

Detecção de Paralelismo em Equações Discretizadas por Aplicação do Método das Diferenças Finitas em Equações Diferenciais Parciais

Fernando A. Pinto¹, Eliana S. de Almeida¹, Leonardo P. Viana¹

¹Laboratório de Computação Científica e Análise Numérica
Centro de Pesquisa em Matemática Computacional – Instituto de Computação
Universidade Federal de Alagoas (UFAL)
Campus A. C. Simões s/n – Tabuleiro dos Martins – 57072.970 – Maceió – AL – Brasil
fernando@maceio.al.gov.br, eliana.almeida@pesquisador.cnpq.br,
lpviana@lccv.ufal.br

Resumo. *A proposta deste trabalho é apresentar a especificação semântica formal de uma linguagem de programação, um arcabouço de alto nível, a partir da detecção automática de condições de execução em paralelo do Método das Diferenças Finitas em equações diferenciais. Esta funcionalidade eximirá o usuário da necessidade de conhecimento prévio de características dos paradigmas de programação paralela. Esta abordagem é fundamentada por técnicas formais da computação como Semântica Denotacional e CSP (Communicating Sequential Processes) como meta-linguagem para suporte às oportunidades de concorrência.*

1. Introdução

O processo da geração automática de programas paralelos, sejam eles científicos ou de propósito geral, tem se tornado um grande desafio na computação. O uso de formalismos sintáticos e semânticos possibilitam a criação de *parsers* entre os paradigmas sequenciais e paralelos facilitando este processo de implementação. Neste trabalho, a partir de uma equação diferencial discretizada e com base em uma gramática descritiva, pretende-se gerar códigos paralelos para processamento de alto desempenho.

2. Equações Diferenciais Parciais

Muitos fenômenos observados na natureza podem ser modelados por Equações Diferenciais Parciais (EDP's) e são amplamente estudados por cientistas e engenheiros. EDP's são equações contendo derivadas de uma função com duas ou mais variáveis. Trataremos de equações diferenciais lineares de segunda ordem com duas variáveis, que podem ser escritas na forma

$$au_{xx} + bu_{xy} + cu_{yy} + du_x + eu_y + fu + g = 0 ,$$

onde u é função das variáveis independentes x e y , a , b , c , d , e , f e g são constantes e os índices subscritos denotam derivadas parciais nas respectivas variáveis.

De modo geral, soluções analíticas para estas equações diferenciais não são possíveis. Assim se faz necessária a aproximação destas soluções através de métodos numéricos computacionais, algoritmos que são implementados em uma linguagem de programação específica. Desta forma, o uso destes métodos numéricos muitas vezes requer conhecimento de técnicas de programação de alto desempenho. A criação de mecanismos que simplifiquem o processo de implementação dos algoritmos para resolução numérica de EDP's, preferencialmente em linguagem alvo que explore o paralelismo usado em clusters de computadores, é desejado e pode produzir ferramentas que aumentem a produtividade

3. Semântica Denotacional

A semântica denotacional é o método mais rigoroso para descrever significado de programas. Para especificar uma linguagem, baseada nesta abordagem é necessária a definição de sua sintaxe abstrata, seus domínios semânticos e funções semânticas. Tanto o domínio semântico quanto as funções semânticas podem ser definidos em termos de CSP (Communicating Sequential Processes) [Hoare 2004].

Um trabalho que demonstra com sucesso a especificação do comportamento de programa com uso do formalismo semântico é mostrado em [Almeida and Haeusler 1994], onde os autores utilizam a Semântica Denotacional associada a meta-linguagem CSP para descrever o comportamento das trocas de mensagens em programa no paradigma orientado a objetos.

4. Metodologia e Resultados Preliminares

Nossa proposta é a definição semântica de uma linguagem capaz de abstrair as implementações de uma equação discretizada, gerando uma semântica para a definição de processos concorrentes através do trabalho desenvolvido em [Martins and Haeusler 2008].

O método das diferenças finitas para resolução numérica de EDP's através de sua discretização pode ser escrito como um modelo de grafos de dependência, como exemplificado na Fig. 1. Estes grafos podem ser mapeados em CSP, de forma a explorar suas possibilidades de processamento paralelo concorrente,

$$\begin{aligned} R_r &= (C_r!PU_{r,s+1} \longrightarrow A) ; S_r = (C_r?PU_{r,s+1} \longrightarrow B) \\ R_l &= (C_l!PU_{r,s-1} \longrightarrow A) ; S_l = (C_l?PU_{r,s-1} \longrightarrow B) \\ R_c &= (C_c!PU_{r,s} \longrightarrow A) ; S_c = (C_c?PU_{r,s} \longrightarrow B) \end{aligned}$$

A computação concorrente destes processos gera um novo estado P , como visto:

$$(R_r || S_r || R_l || S_l || R_c || S_c) \longrightarrow P_{r+1,s}$$

Esta modelagem em CSP apresenta a computação necessária para definir um novo processo P na célula $r + 1, s$ vista na figura 1(a). A figura 1(b) apresenta toda a dependência necessária para a resolução de um problema.

Para a completude das especificações, estamos estudando a semântica de vários esquemas de diferenças finitas usados em métodos numéricos para resolução de equações diferenciais parciais. A construção desta semântica na forma *top-down* e incremental é considerada positiva [Tirelo et al. 2008]. Esta abordagem será utilizada como regra arquitetural na especificação semântica dos métodos numéricos.

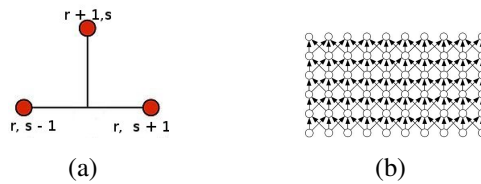


Figura 1. Comunicação entre as células no método das diferenças finitas. (a) Stencil para comunicação em *forward*. (b) Grafo de dependência com o padrão de comunicação entre as células.

5. Conclusões e Perspectivas

Neste trabalho, pretendemos criar uma especificação semântica para a resolução numérica de EDP's através do método das diferenças finitas. Esta especificação será o modelo para implementação de uma ferramenta robusta, compilador, capaz de gerar código paralelo a partir de uma EDP discretizada.

Referências

- Almeida, E. and Haeusler, E. H. (1994). Utilização de csp como ferramenta para especificação semântica de linguagens orientadas a objetos. *Seminário Integrado de Software e Hardware*, (XIV).
- Hoare, C. A. R. (2004). *Communicating Sequential Processes*. Prentice Hall.
- Martins, C. B. and Haeusler, E. H. (2008). Automatic parallel code generation through denotational semantics and dependence graphs. In *II Workshop on Languages and Tools for Parallel and Distributed Programming, LTPD 2008*. Brazilian Computer Society.
- Tirelo, F., Bigonha, R. S., and Saraiva, J. (2008). Disentangling denotational semantics definitions. *J-Jucs*, 14(21):3592–3607.