o último sequente nas árvores de prova dos cenários $Sc_1C1 \oplus T1$ e $Sc_2C1 \oplus T1$ é $o \vdash o$. Desta maneira, as condições a) e b) do passo 1 são satisfeitas, isto é, apenas um átomo o foi produzido nestas árvores de prova e, como o último sequente é um sequente identidade, não há nenhum átomo disponível para consumo nestes cenários, ou seja, a execução destes cenários finaliza sem fichas remanescentes.

Considerando a análise do cenário $Sc_3C1 \oplus T1$, nota-se que o último sequente deste cenário é $p1, VISA, q2, t_6, t_o \vdash o$. É fácil notar que nenhum átomo o foi produzido nesse cenário. Assim, a condição a) do passo 1 não é verificada para este cenário. Este sequente também possui três átomos disponíveis para consumo $p1, VISA$ e $q2$. Consequentemente, este cenário não satisfaz a condição b) do passo 1 de verificação.

Considerando a análise do cenário $Sc_4C1 \oplus T1$, nota-se que o último sequente deste cenário é $p2, eCash, q2, t_7, t_o \vdash o$. Nota-se ainda que nenhum átomo o foi produzido nesse cenário. Assim, a condição a) do passo 1 não é verificada para este cenário. Este sequente também possui três átomos disponíveis para consumo $p2, eCash$ e $q2$. Consequentemente,

este cenário também não satisfaz a condição b) do passo 1 de verificação.

Desta maneira, na segunda parte do processo de verificação, (passo 2), apenas os cenários $Sc_1C1 \oplus T1$ e $Sc_2C1 \oplus T1$ são considerados. Assim, é preciso verificar se cada transição $t \in T$ do sistema composto da Figura 22 aparece em pelo menos um destes cenários. A Tabela 5 mostra, para cada transição $t \in T$ do sistema composto, os cenários nos quais a transição t aparece.

Tabela 7 – Relação entre transições e cenários do sistema composto $C1 \oplus T1$.

Transição	Cenários
t_i	$Sc_1C1 \oplus T1, Sc_2C1 \oplus T1$
t_1	$Sc_1C1 \oplus T1, Sc_2C1 \oplus T1$
t_2	$Sc_2C1 \oplus T1$
t_3	$Sc_1C1 \oplus T1$
t_4	$Sc_1C1 \oplus T1$
t_5	$Sc_2C1 \oplus T1$
t_6	$Sc_1C1 \oplus T1$
t_7	$Sc_2C1 \oplus T1$
t_o	$Sc_1C1 \oplus T1, Sc_2C1 \oplus T1$

Considerando a Tabela 7, verifica-se que cada transição $t \in T$ aparece em pelo menos um cenário. Logo, a condição 2 também é satisfeita e o sistema composto mostrado na Figura 22 satisfaz o critério de correção *Relaxed Soundness*. Os cenários $Sc_1C1 \oplus T1$ e $Sc_2C1 \oplus T1$ são os cenários que finalizam corretamente, sem deixar fichas remanescentes na rede.

4.2.3 Verificação do critério de correção *Weak Soundness* para o sistema composto $C1 \oplus T1$

Para verificar o critério de correção *Weak Soundness* para o sistema composto $C1 \oplus T1$, apresentado na Figura 22, é necessário, inicialmente, representar cada cenário do sistema composto $C1 \oplus T1$ através de um sequente da Lógica Linear.

Para tanto, conforme mostra a seção 4.2.1, é necessário definir e provar quatro sequentes da Lógica Linear, sendo que cada sequente representa um cenário do sistema composto $C1 \oplus T1$. Assim, os cenários do sistema composto, mostrado na Figura 22, são dados pelos seguintes sequentes da Lógica Linear:

$$Sc_1C1 \oplus T1 = i, t_i, t_1, t_3, t_4, t_6, t_o \vdash o,$$

$$Sc_2C1 \oplus T1 = i, t_i, t_1, t_2, t_5, t_7, t_o \vdash o,$$

$$Sc_3C1 \oplus T1 = i, t_i, t_1, t_2, t_4, t_6, t_o \vdash o,$$

$$Sc_4C1 \oplus T1 = i, t_i, t_1, t_3, t_5, t_7, t_o \vdash o.$$

Após a definição dos sequentes da Lógica Linear que representam cada cenário do sistema composto $C1 \oplus T1$, estes sequentes precisam ser provados. Para prová-los, árvores de prova da Lógica Linear devem ser construídas. O Apêndice A.3 apresenta as árvores de prova para cada um destes cenários. É possível observar que as árvores de prova para estes

cenários são as mesmas construídas no contexto das verificações dos critérios de correção *Soundness* e *Relaxed Soundness*, apresentadas nas seções 4.2.1 e 4.2.2. Portanto, tais árvores de prova são simplesmente reutilizadas para a verificação do critério de correção *Weak Soundness*.

O próximo passo, de acordo com o método para verificação do critério de correção *Weak Soundness* proposto na seção 3.3, é analisar as árvores de prova produzidas para cada cenário Sc_i do modelo composto mostrado na Figura 22. Assim, deve-se analisar as árvores de prova para os cenários $Sc_1C1 \oplus T1$, $Sc_2C1 \oplus T1$, $Sc_3C1 \oplus T1$ e $Sc_4C1 \oplus T1$, mostradas no Apêndice A.3, de forma a verificar se a condição 1 é satisfeita. É relevante destacar que a análise das árvores de prova deve considerar exatamente a verificação da condição 1 especificada na seção 3.3, ou seja, nenhuma adaptação ao método de verificação apresentado é necessária.

Prosseguindo com a análise das árvores de prova, nota-se que o último sequente nas árvores de prova dos cenários $Sc_1C1 \oplus T1$ e $Sc_2C1 \oplus T1$ é $o \vdash o$. Desta maneira, as condições a) e b) do passo 1 são satisfeitas, isto é, apenas um átomo o foi produzido nestas árvores de prova e, como o último sequente é um sequente identidade, não há nenhum átomo disponível para consumo nestes cenários.

No entanto, o mesmo não ocorre para os cenários $Sc_3C1 \oplus T1$ e $Sc_4C1 \oplus T1$, onde a condição 1 não é satisfeita, uma vez que nas árvores de prova para os cenários $Sc_3C1 \oplus T1$ e $Sc_4C1 \oplus T1$ do sistema composto:

- a) nenhum átomo o foi produzido nas árvores de prova, ou seja, o sequente identidade $o \vdash o$ não aparece nas árvores de prova para os cenários $Sc_3C1 \oplus T1$ e $Sc_4C1 \oplus T1$. Isso significa que nenhuma ficha foi produzida no lugar de término destes cenários, ou seja, para estes cenários, o sistema composto não finaliza;
- b) há três átomos disponíveis para consumo no último sequente das árvores de prova construídas para os cenários $Sc_3C1 \oplus T1$ e $Sc_4C1 \oplus T1$; para o cenário $Sc_3C1 \oplus T1$ os átomos $p1$, $VISA$ e $q2$ e para o cenário $Sc_4C1 \oplus T1$ os átomos $p2$, $eCash$ e $q2$.

Portanto, como a condição 1 do método de verificação do critério *Weak Soundness*, apresentado na seção 3.3, não é satisfeita para os cenários $Sc_3C1 \oplus T1$ e $Sc_4C1 \oplus T1$, tem-se que o sistema composto $C1 \oplus T1$, apresentado na Figura 22, não satisfaz o critério de correção *Weak Soundness*. De fato, pode-se observar a presença de *deadlock* no sistema composto $C1 \oplus T1$ e, assim, o sistema analisado não poderia satisfazer o critério de *Weak Soundness*.

4.2.4 Detecção de cenários livres de *deadlock* no sistema composto $C1 \oplus T1$

A aplicação do método para detecção de cenários livres de *deadlock* em um sistema composto considera a definição e análise de todos os cenários dos módulos que compõem

o sistema composto.

Assim, considerando a noção de um cenário definida neste trabalho, tem-se que para o módulo de *workflow* C1, apresentado na Figura 21, existem dois cenários distintos: o primeiro cenário Sc_1C1 , onde a tarefa $t2$ será executada e o segundo cenário, Sc_2C1 , onde a execução da tarefa $t3$ é considerada. Já para o módulo T1, há dois cenários distintos: o primeiro cenário, Sc_1T1 , onde as tarefas $t4$ e $t6$ serão executadas e o segundo cenário, Sc_2T1 , onde as tarefas $t5$ e $t7$ serão realizadas, conforme mostra a subseção 4.2.1.

Nesta abordagem, cada um destes cenários é então representado por um sequente da Lógica Linear que considera as marcações inicial e final de um módulo, seus lugares de *input* e uma lista não ordenada de transições envolvidas no cenário.

Assim, considerando o módulo de *workflow* C1 apresentado na Figura 21, dois cenários distintos (e, conseqüentemente, sequentes da Lógica Linear) são definidos:

$$Sc_1C1 = q0, ticket, t_1, t_2 \vdash q2,$$

$$Sc_2C1 = q0, ticket, t_1, t_3 \vdash q2.$$

Já para o módulo de *workflow* T1, mostrado na Figura 21, dois cenários são definidos:

$$Sc_1T1 = p0, eCash, t_4, t_6 \vdash p3,$$

$$Sc_2T1 = p0, VISA, t_5, t_7 \vdash p3.$$

Considerando a adaptação proposta na subseção 4.2.1, estes cenários são um subconjunto dos cenários definidos para a verificação do critério de correção *Soundness* para o sistema composto $C1 \oplus T1$. Conseqüentemente, as árvores de prova construídas para provar tais cenários, apresentadas no Apêndice A.3, podem ser reutilizadas para a detecção dos cenários livres de *deadlock* no sistema composto $C1 \oplus T1$.

É importante notar que, no contexto da detecção de cenários livres de *deadlock*, cada árvore de prova da Lógica Linear possui dois sequentes mais relevantes: o primeiro sequente e o último sequente. Como as árvores de prova da Lógica Linear são construídas e lidas de forma *bottom up*, o primeiro sequente da árvore de prova para o cenário Sc_1C1 , mostrada no Apêndice A.3, é $q0, ticket, t_1, t_2 \vdash q2$ e o último sequente é $VISA, q2 \vdash q2$. O átomo *ticket* no primeiro sequente representa o lugar de *input* para o cenário Sc_1C1 . Isso significa que, para funcionar corretamente, este cenário consome o átomo *ticket*. Já o átomo *VISA*, no último sequente, corresponde a um átomo produzido pelo cenário Sc_1C1 , que não foi consumido por nenhuma fórmula de transição deste cenário, isto é, o átomo remanescente na árvore de prova, ou seja, o átomo que representa o lugar de *output* para o referido cenário.

Assim, considerando o conjunto de árvores de prova para os cenários Sc_1C1 , Sc_2C1 , Sc_1T1 e Sc_2T1 , apresentadas no Apêndice A.3, nota-se que o átomo de *input* do cenário Sc_1C1 , *ticket*, corresponde ao átomo de *output* do cenário Sc_2T1 . Além disso, o átomo de *output* do cenário Sc_1C1 , *VISA*, corresponde ao átomo de lugar de *input* para o cenário Sc_2T1 . Ou seja, $P^I(Sc_1C1) \cup P^I(Sc_2T1) = \{ticket, VISA\} = P^O(Sc_1C1) \cup P^O(Sc_2T1) = \{VISA, ticket\}$. Já para os cenários Sc_2C1 e Sc_1T1 , tem-se que $P^I(Sc_2C1) \cup P^I(Sc_1T1) =$

$\{ticket, eCash\} = P^O(Sc_2C1) \cup P^O(Sc_1T1) = \{eCash, ticket\}$. Desta forma, considerando o sistema composto da Figura 22, existem dois cenários candidatos a serem analisados: $Sc_1C1 \times Sc_2T1$ e $Sc_2C1 \times Sc_1T1$.

Deve-se observar que, de acordo com a Definição 11, um sistema composto considera exatamente dois módulos de *workflow*. Já a Definição 8, mostra que uma IOWF-net possui n LWF-nets. Logo, os cenários candidatos a serem livres de *deadlock*, no caso da composição de serviços Web, considera a composição de exatamente dois cenários. Assim, considerando o método apresentado na seção 3.4 e o fato de que um sistema composto possui exatamente dois módulos de *workflow*, para verificar se um cenário candidato $Sc_i1 \times Sc_j2$ é, de fato, um cenário livre de *deadlock*, é necessário provar o seguinte da Lógica Linear que considera as listas de transições dos cenários Sc_i1 e Sc_j2 , as transições t_i e t_o , juntamente com os lugares globais de início e término i e o , respectivamente, definidos na Definição 11. Assim, o seguinte da Lógica Linear a ser provado é dado por: $i, t_i, \alpha, \beta, t_o \vdash o$, onde α é a lista de transições para o cenário Sc_i1 e β é a lista de transições para o cenário Sc_j2 . Assim, para o cenário candidato $Sc_1C1 \times Sc_2T1$, o seguinte seguinte da Lógica Linear deve ser provado: $i, t_i, t_1, t_2, t_5, t_7, t_o \vdash o$. Para o cenário candidato $Sc_2C1 \times Sc_1T1$, tem-se o seguinte seguinte da Lógica Linear: $i, t_i, t_1, t_3, t_4, t_6, t_o \vdash o$.

Considerando as árvores de prova apresentadas no Apêndice A.3, o cenário candidato $Sc_1C1 \times Sc_2T1$, cujo seguinte a ser provado é $i, t_i, t_1, t_2, t_5, t_7, t_o \vdash o$, corresponde ao cenário $Sc_2C1 \oplus T1$. Ou seja, os seguintes da Lógica Linear a serem provados são iguais. Semelhante situação ocorre para o cenário candidato $Sc_2C1 \times Sc_1T1$. O seguinte a ser provado para este cenário é $i, t_i, t_1, t_3, t_4, t_6, t_o \vdash o$, que é o seguinte provado no cenário $Sc_1C1 \oplus T1$, apresentado no Apêndice A.3. Logo, as árvores de prova para os cenários $Sc_2C1 \oplus T1$ e $Sc_1C1 \oplus T1$ apresentadas no Apêndice A.3 podem ser reutilizadas.

Assim, considerando as árvores de prova para os cenários candidatos, deve-se verificar se as condições a) e b), apresentadas na seção 3.4 são satisfeitas para cada um dos cenários candidatos. Caso positivo, o cenário candidato analisado é um cenário livre de *deadlock*.

Considerando as árvores de prova apresentadas para os cenários candidatos, no Apêndice A.3, nota-se que apenas um átomo o foi produzido em cada árvore de prova. Nota-se também que não há nenhum átomo disponível para consumo no último seguinte destas árvores de prova. Logo, prova-se que os dois cenários analisados finalizam corretamente e, quando a execução de qualquer um destes cenários finaliza, não há tarefas remanescentes a serem tratadas. Assim, os cenários $Sc_1C1 \times Sc_2T1$ e $Sc_2C1 \times Sc_1T1$ são cenários livres de *deadlock*.

Finalmente, deve-se observar que, considerando o sistema composto da Figura 22, os cenários livres de *deadlock* detectados podem prover todas as necessidades de negócio entre o cliente (*customer*) e um serviço de bilheteria (*ticket service*), isto é, o cenário $Sc_1C1 \times Sc_2T1$ é aquele no qual o cliente efetua o pagamento utilizando um cartão de crédito VISA e o cenário $Sc_2C1 \times Sc_1T1$ é aquele no qual dinheiro eletrônico (*eCash*) é utilizado

como forma de pagamento. Assim, os cenários livres de *deadlock* formalizam os serviços oferecidos pelo sistema composto. Para uma melhor visualização destes serviços, deve-se considerar o grafo de precedência associado a cada um deles. Para a construção de tais grafos, deve-se considerar as árvores de prova da Lógica Linear rotuladas, apresentadas no Apêndice A.3.1. O primeiro serviço, $Sc_1C1 \times Sc_2T1$, é formalizado pelo grafo de precedência apresentado na Figura 23. Já o grafo de precedência da Figura 24 formaliza o segundo serviço, $Sc_2C1 \times Sc_1T1$.

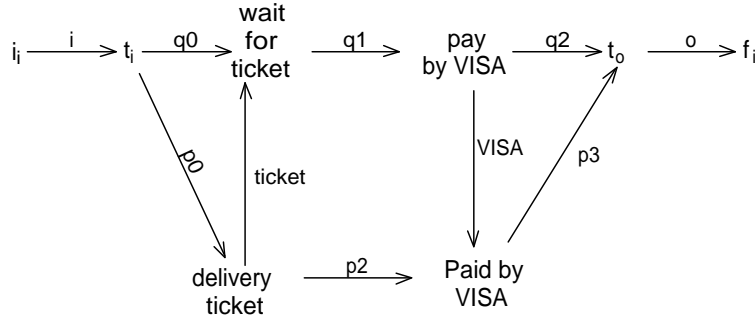


Figura 23 – Grafo de precedência para o serviço $Sc_1C1 \times Sc_2T1$.

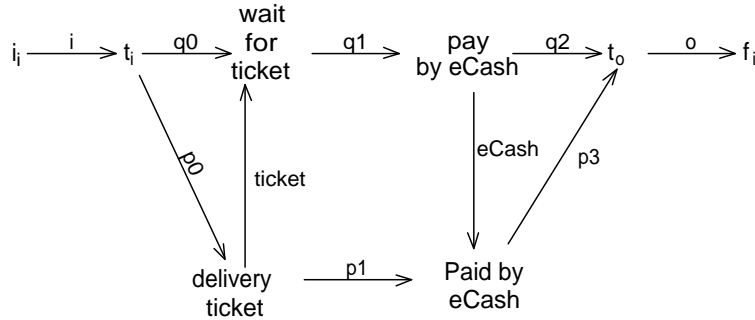


Figura 24 – Grafo de precedência para o serviço $Sc_2C1 \times Sc_1T1$.

4.3 Avaliação dos Resultados

Inicialmente, a seção 4.1 apresentou a relação direta entre a modelagem de processos de *workflow* e a modelagem de serviços Web. Conforme apresentado, um serviço Web pode ser modelado através de uma WorkFlow net suplementada de uma interface, ou seja, canais de comunicação ou mensagens dirigidas. Além disso, foi apresentada a noção de um sistema composto que corresponde a uma WorkFlow net Interorganizacional que considera exatamente dois processos de *workflow* locais.

Assim, foi possível aplicar, na seção 4.2, os métodos propostos no Capítulo 3 à verificação do sistema composto $C1 \oplus T1$.

A subseção 4.2.1 apresentou a verificação do critério de correção *Soundness* para o sistema composto $C1 \oplus T1$. Com a aplicação do método, conclui-se que o sistema analisado não satisfaz o critério de correção *Soundness*. De fato, isso ocorre uma vez que este sistema pode ser levado a situações de *deadlock*. Nesta subseção também foi apresentado como deve ocorrer a adaptação nos sequentes da Lógica Linear para que as árvores de prova construídas para provar tais sequentes possam ser reutilizadas para a verificação dos critérios de correção *Relaxed Soundness* e *Weak Soundness*, bem como no método para detecção de cenários livres de *deadlock*.

Já na subseção 4.2.2, a verificação do critério de correção *Relaxed Soundness* para o sistema composto $C1 \oplus T1$ foi apresentada. Com a aplicação do método, conclui-se que o sistema analisado satisfaz o critério de correção *Relaxed Soundness*. De fato, isso ocorre já que cada transição existente no sistema analisado aparece em pelo menos um cenário que finaliza corretamente.

Na sequência, na subseção 4.2.3, é apresentada a verificação do critério de correção *Weak Soundness* para o sistema composto $C1 \oplus T1$. Conforme mostra a conclusão da aplicação do método, o sistema composto analisado não satisfaz o critério *Weak Soundness*. Mais uma vez o resultado obtido no método está correto uma vez que o critério *Weak Soundness* garante ausência de *deadlock*, o que não ocorre no sistema analisado.

Finalmente, na subseção 4.2.4, os cenários livres de *deadlock* para o sistema composto $C1 \oplus T1$ são detectados. Dois cenários são inicialmente identificados como cenários candidatos. Posteriormente, verifica-se que tais cenários são, de fato, livres de *deadlock*. Conforme apresentado, os cenários livres de *deadlock* podem prover todas as necessidades de negócio entre os serviços compostos.

Considerando tais resultados, algumas observações devem ser destacadas. A primeira diz respeito ao reuso das árvores de prova da Lógica Linear, construídas para a prova do critério de correção *Soundness*. A apresentação do estudo de caso possibilitou a verificação de que o reuso de fato ocorre. Após a construção das árvores de prova para a prova do critério de correção *Soundness*, apenas a análise de tais árvores de prova ocorreu nos métodos subsequentes. A adaptação dos sequentes da Lógica Linear possibilitou não somente a verificação do critério de correção *Soundness*, mas também a verificação dos critérios de correção *Relaxed Soundness* e *Weak Soundness* para o sistema composto analisado. Ou seja, se o sistema composto não é *Sound*, outras propriedades do sistema puderam ser verificadas. Além disso, as árvores de prova foram ainda reutilizadas para a detecção dos cenários livres de *deadlock*. Quando considera-se as abordagens tradicionais para verificação dos critérios de correção *Soundness*, *Relaxed Soundness* e *Weak Soundness*, tal possibilidade de reuso não é alcançada. Isso porque o critério de correção *Soundness* é verificado baseado na construção e análise de um grafo das marcações acessíveis para o sistema composto que deve satisfazer as propriedades de limitabilidade e vivacidade. Tal grafo pode ser reutilizado para a verificação do critério de correção *Weak*

Soundness, mas, no caso da verificação do critério *Relaxed Soundness*, é necessário verificar, além da ausência de transições mortas, invariantes de transição, conforme mostra Dehnert (2003). Logo, o reuso não ocorre de forma direta e suficiente, como no caso dos métodos apresentados no presente trabalho.

Além disso, diferentemente do que ocorre nas abordagens propostas por Martens (2005) e Xiong, Fan e Zhou (2010), não foi necessária a remodelagem do sistema composto, nem mesmo a exclusão da possibilidade de utilização do módulo de *workflow* T1 em qualquer composição de serviços Web. De fato, a aplicação do método para detecção de cenários livres de *deadlock* possibilitou a detecção de cenários suficientes para atender às necessidades do sistema composto analisado. Se somente estes cenários livres de *deadlock* são seguidos, o sistema composto nunca será levado a uma situação de *deadlock*.

Um outro aspecto importante a ser destacado é que os sequentes da Lógica Linear mostram de modo explícito o efeito da composição dos serviços sobre as possibilidades de colaboração existentes, o que não ocorre quando uma abordagem baseada na construção e análise de grafos das marcações acessíveis é utilizada. Para uma melhor compreensão, considere que uma abordagem tradicional, como a proposta por Aalst (1998b), é utilizada para a verificação do critério de correção *Soundness*, por exemplo, para o sistema composto $C1 \oplus T1$. Neste caso, deve-se considerar o grafo das marcações acessíveis para $\overline{C1} \oplus \overline{T1}$ (*short-circuited net* para o sistema composto $C1 \oplus T1$ que, conforme mostram Aalst et al. (2011), é a rede de Petri obtida conectando os lugares o e i através de uma transição auxiliar, tornando, assim, a rede cíclica). Além disso, deve-se considerar o grafo das marcações acessíveis para os módulos $\overline{C1}$ e $\overline{T1}$, individualmente.

A Figura 25, mostra o módulo $\overline{C1}$ implementado utilizando o simulador CPN Tools (RATZER et al., 2003) e o grafo das marcações acessíveis para o mesmo. Já a Figura 26 apresenta a implementação do módulo $\overline{T1}$ e seu respectivo grafo das marcações acessíveis. Considerando a análise destes grafos, tem-se, em ambos os casos, um único componente fortemente conectado e nenhuma marcação morta, ou seja, nenhuma situação de *deadlock*. Logo, as redes de Petri $\overline{C1}$ e $\overline{T1}$ são vivas e satisfazem o critério de correção *Soundness*.

Assim, deve-se analisar a rede que representa o sistema composto $C1 \oplus T1$. A Figura 27 mostra a rede $\overline{C1} \oplus \overline{T1}$ implementada utilizando o simulador CPN Tools. O grafo das marcações acessíveis para o sistema apresentado na Figura 27, gerado pelo CPN Tools, é apresentado na Figura 28. Considerando a análise deste grafo, é possível verificar que os nós 7 e 10 representam as marcações mortas, ou seja, as situações de *deadlock*. As Figuras 29 e 30 mostram, respectivamente, tais situações de *deadlock*. Além disso, é possível identificar três componentes fortemente conectados, o que mostra, neste caso, que a rede de Petri analisada não é viva e, conseqüentemente, não satisfaz o critério de correção *Soundness*. No entanto, nenhuma informação sobre as possibilidades de colaboração existentes entre os serviços C1 e T1 pode ser extraída do grafo apresentado na Figura 28.

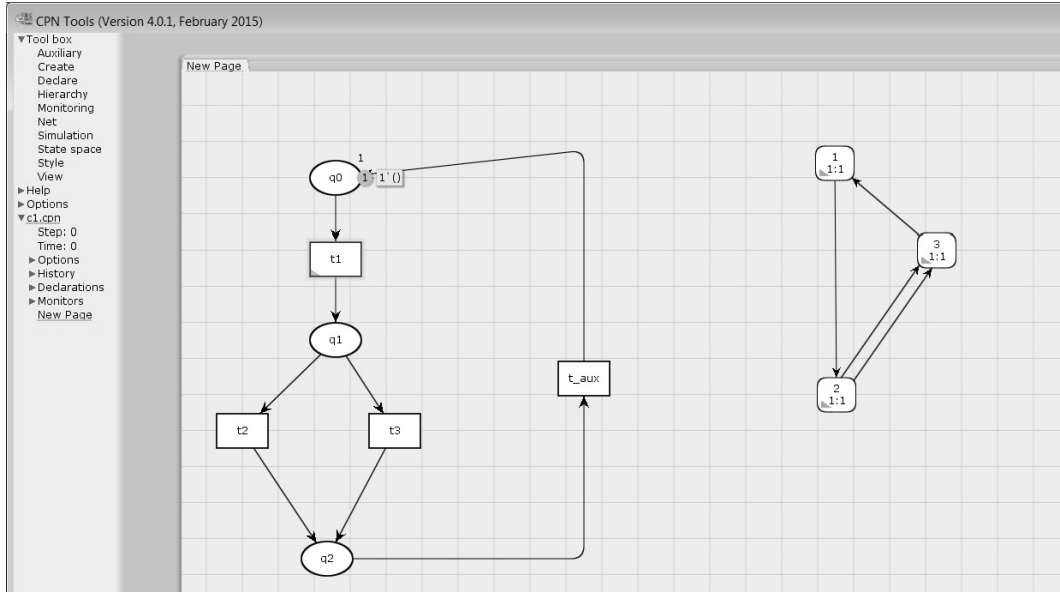


Figura 25 – O módulo $\overline{C1}$ implementado utilizando o simulador CPN Tools e o respectivo grafo das marcações acessíveis.

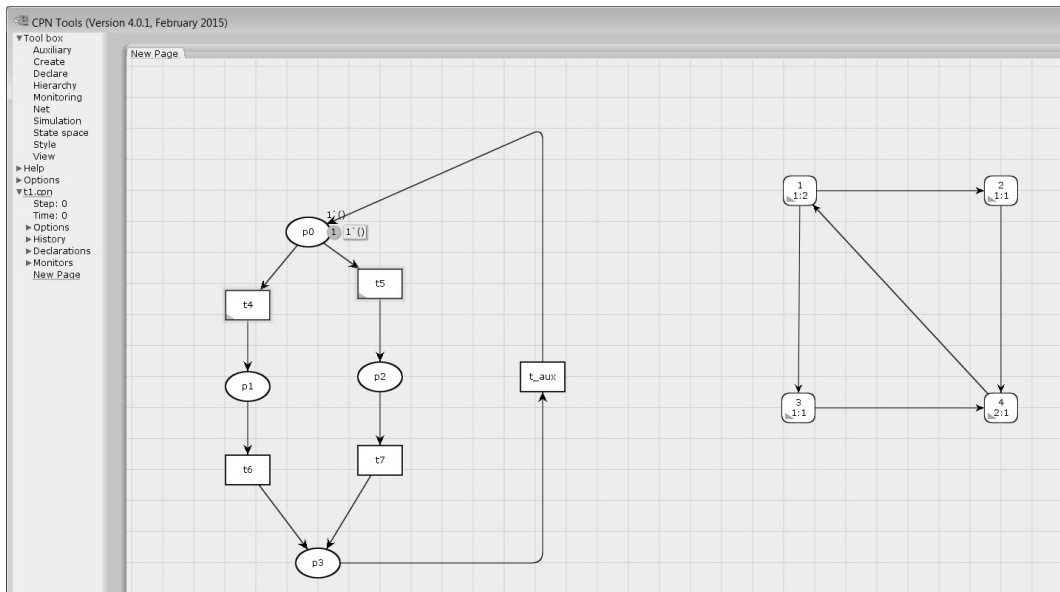


Figura 26 – O módulo $\overline{T1}$ implementado utilizando o simulador CPN Tools e o respectivo grafo das marcações acessíveis.

O fato dos sequentes da Lógica Linear mostrarem de forma explícita as colaborações existentes na composição de serviços decorre do fato das árvores de prova da Lógica Linear manterem a estrutura acíclica dos sistemas compostos. Por exemplo, no caso do sistema composto $C1 \oplus T1$, foram detectados dois cenários livres de *deadlock*, os cenários $Sc_1C1 \times Sc_2T1$ e $Sc_2C1 \times Sc_1T1$, que são definidos, respectivamente, pelos sequentes $i, t_i, t_1, t_2, t_5, t_7, t_o \vdash o$ e $i, t_i, t_1, t_3, t_4, t_6, t_o \vdash o$. Conforme mostra a Figura 20, no primeiro sequente, as transições t_1 , t_2 , t_5 e t_7 correspondem, respectivamente, às tarefas *Wait for Ticket*, *Pay by VISA*, *Delivery Ticket* e *Paid by VISA*. Já no segundo sequente, as

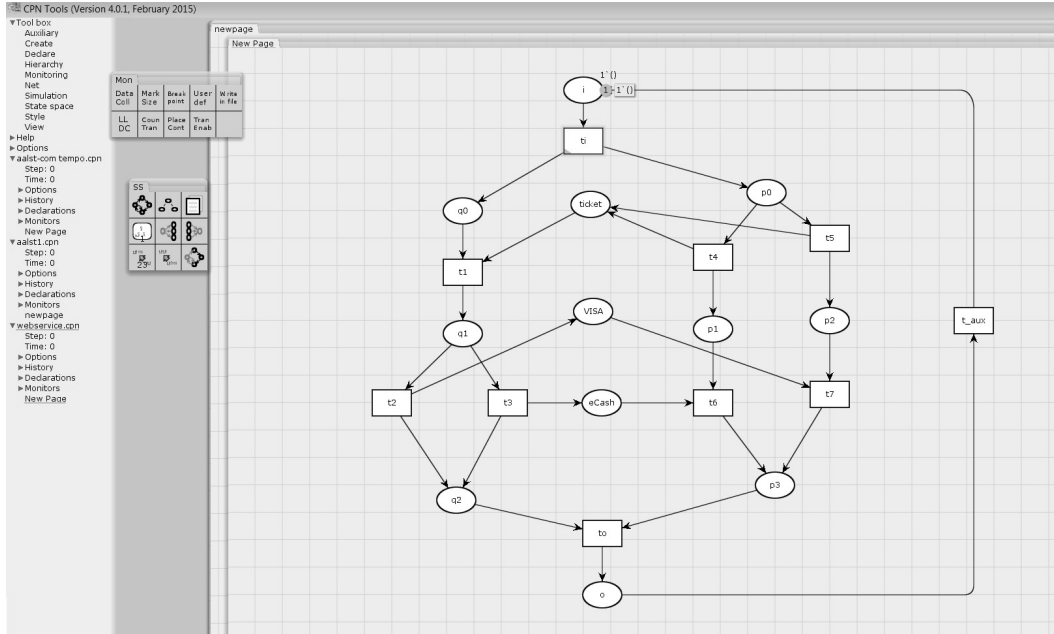


Figura 27 – O sistema composto $\overline{C1} \oplus \overline{T1}$ implementado utilizando o simulador CPN Tools.

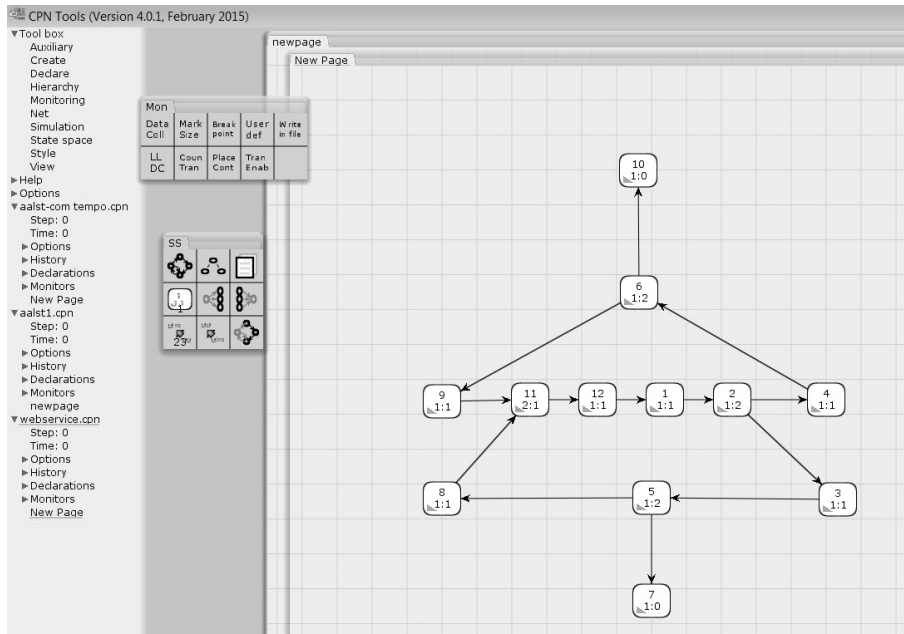


Figura 28 – Grafo das marcações acessíveis para o sistema composto $\overline{C1} \oplus \overline{T1}$, gerado pelo simulador CPN Tools.

transições t_1 , t_3 , t_4 e t_6 correspondem, respectivamente, às tarefas *Wait for Ticket*, *Pay by eCash*, *Delivery Ticket* e *Paid by eCash*. Claramente, o primeiro sequente refere-se ao pagamento do bilhete utilizando um cartão VISA e o segundo sequente ao pagamento do bilhete através de dinheiro eletrônico. Assim, as duas possibilidades de colaboração existentes são a compra do bilhete efetuando o pagamento com cartão VISA ou eCash, quando considera-se a composição dos serviços $C1$ e $T1$. De forma mais objetiva que

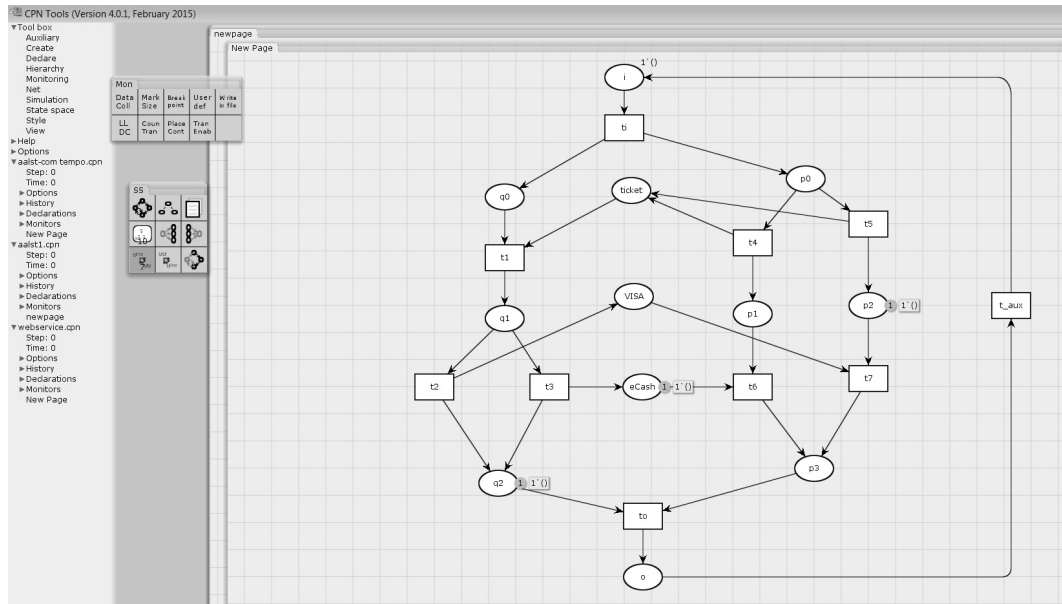


Figura 29 – Marcação referente ao nó 7 do grafo das marcações acessíveis apresentado na Figura 28.

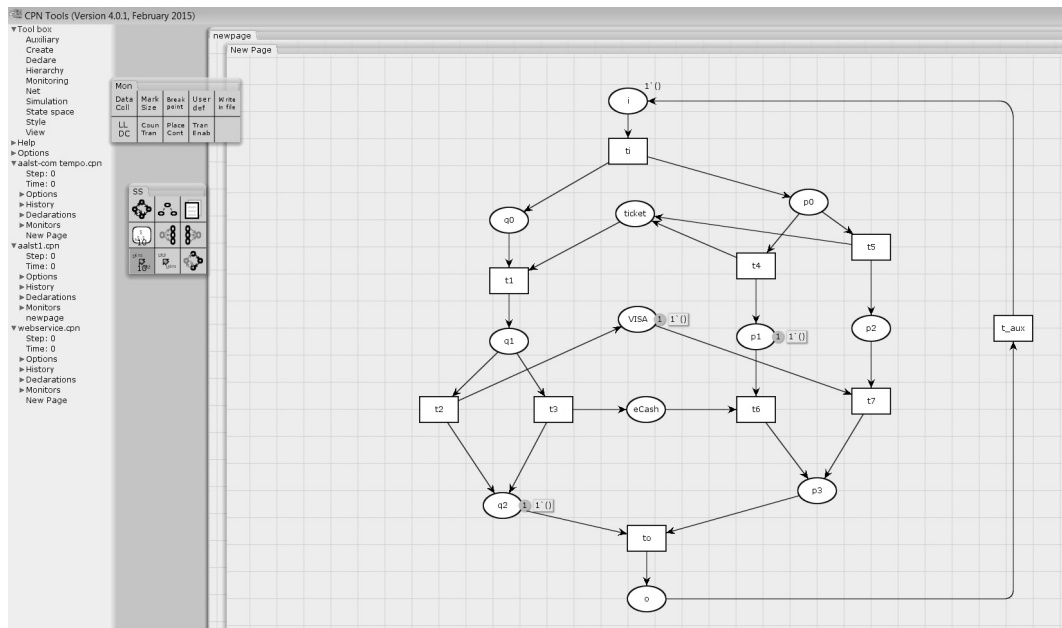


Figura 30 – Marcação referente ao nó 7 do grafo das marcações acessíveis apresentado na Figura 28.

os seguintes, os grafos de precedência associados aos referidos cenários, apresentados nas Figuras 23 e 24, permitem identificar de forma objetiva o efeito da composição dos serviços em relação às possibilidades de colaboração existentes, diferentemente do que ocorre com abordagens baseadas na construção e análise de grafos das marcações acessíveis, como anteriormente apresentado.

Finalmente, o estudo de caso possibilitou uma comparação que considera o número de árvores de prova da Lógica Linear a serem construídas e analisadas. Supondo que,

isoladamente, deseja-se verificar somente o critério de correção *Soundness* para o sistema composto $C1 \oplus T1$, oito árvores de prova da Lógica Linear devem ser construídas e analisadas: as árvores de prova para os cenários Sc_1C1 , Sc_2C1 , Sc_1T1 , Sc_2T1 , $Sc_1C1 \oplus T1$, $Sc_2C1 \oplus T1$, $Sc_3C1 \oplus T1$ e $Sc_4C1 \oplus T1$. Por outro lado, se deseja-se somente detectar os cenários livres de *deadlock* para o sistema composto $C1 \oplus T1$, seis árvores de prova serão construídas e analisadas: as árvores de prova para os cenários Sc_1C1 , Sc_2C1 , Sc_1T1 , Sc_2T1 e as árvores de prova para os cenários candidatos: $Sc_2C1 \times Sc_1T1$ (ou $Sc_1C1 \oplus T1$) e $Sc_1C1 \times Sc_2T1$ (ou $Sc_2C1 \oplus T1$). Assim, no que se diz respeito às árvores de prova para os cenários dos módulos de *workflow*, não há nenhuma redução em relação ao seu número. De fato, os cenários para os módulos de *workflow* são exatamente os mesmos nos métodos para verificação do critério de correção *Soundness* e detecção de cenários livres de *deadlock*. Entretanto, no que se diz respeito às árvores de prova que consideram os cenários do sistema composto, para a verificação de *Soundness* quatro árvores de prova devem ser construídas e, para o método de detecção de cenários livres de *deadlock*, somente duas árvores de prova devem ser construídas, uma vez que há somente dois cenários candidatos. Observa-se assim, no caso do sistema composto $C1 \oplus T1$, uma redução de cinquenta por cento de cenários a serem analisados e, conseqüentemente, das árvores de prova a serem construídas. A redução do número de provas a serem construídas e analisadas ocorre somente quando o sistema considerado para verificação não satisfaz o critério de correção *Soundness*, o que ocorre na maioria das vezes de acordo com os resultados apresentados por Fahland et al. (2011). Caso o sistema satisfaça tal critério, o número de árvores de prova será exatamente o mesmo nos métodos para verificação do critério de correção *Soundness* e detecção de cenários livres de *deadlock*, apresentados nesta pesquisa.

Planejamento de Recursos e Monitoração de Cenários Livres de *Deadlock* utilizando Árvore de Prova da Lógica Linear

Este capítulo mostra como as informações derivadas das árvores de prova da Lógica Linear podem ser utilizadas para o planejamento de recursos em processos de *workflow* interorganizacionais e composições de serviços Web. Além disso, mostra como tais informações podem ser utilizadas na monitoração da execução dos cenários livres de *deadlock* em tais processos e composições.

5.1 Planejamento de Recursos: uma análise quantitativa

O principal objetivo do planejamento de recursos, apresentado neste trabalho, é calcular janelas de datas onde podem ocorrer a utilização de um dado recurso (como uma pessoa, equipe, empresa terceirizada, etc.), para o tratamento das tarefas em um processo de *workflow* interorganizacional ou em uma composição de serviços Web. Assim, é possível prever a disponibilidade desejada dos recursos envolvidos na execução das atividades existentes no processo ou composição analisada.

No caso das IOWF-nets e das composições de serviços Web analisadas utilizando árvores de prova da Lógica Linear, as datas de execução das tarefas podem ser dadas através de datas simbólicas, em vez de datas numéricas. A maior vantagem da utilização de datas simbólicas é que quando já calculadas, estas podem ser utilizadas diretamente para qualquer instância a ser tratada pelo processo de *workflow* interorganizacional ou composição de serviços Web analisada – mesmo considerando casos cujo instante de iniciação

e/ou duração das tarefas sejam diferentes. Através das árvores de prova com cálculo de datas – que consideram diretamente a estrutura da rede analisada e não os intervalos de datas numéricas associados às tarefas –, datas simbólicas para a execução das tarefas envolvidas no processo de *workflow* interorganizacional, ou composição de serviços Web, são derivadas.

O planejamento de recursos deve considerar todos os cenários livres de *deadlock*, que são obtidos através da aplicação do método para Detecção de Cenários Livres de *Deadlock* em Processos de *Workflow* Interorganizacionais, apresentado na seção 3.4.

Para cada tarefa do tipo usuário no modelo analisado, deve-se extrair as datas de produção, D_P , e consumo, D_C , do átomo que representa a pré-condição da transição correspondente a esta tarefa. Quando há mais de uma pré-condição associada à transição, considera-se a data máxima das produções dos átomos correspondentes a estas pré-condições. A data de produção deste átomo, D_P , corresponde ao início da execução da tarefa associada à transição e a data de consumo, D_C , corresponde ao término da execução da mesma. Assim, é gerado um intervalo $[D_P, D_C]$ de datas onde o recurso que executará a referida tarefa deve estar disponível para realizá-la.

Uma vez que as datas de produção e consumo são dependentes de durações de sensibilização d_i , cujo valor pertence a um intervalo de tempo $\Delta_i = [\delta_{i_{min}}, \delta_{i_{max}}]$, pode-se considerar vários intervalos possíveis de execução das tarefas, de acordo com um planejamento estratégico. Por exemplo, o intervalo de execução $I_{Exec} = [D_{P_{min}}, D_{C_{max}}]$ considera que a alocação do recurso para a execução da tarefa poderá ocorrer entre o início ao mais cedo e o término ao mais tarde da tarefa considerada. Este intervalo de datas é o mais flexível, no sentido de que ele considera a janela de tempo de utilização do recurso mais extensa possível.

De acordo com a disponibilidade do recurso envolvido, poderão ser consideradas na estratégia de escalonamento de atividades do processo analisado outras datas fornecidas pela árvore de prova, como $D_{P_{max}}$ (data de início ao mais tarde da tarefa) ou $D_{C_{min}}$ (data de término ao mais cedo da tarefa).

A informação sobre quando o processo global finalizará, para um dado cenário livre de *deadlock*, é dada pela data de produção do átomo o no referido cenário. Assim, pode-se calcular a data ao mais cedo de finalização do processo, dada pela data mínima $D_{P_{min}}$ de produção deste átomo e a data ao mais tarde de finalização do processo, dada pela data máxima $D_{P_{max}}$ de produção do átomo o , para o cenário desejado.

Para exemplificação, considere o processo de *workflow* interorganizacional modelado pela IOWF-net mostrada na Figura 6, cuja $U(\text{IOWF-net})$ é apresentada na Figura 7. Neste processo, como mostra a seção 3.4, há 4 cenários livres de *deadlock*. O cenário $Sc_1A \times Sc_3PC$ é aquele no qual um artigo é rejeitado. O cenário $Sc_2A \times Sc_2PC$ é o cenário no qual um artigo é aceito, a versão final é recebida pelo comitê de programa e o artigo publicado. Já o cenário $Sc_3A \times Sc_1PC$ é aquele no qual o comitê de programa não

recebe a versão final até a data especificada e o autor não prepara a versão final do artigo, sendo então recebida a notificação número 1. Finalmente, o cenário $Sc_4A \times Sc_1PC$ é o cenário no qual o comitê de programa não recebe a versão final até a data especificada e o autor, embora tenha sinalizado estar preparando a versão final, não envia o artigo final, sendo então recebida a notificação número 2.

Para determinar a informação sobre a finalização do processo global e também realizar o planejamento de recursos para os cenários livres de *deadlock*, é necessário construir uma árvore de prova da Lógica Linear com cálculo de datas para cada um destes cenários. As árvores de prova, com cálculo de datas, para os cenários $Sc_1A \times Sc_3PC$, $Sc_2A \times Sc_2PC$, $Sc_3A \times Sc_1PC$ e $Sc_4A \times Sc_1PC$ são apresentadas no Apêndice A.4.

A informação sobre quando o processo global finalizará, para os cenários $Sc_1A \times Sc_3PC$, $Sc_2A \times Sc_2PC$, $Sc_3A \times Sc_1PC$ e $Sc_4A \times Sc_1PC$, é dada pela data de produção do átomo o em cada um destes cenários. Estas datas de produção são extraídas das árvores de prova com cálculo de datas para estes cenários. A Tabela 8 mostra as datas de produção do átomo o em cada um dos cenários livres de *deadlock*.

Tabela 8 – Datas de produção do átomo o nos cenários livres de *deadlock*.

Cenário	Data de produção do átomo o
$Sc_1A \times Sc_3PC$	$D_i + d_i + d_1 + d_{10} + d_{11} + \max\{d_2, d_{12} + d_{13}\} + d_4 + d_o$
$Sc_2A \times Sc_2PC$	$D_i + d_i + d_1 + d_{10} + d_{11} + \max\{d_2, d_{12} + d_{14}\} + d_3 + d_5 + d_7 + d_{16} + d_{17} + d_9 + d_o$
$Sc_3A \times Sc_1PC$	$D_i + d_i + d_1 + d_{10} + d_{11} + \max\{\max\{d_2, d_{12} + d_{14}\} + d_3, d_{12} + d_{14} + d_{15}\} + d_6 + d_o$
$Sc_4A \times Sc_1PC$	$D_i + d_i + d_1 + d_{10} + d_{11} + \max\{\max\{d_2, d_{12} + d_{14}\} + d_3 + d_5, d_{12} + d_{14} + d_{15}\} + d_8 + d_o$

Considerando as datas apresentadas na Tabela 8, pode-se calcular a data ao mais cedo de finalização do processo, dada pela data mínima D_{Pmin} de produção do átomo o e a data ao mais tarde de finalização do mesmo, dada pela data máxima D_{Pmax} de produção deste átomo, para os cenários desejados. A Tabela 9 mostra os intervalos $[D_{Pmin}, D_{Pmax}]$ de produção do átomo o em cada um dos cenários livres de *deadlock*.

Para exemplificação, considere o processo de *workflow* interorganizacional mapeado na $U(IOWF-net)$ com tempo associado às transições, mostrado na Figura 31. A única diferença entre as Figuras 7 e 31 é que a primeira não possui intervalos de tempo associados às transições enquanto que a segunda associa às transições intervalos de tempo numérico. Assim, considerando os intervalos de tempo apresentados na Figura 31 e que o processo a ser tratado inicia-se na data 0, ou seja, $D_i = 0$, os intervalos de datas numéricas para a finalização do processo podem ser calculados apenas substituindo as datas simbólicas presentes na Tabela 9 pelas datas numéricas associadas às transições apresentadas na Figura 31. A Tabela 10 mostra o resultado deste processo.

Como mostra a Tabela 10, o cenário $Sc_1A \times Sc_3PC$ finalizará ao mais cedo na data 2 e ao mais tarde na data 28. Já o cenário $Sc_2A \times Sc_2PC$ terá término ao mais cedo na data 2 e ao mais tarde na data 45. As datas 2 e 31 indicam, respectivamente, as datas de

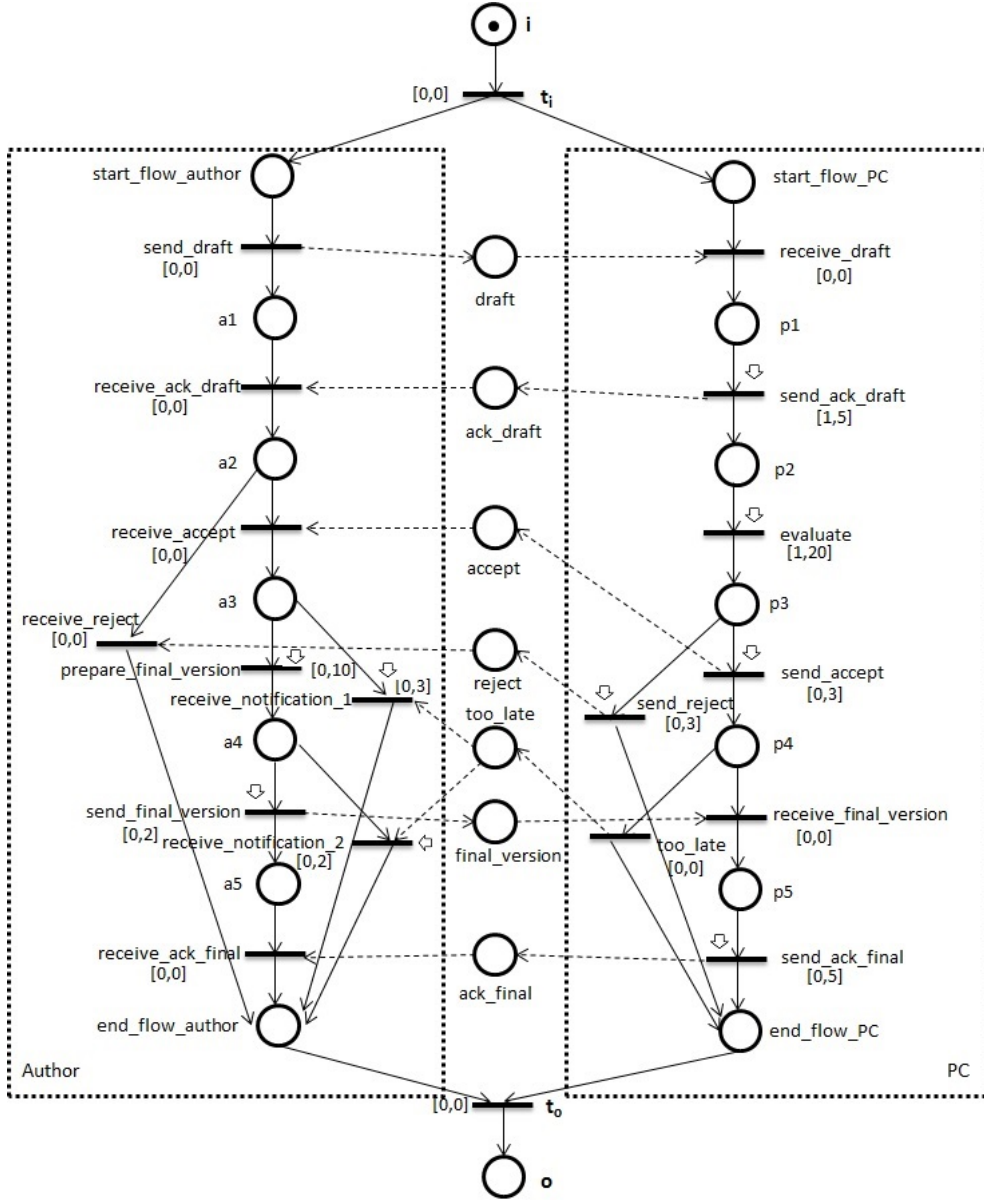


Figura 31 – $U(\text{IOWF-net})$ da Figura 7 com tempo associado às transições.

término ao mais cedo e ao mais tarde para o cenário $Sc_3A \times Sc_1PC$. Da mesma forma, as datas de término ao mais cedo e término ao mais tarde para o cenário $Sc_4A \times Sc_1PC$ são dadas por 2 e 40, respectivamente.

Para demonstrar a efetividade da abordagem proposta, simulações utilizando o simulador CPN Tools (RATZER et al., 2003) foram realizadas. Para tanto, a $U(\text{IOWF-net})$ com tempo associado às transições, apresentada na Figura 31 foi traduzida na rede de Petri colorida mostrada na Figura 32 e simulada no CPN Tools considerando 50 replicações de simulação. No simulador CPN Tools é possível executar uma simulação individual ou um número de simulações especificado através da função $CPN'Replications.nreplications$. Por exemplo, a função executada nesta pesquisa foi $CPN'Replications.nreplications \quad 50$, onde o número 50 representa o número de simulações executadas. Os tempos asso-

Tabela 9 – Intervalos de datas simbólicas, $[D_{Pmin}, D_{Pmax}]$, para a finalização dos cenários livres de *deadlock*.

Cenário	Intervalo
$Sc_1A \times Sc_3PC$	$[D_i + d_{i_{min}} + d_{1_{min}} + d_{10_{min}} + d_{11_{min}} + \max\{d_{2_{min}}, d_{12_{min}} + d_{13_{min}}\} + d_{4_{min}} + d_{o_{min}},$ $D_i + d_{i_{max}} + d_{1_{max}} + d_{10_{max}} + d_{11_{max}} + \max\{d_{2_{max}}, d_{12_{max}} + d_{13_{max}}\} + d_{4_{max}} + d_{o_{max}}]$
$Sc_2A \times Sc_2PC$	$[D_i + d_{i_{min}} + d_{1_{min}} + d_{10_{min}} + d_{11_{min}} + \max\{d_{2_{min}}, d_{12_{min}} + d_{14_{min}}\} + d_{3_{min}} + d_{5_{min}} + d_{7_{min}} + d_{16_{min}} +$ $+ d_{17_{min}} + d_{9_{min}} + d_{o_{min}},$ $D_i + d_{i_{max}} + d_{1_{max}} + d_{10_{max}} + d_{11_{max}} + \max\{d_{2_{max}}, d_{12_{max}} + d_{14_{max}}\} + d_{3_{max}} + d_{5_{max}} + d_{7_{max}} + d_{16_{max}} +$ $+ d_{17_{max}} + d_{9_{max}} + d_{o_{max}}]$
$Sc_3A \times Sc_1PC$	$[D_i + d_{i_{min}} + d_{1_{min}} + d_{10_{min}} + d_{11_{min}} + \max\{\max\{d_{2_{min}}, d_{12_{min}} + d_{14_{min}}\} + d_{3_{min}}, d_{12_{min}} + d_{14_{min}} + d_{15_{min}}\} +$ $+ d_{6_{min}} + d_{o_{min}},$ $D_i + d_{i_{max}} + d_{1_{max}} + d_{10_{max}} + d_{11_{max}} + \max\{\max\{d_{2_{max}}, d_{12_{max}} + d_{14_{max}}\} + d_{3_{max}}, d_{12_{max}} + d_{14_{max}} + d_{15_{max}}\} +$ $+ d_{6_{max}} + d_{o_{max}}]$
$Sc_4A \times Sc_1PC$	$[D_i + d_{i_{min}} + d_{1_{min}} + d_{10_{min}} + d_{11_{min}} +$ $+ \max\{\max\{d_{2_{min}}, d_{12_{min}} + d_{14_{min}}\} + d_{3_{min}} + d_{5_{min}}, d_{12_{min}} + d_{14_{min}} + d_{15_{min}}\} + d_{8_{min}} + d_{o_{min}},$ $D_i + d_{i_{max}} + d_{1_{max}} + d_{10_{max}} + d_{11_{max}} +$ $+ \max\{\max\{d_{2_{max}}, d_{12_{max}} + d_{14_{max}}\} + d_{3_{max}} + d_{5_{max}}, d_{12_{max}} + d_{14_{max}} + d_{15_{max}}\} + d_{8_{max}} + d_{o_{max}}]$

Tabela 10 – Intervalos de datas numéricas para a finalização dos cenários livres de *deadlock*.

Cenário	Intervalo
$Sc_1A \times Sc_3PC$	[2,28]
$Sc_2A \times Sc_2PC$	[2,45]
$Sc_3A \times Sc_1PC$	[2,31]
$Sc_4A \times Sc_1PC$	[2,40]

ciados ao disparo das transições do modelo seguem uma distribuição uniforme (*uniform(a:real,b:real):real*), com limites mínimo e máximo. A distribuição uniforme foi escolhida pois nela a probabilidade de cada tempo ocorrer é constante dentro do intervalo considerado. Além disso, é utilizada a função *round()*, cujo objetivo é converter o resultado da distribuição uniforme no valor inteiro mais próximo a este.

Inicialmente, a simulação foi realizada considerando a verificação da informação referente às datas de finalização do processo global. Desta forma, um monitor, denominado *oPlace*, foi adicionado ao lugar *o* do modelo mostrado na Figura 32. O relatório de replicações de simulação, com as estatísticas da simulação é apresentado na Figura 33.

Considerando a informação sobre a finalização do processo global, os resultados da simulação mostram, conforme área destacada da Figura 33, que o processo modelado na Figura 32 finaliza ao mais cedo na data 5 e ao mais tarde na data 34. Já o método analítico, isto é, o método proposto nesta pesquisa, que considera o cálculo das datas simbólicas e posteriormente a substituição destas pelas datas numéricas, produziu a data

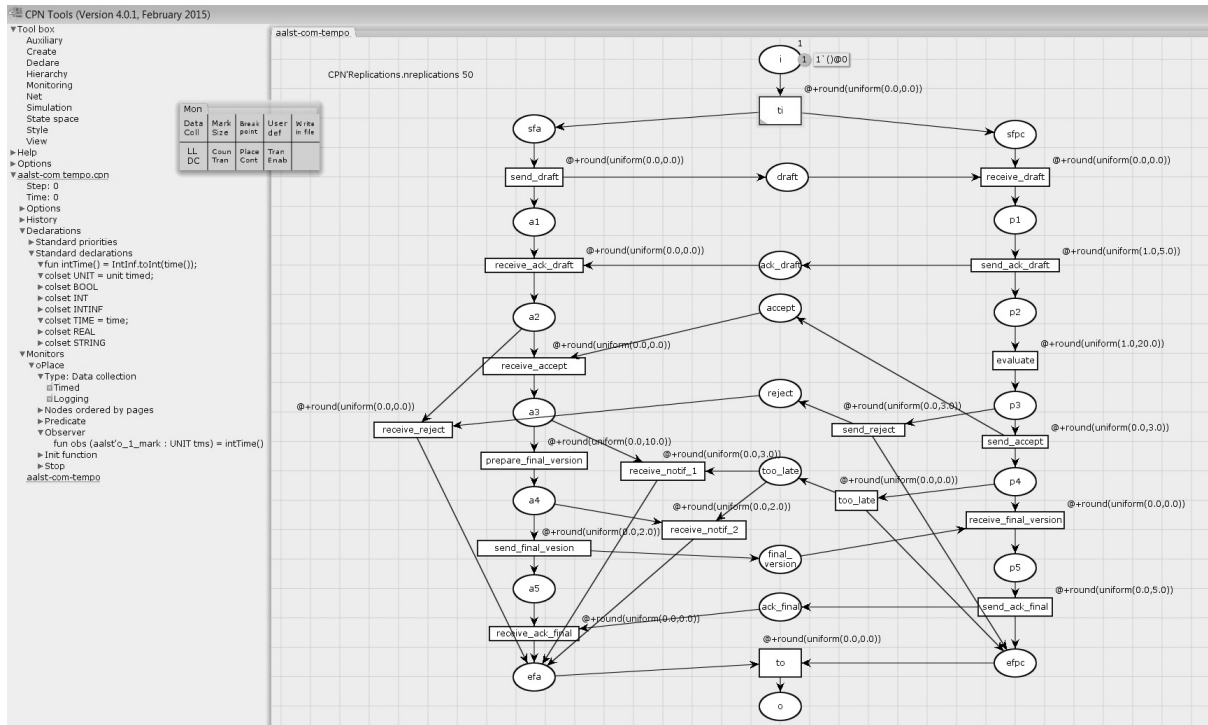


Figura 32 – Implementação no CPN Tools do modelo apresentado na Figura 31.

CPN Tools Performance Report

Net: /cydrive/C/Users/Ligia/Dropbox/Doutorado/cpn tools/aalst-com tempo.cpn
Number of replications: 50

Statistics							
Name	Avrg	90% Half Length	95% Half Length	99% Half Length	Std	Min	Max
oPlace							
count_iid	10.220000	0.281746	0.338129	0.452400	1.183043	9	12
max_iid	16.560000	1.448360	1.738204	2.325632	6.081622	5	34
min_iid	0.000000	0.000000	0.000000	0.000000	0.000000	0	0
avrg_iid	4.359541	0.453034	0.543694	0.727436	1.902275	1.400000	10.558824

Figura 33 – Relatório de replicações de simulação que considera a monitoração do lugar o .

2 de término ao mais cedo e a data 45 de término ao mais tarde, sendo esta a data ao mais tarde para o cenário $Sc_2A \times Sc_2PC$, cenário que possui maior data de término ao mais tarde. Consequentemente, é fácil notar que os valores obtidos na simulação pertencem ao intervalo numérico definido pelo método analítico.

Para validar os intervalos de datas numéricas para a finalização de cada um dos cenários livres de *deadlock*, apresentados na Tabela 10, foi realizada a simulação de cada um destes cenários, utilizando o simulador CPN Tools.

A Figura 34 mostra a implementação do cenário $Sc_1A \times Sc_3PC$ no CPN Tools. O

relatório de replicações de simulação, com as estatísticas da simulação, é apresentado na Figura 35. Como pode-se observar na área destacada desta figura, o término ao mais cedo ocorre na data 5 e o término ao mais tarde na data 25. Os resultados analíticos produziram, conforme mostra a Tabela 10, o intervalo de datas $[2, 28]$, ou seja, o término ao mais cedo na data 2 e o término ao mais tarde na data 28. Desta forma, os valores obtidos na simulação pertencem ao intervalo numérico definido pelo método analítico.

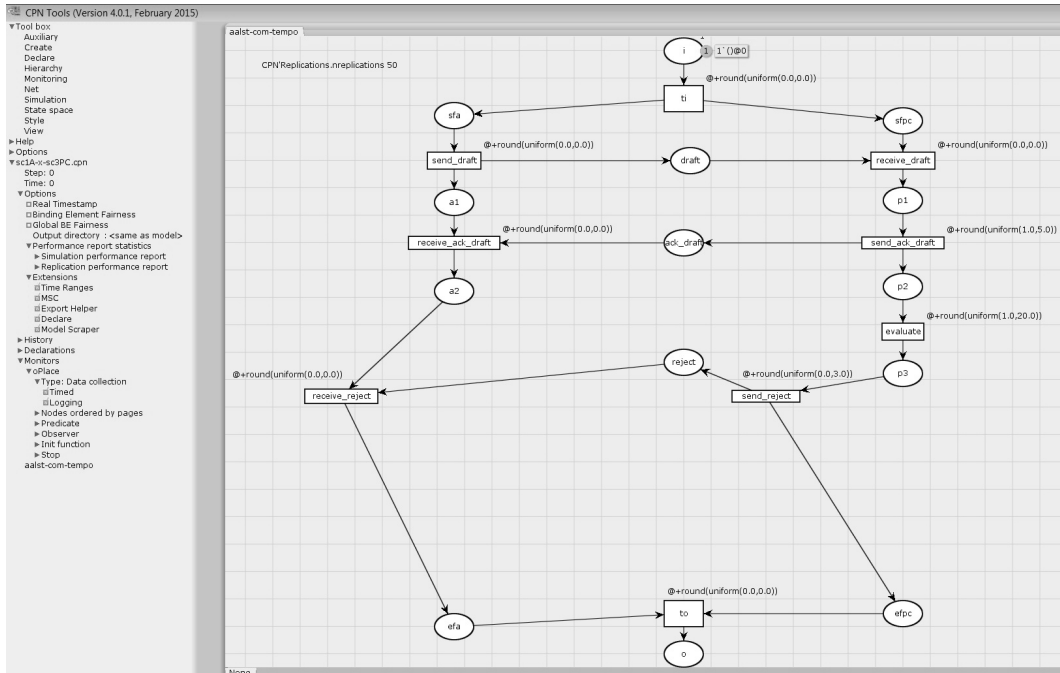


Figura 34 – Implementação do cenário $Sc_1A \times Sc_3PC$ no CPN Tools.

CPN Tools Performance Report

Net: /cygdrive/C/Users/Ligia/Desktop/testes cpn/sc1A-x-sc3PC.cpn

Number of replications: 50

Statistics							
Name	Avrg	90% Half Length	95% Half Length	99% Half Length	Std	Min	Max
oPlace							
count_iid	9.000000	0.000000	0.000000	0.000000	0.000000	9	9
max_iid	13.740000	1.253569	1.504431	2.012856	5.263699	5	25
min_iid	0.000000	0.000000	0.000000	0.000000	0.000000	0	0
avrg_iid	3.122808	0.229037	0.274872	0.367765	0.961721	0.947368	5.391304

Figura 35 – Relatório de replicações de simulação que considera a monitoração do lugar o no cenário $Sc_1A \times Sc_3PC$.

A Figura 36 apresenta a implementação do cenário $Sc_2A \times Sc_2PC$ na referida ferramenta de simulação. O relatório de replicações de simulação, com as estatísticas da simulação, é apresentado na Figura 37. Como pode-se observar na área destacada desta figura, o término ao mais cedo ocorre na data 10 e o término ao mais tarde na data 36. Os

resultados analíticos produziram, como mostra a Tabela 10, o intervalo de datas $[2, 45]$, correspondendo ao término ao mais cedo na data 2 e ao término ao mais tarde na data 45. Novamente, os valores obtidos na simulação pertencem ao intervalo numérico definido pelo método analítico.

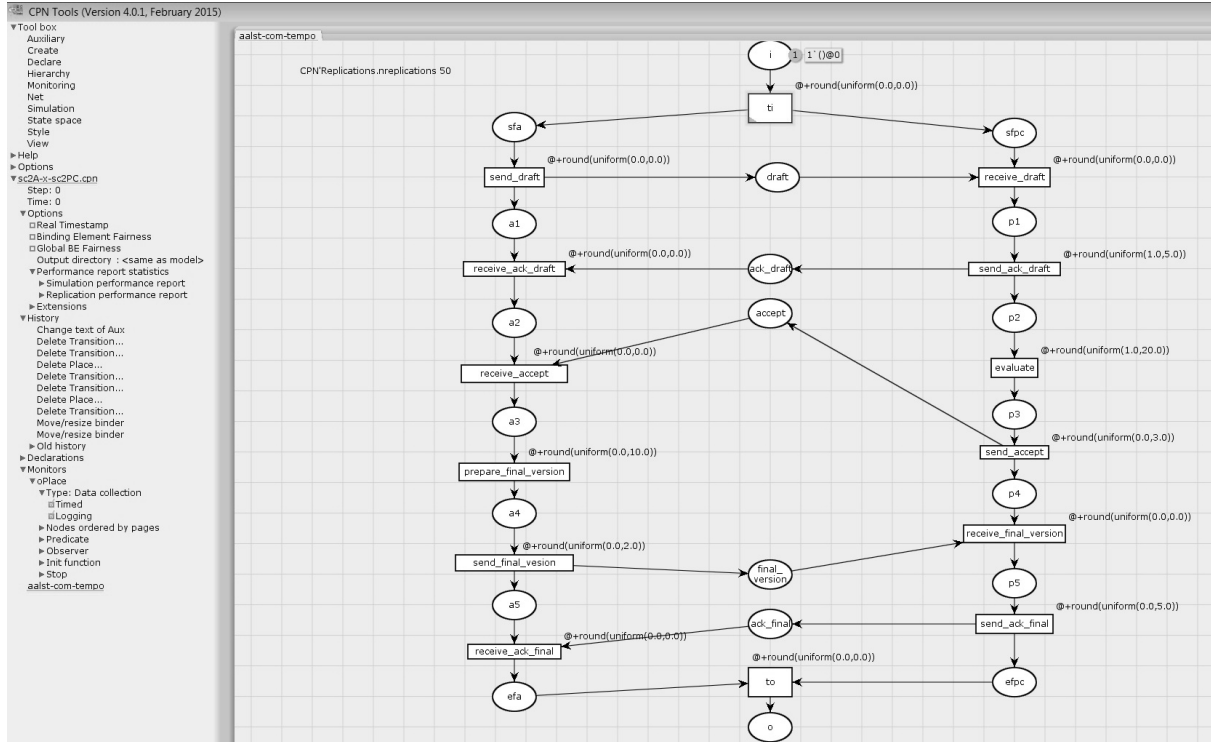


Figura 36 – Implementação do cenário $Sc_2A \times Sc_2PC$ no CPN Tools.

CPN Tools Performance Report
 Net: /cygdrive/C/Users/Ligia/Desktop/testes cpn/sc2A-x-sc2PC.cpn
 Number of replications: 50

Statistics							
Name	Avrg	90% Half Length	95% Half Length	99% Half Length	StD	Min	Max
oPlace							
count_iid	14.000000	0.000000	0.000000	0.000000	0.000000	14	14
max_iid	23.880000	1.524274	1.829310	2.447528	6.400383	10	36
min_iid	0.000000	0.000000	0.000000	0.000000	0.000000	0	0
avrg_iid	7.993288	0.480018	0.576078	0.770765	2.015581	3.400000	12.058824

Figura 37 – Relatório de replicações de simulação que considera a monitoração do lugar o no cenário $Sc_2A \times Sc_2PC$.

Já a Figura 38 mostra a implementação do cenário $Sc_3A \times Sc_1PC$ no CPN Tools. O relatório de replicações de simulação, com as estatísticas da simulação, é apresentado na Figura 39. Como pode-se observar na área destacada desta figura, o término ao mais

cedo ocorre na data 5 e o término ao mais tarde na data 27. Os resultados analíticos produziram, conforme mostra a Tabela 10, o intervalo de datas [2, 31], ou seja, término ao mais cedo na data 2 e término ao mais tarde na data 31. Outra vez, os valores obtidos na simulação pertencem ao intervalo numérico definido pelo método analítico.

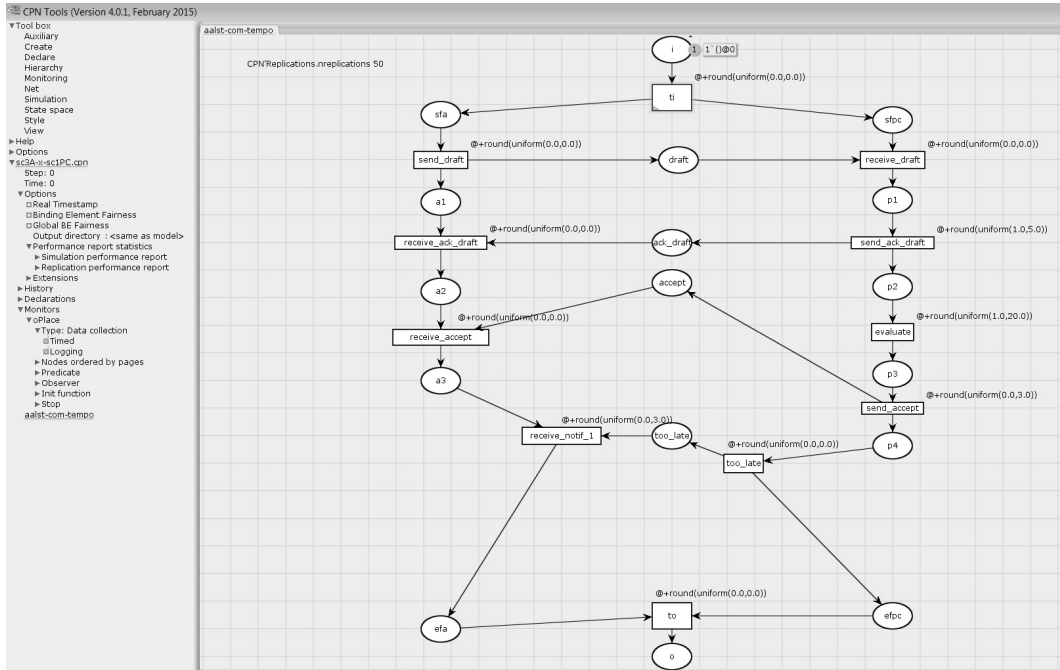


Figura 38 – Implementação do cenário $Sc_3A \times Sc_1PC$ no CPN Tools.

CPN Tools Performance Report
 Net: /cygdrive/C/Users/Ligia/Desktop/testes cpn/sc3A-x-sc1PC.cpn
 Number of replications: 50

Statistics							
Name	Avrg	90% Half Length	95% Half Length	99% Half Length	StD	Min	Max
oPlace							
count_iid	11.000000	0.000000	0.000000	0.000000	0.000000	11	11
max_iid	17.120000	1.428630	1.714525	2.293952	5.998775	5	27
min_iid	0.000000	0.000000	0.000000	0.000000	0.000000	0	0
avrg_iid	4.084460	0.295470	0.354599	0.474436	1.240670	1.400000	7.518519

Figura 39 – Relatório de replicações de simulação que considera a monitoração do lugar o no cenário $Sc_3A \times Sc_1PC$.

Finalmente, a Figura 40 mostra a implementação do cenário $Sc_4A \times Sc_1PC$. O relatório de replicações de simulação, com as estatísticas da simulação, é apresentado na Figura 41. Considerando as informações contidas na área destacada desta figura, o término ao mais cedo ocorre na data 8 e o término ao mais tarde na data 34. Os resultados analíticos produziram, como mostra a Tabela 10, o intervalo de datas [2, 40], isto é, tér-

mino ao mais cedo na data 2 e término ao mais tarde na data 40. Mais uma vez, os valores obtidos na simulação pertencem ao intervalo numérico definido pelo método analítico.

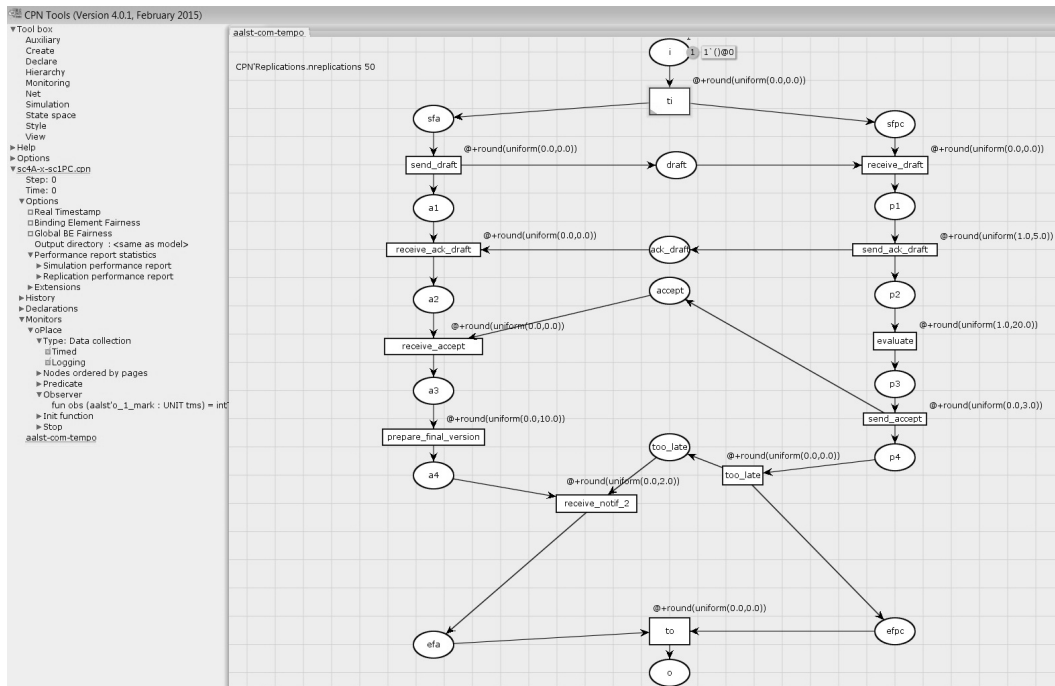


Figura 40 – Implementação do cenário $Sc_4A \times Sc_1PC$ no CPN Tools.

CPN Tools Performance Report
Net: /cygdrive/C/Users/Ligia/Desktop/testes cpn/sc4A-x-sc1PC.cpn
Number of replications: 50

Statistics							
Name	Avrg	90% Half Length	95% Half Length	99% Half Length	StD	Min	Max
oPlace							
count_iid	12.000000	0.000000	0.000000	0.000000	0.000000	12	12
max_iid	20.960000	1.603700	1.924631	2.575063	6.733892	8	34
min_iid	0.000000	0.000000	0.000000	0.000000	0.000000	0	0
avrg_iid	6.529109	0.491450	0.589798	0.789121	2.063583	2.500000	10.750000

Figura 41 – Relatório de replicações de simulação que considera a monitoração do lugar o no cenário $Sc_4A \times Sc_1PC$.

Deve-se observar que os valores obtidos pelo simulador CPN Tools para todos os cenários livres de *deadlock* pertencem aos intervalos produzidos pelas fórmulas da Lógica Linear, como sumarizado na Tabela 11, reforçando a correção do método analítico utilizado para a produção das datas simbólicas e, conseqüentemente, numéricas para a finalização de um dado processo de *workflow* interorganizacional ou composição de serviço Web.

Tabela 11 – Comparação entre as datas numéricas e simuladas para finalização dos cenários livres de *deadlock*.

Cenário	Intervalo de datas numéricas	Intervalo de datas simuladas
$Sc_1A \times Sc_3PC$	[2,28]	[5,25]
$Sc_2A \times Sc_2PC$	[2,45]	[10,36]
$Sc_3A \times Sc_1PC$	[2,31]	[2,27]
$Sc_4A \times Sc_1PC$	[2,40]	[8,34]

Para realizar o planejamento de recursos para os cenários livres de *deadlock*, deve-se considerar as árvores de prova, com cálculo de datas, para os cenários $Sc_1A \times Sc_3PC$, $Sc_2A \times Sc_2PC$, $Sc_3A \times Sc_1PC$ e $Sc_4A \times Sc_1PC$, apresentadas no Apêndice A.4.

Considere ainda que, no cenário $Sc_1A \times Sc_3PC$, as tarefas *send_ack_draft*, *evaluate* e *send_reject* são tarefas do tipo usuário, ou seja, são tarefas acionadas por recursos. Neste caso, os recursos são do tipo humano: o coordenador do evento, que sinaliza o recebimento do artigo e o membro do comitê de programa, que avalia o artigo e decide pela rejeição do mesmo. Já para o cenário $Sc_2A \times Sc_2PC$, as tarefas do tipo usuário são *send_ack_draft*, *evaluate*, *send_accept*, *prepare_final_version*, *send_final_version* e *send_ack_final*. Para o cenário $Sc_3A \times Sc_1PC$, as tarefas *send_ack_draft*, *evaluate*, *send_accept* e *receive_notification_1* são tarefas do tipo usuário. Finalmente, para o cenário $Sc_4A \times Sc_1PC$, tem-se as seguintes tarefas do tipo usuário: *send_ack_draft*, *evaluate*, *send_accept*, *prepare_final_version* e *receive_notification_2*. Logo, deve-se saber quando os recursos humanos associados a tais tarefas do tipo usuário devem estar disponíveis para a execução das mesmas, considerando os referidos cenários livres de *deadlock*.

A Tabela 12 mostra os intervalos de datas simbólicas de execução para as tarefas que são do tipo usuário dos cenários livres de *deadlock* da $U(IOWF-net)$ da Figura 7, considerando o intervalo de execução $I_{Exec} = [D_{Pmin}, D_{Cmax}]$. É importante destacar que os intervalos de datas simbólicas calculados poderão ser utilizados por qualquer caso tratado pelo processo de *workflow* interorganizacional modelado pela IOWF-net da Figura 6.

Considerando os intervalos de tempo apresentados na Figura 31 e considerando que o processo a ser tratado inicia-se na data 0, ou seja, $D_i = 0$, os intervalos de datas para a execução das tarefas podem ser calculados apenas substituindo as datas simbólicas presentes na Tabela 12 pelas datas numéricas associadas às transições apresentadas na Figura 31. A Tabela 13 mostra o resultado deste processo. Por exemplo, o recurso utilizado para executar a tarefa *evaluate* dos cenários livres de *deadlock* deverá ser alocado ao mais cedo na data 1 e deverá ser liberado ao mais tarde na data 25. Assim, o recurso que executará esta tarefa pode ser alocado para desempenhar a atividade de avaliação do artigo neste intervalo de datas.

Tabela 12 – Intervalos de datas simbólicas para execução de tarefas do tipo usuário dos cenários $Sc_1A \times Sc_3PC$, $Sc_2A \times Sc_2PC$, $Sc_3A \times Sc_1PC$ e $Sc_4A \times Sc_1PC$.

Tarefa	Intervalos de datas simbólicas para todos os cenários livres de <i>deadlock</i>
<i>send_ack_draft</i>	$[D_i + \delta_{i_{min}} + \delta_{1_{min}} + \delta_{10_{min}}, D_i + \delta_{i_{max}} + \delta_{1_{max}} + \delta_{10_{max}} + \delta_{11_{max}}]$
<i>evaluate</i>	$[D_i + \delta_{i_{min}} + \delta_{1_{min}} + \delta_{10_{min}} + \delta_{11_{min}}, D_i + \delta_{i_{max}} + \delta_{1_{max}} + \delta_{10_{max}} + \delta_{11_{max}} + \delta_{12_{max}}]$
Intervalo de datas simbólicas para $Sc_1A \times Sc_3PC$	
<i>send_reject</i>	$[D_i + \delta_{i_{min}} + \delta_{1_{min}} + \delta_{10_{min}} + \delta_{11_{min}} + \delta_{12_{min}},$ $D_i + \delta_{i_{max}} + \delta_{1_{max}} + \delta_{10_{max}} + \delta_{11_{max}} + \delta_{12_{max}} + \delta_{13_{max}}]$
Intervalo de datas simbólicas para $Sc_2A \times Sc_2PC$, $Sc_3A \times Sc_1PC$ e $Sc_4A \times Sc_1PC$	
<i>send_accept</i>	$[D_i + \delta_{i_{min}} + \delta_{1_{min}} + \delta_{10_{min}} + \delta_{11_{min}} + \delta_{12_{min}},$ $D_i + \delta_{i_{max}} + \delta_{1_{max}} + \delta_{10_{max}} + \delta_{11_{max}} + \delta_{12_{max}} + \delta_{14_{max}}]$
Intervalos de datas simbólicas para $Sc_2A \times Sc_2PC$	
<i>prepare_final_version</i>	$[D_i + \delta_{i_{min}} + \delta_{1_{min}} + \delta_{10_{min}} + \delta_{11_{min}} + \max\{\delta_{2_{min}}, \delta_{12_{min}} + \delta_{14_{min}}\} + \delta_{3_{min}},$ $D_i + \delta_{i_{max}} + \delta_{1_{max}} + \delta_{10_{max}} + \delta_{11_{max}} + \max\{\delta_{2_{max}}, \delta_{12_{max}} + \delta_{14_{max}}\} + \delta_{3_{max}} + \delta_{5_{max}}]$
<i>send_final_version</i>	$[D_i + \delta_{i_{min}} + \delta_{1_{min}} + \delta_{10_{min}} + \delta_{11_{min}} + \max\{\delta_{2_{min}}, \delta_{12_{min}} + \delta_{14_{min}}\} + \delta_{3_{min}} + \delta_{5_{min}},$ $D_i + \delta_{i_{max}} + \delta_{1_{max}} + \delta_{10_{max}} + \delta_{11_{max}} + \max\{\delta_{2_{max}}, \delta_{12_{max}} + \delta_{14_{max}}\} + \delta_{3_{max}} + \delta_{5_{max}} + \delta_{7_{max}}]$
<i>send_ack_final</i>	$[D_i + \delta_{i_{min}} + \delta_{1_{min}} + \delta_{10_{min}} + \delta_{11_{min}} + \max\{\delta_{2_{min}}, \delta_{12_{min}} + \delta_{14_{min}}\} + \delta_{3_{min}} + \delta_{5_{min}} + \delta_{7_{min}} + \delta_{16_{min}},$ $D_i + \delta_{i_{max}} + \delta_{1_{max}} + \delta_{10_{max}} + \delta_{11_{max}} + \max\{\delta_{2_{max}}, \delta_{12_{max}} + \delta_{14_{max}}\} + \delta_{3_{max}} + \delta_{5_{max}} + \delta_{7_{max}} + \delta_{16_{max}} + \delta_{17_{max}}]$
Intervalo de datas simbólicas para $Sc_3A \times Sc_1PC$	
<i>receive_notification_1</i>	$[D_i + \delta_{i_{min}} + \delta_{1_{min}} + \delta_{10_{min}} + \delta_{11_{min}} + \max\{\max\{\delta_{2_{min}}, \delta_{12_{min}} + \delta_{14_{min}}\} + \delta_{3_{min}}, \delta_{12_{min}} + \delta_{14_{min}} + \delta_{15_{min}}\},$ $D_i + \delta_{i_{max}} + \delta_{1_{max}} + \delta_{10_{max}} + \delta_{11_{max}} + \max\{\max\{\delta_{2_{max}}, \delta_{12_{max}} + \delta_{14_{max}}\} + \delta_{3_{max}}, \delta_{12_{max}} + \delta_{14_{max}} + \delta_{15_{max}}\} + \delta_{6_{max}}]$
Intervalo de datas simbólicas para $Sc_4A \times Sc_1PC$	
<i>prepare_final_version</i>	$[D_i + \delta_{i_{min}} + \delta_{1_{min}} + \delta_{10_{min}} + \delta_{11_{min}} + \max\{\delta_{2_{min}}, \delta_{12_{min}} + \delta_{14_{min}}\} + \delta_{3_{min}},$ $D_i + \delta_{i_{max}} + \delta_{1_{max}} + \delta_{10_{max}} + \delta_{11_{max}} + \max\{\delta_{2_{max}}, \delta_{12_{max}} + \delta_{14_{max}}\} + \delta_{3_{max}} + \delta_{5_{max}}]$
<i>receive_notification_2</i>	$[D_i + \delta_{i_{min}} + \delta_{1_{min}} + \delta_{10_{min}} + \delta_{11_{min}} + \max\{\max\{\delta_{2_{min}}, \delta_{12_{min}} + \delta_{14_{min}}\} + \delta_{3_{min}} + \delta_{5_{min}}, \delta_{12_{min}} + \delta_{14_{min}} + \delta_{15_{min}}\},$ $D_i + \delta_{i_{max}} + \delta_{1_{max}} + \delta_{10_{max}} + \delta_{11_{max}} + \max\{\max\{\delta_{2_{max}}, \delta_{12_{max}} + \delta_{14_{max}}\} + \delta_{3_{max}} + \delta_{5_{max}}, \delta_{12_{max}} + \delta_{14_{max}} + \delta_{15_{max}}\} + \delta_{8_{max}}]$

Tabela 13 – Intervalos de datas numéricas para execução de tarefas do tipo usuário dos cenários $Sc_1A \times Sc_3PC$, $Sc_2A \times Sc_2PC$, $Sc_3A \times Sc_1PC$ e $Sc_4A \times Sc_1PC$.

Tarefa	Intervalos de datas numéricas para todos os cenários livres de <i>deadlock</i>
<i>send_ack_draft</i>	[0,5]
<i>evaluate</i>	[1,25]
	Intervalo de datas numéricas para $Sc_1A \times Sc_3PC$
<i>send_reject</i>	[2,28]
	Intervalo de datas numéricas para $Sc_2A \times Sc_2PC$, $Sc_3A \times Sc_1PC$ e $Sc_4A \times Sc_1PC$
<i>send_accept</i>	[2,28]
	Intervalos de datas numéricas para $Sc_2A \times Sc_2PC$
<i>prepare_final_version</i>	[2,38]
<i>send_final_version</i>	[2,40]
<i>send_ack_final</i>	[2,45]
	Intervalo de datas numéricas para $Sc_3A \times Sc_1PC$
<i>receive_notification_1</i>	[2,31]
	Intervalo de datas numéricas para $Sc_4A \times Sc_1PC$
<i>prepare_final_version</i>	[2,38]
<i>receive_notification_2</i>	[2,40]

Para demonstrar a efetividade da abordagem proposta, simulações foram realizadas para a verificação dos intervalos de data para a execução das tarefas do tipo usuário nos cenários livres de *deadlock*. Para realizar tais simulações, transições auxiliares foram adicionadas aos modelos dos cenários livres de *deadlock* implementados no CPN tools. Tais transições apresentam em sua nomenclatura a palavra “aux”, para indicar sua função simplesmente auxiliar, sem semântica adicional ao processo original. Por exemplo, na Figura 42 é possível identificar as transições *sad_aux*, *e_aux* e *sr_aux*. As transições auxiliares são necessárias para obter informação sobre o início ao mais cedo e o término ao mais tarde das tarefas do tipo usuário, neste caso, *send_ack_draft*, *evaluate* e *send_reject*. Os lugares auxiliares, nomeados conforme as transições auxiliares que estes antecedem, foram adicionados somente para manter a estrutura de uma rede de Petri. Deve-se notar que, como o intervalo de datas associado às transições auxiliares é $[0, 0]$, nenhum tempo adicional foi somado à execução do modelo original. Para coletar os dados necessários, monitores são associados às transições do tipo usuário e às transições auxiliares.

A Figura 42 mostra a implementação do cenário $Sc_1A \times Sc_3PC$ com as respectivas transições auxiliares no CPN tools. O modelo foi simulado no CPN Tools considerando 50 replicações de simulação. Os dados obtidos no relatório de replicações são apresentados

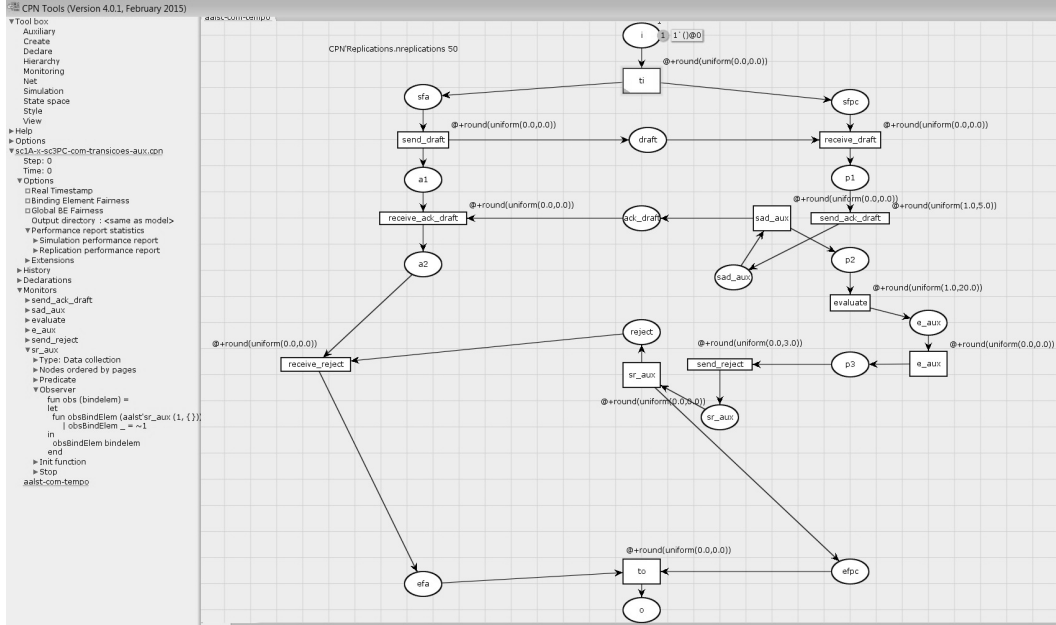


Figura 42 – Implementação cenário $Sc_1A \times Sc_3PC$ com transições auxiliares no CPN Tools.

na Tabela 14.

Tabela 14 – Intervalos de datas simuladas para execução de tarefas do tipo usuário do cenário $Sc_1A \times Sc_3PC$.

Tarefa	Intervalos de datas simuladas
<i>send_ack_draft</i>	[0,5]
<i>evaluate</i>	[1,23]
<i>send_reject</i>	[4,25]

Já a Figura 43 mostra a implementação do cenário $Sc_2A \times Sc_2PC$ com as respectivas transições auxiliares no CPN tools. O modelo foi simulado no CPN Tools considerando 50 replicações de simulação. Os dados obtidos no relatório de replicações são apresentados na Tabela 15.

Tabela 15 – Intervalos de datas simuladas para execução de tarefas do tipo usuário do cenário $Sc_2A \times Sc_2PC$.

Tarefa	Intervalos de datas simuladas
<i>send_ack_draft</i>	[0,5]
<i>evaluate</i>	[1,24]
<i>send_accept</i>	[2,24]
<i>prepare_final_version</i>	[5,31]
<i>send_final_version</i>	[11,32]
<i>send_ack_final</i>	[12,34]

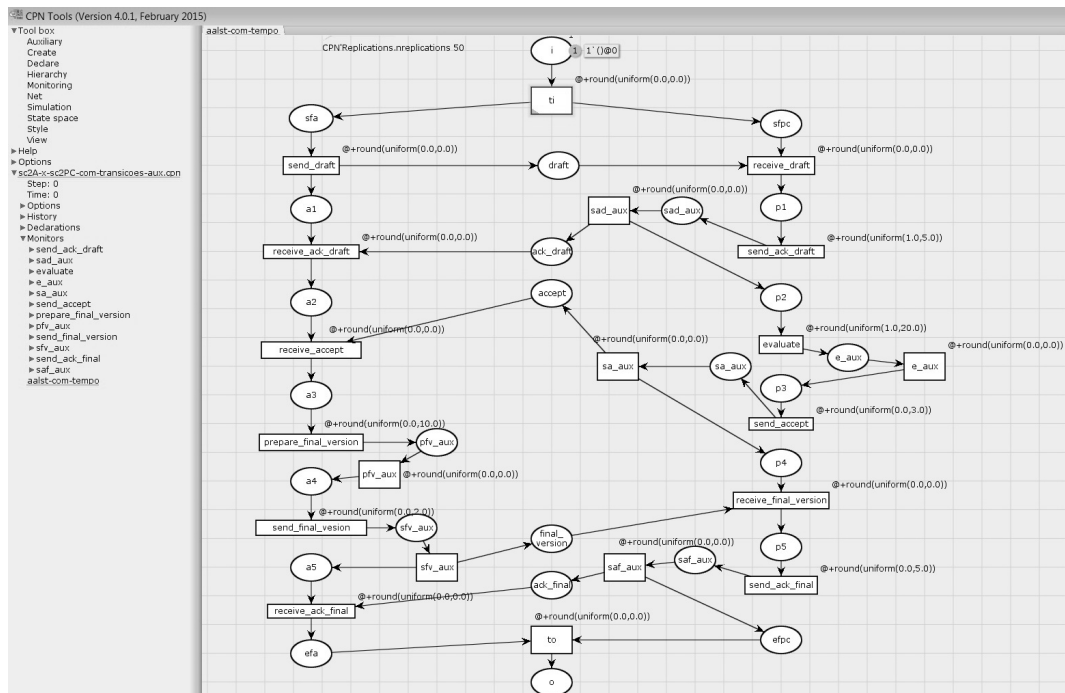


Figura 43 – Implementação cenário $Sc_2A \times Sc_2PC$ com transições auxiliares no CPN Tools.

Na Figura 44 é apresentada a implementação, no CPN Tools, do cenário $Sc_3A \times Sc_1PC$ com as respectivas transições auxiliares. Este modelo também foi simulado na referida ferramenta considerando 50 replicações de simulação. A Tabela 16 mostra os dados obtidos no relatório de replicações.

Tabela 16 – Intervalos de datas simuladas para execução de tarefas do tipo usuário do cenário $Sc_3A \times Sc_1PC$.

Tarefa	Intervalos de datas simuladas
<i>send_ack_draft</i>	[0,5]
<i>evaluate</i>	[1,23]
<i>send_accept</i>	[4,26]
<i>receive_notification_1</i>	[5,26]

Finalmente, a Figura 45 mostra a implementação do cenário $Sc_4A \times Sc_1PC$ no CPN Tools e a Tabela 17 mostra os dados obtidos no relatório de replicações de simulação, que considera um total de 50 replicações de simulação.

A Tabela 18 mostra a comparação entre os intervalos de datas numéricas e simuladas para as tarefas do tipo usuário dos cenários livres de *deadlock*. Considerando as informações apresentadas nesta tabela, conclui-se que os intervalos de datas simuladas pertencem aos intervalos de data numéricas, calculadas através do método apresentado neste trabalho, o que reforça a correção do método proposto para a obtenção de datas simbólicas e numéricas para a execução de tarefas do tipo usuário em cenários livres de *deadlock*.

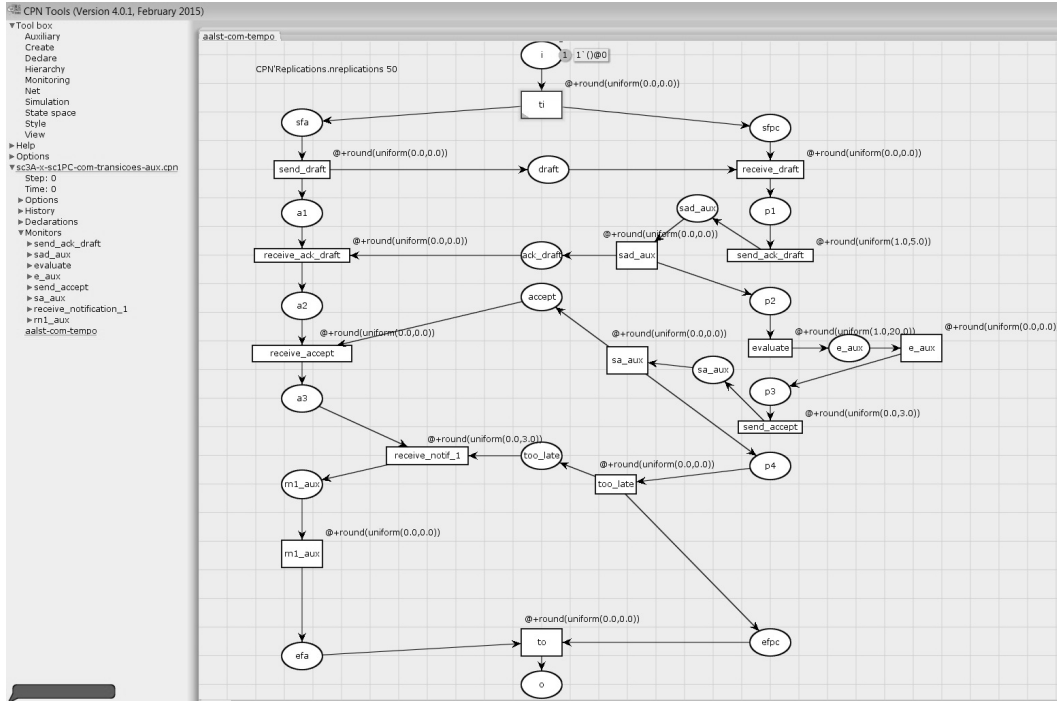


Figura 44 – Implementação cenário $Sc_3A \times Sc_1PC$ com transições auxiliares no CPN Tools.

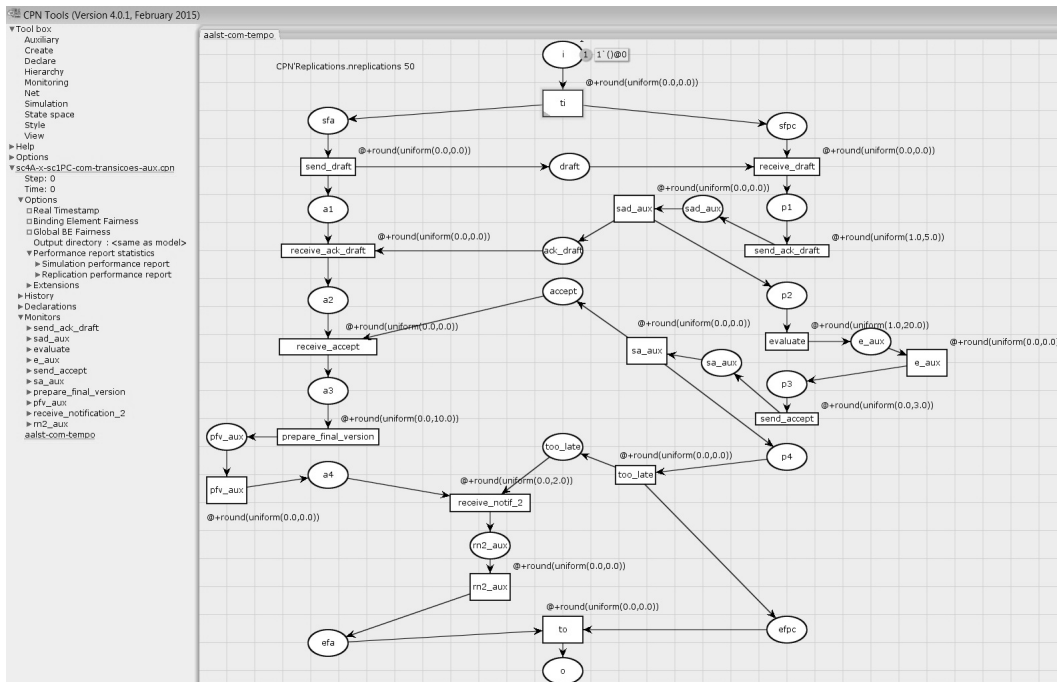


Figura 45 – Implementação cenário $Sc_4A \times Sc_1PC$ com transições auxiliares no CPN Tools.

Para finalizar, destaca-se que não é necessário construir novas árvores de prova da Lógica Linear, que contemplem o cálculo de datas, somente no contexto da análise quantitativa. Os cálculos de datas podem ser efetuados na construção das árvores de prova utilizadas para aplicação dos métodos de análise qualitativa, apresentados no Capítulo 3.

Tabela 17 – Intervalos de datas simuladas para execução de tarefas do tipo usuário do cenário $Sc_4A \times Sc_1PC$.

Tarefa	Intervalos de datas simuladas
<i>send_ack_draft</i>	[0,5]
<i>evaluate</i>	[1,25]
<i>send_accept</i>	[3,27]
<i>prepare_final_version</i>	[4,31]
<i>receive_notification_2</i>	[6,32]

Tabela 18 – Comparação entre os intervalos de datas numéricas e simuladas para execução de tarefas do tipo usuário dos cenários livres de *deadlock*.

Tarefa	Intervalos de datas numéricas	Intervalos de datas simuladas
<i>send_ack_draft</i>	[0,5]	[0,5]
<i>evaluate</i>	[1,25]	[1,23],[1,24],[1,25]
<i>send_reject</i>	[2,28]	[4,25]
<i>send_accept</i>	[2,28]	[2,24],[4,26],[3,27]
<i>prepare_final_version</i>	[2,38]	[5,31],[4,31]
<i>send_final_version</i>	[2,40]	[11,32]
<i>send_ack_final</i>	[2,45]	[12,34]
<i>receive_notification_1</i>	[2,31]	[5,26]
<i>receive_notification_2</i>	[2,40]	[6,32]

Tais cálculos de data não implicam em nenhuma alteração dos métodos apresentados para análise qualitativa. Desta forma, as árvores de prova podem ser reutilizadas tanto no que se refere à prova das variantes do critério de correção *Soundness*, à detecção de cenários livres de *deadlock*, bem como ao planejamento de recursos nesta seção apresentado. Finalmente, é importante notar que esta abrangente possibilidade de reutilização não ocorre quando métodos baseados na construção e análise de grafos das marcações acessíveis são utilizados.

5.2 Árvore de Prova da Lógica Linear e a Monitoração de Cenários Livres de *Deadlock*

O objetivo desta seção é mostrar como informações derivadas das árvores de prova da Lógica Linear podem ser utilizadas na monitoração da execução dos cenários livres de *deadlock* em processos de *workflow* interorganizacionais e composições de serviços Web.

A primeira possibilidade é seguir os grafos de precedência dos cenários livres de *deadlock*, construídos a partir das informações obtidas nas árvores de prova da Lógica Linear rotuladas, de forma que tais grafos fixem as restrições que devem ser seguidas em tempo

de execução.

Por exemplo, considerando o sistema composto $C1 \oplus T1$ apresentado na Figura 22, há dois cenários livres de *deadlock*: o cenário $Sc_1C1 \times Sc_2P1$, no qual o pagamento do bilhete é feito utilizando cartão de crédito VISA e o cenário $Sc_2C1 \times Ac_1P1$, no qual o pagamento é realizado através de dinheiro eletrônico (*eCash*). Os grafos de precedência para estes cenários são apresentados, respectivamente, nas Figuras 46 e 47.

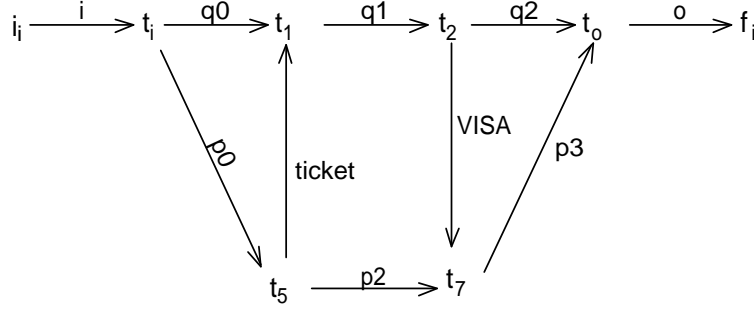


Figura 46 – Grafo de precedência para o cenário $Sc_1C1 \times Sc_2P1$.

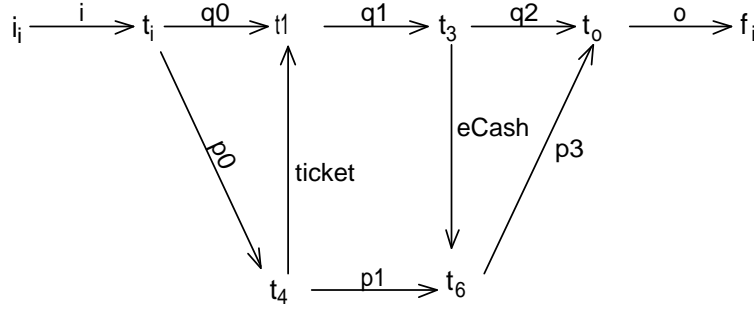


Figura 47 – Grafo de precedência para o cenário $Sc_1C1 \times Sc_2P1$.

Inicialmente, o sistema de monitoração deve considerar os dois grafos de precedência. Se o disparo da transição t_5 ocorre, o grafo de precedência da Figura 46 deve ser seguido e o grafo de precedência da Figura 47 deve ser descartado. Por outro lado, se o disparo da transição t_4 ocorre, deve-se ser seguido o grafo de precedência da Figura 47 e o outro grafo descartado. Desta forma, é fixado que, após o disparo de t_5 , devem ocorrer os disparos de t_1 , t_2 e t_7 e, após o disparo de t_4 , devem ocorrer os disparos de t_1 , t_3 e t_6 . Logo, somente os cenários livres de *deadlock* serão executados e o sistema composto $C1 \oplus T1$ nunca será levado às situações de *deadlock*.

O mesmo ocorre quando considera-se o processo de *workflow* interorganizacional modelado na Figura 7. Neste caso, há quatro cenários livres de *deadlock*: os cenários $Sc_3A \times Sc_1PC$, $Sc_4A \times Sc_1PC$, $Sc_2A \times Sc_2PC$ e $Sc_1A \times Sc_3PC$, cujos grafos de precedência são apresentados, respectivamente, nas Figuras 49, 50, 51 e 48.

Inicialmente, os quatro grafos de precedência devem ser considerados pelo sistema de monitoração. Os disparos das transições t_i , t_1 , t_{10} , t_{11} , t_{12} e t_2 não ocasionam nenhum

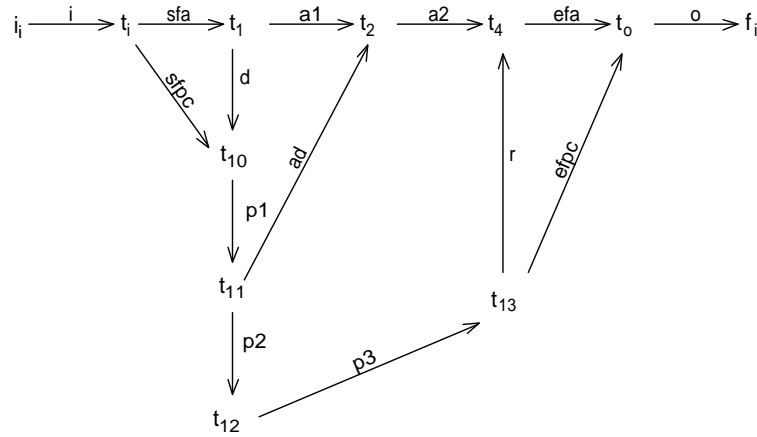


Figura 48 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_1A \times Sc_3PC$.

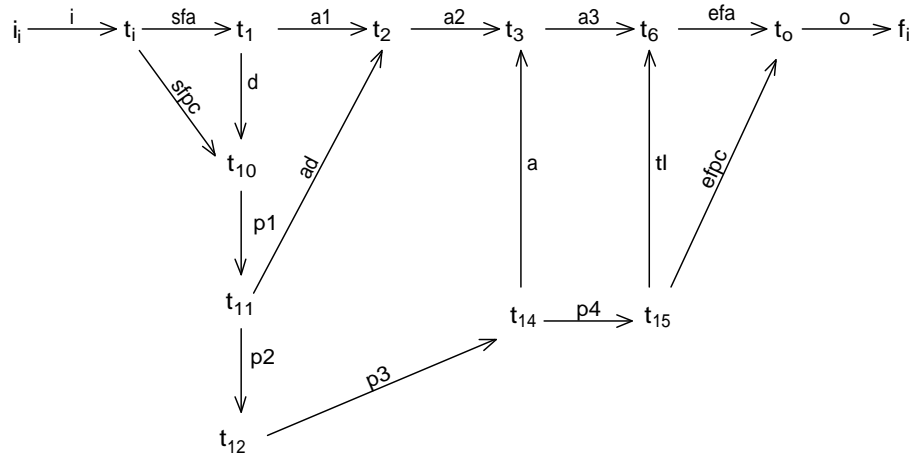


Figura 49 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_3A \times Sc_1PC$.

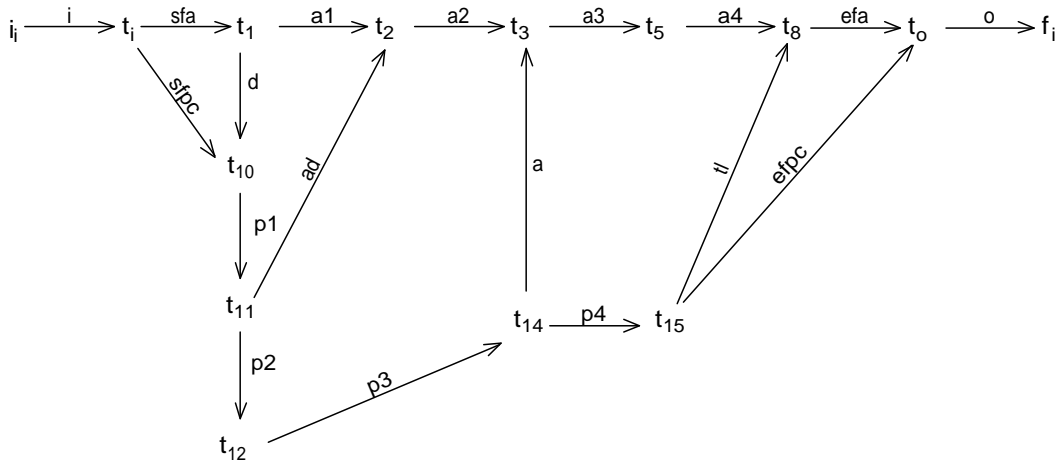


Figura 50 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_4A \times Sc_1PC$.

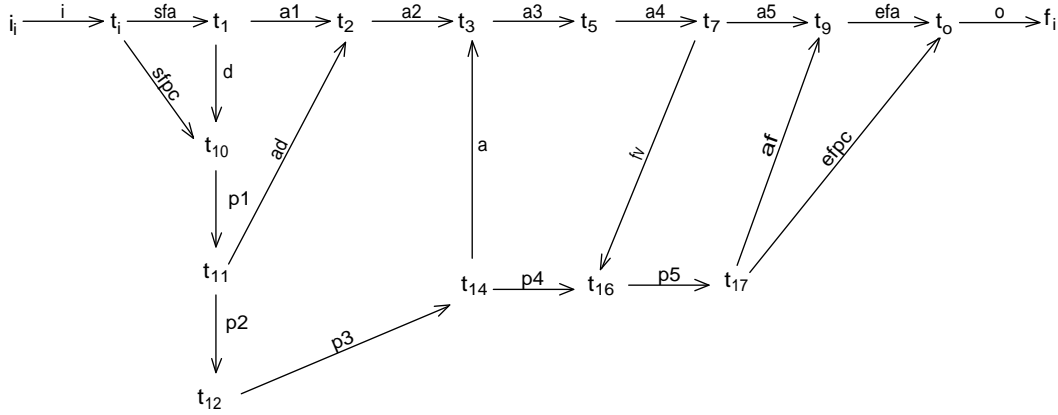


Figura 51 – Grafo de precedência para a árvore de prova rotulada para o cenário $Sc_2A \times Sc_2PC$.

tipo de escolha entre os grafos de precedência, sendo até este momento, os quatro cenários considerados. Depois disso, deve ocorrer o disparo de t_{13} ou t_{14} . Se o disparo de t_{13} ocorre, somente o grafo de precedência da Figura 48 é seguido a partir deste ponto, sendo os demais grafos descartados. Desta forma, as próximas transições a serem disparadas são t_4 e t_o . Se, por outro lado, ocorre o disparo da transição t_{14} , o grafo de precedência da Figura 48 é descartado e os outros três grafos passam a ser considerados. Nestes três cenários, após o disparo de t_{14} ocorre o disparo de t_3 . Até então, os três cenários são considerados. Se, na sequência, ocorre o disparo de t_{15} , há dois cenários a serem considerados: $Sc_3A \times Sc_1PC$ e $Sc_4A \times Sc_1PC$. Se após t_{15} há o disparo de t_5 , o único cenário considerado é $Sc_4A \times Sc_1PC$ e, desta forma, na sequência são disparadas as transições t_8 e t_o . Se após t_{15} há o disparo de t_6 , o cenário a ser considerado é o cenário $Sc_3A \times Sc_1PC$, onde, na sequência, é disparada a transição t_o . Há também a possibilidade de disparar t_5 antes de t_{15} . Neste caso, há dois cenários a serem considerados: $Sc_4A \times Sc_5PC$ e $Sc_2A \times Sc_2PC$, cujos grafos de precedência são apresentados nas Figuras 50 e 51, respectivamente. Neste caso, após o disparo de t_5 , pode ocorrer o disparo de t_7 ou t_{15} . Caso ocorra o disparo de t_{15} , o grafo de precedência da Figura 50 é considerado para ser seguido e o grafo da Figura 51 é descartado. Nesta situação, após o disparo de t_{15} haverá os disparos de t_8 e t_o . Se, por outro lado, ocorre o disparo de t_7 , o grafo de precedência da Figura 51 é seguido e as transições t_{16} , t_{17} , t_9 e t_o disparam na sequência. Desta maneira, somente os cenários livres de *deadlock* serão executados e o processo de *workflow* interorganizacional da Figura 7 nunca será levado à situação de *deadlock*.

Uma outra possibilidade é considerar as informações obtidas nas árvores de prova da Lógica Linear com cálculo de datas. Neste caso, a monitoração da execução dos cenários livres de *deadlock* deve considerar, para cada cenário livre de *deadlock*, as datas de produção dos átomos que correspondem aos lugares de entrada das transições a serem monitoradas. As transições t_i e t_o em um sistema composto não precisam ser monitoradas, pois poderiam ser removidas do modelo sem implicar em alterações semânticas no mesmo.

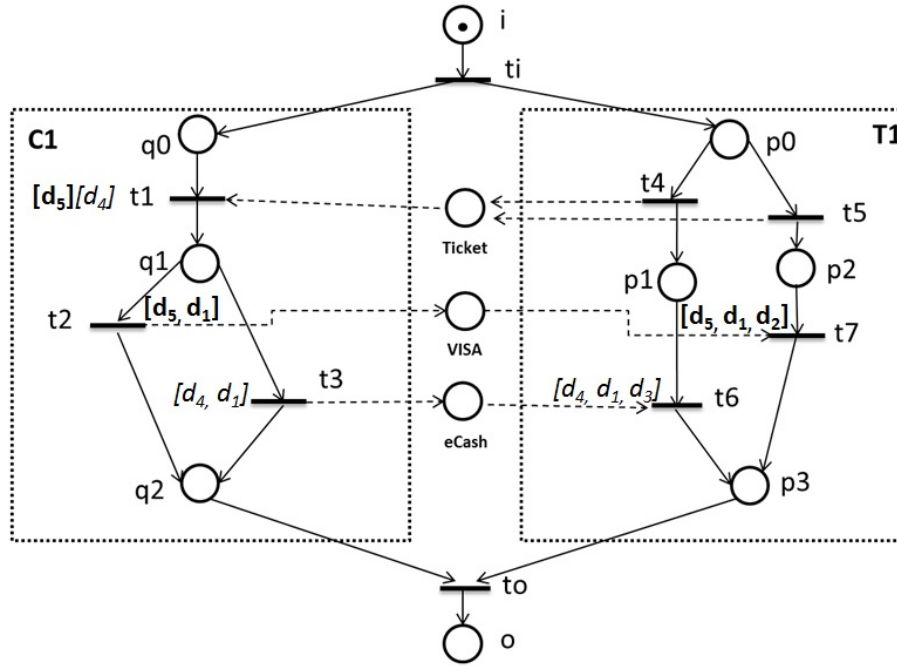


Figura 52 – Sistema composto $C1 \oplus T1$ com guardas associadas às transições.

Desta forma, as datas D_i e d_i não precisam aparecer no modelo a ser monitorado. As datas d_j , onde $j \neq i$ e $j \neq o$, que aparecem nas datas de produção dos átomos que representam os lugares de entrada da transição considerada devem compor a guarda de tal transição. Uma guarda associada a uma transição representa uma condição de disparo da mesma, sendo apresentada por uma expressão ou lista de expressões booleanas entre colchetes.

Para exemplificação, considere o sistema composto $C1 \oplus T1$, apresentado na Figura 22, e as árvores de prova com cálculo de datas para os cenários livres de *deadlock* ($Sc_2C1 \times Sc_1T1$ e $Sc_1C1 \times Sc_2T1$) deste sistema, apresentadas no Apêndice A.5. Considerando o cenário $Sc_2C1 \times Sc_1T1$, o cenário onde o pagamento do bilhete é realizado utilizando *eCash*, e a transição t_4 é disparada, por exemplo, tem-se que data de produção do átomo que representa o lugar de entrada de t_4 , $p0$, é $D_i + d_i$. Como as datas D_i e d_i não devem aparecer no modelo monitorado, nenhuma guarda é associada a esta transição. Já para a transição t_1 , cujos átomos a serem considerados são $q0$ e *ticket*, tem-se as datas de produção $D_i + d_i$ e $D_i + d_i + d_4$, respectivamente. Desconsiderando-se as datas D_i e d_i , tem-se a data d_4 , que deve aparecer na guarda associada à transição t_1 . Considerando a transição t_3 , deve-se verificar a data de produção do átomo $q1$ que, para este cenário, é dada por $D_i + d_i + d_4 + d_1$. Logo, a guarda associada à transição t_3 é $[d_4, d_1]$. Finalmente, para a transição t_6 tem-se as datas $D_i + d_i + d_4$ e $D_i + d_i + d_4 + d_1 + d_3$ de produção dos átomos $p1$ e *eCash*, respectivamente. Logo, a guarda a ser associada à transição t_6 é dada por $[d_4, d_1, d_3]$. Este mesmo procedimento deve ser realizado em relação ao cenário $Sc_1C1 \times Sc_2T1$. A Figura 52 mostra o resultado deste procedimento. É importante observar, na Figura 52, que há duas guardas associadas à transição t_1 , $[d_5]$ e $[d_4]$. A

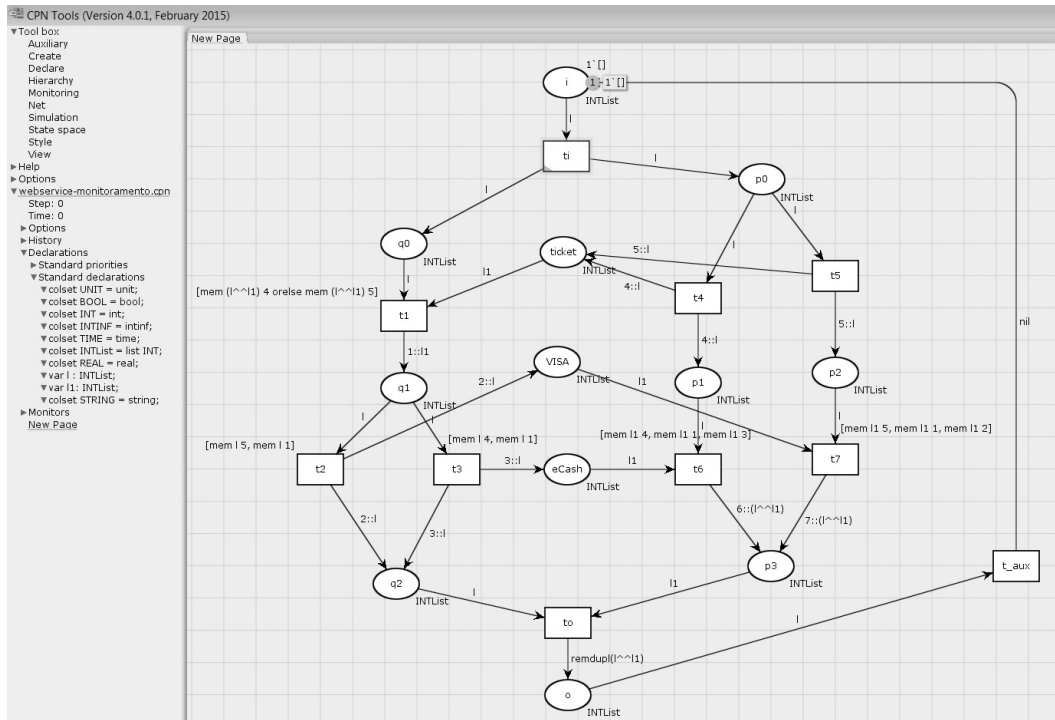


Figura 53 – Sistema composto $\overline{C1 \oplus T1}$ com guardas associadas às transições, implementado no CPN Tools.

primeira corresponde ao cenário $Sc_1C1 \times Sc_2T1$ e a segunda ao cenário $Sc_2C1 \times Sc_1T1$. Ou seja, para ocorrer o disparo de t_1 , ou a transição t_4 ou a transição t_5 deve ter sido disparada anteriormente. Já para a transição t_2 tem-se a guarda $[d_5, d_1]$, indicando que, para t_2 ser disparada, as transições t_5 e t_1 devem ter sido previamente disparadas. Associada à transição t_3 , a guarda $[d_4, d_1]$ indica que, para o disparo de tal transição, as transições t_4 e t_1 devem ser previamente disparadas. Para a transição t_6 , a guarda $[d_4, d_1, d_3]$ indica a necessidade prévia dos disparos de t_4 , t_1 e t_3 . Finalmente, a guarda $[d_5, d_1, d_2]$, associada à transição t_7 , indica a necessidade do disparo prévio das transições t_5 , t_1 e t_2 .

A implementação do modelo apresentado na Figura 52 no CPN Tools é apresentada na Figura 53. Inicialmente, foi criado um tipo `INTList`, que é uma lista de inteiros e duas variáveis deste tipo, l e $l1$, para armazenarem as transições já disparadas. A cada disparo de uma transição t_j onde $j \neq i$ e $j \neq o$, o inteiro j é adicionado à lista de transições já disparadas, através da função $e :: l$, que insere o elemento e na cabeça da lista l . A função $l1 \hat{\sim} l2$ concatena as listas $l1$ e $l2$, que deve ser aplicada quando uma transição tem mais de um lugar de entrada. Já a função $mem\ l\ x$ retorna *true* (verdadeiro) se o elemento x pertence à lista l . Nas guardas, as vírgulas representam conjunções (‘e’ lógico) e o elemento sintático *otherwise* corresponde a uma disjunção (‘ou’ lógico). Finalmente, a função *remdupl* l remove elementos repetidos da lista l , sendo utilizada somente para melhor visualização das transições disparadas em uma dada simulação.

Com a finalidade de verificar se, de fato, a associação de guardas às transições, conforme definido, permite monitorar o sistema composto $C1 \oplus T1$, de forma que somente os

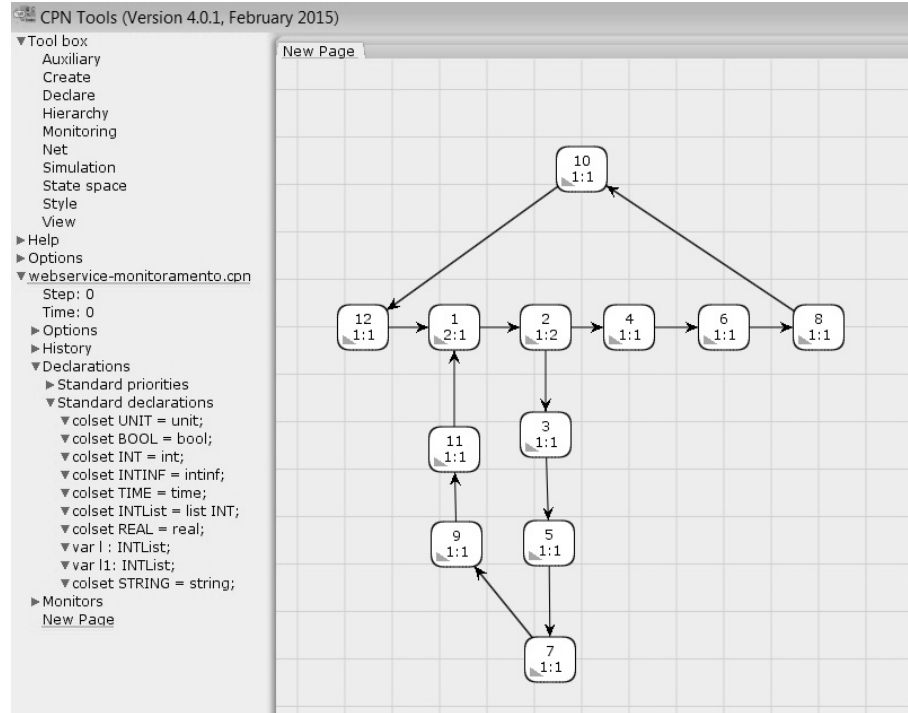


Figura 54 – Grafo das marcações acessíveis para o sistema composto $\overline{C1} \oplus T1$ com guardas associadas às transições, gerado pelo CPN Tools.

cenários livres de *deadlock* são executados em tal sistema composto, o grafo das marcações acessíveis foi gerado, considerando o modelo apresentado na Figura 53. A Figura 54 apresenta o grafo das marcações acessíveis para o sistema composto $\overline{C1} \oplus T1$ com guardas associadas às transições, gerado pelo CPN Tools. A análise deste grafo indica a ausência de nós que representem marcações mortas, ou seja, as situações de *deadlock* são evitadas. Além disso, é possível identificar somente uma componente fortemente conectada, o que mostra, neste caso, que a rede de Petri analisada é viva. Logo, conclui-se que as árvores de prova com cálculo de datas da Lógica Linear fornecem informações que, associadas às condições de disparo das transições, permitem a monitoração de cenários livres de *deadlock*, para o sistema composto $C1 \oplus T1$.

Algumas observações relacionadas ao sistema composto $C1 \oplus T1$ são então realizadas. Conforme mostra Martens (2005), deve-se notar que o módulo $C1$ não conhece o estado interno do módulo $T1$, isto é, $p1$ ou $p2$. Mais especificamente, o módulo $C1$ desconhece a informação sobre qual transição, t_4 ou t_5 , foi inicialmente disparada pelo módulo $T1$. Ainda de acordo com Martens (2005), quando uma decisão interna é tomada e não propriamente comunicada a um serviço Web externo, um problema bem conhecido na literatura ocorre: o problema da escolha não local (BEN-ABDALLAH; LEUE, 1997) e, nestes casos, segundo o referido autor, tais módulos não deveriam ser considerados em composições de serviços Web.

O primeiro ponto a destacar é que, considerando a aplicação do método para detecção de cenários livres de *deadlock* apresentado neste trabalho e a utilização das informações

obtidas nas árvores de prova da Lógica Linear com cálculo de datas para a monitoração dos cenários livres de *deadlock*, o módulo $T1$ pode ser considerado em uma composição de serviços Web, a composição $C1 \oplus T1$, que, conforme demonstrado nesta seção, torna-se livre de bloqueios mortais (*deadlocks*), após a monitoração dos cenários livres de *deadlock*.

Além disso, deve-se observar que a monitoração dos cenários livres de *deadlock* proposta nesta seção comunica propriamente a decisão interna do módulo $T1$ de disparar t_4 ou t_5 ao módulo $C1$. Desta forma, o problema da escolha não local é contornado.

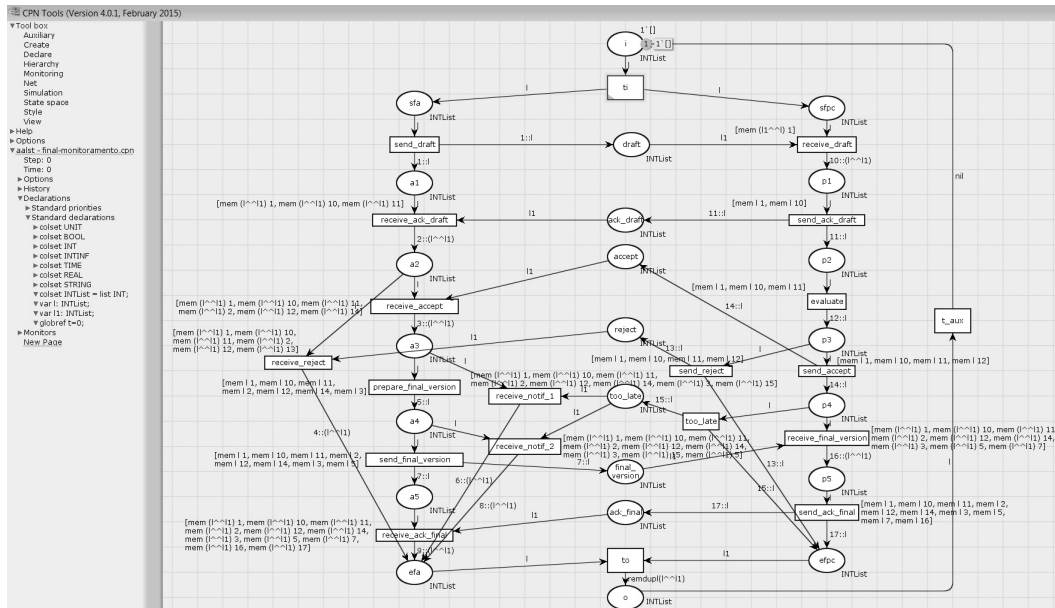


Figura 55 – $U(\text{IOWF-net})$ da Figura 7 com guardas associadas às transições, implementada no CPN Tools.

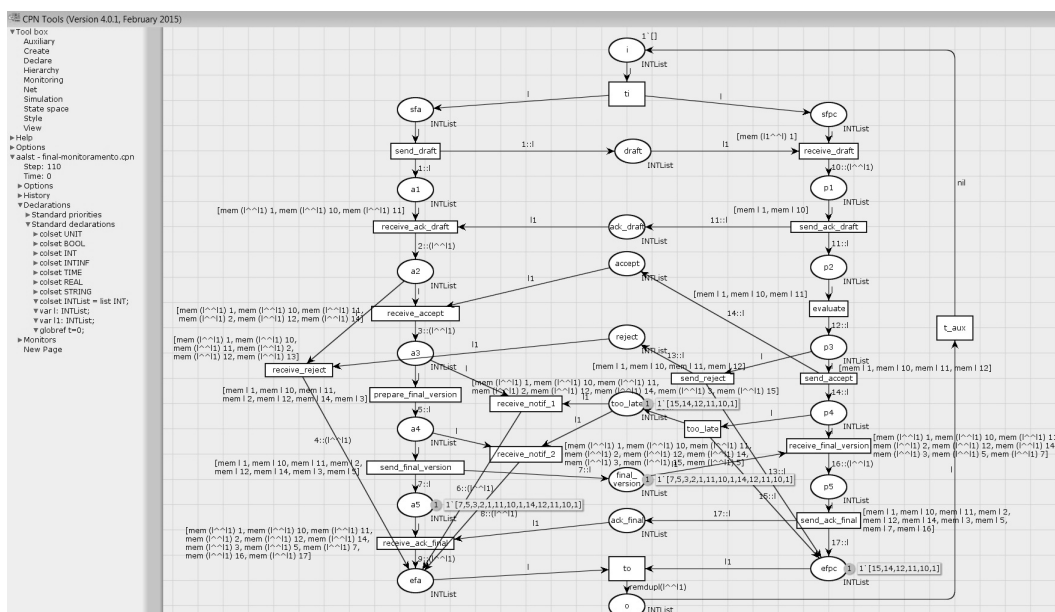


Figura 56 – Situação de *deadlock* para a $U(\text{IOWF-net})$ da Figura 55.

Este mesmo procedimento de monitoração foi implementado no CPN Tools, considerando o processo de *workflow* interorganizacional apresentado na Figura 6, os cenários livres de *deadlock* para tal processo, apresentados na seção 3.4, e as árvores de prova da Lógica Linear com cálculo de datas para tais cenários livres de *deadlock*, apresentadas no Apêndice A.4. O resultado é apresentado na Figura 55. Neste caso, a monitoração realizada não evita a situação de *deadlock*, como mostra a Figura 56. Isso ocorre devido ao fato do *deadlock*, no caso deste processo de *workflow* interorganizacional, não ser decorrente do problema da escolha não local. Trabalhos futuros devem ser desenvolvidos, considerando o controle de situações de *deadlock* não provenientes do problema da escolha não local.

Conclusão

Este capítulo apresenta a conclusão desta pesquisa. Inicialmente, a seção 6.1 apresenta suas principais contribuições. Já a seção 6.2 apresenta os trabalhos futuros que poderão ser desenvolvidos considerando os resultados obtidos nesta pesquisa. Finalmente, a seção 6.3 apresenta as contribuições em produção bibliográfica obtidas no contexto da presente pesquisa.

6.1 Principais Contribuições

Este trabalho formaliza quatro métodos baseados na Lógica Linear para verificação de processos de *workflow* interorganizacionais modelados por WorkFlow net interorganizacional (IOWF-net), que é um tipo de rede de Petri que modela tais processos. Além disso, um estudo de caso é apresentado no contexto da verificação de composições de serviços Web, uma vez que há uma relação estreita entre a modelagem de um processo de *workflow* interorganizacional e uma composição de serviços Web. Finalmente, uma abordagem para planejamento de recursos, baseada no cálculo de datas simbólicas, que considera dados extraídos de árvores de prova da Lógica Linear, é apresentada e validada através de simulações realizadas no simulador CPN Tools. Duas abordagens para a monitoração dos cenários livres de *deadlock* são introduzidas e mostram como dados obtidos nas árvores de prova da Lógica Linear podem ser utilizados para guiar a execução de tais cenários.

O primeiro método baseado na Lógica Linear para verificação de processos de *workflow* interorganizacionais está relacionado com a verificação do critério de correção *Soundness* para processos de *workflow* interorganizacionais. O método é formalizado na seção 3.1 e mostra como a construção e análise de árvores de prova da Lógica Linear permitem a verificação do critério de correção *Soundness* para processos de *workflow* interorganizacionais considerando diretamente a estrutura acíclica de uma IOWF-net. O uso da Lógica Linear permite uma análise parcial do modelo, o que não é possível quando uma abordagem baseada na construção e análise de grafos das marcações acessíveis é utilizada. Além

disso, a avaliação dos resultados do estudo de caso, apresentada na seção 4.3, mostra que a utilização da Lógica Linear para a verificação deste critério de correção permite a visualização das possibilidades de colaborações existentes em um processo de *workflow* interorganizacional, o que não ocorre quando grafos das marcações acessíveis são utilizados para a verificação do critério de correção *Soundness*. Logo, tem-se que a primeira hipótese desta pesquisa foi validada, isto é, *o uso da Lógica Linear, mais especificamente a construção e análise de árvores de prova da Lógica Linear, permite verificar o critério de correção Soundness para processos de workflow interorganizacionais, sem considerar a construção e análise de grafos das marcações acessíveis correspondentes, considerando diretamente a estrutura acíclica da rede utilizada para modelagem do processo.*

O segundo método baseado na Lógica Linear para verificação de processos de *workflow* interorganizacionais é o método para verificação do critério de correção *Relaxed Soundness* para processos de *workflow* interorganizacionais, apresentado na seção 3.2. Este método formaliza como as árvores de prova da Lógica Linear, construídas para a verificação do critério de correção *Soundness* para um dado processo de *workflow* interorganizacional, são reutilizadas, considerando exclusivamente a análise de tais árvores de prova para decidir o critério de correção *Relaxed Soundness* para o dado processo. Além disso, o método apresentado não exige verificação dos processos de *workflow* locais individualmente e também considera a estrutura acíclica de uma IOWF-net. Portanto, conclui-se que a segunda hipótese desta pesquisa também foi validada, ou seja, *o reuso de árvores de prova da Lógica Linear construídas para a verificação do critério de correção Soundness para um processo de workflow interorganizacional e um método de análise para tais árvores de prova reutilizadas, permite a verificação do critério de correção Relaxed Soundness para processos de workflow interorganizacionais não Sound, sem que seja necessária a verificação do critério de correção Relaxed Soundness para os processos de workflow locais de forma singular e considerando diretamente a estrutura acíclica da rede utilizada para modelagem do processo.*

O terceiro método baseado na Lógica Linear para verificação de processos de *workflow* interorganizacionais formalizado nesta pesquisa tem por objetivo validar a terceira hipótese desta pesquisa: *o reuso de árvores de prova da Lógica Linear construídas para a verificação do critério de correção Soundness para um processo de workflow interorganizacional e um método de análise para tais árvores de prova reutilizadas, permite a verificação do critério de correção Weak Soundness para processos de workflow interorganizacionais não Sound, considerando diretamente a estrutura acíclica da rede utilizada para modelagem do processo.* O método para verificação do critério de correção *Weak Soundness* para processos de *workflow* interorganizacionais é apresentado na seção 3.3 e formaliza a análise de árvores de prova da Lógica Linear reutilizadas, que foram construídas para a verificação do critério de correção *Soundness* para um dado processo de *workflow* interorganizacional, de forma que a referida análise permite decidir o critério de

correção *Weak Soundness* para tal processo de *workflow* interorganizacional, considerando diretamente a estrutura acíclica da rede utilizada para modelagem do processo.

Finalmente, o quarto método baseado na Lógica Linear para verificação de processos de *workflow* interorganizacionais formalizado nesta pesquisa é o método para detecção de cenários livres de *deadlock* em processos de *workflow* interorganizacionais, apresentado na seção 3.4. Este método considera, inicialmente, a construção e análise de árvores de prova da Lógica Linear para os cenários das WorkFlow nets locais LWF-nets. Desta análise inicial são obtidos cenários candidatos que devem ser verificados, através da construção e análise de árvores de prova da Lógica Linear, para garantir que, de fato, tais cenários são livres de *deadlock*. O método apresentado identifica todos os cenários livres de *deadlock* em um dado processo de *workflow* interorganizacional, permitindo que as organizações envolvidas no processo possam analisar tais cenários e verificar se os mesmos satisfazem suas necessidades de negócio. Assim, tem-se a validação da quarta hipótese desta pesquisa, isto é, *o uso da Lógica Linear, mais especificamente a construção e análise de árvores de prova da Lógica Linear, permite determinar todos os cenários livres de deadlock em um processo de workflow interorganizacional.*

A quinta hipótese desta pesquisa é validada através do estudo de caso apresentado no Capítulo 4. Inicialmente, é apresentada a relação entre a modelagem de um processo de *workflow* interorganizacional e a modelagem de uma composição de serviços Web. Os métodos formalizados no contexto dos processos de *workflow* interorganizacionais são então aplicados ao contexto das composições de serviços Web. É mostrado como as árvores de prova da Lógica Linear devem ser adaptadas já na prova do critério de correção *Soundness*, para que o reuso das mesmas possa ocorrer tanto nos métodos para verificação dos critérios de correção *Relaxed Soundness* e *Weak Soundness* quanto no método para detecção de cenários livres de *deadlock*. Finalmente, na seção 4.3, é realizada uma avaliação dos resultados obtidos no estudo de caso, mostrando as vantagens da utilização da Lógica Linear, em relação às abordagens clássicas. Inicialmente, o efetivo reuso das árvores de prova da Lógica Linear é apresentado. Uma comparação entre a abordagem tradicional para verificação de *Soundness*, baseada na construção e análise de grafos das marcações acessíveis, e a abordagem apresentada por esta pesquisa é apresentada, destacando o fato dos sequentes mostrarem de maneira explícita as possibilidades de colaborações existentes entre dois serviços Web, diferentemente do que ocorre nas abordagens tradicionais. Finalmente, uma comparação que considera o número de árvores de prova da Lógica Linear construídas para a verificação de *Soundness* e o número de árvores construídas para o método de detecção de cenários livres de *deadlock* é apresentada, mostrando que o número de árvores a serem construídas pode ser significativamente reduzido no método para detecção de cenários livres de *deadlock* proposto nesta pesquisa.

Finalmente, as abordagens para planejamento de recursos e monitoração de cenários livres de *deadlock* utilizando árvores de prova da Lógica Linear, apresentadas no Capí-

tulo 5, validam a sexta hipótese desta pesquisa: *dados derivados das árvores da Lógica Linear permitem realizar um planejamento de recursos para o tratamento de tarefas, de forma que sejam determinados intervalos de datas simbólicas nas quais os recursos devem permanecer disponíveis. Além disso, tais dados derivados das árvores de prova podem ser utilizados em estratégias de monitoração dos cenários livres de deadlock, de forma que o sistema global evite as situações de bloqueio.* Inicialmente, é apresentado como dados obtidos nas árvores de prova da Lógica Linear com cálculo de datas devem ser utilizados para o planejamento de recursos para o tratamento de tarefas em processos de *workflow* interorganizacionais. Intervalos de datas simbólicas são definidos para a execução de tarefas do tipo usuário. Além disso, pode-se obter informações sobre o término ao mais cedo e ao mais tarde do processo analisado. Simulações utilizando a ferramenta de simulação CPN Tools foram realizadas para validar os dados simbólicos obtidos pelo método apresentado. Finalmente, foi mostrado como dados obtidos em árvores de prova da Lógica Linear podem ser utilizados para a monitoração dos cenários livres de *deadlock*. A primeira maneira apresentada é seguir grafos de precedência construídos utilizando dados obtidos em árvores de prova da Lógica Linear rotuladas. Já a segunda abordagem considera a associação de guardas que condicionam os disparos das transições e possuem informações obtidas nas árvores de prova da Lógica Linear com cálculo de datas. Implementações no CPN Tools mostram que este método é válido quando as situações de *deadlock* são decorrentes do problema da escolha não local.

6.2 Trabalhos Futuros

Como trabalho futuro, deve-se investigar como as regras de fusão de transições, apresentadas por Cortabarría (2005), podem ser utilizadas no contexto da monitoração da execução dos cenários livres de *deadlock*, obtidos a partir do método apresentado na presente pesquisa, nos casos em que as situações de *deadlock* não são provenientes do problema da escolha não local.

Além disso, pode-se considerar no processo de monitoração a existência de estruturas de sifões que esvaziam em caso de *deadlock* e, através da utilização das regras de fusão de transições apresentadas em Cortabarría (2005), verificar a possibilidade de monitorar o esvaziamento de tais sifões através da fusão das transições responsáveis por seu esvaziamento. O objetivo então é definir um método sistemático para saber quais transições devem ser fusionadas para que as possíveis situações de *deadlock* sejam evitadas.

Um outro trabalho é utilização da Lógica Linear e da IOWF-net para representar uma proposta de arquitetura do tipo SOA, de forma que os cenários obtidos no modelo de análise de requisitos sejam reproduzidos no modelo de arquitetura, possibilitando a validação e a rastreabilidade entre o que está especificado no modelo de requisitos e o que está reproduzido no modelo de arquitetura.

Finalmente, deve-se aprofundar os estudos no que se diz respeito ao problema da escolha não local, que não é um problema exclusivo do contexto dos processos de *workflow* ou serviços Web, verificando como a Lógica Linear pode auxiliar na solução do referido problema em um contexto mais geral.

6.3 Contribuições em Produção Bibliográfica

Na sequência são apresentadas produções bibliográficas resultantes desta pesquisa.

O artigo (PASSOS; JULIA, 2013), intitulado *Qualitative Analysis of Interorganizational WorkFlow Nets Using Linear Logic: Soundness Verification*, foi publicado na vigésima quinta edição da *IEEE International Conference on Tools with Artificial Intelligence* (ICTAI) em 2013 e apresenta o método para Verificação do Critério de Correção *Soundness* para Processos de *Workflow* Interorganizacionais, formalizado na seção 3.1.

Já o artigo (PASSOS; JULIA, 2014), intitulado *Linear Logic as a Tool for Deadlock-Freeness Scenarios Detection in Interorganizational Workflow Processes*, foi publicado na vigésima sexta edição da *IEEE International Conference on Tools with Artificial Intelligence* (ICTAI) em 2014 e refere-se ao método para Detecção de Cenários Livres de *Deadlock* em Processos de *Workflow* Interorganizacionais, formalizado na seção 3.4.

O artigo (PASSOS; JULIA, 2015b), intitulado *Relaxed Soundness Verification for Interorganizational Workflow Processes*, foi publicado na décima sétima edição da *International Conference on Enterprise Information Systems* (ICEIS) em 2015 e descreve o método para Verificação do Critério de Correção *Relaxed Soundness* para Processos de *Workflow* Interorganizacionais, formalizado na seção 3.2.

Já o artigo (PASSOS; JULIA, 2015), intitulado *Deadlock-Freeness Scenarios Detection in Web Service Composition*, publicado na décima segunda edição da *International Conference on Information Technology - New Generations* (ITNG) em 2015, trata do método para detecção de cenários livres de *deadlock* em serviços Web contemplando, assim, parte do estudo de caso apresentado no Capítulo 4.

O artigo (PASSOS; JULIA, 2015a), intitulado *Linear Logic as a Tool for Qualitative and Quantitative Analysis of Workflow Processes*, contempla os cálculos de datas simbólicas utilizadas para o planejamento de recursos, apresentado na seção 5.1. Este artigo foi aceito para publicação no *International Journal on Artificial Intelligence Tools* e apresenta a formalização da *WorkFlow* net utilizando Lógica Linear, com uma extensão do tipo experimental (usando o simulador *CPN Tools*), a fim de validar através de simulações os resultados analíticos apresentados em (PASSOS, 2009).

Finalmente, o artigo (PASSOS et al., 2013), intitulado *Verification of Workflow Specifications in UML Using Automated Transformations to WF-nets*, publicado na nona edição do Simpósio Brasileiro de Sistemas de Informação (SBSI) em 2013, trata de uma atividade transversal relacionada aos assuntos abordados nesta tese, com o objetivo de

tornar as abordagens apresentadas mais intuitivas e factíveis num ambiente industrial. Este trabalho foi desenvolvido em parceria com outro pesquisador do grupo de Engenharia de Software da FACOM/UFU, o Prof. Dr. Marcelo de Almeida Maia.

Referências

- AALST, W. M. P. van der. Verification of workflow nets. In: **ICATPN '97: Proceedings of the 18th International Conference on Application and Theory of Petri Nets**. London, UK: Springer-Verlag, 1997. p. 407–426. ISBN 3-540-63139-9.
- _____. The application of Petri nets to Workflow Management. In: **The Journal of Circuits, Systems and Computers**. [S.l.: s.n.], 1998. p. 21–66.
- AALST, W. M. P. Van der. Modeling and analyzing interorganizational workflows. In: **Application of Concurrency to System Design, 1998. Proceedings., 1998 International Conference on**. [S.l.: s.n.], 1998. p. 262–272.
- AALST, W. M. P. van der. Woflan: a Petri-net-based workflow analyzer. **Syst. Anal. Model. Simul.**, Gordon and Breach Science Publishers, Inc., Newark, NJ, USA, v. 35, n. 3, p. 345–357, 1999. ISSN 0232-9298.
- AALST, W. M. P. van der et al. Introduction. In: HOFSTEDE, A. H. M. ter et al. (Ed.). **Modern Business Process Automation**. [S.l.]: Springer, 2010. p. 3–19. ISBN 978-3-642-03120-5.
- _____. Soundness of workflow nets: classification, decidability, and analysis. **Form. Asp. Comput.**, Springer-Verlag, London, UK, UK, v. 23, p. 333–363, May 2011. ISSN 0934-5043. Disponível em: <<http://dx.doi.org/10.1007/s00165-010-0161-4>>.
- AALST, W. M. P. van der; HEE, K. van. **Workflow Management: Models, Methods, and Systems**. [S.l.]: The MIT Press, 2004.
- AALST, W. M. P. van der; HOFSTEDE, A. H. M. ter. Yawl: yet another workflow language. **Inf. Syst.**, Elsevier Science Ltd., Oxford, UK, UK, v. 30, n. 4, p. 245–275, jun. 2005. ISSN 0306-4379. Disponível em: <<http://dx.doi.org/10.1016/j.is.2004.02.002>>.
- AALST, W. van der. Structural characterizations of sound workflow nets. **Computing Science Reports/23**, Eindhoven University of Technology, n. 96, 1996.
- _____. Three good reasons for using a petri-net-based workflow management system. In: **Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC'96)**. [S.l.: s.n.], 1996. p. 179–201.
- _____. Loosely coupled interorganizational workflows: modeling and analyzing workflows crossing organizational boundaries. **Inf. Manage.**, Elsevier Science Publishers B. V.,

Amsterdam, The Netherlands, The Netherlands, v. 37, n. 2, p. 67–75, mar. 2000. ISSN 0378-7206.

AALST, W. van der; STAHL, C. **Modeling Business Processes: A Petri Net-Oriented Approach**. [S.l.]: The MIT Press, 2011. ISBN 0262015382, 9780262015387.

BEN-ABDALLAH, H.; LEUE, S. Syntactic detection of process divergence and non-local choice in message sequence charts. In: BRINKSMA, E. (Ed.). **Tools and Algorithms for the Construction and Analysis of Systems**. [S.l.]: Springer Berlin Heidelberg, 1997, (Lecture Notes in Computer Science, v. 1217). p. 259–274. ISBN 978-3-540-62790-6.

CAPTARENCU, O. Modelling and verification of interorganizational workflows with security constraints: A petri nets-based approach. In: BAJEC, M.; EDER, J. (Ed.). **Advanced Information Systems Engineering Workshops**. [S.l.]: Springer Berlin Heidelberg, 2012, (Lecture Notes in Business Information Processing, v. 112). p. 486–493. ISBN 978-3-642-31068-3.

CHAMPAGNAT, R.; PRADIN-CHÉZALVIEL, B.; VALETTE, R. Petri nets and Linear Logic as an aid for scheduling batch processes. In: **ADPM 2000, Automation of mixed processes: Hybrid Dynamic Systems**. Dortmund Germany: [s.n.], 2000. p. 107–112.

CORTABARRÍA, J. C. M. **CONTROL DE SISTEMAS DE EVENTOS DISCRETOS MODELADOS CON REDES DE PETRI**. Tese (Doutorado) — Universidad de Zaragoza, 2005.

DEHNERT, J. **A Methodology for Workflow Modeling: From business process modeling towards sound workflow specification**. Tese (Doutorado) — Technischen Universität Berlin, 2003.

DEHNERT, J.; RITTGEN, P. Relaxed soundness of business processes. In: DITTRICH, K.; GEPPERT, A.; NORRIE, M. (Ed.). **Advanced Information Systems Engineering**. Springer Berlin Heidelberg, 2001, (Lecture Notes in Computer Science, v. 2068). p. 157–170. ISBN 978-3-540-42215-0. Disponível em: <http://dx.doi.org/10.1007/3-540-45341-5_11>.

DIAZ, M. (Ed.). **Petri Nets: Fundamental Models, Verification and Applications**. [S.l.]: Wiley-IEEE Press, 2010.

FAHLAND, D. et al. Analysis on demand: Instantaneous soundness checking of industrial business process models. **Data Knowl. Eng.**, v. 70, n. 5, p. 448–466, 2011.

GIRARD, J.-Y. Linear logic. **Theoretical Computer Science**, Elsevier Science Publishers Ltd., Essex, UK, v. 50, n. 1, p. 1–102, 1987. ISSN 0304-3975.

_____. Linear logic: its syntax and semantics. In: **Proceedings of the workshop on Advances in Linear Logic**. New York, NY, USA: Cambridge University Press, 1995. p. 1–42. ISBN 0-521-55961-8.

GIRAULT, F.; PRADIN-CHÉZALVIEL, B.; VALETTE, R. A logic for Petri nets. **Journal Européen des Systèmes Automatisés**, v. 31, n. 3, p. 525–542, 1997.

GOCHET, P.; GRIBOMONT, P. **Logique: méthodes pour l'informatique fondamentale**. [S.l.]: Hermès, 1990.

- JENSEN, K. Coloured petri nets and the invariant-method. **Theoretical Computer Science**, v. 14, n. 3, p. 317 – 336, 1981. ISSN 0304-3975. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0304397581900499>>.
- KLAI, K.; OCHI, H.; TATA, S. Formal Abstraction and Compatibility Checking of Web Services. In: **Web Services (ICWS), 2013 IEEE 20th International Conference on**. [S.l.: s.n.], 2013. p. 163–170.
- LI, Z.; ZHOU, M. Clarifications on the definitions of elementary siphons in petri nets. **IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans**, v. 36, n. 6, p. 1227–1229, Nov 2006. ISSN 1083-4427.
- LIM, H. W.; KERSCHBAUM, F.; WANG, H. Workflow signatures for business process compliance. **IEEE Trans. Dependable Secur. Comput.**, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 9, n. 5, p. 756–769, set. 2012. ISSN 1545-5971. Disponível em: <<http://dx.doi.org/10.1109/TDSC.2012.38>>.
- LOMAZOVA, I. A. Interacting workflow nets for workflow process re-engineering. **Fundam. Inf.**, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 101, n. 1-2, p. 59–70, jan. 2010. ISSN 0169-2968. Disponível em: <<http://dl.acm.org/citation.cfm?id=1883453.1883459>>.
- MARTENS, A. Usability of web services. In: **Web Information Systems Engineering Workshops, 2003. Proceedings. Fourth International Conference on**. [S.l.: s.n.], 2003. p. 182–190.
- _____. Analyzing web service based business processes. In: CERIOLI, M. (Ed.). **Fundamental Approaches to Software Engineering**. Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3442). p. 19–33. ISBN 978-3-540-25420-1. Disponível em: <http://dx.doi.org/10.1007/978-3-540-31984-9_3>.
- MERLIN, P. M. **A study of the recoverability of computing systems**. Tese (Doutorado) — University of California, Irvine, 1974.
- MURATA, T. Petri nets: Properties, analysis and applications. **Proceedings of the IEEE**, v. 77, n. 4, p. 541–580, 1989.
- OMG. **OMG Business Process Model And Notation (BPMN) – Version 2.0**. [S.l.], 2011. Disponível em: <<http://www.omg.org/spec/BPMN/2.0/>>. Acesso em: 12.1.2015.
- _____. **OMG Unified Modeling Language Specification – Version 2.4.1**. [S.l.], 2011. Disponível em: <<http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF>>. Acesso em: 12.1.2015.
- PASSOS, L. M. S. **Formalização de WorkFlow nets utilizando Lógica Linear: análise qualitativa e quantitativa**. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, Maio 2009.
- PASSOS, L. M. S.; JULIA, S. Qualitative analysis of workflow nets using linear logic: soundness verification. In: **Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics**. Piscataway, NJ, USA: IEEE Press, 2009. (SMC'09), p. 2843–2847. ISBN 978-1-4244-2793-2.

_____. Qualitative Analysis of Interorganizational Workflow Nets Using Linear Logic: Soundness Verification. In: **Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on**. [S.l.: s.n.], 2013. p. 667–673. ISSN 1082-3409.

PASSOS, L. M. S.; JULIA, S. Linear Logic as a Tool for Deadlock-Freeness Scenarios Detection in Interorganizational Workflow Processes. In: **Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on**. [S.l.: s.n.], 2014. p. 316–320. ISSN 1082-3409.

_____. Linear Logic as a Tool for Qualitative and Quantitative Analysis of Workflow Processes. Aceito para publicação no "International Journal on Artificial Intelligence Tools". 2015.

_____. Relaxed soundness verification for interorganizational workflow processes. In: **ICEIS 2015 - Proceedings of the 17th International Conference on Enterprise Information Systems, Volume 3, Barcelona, Spain, 27-30 April, 2015**. [S.l.: s.n.], 2015. p. 221–228.

PASSOS, L. M. S. et al. Verification of Workflow Specifications in UML Using Automated Transformations to WF-nets. In: **Anais do IX Simpósio Brasileiro de Sistemas de Informação (SBSI 2013)**. Sociedade Brasileira de Computação (SBC), 2013. p. 674–685. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2013/0061.pdf>>. Acesso em: 29.1.2015.

PASSOS, L. S.; JULIA, S. Deadlock-freeness scenarios detection in web service composition. In: **Information Technology - New Generations (ITNG), 2015 12th International Conference on**. [S.l.: s.n.], 2015. p. 780–783.

PETRI, C. A. **Kommunikation mit Automaten**. Tese (Doutorado) — Institut für instrumentelle Mathematik, Bonn, Germany, 1962.

PRADIN-CHÉZALVIEL, B.; VALETTE, R.; KUNZLE, L. A. Scenario durations characterization of t-timed Petri nets using linear logic. In: **PNPM '99: Proceedings of the The 8th International Workshop on Petri Nets and Performance Models**. Washington, DC, USA: IEEE Computer Society, 1999. p. 208. ISBN 0-7695-0331-4.

PUHLMANN, F. Soundness verification of business processes specified in the pi-calculus. In: MEERSMAN, R.; TARI, Z. (Ed.). **On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS**. Springer Berlin Heidelberg, 2007, (Lecture Notes in Computer Science, v. 4803). p. 6–23. ISBN 978-3-540-76846-3. Disponível em: <http://dx.doi.org/10.1007/978-3-540-76848-7_3>.

RATZER, A. et al. CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets. In: AALST, W. van der; BEST, E. (Ed.). **Applications and Theory of Petri Nets 2003**. [S.l.]: Springer Berlin Heidelberg, 2003, (Lecture Notes in Computer Science, v. 2679). p. 450–462. ISBN 978-3-540-40334-0.

RIVIERE, N.; PRADIN-CHÉZALVIEL, B.; VALETTE, R. Reachability and temporal conflicts in t-time Petri nets. In: **Petri Nets and Performance Models, 2001. Proceedings. 9th International Workshop on**. [S.l.: s.n.], 2001. p. 229–238.

- SIEGERIS, J.; ZIMMERMANN, A. Workflow model compositions preserving relaxed soundness. In: **Proceedings of the 4th International Conference on Business Process Management**. Berlin, Heidelberg: Springer-Verlag, 2006. (BPM'06), p. 177–192. ISBN 3-540-38901-6, 978-3-540-38901-9.
- SUN, H. C.; DU, Y. Y. Soundness Analysis of Inter-Organizational Workflows. **Information Technology Journal**, v. 7, p. 1194–1199, 2008.
- VALETTE, R. Analysis of Petri Nets by Stepwise Refinements. **Journal of Computer and System Sciences**, v. 18, p. 35–46, 1979.
- VALETTE, R.; CARDOSO, J. **Redes de Petri**. [S.l.]: Editora DAUSFC, 1997.
- XIONG, P.; FAN, Y.; ZHOU, M. A Petri Net Approach to Analysis and Composition of Web Services. **Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on**, v. 40, n. 2, p. 376–387, March 2010. ISSN 1083-4427.
- YAMAGUCHI, S. et al. WF-Net Based Modeling and Soundness Verification of Interworkflows. **IEICE Trans. Fundam. Electron. Commun. Comput. Sci.**, Oxford University Press, Oxford, UK, E90-A, n. 4, p. 829–835, abr. 2007. ISSN 0916-8508.

Apêndices

Árvores de Prova da Lógica Linear

Neste apêndice são apresentadas as árvores de prova da Lógica Linear construídas para exemplificação dos métodos de verificação formalizados nesta pesquisa.

O Apêndice A.1 apresenta as árvores de prova da Lógica Linear construídas para exemplificação dos métodos para verificação dos critérios de correção *Soundness*, *Relaxed Soundness* e *Weak Soundness*. Já o Apêndice A.2 apresenta as árvores de prova construídas no contexto da exemplificação do método para detecção de cenários livres de *deadlock*. O Apêndice A.3 apresenta as árvores de prova da Lógica Linear construídas para o estudo de caso. Já o Apêndice A.4 apresenta as árvores de prova da Lógica Linear com cálculo de datas para o planejamento de recursos. Finalmente, o Apêndice A.5 apresenta as árvores de prova da Lógica Linear com cálculo de datas para a monitoração de cenários livres de *deadlock*.

A.1 Árvores de Prova da Lógica Linear para Aplicação dos Métodos para Verificação dos Critérios de Correção *Soundness*, *Relaxed Soundness* e *Weak Soundness*

Este apêndice apresenta as árvores de prova da Lógica Linear construídas no contexto dos métodos para verificação dos critérios de correção *Soundness*, *Relaxed Soundness* e *Weak Soundness*, apresentados, respectivamente, nas seções 3.1, 3.2 e 3.3.

É importante recordar que as transições da $U(\text{IOWF-net})$ são representadas por:

$$\begin{aligned} t_i &= i \multimap \text{start_flow_author} \otimes \text{start_flow_PC}, \\ t_1 &= \text{send_draft} = \text{start_flow_author} \multimap a1 \otimes \text{draft}, \\ t_2 &= \text{receive_ack_draft} = a1 \otimes \text{ack_draft} \multimap a2, \\ t_3 &= \text{receive_accept} = a2 \otimes \text{accept} \multimap a3, \\ t_4 &= \text{receive_reject} = a2 \otimes \text{reject} \multimap \text{end_flow_author}, \end{aligned}$$

$t_5 = \text{prepare_final_version} = a3 \multimap a4,$
 $t_6 = \text{receive_notification_1} = a3 \otimes \text{too_late} \multimap \text{end_flow_author},$
 $t_7 = \text{send_final_version} = a4 \multimap a5 \otimes \text{final_version},$
 $t_8 = \text{receive_notification_2} = a4 \otimes \text{too_late} \multimap \text{end_flow_author},$
 $t_9 = \text{receive_ack_final} = a5 \otimes \text{ack_final} \multimap \text{end_flow_author},$
 $t_{10} = \text{receive_draft} = \text{start_flow_PC} \otimes \text{draft} \multimap p1,$
 $t_{11} = \text{send_ack_draft} = p1 \multimap \text{ack_draft} \otimes p2,$
 $t_{12} = \text{evaluate} = p2 \multimap p3,$
 $t_{13} = \text{send_reject} = p3 \multimap \text{reject} \otimes \text{end_flow_PC},$
 $t_{14} = \text{send_accept} = p3 \multimap \text{accept} \otimes p4,$
 $t_{15} = \text{too_late} = p4 \multimap \text{too_late} \otimes \text{end_flow_PC},$
 $t_{16} = \text{receive_final_version} = \text{final_version} \otimes p4 \multimap p5,$
 $t_{17} = \text{send_ack_final} = p5 \multimap \text{ack_final} \otimes \text{end_flow_PC},$
 $t_o = \text{end_flow_author} \otimes \text{end_flow_PC} \multimap o.$

Além disso, os átomos nas árvores de prova poderão ser representados apenas pelas letras iniciais de seus nomes completos. Por exemplo, *too_late* e *final_version* poderão ser representados por *tl* e *fv*, respectivamente. Além disso, um conjunto de transições t_i, t_j, t_k and t_l pode ser representado por $t_{i,j,k,l}$. Estas reduções podem ocorrer devido ao tamanho e legibilidade das árvores de prova apresentadas.

A árvore de prova para o cenário Sc_1A é mostrada na sequência.

$$\begin{array}{c}
 \frac{a2 \vdash a2 \quad efa \vdash efa}{\multimap_L} \\
 \frac{a1 \vdash a1 \quad a2, a2 \multimap efa \vdash efa}{\multimap_L} \\
 \frac{sfa \vdash sfa \quad a1, a1 \multimap a2, a2 \multimap efa \vdash efa}{\multimap_L} \\
 sfa, sfa \multimap a1, a1 \multimap a2, a2 \multimap efa \vdash efa
 \end{array}$$

A árvore de prova para o cenário Sc_2A é dada por:

$$\begin{array}{c}
 \frac{a5 \vdash a5 \quad efa \vdash efa}{\multimap_L} \\
 \frac{a4 \vdash a4 \quad a5, a5 \multimap efa \vdash efa}{\multimap_L} \\
 \frac{a3 \vdash a3 \quad a4, a4 \multimap a5, a5 \multimap efa \vdash efa}{\multimap_L} \\
 \frac{a2 \vdash a2 \quad a3, a3 \multimap a4, a4 \multimap a5, a5 \multimap efa \vdash efa}{\multimap_L} \\
 \frac{a1 \vdash a1 \quad a2, a2 \multimap a3, a3 \multimap a4, a4 \multimap a5, a5 \multimap efa \vdash efa}{\multimap_L} \\
 \frac{sfa \vdash sfa \quad a1, a1 \multimap a2, a2 \multimap a3, a3 \multimap a4, a4 \multimap a5, a5 \multimap efa \vdash efa}{\multimap_L} \\
 sfa, sfa \multimap a1, a1 \multimap a2, a2 \multimap a3, a3 \multimap a4, a4 \multimap a5, a5 \multimap efa \vdash efa
 \end{array}$$

A árvore de prova para o cenário Sc_3A é a seguinte:

$$\begin{array}{c}
 \frac{a3 \vdash a3 \quad efa \vdash efa}{\multimap_L} \\
 \frac{a2 \vdash a2 \quad a3, a3 \multimap efa \vdash efa}{\multimap_L} \\
 \frac{a1 \vdash a1 \quad a2, a2 \multimap a3, a3 \multimap efa \vdash efa}{\multimap_L} \\
 \frac{sfa \vdash sfa \quad a1, a1 \multimap a2, a2 \multimap a3, a3 \multimap efa \vdash efa}{\multimap_L} \\
 sfa, sfa \multimap a1, a1 \multimap a2, a2 \multimap a3, a3 \multimap efa \vdash efa
 \end{array}$$

A árvore de prova para o cenário Sc_4A é dada por:

$$\begin{array}{c}
 \frac{a4 \vdash a4 \quad efa \vdash efa}{\multimap_L} \\
 \frac{a3 \vdash a3 \quad a4, a4 \multimap efa \vdash efa}{\multimap_L} \\
 \frac{a2 \vdash a2 \quad a3, a3 \multimap a4, a4 \multimap efa \vdash efa}{\multimap_L} \\
 \frac{a1 \vdash a1 \quad a2, a2 \multimap a3, a3 \multimap a4, a4 \multimap efa \vdash efa}{\multimap_L} \\
 \frac{sfa \vdash sfa \quad a1, a1 \multimap a2, a2 \multimap a3, a3 \multimap a4, a4 \multimap efa \vdash efa}{\multimap_L} \\
 sfa, sfa \multimap a1, a1 \multimap a2, a2 \multimap a3, a3 \multimap a4, a4 \multimap efa \vdash efa
 \end{array}$$

A árvore de prova para o cenário Sc_1PC é apresentada na sequência.

$$\begin{array}{c}
 \frac{p4 \vdash p4 \quad efpct \vdash efpct}{\multimap_L} \\
 \frac{p3 \vdash p3 \quad p4, p4 \multimap efpct \vdash efpct}{\multimap_L} \\
 \frac{p2 \vdash p2 \quad p3, p3 \multimap p4, p4 \multimap efpct \vdash efpct}{\multimap_L} \\
 \frac{p1 \vdash p1 \quad p2, p2 \multimap p3, p3 \multimap p4, p4 \multimap efpct \vdash efpct}{\multimap_L} \\
 \frac{sfpct \vdash sfpc \quad p1, p1 \multimap p2, p2 \multimap p3, p3 \multimap p4, p4 \multimap efpct \vdash efpct}{\multimap_L} \\
 \frac{sfpct \vdash sfpc \quad p1, p1 \multimap p2, p2 \multimap p3, p3 \multimap p4, p4 \multimap efpct \vdash efpct}{\multimap_L} \\
 sfpc, sfpc \multimap p1, p1 \multimap p2, p2 \multimap p3, p3 \multimap p4, p4 \multimap efpct \vdash efpct
 \end{array}$$

A árvore de prova para o cenário Sc_2PC é dada por:

$$\begin{array}{c}
\frac{p5 \vdash p5 \quad efpct \vdash efp}{\multimap_L} \\
\frac{p4 \vdash p4 \quad p5, p5 \multimap efpct \vdash efp}{\multimap_L} \\
\frac{p3 \vdash p3 \quad p4, p4 \multimap p5, p5 \multimap efpct \vdash efp}{\multimap_L} \\
\frac{p2 \vdash p2 \quad p3, p3 \multimap p4, p4 \multimap p5, p5 \multimap efpct \vdash efp}{\multimap_L} \\
\frac{p1 \vdash p1 \quad p2, p2 \multimap p3, p3 \multimap p4, p4 \multimap p5, p5 \multimap efpct \vdash efp}{\multimap_L} \\
\frac{sfpct \vdash sfpc \quad p1, p1 \multimap p2, p2 \multimap p3, p3 \multimap p4, p4 \multimap p5, p5 \multimap efpct \vdash efp}{\multimap_L} \\
sfpc, sfpc \multimap p1, p1 \multimap p2, p2 \multimap p3, p3 \multimap p4, p4 \multimap p5, p5 \multimap efpct \vdash efp
\end{array}$$

A árvore de prova para o cenário Sc_3PC é a seguinte:

$$\begin{array}{c}
\frac{p3 \vdash p3 \quad efpct \vdash efp}{\multimap_L} \\
\frac{p2 \vdash p2 \quad p3, p3 \multimap efpct \vdash efp}{\multimap_L} \\
\frac{p1 \vdash p1 \quad p2, p2 \multimap p3, p3 \multimap efpct \vdash efp}{\multimap_L} \\
\frac{sfpct \vdash sfpc \quad p1, p1 \multimap p2, p2 \multimap p3, p3 \multimap efpct \vdash efp}{\multimap_L} \\
sfpc, sfpc \multimap p1, p1 \multimap p2, p2 \multimap p3, p3 \multimap efpct \vdash efp
\end{array}$$

A árvore de prova para o cenário Sc_1 é como se segue:

$$\begin{array}{c}
\frac{\frac{efa \vdash efa \quad efpct \vdash efp}{efa, efpct \vdash efa \otimes efp} \otimes_R \quad o \vdash o}{\multimap_L} \\
\frac{\frac{a2 \vdash a2 \quad r \vdash r}{a2, r \vdash a2 \otimes r} \otimes_R \quad efp, efa, efa \otimes efp \multimap o \vdash o}{\multimap_L} \\
\frac{\frac{a1 \vdash a1 \quad ad \vdash ad}{a1, ad \vdash a1 \otimes ad} \otimes_R \quad r, efp, a2, a2 \otimes r \multimap efa, t_o \vdash o}{\multimap_L} \\
\frac{a1, ad, r, efp, a1 \otimes ad \multimap a2, t_4, o \vdash o}{\otimes_L} \\
\frac{p3 \vdash p3 \quad a1, ad, r \otimes efp, t_2, 4, o \vdash o}{\multimap_L} \\
\frac{p2 \vdash p2 \quad a1, ad, p3, p3 \multimap r \otimes efp, t_2, 4, o \vdash o}{\multimap_L} \\
\frac{a1, ad, p2, p2 \multimap p3, t_2, 4, 13, o \vdash o}{\otimes_L} \\
\frac{p1 \vdash p1 \quad a1, ad \otimes p2, t_2, 4, 12, 13, o \vdash o}{\multimap_L} \\
\frac{\frac{sfpct \vdash sfpc \quad d \vdash d}{sfpct, d \vdash sfpc \otimes d} \otimes_R \quad a1, p1, p1 \multimap ad \otimes p2, t_2, 4, 12, 13, o \vdash o}{\multimap_L} \\
\frac{sfpct, a1, d, sfpc \otimes d \multimap p1, t_2, 4, 11, 12, 13, o \vdash o}{\otimes_L} \\
\frac{sfa \vdash sfa \quad sfpc, a1 \otimes d, t_2, 4, 10, 11, 12, 13, o \vdash o}{\multimap_L} \\
\frac{sfa, sfpc, sfa \multimap a1 \otimes d, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o}{\otimes_L} \\
\frac{i \vdash i \quad sfa \otimes sfpc, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o}{\multimap_L} \\
i, t_i, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o
\end{array}$$

A árvore de prova para o cenário Sc_2 é dada por:

$$\begin{array}{c}
 \frac{\frac{efa \vdash efa \quad efpc \vdash efpc}{efa, efpc \vdash efa \otimes efpc} \otimes_R \quad o \vdash o}{\vdash o} \multimap_L \\
 \frac{\frac{\frac{a3 \vdash a3 \quad tl \vdash tl}{a3, tl \vdash a3 \otimes tl} \otimes_R \quad efpc, efa, efa \otimes efpc \multimap o \vdash o}{a3, tl, efpc, a3 \otimes tl \multimap efa, t_o \vdash o} \otimes_L \\
 \frac{p4 \vdash p4 \quad a3, tl \otimes efpc, t_6, t_o \vdash o}{\vdash o} \multimap_L \\
 \frac{\frac{\frac{a2 \vdash a2 \quad a \vdash a}{a2, a \vdash a2 \otimes a} \otimes_R \quad p4, a3, t_6, p4 \multimap tl \otimes efpc, t_o \vdash o}{a2, a, p4, a2 \otimes a \multimap a3, t_6, t_{15}, t_o \vdash o} \otimes_L \\
 \frac{p3 \vdash p3 \quad a2, a \otimes p4, t_3, t_6, t_{15}, t_o \vdash o}{\vdash o} \multimap_L \\
 \frac{p2 \vdash p2 \quad a2, p3, t_3, t_6, p3 \multimap a \otimes p4, t_{15}, t_o \vdash o}{\vdash o} \multimap_L \\
 \frac{\frac{\frac{a1 \vdash a1 \quad ad \vdash ad}{a1, ad \vdash a1 \otimes ad} \otimes_R \quad p2, a2, t_3, t_6, p2 \multimap p3, t_{14}, t_{15}, t_o \vdash o}{a1, ad, p2, a1 \otimes ad \multimap a2, t_3, t_6, t_{12}, t_{14}, t_{15}, t_o \vdash o} \otimes_L \\
 \frac{p1 \vdash p1 \quad a1, ad \otimes p2, t_2, t_3, t_6, t_{12}, t_{14}, t_{15}, t_o \vdash o}{\vdash o} \multimap_L \\
 \frac{\frac{\frac{sfpc \vdash sfpc \quad d \vdash d}{sfpc, d \vdash sfpc \otimes d} \otimes_R \quad a1, p1, t_2, 3, 6, p1 \multimap ad \otimes p2, t_{12}, t_{14}, t_{15}, t_o \vdash o}{sfpc, a1, d, t_2, t_3, t_6, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o} \otimes_L \\
 \frac{sfa \vdash sfa \quad sfpc, a1 \otimes d, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o}{\vdash o} \multimap_L \\
 \frac{sfa, sfpc, sfa \multimap a1 \otimes d, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o}{\vdash o} \otimes_L \\
 \frac{i \vdash i \quad sfa \otimes sfpc, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o}{\vdash o} \multimap_L \\
 i, t_i, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o
 \end{array}$$

A árvore de prova para o cenário Sc_3 é a mostrada na sequência.

$$\begin{array}{c}
\frac{\frac{efa \vdash efa \quad efp \vdash efp}{efa, efp \vdash efa \otimes efp} \otimes_R \quad o \vdash o}{\vdash o} \multimap_L \\
\frac{\frac{a4 \vdash a4 \quad tl \vdash tl}{a4, tl \vdash a4 \otimes tl} \otimes_R \quad efp, efa, efa \otimes efp \multimap o \vdash o}{\vdash o} \multimap_L \\
\frac{a3 \vdash a3 \quad tl, efp, a4, a4 \otimes tl \multimap efa, to \vdash o}{\vdash o} \multimap_L \\
\frac{a3, tl, efp, a3 \multimap a4, t8, to \vdash o}{\vdash o} \otimes_L \\
\frac{p4 \vdash p4 \quad a3, tl \otimes efp, t5, t8, to \vdash o}{\vdash o} \multimap_L \\
\frac{\frac{a2 \vdash a2 \quad a \vdash a}{a2, a \vdash a2 \otimes a} \otimes_R \quad p4, a3, t5, t8, p4 \multimap tl \otimes efp, to \vdash o}{\vdash o} \multimap_L \\
\frac{a2, a, p4, a2 \otimes a \multimap a3, t5, t8, t15, to \vdash o}{\vdash o} \otimes_L \\
\frac{p3 \vdash p3 \quad a2, a \otimes p4, t3, t5, t8, t15, to \vdash o}{\vdash o} \multimap_L \\
\frac{p2 \vdash p2 \quad a2, p3, t3, t5, t8, p3 \multimap a \otimes p4, t15, to \vdash o}{\vdash o} \multimap_L \\
\frac{\frac{a1 \vdash a1 \quad ad \vdash ad}{a1, ad \vdash a1 \otimes ad} \otimes_R \quad p2, a2, t3, t5, t8, p2 \multimap p3, t14, t15, to \vdash o}{\vdash o} \multimap_L \\
\frac{a1, ad, p2, a1 \otimes ad \multimap a2, t3, t5, t8, t12, t14, t15, to \vdash o}{\vdash o} \otimes_L \\
\frac{p1 \vdash p1 \quad a1, ad \otimes p2, t2, t3, t5, t8, t12, t14, t15, to \vdash o}{\vdash o} \multimap_L \\
\frac{\frac{sfpc \vdash sfpc \quad d \vdash d}{sfpc, d \vdash sfpc \otimes d} \otimes_R \quad a1, p1, t2, 3, 5, 8, p1 \multimap ad \otimes p2, t12, t14, t15, to \vdash o}{\vdash o} \multimap_L \\
\frac{sfpc, a1, d, t2, t3, t5, t8, sfpc \otimes d \multimap p1, t11, t12, t14, t15, to \vdash o}{\vdash o} \otimes_L \\
\frac{sfa \vdash sfa \quad sfpc, a1 \otimes d, t2, t3, t5, t8, t10, t11, t12, t14, t15, to \vdash o}{\vdash o} \multimap_L \\
\frac{sfa, sfpc, sfa \multimap a1 \otimes d, t2, t3, t5, t8, t10, t11, t12, t14, t15, to \vdash o}{\vdash o} \otimes_L \\
\frac{i \vdash i \quad sfa \otimes sfpc, t1, t2, t3, t5, t8, t10, t11, t12, t14, t15, to \vdash o}{\vdash o} \multimap_L \\
i, t_i, t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, to \vdash o
\end{array}$$

A árvore de prova para o cenário Sc_4 é como se segue:

$$\begin{array}{c}
 \frac{\frac{efa \vdash efa \quad efp \vdash efp}{efa, efp \vdash efa \otimes efp} \otimes_R \quad o \vdash o}{o \vdash o} \multimap_L \\
 \frac{\frac{a5 \vdash a5 \quad af \vdash af}{a5, af \vdash a5 \otimes af} \otimes_R \quad efp, efa, efa \otimes efp \multimap o \vdash o}{a5, af, efp, a5 \otimes af \multimap efa, t_o \vdash o} \multimap_L \\
 \frac{p5 \vdash p5 \quad a5, af \otimes efp, t_9, t_o \vdash o}{p5 \vdash p5 \quad a5, af \otimes efp, t_9, t_o \vdash o} \multimap_L \\
 \frac{\frac{fv \vdash fv \quad p4 \vdash p4}{fv, p4 \vdash fv \otimes p4} \otimes_R \quad a5, p5, p5 \multimap af \otimes efp, t_9, t_o \vdash o}{p4, a5, fv, fv \otimes p4 \multimap p5, t_9, t_{17}, o \vdash o} \otimes_L \\
 \frac{a4 \vdash a4 \quad p4, a5 \otimes fv, t_9, t_{16}, t_{17}, o \vdash o}{a4 \vdash a4 \quad p4, a5 \otimes fv, t_9, t_{16}, t_{17}, o \vdash o} \multimap_L \\
 \frac{a3 \vdash a3 \quad p4, a4, a4 \multimap a5 \otimes fv, t_9, t_{16}, t_{17}, o \vdash o}{a3 \vdash a3 \quad p4, a4, a4 \multimap a5 \otimes fv, t_9, t_{16}, t_{17}, o \vdash o} \multimap_L \\
 \frac{\frac{a2 \vdash a2 \quad a \vdash a}{a2, a \vdash a2 \otimes a} \otimes_R \quad p4, a3, a3 \multimap a4, t_7, t_9, t_{16}, t_{17}, o \vdash o}{a2, a, p4, a2 \otimes a \multimap a3, t_5, t_7, t_9, t_{16}, t_{17}, o \vdash o} \otimes_L \\
 \frac{p3 \vdash p3 \quad a2, a \otimes p4, t_3, t_5, t_7, t_9, t_{16}, t_{17}, o \vdash o}{p3 \vdash p3 \quad a2, a \otimes p4, t_3, t_5, t_7, t_9, t_{16}, t_{17}, o \vdash o} \multimap_L \\
 \frac{p2 \vdash p2 \quad a2, p3, t_3, t_5, t_7, t_9, p3 \multimap a \otimes p4, t_{16}, t_{17}, o \vdash o}{p2 \vdash p2 \quad a2, p3, t_3, t_5, t_7, t_9, p3 \multimap a \otimes p4, t_{16}, t_{17}, o \vdash o} \multimap_L \\
 \frac{\frac{a1 \vdash a1 \quad ad \vdash ad}{a1, ad \vdash a1 \otimes ad} \otimes_R \quad p2, a2, t_3, t_5, t_7, t_9, p2 \multimap p3, t_{14}, t_{16}, t_{17}, o \vdash o}{a1, ad, p2, a1 \otimes ad \multimap a2, t_3, t_5, t_7, t_9, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o} \otimes_L \\
 \frac{p1 \vdash p1 \quad a1, ad \otimes p2, t_2, t_3, t_5, t_7, t_9, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o}{p1 \vdash p1 \quad a1, ad \otimes p2, t_2, t_3, t_5, t_7, t_9, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o} \multimap_L \\
 \frac{\frac{sfpc \vdash sfpc \quad d \vdash d}{sfpc, d \vdash sfpc \otimes d} \otimes_R \quad a1, p1, t_2, 3, 5, 7, 9, p1 \multimap ad \otimes p2, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o}{sfpc, a1, d, t_2, t_3, t_5, t_7, t_9, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o} \otimes_L \\
 \frac{sfa \vdash sfa \quad sfpc, a1 \otimes d, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o}{sfa \vdash sfa \quad sfpc, a1 \otimes d, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o} \multimap_L \\
 \frac{sfa, sfpc, sfa \multimap a1 \otimes d, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o}{sfa, sfpc, sfa \multimap a1 \otimes d, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o} \otimes_L \\
 \frac{i \vdash i \quad sfa \otimes sfpc, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o}{i \vdash i \quad sfa \otimes sfpc, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o} \multimap_L \\
 i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o
 \end{array}$$

E, finalmente, a árvore de prova para o cenário Sc_5 é dada por:

$$\begin{array}{c}
\frac{}{tl,efpc,a5,fv,t9,t_o \vdash o} \otimes_L \\
\frac{a4 \vdash a4 \quad tl,efpc,a5 \otimes fv,t9,t_o \vdash o}{\vdash} \multimap_L \\
\frac{a3 \vdash a3 \quad tl,efpc,a4,a4 \multimap a5 \otimes fv,t9,t_o \vdash o}{\vdash} \multimap_L \\
\frac{}{a3,tl,efpc,a3 \multimap a4,t7,t9,t_o \vdash o} \otimes_L \\
\frac{p4 \vdash p4 \quad a3,tl \otimes efpc,t5,t7,t9,t_o \vdash o}{\vdash} \multimap_L \\
\frac{\frac{a2 \vdash a2}{a2,a \vdash a2 \otimes a} \otimes_R \quad \frac{a \vdash a}{p4,a3,t5,t7,t9,p4 \multimap tl \otimes efpc,t_o \vdash o} \multimap_L}{\vdash} \otimes_L \\
\frac{}{a2,a,p4,a2 \otimes a \multimap a3,t5,t7,t9,t_{15},t_o \vdash o} \otimes_L \\
\frac{p3 \vdash p3 \quad a2,a \otimes p4,t3,t5,t7,t9,t_{15},t_o \vdash o}{\vdash} \multimap_L \\
\frac{p2 \vdash p2 \quad a2,p3,t3,t5,t7,t9,p3 \multimap a \otimes p4,t_{15},t_o \vdash o}{\vdash} \multimap_L \\
\frac{\frac{a1 \vdash a1}{a1,a \vdash a1 \otimes ad} \otimes_R \quad \frac{ad \vdash ad}{p2,a2,t3,t5,t7,t9,p2 \multimap p3,t_{14},t_{15},t_o \vdash o} \multimap_L}{\vdash} \otimes_L \\
\frac{}{a1,ad,p2,a1 \otimes ad \multimap a2,t3,t5,t7,t9,t_{12},t_{14},t_{15},t_o \vdash o} \otimes_L \\
\frac{p1 \vdash p1 \quad a1,ad \otimes p2,t2,t3,t5,t7,t9,t_{12},t_{14},t_{15},t_o \vdash o}{\vdash} \multimap_L \\
\frac{\frac{sfp \vdash sfp}{sfp,d \vdash sfp \otimes d} \otimes_R \quad \frac{d \vdash d}{a1,p1,t2,3,5,7,9,p1 \multimap ad \otimes p2,t_{12},t_{14},t_{15},t_o \vdash o} \multimap_L}{\vdash} \otimes_L \\
\frac{}{sfp,a1,d,t2,t3,t5,t7,t9,sfp \otimes d \multimap p1,t_{11},t_{12},t_{14},t_{15},t_o \vdash o} \otimes_L \\
\frac{sfa \vdash sfa \quad sfp,a1 \otimes d,t2,t3,t5,t7,t9,t_{10},t_{11},t_{12},t_{14},t_{15},t_o \vdash o}{\vdash} \multimap_L \\
\frac{}{sfa,sfp,sfa \multimap a1 \otimes d,t2,t3,t5,t7,t9,t_{10},t_{11},t_{12},t_{14},t_{15},t_o \vdash o} \otimes_L \\
\frac{i \vdash i \quad sfa \otimes sfp,t1,t2,t3,t5,t7,t9,t_{10},t_{11},t_{12},t_{14},t_{15},t_o \vdash o}{\vdash} \multimap_L \\
i,t_i,t_1,t_2,t_3,t_5,t_7,t_9,t_{10},t_{11},t_{12},t_{14},t_{15},t_o \vdash o
\end{array}$$

A.2 Árvores de Prova da Lógica Linear para Aplicação do Método para Detecção de Cenários Livres de *Deadlock*

Este apêndice apresenta as árvores de prova da Lógica Linear construídas no contexto do método para detecção de cenários livres de *deadlock*, apresentado na seção 3.4.

É importante recordar que as transições da $U(\text{IOWF-net})$ são representadas por:

$$\begin{aligned}
t_i &= i \multimap start_flow_author \otimes start_flow_PC, \\
t_1 &= send_draft = start_flow_author \multimap a1 \otimes draft, \\
t_2 &= receive_ack_draft = a1 \otimes ack_draft \multimap a2, \\
t_3 &= receive_accept = a2 \otimes accept \multimap a3, \\
t_4 &= receive_reject = a2 \otimes reject \multimap end_flow_author, \\
t_5 &= prepare_final_version = a3 \multimap a4,
\end{aligned}$$

$$\begin{aligned}
t_6 &= \text{receive_notification_1} = a3 \otimes \text{too_late} \multimap \text{end_flow_author}, \\
t_7 &= \text{send_final_version} = a4 \multimap a5 \otimes \text{final_version}, \\
t_8 &= \text{receive_notification_2} = a4 \otimes \text{too_late} \multimap \text{end_flow_author}, \\
t_9 &= \text{receive_ack_final} = a5 \otimes \text{ack_final} \multimap \text{end_flow_author}, \\
t_{10} &= \text{receive_draft} = \text{start_flow_PC} \otimes \text{draft} \multimap p1, \\
t_{11} &= \text{send_ack_draft} = p1 \multimap \text{ack_draft} \otimes p2, \\
t_{12} &= \text{evaluate} = p2 \multimap p3, \\
t_{13} &= \text{send_reject} = p3 \multimap \text{reject} \otimes \text{end_flow_PC}, \\
t_{14} &= \text{send_accept} = p3 \multimap \text{accept} \otimes p4, \\
t_{15} &= \text{too_late} = p4 \multimap \text{too_late} \otimes \text{end_flow_PC}, \\
t_{16} &= \text{receive_final_version} = \text{final_version} \otimes p4 \multimap p5, \\
t_{17} &= \text{send_ack_final} = p5 \multimap \text{ack_final} \otimes \text{end_flow_PC}, \\
t_o &= \text{end_flow_author} \otimes \text{end_flow_PC} \multimap o.
\end{aligned}$$

Além disso, os átomos nas árvores de prova poderão ser representados apenas pelas letras iniciais de seus nomes completos. Por exemplo, *too_late* e *final_version* poderão ser representados por *tl* e *fv*, respectivamente. Além disso, um conjunto de transições t_i, t_j, t_k and t_l pode ser representado por $t_{i,j,k,l}$. Estas reduções podem ocorrer devido ao tamanho e legibilidade das árvores de prova apresentadas.

A árvore de prova para o cenário Sc_1A é dada por:

$$\begin{array}{c}
\frac{\frac{a2 \vdash a2 \quad r \vdash r}{a2, r \vdash a2 \otimes r} \otimes_R \quad d, efa \vdash efa}{\quad} \multimap_L \\
\frac{\frac{\frac{a1 \vdash a1 \quad ad \vdash ad}{a1, ad \vdash a1 \otimes ad} \otimes_R \quad r, d, a2, a2 \otimes r \multimap efa \vdash efa}{\quad} \multimap_L \\
\frac{\quad}{ad, r, a1, d, a1 \otimes ad \multimap a2, t_4 \vdash efa} \otimes_L \\
\frac{sfa \vdash sfa \quad ad, r, a1 \otimes d, t_2, t_4 \vdash efa}{\quad} \multimap_L \\
ad, r, sfa, sfa \multimap a1 \otimes d, t_2, t_4 \vdash efa
\end{array}$$

A árvore de prova para o cenário Sc_2A é como se segue:

$$\begin{array}{c}
\frac{\frac{af \vdash af}{af, a5 \vdash af \otimes a5} \quad \frac{a5 \vdash a5}{d, fv, efa \vdash efa} \otimes_R \quad \frac{}{af, d, a5, fv, af \otimes a5 \multimap efa \vdash efa} \multimap_L \\
\frac{}{a4 \vdash a4} \quad \frac{}{af, d, a5 \otimes fv, t_9 \vdash efa} \multimap_L \\
\frac{}{a3 \vdash a3} \quad \frac{}{af, d, a4, a4 \multimap a5 \otimes fv, t_9 \vdash efa} \multimap_L \\
\frac{\frac{a2 \vdash a2}{a2, a \vdash a2 \otimes a} \quad \frac{a \vdash a}{af, d, a3, a3 \multimap a4, t_7, 9 \vdash efa} \otimes_R \quad \frac{}{a, af, d, a2, a2 \otimes a \multimap a3, t_5, 7, 9 \vdash efa} \multimap_L \\
\frac{\frac{a1 \vdash a1}{a1, ad \vdash a1 \otimes ad} \quad \frac{ad \vdash ad}{a, af, d, a2, a2 \otimes a \multimap a3, t_5, 7, 9 \vdash efa} \otimes_R \quad \frac{}{ad, a, af, a1, d, a1 \otimes ad \multimap a2, t_3, 5, 7, 9 \vdash efa} \otimes_L \\
\frac{}{sfa \vdash sfa} \quad \frac{}{ad, a, af, a1 \otimes d, t_2, 3, 5, 7, 9 \vdash efa} \multimap_L \\
ad, a, af, sfa, sfa \multimap a1 \otimes d, t_2, t_3, t_5, t_7, t_9 \vdash efa
\end{array}$$

A árvore de prova para o cenário Sc_3A é mostrada na sequência.

$$\begin{array}{c}
\frac{\frac{a3 \vdash a3}{a3, tl \vdash a3 \otimes tl} \quad \frac{tl \vdash tl}{d, efa \vdash efa} \otimes_R \quad \frac{}{tl, d, a3, a3 \otimes tl \multimap efa \vdash efa} \multimap_L \\
\frac{\frac{a2 \vdash a2}{a2, a \vdash a2 \otimes a} \quad \frac{a \vdash a}{tl, d, a3, a3 \otimes tl \multimap efa \vdash efa} \otimes_R \quad \frac{}{a, tl, d, a2, a2 \otimes a \multimap a3, t_6 \vdash efa} \multimap_L \\
\frac{\frac{a1 \vdash a1}{a1, ad \vdash a1 \otimes ad} \quad \frac{ad \vdash ad}{a, tl, d, a2, a2 \otimes a \multimap a3, t_6 \vdash efa} \otimes_R \quad \frac{}{ad, a, tl, a1, d, a1 \otimes ad \multimap a2, t_3, 6 \vdash efa} \otimes_L \\
\frac{}{sfa \vdash sfa} \quad \frac{}{ad, a, tl, a1 \otimes d, t_2, 3, 6 \vdash efa} \multimap_L \\
ad, a, tl, sfa, sfa \multimap a1 \otimes d, t_2, t_3, t_6 \vdash efa
\end{array}$$

A árvore de prova para o cenário Sc_4A é dada por:

$$\begin{array}{c}
\frac{\frac{a4 \vdash a4}{a4, tl \vdash a4 \otimes tl} \quad \frac{tl \vdash tl}{d, efa \vdash efa} \otimes_R \quad \frac{}{tl, d, a4, a4 \otimes tl \multimap efa \vdash efa} \multimap_L \\
\frac{}{a3 \vdash a3} \quad \frac{}{tl, d, a4, a4 \otimes tl \multimap efa \vdash efa} \multimap_L \\
\frac{\frac{a2 \vdash a2}{a2, a \vdash a2 \otimes a} \quad \frac{a \vdash a}{tl, d, a3, a3 \multimap a4, t_8 \vdash efa} \otimes_R \quad \frac{}{a, tl, d, a2, a2 \otimes a \multimap a3, t_5, 8 \vdash efa} \multimap_L \\
\frac{\frac{a1 \vdash a1}{a1, ad \vdash a1 \otimes ad} \quad \frac{ad \vdash ad}{a, tl, d, a2, a2 \otimes a \multimap a3, t_5, 8 \vdash efa} \otimes_R \quad \frac{}{ad, a, tl, a1, d, a1 \otimes ad \multimap a2, t_3, 5, 8 \vdash efa} \otimes_L \\
\frac{}{sfa \vdash sfa} \quad \frac{}{ad, a, tl, a1 \otimes d, t_2, 3, 5, 8 \vdash efa} \multimap_L \\
ad, a, tl, sfa, sfa \multimap a1 \otimes d, t_2, t_3, t_5, t_8 \vdash efa
\end{array}$$

A árvore de prova para o cenário Sc_1PC é mostrada na sequência.

$$\begin{array}{c}
 \frac{}{ad, a, tl, efp \vdash efp} \otimes_L \\
 \frac{p4 \vdash p4 \quad ad, a, tl \otimes efp \vdash efp}{\multimap_L} \\
 \frac{}{ad, a, p4, p4 \multimap tl \otimes efp \vdash efp} \otimes_L \\
 \frac{p3 \vdash p3 \quad ad, a \otimes p4, t_{15} \vdash efp}{\multimap_L} \\
 \frac{p2 \vdash p2 \quad ad, p3, p3 \multimap a \otimes p4, t_{15} \vdash efp}{\multimap_L} \\
 \frac{}{ad, p2, p2 \multimap p3, t_{14}, t_{15} \vdash efp} \otimes_L \\
 \frac{p1 \vdash p1 \quad ad \otimes p2, t_{12}, t_{14}, t_{15} \vdash efp}{\multimap_L} \\
 \frac{\frac{sfpc \vdash sfpc \quad d \vdash d}{sfpc, d \vdash sfpc \otimes d} \otimes_R \quad p1, p1 \multimap ad \otimes p2, t_{12}, t_{14}, t_{15} \vdash efp}{\multimap_L} \\
 d, sfpc, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{14}, t_{15} \vdash efp
 \end{array}$$

A árvore de prova para o cenário Sc_2PC é como se segue:

$$\begin{array}{c}
 \frac{}{ad, a, af, efp \vdash efp} \otimes_L \\
 \frac{p5 \vdash p5 \quad ad, a, af \otimes efp \vdash efp}{\multimap_L} \\
 \frac{\frac{fv \vdash fv \quad p4 \vdash p4}{fv, p4 \vdash fv \otimes p4} \otimes_R \quad ad, a, p5, p5 \multimap af \otimes efp \vdash efp}{\multimap_L} \\
 \frac{}{fv, ad, a, p4, fv \otimes p4 \multimap p5, t_{17} \vdash efp} \otimes_L \\
 \frac{p3 \vdash p3 \quad fv, ad, a \otimes p4, t_{16}, t_{17} \vdash efp}{\multimap_L} \\
 \frac{p2 \vdash p2 \quad fv, ad, p3, p3 \multimap a \otimes p4, t_{16}, t_{17} \vdash efp}{\multimap_L} \\
 \frac{}{fv, ad, p2, p2 \multimap p3, t_{14}, t_{16}, t_{17} \vdash efp} \otimes_L \\
 \frac{p1 \vdash p1 \quad fv, ad \otimes p2, t_{12}, t_{14}, t_{16}, t_{17} \vdash efp}{\multimap_L} \\
 \frac{\frac{sfpc \vdash sfpc \quad d \vdash d}{sfpc, d \vdash sfpc \otimes d} \otimes_R \quad fv, p1, p1 \multimap ad \otimes p2, t_{12}, t_{14}, t_{16}, t_{17} \vdash efp}{\multimap_L} \\
 d, fv, sfpc, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{14}, t_{16}, t_{17} \vdash efp
 \end{array}$$

Finalmente, a árvore de prova para o cenário Sc_3PC é dada por:

$$\begin{array}{c}
 \frac{}{ad, r, efp \vdash efp} \otimes_L \\
 \frac{p3 \vdash p3 \quad ad, r \otimes efp \vdash efp}{\multimap_L} \\
 \frac{p2 \vdash p2 \quad ad, p3, p3 \multimap r \otimes efp \vdash efp}{\multimap_L} \\
 \frac{}{ad, p2, p2 \multimap p3, t_{13} \vdash efp} \otimes_L \\
 \frac{p1 \vdash p1 \quad ad \otimes p2, t_{12}, t_{13} \vdash efp}{\multimap_L} \\
 \frac{\frac{sfpc \vdash sfpc \quad d \vdash d}{sfpc, d \vdash sfpc \otimes d} \otimes_R \quad p1, p1 \multimap ad \otimes p2, t_{12}, t_{13} \vdash efp}{\multimap_L} \\
 d, sfpc, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{13} \vdash efp
 \end{array}$$

As árvores de prova para os cenários candidatos, $Sc_3A \times Sc_1PC$, $Sc_4A \times Sc_1PC$, $Sc_2A \times Sc_2PC$ e $Sc_1A \times Sc_3PC$, são apresentadas na sequência.

A árvore de prova para o cenário $Sc_3A \times Sc_1PC$ é dada por:

$$\begin{array}{c}
\frac{\frac{efa \vdash efa \quad efpcl \vdash efpcl}{efa, efpcl \vdash efa \otimes efpcl} \otimes_R \quad o \vdash o}{\vdash o} \multimap_L \\
\frac{\frac{\frac{a3 \vdash a3 \quad tl \vdash tl}{a3, tl \vdash a3 \otimes tl} \otimes_R \quad efpcl, efa, efa \otimes efpcl \multimap o \vdash o}{a3, tl, efpcl, a3 \otimes tl \multimap efa, to \vdash o} \otimes_L}{\vdash o} \multimap_L \\
\frac{p4 \vdash p4 \quad a3, tl \otimes efpcl, t_6, to \vdash o}{\vdash o} \multimap_L \\
\frac{\frac{\frac{a2 \vdash a2 \quad a \vdash a}{a2, a \vdash a2 \otimes a} \otimes_R \quad p4, a3, t_6, p4 \multimap tl \otimes efpcl, to \vdash o}{a2, a, p4, a2 \otimes a \multimap a3, t_6, t_{15}, to \vdash o} \otimes_L}{\vdash o} \multimap_L \\
\frac{p3 \vdash p3 \quad a2, a \otimes p4, t_3, t_6, t_{15}, to \vdash o}{\vdash o} \multimap_L \\
\frac{p2 \vdash p2 \quad a2, p3, t_3, t_6, p3 \multimap a \otimes p4, t_{15}, to \vdash o}{\vdash o} \multimap_L \\
\frac{\frac{\frac{a1 \vdash a1 \quad ad \vdash ad}{a1, ad \vdash a1 \otimes ad} \otimes_R \quad p2, a2, t_3, t_6, p2 \multimap p3, t_{14}, t_{15}, to \vdash o}{a1, ad, p2, a1 \otimes ad \multimap a2, t_3, t_6, t_{12}, t_{14}, t_{15}, to \vdash o} \otimes_L}{\vdash o} \multimap_L \\
\frac{p1 \vdash p1 \quad a1, ad \otimes p2, t_2, t_3, t_6, t_{12}, t_{14}, t_{15}, to \vdash o}{\vdash o} \multimap_L \\
\frac{\frac{\frac{sfpc \vdash sfpc \quad d \vdash d}{sfpc, d \vdash sfpc \otimes d} \otimes_R \quad a1, p1, t_2, 3, 6, p1 \multimap ad \otimes p2, t_{12}, t_{14}, t_{15}, to \vdash o}{sfpc, a1, d, t_2, t_3, t_6, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{14}, t_{15}, to \vdash o} \otimes_L}{\vdash o} \multimap_L \\
\frac{sfa \vdash sfa \quad sfpc, a1 \otimes d, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, to \vdash o}{\vdash o} \multimap_L \\
\frac{sfa, sfpc, sfa \multimap a1 \otimes d, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, to \vdash o}{\vdash o} \otimes_L \\
\frac{i \vdash i \quad sfa \otimes sfpc, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, to \vdash o}{\vdash o} \multimap_L \\
i, t_i, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, to \vdash o
\end{array}$$

A árvore de prova para o cenário $Sc_4A \times Sc_1PC$ é como se segue:

$$\begin{array}{c}
 \frac{\frac{efa \vdash efa \quad efpc \vdash efpc}{efa, efpc \vdash efa \otimes efpc} \otimes_R \quad o \vdash o}{\quad} \multimap_L \\
 \frac{\frac{a4 \vdash a4 \quad tl \vdash tl}{a4, tl \vdash a4 \otimes tl} \otimes_R \quad efpc, efa, efa \otimes efpc \multimap o \vdash o}{\quad} \multimap_L \\
 \frac{a3 \vdash a3 \quad tl, efpc, a4, a4 \otimes tl \multimap efa, to \vdash o}{\quad} \multimap_L \\
 \frac{\quad}{a3, tl, efpc, a3 \multimap a4, t8, to \vdash o} \otimes_L \\
 \frac{p4 \vdash p4 \quad a3, tl \otimes efpc, t5, t8, to \vdash o}{\quad} \multimap_L \\
 \frac{\frac{a2 \vdash a2 \quad a \vdash a}{a2, a \vdash a2 \otimes a} \otimes_R \quad p4, a3, t5, t8, p4 \multimap tl \otimes efpc, to \vdash o}{\quad} \multimap_L \\
 \frac{\quad}{a2, a, p4, a2 \otimes a \multimap a3, t5, t8, t15, to \vdash o} \otimes_L \\
 \frac{p3 \vdash p3 \quad a2, a \otimes p4, t3, t5, t8, t15, to \vdash o}{\quad} \multimap_L \\
 \frac{p2 \vdash p2 \quad a2, p3, t3, t5, t8, p3 \multimap a \otimes p4, t15, to \vdash o}{\quad} \multimap_L \\
 \frac{\frac{a1 \vdash a1 \quad ad \vdash ad}{a1, ad \vdash a1 \otimes ad} \otimes_R \quad p2, a2, t3, t5, t8, p2 \multimap p3, t14, t15, to \vdash o}{\quad} \multimap_L \\
 \frac{\quad}{a1, ad, p2, a1 \otimes ad \multimap a2, t3, t5, t8, t12, t14, t15, to \vdash o} \otimes_L \\
 \frac{p1 \vdash p1 \quad a1, ad \otimes p2, t2, t3, t5, t8, t12, t14, t15, to \vdash o}{\quad} \multimap_L \\
 \frac{\frac{sfpc \vdash sfpc \quad d \vdash d}{sfpc, d \vdash sfpc \otimes d} \otimes_R \quad a1, p1, t2, 3, 5, 8, p1 \multimap ad \otimes p2, t12, t14, t15, to \vdash o}{\quad} \multimap_L \\
 \frac{\quad}{sfpc, a1, d, t2, t3, t5, t8, sfpc \otimes d \multimap p1, t11, t12, t14, t15, to \vdash o} \otimes_L \\
 \frac{sfa \vdash sfa \quad sfpc, a1 \otimes d, t2, t3, t5, t8, t10, t11, t12, t14, t15, to \vdash o}{\quad} \multimap_L \\
 \frac{\quad}{sfa, sfpc, sfa \multimap a1 \otimes d, t2, t3, t5, t8, t10, t11, t12, t14, t15, to \vdash o} \otimes_L \\
 \frac{i \vdash i \quad sfa \otimes sfpc, t1, t2, t3, t5, t8, t10, t11, t12, t14, t15, to \vdash o}{\quad} \multimap_L \\
 i, t_i, t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, to \vdash o
 \end{array}$$

A árvore de prova para o cenário $Sc_2A \times Sc_2PC$ é dada por:

$$\begin{array}{c}
\frac{\frac{efa \vdash efa \quad efp \vdash efp}{efa, efp \vdash efa \otimes efp} \otimes_R \quad o \vdash o}{\quad} \multimap_L \\
\frac{\frac{a5 \vdash a5 \quad af \vdash af}{a5, af \vdash a5 \otimes af} \otimes_R \quad efp, efa, efa \otimes efp \multimap o \vdash o}{\quad} \multimap_L \\
\frac{\quad}{a5, af, efp, a5 \otimes af \multimap efa, t_o \vdash o} \otimes_L \\
\frac{\quad}{p5 \vdash p5 \quad a5, af \otimes efp, t_9, t_o \vdash o} \multimap_L \\
\frac{\frac{fv \vdash fv \quad p4 \vdash p4}{fv, p4 \vdash fv \otimes p4} \otimes_R \quad a5, p5, p5 \multimap af \otimes efp, t_9, t_o \vdash o}{\quad} \multimap_L \\
\frac{\quad}{p4, a5, fv, fv \otimes p4 \multimap p5, t_9, t_{17}, o \vdash o} \otimes_L \\
\frac{\quad}{a4 \vdash a4 \quad p4, a5 \otimes fv, t_9, t_{16}, t_{17}, o \vdash o} \multimap_L \\
\frac{\quad}{a3 \vdash a3 \quad p4, a4, a4 \multimap a5 \otimes fv, t_9, t_{16}, t_{17}, o \vdash o} \multimap_L \\
\frac{\frac{a2 \vdash a2 \quad a \vdash a}{a2, a \vdash a2 \otimes a} \otimes_R \quad p4, a3, a3 \multimap a4, t_7, t_9, t_{16}, t_{17}, o \vdash o}{\quad} \multimap_L \\
\frac{\quad}{a2, a, p4, a2 \otimes a \multimap a3, t_5, t_7, t_9, t_{16}, t_{17}, o \vdash o} \otimes_L \\
\frac{\quad}{p3 \vdash p3 \quad a2, a \otimes p4, t_3, t_5, t_7, t_9, t_{16}, t_{17}, o \vdash o} \multimap_L \\
\frac{\quad}{p2 \vdash p2 \quad a2, p3, t_3, t_5, t_7, t_9, p3 \multimap a \otimes p4, t_{16}, t_{17}, o \vdash o} \multimap_L \\
\frac{\frac{a1 \vdash a1 \quad ad \vdash ad}{a1, ad \vdash a1 \otimes ad} \otimes_R \quad p2, a2, t_3, t_5, t_7, t_9, p2 \multimap p3, t_{14}, t_{16}, t_{17}, o \vdash o}{\quad} \multimap_L \\
\frac{\quad}{a1, ad, p2, a1 \otimes ad \multimap a2, t_3, t_5, t_7, t_9, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o} \otimes_L \\
\frac{\quad}{p1 \vdash p1 \quad a1, ad \otimes p2, t_2, t_3, t_5, t_7, t_9, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o} \multimap_L \\
\frac{\frac{sfpc \vdash sfpc \quad d \vdash d}{sfpc, d \vdash sfpc \otimes d} \otimes_R \quad a1, p1, t_2, 3, 5, 7, 9, p1 \multimap ad \otimes p2, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o}{\quad} \multimap_L \\
\frac{\quad}{sfpc, a1, d, t_2, t_3, t_5, t_7, t_9, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o} \otimes_L \\
\frac{\quad}{sfa \vdash sfa \quad sfpc, a1 \otimes d, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o} \multimap_L \\
\frac{\quad}{sfa, sfpc, sfa \multimap a1 \otimes d, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o} \otimes_L \\
\frac{\quad}{i \vdash i \quad sfa \otimes sfpc, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o} \multimap_L \\
i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o
\end{array}$$

Finalmente, a árvore de prova para o cenário $Sc_1A \times Sc_3PC$ é dada por:

$$\begin{array}{c}
 \frac{\frac{efa \vdash efa \quad efpc \vdash efpc}{efa, efpc \vdash efa \otimes efpc} \otimes_R \quad o \vdash o}{\vdash o} \multimap_L \\
 \frac{\frac{a2 \vdash a2 \quad r \vdash r}{a2, r \vdash a2 \otimes r} \otimes_R \quad efpc, efa, efa \otimes efpc \multimap o \vdash o}{\vdash o} \multimap_L \\
 \frac{\frac{a1 \vdash a1 \quad ad \vdash ad}{a1, ad \vdash a1 \otimes ad} \otimes_R \quad r, efpc, a2, a2 \otimes r \multimap efa, t_o \vdash o}{\vdash o} \multimap_L \\
 \frac{a1, ad, r, efpc, a1 \otimes ad \multimap a2, t_4, o \vdash o}{\vdash o} \otimes_L \\
 \frac{p3 \vdash p3 \quad a1, ad, r \otimes efpc, t_2, 4, o \vdash o}{\vdash o} \multimap_L \\
 \frac{p2 \vdash p2 \quad a1, ad, p3, p3 \multimap r \otimes efpc, t_2, 4, o \vdash o}{\vdash o} \multimap_L \\
 \frac{a1, ad, p2, p2 \multimap p3, t_2, 4, 13, o \vdash o}{\vdash o} \otimes_L \\
 \frac{p1 \vdash p1 \quad a1, ad \otimes p2, t_2, 4, 12, 13, o \vdash o}{\vdash o} \multimap_L \\
 \frac{\frac{sfpc \vdash sfpc \quad d \vdash d}{sfpc, d \vdash sfpc \otimes d} \otimes_R \quad a1, p1, p1 \multimap ad \otimes p2, t_2, 4, 12, 13, o \vdash o}{\vdash o} \multimap_L \\
 \frac{sfpc, a1, d, sfpc \otimes d \multimap p1, t_2, 4, 11, 12, 13, o \vdash o}{\vdash o} \otimes_L \\
 \frac{sfa \vdash sfa \quad sfpc, a1 \otimes d, t_2, 4, 10, 11, 12, 13, o \vdash o}{\vdash o} \multimap_L \\
 \frac{sfa, sfpc, sfa \multimap a1 \otimes d, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o}{\vdash o} \otimes_L \\
 \frac{i \vdash i \quad sfa \otimes sfpc, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o}{\vdash o} \multimap_L \\
 i, t_i, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o
 \end{array}$$

A.2.1 Árvore de Prova da Lógica Linear Rotuladas

Este apêndice apresenta as árvores de prova da Lógica Linear rotuladas, utilizadas no contexto do método para detecção de cenários livres de *deadlock*, apresentado na seção 3.4.

A árvore de prova rotulada para o cenário $Sc_3A \times Sc_1PC$ é dada por:

$$\begin{array}{c}
\frac{\frac{t_6 \quad t_o \quad t_{15} \quad t_o}{efa \vdash efa} \quad \frac{t_{15} \quad t_o}{efpc \vdash efpc} \quad \frac{t_o \quad f_i}{o \vdash o}}{\frac{t_6 \quad t_{15} \quad t_o \quad t_o}{efa, efpc \vdash efa \otimes efpc} \otimes_R} \multimap_L \frac{t_o \quad f_i}{o \vdash o} \\
\hline
\frac{\frac{t_3 \quad t_6 \quad t_{15} \quad t_6}{a3 \vdash a3} \quad \frac{t_{15} \quad t_6}{efpc, efa, efa \otimes efpc \multimap o \vdash o} \quad \frac{t_3 \quad t_{15} \quad t_6 \quad t_6}{a3, tl \vdash a3 \otimes tl} \otimes_R}{\frac{t_3 \quad t_{15} \quad t_{15}}{a3, tl, efpc, a3 \otimes tl \multimap efa, t_o \vdash o} \otimes_L} \multimap_L \frac{t_3 \quad t_{15} \quad t_{15}}{a3, tl, efpc, a3 \otimes tl \multimap efa, t_o \vdash o} \otimes_L \\
\hline
\frac{\frac{t_{14} \quad t_{15}}{p4 \vdash p4} \quad \frac{t_3 \quad t_{15} \quad t_{15}}{a3, tl \otimes efpc, t_6, t_o \vdash o} \quad \frac{t_{14} \quad t_{15}}{p4 \vdash p4} \otimes_R}{\frac{t_{14} \quad t_{15}}{p4 \vdash p4} \otimes_R} \multimap_L \frac{t_{14} \quad t_{15}}{p4 \vdash p4} \\
\hline
\frac{\frac{t_2 \quad t_3 \quad t_{14} \quad t_3}{a2 \vdash a2} \quad \frac{t_{14} \quad t_3}{p4, a3, t_6, p4 \multimap tl \otimes efpc, t_o \vdash o} \quad \frac{t_2 \quad t_{14} \quad t_3 \quad t_3}{a2, a \vdash a2 \otimes a} \otimes_R}{\frac{t_2 \quad t_{14} \quad t_{14}}{a2, a, p4, a2 \otimes a \multimap a3, t_6, t_{15}, t_o \vdash o} \otimes_L} \multimap_L \frac{t_2 \quad t_{14} \quad t_{14}}{a2, a, p4, a2 \otimes a \multimap a3, t_6, t_{15}, t_o \vdash o} \otimes_L \\
\hline
\frac{\frac{t_{12} \quad t_{14}}{p3 \vdash p3} \quad \frac{t_2 \quad t_{14} \quad t_{14}}{a2, a \otimes p4, t_3, t_6, t_{15}, t_o \vdash o} \quad \frac{t_{12} \quad t_{14}}{p3 \vdash p3} \otimes_R}{\frac{t_{12} \quad t_{14}}{p3 \vdash p3} \otimes_R} \multimap_L \frac{t_{12} \quad t_{14}}{p3 \vdash p3} \\
\hline
\frac{\frac{t_{11} \quad t_{12}}{p2 \vdash p2} \quad \frac{t_2 \quad t_{12}}{a2, p3, t_3, t_6, p3 \multimap a \otimes p4, t_{15}, t_o \vdash o} \quad \frac{t_{11} \quad t_{12}}{p2 \vdash p2} \otimes_R}{\frac{t_{11} \quad t_{12}}{p2 \vdash p2} \otimes_R} \multimap_L \frac{t_{11} \quad t_{12}}{p2 \vdash p2} \\
\hline
\frac{\frac{t_1 \quad t_2 \quad t_{11} \quad t_2}{a1 \vdash a1} \quad \frac{t_{11} \quad t_2}{p2, a2, t_3, t_6, p2 \multimap p3, t_{14}, t_{15}, t_o \vdash o} \quad \frac{t_1 \quad t_{11} \quad t_2 \quad t_2}{a1, ad \vdash a1 \otimes ad} \otimes_R}{\frac{t_1 \quad t_{11} \quad t_{11}}{a1, ad, p2, a1 \otimes ad \multimap a2, t_3, t_6, t_{12}, t_{14}, t_{15}, t_o \vdash o} \otimes_L} \multimap_L \frac{t_1 \quad t_{11} \quad t_{11}}{a1, ad, p2, a1 \otimes ad \multimap a2, t_3, t_6, t_{12}, t_{14}, t_{15}, t_o \vdash o} \otimes_L \\
\hline
\frac{\frac{t_{10} \quad t_{11}}{p1 \vdash p1} \quad \frac{t_1 \quad t_{11} \quad t_{11}}{a1, ad \otimes p2, t_2, t_3, t_6, t_{12}, t_{14}, t_{15}, t_o \vdash o} \quad \frac{t_{10} \quad t_{11}}{p1 \vdash p1} \otimes_R}{\frac{t_{10} \quad t_{11}}{p1 \vdash p1} \otimes_R} \multimap_L \frac{t_{10} \quad t_{11}}{p1 \vdash p1} \\
\hline
\frac{\frac{t_i \quad t_{10} \quad t_1 \quad t_{10}}{sfpc \vdash sfpc} \quad \frac{t_1 \quad t_{10}}{d \vdash d} \quad \frac{t_i \quad t_1 \quad t_{10} \quad t_{10}}{sfpc, d \vdash sfpc \otimes d} \otimes_R}{\frac{t_i \quad t_{10} \quad t_1 \quad t_{10}}{sfpc, d \vdash sfpc \otimes d} \otimes_R} \multimap_L \frac{t_i \quad t_{10} \quad t_1 \quad t_{10}}{sfpc, d \vdash sfpc \otimes d} \otimes_R \\
\hline
\frac{\frac{t_i \quad t_1 \quad t_1}{sfpc, a1, d, t_2, t_3, t_6, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o} \otimes_L}{\frac{t_i \quad t_1 \quad t_1}{sfpc, a1, d, t_2, t_3, t_6, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o} \otimes_L} \multimap_L \frac{t_i \quad t_1 \quad t_1}{sfpc, a1, d, t_2, t_3, t_6, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o} \otimes_L \\
\hline
\frac{\frac{t_i \quad t_1 \quad t_1}{sfa \vdash sfa} \quad \frac{t_i \quad t_1 \quad t_1}{sfpc, a1 \otimes d, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o} \quad \frac{t_i \quad t_1 \quad t_1}{sfa \vdash sfa} \otimes_R}{\frac{t_i \quad t_1 \quad t_1}{sfa \vdash sfa} \otimes_R} \multimap_L \frac{t_i \quad t_1 \quad t_1}{sfa \vdash sfa} \\
\hline
\frac{\frac{t_i \quad t_i}{sfa \vdash sfa} \quad \frac{t_i \quad t_i}{sfa \otimes sfpc, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o} \quad \frac{t_i \quad t_i}{sfa \vdash sfa} \otimes_R}{\frac{t_i \quad t_i}{sfa \vdash sfa} \otimes_R} \multimap_L \frac{t_i \quad t_i}{sfa \vdash sfa} \\
\hline
\frac{\frac{t_i \quad t_i}{i \vdash i} \quad \frac{t_i \quad t_i}{sfa \otimes sfpc, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o} \quad \frac{t_i \quad t_i}{i \vdash i} \otimes_R}{\frac{t_i \quad t_i}{i \vdash i} \otimes_R} \multimap_L \frac{t_i \quad t_i}{i \vdash i} \\
\hline
\frac{t_i \quad t_i}{i \vdash i} \multimap_L \frac{t_i \quad t_i}{i \vdash i}
\end{array}$$

A árvore de prova rotulada para o cenário $Sc_4A \times Sc_1PC$ é como se segue:

$$\begin{array}{c}
 \frac{\frac{\frac{t_8 \quad t_o}{efa \vdash efa} \quad \frac{t_{15} \quad t_o}{efpc \vdash efpc}}{t_8 \quad t_{15} \quad t_o \quad t_o} \otimes_R \quad \frac{t_o \vdash f_i}{o} \multimap_L}{efa, efpc \vdash efa \otimes efpc} \multimap_L \\
 \frac{\frac{\frac{t_5 \quad t_8}{a4 \vdash a4} \quad \frac{t_{15} \quad t_8}{tl \vdash tl}}{t_5 \quad t_{15} \quad t_8 \quad t_8} \otimes_R \quad \frac{t_{15} \quad t_8}{efpc, efa, efa \otimes efpc \multimap o \vdash o} \multimap_L}{a4, tl \vdash a3 \otimes tl} \multimap_L \\
 \frac{t_3 \quad t_5 \quad t_{15} \quad t_{15} \quad t_5}{a3 \vdash a3 \quad tl, efpc, a4, a4 \otimes tl \multimap efa, t_o \vdash o} \multimap_L \\
 \frac{t_3 \quad t_{15} \quad t_{15}}{a3, tl, efpc, a3 \multimap a4, t_8, t_o \vdash o} \otimes_L \\
 \frac{t_{14} \quad t_{15} \quad t_3 \quad t_{15} \quad t_{15}}{p4 \vdash p4 \quad a3, tl \otimes efpc, t_5, t_8, t_o \vdash o} \multimap_L \\
 \frac{\frac{t_2 \quad t_3}{a2 \vdash a2} \quad \frac{t_{14} \quad t_3}{a \vdash a}}{t_2 \quad t_{14} \quad t_3 \quad t_3} \otimes_R \quad \frac{t_{14} \quad t_3}{p4, a3, t_5, t_8, p4 \multimap tl \otimes efpc, t_o \vdash o} \multimap_L \\
 \frac{t_2 \quad t_{14} \quad t_{14}}{a2, a, p4, a2 \otimes a \multimap a3, t_5, t_8, t_{15}, t_o \vdash o} \otimes_L \\
 \frac{t_{12} \quad t_{14} \quad t_2 \quad t_{14}}{p3 \vdash p3 \quad a2, a \otimes p4, t_3, t_5, t_8, t_{15}, t_o \vdash o} \multimap_L \\
 \frac{t_{11} \quad t_{12} \quad t_2 \quad t_{12}}{p2 \vdash p2 \quad a2, p3, t_3, t_5, t_8, p3 \multimap a \otimes p4, t_{15}, t_o \vdash o} \multimap_L \\
 \frac{\frac{t_1 \quad t_2}{a1 \vdash a1} \quad \frac{t_{11} \quad t_2}{ad \vdash ad}}{t_1 \quad t_{11} \quad t_2 \quad t_2} \otimes_R \quad \frac{t_{11} \quad t_2}{p2, a2, t_3, t_5, t_8, p2 \multimap p3, t_{14}, t_{15}, t_o \vdash o} \multimap_L \\
 \frac{t_1 \quad t_{11} \quad t_{11}}{a1, ad, p2, a1 \otimes ad \multimap a2, t_3, t_5, t_8, t_{12}, t_{14}, t_{15}, t_o \vdash o} \otimes_L \\
 \frac{t_{10} \quad t_{11} \quad t_1 \quad t_{11} \quad t_{11}}{p1 \vdash p1 \quad a1, ad \otimes p2, t_2, t_3, t_5, t_8, t_{12}, t_{14}, t_{15}, t_o \vdash o} \multimap_L \\
 \frac{\frac{t_i \quad t_{10}}{sfpc \vdash sfpc} \quad \frac{t_1 \quad t_{10}}{d \vdash d}}{t_i \quad t_1 \quad t_{10} \quad t_{10}} \otimes_R \quad \frac{t_1 \quad t_{10}}{a1, p1, t_2, 3, 5, 8, p1 \multimap ad \otimes p2, t_{12}, t_{14}, t_{15}, t_o \vdash o} \multimap_L \\
 \frac{t_i \quad t_1 \quad t_1}{sfpc, a1, d, t_2, t_3, t_5, t_8, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o} \otimes_L \\
 \frac{t_i \quad t_1 \quad t_i \quad t_1 \quad t_1}{sfa \vdash sfa \quad sfpc, a1 \otimes d, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o} \multimap_L \\
 \frac{t_i \quad t_i}{sfa, sfpc, sfa \multimap a1 \otimes d, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o} \otimes_L \\
 \frac{i_i \quad i_i \quad t_i \quad t_i}{i \vdash i \quad sfa \otimes sfpc, t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o} \multimap_L \\
 i_i \quad i, i \multimap sfa \otimes sfpc, t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o
 \end{array}$$

A árvore de prova rotulada para o cenário $Sc_2A \times Sc_2PC$ é dada por:

$$\begin{array}{c}
\frac{\frac{\frac{t_9 \quad t_o}{efa \vdash efa} \quad \frac{t_{17} \quad t_o}{efpc \vdash efpc}}{efa, efpc \vdash efa \otimes efpc} \otimes_R \quad \frac{t_o \vdash f_i}{t_o} \multimap_L}{t_o} \\
\\
\frac{\frac{\frac{t_7 \quad t_9}{a5 \vdash a5} \quad \frac{t_{17} \quad t_9}{af \vdash af}}{a5, af \vdash a5 \otimes af} \otimes_R \quad \frac{t_{17} \quad t_9}{efpc, efa, efa \otimes efpc \multimap o \vdash o} \multimap_L}{t_9} \\
\\
\frac{\frac{t_7 \quad t_{17} \quad t_{17}}{a5, af, efpc, a5 \otimes af \multimap efa, t_o \vdash o} \otimes_L \quad \frac{t_{16} \quad t_{17} \quad \frac{t_7 \quad t_{17} \quad t_{17}}{a5, af \otimes efpc, t_9, o \vdash o}}{t_{17}} \multimap_L}{t_{17}} \\
\\
\frac{\frac{\frac{t_7 \quad t_{16}}{fv \vdash fv} \quad \frac{t_{14} \quad t_{16}}{p4 \vdash p4}}{t_7, t_{14} \vdash fv \otimes p4} \otimes_R \quad \frac{t_7 \quad t_{16}}{a5, p5, p5 \multimap af \otimes efpc, t_9, t_o \vdash o} \multimap_L}{t_{16}} \\
\\
\frac{\frac{t_{14} \quad t_7 \quad t_7}{p4, a5, fv, fv \otimes p4 \multimap p5, t_9, t_{17}, o \vdash o} \otimes_L \quad \frac{t_5 \quad t_7 \quad \frac{t_{14} \quad t_7 \quad t_7}{p4, a5 \otimes fv, t_9, 16, 17, o \vdash o}}{t_7} \multimap_L}{t_7} \\
\\
\frac{\frac{t_3 \quad t_5}{a3 \vdash a3} \quad \frac{t_{14} \quad t_5}{p4, a4, a4 \multimap a5 \otimes fv, t_9, 16, 17, o \vdash o} \multimap_L}{t_5} \\
\\
\frac{\frac{\frac{t_2 \quad t_3}{a2 \vdash a2} \quad \frac{t_{14} \quad t_3}{a \vdash a}}{t_2, t_{14} \vdash a2 \otimes a} \otimes_R \quad \frac{t_{14} \quad t_3}{p4, a3, a3 \multimap a4, t_7, t_9, t_{16}, t_{17}, t_o \vdash o} \multimap_L}{t_3} \\
\\
\frac{\frac{t_2 \quad t_{14} \quad t_{14}}{a2, a, p4, a2 \otimes a \multimap a3, t_5, t_7, t_9, t_{16}, 17, o \vdash o} \otimes_L \quad \frac{t_{12} \quad t_{14} \quad \frac{t_2 \quad t_{14} \quad t_{14}}{a2, a \otimes p4, t_3, t_5, t_7, t_9, t_{16}, t_{17}, t_o \vdash o}}{t_{14}} \multimap_L}{t_{14}} \\
\\
\frac{\frac{t_{11} \quad t_{12}}{p2 \vdash p2} \quad \frac{t_2 \quad t_{12}}{a2, p3, t_3, t_5, t_7, t_9, p3 \multimap a \otimes p4, t_{16}, t_{17}, t_o \vdash o} \multimap_L}{t_{12}} \\
\\
\frac{\frac{\frac{t_1 \quad t_2}{a1 \vdash a1} \quad \frac{t_{11} \quad t_2}{ad \vdash ad}}{t_1, t_{11} \vdash a1 \otimes ad} \otimes_R \quad \frac{t_{11} \quad t_2}{p2, a2, t_3, t_5, t_7, t_9, p2 \multimap p3, t_{14}, t_{16}, t_{17}, t_o \vdash o} \multimap_L}{t_2} \\
\\
\frac{\frac{t_1 \quad t_{11} \quad t_{11}}{a1, ad, p2, a1 \otimes ad \multimap a2, t_3, t_5, t_7, t_9, t_{12}, t_{14}, t_{16}, 17, o \vdash o} \otimes_L \quad \frac{t_{10} \quad t_{11} \quad \frac{t_1 \quad t_{11} \quad t_{11}}{a1, ad \otimes p2, t_2, t_3, t_5, t_7, t_9, t_{12}, t_{14}, t_{16}, t_{17}, t_o \vdash o}}{t_{11}} \multimap_L}{t_{11}} \\
\\
\frac{\frac{\frac{t_i \quad t_{10}}{sfpc \vdash sfpc} \quad \frac{t_1 \quad t_{10}}{d \vdash d}}{t_i, t_1 \vdash sfpc \otimes d} \otimes_R \quad \frac{t_1 \quad t_{10}}{a1, p1, t_2, 3, 5, 7, 9, p1 \multimap ad \otimes p2, t_{12}, t_{14}, t_{16}, t_{17}, t_o \vdash o} \multimap_L}{t_{10}} \\
\\
\frac{\frac{t_i \quad t_1 \quad t_1}{sfpc, a1, d, t_2, t_3, t_5, t_7, t_9, sfpc \otimes d \multimap p1, t_{11}, t_{12}, t_{14}, t_{16}, 17, o \vdash o} \otimes_L \quad \frac{t_i \quad t_1 \quad \frac{t_i \quad t_1 \quad t_1}{sfpc, a1 \otimes d, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, t_o \vdash o}}{t_1} \multimap_L}{t_1} \\
\\
\frac{\frac{t_i \quad t_i}{sfa \otimes sfpc, sfa \multimap a1 \otimes d, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, 17, o \vdash o} \otimes_L \quad \frac{i_i \quad t_i \quad \frac{t_i \quad t_i}{sfa \otimes sfpc, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, t_o \vdash o}}{t_i} \multimap_L}{t_i} \\
\\
\frac{i_i}{i, i \multimap sfa \otimes sfpc, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, t_o \vdash o}
\end{array}$$

E, finalmente, a árvore de prova rotulada para o cenário $Sc_1A \times Sc_3PC$ é apresentada na sequência:

$$\begin{array}{c}
\frac{\frac{\frac{t_4 \quad t_o \quad t_{13} \quad t_o}{efa \vdash efa} \quad \frac{t_{13} \quad t_o}{efpc \vdash efpc}}{t_4 \quad t_{13} \quad t_o \quad t_o} \otimes_R \quad \frac{t_o \quad f_i}{o \vdash o} \multimap_L}{efa, efpc \vdash efa \otimes efpc} \multimap_L \\
\frac{\frac{\frac{t_2 \quad t_4 \quad t_{13} \quad t_4}{a2 \vdash a2} \quad \frac{t_2 \quad t_{13} \quad t_4 \quad t_4}{a2, r \vdash a2 \otimes r}}{t_2 \quad t_{13} \quad t_4 \quad t_4} \otimes_R \quad \frac{t_{13} \quad t_4}{efpc, efa, efa \otimes efpc \multimap o \vdash o} \multimap_L}{a2, r \vdash a2 \otimes r} \multimap_L \\
\frac{\frac{\frac{t_1 \quad t_2 \quad t_{11} \quad t_2}{a1 \vdash a1} \quad \frac{t_{11} \quad t_2}{ad \vdash ad}}{t_1 \quad t_{11} \quad t_2 \quad t_2} \otimes_R \quad \frac{t_{13} \quad t_{13} \quad t_2}{r, efpc, a2, a2 \otimes r \multimap efa, t_o \vdash o} \multimap_L}{a1, ad \vdash a1 \otimes ad} \multimap_L \\
\frac{t_1 \quad t_{11} \quad t_{13} \quad t_{13}}{a1, ad, r, efpc, a1 \otimes ad \multimap a2, t_4, o \vdash o} \otimes_L \\
\frac{t_{12} \quad t_{13} \quad t_1 \quad t_{11} \quad t_{13} \quad t_{13}}{p3 \vdash p3 \quad a1, ad, r \otimes efpc, t_2, 4, o \vdash o} \multimap_L \\
\frac{t_{11} \quad t_{12} \quad t_1 \quad t_{11} \quad t_{12}}{p2 \vdash p2 \quad a1, ad, p3, p3 \multimap r \otimes efpc, t_2, 4, o \vdash o} \multimap_L \\
\frac{t_1 \quad t_{11} \quad t_{11}}{a1, ad, p2, p2 \multimap p3, t_2, 4, 13, o \vdash o} \otimes_L \\
\frac{t_{10} \quad t_{11} \quad t_1 \quad t_{11} \quad t_{11}}{p1 \vdash p1 \quad a1, ad \otimes p2, t_2, 4, 12, 13, o \vdash o} \multimap_L \\
\frac{\frac{t_i \quad t_{10} \quad t_1 \quad t_{10}}{sfpc \vdash sfpc} \quad \frac{t_1 \quad t_{10}}{d \vdash d}}{t_i \quad t_1 \quad t_{10} \quad t_{10}} \otimes_R \quad \frac{t_1 \quad t_{10}}{a1, p1, p1 \multimap ad \otimes p2, t_2, 4, 12, 13, o \vdash o} \multimap_L}{sfpc, d \vdash sfpc \otimes d} \multimap_L \\
\frac{t_i \quad t_1 \quad t_1}{sfpc, a1, d, sfpc \otimes d \multimap p1, t_2, 4, 11, 12, 13, o \vdash o} \otimes_L \\
\frac{t_i \quad t_1 \quad t_i \quad t_1 \quad t_1}{sfa \vdash sfa \quad sfpc, a1 \otimes d, t_2, 4, 10, 11, 12, 13, o \vdash o} \multimap_L \\
\frac{t_i \quad t_i}{sfa, sfpc, sfa \multimap a1 \otimes d, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o} \otimes_L \\
\frac{i_i \quad t_i \quad t_i \quad t_i}{i \vdash i \quad sfa \otimes sfpc, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o} \multimap_L \\
i_i, i \multimap sfa \otimes sfpc, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o
\end{array}$$

A.3 Árvore de Prova da Lógica Linear para o Estudo de Caso

Este apêndice apresenta as árvores de prova da Lógica Linear construídas no contexto do Estudo de Caso, apresentado no Capítulo 4.

É importante lembrar que, considerando o sistema composto apresentado na Figura 22, cada transição é representada por:

$$t_i = i \multimap q0 \otimes p0,$$

$$t_1 = q0 \otimes ticket \multimap q1,$$

$$t_2 = q1 \multimap VISA \otimes q2,$$

$$t_3 = q1 \multimap eCash \otimes q2,$$

$$t_4 = p0 \multimap ticket \otimes p1,$$

$$\begin{aligned}
t_5 &= p0 \multimap ticket \otimes p2, \\
t_6 &= p1 \otimes eCash \multimap p3, \\
t_7 &= p2 \otimes VISA \multimap p3, \\
t_o &= q2 \otimes p3 \multimap o.
\end{aligned}$$

A árvore de prova para o Sc_1C1 é dada por:

$$\frac{\frac{\frac{q0 \vdash q0 \quad ticket \vdash ticket}{p0, ticket \vdash p0 \otimes ticket} \otimes_R \quad q1, q1 \multimap VISA \otimes q2 \vdash q2}{q0, ticket, q0 \otimes ticket \multimap q1, t_2 \vdash q2} \multimap_L \quad \frac{\frac{q1 \vdash q1 \quad VISA \otimes q2 \vdash q2}{q1, q1 \multimap VISA \otimes q2 \vdash q2} \multimap_L \quad \frac{VISA, q2 \vdash q2}{q1, q1 \multimap VISA \otimes q2 \vdash q2} \otimes_L$$

A árvore de prova para o cenário Sc_2C1 é como se segue:

$$\frac{\frac{\frac{q0 \vdash q0 \quad ticket \vdash ticket}{q0, ticket \vdash q0 \otimes ticket} \otimes_R \quad q1, q1 \multimap eCash \otimes q2 \vdash q2}{q0, ticket, q0 \otimes ticket \multimap q1, t_3 \vdash q2} \multimap_L \quad \frac{\frac{q1 \vdash q1 \quad eCash, q2 \vdash q2}{q1, q1 \multimap eCash \otimes q2 \vdash q2} \multimap_L \quad \frac{eCash, q2 \vdash q2}{q1, q1 \multimap eCash \otimes q2 \vdash q2} \otimes_L$$

A árvore de prova para o cenário Sc_1T1 é dada por:

$$\frac{\frac{\frac{p1 \vdash p1 \quad eCash \vdash eCash}{p1, eCash \vdash p1 \otimes eCash} \otimes_R \quad ticket, p3 \vdash p3}{eCash, ticket, p1, p1 \otimes eCash \multimap p3 \vdash p3} \multimap_L \quad \frac{p0 \vdash p0 \quad eCash, ticket \otimes p1, t_6 \vdash p3}{p0, eCash, p0 \multimap ticket \otimes p1, t_6 \vdash p3} \multimap_L$$

A árvore de prova para o cenário Sc_2T1 é mostrada na sequência.

$$\frac{\frac{\frac{p2 \vdash p2 \quad VISA \vdash VISA}{p2, VISA \vdash p2 \otimes VISA} \otimes_R \quad ticket, p3 \vdash p3}{VISA, ticket, p2, p2 \otimes VISA \multimap p3 \vdash p3} \multimap_L \quad \frac{p0 \vdash p0 \quad VISA, ticket \otimes p2, t_7 \vdash p3}{p0, VISA, p0 \multimap ticket \otimes p2, t_7 \vdash p3} \multimap_L$$

A árvore de prova para o cenário $Sc_1C1 \oplus T1$ (ou $Sc_2C1 \times Sc_1T1$) é como se segue:

$$\begin{array}{c}
\frac{\frac{q2 \vdash q2 \quad p3 \vdash p3}{q2, p3 \vdash q2 \otimes p3} \otimes_R \quad o \vdash o}{\quad} \multimap_L \\
\frac{\frac{p1 \vdash p1 \quad eCash \vdash eCash}{p1, eCash \vdash p1 \otimes eCash} \otimes_R \quad q2, p3, q2 \otimes p3 \multimap o \vdash o}{\quad} \multimap_L \\
\frac{\quad}{p1, eCash, q2, p1 \otimes eCash \multimap p3, t_o \vdash o} \otimes_L \\
\frac{q1 \vdash q1 \quad p1, eCash \otimes q2, t_6, t_o \vdash o}{\quad} \multimap_L \\
\frac{\frac{q0 \vdash q0 \quad ticket \vdash ticket}{q0, ticket \vdash q0 \otimes ticket} \otimes_R \quad p1, q1, q1 \multimap eCash \otimes q2, t_6, t_o \vdash o}{\quad} \multimap_L \\
\frac{\quad}{q0, ticket, p1, q0 \otimes ticket \multimap q1, t_3, t_6, t_o \vdash o} \otimes_L \\
\frac{p0 \vdash p0 \quad q0, ticket \otimes p1, t_1, t_3, t_6, t_o \vdash o}{\quad} \multimap_L \\
\frac{\quad}{q0, p0, t_1, t_3, p0 \multimap ticket \otimes p1, t_6, t_o \vdash o} \otimes_L \\
\frac{i \vdash i \quad q0 \otimes p0, t_1, t_3, t_4, t_6, t_o \vdash o}{\quad} \multimap_L \\
i, i \multimap q0 \otimes p0, t_1, t_3, t_4, t_6, t_o \vdash o
\end{array}$$

A árvore de prova para o cenário $Sc_2C1 \oplus T1$ (ou $Sc_1C1 \times Sc_2T1$) é dada por:

$$\begin{array}{c}
\frac{\frac{q2 \vdash q2 \quad p3 \vdash p3}{q2, p3 \vdash q2 \otimes p3} \otimes_R \quad o \vdash o}{\quad} \multimap_L \\
\frac{\frac{p2 \vdash p2 \quad VISA \vdash VISA}{p2, VISA \vdash p2 \otimes VISA} \otimes_R \quad q2, p3, q2 \otimes p3 \multimap o \vdash o}{\quad} \multimap_L \\
\frac{\quad}{p2, VISA, q2, p2 \otimes VISA \multimap p3, t_o \vdash o} \otimes_L \\
\frac{q1 \vdash q1 \quad p2, VISA \otimes q2, t_7, t_o \vdash o}{\quad} \multimap_L \\
\frac{\frac{q0 \vdash q0 \quad ticket \vdash ticket}{q0, ticket \vdash q0 \otimes ticket} \otimes_R \quad p2, q1, q1 \multimap VISA \otimes q2, t_7, t_o \vdash o}{\quad} \multimap_L \\
\frac{\quad}{q0, ticket, p2, q0 \otimes ticket \multimap q1, t_2, t_7, t_o \vdash o} \otimes_L \\
\frac{p0 \vdash p0 \quad q0, ticket \otimes p2, t_1, t_2, t_7, t_o \vdash o}{\quad} \multimap_L \\
\frac{\quad}{q0, p0, t_1, t_2, p0 \multimap ticket \otimes p2, t_7, t_o \vdash o} \otimes_L \\
\frac{i \vdash i \quad q0 \otimes p0, t_1, t_2, t_5, t_7, t_o \vdash o}{\quad} \multimap_L \\
i, i \multimap q0 \otimes p0, t_1, t_2, t_5, t_7, t_o \vdash o
\end{array}$$

A árvore de prova para o cenário $Sc_3C1 \oplus T1$ é apresentada na sequência.

$$\begin{array}{c}
\frac{}{p1, VISA, q2, t_6, t_o \vdash o} \otimes_L \\
\frac{q1 \vdash q1 \quad p1, VISA \otimes q2, t_6, t_o \vdash o}{\quad} \multimap_L \\
\frac{\frac{q0 \vdash q0 \quad ticket \vdash ticket}{q0, ticket \vdash q0 \otimes ticket} \otimes_R \quad p1, q1, q1 \multimap VISA \otimes q2, t_6, t_o \vdash o}{\quad} \multimap_L \\
\frac{}{q0, ticket, p1, q0 \otimes ticket \multimap q1, t_2, t_6, t_o \vdash o} \otimes_L \\
\frac{p0 \vdash p0 \quad q0, ticket \otimes p1, t_1, t_2, t_6, t_o \vdash o}{\quad} \multimap_L \\
\frac{}{q0, p0, t_1, t_2, p0 \multimap ticket \otimes p1, t_6, t_o \vdash o} \otimes_L \\
\frac{i \vdash i \quad q0 \otimes p0, t_1, t_2, t_4, t_6, t_o \vdash o}{\quad} \multimap_L \\
i, i \multimap q0 \otimes p0, t_1, t_2, t_4, t_6, t_o \vdash o
\end{array}$$

A árvore de prova para o cenário $Sc_4C1 \oplus T1$ é dada por:

$$\begin{array}{c}
\frac{}{p2, eCash, q2, t_7, t_o \vdash o} \otimes_L \\
\frac{q1 \vdash q1 \quad p2, eCash \otimes q2, t_7, t_o \vdash o}{\quad} \multimap_L \\
\frac{\frac{q0 \vdash q0 \quad ticket \vdash ticket}{q0, ticket \vdash q0 \otimes ticket} \otimes_R \quad p2, q1, q1 \multimap eCash \otimes q2, t_7, t_o \vdash o}{\quad} \multimap_L \\
\frac{}{q0, ticket, p2, q0 \otimes ticket \multimap q1, t_3, t_7, t_o \vdash o} \otimes_L \\
\frac{p0 \vdash p0 \quad q0, ticket \otimes p2, t_1, t_3, t_7, t_o \vdash o}{\quad} \multimap_L \\
\frac{}{q0, p0, t_1, t_3, p0 \multimap ticket \otimes p2, t_7, t_o \vdash o} \otimes_L \\
\frac{i \vdash i \quad q0 \otimes p0, t_1, t_3, t_5, t_7, t_o \vdash o}{\quad} \multimap_L \\
i, i \multimap q0 \otimes p0, t_1, t_3, t_5, t_7, t_o \vdash o
\end{array}$$

A.3.1 Árvores de Prova da Lógica Linear Rotuladas para o Estudo de Caso

A árvore de prova rotulada para o cenário $Sc_1C1 \oplus T1$ (ou $Sc_2C1 \times Sc_1T1$) é como se segue:

$$\begin{array}{c}
\frac{\frac{t_3 \ t_o \ t_6 \ t_o}{q2 \vdash q2 \ p3 \vdash p3} \otimes_R \frac{t_o \ f_i}{o \vdash o} \multimap_L}{\frac{t_3 \ t_6 \ t_o \ t_o}{q2, p3 \vdash q2 \otimes p3} \otimes_R} \multimap_L \\
\frac{\frac{t_4 \ t_6}{p1 \vdash p1} \frac{t_3 \ t_6}{eCash \vdash eCash} \otimes_R \frac{t_3 \ t_6}{q2, p3, q2 \otimes p3 \multimap o \vdash o} \multimap_L}{\frac{t_4 \ t_3 \ t_6 \ t_6}{p1, eCash \vdash p1 \otimes eCash} \otimes_R} \multimap_L \\
\frac{\frac{t_4 \ t_3 \ t_3}{p1, eCash, q2, p1 \otimes eCash \multimap p3, t_o \vdash o} \otimes_L}{\frac{t_1 \ t_3 \ t_4 \ t_3 \ t_3}{q1 \vdash q1 \ p1, eCash \otimes q2, t_6, t_o \vdash o} \multimap_L} \multimap_L \\
\frac{\frac{t_i \ t_1 \ t_4 \ t_1}{q0 \vdash q0 \ ticket \vdash ticket} \otimes_R \frac{t_4 \ t_1}{p1, q1, q1 \multimap eCash \otimes q2, t_6, t_o \vdash o} \multimap_L}{\frac{t_i \ t_4 \ t_4}{q0, ticket \vdash q0 \otimes ticket} \otimes_R} \multimap_L \\
\frac{\frac{t_i \ t_4 \ t_4}{q0, ticket, p1, q0 \otimes ticket \multimap q1, t_3, t_6, t_o \vdash o} \otimes_L}{\frac{t_i \ t_4 \ t_4}{p0 \vdash p0 \ q0, ticket \otimes p1, t_1, t_3, t_6, t_o \vdash o} \multimap_L} \multimap_L \\
\frac{\frac{t_i \ t_i}{q0, p0, t_1, t_3, p0 \multimap ticket \otimes p1, t_6, t_o \vdash o} \otimes_L}{\frac{i \vdash i \ t_i \ t_i}{q0 \otimes p0, t_1, t_3, t_4, t_6, t_o \vdash o} \multimap_L} \multimap_L \\
\frac{i \vdash i \ t_i \ t_i}{i, i \multimap q0 \otimes p0, t_1, t_3, t_4, t_6, t_o \vdash o} \multimap_L
\end{array}$$

A árvore de prova rotulada para o cenário $Sc_2C1 \oplus T1$ (ou $Sc_1C1 \times Sc_2T1$) é dada por:

$$\begin{array}{c}
\frac{\frac{t_2 \ t_o \ t_7 \ t_o}{q2 \vdash q2 \ p3 \vdash p3} \otimes_R \frac{t_o \ f_i}{o \vdash o} \multimap_L}{\frac{t_2 \ t_7 \ t_o \ t_o}{q2, p3 \vdash q2 \otimes p3} \otimes_R} \multimap_L \\
\frac{\frac{t_5 \ t_7}{p2 \vdash p2} \frac{t_2 \ t_7}{VISA \vdash VISA} \otimes_R \frac{t_2 \ t_7}{q2, p3, q2 \otimes p3 \multimap o \vdash o} \multimap_L}{\frac{t_5 \ t_2 \ t_7 \ t_7}{p2, VISA \vdash p2 \otimes VISA} \otimes_R} \multimap_L \\
\frac{\frac{t_5 \ t_2 \ t_2}{p2, VISA, q2, p2 \otimes VISA \multimap p3, t_o \vdash o} \otimes_L}{\frac{t_1 \ t_2 \ t_5 \ t_2}{q1 \vdash q1 \ p2, VISA \otimes q2, t_7, t_o \vdash o} \multimap_L} \multimap_L \\
\frac{\frac{t_i \ t_1 \ t_5 \ t_1}{q0 \vdash q0 \ ticket \vdash ticket} \otimes_R \frac{t_5 \ t_1}{p2, q1, q1 \multimap VISA \otimes q2, t_7, t_o \vdash o} \multimap_L}{\frac{t_i \ t_5 \ t_1 \ t_1}{q0, ticket \vdash q0 \otimes ticket} \otimes_R} \multimap_L \\
\frac{\frac{t_i \ t_5 \ t_5}{q0, ticket, p2, q0 \otimes ticket \multimap q1, t_2, t_7, t_o \vdash o} \otimes_L}{\frac{t_i \ t_5 \ t_5}{p0 \vdash p0 \ q0, ticket \otimes p2, t_1, t_2, t_7, t_o \vdash o} \multimap_L} \multimap_L \\
\frac{\frac{t_i \ t_i}{q0, p0, t_1, t_2, p0 \multimap ticket \otimes p2, t_7, t_o \vdash o} \otimes_L}{\frac{i \vdash i \ t_i \ t_i}{q0 \otimes p0, t_1, t_2, t_5, t_7, t_o \vdash o} \multimap_L} \multimap_L \\
\frac{i \vdash i \ t_i \ t_i}{i, i \multimap q0 \otimes p0, t_1, t_2, t_5, t_7, t_o \vdash o} \multimap_L
\end{array}$$

A.4 Árvores de prova da Lógica Linear com Cálculo de Datas para o Planejamento de Recursos

Este apêndice apresenta as árvores de prova da Lógica Linear com cálculo de datas, construídas no contexto da análise quantitativa, ou seja, para o planejamento de recursos, apresentada na seção 5.1.

É importante recordar que as transições da $U(\text{IOWF-net})$ são representadas por:

$$\begin{aligned}
 t_i &= i \multimap \text{start_flow_author} \otimes \text{start_flow_PC}, \\
 t_1 &= \text{send_draft} = \text{start_flow_author} \multimap a_1 \otimes \text{draft}, \\
 t_2 &= \text{receive_ack_draft} = a_1 \otimes \text{ack_draft} \multimap a_2, \\
 t_3 &= \text{receive_accept} = a_2 \otimes \text{accept} \multimap a_3, \\
 t_4 &= \text{receive_reject} = a_2 \otimes \text{reject} \multimap \text{end_flow_author}, \\
 t_5 &= \text{prepare_final_version} = a_3 \multimap a_4, \\
 t_6 &= \text{receive_notification_1} = a_3 \otimes \text{too_late} \multimap \text{end_flow_author}, \\
 t_7 &= \text{send_final_version} = a_4 \multimap a_5 \otimes \text{final_version}, \\
 t_8 &= \text{receive_notification_2} = a_4 \otimes \text{too_late} \multimap \text{end_flow_author}, \\
 t_9 &= \text{receive_ack_final} = a_5 \otimes \text{ack_final} \multimap \text{end_flow_author}, \\
 t_{10} &= \text{receive_draft} = \text{start_flow_PC} \otimes \text{draft} \multimap p_1, \\
 t_{11} &= \text{send_ack_draft} = p_1 \multimap \text{ack_draft} \otimes p_2, \\
 t_{12} &= \text{evaluate} = p_2 \multimap p_3, \\
 t_{13} &= \text{send_reject} = p_3 \multimap \text{reject} \otimes \text{end_flow_PC}, \\
 t_{14} &= \text{send_accept} = p_3 \multimap \text{accept} \otimes p_4, \\
 t_{15} &= \text{too_late} = p_4 \multimap \text{too_late} \otimes \text{end_flow_PC}, \\
 t_{16} &= \text{receive_final_version} = \text{final_version} \otimes p_4 \multimap p_5, \\
 t_{17} &= \text{send_ack_final} = p_5 \multimap \text{ack_final} \otimes \text{end_flow_PC}, \\
 t_o &= \text{end_flow_author} \otimes \text{end_flow_PC} \multimap o.
 \end{aligned}$$

Além disso, os átomos nas árvores de prova poderão ser representados apenas pelas letras iniciais de seus nomes completos. Por exemplo, *too_late* e *final_version* poderão ser representados por *tl* e *fv*, respectivamente. Além disso, um conjunto de transições t_i, t_j, t_k and t_l pode ser representado por $t_{i,j,k,l}$. Estas reduções podem ocorrer devido ao tamanho e legibilidade das árvores de prova apresentadas.

A árvore de prova da Lógica Linear com cálculo de datas, para o cenário $Sc_1A \times Sc_3PC$, onde $Seq = D_i + d_i + d_1 + d_{10} + d_{11}$, é dada por:

$$\begin{array}{c}
\frac{\frac{efa(Seq+max\{d_2,d_{12}+d_{13}\}+d_4,Seq+max\{d_2,d_{12}+d_{13}\}+d_4+d_o) \vdash efa \quad efp(Seq+d_{12}+d_{13},Seq+max\{d_2,d_{12}+d_{13}\}+d_4+d_o) \vdash efp}{efa(Seq+max\{d_2,d_{12}+d_{13}\}+d_4,Seq+max\{d_2,d_{12}+d_{13}\}+d_4+d_o),efp(Seq+d_{12}+d_{13},Seq+max\{d_2,d_{12}+d_{13}\}+d_4+d_o) \vdash efa \otimes efp} \otimes_R \quad o(Seq+max\{d_2,d_{12}+d_{13}\}+d_4+d_o,.) \vdash o}{\vdash o} \multimap_L \\
\frac{\frac{a2(Seq+d_2,Seq+max\{d_2,d_{12}+d_{13}\}+d_4) \vdash a2 \quad r(Seq+d_{12}+d_{13},Seq+max\{d_2,d_{12}+d_{13}\}+d_4) \vdash r}{a2(Seq+d_2,Seq+max\{d_2,d_{12}+d_{13}\}+d_4),r(Seq+d_{12}+d_{13},Seq+max\{d_2,d_{12}+d_{13}\}+d_4) \vdash a2 \otimes r} \otimes_R \quad efp(Seq+d_{12}+d_{13},.),efa(Seq+max\{d_2,d_{12}+d_{13}\}+d_4,.) \vdash efa \otimes efp}{\vdash o} \multimap_L \\
\frac{\frac{\frac{a1(D_i+d_i+d_1,Seq+d_2) \vdash a1 \quad ad(Seq,Seq+d_2) \vdash ad}{a1(D_i+d_i+d_1,Seq+d_2),ad(Seq,Seq+d_2) \vdash a1 \otimes ad} \otimes_R \quad r(Seq+d_{12}+d_{13},.),efp(Seq+d_{12}+d_{13},.),a2(Seq+d_2,.) \vdash a2 \otimes r}{\vdash o} \multimap_L \quad efa,t_o \vdash o}{\vdash o} \multimap_L \\
\frac{a1(D_i+d_i+d_1,.) \vdash a1,ad(Seq,.) \vdash ad, r(Seq+d_{12}+d_{13},.) \vdash r,efp(Seq+d_{12}+d_{13},.) \vdash efp}{\vdash o} \multimap_L \\
\frac{p3(Seq+d_{12},Seq+d_{12}+d_{13}) \vdash p3 \quad a1(D_i+d_i+d_1,.) \vdash a1,ad(Seq,.) \vdash ad, r(Seq+d_{12}+d_{13},.) \vdash r,efp(Seq+d_{12}+d_{13},.) \vdash efp}{\vdash o} \multimap_L \\
\frac{p2(Seq,Seq+d_{12}) \vdash p2 \quad a1(D_i+d_i+d_1,.) \vdash a1,ad(Seq,.) \vdash ad, p3(Seq+d_{12},.) \vdash p3}{\vdash o} \multimap_L \\
\frac{a1(D_i+d_i+d_1,.) \vdash a1,ad(Seq,.) \vdash ad, p2(Seq,.) \vdash p2}{\vdash o} \multimap_L \\
\frac{p1(D_i+d_i+d_1+d_{10},Seq) \vdash p1 \quad a1(D_i+d_i+d_1,.) \vdash a1,ad(Seq,.) \vdash ad, p2(Seq,.) \vdash p2}{\vdash o} \multimap_L \\
\frac{\frac{sfpc(D_i+d_i,D_i+d_i+d_1+d_{10}) \vdash sfpc \quad d(D_i+d_i+d_1,D_i+d_i+d_1+d_{10}) \vdash d}{sfpc(D_i+d_i,D_i+d_i+d_1+d_{10}),d(D_i+d_i+d_1,D_i+d_i+d_1+d_{10}) \vdash sfpc \otimes d} \otimes_R \quad a1(D_i+d_i+d_1,.) \vdash a1,p1(D_i+d_i+d_1+d_{10},.) \vdash p1}{\vdash o} \multimap_L \\
\frac{sfpc(D_i+d_i,.) \vdash sfpc, a1(D_i+d_i+d_1,.) \vdash a1, d(D_i+d_i+d_1,.) \vdash d, sfpc \otimes d \vdash p1}{\vdash o} \multimap_L \\
\frac{sfa(D_i+d_i,D_i+d_i+d_1) \vdash sfa \quad sfpc(D_i+d_i,.) \vdash sfpc, a1(D_i+d_i+d_1,.) \vdash a1, d(D_i+d_i+d_1,.) \vdash d, sfpc \otimes d \vdash p1}{\vdash o} \multimap_L \\
\frac{sfa(D_i+d_i,.) \vdash sfa, sfpc(D_i+d_i,.) \vdash sfpc, sfa \otimes a1 \vdash d, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o}{\vdash o} \multimap_L \\
\frac{i(D_i,D_i+d_i) \vdash i \quad sfa(D_i+d_i,.) \vdash sfa, sfpc(D_i+d_i,.) \vdash sfpc, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o}{\vdash o} \multimap_L \\
i(D_i,.) \vdash i, sfa \otimes sfpc, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o
\end{array}$$

A árvore de prova da Lógica Linear com cálculo de datas, para o cenário $Sc_2A \times Sc_2PC$, onde $Seq = d_{10} + d_{11} + d_{12} + d_{14}$ e $Seq1 = D_i + d_i + d_1 + d_{10} + d_{11} + \max\{d_2, d_{12} + d_{14}\} + d_3$, é dada por:

$$\begin{array}{c}
\frac{\frac{efa(Seq1+d_5+d_7+d_{16}+d_{17}+d_9, Seq1+d_5+d_7+d_{16}+d_{17}+d_9+d_o) \vdash efa \quad efp(Seq1+d_5+d_7+d_{16}+d_{17}, Seq1+d_5+d_7+d_{16}+d_{17}+d_9+d_o) \vdash efp}{efa(Seq1+d_5+d_7+d_{16}+d_{17}+d_9, Seq1+d_5+d_7+d_{16}+d_{17}+d_9+d_o), efp(Seq1+d_5+d_7+d_{16}+d_{17}, Seq1+d_5+d_7+d_{16}+d_{17}+d_9+d_o) \vdash efa \otimes efp} \otimes_R \quad o(Seq1+d_5+d_7+d_{16}+d_{17}+d_9+d_o,.) \vdash o}{\multimap_L} \\
\frac{\frac{a5(Seq1+d_5+d_7, Seq1+d_5+d_7+d_{16}+d_{17}+d_9) \vdash a5 \quad af(Seq1+d_5+d_7+d_{16}+d_{17}, Seq1+d_5+d_7+d_{16}+d_{17}+d_9) \vdash af}{a5(Seq1+d_5+d_7, Seq1+d_5+d_7+d_{16}+d_{17}+d_9), af(Seq1+d_5+d_7+d_{16}+d_{17}, Seq1+d_5+d_7+d_{16}+d_{17}+d_9) \vdash a5 \otimes af} \otimes_R \quad efp(Seq1+d_5+d_7+d_{16}+d_{17},.), efa(Seq1+d_5+d_7+d_{16}+d_{17}+d_9,.), efa \otimes efp \multimap o \vdash o}{\multimap_L} \\
\frac{\quad}{a5(Seq1+d_5+d_7,.), af(Seq1+d_5+d_7+d_{16}+d_{17},.), efp(Seq1+d_5+d_7+d_{16}+d_{17},.), a5 \otimes af \multimap efa, t_o \vdash o} \otimes_L \\
\frac{p5(Seq1+d_5+d_7+d_{16}, Seq1+d_5+d_7+d_{16}+d_{17}) \vdash p5 \quad a5(Seq1+d_5+d_7,.), af(Seq1+d_5+d_7+d_{16}+d_{17},.) \otimes efp(Seq1+d_5+d_7+d_{16}+d_{17},.), t_9, o \vdash o}{\multimap_L} \\
\frac{\frac{fv(Seq1+d_5+d_7, Seq1+d_5+d_7+d_{16}) \vdash fv \quad p4(D_i+d_i+d_1+Seq, Seq1+d_5+d_7+d_{16}) \vdash p4}{fv(Seq1+d_5+d_7, Seq1+d_5+d_7+d_{16}), p4(D_i+d_i+d_1+Seq, Seq1+d_5+d_7+d_{16}) \vdash fv \otimes p4} \otimes_R \quad a5(Seq1+d_5+d_7,.), p5(Seq1+d_5+d_7+d_{16},.), p5 \multimap af \otimes efp, t_9, o \vdash o}{\multimap_L} \\
\frac{\quad}{p4(D_i+d_i+d_1+Seq,.), a5(Seq1+d_5+d_7,.), fv(Seq1+d_5+d_7,.), fv \otimes p4 \multimap p5, t_9, 17, o \vdash o} \otimes_L \\
\frac{a4(Seq1+d_5, Seq1+d_5+d_7) \vdash a4 \quad p4(D_i+d_i+d_1+Seq,.), a5(Seq1+d_5+d_7,.) \otimes fv(Seq1+d_5+d_7,.), t_9, 16, 17, o \vdash o}{\multimap_L} \\
\frac{a3(Seq1, Seq1+d_5) \vdash a3 \quad p4(D_i+d_i+d_1+Seq,.), a4(Seq1+d_5,.), a4 \multimap a5 \otimes fv, t_9, 16, 17, o \vdash o}{\multimap_L} \\
\frac{\frac{a2(D_i+d_i+d_1+d_{10}+d_{11}+d_2, Seq1) \vdash a2 \quad a(D_i+d_i+d_1+Seq, Seq1) \vdash a}{a2(D_i+d_i+d_1+d_{10}+d_{11}+d_2, Seq1), a(D_i+d_i+d_1+Seq, Seq1) \vdash a2 \otimes a} \otimes_R \quad p4(D_i+d_i+d_1+Seq,.), a3(Seq1,.), a3 \multimap a4, t_7, 9, 16, 17, o \vdash o}{\multimap_L} \\
\frac{\quad}{a2(D_i+d_i+d_1+d_{10}+d_{11}+d_2,.), a(D_i+d_i+d_1+Seq,.), p4(D_i+d_i+d_1+Seq,.), a2 \otimes a \multimap a3, t_5, 7, 9, 16, 17, o \vdash o} \otimes_L \\
\frac{p3(D_i+d_i+d_1+d_{10}+d_{11}+d_{12}, D_i+d_i+d_1+Seq) \vdash p3 \quad a2(D_i+d_i+d_1+d_{10}+d_{11}+d_2,.), a(D_i+d_i+d_1+Seq,.) \otimes p4(D_i+d_i+d_1+Seq,.), t_3, 5, 7, 9, 16, 17, o \vdash o}{\multimap_L} \\
\frac{p2(D_i+d_i+d_1+d_{10}+d_{11}, D_i+d_i+d_1+d_{10}+d_{11}+d_{12}) \vdash p2 \quad a2(D_i+d_i+d_1+d_{10}+d_{11}+d_2,.), p3(D_i+d_i+d_1+d_{10}+d_{11}+d_{12},.), p3 \multimap a \otimes p4, t_3, 5, 7, 9, 16, 17, o \vdash o}{\multimap_L} \\
\frac{\frac{a1(D_i+d_i+d_1, D_i+d_i+d_1+d_{10}+d_{11}+d_2) \vdash a1 \quad ad(D_i+d_i+d_1+d_{10}+d_{11}, D_i+d_i+d_1+d_{10}+d_{11}+d_2) \vdash ad}{a1(D_i+d_i+d_1, D_i+d_i+d_1+d_{10}+d_{11}+d_2), ad(D_i+d_i+d_1+d_{10}+d_{11}, D_i+d_i+d_1+d_{10}+d_{11}+d_2) \vdash a1 \otimes ad} \otimes_R \quad p2(D_i+d_i+d_1+d_{10}+d_{11},.), a2(D_i+d_i+d_1+d_{10}+d_{11}+d_2,.), p2 \multimap p3, t_3, 5, 7, 9, 14, 16, 17, o \vdash o}{\multimap_L} \\
\frac{\quad}{a1(D_i+d_i+d_1,.), ad(D_i+d_i+d_1+d_{10}+d_{11},.), p2(D_i+d_i+d_1+d_{10}+d_{11},.), a1 \otimes ad \multimap a2, t_3, 5, 7, 9, 12, 14, 16, 17, o \vdash o} \otimes_L \\
\frac{p1(D_i+d_i+d_1+d_{10}, D_i+d_i+d_1+d_{10}+d_{11}) \vdash p1 \quad a1(D_i+d_i+d_1,.), ad(D_i+d_i+d_1+d_{10}+d_{11},.) \otimes p2(D_i+d_i+d_1+d_{10}+d_{11},.), t_2, 3, 5, 7, 9, 12, 14, 16, 17, o \vdash o}{\multimap_L} \\
\frac{\frac{sfpc(D_i+d_i, D_i+d_i+d_1+d_{10}) \vdash sfpc \quad d(D_i+d_i+d_1, D_i+d_i+d_1+d_{10}) \vdash d}{sfpc(D_i+d_i, D_i+d_i+d_1+d_{10}), d(D_i+d_i+d_1, D_i+d_i+d_1+d_{10}) \vdash sfpc \otimes d} \otimes_R \quad a1(D_i+d_i+d_1,.), p1(D_i+d_i+d_1+d_{10},.), p1 \multimap ad \otimes p2, t_2, 3, 5, 7, 9, 12, 14, 16, 17, o \vdash o}{\multimap_L} \\
\frac{\quad}{sfpc(D_i+d_i,.), a1(D_i+d_i+d_1,.), d(D_i+d_i+d_1,.), sfpc \otimes d \multimap p1, t_2, 3, 5, 7, 9, 11, 12, 14, 16, 17, o \vdash o} \otimes_L \\
\frac{sf a(D_i+d_i, D_i+d_i+d_1) \vdash sf a \quad sfpc(D_i+d_i,.), a1(D_i+d_i+d_1,.) \otimes d(D_i+d_i+d_1,.), t_2, 3, 5, 7, 9, 10, 11, 12, 14, 16, 17, o \vdash o}{\multimap_L} \\
\frac{\quad}{sf a(D_i+d_i,.), sfpc(D_i+d_i,.), sf a \multimap a1 \otimes d, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, t_o \vdash o} \otimes_L \\
\frac{i(D_i, D_i+d_i) \vdash i \quad sf a(D_i+d_i,.) \otimes sfpc(D_i+d_i,.), t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, t_o \vdash o}{\multimap_L} \\
\frac{\quad}{i(D_i,.), i \multimap sf a \otimes sfpc, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, t_o \vdash o}
\end{array}$$

A árvore de prova da Lógica Linear com cálculo de datas, para o cenário $Sc_3A \times Sc_1PC$, onde $Seq = D_i + d_i + d_1 + d_{10} + d_{11}$, $Seq1 = \max\{d_2, d_{12} + d_{14}\} + d_3$ e $Seq2 = d_{12} + d_{14} + d_{15}$, é dada por:

$$\begin{array}{c}
\frac{\frac{efa(Seq+\max\{Seq1,Seq2\}+d_6,Seq+\max\{Seq1,Seq2\}+d_6+d_o) \vdash efa \quad efp(Seq+Seq2,Seq+\max\{Seq1,Seq2\}+d_6+d_o) \vdash efp}{efa(Seq+\max\{Seq1,Seq2\}+d_6,Seq+\max\{Seq1,Seq2\}+d_6+d_o),efp(Seq+Seq2,Seq+\max\{Seq1,Seq2\}+d_6+d_o) \vdash efa \otimes efp} \otimes_R \quad o(Seq+\max\{Seq1,Seq2\}+d_6+d_o,.) \vdash o}{\vdash o} \multimap_L \\
\frac{\frac{a3(Seq+Seq1,Seq+\max\{Seq1,Seq2\}+d_6) \vdash a3 \quad tl(Seq+Seq2,Seq+\max\{Seq1,Seq2\}+d_6) \vdash tl}{a3(Seq+Seq1,Seq+\max\{Seq1,Seq2\}+d_6),tl(Seq+Seq2,Seq+\max\{Seq1,Seq2\}+d_6) \vdash a3 \otimes tl} \otimes_R \quad efp(Seq+Seq2,.) , efa(Seq+\max\{Seq1,Seq2\}+d_6,.) , efa \otimes efp \vdash o \vdash o}{\vdash o} \multimap_L \\
\frac{\vdash o}{a3(Seq+Seq1,.) , tl(Seq+Seq2,.) , efp(Seq+Seq2,.) , a3 \otimes tl \vdash o} \otimes_L \\
\frac{\vdash o}{p4(Seq+d_{12}+d_{14},Seq+Seq2) \vdash p4 \quad a3(Seq+Seq1,.) , tl(Seq+Seq2,.) \otimes efp(Seq+Seq2,.) , t_6, o \vdash o} \multimap_L \\
\frac{\frac{a2(Seq+d_2,Seq+Seq1) \vdash a2 \quad a(Seq+d_{12}+d_{14},Seq+Seq1) \vdash a}{a2(Seq+d_2,Seq+Seq1),a(Seq+d_{12}+d_{14},Seq+Seq1) \vdash a2 \otimes a} \otimes_R \quad p4(Seq+d_{12}+d_{14},.) , a3(Seq+Seq1,.) , p4 \vdash o \otimes efp, t_6, o \vdash o}{\vdash o} \multimap_L \\
\frac{\vdash o}{a2(Seq+d_2,.) , a(Seq+d_{12}+d_{14},.) , p4(Seq+d_{12}+d_{14},.) , a2 \otimes a \vdash o} \otimes_L \\
\frac{\vdash o}{p3(Seq+d_{12},Seq+d_{12}+d_{14}) \vdash p3 \quad a2(Seq+d_2,.) , a(Seq+d_{12}+d_{14},.) \otimes p4(Seq+d_{12}+d_{14},.) , t_3, t_6, t_{15}, o \vdash o} \multimap_L \\
\frac{\vdash o}{p2(Seq,Seq+d_{12}) \vdash p2 \quad a2(Seq+d_2,.) , p3(Seq+d_{12},.) , p3 \vdash o \otimes p4, t_3, t_6, t_{15}, o \vdash o} \multimap_L \\
\frac{\frac{a1(D_i+d_i+d_1,Seq+d_2) \vdash a1 \quad ad(Seq,Seq+d_2) \vdash ad}{a1(D_i+d_i+d_1,Seq+d_2),ad(Seq,Seq+d_2) \vdash a1 \otimes ad} \otimes_R \quad p2(Seq,.) , a2(Seq+d_2,.) , p2 \vdash o \otimes p3, t_3, t_6, t_{15}, o \vdash o}{\vdash o} \multimap_L \\
\frac{\vdash o}{a1(D_i+d_i+d_1,.) , ad(Seq,.) , p2(Seq,.) , a1 \otimes ad \vdash o} \otimes_L \\
\frac{\vdash o}{p1(D_i+d_i+d_1+d_{10},Seq) \vdash p1 \quad a1(D_i+d_i+d_1,.) , ad(Seq,.) \otimes p2(Seq,.) , t_2, t_3, t_6, t_{12}, t_{14}, t_{15}, o \vdash o} \multimap_L \\
\frac{\frac{\frac{sfpc(D_i+d_i,D_i+d_i+d_1+d_{10}) \vdash sfpc \quad d(D_i+d_i+d_1,D_i+d_i+d_1+d_{10}) \vdash d}{sfpc(D_i+d_i,D_i+d_i+d_1+d_{10}),d(D_i+d_i+d_1,D_i+d_i+d_1+d_{10}) \vdash sfpc \otimes d} \otimes_R \quad a1(D_i+d_i+d_1,.) , p1(D_i+d_i+d_1+d_{10},.) , p1 \vdash o \otimes p2, t_2, t_3, t_6, t_{12}, t_{14}, t_{15}, o \vdash o}{\vdash o} \multimap_L \\
\frac{\vdash o}{sfpc(D_i+d_i,.) , a1(D_i+d_i+d_1,.) , d(D_i+d_i+d_1,.) , sfpc \otimes d \vdash o} \otimes_L \\
\frac{\vdash o}{sfa(D_i+d_i,D_i+d_i+d_1) \vdash sfa \quad sfpc(D_i+d_i,.) , a1(D_i+d_i+d_1,.) \otimes d(D_i+d_i+d_1,.) , t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, o \vdash o} \multimap_L \\
\frac{\vdash o}{sfa(D_i+d_i,.) , sfpc(D_i+d_i,.) , sfa \vdash o \otimes d, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, o \vdash o} \otimes_L \\
\frac{\vdash o}{i(D_i,D_i+d_i) \vdash i \quad sfa(D_i+d_i,.) \otimes sfpc(D_i+d_i,.) , t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, o \vdash o} \multimap_L \\
\frac{\vdash o}{i(D_i,.) , i \vdash o \otimes sfa \otimes sfpc, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, o \vdash o}
\end{array}$$

A árvore de prova da Lógica Linear com cálculo de datas, para o cenário $Sc_4A \times Sc_1PC$, onde $Seq = D_i + d_i + d_1 + d_{10} + d_{11}$, $Seq1 = \max\{d_2, d_{12} + d_{14}\} + d_3$ e $Seq2 = d_{12} + d_{14} + d_{15}$, é apresentada na sequência.

$$\begin{array}{c}
\frac{\frac{efa(Seq + \max\{Seq1 + d_5, Seq2\} + d_8, Seq + \max\{Seq1 + d_5, Seq2\} + d_8 + d_o) \vdash efa \quad efp(Seq + Seq2, Seq + \max\{Seq1 + d_5, Seq2\} + d_8 + d_o) \vdash efp}{efa(Seq + \max\{Seq1 + d_5, Seq2\} + d_8, Seq + \max\{Seq1 + d_5, Seq2\} + d_8 + d_o) \vdash efa \otimes efp} \otimes_R \quad o(Seq + \max\{Seq1 + d_5, Seq2\} + d_8 + d_o) \vdash o}{\frac{a4(Seq + Seq1 + d_5, Seq + \max\{Seq1 + d_5, Seq2\} + d_8) \vdash a4 \quad tl(Seq + Seq2, Seq + \max\{Seq1 + d_5, Seq2\} + d_8) \vdash tl}{a4(Seq + Seq1 + d_5, Seq + \max\{Seq1 + d_5, Seq2\} + d_8), tl(Seq + Seq2, Seq + \max\{Seq1 + d_5, Seq2\} + d_8) \vdash a3 \otimes tl} \otimes_R \quad efp(Seq + Seq2, .), efa(Seq + \max\{Seq1 + d_5, Seq2\} + d_8, .), efa \otimes efp \multimap o \vdash o} \multimap_L} \\
\frac{a3(Seq + Seq1, Seq + Seq1 + d_5) \vdash a3 \quad tl(Seq + Seq2, .), efp(Seq + Seq2, .), a4(Seq + Seq1 + d_5, .), a4 \otimes tl \multimap efa, t_o \vdash o}{a3(Seq + Seq1, .), tl(Seq + Seq2, .), efp(Seq + Seq2, .), a3 \multimap a4, t_8, o \vdash o} \otimes_L} \\
\frac{p4(Seq + d_{12} + d_{14}, Seq + Seq2) \vdash p4 \quad a3(Seq + Seq1, .), tl(Seq + Seq2, .) \otimes efp(Seq + Seq2, .), t_5, s, o \vdash o}{a2(Seq + d_2, Seq + Seq1) \vdash a2 \quad a(Seq + d_{12} + d_{14}, Seq + Seq1) \vdash a}{a2(Seq + d_2, Seq + Seq1), a(Seq + d_{12} + d_{14}, Seq + Seq1) \vdash a2 \otimes a} \otimes_R \quad p4(Seq + d_{12} + d_{14}, .), a3(Seq + Seq1, .), p4 \multimap tl \otimes efp, t_5, s, o \vdash o} \multimap_L} \\
\frac{a2(Seq + d_2, .), a(Seq + d_{12} + d_{14}, .), p4(Seq + d_{12} + d_{14}, .), a2 \otimes a \multimap a3, t_5, s, 15, o \vdash o}{p3(Seq + d_{12}, Seq + d_{12} + d_{14}) \vdash p3 \quad a2(Seq + d_2, .), a(Seq + d_{12} + d_{14}, .) \otimes p4(Seq + d_{12} + d_{14}, .), t_3, s, 5, 8, 15, o \vdash o} \multimap_L} \\
\frac{p2(Seq, Seq + d_{12}) \vdash p2 \quad a2(Seq + d_2, .), p3(Seq + d_{12}, .), p3 \multimap a \otimes p4, t_3, s, 5, 8, 15, o \vdash o}{\frac{a1(D_i + d_i + d_1, Seq + d_2) \vdash a1 \quad ad(Seq, Seq + d_2) \vdash ad}{a1(D_i + d_i + d_1, Seq + d_2), ad(Seq, Seq + d_2) \vdash a1 \otimes ad} \otimes_R \quad p2(Seq, .), a2(Seq + d_2, .), p2 \multimap p3, t_3, s, 5, 8, 14, 15, o \vdash o} \multimap_L} \\
\frac{a1(D_i + d_i + d_1, .), ad(Seq, .), p2(Seq, .), a1 \otimes ad \multimap a2, t_3, s, 5, 8, 12, 14, 15, o \vdash o}{p1(D_i + d_i + d_1 + d_{10}, Seq) \vdash p1 \quad a1(D_i + d_i + d_1, .), ad(Seq, .) \otimes p2(Seq, .), t_2, s, 5, 8, 12, 14, 15, o \vdash o} \multimap_L} \\
\frac{\frac{sfpc(D_i + d_i, D_i + d_i + d_1 + d_{10}) \vdash sfpc \quad d(D_i + d_i + d_1, D_i + d_i + d_1 + d_{10}) \vdash d}{sfpc(D_i + d_i, D_i + d_i + d_1 + d_{10}), d(D_i + d_i + d_1, D_i + d_i + d_1 + d_{10}) \vdash sfpc \otimes d} \otimes_R \quad a1(D_i + d_i + d_1, .), p1(D_i + d_i + d_1 + d_{10}, .), p1 \multimap ad \otimes p2, t_2, s, 5, 8, 12, 14, 15, o \vdash o} \multimap_L} \\
\frac{sfpc(D_i + d_i, .), a1(D_i + d_i + d_1, .), d(D_i + d_i + d_1, .), sfpc \otimes d \multimap p1, t_2, s, 5, 8, 11, 12, 14, 15, o \vdash o}{sfa(D_i + d_i, D_i + d_i + d_1) \vdash sfa \quad sfpc(D_i + d_i, .), a1(D_i + d_i + d_1, .) \otimes d(D_i + d_i + d_1, .), t_2, s, 5, 8, 10, 11, 12, 14, 15, o \vdash o} \multimap_L} \\
\frac{sfa(D_i + d_i, .), sfpc(D_i + d_i, .), sfa \multimap a1 \otimes d, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o}{i(D_i, D_i + d_i) \vdash i \quad sfa(D_i + d_i, .) \otimes sfpc(D_i + d_i, .), t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o} \multimap_L} \\
i(D_i, .), i \multimap sfa \otimes sfpc, t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o
\end{array}$$

A.5 Árvores de Prova da Lógica Linear com Cálculo de Datas para Monitoração de Cenários Livres de *Deadlock*

Este apêndice apresenta as árvores de prova da Lógica Linear construídas no contexto da monitoração de cenários livres de *deadlock*, apresentado na seção 5.2.

É importante lembrar que, considerando o sistema composto apresentado na Figura 22, cada transição é representada por:

$$t_i = i \multimap q0 \otimes p0,$$

$$t_1 = q0 \otimes ticket \multimap q1,$$

$$t_2 = q1 \multimap VISA \otimes q2,$$

$$t_3 = q1 \multimap eCash \otimes q2,$$

$$t_4 = p0 \multimap ticket \otimes p1,$$

$$t_5 = p0 \multimap ticket \otimes p2,$$

$$t_6 = p1 \otimes eCash \multimap p3,$$

$$t_7 = p2 \otimes VISA \multimap p3,$$

$$t_o = q2 \otimes p3 \multimap o.$$

Na sequência são apresentadas as árvores de prova da Lógica Linear com cálculo de datas para os cenários livres de *deadlock* do sistema composto apresentado na Figura 22.

A árvore de prova com cálculo de datas para o cenário $Sc_1C1 \oplus T1$ (ou $Sc_2C1 \times Sc_1T1$) é como se segue:

$$\begin{array}{c}
 \frac{\frac{q2(D_i+d_i+d_4+d_1+d_3, D_i+d_i+d_4+d_1+d_3+d_6+d_o) \vdash q2 \quad p3(D_i+d_i+d_4+d_1+d_3+d_6, D_i+d_i+d_4+d_1+d_3+d_6+d_o) \vdash p3}{q2(D_i+d_i+d_4+d_1+d_3, D_i+d_i+d_4+d_1+d_3+d_6+d_o), p3(D_i+d_i+d_4+d_1+d_3+d_6, D_i+d_i+d_4+d_1+d_3+d_6+d_o) \vdash q2 \otimes p3} \otimes_R \quad o(D_i+d_i+d_4+d_1+d_3+d_6+d_o,.) \vdash o}{\frac{p1(D_i+d_i+d_4, D_i+d_i+d_4+d_1+d_3+d_6) \vdash p1 \quad eCash(D_i+d_i+d_4+d_1+d_3, D_i+d_i+d_4+d_1+d_3+d_6) \vdash eCash}{p1(D_i+d_i+d_4, D_i+d_i+d_4+d_1+d_3+d_6), eCash(D_i+d_i+d_4+d_1+d_3, D_i+d_i+d_4+d_1+d_3+d_6) \vdash p1 \otimes eCash} \otimes_R \quad q2(D_i+d_i+d_4+d_1+d_3,.), p3(D_i+d_i+d_4+d_1+d_3+d_6,.), q2 \otimes p3 \multimap o \vdash o}{\frac{p1(D_i+d_i+d_4,.) \vdash p1 \quad eCash(D_i+d_i+d_4+d_1+d_3,.) \vdash eCash}{p1(D_i+d_i+d_4,.) \vdash p1 \otimes eCash} \otimes_R \quad q2(D_i+d_i+d_4+d_1+d_3,.) \vdash q2 \otimes p3 \multimap o \vdash o}{\frac{q1(D_i+d_i+d_4+d_1, D_i+d_i+d_4+d_1+d_3) \vdash q1 \quad p1(D_i+d_i+d_4,.) \vdash p1 \otimes eCash(D_i+d_i+d_4+d_1+d_3,.) \vdash eCash}{q1(D_i+d_i+d_4+d_1,.) \vdash q1 \otimes p1 \otimes eCash(D_i+d_i+d_4+d_1+d_3,.) \vdash eCash} \otimes_R \quad q2(D_i+d_i+d_4+d_1+d_3,.) \vdash q2 \otimes p3 \multimap o \vdash o}{\frac{q0(D_i+d_i, D_i+d_i+d_4+d_1) \vdash q0 \quad ticket(D_i+d_i+d_4, D_i+d_i+d_4+d_1) \vdash ticket}{q0(D_i+d_i, D_i+d_i+d_4+d_1), ticket(D_i+d_i+d_4, D_i+d_i+d_4+d_1) \vdash q0 \otimes ticket} \otimes_R \quad p1(D_i+d_i+d_4,.) \vdash p1 \otimes eCash(D_i+d_i+d_4+d_1+d_3,.) \vdash eCash}{\frac{q0(D_i+d_i,.) \vdash q0 \quad ticket(D_i+d_i+d_4,.) \vdash ticket}{q0(D_i+d_i,.) \vdash q0 \otimes ticket} \otimes_R \quad p1(D_i+d_i+d_4,.) \vdash p1 \otimes eCash(D_i+d_i+d_4+d_1+d_3,.) \vdash eCash}{\frac{p0(D_i+d_i, D_i+d_i+d_4) \vdash p0 \quad q0(D_i+d_i,.) \vdash q0 \otimes ticket(D_i+d_i+d_4,.) \vdash ticket}{p0(D_i+d_i, D_i+d_i+d_4) \vdash p0 \otimes ticket(D_i+d_i+d_4,.) \vdash ticket} \otimes_R \quad p1(D_i+d_i+d_4,.) \vdash p1 \otimes eCash(D_i+d_i+d_4+d_1+d_3,.) \vdash eCash}{\frac{q0(D_i+d_i,.) \vdash q0 \quad p0(D_i+d_i,.) \vdash p0 \otimes ticket(D_i+d_i+d_4,.) \vdash ticket}{q0(D_i+d_i,.) \vdash q0 \otimes p0 \otimes ticket(D_i+d_i+d_4,.) \vdash ticket} \otimes_R \quad p1(D_i+d_i+d_4,.) \vdash p1 \otimes eCash(D_i+d_i+d_4+d_1+d_3,.) \vdash eCash}{\frac{i(D_i, D_i+d_i) \vdash i \quad q0(d_i+d_i,.) \vdash q0 \otimes p0 \otimes ticket(D_i+d_i+d_4,.) \vdash ticket}{i(D_i, D_i+d_i) \vdash i \otimes q0 \otimes p0 \otimes ticket(D_i+d_i+d_4,.) \vdash ticket} \otimes_R \quad p1(D_i+d_i+d_4,.) \vdash p1 \otimes eCash(D_i+d_i+d_4+d_1+d_3,.) \vdash eCash}{i(D_i,.) \vdash i \otimes q0 \otimes p0 \otimes ticket(D_i+d_i+d_4,.) \vdash ticket} \otimes_R \quad p1(D_i+d_i+d_4,.) \vdash p1 \otimes eCash(D_i+d_i+d_4+d_1+d_3,.) \vdash eCash} \otimes_L
 \end{array}$$

A árvore de prova com cálculo de datas para o cenário $Sc_2C1 \oplus T1$ (ou $Sc_1C1 \times Sc_2T1$) é dada por:

$$\begin{array}{c}
\frac{\frac{q2(D_i+d_i+d_5+d_1+d_2, D_i+d_i+d_5+d_1+d_2+d_7+d_o) \vdash q2 \quad p3(D_i+d_i+d_5+d_1+d_2+d_7, D_i+d_i+d_5+d_1+d_2+d_7+d_o) \vdash p3}{q2(D_i+d_i+d_5+d_1+d_2, D_i+d_i+d_5+d_1+d_2+d_7+d_o), p3(D_i+d_i+d_5+d_1+d_2+d_7, D_i+d_i+d_5+d_1+d_2+d_7+d_o) \vdash q2 \otimes p3} \otimes_R \quad o(D_i+d_i+d_5+d_1+d_2+d_7+d_o, \cdot) \vdash o}{\frac{p2(D_i+d_i+d_5, D_i+d_i+d_5+d_1+d_2+d_7) \vdash p2 \quad VISA(D_i+d_i+d_5+d_1+d_2, D_i+d_i+d_5+d_1+d_2+d_7) \vdash VISA}{p2(D_i+d_i+d_5, D_i+d_i+d_5+d_1+d_2+d_7), VISA(D_i+d_i+d_5+d_1+d_2, D_i+d_i+d_5+d_1+d_2+d_7) \vdash p2 \otimes VISA} \otimes_R \quad q2(D_i+d_i+d_5+d_1+d_2, \cdot), p3(D_i+d_i+d_5+d_1+d_2+d_7, \cdot), q2 \otimes p3 \multimap o \vdash o}{\frac{p2(D_i+d_i+d_5, \cdot), VISA(D_i+d_i+d_5+d_1+d_2, \cdot), q2(D_i+d_i+d_5+d_1+d_2, \cdot), p2 \otimes VISA \multimap p3, t_o \vdash o}{\otimes_L} \\
\frac{q1(D_i+d_i+d_5+d_1, D_i+d_i+d_5+d_1+d_2) \vdash q1 \quad p2(D_i+d_i+d_5, \cdot), VISA(D_i+d_i+d_5+d_1+d_2, \cdot) \otimes q2(D_i+d_i+d_5+d_1+d_2, \cdot), t_7, t_o \vdash o}{\multimap_L} \\
\frac{\frac{q0(D_i+d_i, D_i+d_i+d_5+d_1) \vdash q0 \quad ticket(D_i+d_i+d_5, D_i+d_i+d_5+d_1) \vdash ticket}{q0(D_i+d_i, D_i+d_i+d_5+d_1), ticket(D_i+d_i+d_5, D_i+d_i+d_5+d_1) \vdash q0 \otimes ticket} \otimes_R \quad p2(D_i+d_i+d_5, \cdot), q1(D_i+d_i+d_5+d_1, \cdot), q1 \multimap VISA \otimes q2, t_7, t_o \vdash o}{\multimap_L} \\
\frac{q0(D_i+d_i, \cdot), ticket(D_i+d_i+d_5, \cdot), p2(D_i+d_i+d_5, \cdot), q0 \otimes ticket \multimap q1, t_2, t_7, t_o \vdash o}{\otimes_L} \\
\frac{p0(D_i+d_i, D_i+d_i+d_5) \vdash p0 \quad q0(D_i+d_i, \cdot), ticket(D_i+d_i+d_5, \cdot) \otimes p2(D_i+d_i+d_5, \cdot), t_1, t_2, t_7, t_o \vdash o}{\multimap_L} \\
\frac{q0(D_i+d_i, \cdot), p0(D_i+d_i, \cdot), t_1, t_2, p0 \multimap ticket \otimes p2, t_7, t_o \vdash o}{\otimes_L} \\
\frac{i(D_i, D_i+d_i) \vdash i \quad q0(d_i+d_i, \cdot) \otimes p0(D_i+d_i), t_1, t_2, t_5, t_7, t_o \vdash o}{\multimap_L} \\
i(D_i, \cdot), i \multimap q0 \otimes p0, t_1, t_2, t_5, t_7, t_o \vdash o
\end{array}$$