

Ocaml

Autores : Felipe Assad , Jorge Chagas, Thiago

Data : 26/04/2019

Universidade Federal Fluminense



Objective Categorical Abstract Machine Language

- Ocaml é uma linguagem multiparadigma (Orientado a objetos, Imperativo e Funcional)
- Robert Milner em 1978 cria ML(Meta Language) predecessora da CAM
- Compilação nativa(ocamlopt) ou para bytecode(ocamlc)

- PROS:

- Tipagem Estática e Forte aliada a um bom parser.
- Definição de tipos e Pattern Matching
- Confiabilidade e Abstração

- CONS:

- Não tem tantas bibliotecas quanto as linguagens mais populares
- Não tem um bom suporte a processadores multicore em relação a outras linguagens

Objetivos desta apresentação

- Mostrar o Lexer e o Parser funcionando através do exemplo da calculadora
- Funcionamento de expressões booleanas
- Funcionamento de comandos

O que foi feito

- Configuração do lexer(`ocamllex`) e o parser(`ocamlyacc`)
- Parser para a linguagem Imp com operações aritméticas, Booleanas e Comandos

O que não foi feito e porque

- Automato(em andamento)

Por quê?

- Estávamos com dificuldades na hora de definir expressões aritméticas e booleanas
- Definir o modo que o automato irá operar(Pattern matching ou Imperativo)

- Construção do automato
- Comandos e Memória

Avaliação da evolução do trabalho

- Ambientar-se com a linguagem escolhida e suas ferramentas.(OK)
- Implementar um parser para a linguagem Imp-0, contendo operações aritméticas, Booleanas e comandos.(OK)
- Automato(em andamento)
 - *FALTANDO: Implementar lib-mark0: (i) Interpreting Automata
Implementar um compilador de Imp-0 com operações aritméticas, Booleanas e comandos para lib-mark0.*