

MGASet - Uma Ferramenta para Apoiar o Teste e Validação de Especificações Baseadas em Máquinas de Estado Finito ¹

Marco Arthur Pereira Candolo

Adenilso da Silva Simão

José Carlos Maldonado

{mapc, adenilso, jcmaldon}@icmc.sc.usp.br

Departamento de Computação

Instituto de Ciências Matemáticas e de Computação

Universidade de São Paulo

Av. Trabalhador São-carlense, 400

Cx. Postal 668

CEP. 13560-970

São Carlos — São Paulo

Resumo

A atividade de teste é fundamental no desenvolvimento de sistemas críticos, tais como controladores de trânsito ou médico-hospitalares, uma vez que a ocorrência de falhas em tais sistemas é inadmissível, devido as graves conseqüências que acarretam para o ambiente em que estão inseridos. Dessa forma, são essenciais a utilização de técnicas rigorosas de teste e validação, e, em especial, a disponibilidade de ferramentas de apoio. Em geral, a especificação do aspecto comportamental desses sistemas é feita com uso de máquinas de transição de estado, tais como Máquinas de Estado Finito, Statecharts, Redes de Petri, etc. Com o objetivo de apoiar o teste e a validação de sistemas reativos especificados com o uso de Máquinas de Estado Finito, desenvolveu-se a ferramenta MGASet, que realiza a verificação de propriedades e a geração de seqüências típicas e de seqüências de teste.

1. Introdução

Máquinas de Estado Finito (MEFs) [5] são utilizadas em diversos contextos, tais como na descrição de aspectos comportamentais em modelos orientados a objetos e na especificação de sistemas reativos. Sistemas reativos podem ser encontrados em diversas áreas e em geral a ocorrência de falhas nesses sistemas é inadmissível. Para se garantir a correção de tais sistemas é necessário aplicarem-se técnicas de teste e validação, dentre as quais está a aplicação de seqüências de teste.

¹Trabalho desenvolvido com suporte financeiro parcial do CNPq, FAPESP, CAPES e Telcordia (EUA).

As operações envolvidas na atividade teste, como as requeridas pelo Método W [4] para a geração de seqüências de teste, são, em geral, muito extensas e inadequadas para serem feitas manualmente, levando-se em conta o tempo, o custo de aplicação e a probabilidade de se cometer erros na própria atividade. A disponibilidade de ferramentas de apoio é, dessa forma, imprescindível para melhorar a qualidade da atividade de teste e reduzir o tempo e o custo de aplicação, aumentando a produtividade.

Nesse contexto, a ferramenta MGASet (Módulo de Geração de Seqüências de Teste) foi desenvolvida por Nakazato [7, 8] em linguagem C, para o ambiente Unix. Essa ferramenta inicialmente foi implementada sem interface gráfica, sendo utilizada por meio de linha de comando.

Este trabalho apresenta uma nova versão da ferramenta MGASet desenvolvida em linguagem Java [2, 3], com base na ferramenta MGASet versão C. Dentre as principais funcionalidades dessa nova versão, incluem-se a verificação de propriedades de MEFs e a geração de seqüências básicas de teste, em particular, as seqüências do método W. Visando a melhorar a usabilidade, a ferramenta possui também recursos para a edição e simulação gráficas de MEFs. A partir deste ponto, quando mencionado MGASet, subentende-se MGASet desenvolvida em Java.

Este artigo está organizado da seguinte forma. Na Seção 2, as funcionalidades da ferramenta MGASet são apresentadas. Na Seção 3, apresentam-se os aspectos operacionais e de implementação da ferramenta. Algumas considerações finais são apresentadas na Seção 4.

2. Funcionalidades da Ferramenta

As principais funcionalidades da ferramenta MGASet são:

- Edição gráfica da MEF;
- Simulação interativa;
- Verificação de propriedades [9, 10];
- Geração de Seqüência de Sincronização [10];
- Geração do Conjunto de Caracterização [10];
- Geração de Seqüências de Teste pelo Método W [4] e
- Aplicação automática das seqüências de teste.

A interface gráfica da ferramenta MGASet integra todas as suas funcionalidades em um único ambiente. Visando a facilitar a sua utilização, a ferramenta possui ajuda passo a passo, que instrui o usuário na sua operação. A janela principal da ferramenta está reproduzida na Figura 1.

Esta seção apresenta os principais aspectos funcionais da ferramenta, abrangendo a edição (Seção 2.1) e simulação (Seção 2.2) de MEFs, a geração de tabelas e verificação de propriedades (Seção 2.3) e a geração de seqüências básicas e de teste (Seção 2.4).

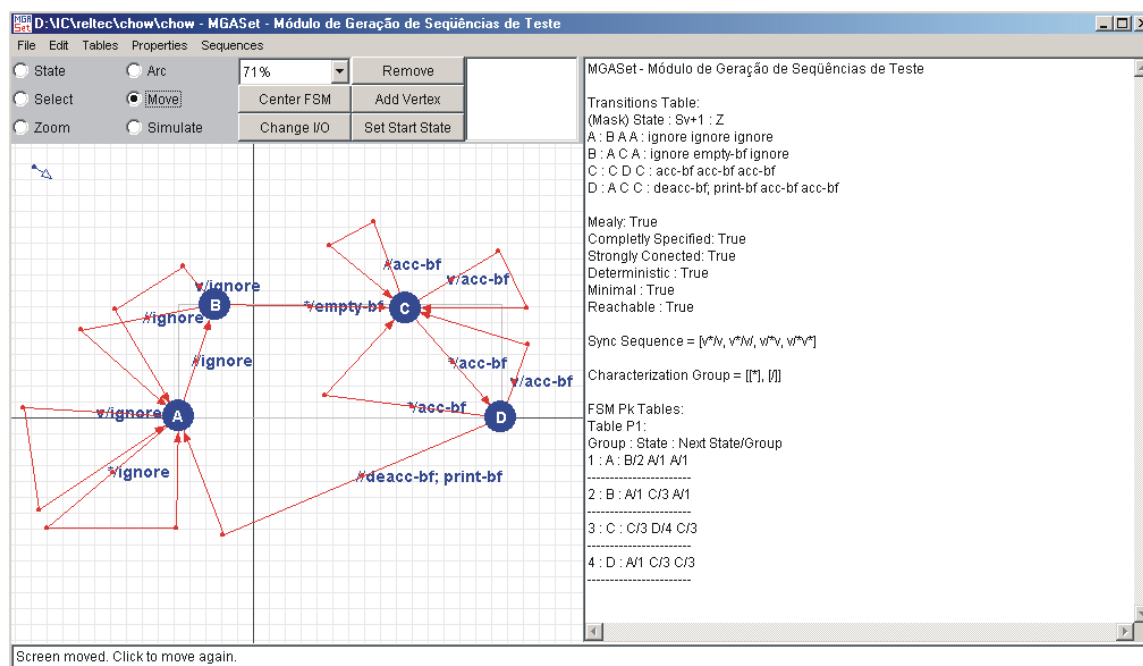


Figura 1: Janela principal da ferramenta MGASet.

2.1. Edição

Para editar uma MEF, o usuário deve definir os elementos dos conjuntos de entrada e saída da mesma, utilizando a janela *FSM I/O Editor*, mostrada na Figura 2.a. A janela pode ser visualizada pelo menu *Edit* → *FSM I/O Editor*. Para fins de documentação, pode-se também definir características de autoria da MEF, tais como o título e o autor, na janela *FSM Data* (Figura 2.b), acionada pelo menu *Edit* → *FSM Data*.

Após especificar os elementos dos conjuntos de entrada e saída, o usuário pode definir os estados e transições da MEF diretamente na janela principal da ferramenta, utilizando os modos de edição disponíveis: **State** (para a inclusão de novos estados), **Arc** (para a inclusão de transições), **Select** (para seleção de um estado, uma transição ou um vértice de transição), **Move** (para se mover um estado, um vértice de uma transição ou toda a MEF) e **Zoom** (para alterar o grau de ampliação da MEF no ambiente de trabalho, gradualmente de 25% até 1600%). Também estão disponíveis para auxiliar no processo de edição os botões **Remove** (para a remoção de um estado, transição ou vértice de transição), **Center FSM** (para centralizar a MEF no ambiente de trabalho), **Add Vertex** (para adicionar um vértice em uma transição), **Change I/O** (para alterar a entrada e a saída de uma transição) e **Set Start State** (para definir o estado inicial da MEF) e as opções pré-definidas de **Zoom** (25%, 50%, 100%, 200%, 400%, 800% e 1600%)

Para o gerenciamento das MEFs editadas, o menu *File* oferece opções para iniciar a edição de uma nova MEF (*New*), salvar a MEF que foi editada (*Save* e *Save As*) e abrir uma MEF (*Open*). Também é possível finalizar a execução da ferramenta (*Exit*).

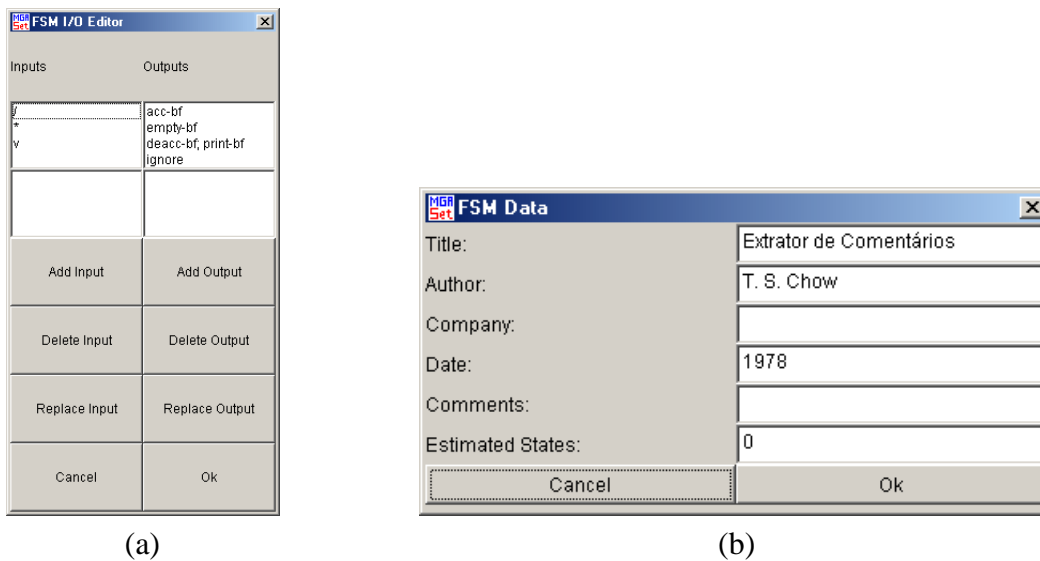


Figura 2: Janelas auxiliares: (a) FSM I/O Editor e (b) FSM Data.

2.2. Simulação

Estando a MEF editada, o usuário pode simular o seu comportamento por meio de uma seqüência de transições, verificando por quais estados a MEF transitará e quais serão as saídas geradas. Para realizar a simulação da MEF, o modo **Simulate** deve ser ativado. Em seguida, o primeiro estado da simulação deve ser selecionado (por *default*, a ferramenta seleciona o estado inicial da MEF). As transições a serem executadas devem, então, ser selecionadas interativamente, uma a uma. As saídas geradas são apresentadas na parte direita da ferramenta, a cada transição executada.

2.3. Geração de Tabelas e Verificação de Propriedades

Como produtos intermediários no processo de teste e validação de especificações de MEFs a ferramenta possui a função de gerar a Tabela de Transição de uma MEF, suas Tabelas P_k e verificar propriedades da MEF.

A Tabela de Transição de uma MEF é a sua especificação na forma tabular, gerada através do menu `Tables → Transition Table`. As Tabelas P_k são utilizadas para a geração de seqüências de teste do Método W e podem ser obtidas acionando-se o menu `Tables → Pk Tables`.

Para gerar seqüências básicas e de teste, é necessário verificar se a MEF em questão apresenta determinadas propriedades. Além de serem utilizadas internamente para verificar a aplicabilidade dos algoritmos, tais propriedades podem ser visualizadas pelo usuário através do menu `Properties`. As propriedades verificadas pela ferramenta são (a definição de cada uma dessas propriedades pode ser encontrada em [9, 10]):

- Mealy;
- Completamente Especificada;

- Fortemente Conectada;
- Determinística;
- Minimal e
- Alcançabilidade.

2.4. Geração de Seqüências Básicas e de Teste

A ferramenta MGASet possibilita a geração de seqüências básicas e seqüências de teste. As seqüências básicas geradas pela ferramenta são: a **Seqüência de Sincronização** (seqüência que leva a MEF a um estado final especificado), a **Seqüência de Distinção** (seqüência que produz saídas diferentes para a sua aplicação em cada um dos estados da MEF) e o **Conjunto de Caracterização** (conjunto de seqüências tal que os símbolos de saída observados após a aplicação dessas seqüências, numa ordem fixada, sobre cada um dos estados são diferentes).

As seqüências de teste são geradas com base no Método W, proposto por Chow [4]. O critério de teste W verifica a estrutura de controle da MEF. Para a sua aplicação é necessário que a MEF possua as propriedades: minimal, determinística, Mealy, completamente especificada e alcançabilidade. O Método W baseia-se na geração de seqüências de teste, que devem ser aplicadas à MEF, e na análise das saídas obtidas. Após terem sido geradas, as seqüências de teste são então simuladas, uma a uma. A ferramenta MGASet automatiza completamente esse processo.

3. Aspectos Operacionais e de Implementação

A ferramenta foi desenvolvida utilizando-se o *Java Development Kit 1.1.8*. Um dos benefícios da utilização da linguagem Java é a obtenção de uma ferramenta portátil, podendo ser utilizada em diversos sistemas operacionais, tais como Linux e Windows. Além disso, com a utilização de uma linguagem orientada a objetos, buscou-se facilitar a manutenção da ferramenta e a inclusão de eventuais extensões.

O código da ferramenta está dividido em 23 módulos, totalizando cerca de 7000 linhas na versão atual, assim distribuídas: a edição e simulação representam 41,4% do código, a interface da ferramenta, incluindo suas janelas, menus, botões, entre outros, representa 16,6% do código e os algoritmos utilizados para o cálculo dos produtos finais e intermediários da ferramenta, 42,0% do código.

Quando comparada à ferramenta MGASet versão desenvolvida em linguagem C [8], os aprimoramentos operacionais que se destacam na ferramenta desenvolvida em Java incluem:

- A disponibilidade de um ambiente gráfico para edição da MEF, incluindo recursos tais como *zoom*, movimentação de elementos e gerenciamento de elementos de entrada e saída;
- A integração dos processos de edição, simulação e teste em um mesmo ambiente, facilitando a análise das conseqüências de eventuais alterações na MEF e

- A inclusão de características de autoria, melhorando a documentação da MEF e possibilitando que comentários do usuário possam ser registradas no próprio arquivo da máquina.

4. Considerações Finais

Neste trabalho foi apresentada a ferramenta MGASet, desenvolvida para apoiar a validação de especificações baseadas em Máquinas de Estado Finito. A motivação do desenvolvimento de ferramentas de apoio como a MGASet é a busca de custos menores, tempos mais curtos e menos erros na atividade de teste, visando ao aumento da produtividade.

Como próximos passos nesta linha de trabalho, incluem-se o aprimoramento da ferramenta com a implementação da geração de outras seqüências de teste como a proposta por Gönenç [6] e a inclusão de algoritmos para a minimização do conjunto de teste [1].

Referências

- [1] ANIDO, R.; CAVALLI, A.; LIMA, L. P.; YEVTUSHENKO, N. *Test suite minimization for testing in context*. Rel. Téc., Instituto de Computação, Universidade de Campinas, (draft version), 2000.
- [2] CANDOLO, M. A. P.; SIMÃO, A. S.; MALDONADO, J. C. MGASet-Java: Geração de Seqüências de Teste para Validação de Especificações Baseada em Máquina de Estado Finito. In: *Simpósio Internacional de Iniciação Científica da USP - SIICUSP*, 2000.
- [3] CANDOLO, M. A. P.; SIMÃO, A. S.; MALDONADO, J. C. *Manual do Usuário da Ferramenta MGASet-Java*. Rel. Téc. (em elaboração), ICMC, São Carlos, 2001.
- [4] CHOW, T. S. Testing Software Design Modeled by Finite State Machines. *IEEE Transactions on Software Engineering*, v. 4, n. 3, p. 178–187, 1978.
- [5] GILL, A. *Introduction to the Theory of Finite State Machines*. New York : McGrall Hill, 1962.
- [6] GÖNENÇ, G. A Method for Design of Fault-Detection Experiments. *IEEE Transactions on Computers*, v. 19, n. 6, p. 551–558, 1970.
- [7] NAKAZATO, K. K. *Módulo de geração de seqüências de teste baseada em máquinas de estado finito*. Dissertação de Mestrado, ICMC/USP, São Carlos/SP, 1995.
- [8] NAKAZATO, K. K.; ALEXANDRINO, M.; MALDONADO, J. C.; FABBRI, S. C. P. F.; MASIERO, P. C. MGASet — Módulo de Geração de Seqüências de Teste. In: *Anais da Sessão de Ferramentas do IX Simpósio Brasileiro de Engenharia de Software*, Recife, Pernambuco, 1995, p. 479–482.
- [9] NAKAZATO, K. K.; MALDONADO, J. C.; FABBRI, S. C. P. F.; MASIERO, P. C. *Propriedades de máquinas de estado finito relevantes para critérios de geração de seqüências de teste*. Rel. Téc. 27, ICMC, São Carlos, 1994.
- [10] NAKAZATO, K. K.; MALDONADO, J. C.; FABBRI, S. C. P. F.; MASIERO, P. C. *Seqüências básicas de máquinas de estado finito*. Rel. Téc. 28, ICMC, São Carlos, 1994.