EasyChair Preprint

№ 15155

Introducing New Definitions of Systems Language Opacity for Discrete Events Modeled by a Class of Timed Automata

Mariana Marques, Raphael Barcelos and João Carlos Basilio

September 28, 2024

Introduzindo novas definições de opacidade de linguagem de sistemas a eventos discretos modelados por uma classe de autômatos temporizados ⋆

Mariana Guimarães Marques * **Raphael Julio Barcelos** *
João Carlos dos Santos Basilio *

* *Universidade Federal do Rio de Janeiro, RJ, Brasil, (e-mail: mariana.marques@coppe.ufrj.br; raphael.barcelos@poli.ufrj.br;basilio@dee.ufrj.br).*

Abstract: This paper addresses the problem of language opacity for a class of discrete-event systems whose transitions occur within some known time interval. For this purpose, a class of timed automata is proposed, time-interval automata (TIA), in which the elapsed time is associated with a global clock, allowing each transition to be associated with a time interval of occurrence. Procedures to do operations with TIA and language-based definition of opacity notions for this class of timed automata, have been presented in previous works as well as an algorithm for its verification. In this paper, we improve these notions of timed language-based opacity (TLBO), in order to capture more nuances that appear when time is taken into account.

Resumo: Este artigo aborda o problema de opacidade de linguagem para uma classe de sistemas a eventos discretos cujas transições ocorrem em um tempo específico contido em um intervalo de tempo conhecido. Para tanto, é utilizada uma classe de autômatos temporizados, os chamados autômatos temporizados por intervalos (ATI), nos quais um intervalo de tempo é associado a cada evento que rotula uma transição. Em artigos anteriores foram apresentados novos procedimentos para a realização de operações com ATIs, a definição usual de opacidade com base em linguagem foi estendida para esta classe de autômatos, foram apresentadas condições necessárias e suficientes para opacidade com base em linguagem temporizada (TLBO, do inglês *Timed Language-Based Opacity*) e um algoritmo para sua verificação foi proposto. Neste artigo, a definição existente de TLBO é expandida com o objetivo de captar mais nuances relacionadas ao modelo temporizado.

Keywords: Discrete event systems; opacity; timed automata; time-interval automata.

Palavras-chaves: Sistemas a eventos discretos; opacidade; autômatos temporizados; autômatos temporizados com intervalo de tempo.

1. INTRODUÇÃO

Um Sistema a Eventos Discretos (SED) é dito opaco se um observador externo, considerado um intruso com intenções maliciosas, for incapaz de inferir o comportamento secreto do sistema (Mazaré, 2004). Considera-se que o intruso possui conhecimento completo da planta, ou seja, ele consegue inferir o comportamento secreto apenas observando o fluxo de informações.

Existem diversas noções de opacidade, cada uma definida de acordo com a natureza do comportamento secreto. As mais utilizadas são opacidade de linguagem (Lin, 2011), de estado atual (Saboori e Hadjicostis, 2007), de estado inicial, *k-step* e *infinite-step* (Jacob et al., 2016; Lafortune et al., 2018). Transformações de complexidade polinomial entre essas noções são definidas em Wu e Lafortune (2013) e Balun e Masopust (2021).

Neste artigo é considerada a opacidade com base em linguagem para uma nova classe de autômatos abstraída a partir do modelo original e de informações sobre o tempo em que os eventos ocorrem. Mesmo que os eventos continuem sendo gerados espontaneamente, é esperado que ocorram dentro de um intervalo de tempo. Em Alves et al. (2020) e Viana et al. (2021) considera-se um tempo mínimo no qual cada transição pode ocorrer; contudo, não há uma limitação para um tempo máximo para ocorrência dos eventos. Em Marques e Basilio (2022) e Marques et al. (2023) foi usada uma definição similar à proposta em Wang et al. (2018), chamados de autômatos temporizados por intervalos (ATIs), onde há um *clock* que é reinicializado a cada transição cujos tempos mínimo e máximo para a ocorrência dos eventos são definidos pelos limites inferior e superior de um intervalo real. Neste contexto, as operações padrões envolvendo autômatos (Cassandras e Lafortune, 2008) foram estendidas em Marques e Basilio (2022) e

⋆ Este trabalho foi, em parte, financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), código de financiamento 001, pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processo número 316881/2021-0.

Marques et al. (2023) para essa nova classe de SEDs temporizados, cuja metodologia das operações envolvendo os autômatos temporizados é diferente daquela proposta em Wang et al. (2018). Autômatos temporizados com guardas que definem um limite inferior e superior para o tempo de ocorrência das transições são usados em Ammar et al. (2021) e Schneider et al. (2012), e se baseiam no modelo definido emAlur e Dill (1994).

Visto que as três noções de opacidade com base em linguagens para autômatos temporizados por intervalo (fortemente opaco, fracamente opaco, não opaco) abordadas em Marques e Basilio (2022) e Marques et al. (2023) classificam comportamentos diferentes com a mesma noção, neste trabalho são propostas cinco novas noções de opacidade para ATI, uma vez que a distinção de sequências secretas de não secretas pode ocorrer observando-se os sinais ou o tempo em que eles ocorrem.

Este artigo está estruturado da seguinte forma. Na seção 2 são revistos os conceitos fundamentais de ATIs necessários para tornar o artigo autocontido. Nas seções 3 e 4 é definida a opacidade com base em linguagem temporizada e é proposto um método para sua verificação em autômatos temporizados por intervalos, respectivamente. A seção 5 apresenta um exemplo. Por fim, a seção 6 resume a contribuição do artigo.

2. FUNDAMENTOS TEÓRICOS DE AUTÔMATO TEMPORIZADO POR INTERVALOS

Neste trabalho será considerada uma classe de autômatos temporizados cujos disparos ocorrem dentro de um intervalo de tempo, como definido em Marques e Basilio (2022) e Marques et al. (2023).

Definição *1.* (Autômato temporizado por intervalos). Um autômato temporizado por intervalos (ATI) é uma sêxtupla $G_T = (X, \Sigma, f, x_0, X_m, \mu)$ na qual X é o conjunto de estados, Σ é o conjunto de eventos, $f : X \times \Sigma \to 2^X$ é a função de transição, x_0 é o estado inicial e X_m é o conjunto de estados marcados. A função $\mu : X \times \Sigma \times X \to 2^{\mathbb{R}_+}$ é a função de rotulamento, de forma que para um par de estados $(x, y) \in X \times X$ e $\sigma \in \Sigma$, $\mu(x, \sigma, y) = I \subseteq 2^{\mathbb{R}_+}$, se $f(x, \sigma) = y$, e não definida caso contrário.

De acordo com a definição 1, é associado um intervalo de tempo I, medido em unidade de tempo (UT), a cada transição $y = f(x, \sigma)$, que determina o intervalo no qual se espera que o evento σ aconteça após o sistema ter alcançado o estado x. Note que é possível que existam transições com o mesmo evento partindo de um mesmo estado $x \in X$, mas rotuladas por intervalos distintos e que levem a estados diferentes de G_T.

Assim como nos autômatos não temporizados, define-se o conjunto de eventos ativos $\Gamma(x) = \{\sigma \in \Sigma | \exists y \in X, f(x, \sigma) = y\} \subseteq 2^{\Sigma}$. Ao longo do texto, denota-se por Δ o conjunto de transições, definido como $\Delta = \{(x, \sigma, y) \in (X \times \Sigma \times X) | f(x, \sigma) = y\}$. Dado um estado $x \in X$, define-se o conjunto formado por eventos que rotulam transições que chegam em x como $\Upsilon(x) = \{\sigma \in \Sigma | \exists x' \in X, f(x', \sigma) = x\}$. As definições padrões de acessibilidade e coacessibilidade (Cassandras e Lafortune, 2008) são mantidas para ATIs.

Seja $\sigma^T = (\sigma, I) \in \Sigma \times 2^{\mathbb{R}_+}$ um evento temporizado. Define-se Δ_T, como o conjunto de transições temporizadas $\Delta_T = \{(x, \sigma^T, y) \in X \times \Sigma \times 2^{\mathbb{R}_+} \times X | (\sigma^T = (\sigma, I)) \wedge ((x, \sigma, y) \in \Delta) \wedge (\mu(x, \sigma, y) = I)\}$. A função de eventos ativos pode ser estendida para $\Gamma_T(x) = \{\sigma^T \in \Sigma \times 2^{\mathbb{R}_+} | (x, \sigma^T, y) \in \Delta_T\}$, e portanto $\Delta_T(x) = \{(x, \sigma^T, y) \in \Delta_T | \sigma^T \in \Gamma_T(x)\}$.

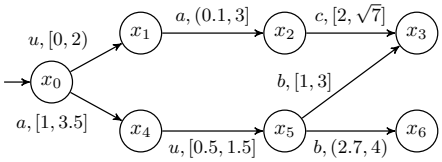


Figura 1. Autômato temporizado G_T.

Um ATI G_T é determinístico se as seguintes condições forem satisfeitas: *(i)* $X_0 = \{x_0\}$; *(ii)* não há transições rotuladas por ε; *(iii)* para todo $\delta \in \Delta_T(x)$, se existirem $\delta_1 = (x, \sigma_1^T, x_1)$ e $\delta_2 = (x, \sigma_2^T, x_2)$ tais que $(x_1 \neq x_2) \wedge (\sigma_1^T = (\sigma, I_1) \wedge \sigma_2^T = (\sigma, I_2))$, então $I_1 \cap I_2 = \emptyset$. Quando pelos menos uma das condições *(i)–(iii)* acima for violada, o ATI é não determinístico. Neste caso, a palavra nula ε deverá ser acrescentada ao conjunto dos eventos de G_T.

Assim como para os autômatos não temporizados, pode-se estender o domínio de f recursivamente para $X \times \Sigma^*$, de forma que $f(x, \epsilon) = x$ e $f(x, s\sigma) = f(f(x, s), \sigma)$, para $s \in \Sigma^*$ e $\sigma \in \Sigma$. Por convenção, a transição (x, ϵ, x) é associada ao intervalo $\emptyset$.

Dada uma sequência temporizada $s = (\sigma_1, I_1) \dots (\sigma_n, I_n)$ em $(\Sigma \times 2^{\mathbb{R}_+})^*$, pode-se recuperar a sequência original de eventos em Σ^* por meio da função $\pi_\Sigma(s) = \sigma_1 \sigma_2 \dots \sigma_n$ e seus intervalos de tempo por $\pi_T(s) = I_1 I_2 \dots I_n$. O i-ésimo evento de $\pi_\Sigma(s)$ e o i-ésimo intervalo de $\pi_T(s)$ são definidos como $\pi_\Sigma^i(s) = \sigma_i$ e $\pi_T^i(s) = I_i$, respectivamente. Dado um conjunto de eventos Σ, o fecho de Kleene para um conjunto de eventos temporizados $E \subseteq \Sigma \times 2^{\mathbb{R}_+}$ é definido como $E^* = \{(\varepsilon, \emptyset)\} \cup E \cup EE \cup \dots$. A linguagem temporizada gerada por G_T é definida por $\mathcal{L}_T(G_T) = \{s_t = (\sigma_1, I_1) \dots (\sigma_n, I_n) \in (\Sigma \times 2^{\mathbb{R}_+})^* | (f(x_0, \sigma_1 \dots \sigma_n)!) \wedge (I_i = \mu(x_{i-1}, \sigma_i, x_i), i = 1, \dots, n) \wedge (x_0 \in X_0) \wedge (x_i = f(x_{i-1}, \sigma_i), i = 1, \dots, n)\} \cup \{(\epsilon, \emptyset)\}$. A linguagem temporizada marcada de G_T é dada por $\mathcal{L}_{mT}(G_T) = \{s \in \mathcal{L}_T(G_T) | f(x_0, \pi_\Sigma(s)) \in X_m\}$. A linguagem não temporizada gerada (ou simplesmente linguagem gerada) por G_T é definida como $\mathcal{L}(G_T) = \{\pi(s), \forall s \in \mathcal{L}_T(G_T)\}$ e a linguagem marcada não temporizada é $\mathcal{L}_m(G_T) = \{\pi_\Sigma(s), \forall s \in \mathcal{L}_{mT}(G_T)\}$.

Exemplo *1.* No autômato da figura 1, o conjunto de estados é $X = \{x_0, x_1, x_2, x_3, x_4, x_5, x_6\}$, $X_0 = \{x_0\}$, o conjunto dos eventos $\Sigma = \{u, a, b, c\}$. Os intervalos de tempo atribuídos a cada transição estão especificados na figura 1 ao lado de seu evento correspondente. Por exemplo, na transição $\delta = (x_1, a, x_2)$, o evento a acontecerá entre 0.1 (aberto) e 3 UTs, representado na figura como $a, (0.1, 3]$. A linguagem temporizada gerada por G_T é igual a $\mathcal{L}_T(G_T) = \{(\varepsilon, [0, 0]); (u, [0, 2)); (a, [1, 3.5]); (u, [0, 2))(a, (0.1, 3]); (a, [1, 3.5])\ (u, [0.5, 1.5]); (u, [0, 2))(a, (0.1, 3])(c, [2, \sqrt{7}]); (a, [1, 3.5])(u, [0.5, 1.5])(b, (2.7, 4)); (a, [1, 3.5])(u, [0.5, 1.5])\ (b, [1, 3])\}$.

Definição *2.* (Interseção de sequências temporizadas). Sejam $s_1 = (\sigma_1, I_1)(\sigma_2, I_2) \dots (\sigma_n, I_n)$ e $s_2 = (\sigma_1', I_1')(\sigma_2', I_2') \dots (\sigma_n', I_n')$ duas sequências temporizadas definidas em $(\Sigma \times 2^{\mathbb{R}_+})^*$. A interseção entre s_1 e s_2 é definida por: $s_1 \cap_T s_2 = (\sigma_1, I_1 \cap I_1')(\sigma_2, I_2 \cap I_2') \dots (\sigma_n, I_n \cap I_n')$, se $(\pi_\Sigma(s_1) =$

$\pi_\Sigma(s_2)) \wedge (I_i \cap I_i' \neq \emptyset, i = 1, 2, \ldots, n)$; e não definida, caso contrário.

Definição *3.* (Interseção entre linguagens temporizadas). Dadas duas linguagens temporizadas L_{T_1} e L_{T_2}, sua interseção é definida como $L_1 \cap_T L_2 = \{s \in (\Sigma \times 2^{\mathbb{R}_+})^* | (\exists u \in L_{T_1}) \wedge (\exists v \in L_{T_2}), s = u \cap_T v\}$.

Definição *4.* (Soma de intervalos de tempo). A soma de n intervalos $I_i = [l_i, u_i]$, tais que $l_i, u_i \in \mathbb{R}_+$ para $i = 1, \ldots, n$, é definida como $\sum_{i=1}^{n} I_i = [\sum_{i=1}^{n} l_i, \sum_{i=1}^{n} u_i]$. Se alguma das extremidades de algum I_i for aberta, esta mesma extremidade também será aberta em $\sum_{i=1}^{n} I_i$.

Definição *5.* (Projeção de sequência temporizada). Seja $\Sigma_s \subseteq \Sigma_l$ e $\Sigma_\varepsilon = (\Sigma_l \setminus \Sigma_s) \cup \{\varepsilon\}$. Então, a projeção de uma sequência $s \in (\Sigma_l \times 2^{\mathbb{R}_+})^*$ é o mapeamento $P_T : ((\Sigma_l \cup \{\varepsilon\}) \times 2^{\mathbb{R}_+})^* \rightarrow ((\Sigma_s \cup \{\varepsilon\}) \times 2^{\mathbb{R}_+})^*$ definido recursivamente como: *(i)* $P_T((\varepsilon, I)) = (\varepsilon, I)$; *(ii)* $P_T((\sigma, I)) = (\sigma, I)$, se $\sigma \in \Sigma_s$, ou $P_T((\sigma, I)) = (\varepsilon, I)$, se $\sigma \in \Sigma_\varepsilon$; *(iii)* $P_T((\sigma_1, I_1)(\sigma_2, I_2)) = (\sigma_1, I_1)(\sigma_2, I_2)$, se $\sigma_1, \sigma_2 \in \Sigma_s$, ou $P_T((\sigma_1, I_1)(\sigma_2, I_2)) = (\sigma_2, I_1 + I_2)$, se $(\sigma_1 \in \Sigma_\varepsilon) \wedge (\sigma_2 \in \Sigma_s)$; ou $P_T((\sigma_1, I_1)(\sigma_2, I_2)) = (\sigma_1, I_1)(\varepsilon, I_2)$, se $(\sigma_1 \in \Sigma_s) \wedge (\sigma_2 \in \Sigma_\varepsilon)$; ou $P_T((\sigma_1, I_1)(\sigma_2, I_2)) = (\varepsilon, I_1 + I_2)$, se $\sigma_1, \sigma_2 \in \Sigma_\varepsilon$; *(iv)* $P_T(s(\sigma, I)) = P_T(P_T(s)(\sigma, I))$, $s \in (\Sigma_l \times 2^{\mathbb{R}_+})^*$, $|s| \geq 2$, e $\sigma \in \Sigma_l \times 2^{\mathbb{R}_+}$.

Note que a projeção de uma sequência elimina os eventos não pertencentes ao conjunto menor e adiciona seus intervalos de tempo ao próximo evento em Σ_s. De acordo com a definição 5, a projeção de uma sequência formada apenas por eventos em Σ_ε resulta em (ε, I), em que I é o resultado da soma de todos os intervalos associados aos eventos da sequência.

2.1 Operações com Autômatos Temporizados

A fim de definir operações envolvendo autômatos temporizados, a seguinte hipótese é necessária.

H1. Os autômatos temporizados não possuem ciclos de estados conectados por transições rotuladas por ε (transições-ε).

O motivo dessa restrição é que ciclos e autolaços como os descritos acima podem atrasar a observação de um evento indefinidamente, resultando, de acordo com a definição 5, em infinitas possibilidades de projeções de uma sequência.

Visto que ATIs possuem intervalos de tempo associados a suas transições, conforme a definição 1, será usado neste trabalho a generalização das operações de complemento, observador e produto para autômatos temporizados (Marques e Basilio, 2022; Marques et al., 2023)

1) Projeção. Considere um ATI não determinístico G_T, cujo conjunto de eventos é Σ, e o conjunto $\Sigma_s \subset \Sigma$. O ATI projeção G_T em relação a Σ_s, $Proj(G_T, \Sigma_s)$, é obtido em duas etapas: (1) Obter $G_{T,\varepsilon} = \mathcal{E}(G_T) = (X, \Sigma_s \cup \{\varepsilon\}, f_{T,\varepsilon}, X_0, X_m, \mu_\varepsilon)$, tal que $\mu_\varepsilon(x, \sigma, y) = \mu(x, \sigma, y)$, se $\sigma \in \Sigma_s$ ou $\mu_\varepsilon(x, \varepsilon, y) = \mu(x, \sigma, y)$, se $\sigma \in \Sigma \setminus \Sigma_s$ e $f_{T,\varepsilon}(x, \sigma) = f(x, \sigma)$, se $\sigma \in \Sigma_s$, ou $f_{T,\varepsilon}(x, \varepsilon) = f(x, \sigma)$, se $\sigma \in \Sigma \setminus \Sigma_s$; (2) A partir de $G_{T,\varepsilon}$ obter $G_{T_P} = \{X_P, \Sigma_s, f_P, x_{0P}, X_{mP}, \mu_P\}$, tal que $X_P = \{x \in X | \exists \sigma \in \Sigma, \sigma \in \Upsilon(x)\} \cup X_0$, $f_P(x_P, \sigma) = \{x \in X_P | (\exists n \in \mathbb{N})(\exists s = \varepsilon^n \sigma)[x = f_{T,\varepsilon}(x_P, s)]\}$ e dado estados $x, y \in X_P$ e uma sequência temporizada $s_t = (\varepsilon, I_1) \ldots (\varepsilon, I_n)(\sigma, I)$, tal que $f_{T,\varepsilon}(x, \varepsilon^n \sigma) = y$, tem-se que $\mu_P(x, \sigma, y) = I + \sum_{i=1}^{n} I_i$.

2) ATI determinístico equivalente a um ATI não determinístico. Considere um ATI não determinístico $G_T = (X, \Sigma, f, X_0, X_m, \mu)$ e suponha, sem perda de generalidade, que G_T não tenha transições-ε [1]. Seu autômato determinístico equivalente $Det(G_T) = G_T^{det} = (X_D, \Sigma, f_D, x_{0_D}, X_{m_D}, \mu_D)$, tal que $\mathcal{L}(G_T)$ e $\mathcal{L}_m(G_T)$, pode ser obtido de acordo com o algoritmo definido em Marques e Basilio (2022).

3) Complementar de um ATI. Considere, sem perda de generalidade, um autômato temporizado determinístico $G_T = (X, \Sigma, f, x_0, X_m, \mu)$ e considere o problema de se calcular o autômato $G_T^c = (X \cup \{x_d\}, \Sigma, f^c, x_0, \{X \cup \{x_d\}\} \backslash X_m, \mu^c)$ cuja linguagem marcada temporizada é $\mathcal{L}_{mT}(G_T^c) = (\Sigma \times 2^{\mathbb{R}_+})^* \setminus \mathcal{L}_{mT}(G_T)$. Para cada estado $x \in X \cup \{x_d\}$, $f^c(x, \sigma) = f(x, \sigma)$ e $(\mu^c(x, \sigma, y) = \mu(x, \sigma, y))[\forall y \in f(x, \sigma)]$, e $f^c(x, \sigma) = x_d$ e $\mu^c(x, \sigma, x_d) = \mathbb{R}^2 \setminus \bigcup_{y \in f(x,\sigma)} \mu(x, \sigma, y)$, se $f(x, \sigma)!$; ou, $f^c(x, \sigma) = x_d$ e $\mu^c(x, \sigma, x_d) = [0, +\infty)$, caso contrário, e $f^c(x_d, \sigma) = x_d$ e $\mu^c(x_d, \sigma, x_d) = [0, +\infty)$ para todo $\sigma \in \Sigma$.

A partir de G_T é feito o seguinte procedimento para obter-se o seu complementar G_T^c. *Passo 1:* criar uma cópia de G_T e um estado novo x_d, de forma que para cada $x \in X$ e $\sigma \in \Sigma$, é definida uma nova transição $(x, (\sigma, I), x_d)$, tal que $I = \mathbb{R}^2 \setminus \bigcup_{y \in f(x,\sigma)} \mu(x, \sigma, y)$, se $f(x, \sigma)!$ ou $I = [0, +\infty)$, caso contrário. *Passo 2:* para cada $\sigma \in \Sigma$, criar autolaços $(x_d, (\sigma, [0, +\infty)), x_d)$. *Passo 3:* desmarca-se os estados originalmente marcados X_m e marca-se os estados em $(X \cup \{x_d\}) \backslash X_m$.

4) Produto de dois ATIs. O produto entre dois ATIs deve não só sincronizar os eventos em comuns, como no caso de autômatos não temporizados (Cassandras e Lafortune, 2008), mas também sincronizar o intervalo de tempo associado às transições.

Formalmente, o produto entre dois ATIs $G_{1T} = (X_1, \Sigma_1, f_1, x_{01}, X_{m1}, \mu_1)$ e $G_{2T} = (X_2, \Sigma_2, f_2, x_{02}, X_{m2}, \mu_2)$, tais que Γ_1 e Γ_2 são suas funções de conjunto de eventos ativos, respectivamente, é dado pelo ATI $G_{1T} \times G_{2T} = Ac(X_1 \times X_2, \Sigma_1 \cup \Sigma_2, f_{1\times2}, (x_{01}, x_{02}), X_{m1} \times X_{m2}, \mu_{1\times2})$, tal que para $x_1 \in X_1$ e $x_2 \in X_2$, define-se $f_{1\times2}((x_1, x_2), \sigma) = (f_1(x_1, \sigma), f_2(x_2, \sigma))$ se $\sigma \in \Gamma_1(x_1) \cap \Gamma_2(x_2)$ e $\mu_1(x_1, \sigma, y_1) \cap \mu_2(x_2, \sigma, y_2) \neq \emptyset$, ou não definida caso contrário, e $\mu_{1\times2}((x_1, x_2), \sigma, (y_1, y_2)) = \mu_1(x_1, \sigma, y_1) \cap \mu_2(x_2, \sigma, y_2)$.

3. OPACIDADE EM AUTÔMATOS TEMPORIZADOS

Nesta seção as noções de opacidade com base em linguagem temporizada feitas anteriormente em Marques e Basilio (2022) e Marques et al. (2023) são refinadas, resultando em cinco novas noções para a classe de autômatos temporizados por intervalo de tempo. Para este fim, o conjunto de eventos Σ é particionado em $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, tal que Σ_o é o conjunto de eventos observáveis e Σ_{uo}, o conjunto de eventos não observáveis. A projeção temporizada observável é dada por $P_{T_o} : (\Sigma \times (2^{\mathbb{R}_+}))^* \rightarrow (\Sigma_o \cup \{\varepsilon\} \times 2^{\mathbb{R}_+})^*$. Dada uma sequência temporizada $s \in (\Sigma \times 2^{\mathbb{R}_+})^*$, seja $N = |\pi_\Sigma(P_{T_o}(s))|$ e o conjunto $I_N = \{1, 2, \ldots, N\}$.

[1] Caso o ATI não determinístico G_T possua transições-ε, basta computar seu ATI projeção G_{T_P} para então calcular seu equivalente determinístico

Definição *6.* (Opacidade com base em linguagem temporizada (TLBO)). Sejam L_S e L_{NS} as linguagens temporizadas secreta e não secreta, respectivamente, ambas definidas em $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$. Dada uma sequência temporizada $s \in L_S$, considere o conjunto $S_{NS}(s) = \{s' \in L_{NS} | \pi_{\Sigma_o}(P_{T_o}(s')) = \pi_{\Sigma_o}(P_{T_o}(s))\}$, composto por sequências temporizadas não secretas que se confundem com a sequência secreta s quando observados somente os eventos de Σ_o. Então, L_S é, em relação a L_{NS} e Σ_o:

(*i*) Fortemente Opaca Independente do Tempo (TISO, *Time Independent Strongly Opaque*), se $(\forall s \in L_S)(\forall t = (t_1, t_2, \ldots, t_n) \in X_{k=1}^{n} \pi_T^k(s))$ $(\exists s' \in S_{NS}(s))[(\forall k \in I_N)(t_k \in \pi_T^k(s'))]$;

(*ii*) Fortemente Opaca Dependente do Tempo (TDSO, *Time Dependent Strongly Opaque*) se $(\forall s \in L_S)(\exists t = (t_1, t_2, \ldots, t_n) \in X_{k=1}^{n} \pi_T^k(s))(\exists s' \in S_{NS}(s))[(\forall k \in I_N)(t_k \in \pi_T^k(s'))]$;

(*iii*) Fracamente Opaca Independente do Tempo (TIWO, *Time Independent Weakly Opaque*) se $(\exists s \in L_S)(\forall t = (t_1, t_2, \ldots, t_n) \in X_{k=1}^{n} \pi_T^k(s))(\exists s' \in S_{NS}(s))[(\forall k \in I_N)(t_k \in \pi_T^k(s'))]$;

(*iv*) Fracamente Opaca Dependente do Tempo (TDWO, *Time Dependent Weakly Opaque*) se $(\exists s \in L_S)(\exists t = (t_1, t_2, \ldots, t_n) \in X_{k=1}^{n} \pi_T^k(s))(\exists s' \in S_{NS}(s))[(\forall k \in I_N)(t_k \in \pi_T^k(s'))]$;

(*v*) Não opaca, caso contrário.

Em que $X_{k=1}^{n} \pi_T^k(s)$ denota o produto cartesiano entre todos os elementos da sequência de intervalos de tempo $\pi_T(s)$.

De acordo com a definição 6, dada uma linguagem temporizada não secreta L_{NS}, uma linguagem temporizada secreta L_S e o conjunto de eventos observáveis Σ_o, a linguagem L_S pode ser classificada da seguinte forma. *(i)* L_S é fortemente opaca independente do tempo se todas as sequências em L_S forem completamente ofuscadas, isto é, para cada sequência secreta s existe uma não secreta s' tal que $\pi_{\Sigma_o}(P_{T_o}(s)) = \pi_{\Sigma_o}(P_{T_o}(s'))$ e $\pi_T^k(P_{T_o}(s)) \subseteq \pi_T^k(P_{T_o}(s'))$. *(ii)* L_S é fortemente opaca dependente do tempo se todas as sequências secretas forem completamente ou parcialmente ofuscadas, isto é, para cada sequência secreta s existe uma não secreta s' tal que $\pi_{\Sigma_o}(P_{T_o}(s)) = \pi_{\Sigma_o}(P_{T_o}(s'))$ e $\pi_T^k(P_{T_o}(s)) \cap \pi_T^k(P_{T_o}(s')) \neq \emptyset$. *(iii)* L_S fracamente opaca independente do tempo se existir uma sequência secreta que é completamente ofuscada. *(iv)* L_S é fracamente opaca dependente do tempo se existir uma sequência secreta que é parcialmente ofuscada. *(v)* L_S não é opaca se todas as sequências forem completamente reveladas.

4. VERIFICAÇÃO DE OPACIDADE COM BASE EM LINGUAGEM TEMPORIZADA

Considere, sem perda de generalidade, os ATIs $G_{S_{T_o}} = (X_S, \Sigma_o, f_S, x_{0_S}, X_{m_S}, \mu_S)$, que marca a projeção observável da linguagem secreta, e $G_{NS_{T_o}} = (X_{NS}, \Sigma_o, f_{NS}, x_{0_{NS}}, X_{m_{NS}}, \mu_{NS})$, que marca a projeção observável da linguagem não secreta, e cujos conjuntos de transições são Δ_S e Δ_{NS}, respectivamente. A seguir são definidas duas composições entre eles.

Definição *7.* (Composição produto temporizada rotulada ofuscada). O resultado da composição produto temporizada rotulada entre $G_{S_{T_o}}$ e $G_{NS_{T_o}}$ é o ATI $G_{S_{T_o}} \times_{ofs} G_{NS_{T_o}} = Ac(X, \Sigma_l, f, X_0, X_m, \mu)$, $X = X_S \times X_{NS}$, $\Sigma_l = \{\sigma_l | (\sigma \in \Sigma_o) \wedge (l \in \{co, po\})\}$, $X_0 = X_{0_S} \times X_{0_{NS}}$.

Para $(x_S, \sigma, x'_S) \in \Delta_S$, $(x_{NS}, \sigma, x'_{NS}) \in \Delta_{NS}$, $I_S = \mu_S(x_S, \sigma, x'_S)$ e $I_{NS} = \mu_{NS}(x_{NS}, \sigma, x'_{NS})$, as funções f e μ são definidas como $f((x_S, x_{NS}), \sigma_l) = (x'_S, x'_{NS})$, de forma que, $l = co$, se $I_S \subseteq I_{NS}$; $l = po$, se $I_S \setminus I_{NS} \neq \emptyset \wedge I_S \cap I_{NS} \neq \emptyset$; ou, l não é definida, caso contrário. E $\mu((x_S, x_{NS}), \sigma_l, (x'_S, x'_{NS})) = I_S \cap I_{NS}$, para $l \in \{co, po\}$. Ou seja, $G_{S_{T_o}} \times_{ofs} G_{NS_{T_o}}$ é o produto entre entre os ATIs que modelam os comportamentos observáveis secreto e não secreto, em que as transições do ATI resultante são rotuladas com *co*, caso a sequência secreta correspondente seja completamente ofuscada, ou com *po*, caso ela seja parcialmente ofuscada.

Em seguida, seja o ATI complementar $(G^{det}_{NS_{T_o}})^c = (X_{NS} \cup \{x_d\}, \Sigma_o, f^c_{NS}, x^{det}_{0_{NS}}, (X_{NS} \cup \{x_d\}) \setminus X_{m_{NS}}, \mu^c_{NS})$ que marca o complemento da projeção da linguagem não secreta, $P_{T_o}(L_{NS})$, e cujo conjunto de transições é Δ^c_{NS}.

Definição *8.* (Composição produto temporizada rotulada revelada). O resultado da composição produto temporizada revelada entre $G_{S_{T_o}}$ e $(G^{det}_{NS_{T_o}})^c$ é $G_{S_{T_o}} \times_{rev} (G^{det}_{NS_{T_o}})^c = Ac(X, \Sigma_l, f, X_0, X_m, \mu)$, tal que $X = X_S \times X_{NS} \cup \{x_d\}$, $\Sigma_l = \{\sigma_l | (\sigma \in \Sigma_o) \wedge (l \in \{co, po, cr, pr\})\}$, $X_0 = X_{0_S} \times \{x^{det}_{0_{NS}}\}$. Para $(x_S, \sigma, x'_S) \in \Delta_S$, $(x_{NS}, \sigma, x'_{NS}) \in \Delta^c_{NS}$, $I_S = \mu_S(x_S, \sigma, x'_S)$, e $I_{NS} = \mu^c_{NS}(x_{NS}, \sigma, x'_{NS})$, as funções f e μ são definidas como $f((x_S, x_{NS}), \sigma_l) = (x'_S, x'_{NS})$, de forma que

$$l = \begin{cases} co, \text{se } (I_S \subseteq I_{NS}) \wedge (x'_{NS} \notin X^c_{m_{NS}}); \\ po, \text{se } (I_S \setminus I_{NS} \neq \emptyset \wedge I_S \cap I_{NS} \neq \emptyset) \wedge (x'_{NS} \notin X^c_{m_{NS}}); \\ cr, \text{se } (I_S \subseteq I_{NS}) \wedge (x'_{NS} \in X_{m_{NS}}) \wedge (x_{NS} \neq x_d); \\ pr, \text{se } (I_S \setminus I_{NS} \neq \emptyset \wedge I_S \cap I_{NS} \neq \emptyset) \wedge \\ \qquad\qquad (x'_{NS} \in X_{m_{NS}}) \wedge (x_{NS} \neq x_d); \\ co, \text{se } (I_S \cap I_{NS} \neq \emptyset) \wedge (x_{NS} = x'_{NS} = x_d); \\ \text{não definida, caso contrário.} \end{cases}$$

E $\mu((x_S, x_{NS}), \sigma_l, (x'_S, x'_{NS})) = I_S \cap I_{NS}$, para $l \in \{co, po, cr, pr\}$.

Dados os ATIs G_{S_T} e G_{NS_T}, que marcam as linguagens temporizadas secreta e não secreta, respectivamente, e seus ATI projeção $G_{S_{T_o}} = Proj(G_S)$, $G_{NS_{T_o}} = Proj(G_{NS})$, seja $G_{ofs,l} = CoAc(G_{S_{T_o}} \times_{ofs} G_{NS_{T_o}}) = (X_{ofs}, \Sigma_l, f_{ofs}, X_{0_{ofs}}, X_{m_{ofs}})$ e $G_{rev,l} = CoAc(G_{S_{T_o}} \times_{rev} (G^{det}_{NS_{T_o}})^c) = (X_{rev}, \Sigma_l, f_{rev}, X_{0_{rev}}, X_{m_{rev}})$ as partes coacessiveis dos ATI construídos conforme as definições 7 e 8, respectivamente.

Em seguida, seja $G = (X, \Sigma, f, X_0, X_m)$ um autômato cujos estados são representados por tuplas da forma $x = (x^1, x^2) \in X$ e seja w um valor constante. O conjunto $X^{1,w} = \{x \in X | x^1 = w\}$ é definido como o conjunto de estados de G cuja primeira componente é igual a w. Esta definição pode ser estendida para a função de transição f, uma vez que sua saída é um estado de G, tal que para $x \in X$ e $\sigma \in \Sigma$, $f^{1,w}(x, \sigma) = \{y = (y^1, y^2) \in f(x, \sigma) | y^1 = w\}$.

A fim de se verificar qual noção de opacidade uma linguagem temporizada satisfaz, conforme a definição 6, é necessário que se construa um autômato verificador a partir dos ATIs $G_{ofs,l}$ e $G_{rev,l}$, obtidos a partir de $G_{S_{T_o}}$ e $G_{NS_{T_o}}$ por meio das definições 7 e 8. O autômato verificador

$G_v = (X_v, \Sigma_v, f_v, X_{0_v}, X_{m_v})$ é tal que seus estados são tuplas da forma (x_1, x_2), em que o primeiro elemento tem origem em $G_{ofs,l}$ e o segundo em $G_{rev,l}$.

Definição *9.* (Autômato verificador rotulado). O verificador rotulado $G_v = (X_v, \Sigma_v, f_v, X_{0_v}, X_{m_v})$ é um autômato não temporizado, cujos elementos são definidos a seguir: $X_v = \{x_v = (x_{v_1}, x_{v_2}) | (\exists x_s \in X_S)[(x_{v_1} = X_{ofs}^{1,x_s}) \wedge (x_{v_2} = X_{rev}^{1,x_s})]\}$; $\Sigma_v = \{\sigma_l | (\sigma \in \Sigma_o) \wedge (l \in \{co, por, cr\})\}$; $X_{0_v} = \{x_{0_v} = (x_{0_{v_1}}, x_{0,v_2}) | (\exists x_0 \in X_{0_S})[(x_{0_{v_1}} = X_{0_{ofs}}^{1,x_0}) \wedge (x_{0_{v_2}} = X_{0_{rev}}^{1,x_0})]\}$.

Dado um estado $x_s \in X_S$ in $G_{S_{To}}$, a função de transição de G_v é definida como:

(i) $f_v((x_{v_1}, x_{v_2}), \sigma_{co}) = \Big((f_{ofs}^{1,x_s}(x, \sigma_{l_1}), f_{rev}^{1,x_s}(y, \sigma_{l_2})),$ se $(\exists x \in x_{v_1})(\exists x_s \in X_s)(\exists l_1 \in \{co, po\})[f_{ofs}^{1,x_s}(x, \sigma_{l_1})!] \wedge [(\exists y \in x_{v_2})(\exists l_2 \in co, po)[f_{rev}^{1,x_s}(x, \sigma_{l_2})!] \Big) \vee \Big((f_{ofs}^{1,x_s}(x, \sigma_{l_1}), \emptyset),$ se $(\exists x \in x_{v_1})(\exists x_s \in X_s)(\exists l_1 \in \{co, po\})[f_{ofs}^{1,x_s}(x, \sigma_{l_1})!] \wedge [(f_{rev}^{1,x_s}(x, \sigma_{l_2}) \neg !, \forall y \in x_{v_2})] \Big)$;

(ii) $f_v((x_{v_1}, x_{v_2}), \sigma_{cr}) = (\emptyset, f_{rev}^{1,x_s}(y, \sigma_{l_2}))$, se $(\exists y \in x_{v_2})(\exists x_s \in X_s)(\exists l_2 \in \{pr, cr\})[f_{rev}^{1,x_s}(x, \sigma_{l_2})!] \wedge [f_{ofs}^{1,x_s} \neg !, \forall x \in x_{v_1}]$

(iii) $f_v((x_{v_1}, x_{v_2}), \sigma_{por}) = (f_{ofs}^{1,x_s}(x, \sigma_{l_1}), f_{rev}^{1,x_s}(y, \sigma_{l_2}))$, se $(\exists x \in x_{v_1})(\exists x_s \in X_s)(\exists l_1 \in \{co, po\})[f_{ofs}^{1,x_s}(x, \sigma_{l_1})!] \wedge (\exists y \in x_{v_2})(\exists l_2 \in \{cr, pr\})[f_{rev}^{1,x_s}(x, \sigma_{l_2})!]$

A verificação da opacidade com base em linguagem temporizada é feita de acordo com o algoritmo 1. Dados os ATIs G_{S_T} e G_{NS_T}, que marcam a linguagem temporizada secreta L_S e a linguagem temporizada não secreta L_{NS}, respectivamente, e um conjunto de eventos observáveis Σ_o, o algoritmo 1 começa, na linha 2, com a construção dos ATI projeção $G_{S_{To}} = Proj(G_{S_T})$ e $G_{NS_{To}} = Proj(G_{NS_T})$.

Em seguida, na linha 3, é obtido o ATI rotulado $G_{ofs,r} = CoAc(Ac(G_{S_{To}} \times_{ofs} G_{NS_{To}}))$, de acordo com a definição 7. Este ATI marca a interseção entre as projeções observáveis de L_S e L_{NS}, ou seja, as sequências secretas que são ofuscadas completamente ou parcialmente. Portanto, se $G_{of,r} = \emptyset$, então não há sequências secretas ofuscadas e L_S não é opaca em relação a L_{NS} e Σ_o (linhas 4-5).

Porém, se $G_{ofs,r} \neq \emptyset$, L_S satisfaz alguma noção de opacidade apresentada na definição 6, e nas linhas 7-9 do algoritmo 1 são obtidos o ATI determinístico equivalente $Det(G_{NS_{To}}) = G_{NS_{To}}^{det}$, seu complementar $(G_{NS_{To}}^{det})^c$, e o ATI rotulado $G_{rev,r} = CoAc(Ac(G_{S_{To}} \times_{rev} (G_{NS_{To}}^{det})^c))$, que marca as sequências secretas que são parcialmente ou completamente reveladas. Portanto, se $G_{rev,r} = \emptyset$, então L_S é fortemente opaca independente do tempo em relação a L_{NS} e Σ_o, visto que não há sequências de L_s que são reveladas, ou seja, todas as sequências secretas são ofuscadas (linhas 10-11).

Por outro lado, se $G_{rev,r} \neq \emptyset$, então existe pelo menos uma sequência secreta que é ou parcialmente ou completamente revelada. Então, o próximo passo (linha 13) é calcular o autômato verificador rotulado G_v, de acordo com a definição 9 e seguindo o algoritmo 2.

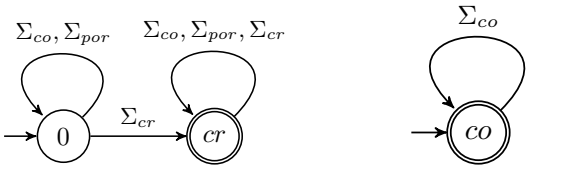
Cada estado de G_v é uma tupla $x_v = (x_{ofs}, x_{rev})$, cuja primeira componente é um conjunto de estados de $G_{ofs,l}$ e a segunda componente é um conjunto de estados de $G_{rev,l}$. Ambas também podem ser, não simultaneamente, conjuntos vazios. Os estados em x_{ofs} e x_{rev}, são estados de $G_{ofs,l}$ e $G_{rev,r}$, respectivamente, cujas primeiras componentes são iguais a um dado estado em $G_{S_{To}}$.

Desta forma, cada transição em G_v representa uma transição δ em $G_{S_{To}}$, cujo rótulo é *co*, *cr* ou *por*, dependendo dos rótulos associados às transições em $G_{of,l}$ e $G_{rev,l}$ que representam δ, de acordo com a definição 9. Então, na próxima checa-se se existem sequências completamente reveladas, parcialmente reveladas/ofuscadas ou completamente ofuscadas. Para isso, o autômato G_{cr} (figura 2a) é construído, tal que $\Sigma_{co} = \{\sigma_{co} | \sigma \in \Sigma\}$, $\Sigma_{por} = \{\sigma_{por} | \sigma \in \Sigma\}$ e $\Sigma_{cr} = \{\sigma_{cr} | \sigma \in \Sigma\}$, e cuja linguagem marcada consiste em sequências que têm pelo menos uma transição de evento rotulado com *cr*. Então, é feito na linha 14 o produto entre G_v e G_{cr}. O resultado $G_{v,cr}$ indica a existência de transições completamente ofuscadas. Se $G_{v,cr} = \emptyset$ (linhas 15-16), não há sequências desta natureza, portanto L_S é fortemente opaca dependente do tempo. Caso $G_{v,cr} \neq \emptyset$, na linha 18 é feita a operação produto entre G_v e G_{co} (figura 2b), cujo resultado é o autômato $G_{v,co}$, que indica a existência de transições completamente ofuscadas. Se $G_{v,co} \neq \emptyset$ (linhas 19-20), então L_S é fracamente opaca independente do tempo. Caso contrário (linhas 21-22), L_{NS} é fracamente opaca e dependente do tempo.

Algoritmo 1: Verificação de TLBO

Entrada: G_{S_T}, G_{NS_T}, Σ_o
Saída : TISO (V/F), TDSO (V/F), TIWO (V/F), TDWO (V/F), NO (F)

1 Definir TISO = F, TDSO = F, TIWO = F, TDWO = F, NO = F
2 Calcular $G_{S_{To}} = Proj(G_{S_T}, \Sigma_o)$ e $G_{NS_{To}} = Proj(G_{NS_T}, \Sigma_o)$
3 Calcular $G_{ofs,r} = CoAc(Ac(G_{S_{To}} \times_{ofs} G_{NS_{To}}))$
4 **se** $G_{ofs,r} = \emptyset$ **então**
5 | NO = V
6 **senão**
7 | Calcular $G_{NS_{To}}^{det} = Det(G_{NS_T})$
8 | Calcular $(G_{NS_{To}}^{det})^c$
9 | Calcular $G_{rev,r} = CoAc(Ac(G_{S_{To}} \times_{rev} G_{NS_{To}}^{det}))$
10 | **se** $G_{rev,r} = \emptyset$ **então**
11 | | TISO = V
12 | **senão**
13 | | Calcular verificador $G_v = V(G_{ofs,l}, G_{rev,l})$
14 | | Calcular $G_{v,cr} = CoAc(G_v \times G_{cr})$
15 | | **se** $G_{v,cr} = \emptyset$ **então**
16 | | | TDSO = V
17 | | **senão**
18 | | | Calcular $G_{v,co} = CoAc(G_v \times G_{co})$
19 | | | **se** $G_{v,co} \neq \emptyset$ **então**
20 | | | | TIWO = V
21 | | | **senão**
22 | | | | TDWO = V



(a) Autômato G_{cr}. (b) Autômato G_{co}.

Figura 2. Autômatos utilizados no algoritmo 1.

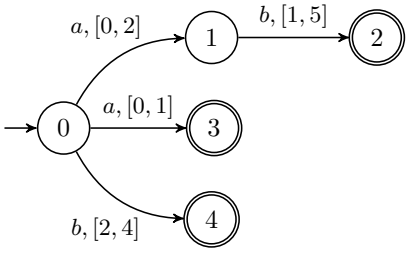


Figura 3. ATI G_{S_T}.

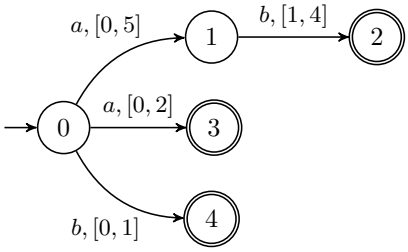


Figura 4. ATI G_{NS_T}.

5. EXEMPLO

Sejam as linguagens temporizadas por intervalos secreta $L_S = \{(a, [0, 1]); (a, [0, 2])(b, [1, 5]); (b, [2, 4])\}$ e não secreta $L_{NS} = \{(a, [0, 2]); (a, [0, 5])(b, [1, 4]); (b, [0, 1])\}$, ambas definidas no conjunto $\Sigma = \Sigma_o = \{a, b\}$. Analisando as sequências de L_S é possível concluir que $s_1 = (a, [0, 1])$ é completamente ofuscada por $s'_1 = (a, [0, 2])$, $s_2 = (a, [0, 2])(b, [1, 5])$ é parcialmente ofuscada por $s'_2 = (a, [0, 5])(b, [1, 4])$ e $s_3 = (b, [2, 4])$ é completamente revelada. Portanto, L_s é fracamente opaca e é independente do tempo, por causa de s_1. Note que a presença de sequências completamente reveladas, como por exemplo s_3, em linguagens temporizadas faz com que elas não satisfaçam os critérios de opacidade forte, isto é, TISO e TDSO. Desta forma, de acordo com a definição 6, L_S é fracamente opaca independente do tempo (TIWO) em relação a L_{NS} e Σ_o.

Agora, utilizando o algoritmo 1 para verificar a opacidade, constrói-se os autômatos G_{S_T} (figura 3), que marca L_S e G_{NS_T} (figura 4), que marca L_{NS}. Como todos os eventos são observáveis, estes ATI são seus próprios ATI projeção. Em seguida, são obtidos os ATIs rotulados $G_{ofs,l}$ e $G_{rev,l}$, mostrados nas figuras 5 e 7, respectivamente. Como ambos são não vazios, a próxima etapa é construir o autômato verificador G_v, representado na figura 9 (o autômato G_v com os rótulos originais é representado na figura 8). Em seguida, o resultado da composição $G_{v,cr} = CoAc(G_v \times G_{cr})$ é diferente de vazio, devido à existência da transição b_{cr} em G_v. Da mesma forma, Em seguida, o resultado da composição $G_{v,co} = CoAc(G_v \times G_{co})$ é diferente de vazio, devido à existência da transição a_{co} em G_v. Finalmente, é possível concluir que L_S é TIWO em relação a L_{NS} e Σ_o (linha 20 do algoritmo 1), resultado que condiz com a análise feita anteriormente.

Algoritmo 2: Verificador Rotulado

1 **Função** $V(G_s, G_{ns}, G_{of,r}, G_{rev,r})$ **é**
2 $X^1_{0_v} = X^2_{0_v} = \emptyset$
3 **para** $x_0 \in X_{0_s}$ **faça**
4 **para** $x_{0,of} \in X_{0_{ofs}}$ **faça**
5 **se** $x_{0,of}[0] = x_0$ **então**
6 $X^1_{0_v} = X^1_{0_v} \cup \{x_{0,of}\}$
7 **para** $x_{0,rev} \in X_{0_{rev}}$ **faça**
8 **se** $x_{0,rev}[0] = x_0$ **então**
9 $X^2_{0_v} = X^2_{0_v} \cup \{x_{0,rev}\}$
10 $X_{0_v} = (X^1_{0_v}, X^2_{0_v})$, $\Delta_v = \emptyset$, $X_v = \{X_0\}, X_{m_v} = \emptyset$
11 Criar dicionário $dic = \{\}$
12 $\Sigma_v = \Sigma_s \cup \Sigma_{ns}$
13 $S = \{X_0\}$
14 **enquanto** $S \neq \emptyset$ **faça**
15 $p = (p_{ofs}, p_{rev}) = pop(S)$
16 **para** $e \in \Gamma_{ofs}(s_{ofs} \in p_{ofs}) \cup \Gamma_{rev}(s_{rev} \in p_{rev})$ **faça**
17 $F_{ofs} = \{x, x \in f_{ofs}(s_{ofs} \in p_{ofs}, e)\}$
18 $F_{rev} = \{x, x \in f_{rev}(s_{rev} \in p_{rev}, e)\}$
19 **para** $x_s \in X_S$ **faça**
20 $Prox_{ofs} = \{x \in F_{ofs}, x[0] = x_s\}$
21 $Prox_{rev} = \{x \in F_{rev}, x[0] = x_s\}$
22 $Novo_{estado} = [Prox_{ofs}, Prox_{rev}]$
23 $dic[Novo_{estado}] = x_s$
24 **se** $\exists f_s(dic[p], e) = x_s$ **então**
25 $r_{ofs} = \{r | f_{ofs}(x \in p_{ofs}, e_r) = x_s\}$
26 $r_{rev} = \{r | f_{rev}(x \in p_{rev}, e_r) = x_s\}$
27 $r_v = (r_{ofs}, r_{rev})$
28 $Novo_t = (p, e_{r_v}, Novo_{estado})$
29 **se** $x_s \in X_{m_s}$ **então**
30 $X_{m_v} = X_{m_v} \cup Novo_{estado}$
31 $X_v = X_v \cup Novo_{estado}$
32 $S = S \cup Novo_{estado}$
33 $\Delta_v = \Delta_v \cup Novo_t$
34 **para** $t \in \Delta_v$ **faça**
35 $t = (x, e_{r_v}, x')$
36 $r_v = (r_{ofs}, r_{rev]})$
37 **se** $r_{rev} = \emptyset$ **então**
38 $R = co$
39 **se** $r_{ofs} = \emptyset$ **então**
40 $R = cr$
41 **se** $r_{ofs} \neq \emptyset \wedge cr, pr \subset r_{rev}$ **então**
42 $R = p$
43 **se** $r_{ofs} \neq \emptyset \wedge co, po \subset r_{rev}$ **então**
44 $R = co$
45 $t \leftarrow (x, e_R, x')$
46 Obter f_v a partir de Δ_v
47 **retorna** $G_v = (X_v, \Sigma_v, f_v, X_{0_v}, X_{m_v})$

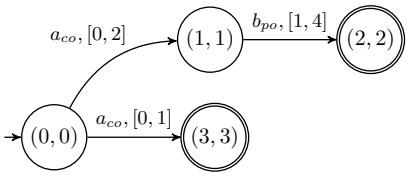


Figura 5. $G_{ofs,l}$

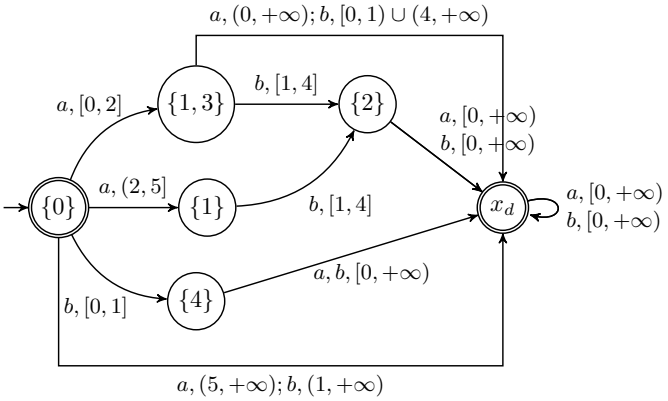


Figura 6. $(G^{det}_{NS_{T_o}})^c$

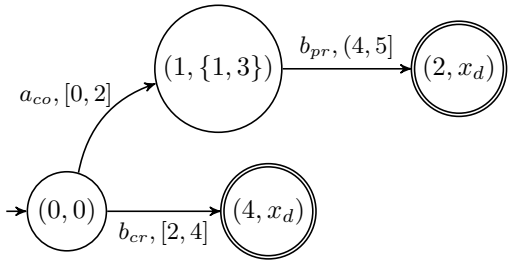


Figura 7. $G_{rev,l}$

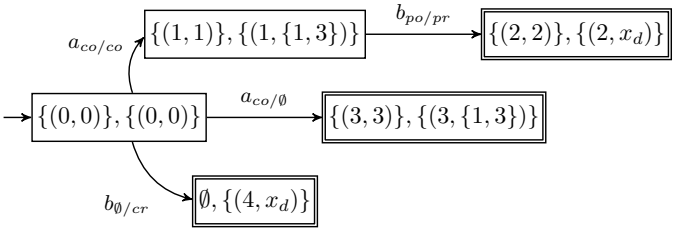


Figura 8. Verificador G_v com rótulos originais.

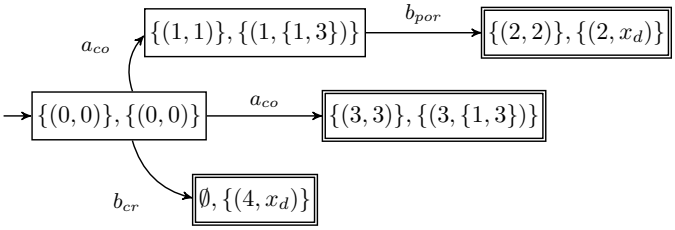


Figura 9. Verificador G_v com rótulos finais.

6. COMENTÁRIOS FINAIS

Neste artigo foi utilizada a teoria de uma classe de autômatos temporizados nos quais há um intervalo de tempo associado a cada transição (Marques e Basilio, 2022; Marques et al., 2023). A definição existente opacidade com base em linguagem temporizada e sua verificação foram estendidas em relação aos artigos anteriores de forma a considerar-se mais informações relativas ao tempo.

REFERÊNCIAS

Alur, R. e Dill, D.L. (1994). A theory of timed automata. *Theoretical Computer Science*, 126(2), 183–235.

Alves, M.V., Carvalho, L.K., e Basilio, J.C. (2020). Supervisory control of networked discrete event systems with timing structure. *IEEE Transactions on Automatic Control*, 66(5), 2206–2218.

Ammar, I., El Touati, Y., Yeddes, M., e Mullins, J. (2021). Bounded opacity for timed systems. *Journal of Information Security and Applications*, 61, 102926.

Balun, J. e Masopust, T. (2021). Comparing the notions of opacity for discrete-event systems. *Discrete Event Dynamic Systems*, 31(4), 553–582.

Cassandras, C.G. e Lafortune, S. (2008). *Introduction to Discrete Events Systems*. Springer, New York, NY : USA, 2nd edition.

Jacob, R., Lesage, J.J., e Faure, J.M. (2016). Overview of discrete event systems opacity: Models, validation, and quantification. *Annual Reviews in Control*, 41, 135–146.

Lafortune, S., Lin, F., e Hadjicostis, C.N. (2018). On the history of diagnosability and opacity in discrete event systems. *Annual Reviews in Control*, 45, 257–266.

Lin, F. (2011). Opacity of discrete event systems and its applications. *Automatica*, 47(3), 496–503.

Marques, M.G., Barcelos, R.J., e Basilio, J.C. (2023). The use of time-interval automata in the modeling of timed discrete event systems and its application to opacity. *IFAC-PapersOnLine*, 56(2), 8654–8659.

Marques, M.G. e Basilio, J.C. (2022). Opacidade de sistemas a eventos discretos modelados por uma classe de autômatos temporizados. In *Congresso Brasileiro de Automática-CBA. Anais do Congresso Brasileiro de Automática 2022. Fortaleza: Brasil, 2022. v. 1.*

Mazaré, L. (2004). Using unification for opacity properties. *Proceedings of the 4th IFIP WG1*, 7, 165–176.

Saboori, A. e Hadjicostis, C.N. (2007). Notions of security and opacity in discrete event systems. In *46th IEEE Conference on Decision and Control*, 5056–5061.

Schneider, S., Litz, L., e Lesage, J.J. (2012). Determination of timed transitions in identified discrete-event models for fault detection. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 5816–5821. IEEE.

Viana, G.S., Alves, M.V., e Basilio, J.C. (2021). Codiagnosability of networked discrete event systems with timing structure. *IEEE Transactions on Automatic Control*, 67(8), 3933–3948.

Wang, L., Zhan, N., e An, J. (2018). The opacity of real-time automata. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(11), 2845–2856.

Wu, Y.C. e Lafortune, S. (2013). Comparative analysis of related notions of opacity in centralized and coordinated architectures. *Discrete Event Dynamic Systems: Theory and Applications*, 23(3), 307–339.