Model checking de sistemas de segurança especificados por GRAFCET e Matriz de Causa e Efeito

Rafael C. dos Santos *, **Max H. de Queiroz** **

* *Instituto de Pesquisas da Marinha – IPqM, Marinha do Brasil – MB, 21931-090, Rio de Janeiro, RJ, (e-mail: celestino@marinha.mil.br).*
** *Departamento de Automação e Sistemas – DAS, Universidade Federal de Santa Catarina – UFSC 88040-900, Florianópolis, SC, (e-mail: max.queiroz@ufsc.br).*

Abstract: This paper presents a methodology assisted by *model checking* for the development of safety systems implemented in Programmable Logical Controllers based on specifications prepared in Binary Logical Diagram (BLD). In order to validate the DLB in the initial design stage, the safety requirements are represented by Cause and Effect Matrix (CEM) and sequential logics by GRAFCET. A method is proposed that systematizes all these requirements in an expanded CEM and a set of GRAFCET structural properties, from which the properties are extracted into temporal logic formulas for the MC algorithm. The proposal is elucidated by the development of the BLD for a motor start and stop logic, where the MC of properties automatically extracted from the MCE and the GRAFCET allowed the identification and correction of errors in the DLB through counterexamples until converging to a formally validated DLB.

Resumo: Este trabalho apresenta uma metodologia assistida por *Model Checking* (MC) para o desenvolvimento de sistemas de segurança implementados em Controladores Lógicos Programáveis a partir de especificações elaboradas em Diagrama Lógico Binário (DLB). A fim de validar o DLB na etapa inicial do projeto, os requisitos de segurança são representados por Matriz de Causa e Efeito (MCE) e as lógicas sequenciais por GRAFCET. Propõe-se um método que sistematiza todos esses requisitos numa MCE expandida e num conjunto de propriedades estruturais do GRAFCET, a partir dos quais são extraídas as propriedades em fórmulas de lógica temporal para o algoritmo de MC. A proposta é elucidada pelo desenvolvimento do BLD para uma lógica de partida e parada de motor, em que o MC de propriedades extraídas automaticamente da MCE e do GRAFCET permitiu identificar e corrigir erros no DLB por meio de contraexemplos até se convergir para um DLB formalmente validado.

Keywords: Model Checking; GRAFCET; Cause and Effect Matrix; Linear Temporal Logic; Programmable Logic Controller; Binary Logic Diagram.

Palavras-chaves: Model Checking; GRAFCET; Matriz de Causa e Efeito; Lógica Temporal Linear; Controlador Lógico Programável; Diagrama Lógico Binário.

1. INTRODUÇÃO

Model checking (MC) é uma técnica de verificação formal automática e exaustiva utilizada para validar sistemas concorrentes de estados finitos. O MC é utilizado de forma a complementar as simulações e testes, uma vez que realiza uma exploração matemática de todo espaço de estados do sistema, para provar se determinada propriedade é atendida em todos possíveis cenários do sistema e, caso não seja satisfeita, um contraexemplo é apresentado (Clarke, 1997), subsidiando a equipe de projeto na identificação e na correção de possíveis erros.

Algumas normas como IEC-61508 (2010) e IEC-61511 (2023) recomendam o MC para o desenvolvimento, verificação e validação dos sistemas críticos de segurança, como por exemplo: os setores militares, aeroespacial e industriais de processos petroquímicos e nucleares, onde um erro no sistema pode levar a consequências catastróficas, como perda de vidas humanas, impactos ambientais e prejuízos materiais e financeiros. Os Controladores Lógicos Programados (CLP) são muito utilizados na arquitetura desses sistemas e um dos principais desafios para equipe de projeto é garantir que os requisitos de segurança sejam atendidos pela especificação e programação do controlador.

Na literatura são encontrados diversos trabalhos sobre aplicação do *model checking* para sistemas de segurança controlados por CLP, visando encontrar erros nas diversas etapas do projeto, como: Pakonen and Björkman (2017) apresenta uma metodologia de verificação formal, utilizando MC na etapa de especificação das lógicas de controle, com o objetivo de proteger os sistemas de instrumentação e controle industrial voltados para aplicações em indústrias nucleares finlandesas, a fim de evitar falhas ocasionadas por ativações espúrias; Chadwick et al. (2018) propõe uma modelagem de verificação automática, baseada em MC, para aplicação em sistema de controle de trens, com o objetivo de verificar se requisitos de intertravamento de segurança são cumpridos pelo sistema; Adiego and Lopez-Miguel (2022) apresenta uma metodologia baseada emm MC para validar a programação em ST do controlador de segurança de um acelerador de partículas no CERN.

Quanto à limitação da utilização do *model checking*, a literatura destaca como principais pontos: 1) explosão combinacional de estados, que está relacionado com o cres-

cimento exponencial do tamanho do espaço de estados do sistema, impossibilitando sua representação na memória disponível nas ferramentas de verificação; e 2) dificuldade de entendimento pela equipe de projetos do formalismo matemático utilizado para especificar as propriedades durante a verificação, como por exemplo as lógicas temporais LTL (*Linear temporal Logic)* e CTL (*Computacion Tree Logic*). A fim de contribuir para solucionar a limitação citada no item 2, Reis et al. (2019) e Lázaro et al. (2019) propõem uma representação de fácil entendimento que padroniza a comunicação entre as equipes envolvidas nos projetos de sistemas de segurança, baseada em Matriz de Causa e Efeito (MCE), a qual é normatizada pela IEC-62881 (2018).

Apesar da relevância científica desses trabalhos, eles possuem uma limitação quanto à representação de propriedades que envolvem lógicas sequenciais. Essas lógicas estão presentes especialmente nos modos transitórios do sistema, como *shutdown*, *startup* e transições do processo, podendo ser especificadas por GRAFCET (IEC-60848, 2002) conforme mostrado por Saito (2019). A limitação pode trazer riscos de segurança impactantes ao sistema, levando em consideração os dados apresentados por Bridges and Clark (2011) de que 70% dos incidentes e acidentes ocorridos num sistema automático de segurança são registrados nos modos transitórios, apesar de operar por apenas 5% do tempo nesses modos.

Diante do exposto, esse trabalho visa contribuir com uma proposta de metodologia assistida por *model checking* que seja capaz de representar e verificar tanto as propriedades que envolvam lógicas combinacionais como sequenciais do sistema, baseada em MCE e GRAFCET. Além de sistematizar a transformação das propriedades estruturais do GRAFCET e de segurança para LTL.

2. MODEL CHECKING PARA SISTEMAS DE SEGURANÇA ESPECIFICADOS POR MCE

A Figura 1 mostra uma metodologia para projetos de segurança com CLP assistidos por *model checking*, com as propriedades especificadas por MCE. A metodologia incorpora as cadeias de MC propostas por Lázaro et al. (2019) e Reis et al. (2019). As cadeias são baseadas na ferramenta de verificação formal (*model checker*) TINA/SELT e diferem apenas quanto à etapa de aplicação. A abordagem do Lázaro et al. (2019) é aplicada na etapa de especificação do código para sistema de controle e monitoração dos navios da Marinha do Brasil. Já a proposta do Reis et al. (2019) é aplicada na etapa de programação para os sistemas integrados de segurança da indústria de Petróleo e Gás.

A parte não destacada na Figura 1 mostra de forma genérica a atual abordagem utilizada no desenvolvimento de projetos de sistemas de segurança. Ela é iniciada a partir de dois documentos importantes, elaborados pelas equipes de projeto: memorial descritivo e MCE. O memorial descritivo é um documento com informações básicas sobre o comportamento do sistema e a MCE representa os requisitos de segurança a serem satisfeitos. De posse desses dois documentos, a equipe técnica elabora o Diagrama Lógico Binário (BLD), padronizado pela ISA-5.2 (2016), o qual expressa toda a lógica de controle e intertravamento. Após essa etapa, o código de programação dos CLP é desenvolvido. Por fim, testes e simulações são realizados com o objetivo de verificar se o sistema está em conformidade com os requisitos de segurança. Caso aprovado, o sistema é instalado na planta. E se reprovado, correções são realizadas no DLB ou código de CLP e novos testes e simulações são realizados. As partes destacadas mostram as cadeias de verificação formal propostas pelos autores citados. O *model checking* integra-se com a atual metodologia de desenvolvimento de sistemas de segurança, de forma complementar aos testes e às simulações, na validação da especificação do código ou do programa dos CLP envolvidos no sistema de segurança, dependendo da cadeia utilizada.

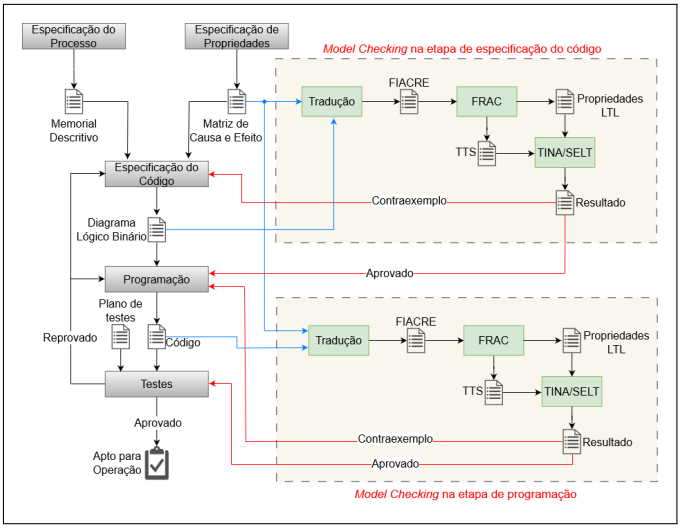


Figura 1. Metodologias assistidas por *Model Checking* para sistemas de segurança especificados por MCE.

A cadeia de *model checking* é composta por: uma tradução intermediária entre os modelos de alto nível (DLB ou código) para os modelos formais de baixo nível aceitos pelo model checker, por meio da linguagem FIACRE; O documento gerado em FIACRE é traduzido automaticamente para os formalismos de verificação TTS (Sistema de Transição Temporizado) e LTL, por meio da ferramenta FRAC; e o TINA/SELT converte o modelo formal TTS para KTZ (estrutura de Kripke), um autômato com proposições atômicas associadas aos estados, e realiza um verificação automática e exaustiva para provar se as propriedades em LTL são atendidas pelo modelo KTZ. Em caso de alguma propriedade não ser satisfeita, um contraexemplo é apresentado e convertido na forma de diagrama de sinais.

2.1 Matriz de Causa e Efeito

Matriz Causa e Efeito (MCE) é uma forma matricial de representar as lógicas de dependências entre as causas (eventos) e os efeitos (ações), que devem ocorrer de forma automática pelo sistema. As linhas da matriz correspondem às causas e as colunas correspondem aos efeitos. A lógica de ativação de cada efeito é especificada pelas relações indicadas nas interseções da matriz na respectiva coluna através dos símbolos que expressam operações lógicas booleanas e temporizações. A norma IEC-62881 (2018) estabelece diretrizes gerais para configuração e implementação de MCE, sem no entanto fixar um padrão de símbolos, que dever adequado a cada setor de aplicação. A simbologia a seguir foi adotada por Lázaro et al. (2019) para a especificação de sistemas de segurança embracados em navios:

1) x: as causas relacionadas por este símbolo ativam diretamente o efeito na respectiva coluna;
2) N: a negação da causa relacionada por este símbolo ativa o efeito na respectiva coluna;
3) Ai: representa a lógica AND entre as causas relacionadas ao índice $i = 1, 2, 3,...$ na mesma coluna;
4) NAi: representa a lógica AND NOT entre as causas relacionadas ao índice $i = 1, 2, 3,...$ na mesma coluna;
5) Tk: as causas relacionadas por este símbolo deverão se manter ativas por k segundos para ativar o efeito.

Para exemplificar essa simbologia, a MCE da Tabela 1 expressa as seguintes relações: o efeito Ea deve ser ativado enquanto a causa Ca estiver ativa e Cb desativada, ou as causas Cb e Ca estiverem ativas. Já o efeito Eb deve ser ativado enquanto Ca estiver ativa ou Cc estiver ativa por mais de 5s.

Tabela 1. Exemplo de uma MCE com lógicas combinacionais e temporização.

Causas \ Efeitos	Ea	Eb
Ca	A1	x
Cb	NA1,A2	
Cc	A2	T5

2.2 Tradução de MCE para LTL

As propriedades em LTL são elaboradas a partir dos requisitos definidos na MCE, em conformidade com a IEC-61511 (2023). Essa norma destaca dois possíveis tipos de falhas para um sistema de segurança: Falha por Ativação Espúria (ST - *Spurious Trip*) e Falha sob Demanda (FD - *Failure on Demand*). ST é uma falha gerada quando o sistema atua sem ser demandado (não possui uma causa ativa), gerando perda de produção ou mesmo situações perigosas. FD é uma falha ocasionada por uma falta de ação do sistema diante de uma situação com potencial risco à segurança do processo.

Desta forma, duas propriedades de segurança são definidas para verificação: Livre de Ativação Espúria (STF - *Spurious Trip Freeness*) e Livre de Falha sob Demanda (FDF - *Failure on Demand Freeness*). STF expressa que para todos os possíveis estados alcançados pelo sistema nunca existirá um efeito (ação) sem a causa (motivação). E FDF expressa que para todos os possíveis estados, o sistema nunca será demandado (ocorrência de uma causa) e não atuará (sem efeito). Essas propriedades podem ser extraídas sistematicamente da MCE para cada efeito (coluna) através das seguintes fórmulas LTL:

$$STF = G\neg(probe \wedge \neg causa \wedge efeito) \quad (1)$$

$$FDF = G\neg(probe \wedge causa \wedge \neg efeito) \quad (2)$$

Onde:
- G é o combinador temporal LTL "*Globally*", que afirma que todos os estados no futuro satisfazem uma determinada proposição;
- $\wedge$, $\vee$ e $\neg$ são os combinadores booleanos de conjunção, disjunção e negação, respectivamente;
- *probe* é o estado do modelo em FIACRE onde as saídas do CLP são atualizadas conforme os valores calculados para as memórias correspondentes no último ciclo de varredura; e
- *causa* representa a combinação lógica das entradas para uma mesma coluna da MCE.

3. PROPOSTA DE MODEL CHECKING PARA SISTEMAS DE SEGURANÇA ESPECIFICADOS POR GRAFCET E MCE

A Figura 2 mostra a proposta de metodologia assistida por *MC* para os sistemas de segurança com especificação de propriedades tanto combinacionais como sequenciais, capaz de verificar e validar propriedades de segurança e estruturais. As principais contribuições em relação às metodologias apresentadas na Figura 1 são: a adição do GRAFCET para representar as propriedades sequenciais; transformação da MCE e GRAFCET para uma MCE expandida que expresse as especificações das propriedades de forma compilada, tanto combinacional como sequencial; e representação e verificação de propriedades estruturais em LTL do GRAFCET. De forma similar a cadeia de *model checking* da Figura 1, a cadeia da Figura 2 é baseada no TINA/SELT. Ela recebe o código ou DLB que expressa o comportamento do sistema e a MCE expandida juntamente com as propriedades estruturais do GRAFCET. Utilizando a linguagem FIACRE, O DLB/Código é traduzido pela ferramenta FRAC para TTS e, posteriormente, para KTZ. Enquanto, a MCE expandida e as propriedades estruturais são traduzidas para propriedades LTL. Por fim, a ferramenta TINA/SELT verifica se as propriedades LTL são satisfeitas pelo modelo do sistema em KTZ e se o GRAFCET atende as propriedades estruturais. Essa cadeia tem a capacidade de realizar a verificação formal tanto no domínio da etapa de especificação do código como de programação.

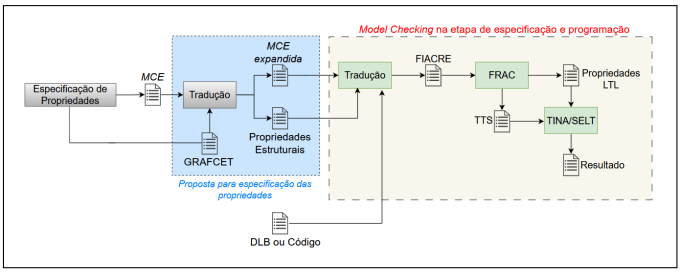


Figura 2. Proposta de metologia assistida por MC para sistemas de segurança especificados por GRAFCET e MCE.

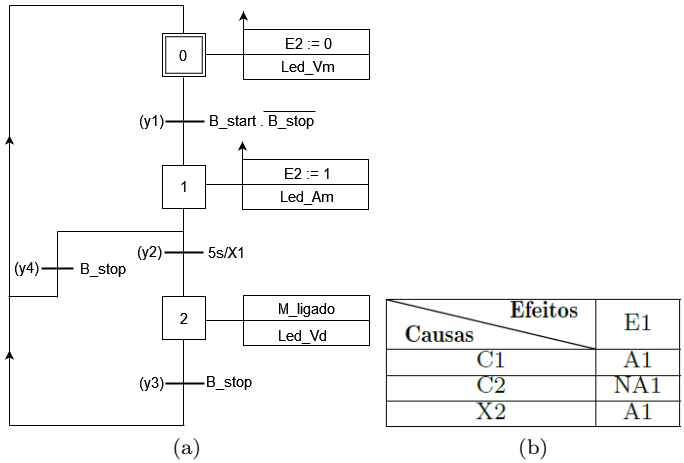
3.1 GRAFCET

GRAFCET (*GRAphe Fonctionnel de Commande Etape Transition*) é uma linguagem gráfica que permite representar, de forma simples e compacta, especificações de sistemas industriais automatizados, com capacidade de representar as combinações de lógicas de controle sequenciais (David, 1995). O GRAFCET é padronizado pela IEC-60848 (2002). Ele é composto por cinco elementos: etapa, transição, arco, ação e receptividade.

A etapa representa um estado, fase ou modo e tem ações associadas. A etapa é representada graficamente por quadrado e expressa o estado, fase ou modo e tem ações associadas. Em qualquer instante, a etapa pode estar ativa ou inativa. A ação, representada por retângulos colocados a direita da etapa, exprime os efeitos que devem ser obtidos em uma determinada situação e também ordens de comando. Ela está ativa apenas quando a etapa correspondente está ativa. A transição é representa graficamente por traço horizontal e mostra a evolução do GRAFCET de uma situação para outra. Em um dado instante, uma transição pode está válida ou não. A receptividade expressa as condições que terão de ser satisfeitas para que a transição seja executada. O arco indica a sequência do GRAFCET pela interligação de uma etapa a uma transição e vice-versa. A Figura 3a mostra um exemplo de um GRAFCET utilizado para especificar a lógica de controle envolvida num sistema de segurança. Ele possui 3 etapas (0, 1, 2), 4 transições (y1, y2, y3, y4), cada transição possui sua receptividade (ex.: y4 tem como receptividade B_stop) e cada etapa possui suas ações relacionadas (ex.: etapa 2 tem como ações ligar led verde e ligar motor).

3.2 Geração de LTL pela Matriz de Causa e Efeito Expandida

As propriedades de segurança para a metodologia proposta neste trabalho estão representadas pela MCE e pelo GRAFCET, respectivamente por lógicas combinacionais



(a) (b)

Figura 3. Lógicas envolvidas na partida e parada de um motor: a)GRAFCET para as lógicas sequenciais; e b)MCE para a lógica combinacional.

e sequenciais. Entretanto, para extrair automaticamente as fórmulas LTL para STF e FDF, propõe-se converter a MCE e o GRAFCET numa única MCE expandida como mostrado na Tabela 2. Para isso, propõe-se incrementar a simbologia da MCE expandida com operadores voltados à lógica sequencial e que represente características do GRAFCET, conforme mostrado a seguir:

1) $l+$: identificação das lógicas l que levam à ativação retentiva do efeito;
2) $l-$: identificação das lógicas l que levam à desativação retentiva do efeito;
3) Xi, $i = 0, 1, 2...$: variável que representa a etapa i do GRAFCET;
4) K': variável que representa o valor de K no ciclo de varredura anterior;
6) yj, $j = 1, 2...$: representa a transição j do GRAFCET;
7) Xi*: utilizado para destacar a(s) etapa(s) inicial(is) do GRAFCET.

Para ilustrar essa simbologia, destacam-se as seguintes relações na Tabela 2: a segunda coluna especifica que a saída E2 deve ser ativada quando o GRAFCET estiver na etapa X1 e desativada na estapa X0; e a oitava coluna especifica que a transição y2 deve permanecer ativa enquanto o GRAFCET tiver permanecido por mais de 5s na etapa X1 no ciclo de varredura anterior.

Tabela 2. MCE expandida para as lógicas sequenciais e combinacional.

C \ E	E1	E2	Vm	Am	Vd	M	y1	y2	y3	y4	*X0	X1	X2
C1	A												
C2	NA												
B_start							A						
B_stop							NA		A	A			
y1											x-	x+	
y2												x-	x+
y3											x+		x-
y4											x+	x-	
X0'							A						
X1'								T5		A			
X2'									A				
X0		x-	x										
X1		x+		x									
X2	A				x	x							

A partir da MCE expandida as propriedades STF e FDF são sistematicamente traduzidas para LTL, para cada efeito, por meio das Equações (1) e (2). No entanto, para as lógicas sequenciais, o valor da causa está relacionada com o valor do efeito no ciclo de varredura anterior e das combinações lógicas das variáveis de entrada que ativam e desativam o efeito (Lázaro et al., 2019), como apresentada na Equação (3).

$$causa = [(efeito' \vee (logicas_ativam_efeito)) \wedge \neg(logicas_desativam_efeito)] \quad (3)$$

3.3 Representação das propriedades estruturais do GRAFCET

O GRAFCET é inspirado em Redes de Petri e herda características estruturais destas (David, 1995). A metodologia da Figura 2 propõe a verificação e validação, por meio da cadeia do *model checking*, de duas importantes propriedades estruturais que devem ser atendidas pelo GRAFCET: a) propriedade de reversibilidade (PR): afirma que para qualquer etapa alcançada a partir da etapa inicial, sempre será possível o retorno à etapa inicial; e b) propriedade exclusão mútua (PEM): assegura que não haja conflitos entre as possíveis seleções de sequências nas divergências do GARFCET, garantindo que as transições sejam mutuamente exclusivas. As Equações (4) e (5) foram elaboradas para sistematizar a tradução dessas propriedades estruturais para LTL. A equação (5) é uma forma simplificada de expressar a PEM para os casos em que não houver paralelismo no GRAFCET, de modo que a PEM possa ser verificada pela não ocorrência de etapas concorrentes.

$$PR = GF(causa_reiniciar) \rightarrow GF(probe \wedge X0^*) \quad (4)$$

$$\begin{aligned} PEM = G[probe \rightarrow & ((X0^* \wedge \neg X1... \wedge \neg Xn) \\ & \vee (\neg X0^* \wedge X1... \wedge \neg Xn) \\ & \vee (\neg X0^* \wedge \neg X1... \wedge Xn))] \end{aligned} \quad (5)$$

Onde:
- F é o combinador temporal LTL "*Future*", que afirma que uma determinada proposição será atendida por um estado futuro, sem especificá-lo.
- GF é o aninhamento dos combinadores G e F, utilizado para expressar que sempre existirá um estado no futuro que satisfaz determinada proposição;
- $\rightarrow$ é combinador lógico implicação.

As equações (1) a (5) serão ilustradas no contexto de um caso prático na próxima seção.

4. ESTUDO DE CASO: LÓGICA DE CONTROLE PARA PARTIDA E PARADA DE UM MOTOR

A metodologia proposta apresentada na Figura 2 é utilizada para validar uma lógica de controle de partida e parada de um motor integrante de um sistema de segurança controlado por CLP. A especificação da lógica envolve relações combinacionais e sequenciais. A metodologia é aplicada no domínio da especificação do código de CLP, no Diagrama de Lógico Binário (DLB).

De forma genérica e com algumas abstrações, a lógica de controle para a partida e parada do motor possui os seguintes requisitos: no estado inicial, o motor está parado, um led vermelho está acesso e uma variável de saída E2 é resetada (valor lógico 0). Quando o botão *start* é acionado e o botão *stop* está desacionado, o motor vai para o estado partindo, a variável E2 é setada (valor lógico 1) e um led amarelo é ligado. A partir desse estado, dois possíveis cenários podem acontecer: se o sistema permanecer no estado partindo por mais de 5s, um led verde é ligado e o motor entra em funcionamento, sendo desligado apenas quando o botão *stop* for acionado; e se no estado partindo o botão *stop* for acionado antes dos 5s, o motor retorna para o estado inicial desligado. Quando o motor está em funcionamento e as variáveis do sistema C1 está setada e C2 está resetada, uma variável E1 é setada. As variáveis E1, E2, C1 e C2 são variáveis do sistema envolvidas em lógicas de intertravamento.

Seguindo a metodologia proposta, o primeiros passo é a definição das especificações em GRAFCET e MCE que

expressem as propriedades a serem atendidas pela lógica de partida e parada do motor. E em seguida, o DLB é modelado para expressar o comportamento da lógica.

4.1 GRAFCET e MCE

A Figura 3a mostra o GRAFCET elaborado para especificar as propriedades que possuem lógicas sequenciais envolvidas. A etapa inicial X0 expressa a situação onde o motor está parado, com suas ações envolvidas de ligar o led vermelho e resetar E2. Quando a transição y1 (botão *start* acionado e botão *stop* desacionado) acontece, o motor vai para a etapa X1, partindo o motor, com as ações de ligar led amarelo e setar E2. Neste momento, se acontece y4 (botão *stop* é acionado), o motor retorna para etapa X0. Já se acontece y2 (o motor permaneceu partindo por 5s), o motor vai para etapa X2 onde entra em funcionamento e o led verde é ligado. Estando em X2, se o botão *stop* é acionado, o motor retorna para etapa X0.

A Figura 3b mostra a MCE elaborada para especificar a propriedade da lógica de controle de partida e parada do motor que possui relação combinacional. A variável E1 é ativada quando está na etapa X2 e C1 está ativada e C2 está desativada.

4.2 Diagrama Lógico Binário

A Figura 4 mostra uma sugestão inicial de DLB para modelar o comportamento da lógica de controle envolvida na partida e parada do motor, em conformidade com a ISA-5.2 (2016). As variáveis de entrada B_start e B_stop representam as condições dos botões *start* e *stop*, respectivamente, acionado (valor lógico 1) ou desacionado (valor lógico 0). Aux1 a Aux16 são variáveis intermediárias criadas para facilitar o entendimento e análise do diagrama. X0, X1, X2 representam as etapas e y1, y2, y3, y4 as transições do GRAFCET. Led_Vm, Led_Am, Led_Vd representam as ações de ligar (valor lógico 1) e desligar (valor lógico) os leds vermelho, amarelo e verde, respectivamente. M_ligado é uma variável para expressar a condição de operação do motor. E1, E2, C1 e C2 são variáveis relacionadas a intertravamento de segurança. O bloco DI é o temporizador com atraso na inicialização de 5s. O tempo no temporizador DI foi propositalmente modificado de 5s para 3s, a fim de testar se a metodologia consegue identificar o erro. E os blocos AND e OR são as portas lógicas.

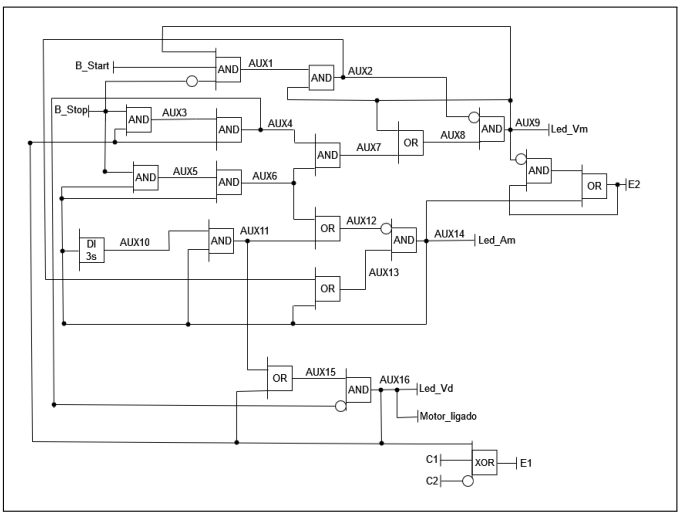


Figura 4. DLB sugerido para representar o comportamento da lógica de controle do motor.

4.3 MCE expandida e tradução para LTL

A Tabela 2 mostra a MCE expandida traduzida a partir da MCE da Figura 3b e do GRAFCET da Figura 3a. Ela representa a especificação das propriedades relacionadas tanto por lógicas combinacionais como sequenciais envolvidas na partida e parada do motor. A MCE expandida da Tabela 2 é sistematicamente traduzida para as propriedades de segurança STF e FDF por meio das Equações (1) a (3), a. De formar a exemplificar, é mostrada a seguir a tradução para o efeito X1, décimo segundo efeito da Matriz:

$$efeito_12 = X1 \tag{6}$$

$$causa_12 = [(y1 \vee X1') \wedge \neg(y2 \vee y4)] \tag{7}$$

$$STF = G\neg[probe \wedge \neg((y1 \vee X1') \wedge \neg(y2 \vee y4)) \wedge X1] \tag{8}$$

$$FDF = G\neg[probe \wedge ((y1 \vee X1') \wedge \neg(y2 \vee y4)) \wedge \neg X1] \tag{9}$$

4.4 Tradução das propriedades estruturais para LTL

As propriedades estruturais PR e PEM são sistematicamente traduzidas para LTL por meio das Equações (4) e (5), conforme mostradas a seguir:

$$PR = GF(B_stop) \rightarrow GF(probe \wedge X0) \tag{10}$$

$$PEM = G[probe \rightarrow ((X0 \wedge \neg X1 \wedge \neg X2) \vee (\neg X0 \wedge X1 \wedge \neg X2) \vee (\neg X0 \wedge \neg X1 \wedge X2))] \tag{11}$$

5. RESULTADOS E DISCUSSÃO

A Tabela 3 mostra o resultado para aplicação da metodologia da Figura 2 ao estudo de caso. A primeira linha da tabela mostra que para o DLB sugerido na Figura 4, a propriedade de segurança STF é atendida para todos efeitos da MCE expandida da Tabela 2, com a única exceção para o efeito y2. Já a propriedade FDF não é atendida para o efeito X0, porém ela é satisfeita pelo DLB para todos os outros efeitos. As propriedades não atendidas expressam que existem erros de especificação do DLB sugerido: o efeito y2 é ativado indevidamente, sem possuir causa motivadora; e o efeito X0 possui causa motivadora e não é ativado. Quanto às propriedades estruturais PR (propriedade de reversibilidade) e PEM (propriedade de exclusão mútua). As duas propriedades não são satisfeitas pelo DLB sugerido. Isso significa que o DLB possui conflitos em seu comportamento que envolve lógicas sequenciais. O TINA/SELT explorou 1617 estados e 4945 transições, com tempo de execução 47ms. O computador utilizado na verificação possui um processador 13th Gen Intel(R) Core(TM) i5-13450HX 2.40 GHz e RAM de 8GB.

Tabela 3. Resultado do MC para o estudo de caso.

	Falhas Verificadas			Custo Computacional		
DLB	STF	FDF	Estruturais	Estados	Transições	Tempo(ms)
Sugerido	y2	X0	PR, PEM	1617	4945	47
Modificado	ok	ok	PEM	3057	9505	94
Corrigido	ok	ok	ok	1401	2409	24

A 5a mostra um contraexemplo disponibilizado pela ferramenta de verificação TINA/SELT para o DLB sugerido. O contraexemplo mostra que, no tempo 4s, y2 é disparado depois do sistema permanecer por 3s com X1 ativa. De acordo com a MCE expandida, y2 deve apenas disparar quando X1 estiver ativa por 5s. Esse erro já era esperado (falha por ativação espúria), uma vez que foi modificado

o tempo do temporizador DI do DLB sugerido, de forma proposital. Já em 6s, X2 está ativa e y3 é disparada, no entanto o sistema não retorna para X0, ocasionando na falha sob demanda. Além dessa falha, ocorrem também falhas estruturais, uma vez que no tempo 7s X0, X1 e X2 estão inativas, não atendendo assim as propriedades PR e PEM.

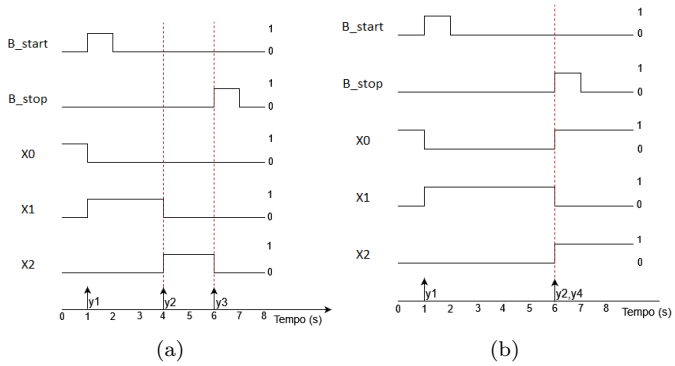


Figura 5. a) contraexemplo para DLB sugerido; b) contraexemplo para DLB modificado.

Após análise do contraexemplo da Figura 5a e visando corrigir os erros, o BLD sugerido foi modificado com as seguintes alterações: a porta lógica AND que possui como saída a variável Aux7 foi substituída por uma porta OR; e o tempo do temporizador DI foi corrigido para 5s. Após correções, uma nova verificação foi realizada e o resultado (na Tabela 3) mostra que as propriedades STF (relacionada a y2), FDF (relacionada a X0) e PR foram dessa vez satisfeitas pelo DLB modificado, assim como as propriedades que já estavam sendo satisfeitas anteriormente. No entanto, a propriedade PEM continua não sendo satisfeita pelo DLB modificado. O TINA/SELT explorou 3057 estados e 9505 transições, com tempo de execução 94ms.

Um novo contraexemplo é apresentado, conforme mostra a 5b. O contraexemplo mostra que no tempo 6s, y2 e y4 estão ocorrendo ao mesmo tempo, ativando assim X0 e X2 simultaneamente, ocasionado o erro da propriedade estrutural de exclusão mútua. Após identificar do erro, o GRAFCET foi corrigido: a receptividade do y2 passa a ser função da lógica AND entre 5s/X1 e NOT B_stop e, consequentemente, o DLB também foi corrigido: AUX10 passa a ser função da lógica AND entre o DI e NOT B_stop. Uma nova verificação foi realizada. Dessa vez, o DLB corrigido satisfaz todas propriedades como mostra a Tabela 3. A ferramenta de verificação explorou 1401 estados e 2409 transições, com tempo de execução 24ms.

6. CONCLUSÃO

Os resultados da metodologia proposta aplicada ao estudo de caso (lógica de controle de partida e parada de um motor) mostram que ela é eficiente na identificação de erros na etapa inicial para os projetos de sistemas de segurança. A metodologia conseguiu identificar falhas de não atendimento as propriedades de segurança STF (Livre de Ativação Espúria) e FDF (Livre de Falha sob Demanda) e das propriedades estruturais PR (Propriedade de Reversibilidade) e PEM (Propriedade de Exclusão mútua). Para essas falhas, contraexemplos são disponibilizados pela ferramenta de verificação, auxiliando a equipe técnica na correção dos erros. Além disso, a metodologia é capaz de validar propriedades a serem atendidas pelo sistema, aumentando assim a confiabilidade e reduzindo o risco de possíveis incidentes.

Para trabalhos futuros, pretende-se introduzir etapas de redução para lidar com possíveis problemas de explosão combinacional de estado, elaborar propriedades estruturais para lidar com o paralelismo do GRAFCET e aplicar a metodologia a um sistema real de segurança.

AGRADECIMENTOS

Os autores agradecem à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), à Universidade Federal de Santa Catarina (UFSC) e à Marinha do Brasil (MB) pelo apoio financeiro, acesso às instalações e recursos laboratoriais essenciais para a condução do estudo.

REFERÊNCIAS

Adiego, B. and Lopez-Miguel, J.C. (2022). Applying model checking to highly-configurable safety critical software: The SPS-PPS PLC Program. JACoW ICALEPCS.

Bridges, W. and Clark, T. (2011). How to efficiently perform the hazard evaluation (pha) required for non-routine modes of operation (startup, shutdown, online maintenance). In 7th Global Congress on Process Safety.

Chadwick, S., James, P., Roggenbach, M., and Wetner, T. (2018). Formal methods for industrial interlocking verification. In 2018 International Conference on Intelligent Rail Transportation (ICIRT), 1–5. IEEE.

Clarke, E.M. (1997). Model checking. In Foundations of Software Technology and Theoretical Computer Science: 17th Conference Kharagpur, India, December 18–20, 1997 Proceedings 17.

David, R. (1995). Grafcet: A powerful tool for specification of logic controllers. IEEE Transactions on Control Systems Technology, 3(3), 253–268.

IEC-60848 (2002). GRAFCET specification language for sequential function charts. International Electrotechnical Commission.

IEC-61508 (2010). Functional safety of electrical/electronic/programmable electronic safety-related systems. International Electrotechnical Commission.

IEC-61511 (2023). Functional safety - Safety Instrumented Systems for the process industry. International Electrotechnical Commission.

IEC-62881 (2018). Cause and Effect Matrix. International Electrotechnical Commission.

ISA-5.2 (2016). Binary Logic Diagrams for Process Operation. International Society of Automation.

Lázaro, F.d.S., De Queiroz, M.H., and Farines, J.M. (2019). Metodologia para desenvolvimento assistido por model checking de sistema de controle e monitoração de navios. In Congresso Brasileiro de Automática-CBA, volume 1.

Pakonen, A. and Björkman, K. (2017). Model checking as a protective method against spurious actuation of industrial control systems. In 27th European Safety and Reliability Conference, ESREL 2017, 3189–3196. CRC Press.

Reis, L.P.E., De Queiroz, M.H., Farines, J.M., LIMA, M.L., and De Campos, M.C.M.M. (2019). Verificação formal de sistemas instrumentados de segurança na indústria de petróleo e gás natural. In Congresso Brasileiro de Automática-CBA, volume 1.

Saito, G.K. (2019). Método para Análise de Riscos e Especificação de Requisitos de Segurança em Processos Industriais com Múltiplos Modos de Operação. Master's thesis, Universidade Federal de Santa Catarina.