

Geração de argumentos válidos da Lógica Proposicional

Filipe da Silva Oliveira¹

Elthon Oliveira²

Ciência da Computação/UFAL, Arapiraca, AL

1 Introdução

MOOCs - *Massive Open Online Courses* são cursos *online* abertos e acessíveis a qualquer pessoa com acesso à internet [1]. Esses ambientes possuem uma grande quantidade de estudantes, o que torna inviável a criação de exercícios únicos para cada aluno. Neste trabalho é apresentada uma abordagem à geração automática de argumentos válidos da Lógica Matemática Proposicional. É utilizada uma combinação de especificações matemáticas com uma técnica de Síntese de Programas chamada Geração por Esboço [3]. Síntese de Programas consiste em gerar automaticamente um programa que satisfaça certas especificações. Na Geração por Esboço, é fornecida a uma linguagem subjacente a um esboço de um programa juntamente com uma especificação. A partir disso, é obtido o melhor programa que satisfaça ao esboço e à especificação.

Por programa, neste contexto, entenda-se um argumento válido da Lógica Proposicional. Um Argumento é composto por um conjunto de premissas e uma conclusão, que são *fórmulas bem formadas* - *fbf's*. Formalmente uma *fbf* é definida como: (i) toda letra do alfabeto é uma *fbf*, conhecido como átomo; (ii) se α é uma *fbf*, $\neg(\alpha)$ também é; e (iii) se α e β são *fbf's*, α o β também é, onde o corresponde a alguma das operações: \wedge | \vee | \rightarrow | \leftrightarrow .

As especificações matemáticas usadas para definir um argumento são escritas utilizando a linguagem Alloy [2], que é baseada em lógica de primeira ordem e capaz de descrever estruturas e relações. Há um suporte ferramental chamado *Alloy Analyzer*³, que é um solucionador que analisa estruturas, relações, propriedades e restrições em busca de modelos que as satisfaçam.

2 Especificações e Algoritmo

Na Figura 1 é apresentado parte do código Alloy que especifica formalmente um argumento. O código ilustrado na Figura 1(a) especifica que há *fbf's* unárias e binárias. Na Figura 1(b) é especificado que uma *fbf* pode ser: átomo, negação, conjunção, disjunção, implicação ou biimplicação. Já na Figura 1(c), é especificada a estrutura da regra de inferência denominada *Modus Ponens* e como se dá seu funcionamento. Este último é definido como um **fato**: $(\alpha \rightarrow \beta) \wedge \alpha \Rightarrow \beta$.

O algoritmo para geração de argumentos é brevemente descrito a seguir. É fornecido ao *Alloy Analyzer* toda a especificação contendo os elementos, estruturas, propriedades e restrições sobre o que pode ser um argumento logicamente válido. O *Alloy Analyzer* é integrado a outra ferramenta que recebe do usuário humano informações sobre características que os argumentos gerados devem ter em comum. A partir da entrada fornecida pelo usuário e das especificações em Alloy,

¹filipe.oliveira@arapiraca.ufal.br - Bolsista PIBIC/CNPq-Af do Edital Propep/UFAL 2020-2021.

²elthon@arapiraca.ufal.br

³<https://alloytools.org/>

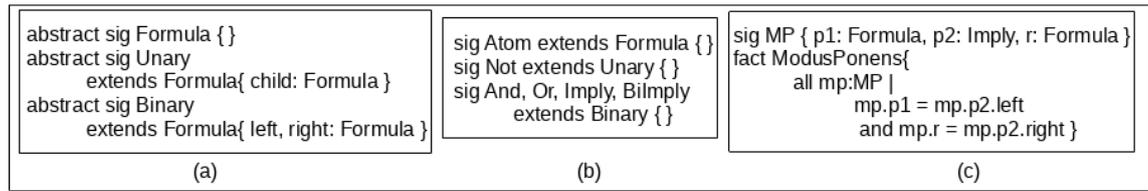


Figura 1: Trechos da especificação em Alloy para argumento.

o analisador retorna diversos modelos que serão instanciados na forma de argumentos. Na Figura 2 é ilustrado parte deste processo. Na Figura 2(a) é apresentado trecho de código Alloy gerado a partir das entradas fornecidas pelo usuário. O usuário escolheu gerar argumentos cujas premissas apresentam dois átomos distintos e que precisem apenas da aplicação do *Modus Ponens* nos processos de prova. Na Figura 2(b) é apresentado um dos vários modelos gerados a partir do código apresentado no passo anterior. Por último, na Figura 2(c), é apresentado um argumento válido instanciado a partir do modelo obtido no passo anterior. Os argumentos são instanciados por meio de uma API (*Interface de programação de Aplicações*, em inglês) desenvolvida em Java que acessa e utiliza a API da ferramenta *Alloy Analyzer*.

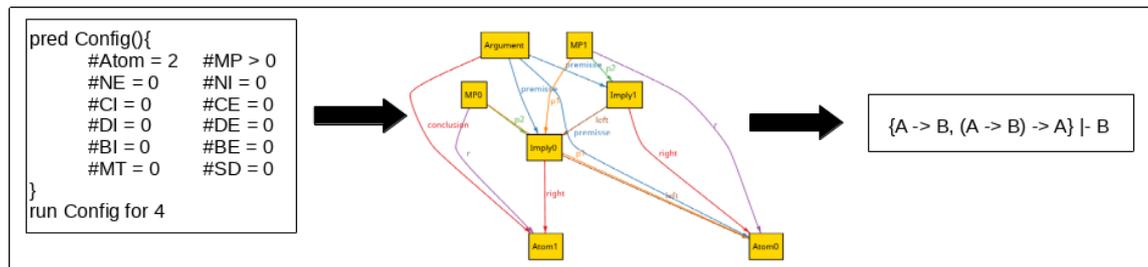


Figura 2: Passos na geração de um argumento.

3 Resultados e conclusões

A abordagem apresentada sintetiza argumentos válidos e com características similares a partir de especificações formais fornecidas. Apesar de existir um certo grau de aleatoriedade na geração, a validade dos argumentos é garantida por construção devido às especificações em Alloy. É possível então gerar questões únicas e com mesmo nível de dificuldade para ser utilizadas em ambientes de MOOCs, inicialmente para a disciplina de Lógica Matemática. Por fim, é importante ressaltar que existe um aplicativo móvel em desenvolvimento que poderá ser utilizado em qualquer celular com sistema Android.

Referências

- [1] Fassbinder, A., Delmiro, M. E. e Barbosa, E. F. Construção e Uso de MOOCs: Uma Revisão Sistemática, *TEMA*, 25:332–341, 2014. DOI: <http://dx.doi.org/10.5753/cbie.sbie.2014.332>.
- [2] Jackson, D. *Software Abstractions: logic, language, and analysis*. MIT press, 2012.
- [3] Solar-Lezama, A. *Program synthesis by sketching*. University of California, Berkeley, 2008.