MÉTODOS DE DECISÃO PARA A ARITMÉTICA DE PRESBURGER NA DEDUÇÃO AUTOMÁTICA COM TÉCNICAS DE REESCRITA

Luiz M. R. Gadelha Jr.[1], **Mauricio Ayala Rincón**[2]

Departamento de Matemática, Universidade de Brasília

Introdução A aritmética de Presburger ($\mathcal{AP}$) consiste dos números inteiros munidos da adição e de um predicado de ordem. A teoria da $\mathcal{AP}$ é de grande relevância na história da dedução automática por ser a primeira teoria provada efetivamente decidível, usando o método de eliminação de quantificadores. Apesar de que este método seja completo, mostrou-se inviável devido à elevada complexidade ($O(2^{2^{2^n}})$ no tamanho da fórmula). Um algoritmo de semi-decisão, desenvolvido por Shostak, para fórmulas da teoria da $\mathcal{AP}$ livres de quantificadores, baseado na transformação lógica das fórmulas em problemas de programação linear inteira e resolução destes pelo método *SUP-INF* de Bledsoe, resulta relevante na prática no contexto da dedução automática por apresentar complexidade consideravelmente inferior. Em [GA96] apresentou-se uma implementação[3] deste algoritmo aprimorado com o objetivo de aumentar a classe de fórmulas efetivamente decidíveis. Apresentam-se considerações sobre a utilidade deste algoritmo de decisão quando combinado às técnicas dedutivas dos sistemas de reescrita de termos, de uso intensivo na manipulação de teorias condicionais que incluem parâmetros aritméticos [Aya97]. A $\mathcal{AP}$ é um exemplo excelente de parâmetro em especificações condicionais implementadas por técnicas de reescrita, devido a que a aritmética é a base da maioria dos sistemas dedutivos e a que esta não pode ser especificada de maneira simples por sistemas de reescrita condicionais (SRC) ou não-condicionais (SR) *convergentes* [Vor88].

Resultados Ilustra-se a aplicação do algoritmo de decisão para a $\mathcal{AP}$ no processo de *completação* de Knuth-Bendix de especificações equacionais condicionais e na dedução de teoremas indutivos. O seguinte SRC apresenta o predicado *divide* sobre os naturais, onde S especifica a função sucessor, $+$ uma função associativa-comutativa e $<$ um predicado de ordem com a semântica usual. $\{1.\ x + 0 \to x;\ 2.\ x + S(y) \to S(x + y);\ 3.\ 0 < S(x) \to true;\ 4.\ x < 0 \to false;\ 5.\ S(x) < S(y) \to x < y;\ 6.\ divide(u, 0) \to true;\ 7.\ divide(u, v) \to false\ if\ ((v < u)\ and\ (v \neq 0));\ 8.\ divide(u, u + v) \to divide(u, v);\}$. O processo de completação (que pode ser realizado com o *RRL* [KZ89]) gera *pares críticos* resultantes da sobreposição das regras 8 com 7, 8 com 1 e 2 com 8: $divide(u, v) = false\ if\ u + v \neq 0\ and\ u + v < u$, $divide(u, u) = true$, e $divide(u, S((u + y))) = divide(u, S(y))$. A completação termina gerando um SRC "equivalente" ao original incluindo os três pares críticos anteriores orientados da esquerda para a direita. No entanto, se se trabalha com um parâmetro pré-construído para a $\mathcal{AP}$, restringida aos inteiros não-negativos, pode-se eliminar o primeiro par crítico, já que sua premissa é inconsistente, assim como o segundo, já que este subsume semanticamente (com respeito à $\mathcal{AP}$) nas regras sexta e oitava; de fato, observe que $divide(u, u) = divide(u, u + 0)$ módulo $\mathcal{AP}$. É importante mencionar que o algoritmo de decisão para $\mathcal{AP}$ trata fórmulas aritméticas com símbolos de função (e predicado) não-interpretados o que permite unificar expressões módulo o parâmetro aritmético. O desenvolvimento de procedimentos para gerar esquemas indutivos é relevante na dedução automática (via reescrita), já que muitos teoremas indutivos não podem ser demonstrados pelo esquema indutivo usual. Dados um SR(C) *noether-*

[1]gadelha@mat.unb.br. Bolsista IC PIBIC, CNPq-UnB
[2]ayala@mat.unb.br. Financiamento parcial FAP-DF
[3]Disponível em http://www.mat.unb.br/~ gadelha

iano, especificando uma função f, e uma conjectura acerca de f, o *método de conjuntos de cobertura* (MCC) gera um esquema indutivo baseado na estrutura da especificação de f. Considere o seguinte SR definindo o "maior divisor comum": $\{1.\ mdc(x,0) \to x;\ 2.\ mdc(0,x) \to x;\ 3.\ mdc(x,x+y) \to mdc(x,y);\ 4.\ mdc(x+y,x) \to mdc(y,x)\}$. Observe que o esquema indutivo usual ($mdc(0,y) = mdc(y,0)$ e $mdc(x,y) = mdc(y,x) \Rightarrow mdc(x+1,y) = mdc(y,x+1)$) não pode ser usado para demonstrar a comutatividade de mdc, já que a conclusão do passo indutivo não pode ser simplificada com as regras do SR. O MCC usa a estrutura da especificação do mdc no próprio SR para gerar o esquema: $mdc(0,y) = mdc(y,0)$ e $mdc(x,y) = mdc(y,x) \Rightarrow mdc(x,x+y) = mdc(x+y,x)$ que permite provar a comutatividade de mdc. Observe que $mdc(x,x+y) = mdc(x+y,x)$ reduz à hipótese de indução aplicando as regras 3 e 4. A geração de esquemas indutivos apropriados quando se trabalha com SRCs com parâmetros aritméticos depende muitas vezes do uso de unificação semântica modulo o parâmetro aritmético [KS96]. O seguinte exemplo ilustra a aplicação do algoritmo de decisão da $\mathcal{AP}$ para geração de esquemas indutivos apropriados. Considere o SRC conformado pelas regras 6, 7 e 8 do SRC especificando o predicado "*divide*" sobre o parâmetro dos naturais. Para demonstrar a conjectura $divide(2,x) = not(divide(2,S(x)))$ obtem-se o seguinte esquema indutivo: $divide(2,0) = not(divide(2,S(0)))$, $divide(2,v) = not(divide(2,S(v)))\ \ if\ \ v < 2 \wedge v \neq 0$, $divide(2,v) = not(divide(2,S(v))) \Rightarrow divide(2,2+v) = not(divide(2,S(2+v)))$. O primeiro caso da base da indução pode ser provado pelas regras 6 e 7. Observe que a regra 7 aplica devido a que sua condição, $S(0) < 2 \wedge S(0) \neq 0$, vale. O segundo caso reduz-se a $false = not(divide(2,S(v)))\ \ if\ \ v < 2 \wedge v \neq 0$ aplicando a regra 7. O lado direito da igualdade não pode ser simplificado, mas aplicando o algoritmo de decisão para $\mathcal{AP}$ pode-se solucionar a premissa obtendo $false = not(divide(2,S(1)))$ que se reduz a $false = not(true)$ aplicando as regras 8 e 6. A conclusão do passo indutivo reduz para $divide(2,v) = not(divide(2,S(v+2)))$ aplicando a regra 8 que "casa" módulo $\mathcal{AP}$ com o lado esquerdo da igualdade. Para reduzir $divide(2,S(v+2))$ também é necessário casamento módulo $\mathcal{AP}$ para detectar que $S(v+2) = 2+S(v)$. Desta forma obtem-se a hipótese de indução completando a prova.

Conclusão A completação condicional é aprimorada eliminando pares críticos inconsistentes e triviais módulo a $\mathcal{AP}$ e a classe de teoremas indutivos dedutíveis por meio de técnicas de reescrita é aumentada solucionando premissas aritméticas e resolvendo casamento e unificação módulo a $\mathcal{AP}$. Uma das aplicações potenciais dos algoritmos de decisão para $\mathcal{AP}$ nos sistemas dedutivos baseados em técnicas de reescrita é a geração de lemas intermediários necessários para concluir provas de conjecturas (indutivas).

Referências

[Aya97] M. Ayala. A Decision Procedure for Conditional Rewriting Systems with Built-in Predicates. In *Anais XXIV Seminário Integrado de Software e Hardware, Brasília, Brazil*, August 1997.

[GA96] L. M. R. Gadelha and M. Ayala. Aplicação de Métodos de Programação Linear Inteira para Decisão na Teoria da Aritmética de Presburger. In *Anáis do XIX Congresso Nacional de Matemática Aplicada e Computacional, Goiânia, Brazil, September, 1996*, 1996. In Portuguese.

[KS96] D. Kapur and M. Subramaniam. New Uses of Linear Arithmetic in Automated Theorem Proving by Induction. *Journal of Automated Reasoning*, 16(1/2), 1996.

[KZ89] D. Kapur and H. Zhang. An overview of Rewrite Rule Laboratory (RRL). In N. Dershowitz, editor, *Proc. Third Int. Conf. on Rewriting techniques and Applications, Chapel-Hill, NC*, volume 355 of *LNCS*. Springer, April 1989.

[Vor88] S. G. Vorobyov. On the Arithmetic Inexpressiveness of Term Rewriting Systems. In *Third Symp. on Logics in Computer Sciences, Edinburgh Scotland*, pages 212–217, July 1988.