

# Semântica Denotacional Legível

Roberto da Silva Bigonha  
Departamento de Ciência da Computação  
Universidade Federal de Minas Gerais

O baixo grau de legibilidade e a grande dificuldade para se entender, formular ou validar descrições denotacionais de semântica de linguagens de programação devem-se principalmente à maneira em que são apresentadas. Tradicionalmente, definições denotacionais consistem em quatro partes: sintaxe abstrata, domínios semânticos, equações semânticas e equações auxiliares. Estas partes são usualmente organizadas como uma coleção de definições de domínios sintáticos e semânticos, seguida por um grupo de equações definindo funções semânticas e auxiliares. Para definições de pequeno porte, esta organização funciona bem, mas para definições de linguagens de porte realista, o método é inadequado. O principal problema para entender, e também para formular, uma definição semântica formal de grande porte, estruturada monoliticamente, está relacionado com o fato de os métodos formais serem inerentemente complicados e de o leitor dever ter controle sobre um volume excessivo de informação para entender qualquer fragmento de definição.

Outro problema relacionado com o estilo usual de definições denotacionais é a dificuldade de manutenção e depuração. Como domínios podem ser utilizados em qualquer equação, uma alteração em sua especificação pode acarretar mudanças em toda a definição. Note-se também que definições formais de linguagens de interesse prático apresentam, além da *ilegibilidade*, baixo nível de *confiabilidade* e de *correção*, devido, principalmente, à complexidade dos métodos formais, que enseja incidência de erros de especificação.

As pesquisas que estamos desenvolvendo buscam soluções baseadas em técnicas de modularização e que usem o computador para automatizar várias das tarefas envolvidas no processo de formulação de definições formais denotacionais. Em particular, investigamos as seguintes soluções:

- formulação de semântica com auxílio de computador;
- mecanização do processo de validação de definições;
- uso de mecanismos de abstração para implementar recursos de modularização em linguagens funcionais;
- produção de definições formais legíveis para não-especialistas em semântica;
- uso do computador para o projeto de linguagens de programação, através da automatização da produção do compilador a partir da definição formal da linguagem.

O resultado alcançado é o projeto, especificação formal e implementação de um sistema integrado **xlds** para formulação, teste e documentação de definições denotacionais, baseado em uma linguagem de especificação de semântica modular, a linguagem **LDS**, que privilegia legibilidade, suporta formulação de definições denotacionais, garante a verificação em tempo de compilação da consistência de tipos polimórficos e permite a execução de definições de semântica através da geração automática de compiladores.

O sistema **xlds** adota, no contexto de definições semânticas, o estilo da metodologia de *Literate Programming* desenvolvida por D. Knuth para a área de construção e documentação de programas.

Nossa solução procura resolver o aparente conflito inerente a definições denotacionais, que, por um lado, devem ser formuladas para serem lidas por pessoas, muitas vezes não-especialistas na área, e por isto, devem ser apresentadas de forma legível, provavelmente como um artigo técnico, contendo todas as explicações, comentários e recursos tipográficos que facilitem sua leitura e entendimento. Por outro lado, estas mesmas definições formais devem ser precisas, concisas e processáveis mecanicamente, o que normalmente dificulta a leitura.

A linguagem **LDS**, que projetamos, permite combinar textos e equações semânticas de uma forma logicamente integrada e modular, exercendo no âmbito de formulação de definição semântica o mesmo papel do **web** de Donald Knuth. A partir de definições **LDS** é possível gerar arquivos no formato  $\text{\LaTeX}$  contendo a definição e arquivos com as equações semânticas numa linguagem executável, denominada **SDL** e baseada em  $\lambda$ -cálculo. **SDL** também é uma linguagem modular, polimórfica, puramente funcional, com tipos fortes e dotadas de recursos para expressar o mapeamento de programas na forma de sintaxe abstrata a domínios semânticos apropriados.

O resultado final do trabalho é um ambiente integrado para formulação de semântica denotacional, baseado na linguagem **LDS**, e que possui, entre outras, as seguintes características:

- editor de definição denotacionais integrado;
- interface com formatador  $\text{\LaTeX}$ ;
- interface homem-máquina de alto nível;
- suporte a formulação modular de definições semânticas;
- processamento de definições semânticas;
- consistência de tipos fortes e polimórficos;
- geração automática de compiladores a partir da definição em **LDS**;
- produção de documentação a partir de definição em **LDS**.

O sistema **xlds** encontra-se em fase de implementação em estações de trabalho gráfica e ambiente **UNIX**.