Superoperators in Interpreters

Sam Martin
January 27 2011

Important Questions

What do we care about with interpreters?
How can we make programs take less space?

hti

Hybrid translator/interpreter
Input: ANSI C
Outputs: SPARC and MIPS R3000
Written in Assembly

hti

Hybrid translator/interpreter
Input: ANSI C
Outputs: SPARC and MIPS R3000
Written in Assembly
why?

Superoperators

x*2 + 1

z*3 + 1

y + 5

x*2 + 1

t + t + x

x*2 + 1

Do they help?

How much slower can we expect interpreted code

Do they help?

How much slower can we expect interpreted code
Around 8 to 16 times slower (eep)

Do they help?

How much slower can we expect interpreted code
Around 8 to 16 times slower (eep)
What if we use superoperators?

Do they help?

How much slower can we expect interpreted code
Around 8 to 16 times slower (eep)
What if we use superoperators?
Only 3 to 9 times slower (yay!)

How hti works

hti actually compiles function prologues
hti uses the output of lcc's intermediate representation
lcc produces Syntax Trees
How many evaluation stacks does hti use?

Room for Superoperators

Lcc only produces 109 operators in its standard trees

We have enough room for 147 Superoperators now (256-109)

Can we make more room?

Optimizations

What do we optimize for...

Optimizations

What do we optimize for...
Size?

Optimizations

What do we optimize for...
Size?
Time?

Unused bytecodes

We can eliminate opcodes that are never used in a particular program

Now we have more possibilities for Superoperators

Drawbacks?

Questions?