Modelando e verificando o cálculo formal de Planos de Corte com Lean 4

Projeto Orientadado em Computação I - Pesquisa Científica

Bernardo Borges
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
bernardoborges@dcc.ufmg.br

***Abstract*—Esta pesquisa envolve a definição e verificação da lógica pseudo-booleana de Planos de Corte utilizando o provador de teoremas Lean 4**

***Index Terms*—lean, cutting planes, formal methods, pseudo boolean reasoning**

I. Introdução

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris luctus lacus eu varius fermentum. Nullam lacus urna, semper a euismod sed, hendrerit eu nisi. Donec nec risus turpis. Vestibulum placerat justo sit amet aliquam mattis. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed vulputate vulputate maximus. Sed euismod turpis vitae libero consectetur gravida. Pellentesque eu nibh ipsum. Ut porta tortor vel pharetra euismod. Nulla quis nunc bibendum, condimentum sem a, tempus lectus.

II. Referencial Teórico

A. Satisfiabilidade

O problema da satisfabilidade booleana (SAT) é muito importante para a Ciência da Computação, sendo o primeiro demonstrado ser da class NP-Completo. Ele é o problema de decidir para dada expressão booleana, se é possível escolher valores para as variáveis de forma que a expressão como um todo seja verdadeira. Este problema é extensivamente pesquisado na área de métodos formais [1] e existem inclusive competições para definir qual é o melhor solucionador do mundo [2].

Um formato comumente utilizado para representar essas expressões é a Forma Normal Conjuntiva (CNF), que consiste da conjunção de cláusulas, em que cada cláusula é a disjunção de variáveis ou negação de variáveis.

$$C_1 \wedge C_2 \wedge \cdots \wedge C_n \tag{1}$$

onde,

$$C_i = a_1^i \vee a_2^i \vee \cdots \vee a_k^i \tag{2}$$

e para cada j

$$a_j^i \in \{x_1, x_2, \ldots, x_m, \neg x_1, \neg x_2, \ldots, \neg x_m\} \tag{3}$$

B. Pseudo-Booleans

Neste trabalho, introduzimos uma outra representação para expressões, chamados Pseudo-Booleanos, funções estudadas desde os anos 1960 na área de pesquisa operacional, em programação inteira. Este formato consiste de:

$$\sum_i a_i l_i \geq A \tag{4}$$

onde,

$$\begin{gathered} A, a_i \in \mathbb{N} \\ l_i \in \{x_i, \overline{x}_i\}, \qquad x_i + \overline{x}_i = 1 \end{gathered} \tag{5}$$

Este formato é exponencialmente mais compacto que o CNF, o que motiva seu uso [3].

C. Pseudo-Boolean Reasoning

A lógica formal de Cutting Planes introduz 2 axiomas e 4 regras de inferência, nomeadamente **Adição**, **Multiplicação**, **Divisão** e **Saturação**, que permitem derivar novas inequações a partir de outras [4].

D. Lean Theorem Prover

Lean é uma linguagem de programação e provador de teoremas criado por Leonardo de Moura em 2013 [5], com influência de ML, Coq e Haskell. A sua versão mais atual de 2021, Lean 4 [6], se tornou proeminente entre matemáticos, por permitir a **formalização** e **verificação** de teoremas, auxiliando o trabalho teórico. Em 2021, uma equipe de pesquisadores usou o Lean para verificar a correção de uma prova de Peter Scholze na área de matemática condensada [7], o que atraiu atenção por formalizar um resultado na vanguarda da pesquisa matemática. Em 2023, Terence Tao usou o Lean para formalizar uma prova da conjectura Polinomial de Freiman-Ruzsa [8], um resultado publicado por Tao e colaboradores no mesmo ano.

III. Contribuições

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris luctus lacus eu varius fermentum. Nullam lacus urna, semper a euismod sed, hendrerit eu nisi. Donec nec risus turpis. Vestibulum placerat justo sit amet aliquam mattis. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed vulputate vulputate maximus. Sed euismod turpis vitae libero consectetur gravida. Pellentesque eu nibh ipsum. Ut porta tortor vel pharetra euismod. Nulla quis nunc bibendum, condimentum sem a, tempus lectus.

Conclusões

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris luctus lacus eu varius fermentum. Nullam lacus urna, semper a euismod sed, hendrerit eu nisi. Donec nec risus turpis. Vestibulum placerat justo sit amet aliquam mattis. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed vulputate vulputate maximus. Sed euismod turpis vitae libero consectetur gravida. Pellentesque eu nibh ipsum. Ut porta tortor vel pharetra euismod. Nulla quis nunc bibendum, condimentum sem a, tempus lectus.

Referências Bibliográficas

[1] D. Le Berre, “Sat Live! keep up to date with research on the satisfiability problem”, Acesso em http://www.satlive.org/.

[2] M. Heule, M. Jävisalo, M. Suda, “The International SAT Competition Web Page”, Acesso em https://satcompetition.github.io/.

[3] J. Nordström, “Pseudo-Boolean Solving and Optimization”, Fevereiro de 2021.

[4] J. Nordström, “A Unified Proof System for Discrete Combinatorial Problems”, Novembro de 2023.

[5] L. de Moura, S. Kong, J. Avigad, F. van Doorn, J. von Raumer “The Lean Theorem Prover”, 25th International Conference on Automated Deduction (CADE-25), Berlin, Germany, 2015. Acesso em https://lean-lang.org/papers/system.pdf.

[6] L. de Moura, S. Ullrich “The Lean 4 Theorem Prover and Programming Language”, 28th International Conference on Automated Deduction (CADE-28), Pittsburgh, USA, 2021. Acesso em https://lean-lang.org/papers/lean4.pdf.

[7] J. Commelin, P. Scholze “Liquid Tensor Experiment”. Acesso em https://math.commelin.net/files/LTE.pdf

[8] T. Tao, “The Polynomial Freiman-Ruzsa Conjecture”, Novembro de 2023. Acesso em https://teorth.github.io/pfr/.