RICE'S THEOREM

AND WHAT IT MEANS FOR YOU! :)

Created by **Simon Cross** / **@hodgestar**

WHAT THE HECK IS RICE'S THEOREM?

SOME THINGS ARE JUST UNPOSSIBLE!

Which things? Glad you asked! :)

PROPERTIES OF COMPUTABLE PARTIAL FUNCTIONS

FORMAL(ISH) STATEMENT

Given a **non-trivial property** of computable partial **functions** the problem of whether a particular program computes such a function is undecidable

NON-TRIVIAL

- A least one function that **has** the property
- A least one function that **doesn't have** the property

See **Wikipedia** for the gory details. :)

KINDS OF PROPERTIES

WORKS FOR:

Properties that refer to behaviour

Properties that refer to input and output only

DOESN'T WORK FOR:

Properties that refer to the implementation

- Number of lines of code
- Number of for-loops
- Whether the code is syntactically correct

SKETCH OF A PROOF ...

... IN PYTHON!

```
def has_cool_property(f):
    """Returns True if f has this cool property,
       False otherwise."""
    # XXX: Put magic here!

def tiny_cool_function(*args, **kw):
    """Tiny function we wrote to show the cool property
       is sometimes met."""
    # XXX: Write this.

def halts(a, i):
    """Returns True if a halts given input i."""
    def t(*args, **kw):
        a(i)  # this is just here to mess with has_cool_property
        return tiny_cool_function(*args, **kw)
    return has_cool_property(t)
```

SO THESE IMPLICATIONS YOU PROMISED?

QA AND TESTING

SECURITY AND SANDBOXES

COMPILING AND STATIC ANALYSIS

NON-IMPLICATIONS

a.k.a. things that might work

UPFRONT DESIGN

If you want a property, design it in up front ...

because you won't be able to check later. :/

HALT?

BY SIMON CROSS / HODGESTAR.ZA.NET

WHAT MADE YOU LOOK INTO ALL THIS?!

BLAME PERL!

It's unparsable. :/

ITS GRAMMAR IS AMBIGUOUS IN CRAZY WAYS:

```
whatever / 25 ; # / ; die "this dies!";
```

IF YOU EVER NEED TO FEEL GREAT ABOUT USING PYTHON:

- **On Parsing Perl** by *Randal Schwartz* (2000)
- **PPI man page** by *Adam Kennedy*
- **Perl Cannot be Parsed: A Formal Proof** by *Jeffrey Kegler* (2008)

AN UNRELATED COOL THING:

- **Python as a First Language** by *John M. Zelle*

“Compiling a program and staring at a screenfull of nagging messages is a dull and exasperating activity.”

“In fact, I would venture to say that after a couple days playing with Python, many faculty who are now teaching C++ would know Python better.”

IDEAS GENERATED BY QUESTIONS AND COMMENTS AFTER THE TALK

- Scalability is another property one can't tack on to programs afterwards.
- JIT compilation also benefits from Rice's Theorem since it imposes strong bounds on what static compilation can prove about the algorithm implemented.