

1.03.04 - Ciência da Computação / Sistemas de Computação.

GERAÇÃO DE QUESTÕES DE LÓGICA UTILIZANDO A LINGUAGEM DE GERAÇÃO DE MODELOS ALLOY

Filipe da Silva Oliveira^{1*}, Elthon Allex da Silva Oliveira²

1. Estudante de Ciência da Computação na Universidade Federal de Alagoas (UFAL) – *Campus Arapiraca*
2. Professor da UFAL Campus Arapiraca – Ciência da Computação/Orientador.

Resumo

A elaboração de problemas com características específicas da disciplina de Lógica é tida como uma tarefa tediosa, principalmente quando há a necessidade por problemas únicos para cada estudante. Neste trabalho é apresentada uma abordagem para geração automática de fórmulas e argumentos da Lógica Proposicional. Para possibilitar a geração automática mantendo características comuns, foi feito uso da linguagem de geração de modelos Alloy, que facilita na utilização da técnica conhecida como Geração por Esboço, advinda da área de Síntese de Programas. Como prova de conceitos, foi desenvolvido um sintetizador capaz de gerar as fórmulas e os argumentos a partir de informações chaves fornecidas pelo usuário. A abordagem utilizada gerou excelentes resultados tanto na síntese de fórmulas como de argumentos.

Autorização legal: Não se aplica.

Palavras-chave: Síntese de Fórmulas; Síntese de Argumentos; Lógica Proposicional.

Apoio financeiro: Programa de Bolsa de Iniciação Científica - PIBIC/UFAL.

Trabalho selecionado para a JNIC: Pró-reitoria de Pesquisa e Pós-Graduação - Propep/UFAL.

Introdução

Síntese de Programas é a tarefa de encontrar automaticamente um programa numa determinada linguagem de programação que satisfaça a intenção do usuário descrita na forma de alguma especificação [1]. Este problema tem sido considerado o *santo graal* da Ciência da Computação. Contudo, atualmente dois problemas se destacam nesta área: a ambiguidade na descrição da intenção do usuário, e um espaço de pesquisa de programas quase sempre enorme. Apesar destes desafios inerentes ao problema, pesquisadores no campo de Síntese de Programas têm desenvolvido muitas técnicas diferentes que permitem sua aplicação em diversos domínios de aplicação da vida real.

Dentre os domínios de aplicação, destacam-se engenharia de software, descoberta biológica, programação de usuário final e educação assistida por computador. Este projeto de pesquisa foca neste último, mais especificamente, na síntese de elementos da Lógica Proposicional. Esses elementos podem ser usados como problemas na educação formal de alunos em disciplinas de Lógica. Sabe-se que gerar novos problemas que tenham características específicas de solução (como um determinado nível de dificuldade ou que envolva o uso de um determinado conjunto de conceitos) é uma tarefa tediosa para o professor. Mais claramente, foi utilizado no projeto a linguagem de geração de modelos Alloy [2] junto a abordagem de Geração de Esboço [3, 4] para conseguir sintetizar os elementos da Lógica Proposicional. A ideia central deste projeto foi investigar a área de Síntese de Programas e aplicá-la. Este cenário foi o meio para a investigação, entendimento dos problemas envolvidos e as soluções existentes na área de pesquisa em questão.

Como objetivo principal deste trabalho teve-se a síntese de fórmulas e argumentos válidos da Lógica Proposicional utilizando a técnica de Geração por Esboço da Síntese de Programas. Objetivos específicos: (i) revisar os problemas envolvidos no campo de Síntese de Programas e as técnicas para tratar os problemas; (ii) aprender as técnicas mais adequadas à síntese de fórmulas e argumentos da maneira proposta para este projeto; (iii) realizar um levantamento, estudo e análise comparativa do suporte ferramental existente para Síntese de Programas; (iv) aprimoramento do sintetizador de fórmulas e argumentos da Lógica Proposicional.

Metodologia

Estudo dos conceitos de Teoria da Computação e Compiladores. Nesta etapa foi realizado a revisão de alguns conceitos da área da computação, com um foco especial na base teórica da pesquisa científica em questão. Sendo assim, o estudo foi baseado em conceitos de Teoria da Computação e Compiladores, duas disciplinas do curso de Ciência da Computação. Sendo assim, os conceitos abaixo foram estudados e revisados por parte do bolsista: Expressões Regulares; Gramática Regular; Gramática livre de contexto; e DSL - Linguagem Específica de Domínio. Algumas implementações realizadas com abordagem desenvolvida nesta pesquisa utilizam tais conceitos como base.

Planejamento e execução de seminários. Esta etapa teve como objetivo a elaboração e execução de

seminários. Foram elencados os principais conceitos estudados ao longo da pesquisa em apresentações elaboradas para comunidade acadêmica. Dentre as apresentações realizadas estão: apresentações individuais para alguns alunos e apresentações em eventos internos da universidade.

Estudo das ferramentas de software existentes sobre Síntese de Programas. Neste ponto, analisou-se as ferramentas existentes relacionadas à Síntese de Programas. As ferramentas encontradas e escolhidas para serem apresentadas foram as que funcionam como base para geração de elementos, utilizando as técnicas de Síntese de Programas.

Aprimoramento/desenvolvimento do sintetizador de fórmulas e argumentos. Esta etapa da pesquisa foi a que exigiu mais tempo neste ciclo, pois ela foi executada do início ao fim. Primeiramente, foi realizado um estudo direcionado a linguagem de geração de modelos Alloy. No estudo, foi observado que a linguagem poderia auxiliar no processo de geração dos elementos da Lógica Proposicional. Então, foi decidido investir tempo em alguns testes com a linguagem. Dentre esses testes, especificações para geração de fórmulas e argumentos foram criados e testados, conseguindo com eles gerar tais elementos. As especificações foram aperfeiçoadas e utilizadas como base da ferramenta desenvolvida com a abordagem. Foram criados alguns mecanismos capazes de melhorar a qualidade da ferramenta, como uma API - Interface de Programação de Aplicações. Tal API é a camada intermediária entre a linguagem Alloy e a ferramenta gráfica desenvolvida, totalizando três componentes desenvolvidos.

Escrita de artigos, confecção de pôsteres, etc. Nesta etapa, foi escrito alguns trabalhos científicos com intuito de divulgar os recursos que foram desenvolvidos neste projeto. Então, tal etapa foi realizada a partir da escrita de artigos científicos com os seguintes resultados: especificações em Alloy, uso da ferramenta Alloy como parte das técnicas de Síntese de Programas utilizadas na abordagem desta pesquisa, a API desenvolvida, e um aplicativo móvel desenvolvido com o objetivo de popularizar o alcance do poder de síntese no contexto de geração de artefatos da Lógica Proposicional.

Resultados e Discussão

Resultados:

Revisão de conceitos de Teoria da Computação e Compiladores. Foram revisados os seguintes conceitos: Expressões Regulares; Gramática Regular; Gramática livre de contexto; e DSL - Linguagem Específica de Domínio.

Estudo das ferramentas de software existentes sobre Síntese de Programas. As seguintes ferramentas foram identificadas: SKETCH, Rossete, Prose e Alloy.

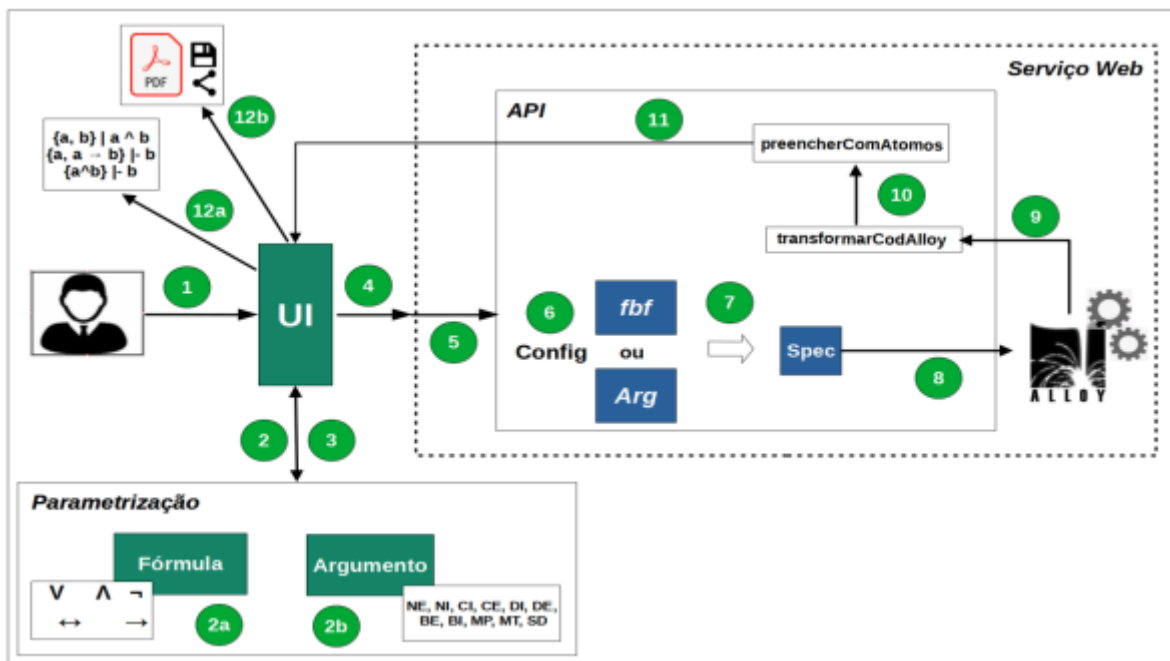


Figura 1: Arquitetura e funcionamento do sintetizador de elementos da Lógica Proposicional.

Aprimoramento/desenvolvimento do sintetizador de fórmulas e argumentos. Na Figura 1 em (1) o usuário acessa a interface do aplicativo móvel desenvolvido. Em (2) informa se quer gerar fórmulas ou argumentos. Depois de passar as informações necessárias para a síntese do elemento escolhido, o aplicativo junta-as em (3)

e acessa o Serviço Web (API) em (4). Com isso, a API recebe as informações do aplicativo em (5) e vai acessar o componente do elemento a ser sintetizado em (6) e (7). A especificação completa é passada para o interpretador da Alloy em (8). A Alloy gera os modelos e retorná-os para a API. A API precisa organizar os modelos (9) para conseguir retornar para o usuário, principalmente preenchendo com átomos em (10). A API retorna os elementos gerados para o aplicativo em (11). O aplicativo carrega uma pré-visualização dos elementos em (11a) e o usuário pode visualizar, compartilhar e salvar um PDF com todos elementos solicitados e gerados em (11b).

```

1  [
2  {
3      "id": 0,
4      "fbf": "(B ^ A) <-> ~(A ^ B)"
5  },
6  {
7      "id": 1,
8      "fbf": "~((B ^ A) -> (B ^ A)) ^ B"
9  },
10 {
11    "id": 2,
12    "fbf": "((~A ^ B) -> (~A ^ B)) ^ A"
13  },
14 {
15    "id": 3,
16    "fbf": "~((B ^ A) <-> (A ^ B))"
17  },
18 ]

```

Figura 2: Exemplos de fbfs sintetizadas com Alloy e transportada com a API.

As Figuras 2 e 3 representam um exemplo da API e Aplicativo móvel. Mais informações sobre os artefatos produzidos nesta abordagem podem ser visualizadas no seguinte link: <https://github.com/OliviFilipeOliveira/logic-elements>.

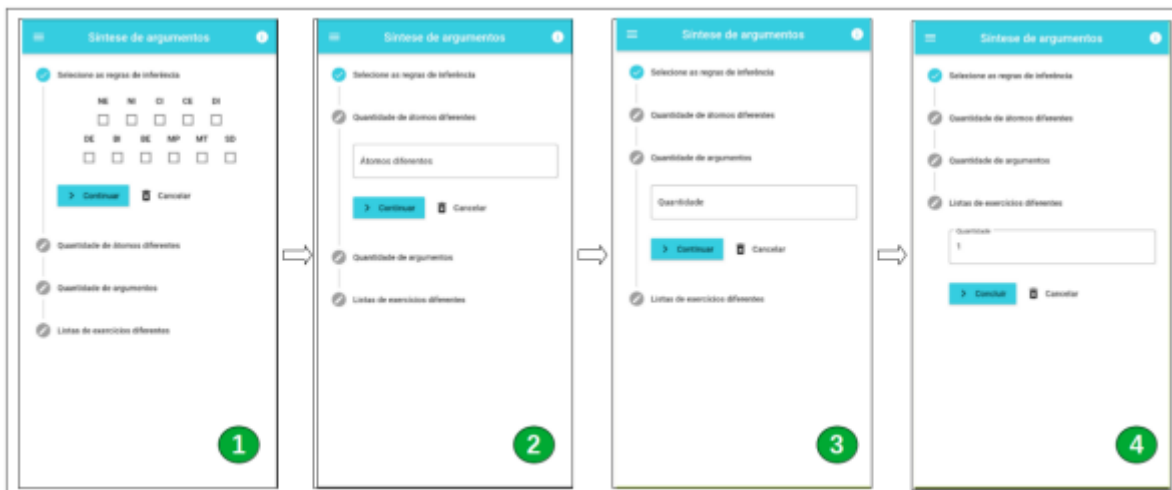


Figura 3: Exemplo de configuração para síntese e acesso de argumentos

Discussões:

No ciclo passado dedicamos muito tempo ao estudo dos conceitos, técnicas, problemas e soluções de problemas. Conseguimos gerar fórmulas com eficácia e argumentos de uma maneira não tão eficaz. No ciclo atual nós dedicamos um tempo também na ideia de utilizar a Alloy e isso nos fez conseguir ter eficácia tanto na síntese de fórmulas como na de argumentos. Além disso, conseguimos criar diversos recursos que somaram na nossa abordagem de geração dos elementos da Lógica Proposicional, como por exemplo, a criação da API e do aplicativo móvel.

Começar estudar Síntese de Programas com intuito de resolver problemas relativamente simples é muito importante. Dito isso, hoje conseguimos resolver o problema da geração tediosa de questões relacionadas à Lógica Proposicional e conseguimos visualizar diversos outros problemas maiores para resolver utilizando as técnicas de Síntese de Programas.

Conclusões

O projeto chega ao fim do ciclo gerando o que foi proposto, que são os elementos da lógica proposicional através da adaptação da técnica de Geração por Esboço da Síntese de Programas em conjunto com a linguagem de geração de modelos Alloy. Foram criados alguns componentes no desenvolvimento da abordagem, dentre eles as especificações Alloy, o aplicativo móvel e API.

A pesquisa atende uma necessidade real, que é a geração automática de questões de Lógica Proposicional. Porém, sua importância também está no fato de que esta necessidade está sendo atendida através da aplicação de uma área antiga da computação, mas relativamente desconhecida, que é a Síntese de Programas. Em um futuro próximo a Síntese de Programas estará resolvendo problemas que antes não se imaginava uma resolução. Para isso, é preciso iniciar a síntese de tarefas relativamente simples para que posteriormente possamos solucionar problemas maiores, e é o que esta pesquisa propôs.

A abordagem desenvolvida consegue gerar com eficiência as fórmulas e os argumentos válidos da Lógica Proposicional. A eficiência é garantida através da definição formal de fórmula juntamente com a definição das regras de inferência, todas implementadas nas especificações Alloy. Isso garante que as fórmulas geradas sejam estruturalmente válidas e os argumentos consigam a partir das premissas chegar na conclusão. Os componentes desenvolvidos são peças fundamentais para uma aplicação direta da abordagem no mundo real, permitindo através do aplicativo que professores e alunos da disciplina de Lógica tenham acesso a um gerador automático de exercícios.

Durante o desenvolvimento desta pesquisa, foi produzida uma quantidade considerável de documentação sobre a área. Pode-se citar mapas conceituais, propostas de arquiteturas para o sintetizador, assim como um dicionário contendo conceitos essenciais para o entendimento da abordagem, resumos de artigos, livros e alguns testes de técnicas de síntese

Para o responsável que trabalhou no plano de trabalho atual, a pesquisa teve grandes resultados positivos. De modo geral conseguiu-se concluir a proposta da pesquisa, que resume-se em estudar de forma simplificada a área de Síntese de Programas e desenvolver o sintetizador. A pesquisa contribuiu diretamente para o desenvolvimento do aluno, levando-o aprender conceitos, técnicas e abordagens de estudo dos mais diversos tipos. Somado a isso, é importante ressaltar também que a estrutura estável do ambiente junto a relação com o orientador, contribuíram fortemente para o responsável pudesse continuar aprendendo os conteúdos.

Referências bibliográficas

- [1]. GULWANI, S., POLOZOV, O., & SINGH, R. (2017). Program synthesis. Foundations and Trends® in Programming Languages, v. 4, n. 1-2, p. 1-119, 2017.
- [2]. Jackson, Daniel. Software Abstractions: logic, language, and analysis. MIT press, 2012.
- [3]. SOLAR-LEZAMA, Armando. The sketching approach to program synthesis. In: Asian Symposium on Programming Languages and Systems. Springer, Berlin, Heidelberg, 2009. p. 4-13.
- [4]. SOLAR-LEZAMA, Armando; BODIK, Rastislav. Program synthesis by sketching. University of California, Berkeley, 2008.