



**Universidade Federal de Pernambuco
Centro de Informática
Graduação em Ciência da Computação**

Síntese de programas utilizando a linguagem Alloy

Proposta de Trabalho de Graduação

Aluno: João Pedro Marcolino Cordeiro (jpmc@cin.ufpe.br)

Orientador: Alexandre Cabral Mota (acm@cin.ufpe.br)

Recife, Abril de 2017

Resumo

Desenvolvimento de sistemas de software ou hardware é um processo complexo e caro. Nos últimos anos, abordagens formais para especificação, desenvolvimento e verificação de software vêm sendo estudadas e desenvolvidas com o objetivo de diminuir o custo e esforço total requerido por esse processo. Uma dessas abordagens é a síntese de programas, que tem por objetivo automatizar a geração de programas executáveis a partir de especificações.

Este trabalho se propõe a implementar uma ferramenta para sintetizar programas na linguagem imperativa IMP utilizando a linguagem de especificação Alloy, além de apresentar exemplos de algoritmos que podem ser gerados com essa ferramenta.

Sumario

1. Contexto
2. Objetivos
3. Cronograma
4. Referencias
5. Possíveis avaliadores
6. Assinaturas

1. Contexto

Métodos formais são técnicas e ferramentas usadas para a especificação, design e verificação de sistemas de Software ou Hardware que vem sendo desenvolvidas e estudadas desde a década de 60 [2]. Baseados em formalismos matemáticos, métodos formais comumente são usados para garantir certas propriedades sobre um sistema, como conformidade de um produto final com sua especificação inicial e a correteude dessa especificação. Nos últimos anos, uma aplicação bastante abordada na literatura que usa essas técnicas é a síntese de programas.

Ferramentas e métodos de síntese de programas tem como objetivo automatizar parte do (ou todo) processo de desenvolvimento de software. Em termos mais claros: dado uma especificação, síntese de programas procura gerar um programa executável que atende essa especificação [1]. Como essas especificações são escritas e também o conceito de “executável” varia de acordo com a abordagem de síntese utilizada.

Existem diferentes maneiras para avaliar e classificar as abordagens de síntese de programas presentes na literatura. Em 2004, Basin *et al.* [1] apresentaram uma dessas possíveis classificações. Segundo eles, abordagens podem ser divididas em três categorias com respeito à técnica de síntese utilizada: síntese construtiva/dedutiva, síntese guiada por *schemas* e síntese indutiva. Também é apresentada uma comparação entre as abordagens analisadas e as especificações dos programas que são gerados por ela. Dois exemplos dessas especificações são: contratos, que são restrições e propriedades sobre os estados iniciais e finais do programa, e exemplos, que são pares de entrada e saída esperada do programa a ser sintetizado.

Cada ferramenta de síntese também aborda pelo menos duas linguagens. Uma para implementação do processo de síntese em si, normalmente uma linguagem formal baseada em lógica, como Alloy; e outra utilizada para representação dos programas sintetizados, que varia de acordo com o tipo de programa que será gerado. Se os programas sintetizados forem imperativos, por exemplo, pode-se usar uma linguagem imperativa simples, como IMP.

IMP é uma linguagem de programação imperativa com uma estrutura extremamente simples quando comparada à outras linguagens imperativas mais conhecidas, como C [6]. Seu corpo sintático é composto de números, valores booleanos, nomes de variáveis, expressões aritméticas, expressões booleanas e comandos. Em sua definição pura, ela não suporta operadores aritméticos comuns como divisão e resto. Para comparações de valores, apenas igualdade e menor ou igual são descritos. Os comandos definidos são *skip*, atribuição, composição, *if-then-else*, e *while*.

Alloy, publicada por Daniel Jackson em 2002 [3] e abordada em mais detalhes em seu livro em 2006 [4], é tanto uma linguagem para descrever estruturas como também uma ferramenta para explorá-las. Um modelo na linguagem Alloy é um conjunto de restrições que descrevem implicitamente um conjunto de estruturas. O analisador Alloy é uma ferramenta que toma como entrada um modelo Alloy e dá como saída exemplos de estruturas que atendem as restrições descritas no modelo e também é capaz de checar propriedades desse modelo através de contraexemplos.

Em 2015, nove anos depois da publicação do livro que descreve Alloy, Milicevic *et al.* [5] publicaram um artigo apresentando Alloy*, uma segunda linguagem que tem como base a sintaxe de Alloy mas estende sua semântica para suportar quantificadores de alta ordem.

2. Objetivo

Este trabalho tem como objetivo analisar e compreender as abordagens mais comuns para síntese de programas utilizadas atualmente na literatura. Além disso, será desenvolvida uma ferramenta, implementada utilizando Alloy, a partir de uma ferramenta de síntese de programas na linguagem IMP já existente, escrita em Alloy*. Também serão apresentados exemplos de algoritmos sintetizados com a ferramenta desenvolvida.

3. Cronograma

Atividade	Março	Abril	Maio	Junho	Julho
Estudo da linguagem Alloy e Alloy*	■	■			
Revisão bibliográfica		■	■		
Elaboração da proposta		■	■		
Desenvolvimento da ferramenta em Alloy		■	■	■	
Síntese de algoritmos utilizando a ferramenta				■	■
Elaboração do relatório final				■	■
Apresentação					■

4. Referências

- [1] D. Basin, Y. Deville, P. Flener, A. Hamfelt e J. F. Nilsson, "Synthesis of programs in computational logic," in *Program Development in Computational Logic*. Berlin, Heidelberg: Springer, 2004, pp. 30-65.
- [2] J. McCarthy, "A basis for a mathematical theory of computation," presented at the *western joint IRE-AIEE-ACM computer conf.*, Los Angeles, California, May 9-11, 1961, pp 225-238.
- [3] D. Jackson. "Alloy: a lightweight object modelling notation," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 11, no. 2, pp 256-290, Apr 2002
- [4] D. Jackson, *Software Abstractions: Logic, Language, and Analysis*. Cambridge, United Kingdom: MIT Press, 2006.
- [5] A. Milicevic, J. P. Near, E. Kang and D. Jackson, "Alloy*: A general-purpose higher-order relational constraint solver." in *37th IEEE Int. Conf. on Software Engineering*, Florence, Italy, 2015, pp. 609-619.
- [6] G. Winskel, *The Formal Semantics of Programming Languages: An Introduction*. Cambridge, United Kingdom: MIT Press, 1993.

5. Possíveis avaliadores

Juliano Iyoda

Márcio Cornélio

Ricardo Martins

6. Assinaturas

João Pedro Marcolino Cordeiro
Aluno

Alexandre Cabral Mota
Orientador