

# Pistache: Uma Implementação do $\pi$ -Calculus como Linguagem de Domínio Específico para Scala

## Parte Subjetiva

Pedro Matiello  
pmatIELLO@gmail.com

Professora Orientadora:  
Ana C. V. de Melo  
acvm@ime.usp.br

## 1 Desafios e frustrações

O trabalho se mostrou, de forma geral, mais difícil do que o imaginado no primeiro momento. Em particular, a implementação do operador de soma exigiu um esforço muito grande e, mesmo após muito trabalho e várias tentativas, não ficou totalmente do meu agrado, funcionando em uma estratégia de *spin lock*, que consome recursos de processamento que poderiam ser economizados em uma implementação mais eficiente.

A implementação da linguagem de domínio específico, apesar de também trabalhosa e de ter tomado uma quantidade de tempo significativa, ficou quase tão próxima do  $\pi$ -calculus quanto eu poderia desejar. Em alguns momentos, determinadas sutilezas do cálculo (e seu entendimento adequado para os propósitos do trabalho) criaram dificuldades, mas que puderam ser superadas com o auxílio da orientadora.

## 2 Disciplinas do curso mais relevantes para este trabalho

### Conceitos Fundamentais de Linguagens de Programação

Nesta disciplina tomei contato com o modelo funcional de programação e com a estratégia de avaliação preguiçosa de expressões (*lazy evaluation*). Os temas apresentados não somente tiveram utilidade prática no desenvolvimento deste trabalho, como também despertaram meu interesse para novas formas de estruturar programas.

Entendo, contudo, que temas de linguagens de programação — em especial, programação funcional, sistemas de tipos, e orientação a objetos — poderiam

ter um espaço maior na grade do curso. Estes me parecem mais relevantes para a Ciência da Computação do que Física ou Cálculo com múltiplas variáveis.

### **Estruturas de Dados e Análise de Algoritmos**

Nestas disciplinas aprendi a estruturar informações e a tratá-las de forma eficiente. A grande quantidade de algoritmos e estruturas apresentadas fez com que eu desenvolvesse uma conveniente intuição para o assunto.

Neste trabalho, as expressões do  $\pi$ -calculus são representadas internamente como árvores e sua execução é, em essência, um percurso em-ordem nesta árvore. Trata-se de uma abordagem bastante simples, até óbvia, e com a qual me sinto confortável por causa estas disciplinas.

### **Métodos Formais em Programação**

Tomei, nesta disciplina, meu primeiro contato com lógica formal. Esta forma estruturada de raciocinar, por meio de regras de inferência, me ajudou posteriormente a entender o  $\pi$ -calculus e as regras de transição que determinam sua semântica.

### **Métodos Formais para Especificação e Construção de Programas**

O  $\pi$ -calculus, teoria sobre a qual este trabalho se sustenta, foi-me apresentado nesta disciplina. É natural, portanto, que esta seja mencionada aqui pois, sem ela, eu dificilmente teria tomado conhecimento deste cálculo de processos.

### **Programação Concorrente**

Nesta disciplina aprendi os mecanismos e conceitos fundamentais que sustentam programas concorrentes. Muitos deles, tais como *threads*, *locks* e seções críticas, foram utilizados e exercitados constantemente no código produzido neste trabalho.

Além disso, a grande dificuldade em trabalhar neste nível de abstração se tornou uma motivação a mais para o desenvolvimento do TCC, despertando meu interesse para novas formas de pensar e estruturar programas que fazem uso de concorrência.

## **3 Continuidade**

Pretendo continuar trabalhando, sob a orientação da professora Ana Cristina, neste projeto. Ainda há uma quantidade significativa de pontos nos quais ele pode ser melhorado e acreditamos, também, que ele possa ser utilizado como ponto de partida para outros estudos no tema.