Fourth Year Project

Scheme To Lua Translator

George Boyle

Basic Info

- Exploration of the similarities and differences between the languages, using the translator as a vehicle
- Written in Lua
- Git - http://github.com/Dockheas23/scheme2lua

Languages

Scheme (1975)

- Dialect of Lisp (1958)
- Minimalist design
- Small standard core
- Widely used in education
- Used as a scripting language for applications

Lua (1993)

- Small embeddable scripting language
- Originally designed for use by engineers in an oil company
- Fast and portable
- Very popular for games

Language Features

Scheme

- Data structure: List
- First class functions
- Functional in spirit
- Heavily expression-based
- Continuations

Lua

- Data structure: Table
- First class functions
- Imperative, but flexible
- Expressions and statements
- Coroutines

Approaches To Translation

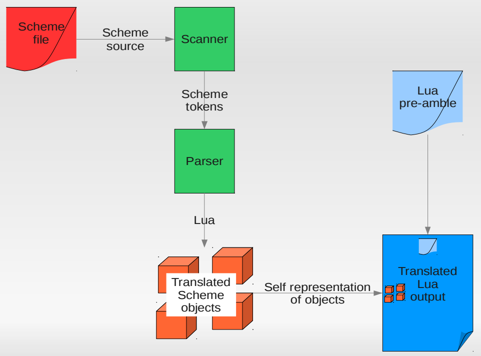
Human ("le mot juste")

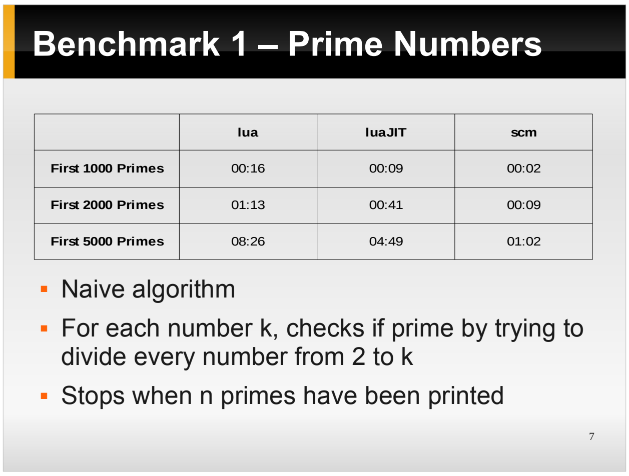
- Flexible
- Handles exceptions
- Very precise and efficient
- Requires substantial experience and intelligence
- Difficult to implement

Structured

- Abstract framework
- Library-based
- Uniform
- Neater design
- Less efficient
- Easier to implement and extend

Overview Of Translator

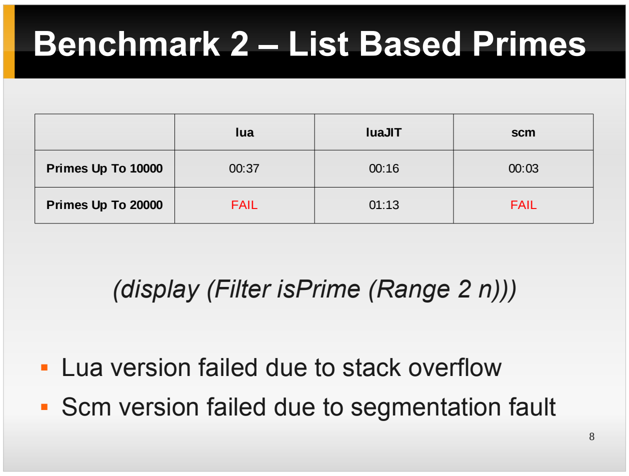




Benchmark 1 – Prime Numbers

	lua	luaJIT	scm
First 1000 Primes	00:16	00:09	00:02
First 2000 Primes	01:13	00:41	00:09
First 5000 Primes	08:26	04:49	01:02

- Naive algorithm
- For each number k, checks if prime by trying to divide every number from 2 to k
- Stops when n primes have been printed

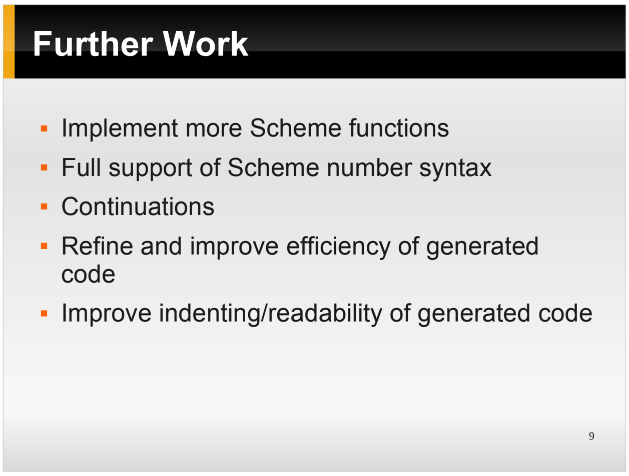


Benchmark 2 – List Based Primes

	lua	luaJIT	scm
Primes Up To 10000	00:37	00:16	00:03
Primes Up To 20000	FAIL	01:13	FAIL

(display (Filter isPrime (Range 2 n)))

- Lua version failed due to stack overflow
- Scm version failed due to segmentation fault



Further Work

- Implement more Scheme functions
- Full support of Scheme number syntax
- Continuations
- Refine and improve efficiency of generated code
- Improve indenting/readability of generated code