Compiler Construction
[VSS]: I'm building stacks

Flip van Spaendonck & Lars Kuijpers

May 14, 2018

Semantics

- Call-by-reference
- Print only prints basic types (others as pointers)

Code generation

- Each token has an addCodeToStack(stack, counter) function
- Call this for tree node
- Each node asks his children to add their code to the codestack
- Counter used to make unique labels

Compilation Schemes (1/2)

Function calls

- Give a unique name to each function to use as label
- Put arguments' code on the codestack
- Results will be put as variables on memory stack
- Jump to code for function body (save origin point to stack)
- Return will set PC back to origin point
- Result of function (if any) will be on top of stack

Compilation Schemes (2/2)

Tuples

- Two adjacent addresses
- Left is pointer address -1, right is pointer address

Lists

- Nested tuples
- Left is pointer to the variable in the current position
- Right is pointer to the next tuple in the list
- Empty list is 0

Extensions

Memory management

- Garbage collection on function exit
- Efficient use of registers

Higher-order functions

- How to pass functions as argument?

Custom structures

- Multuples instead of tuples
- Structs

IN·DEI·NOMINE

Fun Metrics

LOC	3616	a
Word count	9684	a
Character count	98470	aaa
Slides made	17	25
Coffee consumed	0	Still 0
Sanity lost	A bunch	Even more
Experience	Priceless	Pricelesser

Questions ?