

## VERIFICAÇÃO DE CERTIFICADOS DE MODELOS MILP

André Luiz Feijó dos Santos<sup>1</sup> Luiz Carlos de Abreu Albuquerque<sup>2</sup> André Gustavos dos Santos<sup>3</sup>  
Universidade Federal de Viçosa - Departamento de Informática. {andre.santos1, <sup>2</sup>lcaa, <sup>3</sup>andre.santos}@ufv.br

ODS9: Dimensões Econômicas

Categoria: Pesquisa

### Introdução

Em alguns cenários, o uso de Programação Inteira **não** é tolerável a erros (tanto de código quanto de precisão), como em *Automated Theorem Provers* e Produtos de software financeiros. Para esses contextos, o *solver* SCIP resolve problemas considerando aritmética racional, além de emitir **certificados** para verificar resultados obtidos.

Estes certificados seguem o formato VIPR, cuja estrutura é de uma prova formal para os limites do modelo. O exemplo a seguir prova que 1 é um *lower bound* para o modelo abaixo. Para garantir que o resultado retornado é correto, é necessário verificar esta prova.

<b>Derivado Justificativa</b>	$\begin{aligned} &\min x_2 \\ &\text{s.t.} \\ &C_1) 2x_1 + x_2 \geq 1 \\ &C_2) 2x_1 - 3x_2 \leq 1 \\ &x_1, x_2 \in \mathbb{Z} \end{aligned}$
-------------------------------	--

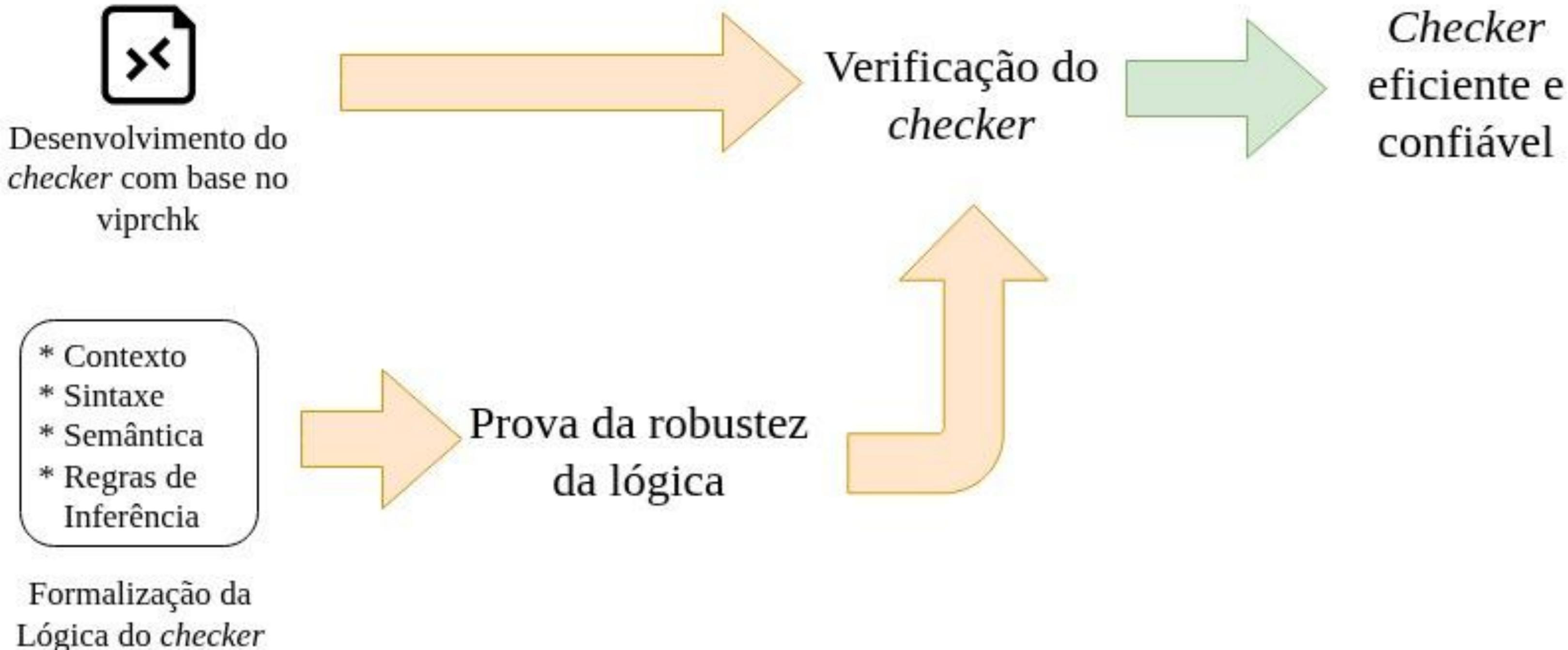
$a_1) x_1 \leq 0$  (asm) Hipótese  
 $a_2) x_1 \geq 1$  (asm) Hipótese  
 $d_3) x_2 \geq 1$  (lin)  $C_1 - 2 \times a_1$   
 $d_4) x_2 \geq \frac{1}{3}$  (lin)  $-\frac{1}{3} \times C_2 + \frac{2}{3} \times a_2$   
 $r_5) x_2 \geq 1$  (rnd) round-up em  $d_4$  ( $x_2 \in \mathbb{Z}$ )  
 $d_6) x_2 \geq 1$  (uns) Unsplit em  $d_3$  e  $r_5$  ( $a_1 \vee a_2 = \top$ )

### Objetivos

Nossos objetivos com esta pesquisa são:

- Formalizar a Lógica empregada pelo VIPR e provar sua robustez;
- Propor um *checker* para certificados VIPR eficiente e seguro;
- Apresentar uma revisão detalhada da literatura sobre certificação de modelos MILP.

### Metodologia



Para a formalização da lógica, foram definidos a **sintaxe**, a **semântica**, o **contexto** sobre o qual as provas são realizadas (i.e., o modelo MILP) e poucas **regras de inferência**, tais como a regra LIN a seguir, onde  $\Sigma$  é um conjunto de restrições de hipótese e cada  $C_i, 1 \leq i \leq k+1$ , uma restrição derivada.

$$\frac{\Sigma \vdash C_1 \quad \Sigma \vdash C_2 \quad \dots \quad \Sigma \vdash C_k}{\Sigma \vdash C_{k+1}} \text{ (LIN)}$$

$$\text{if } \exists \alpha \in \mathbb{R}^k, \text{slc}(\alpha, C) \wedge \text{dom}\left(\sum_{i=1}^k \alpha_i C_i, C_{k+1}\right)$$

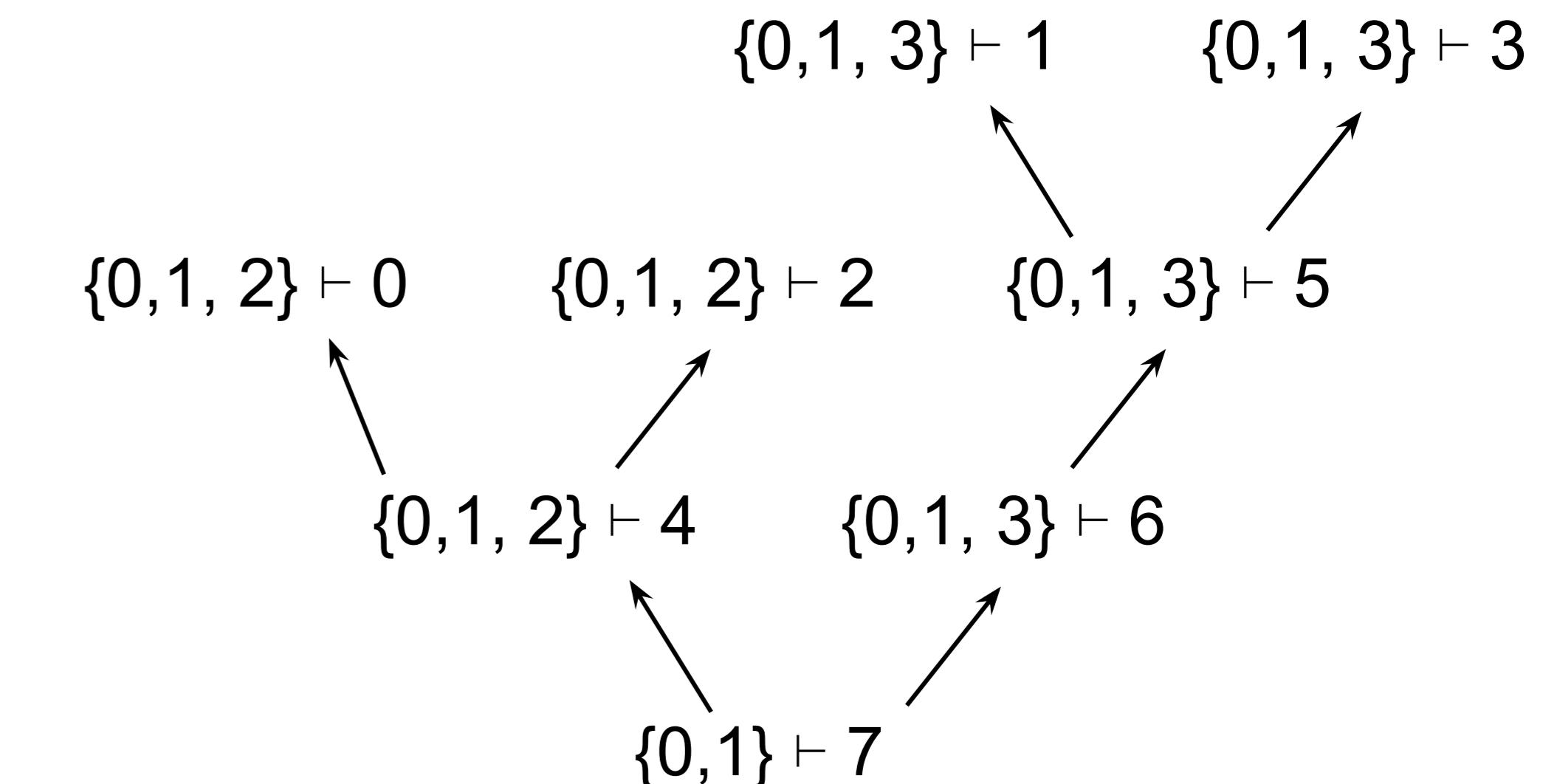
A verificação da robustez da lógica foi realizada utilizando o assistente de provas Lean 4 e o *checker* está sendo desenvolvido em C++.

### Resultados

A formalização da lógica VIPR levou a três principais resultados:

1. A lógica é robusta;
2. Esta formalização contorna especificações permissivas da documentação oficial do VIPR, como possibilidade de *splitting* em restrições que não formam disjunções;
3. Uma das etapas da verificação de um certificado é garantir que a última derivação não assume hipóteses (além do MILP). Com a modelagem, foi possível extraír um algoritmo de busca em grafos de prova que pode levar à uma verificação mais eficiente das hipóteses, se comparado a checkers já existentes, que usam operações de conjuntos.

Ao lado, há um exemplo de prova, em que cada número é o índice de uma restrição do exemplo de prova da Introdução.



### Conclusões

O resultado sobre a robustez da lógica VIPR garante a verificação correta do certificado, ao ser implementado um *checker* fiel à modelagem apresentada, aumentando a segurança acerca dos resultados retornados pelo *solver*. Ainda, a proposta do algoritmo de verificação de hipóteses pode otimizar consideravelmente esta etapa. A pesquisa continua com a implementação total do *checker* e experimentos sobre sua eficiência.

### Principais Referências

- [1] HEUNG, K. K. H.; GLEIXNER, A.; STEFFY, D. E. Verifying integer programming. Springer International Publishing, 2017.
- [2] WOOD, K. et al. Satisfiability Modulo Theories for Verifying MILP Certificates. arXiv, 2024.