

Proposta de trabalhos

1. The Curry-Howard correspondence [9]
A descoberta que programas e provas tem uma relação íntima.
Objetivos: (i) Exibir as principais idéias da correspondência entre programas e provas.
2. The Daikon Invariant Detector [3]
Daikon é um sistema capaz de detectar algumas invariantes usando a partir de um rastreamento da execução de um programa.
Objetivos: (i) Exibir as características e a capacidade da ferramenta (ii) Mostrar exemplos da aplicação (iii) Exibir a técnicas usadas e o limite delas para detectar invariantes.
3. Avaliação parcial [2].
Algumas entradas de um programa estão conhecidas no tempo de compilação ou no começo da execução, sem o compilador o ambiente de execução aproveitar desse conhecimento (por exemplo se $\text{pow}(x,2)$ ou $\text{pow}(x,3)$ sempre chama a mesma função). A avaliação parcial visa melhorar o desempenho de programas gerando uma versão parcialmente avaliada.
Objetivos: (i) Exibir a motivação e os principais técnicas de avaliação parcial. (ii) Mostrar exemplos de ferramentas e aplicações.
4. Semântica algébrica [8, cap. 12].
Objetivos: (i) Exibir a motivação e os principais técnicas dessa semântica. (ii) Comparar a abordagem com outras semânticas.
5. Semântica de ações [8, cap. 13].
Objetivos: (i) Exibir a motivação e os principais técnicas dessa semântica. (ii) Comparar a abordagem com outras semânticas.
6. Semântica de sistemas distribuídos: CCS [10, cap. 14].
CCS permite especificar sistemas distribuídas e concorrentes (processos comunicantes) e, usando uma semântica operacional, analisar-lhes.
Objetivos: (i) Exibir a motivação e os principais técnicas de CCS. (ii) Mostrar exemplos da aplicação de CCS.
7. Semântica axiomática de arrays [7, cap. 4].
Uma extensão da semântica axiomática que permite provar características de programas com campos.
Objetivos: (i) Exibir a sintaxe e a semântica axiomática de IMP com campos. (ii) Mostrar exemplos da aplicação dessa extensão.
8. Semântica do cálculo π [6].
O cálculo π permite de especificar, modelar e analisar sistemas de processos comunicantes com mobilidade: a relação de comunicação entre os processos pode mudar com o tempo.
Objetivos: (i) Exibir a motivação e os principais técnicas da cálculo π . (ii) Mostrar exemplos da aplicação do cálculo π .

9. Programação estruturada com Goto [5].
A contribuição de Knuth à discussão do perigo de “goto”.
Objetivos: (i) Exibir as motivações e principais argumentos na discussão do “goto” no anos 70. (ii) Comparar exemplos de algoritmos com e sem “goto”.
10. Semântica e sistema de tipos de uma linguagem orientada a objetos para busca em banco de dados [1]
O artigo introduz a linguagem OQL (Object query language) para busca em bancos de dados. Objetivos: (i) Exibir a sintaxe, semântica e o sistema de tipos. (ii) Discutir a utilidade dessa abordagem.
11. Semântica denotacional de XPath.
XPath é uma linguagem de busca em documentos XML.
Objetivos: (i) Introduzir a linguagem XPath com exemplos. (ii) Exibir e explicar a semântica denotacional dessa linguagem.

Referências

- [1] G. Bierman. Formal semantics and analysis of object queries. In *Proceedings of ACM SIGMOD*, pages 407–418, 2003.
- [2] C. Consel and O. Danvy. Tutorial notes on partial evaluation. In *POPL '93: Proceedings of the 20th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 493–501, New York, NY, USA, 1993. ACM Press.
- [3] M. D. Ernst, J. H. Perkins, P. J. Guo, S. McCamant, C. Pacheco, M. S. Tschanz, and C. Xiao. The daikon system for dynamic detection of likely invariants. *Science of computer programming*, 2006.
- [4] J. Heering. The role of programming language semantics. Internet, 1999. <http://homepages.cwi.nl/~jan/semantics/node1.html>.
- [5] D. E. Knuth. Structured programming with go to statements. *ACM computing surveys*, 6(4), 1974.
- [6] Milner. *Communicating and mobile processes: The π* . Cambridge University Press, 1999.
- [7] J. C. Reynolds. *Theories of Programming Languages*. Cambridge University Press, 1998.
- [8] K. Slonneger and B. Kurtz. *Formal syntax and semantics of programming languages: a laboratory based approach*. Addison-Wesley, 1995. INF: 681.32.06 S634f.
- [9] P. Wadler. Proofs are programs: 19th century logic and 21st century computing. 2000.
- [10] G. Winskel. *The formal semantics of programming languages*. MIT Press, Feb. 1993. Biblioteca: 681.32.06 W777f.