Teste de Conformidade para Sistemas Reativos

Denise Figueiredo de Rezende[1], **Adilson Luiz Bonifácio**[1]

[1]Departamento de Computação – Universidade Estadual de Londrina (UEL)
Caixa Postal 10.011 – CEP 86057-970 – Londrina – PR – Brasil

denise.rezende@uel.br, bonifacio@uel.br

Abstract. *Technological advances have driven a significant transformation in several areas of our society, bringing countless benefits and opportunities. However, as technologies become increasingly integrated with our lives and in many critical sectors, it becomes crucial to guarantee the reliability of these systems. This ranges from ensuring information privacy to preventing accidents due to failures in transit systems. Faced with these challenges, it became necessary to seek solutions that ensure that the software testing process will more efficiently detect faults. Conformance testing and formalisms have been used to perform tests on reactive systems. This work aims to study and improve conformance testing techniques to optimize and increase the efficiency of the testing process.*

Resumo. *Os avanços tecnológicos têm impulsionado uma transformação significativa em diversas áreas da nossa sociedade, trazendo inúmeros benefícios e oportunidades. No entanto, à medida que as tecnologias se tornam cada vez mais integradas às nossas vidas e em muitos setores críticos, torna-se crucial garantir a confiabilidade desses sistemas. Isso vai desde a garantia da privacidade das informações até a prevenção de acidentes por falhas nos sistemas de trânsito. Diante desses desafios, tornou-se necessário buscar soluções que garantam que o processo de teste de software detecte falhas com mais eficiência. Testes de conformidade e formalismos têm sido usados para realizar testes em sistemas reativos. Este trabalho visa estudar e aprimorar técnicas de testes de conformidade para otimizar e aumentar a eficiência do processo de teste.*

1. Introdução

Sistemas reativos, tanto de software quanto de hardware, são caracterizados pela interação contínua com o ambiente externo. Há inúmeros sistemas reativos que também são críticos. Sistemas críticos são caracterizados pelo significativo impacto na segurança, saúde, infraestrutura da vida em sociedade. Por isso, falhas na detecção de erros nesses sistemas durante a fase de testes pode causar consequências drásticas. O controle de dispositivos e o processamento de informações sensíveis são alguns dos diversos sistemas reativos, onde uma falha pode resultar em gravíssimas consequências. Como por exemplo, num sistema de controle de tráfego, onde a ocorrência de falhas pode acarretar em acidentes de trânsito; em sistemas que lidam com dados financeiros, onde falhas não identificados podem levar a exposição de informações confidenciais. Considerando a importância crucial da detecção de falhas em sistemas reativos críticos, fica evidente o desafio de aprimorar as abordagens de testes para tais sistemas.

Uma das abordagens usadas para sistemas reativos é a verificação de conformidade [Bonifácio and Moura 2017]. O objetivo é garantir que o comportamento de um

sistema reativo esteja de acordo com o comportamento de sua respectiva especificação. Uma abordagem para verificação de conformidade em sistemas reativos usa a relação IOCO (Input/Output Conformance), cujo objetivo é verificar se as saídas geradas pelo sistema são aquelas esperadas na especificação. O formalismo usado para especificar tais sistemas é modelo IOLTS (Input Output Labeled Transition Systems), um sistema de transição rotulado com entradas e saídas, permitindo a análise formal e a verificação de conformidade sobre estes modelos.

O IOVPTS (Input/Output-Visible Pushdown Transition System), por sua vez, estende o IOLTS, para abranger uma classe de modelos mais complexa, usando uma memória auxiliar [Bonifacio and Moura 2023]. Essa complexidade adicionada pelo uso de uma memória de pilha também impacta consideravelmente na relação de conformidade adotada para estes modelos e por consequência na forma como a verificação de conformidade é realizada entre implementações e suas respectivas especificações.

Nessa linha, este trabalho visa aprimorar os métodos de verificação de conformidade para sistemas reativos bem como desenvolver ferramentas de apoio e a aplicação prática dessas técnicas em estudos de caso.

O restante desse projeto está organizado da seguinte forma. A Seção 2 descreve a fundamentação teórica dos modelos mais conhecidos. Já os objetivos dessa proposta estão descritos na Seção 3. Os procedimentos e o cronograma das atividades são apresentados na Seção 4. A Seção 5 lista algumas das contribuições esperadas desse trabalho. A proposta termina com uma lista de referências usadas na elaboração desse projeto.

2. Fundamentação

Sistemas reativos são projetados para interagir de forma contínua com o ambiente externo em tempo real. Tais sistemas precisam ser capazes de receber estímulos do ambiente, processá-los e reagir de maneira apropriada, seja realizando uma ação, gerando uma resposta ou atualizando seu estado interno [Bonifácio and Moura 2017, Bonifacio and Moura 2023].

Esses sistemas funcionam normalmente em ciclos de execução contínuos. Assim podem manter uma interação constante com o ambiente. Os desafios associados aos sistemas reativos, em geral, incluem a garantia de propriedades críticas como segurança e confiabilidade. Uma das atividades que dão suporte para enfrentar estes desafios são os testes baseados em modelos. Os testes, uma atividade importante do desenvolvimento de sistemas, juntamente com o apoio de modelos formais, agregam precisão e robustez à atividade de teste.

2.1. Modelos Formais

A fase de teste é uma etapa fundamental no processo do desenvolvimento de software. No caso de testes em sistemas reativos, existem alguns formalismos apropriados para lidarem com essa etapa de forma rigorosa. Entre esses formalismos existem alguns sistemas de transição que capturam o comportamento de sistemas reativos: os IOLTSs (Input Output Labeled Transition Systems) e os IOVPTSs (Input/Output-Visible Pushdown Transition Systems).

2.1.1. IOLTS

O IOLTS é uma extensão dos LTSs [Tretmans 2008], um sistema de transições rotuladas para descrever o comportamento de sistemas. Porém, nos IOLTSs [Bonifacio and Moura 2021] os rótulos, ou ações, são particionados em entradas e saídas, representando as interações do sistema com o ambiente. A partir deste modelo é possível: aplicar técnicas de geração de testes para criar casos de teste que contemplem diferentes cenários de interações de entrada e saída; verificação de conformidade; análise de cobertura de falhas; completude de conjuntos de testes; entre outros.

Testes baseados nesse modelo tem sido amplamente utilizados como um framework formal para verificar se uma implementação em teste (IUT) está em conformidade com uma especificação fornecida, de acordo com um determinado modelo de falha e uma relação de conformidade específica [Bonifacio and Moura 2021]. A ideia geral é que os comportamentos observados numa IUT sejam comparados ao comportamento modelado pela especificação. Quando algum comportamento distinto, de acordo com a relação de conformidade, é identificado, uma falha é encontrada [Bonifácio and Nascimento]. Este processo é realizado fornecendo as entradas (estímulos) para a IUT e sua respectiva especificação, e comparando as saídas (observáveis) geradas por ambas, a fim de identificar possíveis falhas.

O modelo IOLTS é definido, formalmente, por $M = (S, L_I, L_U, T, s_0)$, onde

- S é o conjunto contável de estados.
- L_I é o conjunto contável de rótulos de entradas.
- L_U é o conjunto contável de rótulos de saídas.
- T é uma relação do tipo $T \subseteq S \times (L \cup \tau) \times S$ que define o conjunto de transições, tal que $L = L_I \cup L_U$, $L = L_I \cup L_U \neq \emptyset$ e $L_I \cap L_U = \emptyset$.
- s_0 é o estado inicial.

Nesse modelo os estados representam eventos e contextos; as transições são as mudanças entre estados desencadeadas por estímulos do ambiente; os rótulos se referem aos símbolos representativos de um dado estímulo ou resposta.

O comportamento de um sistema reativo é definido pelo conjunto de *traces*, sequências de entradas e saídas, do IOLTS que o modela. Este conceito é a base para definir as relações de conformidade entre os modelos, pois permite a comparação das ações sobre o mesmo processamento. Um *trace*, ou computação, pode ser representado por $\sigma = \{a_1, a_2, \ldots, a_n\}$, uma sequência de ações tal que $s_0 \xrightarrow{\sigma} s$, ou seja, transições do estado inicial s_0 até o estado s.

2.1.2. IOVPTS

O modelo IOVPTS (Input/Output Visibly Pushdown Transition System) é um sistema de transição com uma memória auxiliar, permitindo a modelagem e análise de sistemas com comportamentos assíncronos mais complexos [Bonifacio and Moura 2023]. Este modelo é uma variante do VPTS (Visibly Pushdown Labeled Transition System) [Bonifacio and Moura 2022], que por sua vez podem ser associados aos VPAs (Visibly Pushdown Automata) [Alur and Madhusudan 2004], uma classe de autômatos de pilha mais restrito que os tradicionais PDAs [Hopcroft and Ullman 1979].

Com uma memória auxiliar, diferente dos IOLTS, os IOVPTSs podem modelar sistemas reativos mais expressivos, considerando tanto as interações com o ambiente externo quanto o estado interno do sistema. Este cenário é muito mais desafiador, pois o modelo base possui uma pilha para capturar comportamentos mais complexos, comumente encontrados em sistemas reativos complexos [Bonifacio and Moura 2023].

O modelo VPTS consiste em um conjunto de estados, uma pilha (que pode ser observada e manipulada externamente) e um conjunto de transições rotuladas que descrevem as operações realizadas na pilha. Logo, um VPTS é definido formalmente por $\mathcal{S} = \langle S, S_{in}, L, \Gamma, T\rangle$, onde:

- S é um conjunto finito de estados;
- $S_{in} \subseteq S$ é o conjunto de estados iniciais;
- L é um alfabeto;
- $\varsigma \notin L$ é um símbolo especial que indica ação interna;
- Γ é o alfabeto da pilha (também conhecido como alfabeto de empilhamento), onde $\perp \notin \Gamma$ é um símbolo especial que indica fundo da pilha;
- $T = T_c \cup T_r \cup T_i$, onde $T_c \subseteq S \times L_c \times \Gamma \times S, T_r \subseteq S \times L_r \times \Gamma_\perp \times S$ e $T_i \subseteq S \times (L_i \cup \varsigma) \times \sharp \times S$, onde $\sharp \notin \Gamma_\perp$ é um símbolo reservado.

Já um IOVPTS é definido por $\mathcal{J} = \langle S, S_{in}, L_I, L_U, \Gamma, T\rangle$, onde:

- L_I é um conjunto finito de ações de entrada;
- L_U é um conjunto finito de ações de saída;
- $L_I \cap L_U = \emptyset$, e $L = L_I \cup L_U$ é o conjunto de ações; e
- $\langle S, S_{in}, L, \Gamma, T\rangle$ é o VPTS subjacente associado a $\mathcal{J}$.

Seja $t = (p, x, Z, q)$ uma transição de T. Uma transição *push*, $t \in T_c$, significa que uma entrada x está sendo lida quando o controle se move do estado p para q em $\mathscr{S}$, e empilha Z. Já uma transição *pop* onde $t \in T_r$, diz que uma mudança no controle de $\mathscr{S}$ de p para q, $x \in L_r$ é lida e desempilha o símbolo Z. Uma transição do tipo pop pode ser realizada com a pilha vazia, quando o símbolo é $\perp$. Por fim, uma transição simples $t \in T_i$ ocorre quando $x \in L_i$ ou por transição interna quando $t \in T_i$ com $x = \varsigma$. No primeiro caso, t lê um x se movendo de p para q sem modificar a pilha. De forma similar, no segundo caso a pilha também fica inalterada, porém nenhum símbolo de entrada é lido.

O comportamento de um VPTS é dado pela noção de configuração.

Definição 1. *Seja $\mathscr{S} = \langle S, S_{in}, L, \Gamma, T\rangle$ um VPTS. Uma* configuração *de $\mathscr{S}$ é um par $(p, \alpha) \in S \times (\Gamma^\star\{\perp\})$. Quando $p \in S_{in}$ e $\alpha = \perp$, (p, α) é uma* configuração inicial *de $\mathscr{S}$. O conjunto de todas as configurações de $\mathscr{S}$ é dado por $\mathscr{C}_\mathscr{S}$. Seja $(q, \alpha) \in \mathscr{C}_\mathscr{S}$ e $\ell \in L_\varsigma$, escrevemos $(p, \alpha) \xrightarrow{\ell} (q, \beta)$ se existe uma transição $(p, \ell, Z, q) \in T$, tal que:*

1. *$\ell \in L_c$, e $\beta = Z\alpha$;*
2. *$\ell \in L_r$, e ambos (i) $Z \neq \perp$ e $\alpha = Z\beta$, ou (ii) $Z = \alpha = \beta = \perp$;*
3. *$\ell \in L_i \cup \{\varsigma\}$ e $\alpha = \beta$.*

Assim, um movimento simples de $\mathscr{S}$ é representado por $(p, \alpha) \xrightarrow{\ell} (q, \beta)$ quando uma transição $(p, \ell, Z, q) \in T$ é usada neste movimento. Após este movimento simples, $(q, \beta) \in \mathscr{C}_\mathscr{S}$ também é uma configuração de $\mathscr{S}$.

Uma representação gráfica de um VPTS toma uma transição push (s, x, Z, q) com x/Z_+ próximo a aresta correspodente de s para q na figura. De forma semelhante, a transição pop (s, x, Z, q) terá um rótulo x/Z_- próximo a aresta de s à q. As transições simples e intena, $(s, x, \sharp, q)$, terão o rótulo x próximo a aresta correspondente.

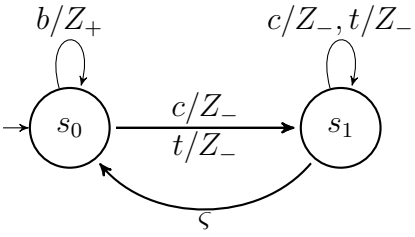

Figure 1. A VPTS $\mathscr{S}_1$, with $L_c = \{b\}$, $L_r = \{c, t\}$, $L_i = \emptyset$.

Exemplo 1. *A Figura 1 representa um VPTS $\mathscr{S}$ com $S = \{s_0, s_1\}$, $S_{in} = \{s_0\}$. Os conjunto de rótulos são $L_c = \{b\}$, $L_r = \{c, t\}$, $L_i = \{\}$, e $\Gamma = \{Z\}$. Existe uma transição push (s_0, b, Z, s_0), as transição pop (s_0, c, Z, s_1), (s_0, t, Z, s_1), (s_1, c, Z, s_1), (s_1, t, Z, s_1), e a transição interna $(s_1, \varsigma, \sharp, s_0)$. O comportamento de $\mathscr{S}$ diz que o símbolo b ocorre tantas vezes quanto necessário, empilhando o símbolo Z. Em seguida, ao menos um c ou t correspondente deve ocorrer, e então Z é desempilhado, ou então vários símbolos c e t ocorrerem enquanto a pilha não estiver vazia. Na sequência, este processo se reinicia com $\mathscr{S}$ voltando o controle para o estado s_0, através de um rótulo internal ς.*

Assim como nos IOLTSs, o conjunto de traces, ou comportamentos, definem a semântica de um VPTS.

Definição 2. *Seja $\mathscr{S} = \langle S, S_{in}, L, \Gamma, T\rangle$ um VPTS e $(p, \alpha), (q, \beta) \in \mathscr{C}_{\mathscr{S}}$.*

1. *Seja $\sigma = l_1, \ldots, l_n$ uma palavra em $L_\varsigma^\star$. σ é um* trace *de (p, α) até (q, β) se existem as configurações $(r_i, \alpha_i) \in \mathscr{C}_{\mathscr{S}}$, $0 \leq i \leq n$, tal que $(r_{i-1}, \alpha_{i-1}) \xrightarrow{l_i} (r_i, \alpha_i)$, $1 \leq i \leq n$, com $(r_0, \alpha_0) = (p, \alpha)$ e $(r_n, \alpha_n) = (q, \beta)$.*
2. *Seja $\sigma \in L^\star$. σ é um* trace observável *de (p, α) à (q, β) em $\mathscr{S}$ se existe um trace μ de (p, α) até (q, β) em $\mathscr{S}$ tal que $\sigma = h_\varsigma(\mu)$.*

Logo, o trace se inicia em (p, α) e termina em (q, β), e a configuração (q, β) é dita alcançável *a partir de (p, α). Quando (q, β) é* alcançável *em $\mathscr{S}$, então ela é alcançável de uma configuração inicial de $\mathscr{S}$.*

Como o símbolo ς pode ocorrer num trace, então quando esses símbolos são removidos, o trace é chamado de observável. Um trace σ de (p, α) to (q, β) é representado por $(p, \alpha) \xrightarrow{\sigma} (q, \beta)$. Quando σ é um trace observável de (p, α) para (q, β), então pode ser representado por $(p, \alpha) \stackrel{\sigma}{\Rightarrow} (q, \beta)$.

2.2. Relação de Conformidade

Uma das abordagens de teste com base em modelos formais é a verificação de conformidade. Uma relação de conformidade é definida com base em um modelo específico. A relação *ioco* (input/output conformance) [Tretmans 2008] é definida sobre o modelo IOLTS e permite a comparação entre uma implementação e sua respectiva especificação IOLTS. Essa comparação é realizada com base nos comportamentos definidos pelos estímulos de entrada e as saídas produzidas desse modelos.

De forma intuitiva, a verificação de conformidade baseada na relação *ioco* consiste basicamente na verificação de dois aspectos: (i) a implementação sob teste pode produzir apenas saídas que também são produzidas pela especificação após uma sequência de estímulos; (ii) se a especificação produz uma saída após uma sequência de estímulos de entrada, a implementação deve produzir esta mesma saída.

A intuição da relação de conformidade para modelos IOVPTSs segue a mesma lógica, porém num cenário bem mais complexo devido a memória de pilha do modelo associado. A conformidade **ioco**-like para IOVPTSs é que dada uma especificação $\mathscr{S}$ e uma implementação $\mathscr{I}$, $\mathscr{I}$ está em conformidade com $\mathscr{S}$ quando, para qualquer comportamento observável σ de $\mathscr{S}$, qualquer símbolo de saída que $\mathscr{I}$ pode produzir após uma execução sobre σ está, necessariamente, entre os símbolos que $\mathscr{S}$ pode produzir após a execução do mesmo σ. Observe que está intuição é similar a da relação **ioco** para IOLTSs, porém agora os modelos, IOVPTSs, possuem uma memória auxiliar de pilha.

Dessa forma, a relação de conformidade entre uma especificação $\mathscr{S}$ e uma IUT $\mathscr{I}$ diz que se um comportamento σ leva $\mathscr{I}$ para uma configuração da qual pode produzir a saída ℓ, então o mesmo deve ocorrer para $\mathscr{S}$.

3. Objetivos

O objetivo desse projeto é aprimorar as técnicas de teste baseadas em relações de conformidade aplicadas a sistemas reativos. Sistemas dessa natureza, tanto de software quanto de hardware, são caracterizados pela interação contínua com o ambiente externo, tornando a atividade de teste bastante complexa, em especial quando os modelos são mais expressivos usando memória auxiliar.

A ideia é compreender de forma mais aprofundada as técnicas existentes, suas relações de conformidade e os métodos de verificação. Em seguida, o trabalho deve explorar novas técnicas e ferramentas que possam otimizar e aumentar a eficiência do processo de teste usando as relações mencionadas.

Os objetivos específicos do projeto são:

1. Estudar os métodos de verificação de conformidade para sistemas reativos.
2. Pesquisar sobre o método tradicional IOCO, para compreender seu funcionamento e as condições em que pode ser aplicado.
3. Estudar a extensão IOCO para linguagens e suas possíveis aplicações.
4. Pesquisar o modelo de pilha IOVPTS para sistemas reativos complexos e a relação de conformidade associada.
5. Avaliar os cenários práticos de aplicação para as relações de conformidade associados a IOLTS bem como para os IOVPTS.
6. Propor uma arquitetura de teste específica para sistemas reativos de acordo com o estudo realizado.
7. Implementar módulos adicionais com algoritmos aprimorados de teste usando as relações de conformidade estudadas.

4. Procedimentos metodológicos

O desenvolvimento deste trabalho seguirá um conjunto de atividades estabelecidas com o intuito de alcançar os objetivos propostos. As atividades estão previstas para serem realizadas durante o período planejado no cronograma da Tabela 1 e descritas abaixo:

1. Revisão bibliográfica dos modelos formais IOLTS e IOVPTS;
2. Revisão bibliográfica das relações de conformidade para teste de sistemas reativos;
3. Levantamento e estudo da relação de conformidade clássica IOCO;
4. Estudo aprofundado da relação de conformidade IOCO estendida e suas aplicações;

5. Pesquisa e estudo aprofundado da relação de conformidade para modelos IOVPTS e suas aplicações;
6. Análise e avaliação das técnicas estudadas;
7. Elaboração e aprimoramento dos mecanismos de teste para sistemas reativos já implementados;
8. Implementação dos módulos de teste para verificação de conformidade para sistemas que usam memória de pilha;
9. Aplicação prática e análise comparativa das técnicas;
10. Divulgação dos resultados através de publicações.

Atividade	2023					2024						
	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul
1	•											
2	•											
3	•	•										
4		•	•									
5		•	•	•								
6			•	•	•							
7				•	•	•						
8						•	•	•	•			
9								•	•	•	•	
10									•	•	•	•

Table 1. Cronograma de execução

5. Contribuições e/ou Resultados esperados

Entre os resultados esperados desse projeto estão: os estudos aprofundados que abrange os formalismos usados para modelar sistemas reativos com e sem memória; estudo aprofundado das relações de conformidade IOCO clássica, estendida e para modelos com memória de pilha; a proposta de um novo sistema baseado nos modelos conhecidos e estudados; a aplicação prática e estudos de caso usando as técnicas de teste baseada em conformidade; e a avaliação comparativa do desenvolvimento mesmo que parcial das técnicas, bem como o aprimoramento dos algoritmos e estruturas existentes.

Espera-se que os métodos e ferramentas desenvolvidas contribuam para aumentar a confiabilidade e qualidade dos testes para sistemas reativos.

References

Alur, R. and Madhusudan, P. (2004). Visibly pushdown languages. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 202–211, New York, NY, USA. ACM.

Bonifacio, A. L. and Moura, A. V. (2021). Testing asynchronous reactive systems: Beyond the ioco framework. *CLEI Electronic Journal*, 24(13).

Bonifacio, A. L. and Moura, A. V. (2022). Conformance checking and pushdown reactive systems. *CLEI Electronic Journal*, 25(3).

Bonifacio, A. L. and Moura, A. V. (2023). Conformance checking and pushdown reactive systems. *CLEI Electronic Journal*, 25(2).

Bonifácio, A. L. and Moura, A. V. (2017). Test suite completeness and black box testing. *Software Testing, Verification and Reliability*, 27(1-2):e1626. e1626 stvr.1626.

Bonifácio, A. L. and Nascimento, C. Extração de propósitos de teste para modelos reativos.

Hopcroft, J. E. and Ullman, J. D. (1979). *Introduction to Automata Theory, Languages, and Comutation*. Addison Wesley.

Tretmans, J. (2008). Model based testing with labelled transition systems. pages 1–38.