

ELEMENTOS LÓGICOS DA PROGRAMAÇÃO II

(7004N4)

JOSÉ CARLOS BACELAR ALMEIDA

PROGRAMA DETALHADO

1. λ -calculus: teoria básica

- (1) Definição indutiva dos termos; variáveis livres e ligadas; α -equivalência.
- (2) Árvores sintáctivas dos termos e representação dos termos por grafos.
- (3) Convenção das variáveis de Barendregt; meta-operação da substituição de variáveis por termos; problema da captura de variáveis.
- (4) Axiomatização da teoria- λ de convertibilidade dos termos; Axioma β .
- (5) Teorema do ponto Fixo;
- (6) Técnicas de demonstração por indução no comprimento de provas (da teoria- λ).
- (7) Extensionalidade: Teoria- η .

2. Redução de termos no λ -calculus.

- (1) As relações de redução \rightarrow_β num passo e \rightarrow_β^* em zero ou mais passos. Formas β -normais.
- (2) Noções de confluência local e Church-Rosser. Classificação de \rightarrow_β à luz dessas definições. Teorema de Church-Rosser. Unicidade das formas β -normais.
- (3) Noções de termo normalizável, fortemente normalizável e de estratégia normal.
- (4) Formas normais à cabeça; noção de variável e de redex à cabeça. Redução à cabeça; forma normal principal à cabeça. Formas normais fracas à cabeça como o resultado do cálculo em programação funcional.

3. Estratégias, Avaliação, e Implementação do λ -calculus.

- (1) Ordem normal (“call-by-name”) e aplicativa (“eager”, “call-by-value”); discussão comparativa.
- (2) Noção de avaliação de termos fechados. Especificação da avaliação por uma estratégia de redução e uma noção de formas canónicas.

- (3) Especificação do processo de avaliação por regras de inferência: semântica natural “big-step” para avaliação “call-by-name” e “call-by-value”. Algoritmos de avaliação sugeridos pela semântica.
- (4) Conceito de “closure” na compilação de linguagens funcionais.
- (5) Semântica natural “big-step” para a avaliação “call-by-name” com closures.
- (6) Máquina abstracta de Krivine.

4. Tipos.

- (1) Motivação para o estudo do λ -calculus com tipos simples.
- (2) Formulação de Church. Definição dos tipos e dos termos; unicidade dos tipos; Teorema da normalização forte.
- (3) Formulação de Curry. Definição dos termos e dos tipos; relação de habitabilidade; juízos e sistema de inferência de tipos; tipos principais; propriedades do sistema.

5. Uma Linguagem de Programação Baseada no λ -calculus.

- (1) Booleanos e Inteiros: sintaxe abstracta; regras de inferência de tipos; regras da semântica de avaliação “big-step”.
- (2) Tuplos e Alternativas: sintaxe abstracta; regras de inferência de tipos; regras da semântica de avaliação “big-step”.
- (3) Definições `let...in...` e padrões.
- (4) Recursividade: construção `letrec`.
- (5) Listas: definição directa e como “açucar sintáctico” com tuplos e alternativas.

6. Outros Tópicos.

- (1) O poder expressivo do λ -calculus sem tipos: codificação de tipos inductivos e de recursividade genérica.
- (2) O sistema- T : recursividade primitiva e normalização forte.