Título Proposta

Marino Souza S.[1]

[1]Departamento de Ciência da Computação – Instituto de Matemática
Universidade Federal Bahia (UFBA) 40.170-110 – Salvador – BA – Brazil

marino@dcc.ufba.br

1. Introdução

Quando trabalhamos com softwares de situações críticas, onde prejuízos financeiros ou danos a vidas estão em risco, erros são inadmissíveis e se faz necessário o uso de verificação formal para garantir que aquele software crítico é correto. Entretanto, é preciso assumir que a especificação daquele software já era correta, e esta pode acabar possuindo inconsistências iniciando a inserção de erros no software antes mesmo da sua implementação.

A técnica formal proposta por [CLARK] torna possível a verificação automática de um modelo de software, permitindo que na etapa de especificação do programa o projetista possa encontrar incoerências, a serem corrigidas na sua modelagem, ainda nesta fase. Aqui a especificação é feita através de estruturas de Kripke e as propriedades desejadas no modelo podem ser escritas em alguma lógica formal, geralmente lógicas temporais como a CTL. Quando esta verificação retorna que o modelo não respeita a propriedade requerida, o projetista obtém um contra-exemplo e deve, então, revisar o modelo manualmente.

Em alguns casos em que não é possível conhecer todas as informações do sistema a modelagem pode ficar com informação parcial. Nesta situação, é necessário recorrer a modelos mais expressivos que as estruturas de Kripke, capazes de explicitar estas indefinições. No trabalho de [GRUMBERG] é proposto um verificador de modelos que faz uso de Estruturas de Transição Modal de Kripke (KMTS) para a modelagem, os quais tornam possíveis a interpretação de um conjunto de estruturas de Kripke através de suas indeterminações, como é mostrado por [ANDRADE; GUERRA], entretanto nesta interpretação que lida com informação incompleta também se faz necessário recorrer ao uso de lógica de três valores, e o verificador de [GRUMBERG] não lida muito bem com esta semântica.

1.1. Trabalhos anteriores

No trabalho de IC com o título `Uma Ferramenta para Refinamento de Modelos KMTS` concluído e apresentado durante o `SEMENTE`[1], pelo graduando desta proposta, é apresentado uma ferramenta de verificação de modelos com lógica de três valores e modelos KMTS, que são estruturas capazes de representar um conjunto de estruturas de Kripke, onde as respostas para a verificação poderia ser uma das três: Verdadeiro (T), Falso (F) ou Indefinido ($\perp$). Nos dois últimos casos é necessário revisar o modelo, e a ferramenta realizava a revisão para o caso $\perp$, usando a abordagem proposta por [RIBEIRO] chamada de refinamento. Este refinador tem um desempenho melhor que

[1]Seminário Integrado de Ensino, Pesquisa e Extensão da UFBA

o proposto em [ANDRADE; GUERRA] para o caso médio, uma vez que a natureza deste problema é da classe NP-Completo.

Quando a verificação retornava F significava que nenhuma das estruturas de Kripke presentes no KMTS atendiam à fórmula, e no caso de $\perp$ tínhamos que algumas estruturas atendiam e outras não. Desta forma, o processo de refinamento era revisar o modelo KMTS e retornar um ou mais modelos que representassem apenas as estruturas de Kripke presentes no modelo anterior que satisfaziam a propriedade buscada. Contudo, a verificação de modelos aplicada baseava-se no trabalho de [GRUMBERG] que além de não ter como finalidade verificar um KMTS interpretado como um conjunto de estruturas de Kripke, ainda retornava o que chamamos de Falsos Indefinidos,[RIBEIRO] que era a resposta $\perp$ quando deveria ser F ou T, sendo necessário executar o refinamento em um modelo KMTS que não encontraria nenhuma estrutura de Kripke ou que encontraria todas elas, respectivamente.

1.2. A Proposta Atual

No trabalho de [RIBEIRO] é proposto uma técnica de verificação de modelos voltada para Modelos KMTS representando um conjunto de estruturas de Kripke que tem melhor desempenho em comparação ao verificador usado na ferramenta anterior, no que diz respeito os Falsos Indefinidos, cuja implementação ainda não é feita. Assim o objetivo deste trabalho é implementar o verificador proposto e integrá-lo à ferramenta de refinamento de modelos KMTS do trabalho anterior, tendo em vista que a verificação de modelos KMTS na semântica desejada é um problema de complexidade NP-completo.

2. CTL x KMTS

A lógica formal CTL, Árvore de Lógica Computacional (Computational Tree Logic), é usada para expressar as propriedades buscadas no processo de verificação. [HUTH]

Definição 1: Seja l um literal, a fórmula CTL ϕ em sua forma normal negativa é definida como:

$$\phi ::= \top \mid F \mid l \mid (\phi \vee \phi) \mid (\phi \wedge \phi) \mid EX\phi \mid AX\phi \mid$$
$$E[\phi U \phi] \mid A[\phi U \phi] \mid E[\phi R \phi] \mid A[\phi R \phi]$$

Na Definição 1, A e E são operadores de caminhos futuros significando para todos os caminhos e existe um caminho possível, respectivamente. Os operadores X, U e R significam: próximo estado, *until* (o ϕ esquerdo é verdade até que o ϕ direito seja verdade) e *release* (vale como o operador U dual)

As estruturas de Kripke são modelos finitos de transição de estados utilizados na especificação de comportamento de sistemas. Contudo, os primeiros estágios de uma especificação de sistemas, normalmente, possuem informações parciais e incompletas, tornando o poder expressivo dessas estruturas insuficientes para lidar com essa categoria de informações. Para contornar, modelos mais expressivos são utilizados, como as estruturas modais de Kripke (Kripke Modal Transitions Systems). KMTS, são modelos finitos de transição de estados capazes de representar estas informações parciais incompletas, de forma que essas características lhe dão poder expressivo capaz de representam um conjunto de estruturas de Kripke.

Definição 2: Seja AP um conjunto de proposições atômicas e $Lit = AP \cup \{\neg p \mid p \in AP\}$ o conjunto de literais sobre AP. Uma Estrutura de Transição Model de Kripke (KMTS) é uma tupla $M = (AP, S, R^+, R^-, L)$, onde S é um conjunto de estados finitos, $R^+ \subseteq S \times S$ e $R^- \subseteq S \times S$ são transições de relação, tal que $R^+ \subseteq R^-$, e $L : S \to 2^{Lit}$ é uma function parcial de rotulação, que assina p ou $\neg p$ num estado s. As relações R^+ e R^- correspondem às transições *must* e *may* respectivamente.

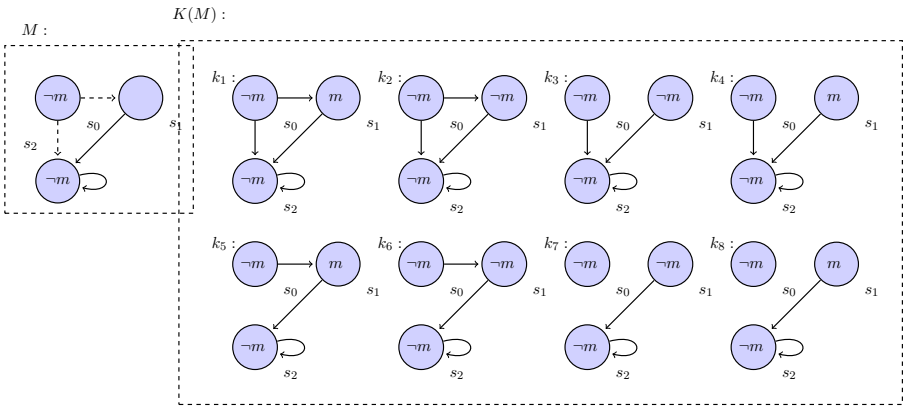

Figure 1. Expanão K(M) de um KMTS M. As linhas pontilhadas representam as transições *may* e as linhas mais sólidas representam as transições *must*.

Desta forma, podemos chamar de $K(M)$ o conjunto de estruturas de Kripke representadas pelo KMTS M. A Figura 1 ilustra um KMTS M e seu conjunto expansão $K(M)$. Uma vez que o estado s_1 de M não está rotulado com m nem $\neg m$, e também existem duas transições tipo *may* (estados: $s_0 \to s_1$ e $s_0 \to s_2$), M nos leva a um conjunto de oito modelos CTL (ou seja 2_x onde x é a quantidade de indeterminações do KMTS).

3. Abordagem da Verificação de Modelos

A técnica de *Model Checking*, abordada aqui, faz uso de modelos KMTS para abstrair um sistema que queremos verificar uma certa propriedade. As propriedades buscadas são expressas na lógica formal CTL.

Quando o algoritmo de *Model Checking* é executado para verificar se a propriedade é satisfeita no modelo, temos três resultados possíveis: Verdadeiro (T), Falso (F) ou Indefinido ($\perp$) Quando o resultado é T, temos que todos os modelos de Kripke representados pelo KMTS satisfazem a propriedade. O resultado F implica que todos os modelos não satisfazem a propriedade. No caso de $\perp$, é preciso aplicar um refinamento a fim de selecionar os modelos que satisfazem a propriedade, quando estes existem, e então retornar um outro KMTS M_1, ou mais de um, que represente o conjunto destes modelos refinados.

O trabalho anterior apresentado em [SEMENTE] implementava o verificador de modelos adaptado do proposto em [GRUMBERG], sendo assim, no caso em a verificação de M retorna $\perp$ ainda é possível que todos os modelos do conjunto K(M) satisfaçam a propriedade requerida, ou ainda que nenhum dos modelos a satisfaça. Neste caso dizemos que a verificação retornou um Falso Indefinido, fazendo-se necessário executar a verificação em cada modelo do conjunto a fim de descobrir qual o resultado correto.

... adicionar exemplo de KMTS VS Fórmula que retorna Falso indefinido...

O verificador de modelos que este trabalho tem finalidade de implementar difere na semântica composicional sobre as operações de conjunção e disjunção, definidas em [RIBEIRO]

... adicionar sessão sobre as operações composicionais ...

4. Metodologia

5. Código