Complexity metrics

Designing and Maintaining Software (DAMS)

Louis Rose

Size != Complexity

“Imagine a small (50 line) program comprising 25 consecutive "IF THEN" constructs. Such a program could have as many as 33.5 million distinct control paths.”

- Thomas J. McCabe

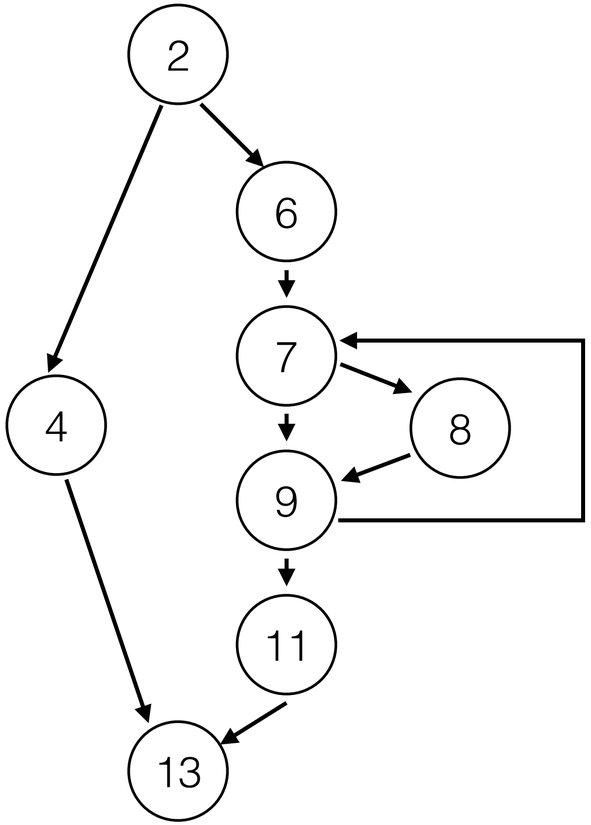
IEEE Transactions on Software Engineering, 2:4, 1976

http://literateprogramming.com/mccabe.pdf

Cyclomatic Complexity

Counts independent paths in (part of) a program

```
1 class Pizza
2  def title
3    if @toppings.empty?
4      "Margherita"
5    else
6      title = ""
7      @toppings.each do |topping|
8        title += " and " unless title.empty?
9        title += topping
10     end
11     title
12   end
13 end
14 end
```



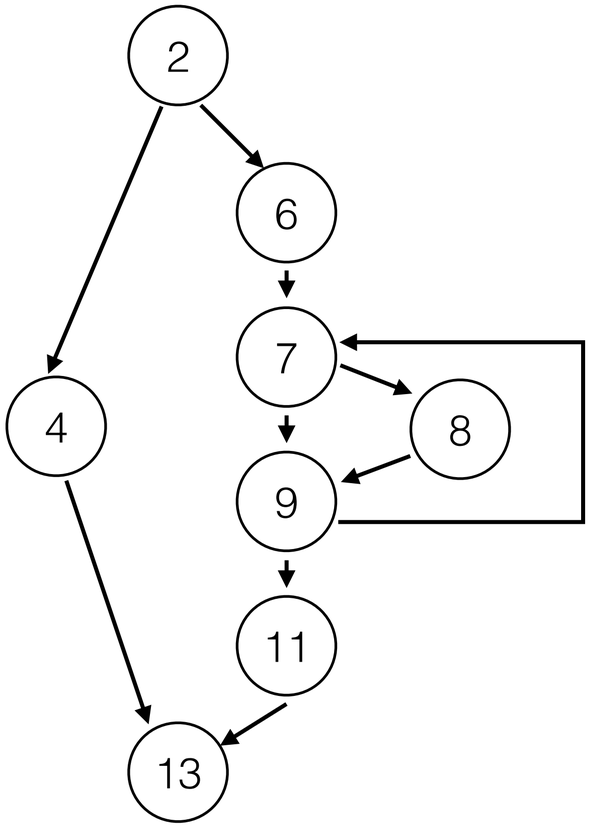
Cyclomatic Complexity

Counts independent paths in (part of) a program

$$m = e - n + 2$$

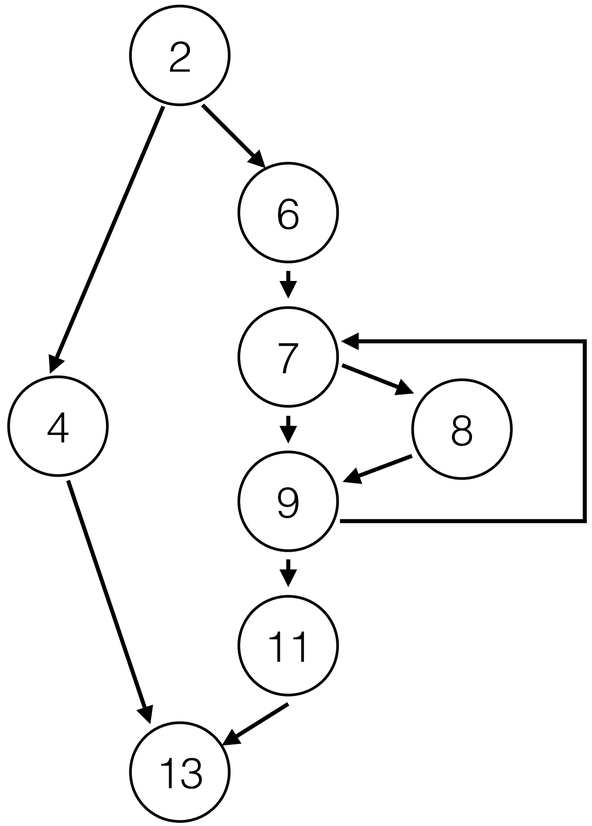
$$m = 10 - 8 + 2$$

$$m = 4$$



Problem

Constructing CFGs isn’t straightforward.



Solution

Approximate complexity from AST by counting:
Assignments, **B**ranches, **C**onditions

```
1 class Pizza
2  def title
3    if @toppings.empty?
4      "Margherita"
5    else
6      title = ""
7      @toppings.each do |topping|
8        title += " and " unless title.empty?
9        title += topping
10      end
11      title
12    end
13  end
14 end
```

ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions

```
1 class Pizza
2  def title
3    if @toppings.empty?
4      "Margherita"
5    else
6      title = ""
7      @toppings.each do |topping|
8        title += " and " unless title.empty?
9        title += topping
10      end
11      title
12    end
13  end
14 end
```

- Jerry Fitzpatrick
http://www.softwarerenovation.com/ABCMetric.pdf

ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions

```
1 class Pizza
2  def title
3    if @toppings.empty?
4      "Margherita"
5    else
6      title = ""
7      @toppings.each do |topping|
8        title += " and " unless title.empty?
9        title += topping
10     end
11     title
12   end
13 end
14 end
```

ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions

```
1 class Pizza
2  def title
3    if @toppings.empty?
4      "Margherita"
5    else
6      title = ""
7      @toppings.each do |topping|
8        title += " and " unless title.empty?
9        title += topping
10      end
11      title
12    end
13  end
14 end
```

ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions

```ruby
class Pizza
 def title
  if @toppings.empty?
    "Margherita"
  else
    title = ""
    @toppings.each do |topping|
      title += " and " unless title.empty?
      title += topping
    end
    title
  end
end
end
```

ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions

```
1 class Pizza
2  def title
3    if @toppings.empty?
4      "Margherita"
5    else
6      title = ""
7      @toppings.each do |topping|
8        title += " and " unless title.empty?
9        title += topping
10     end
11     title
12   end
13 end
14 end
```

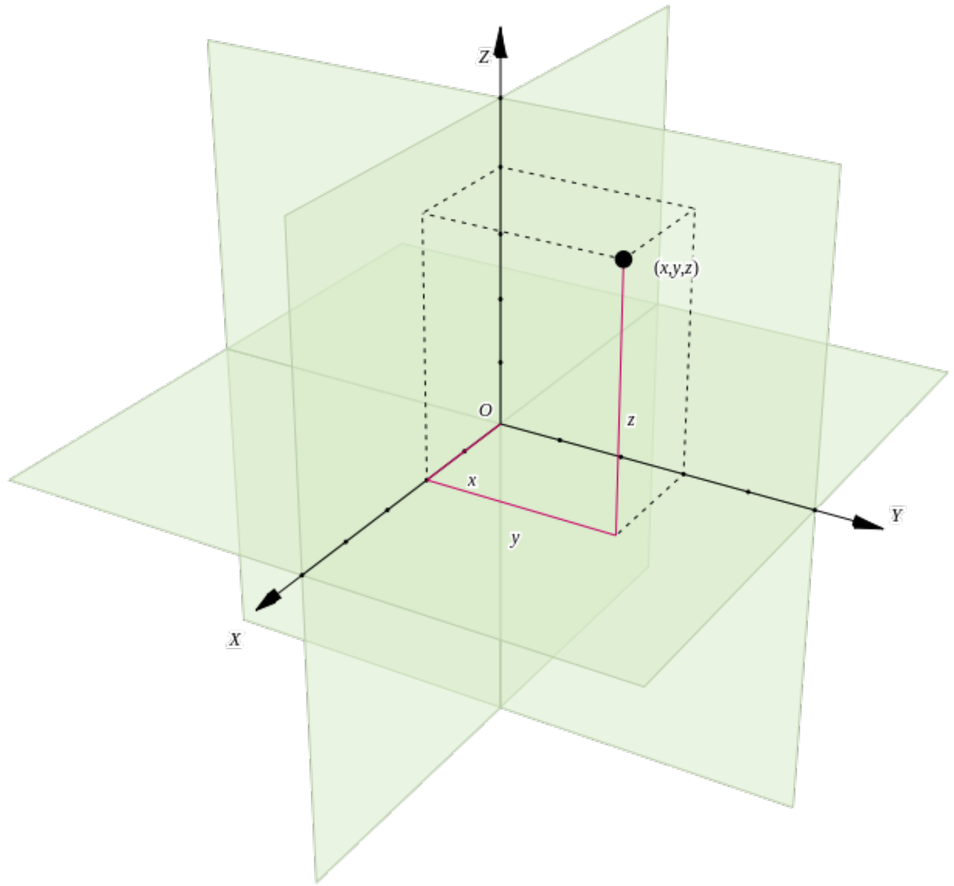
ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions

```
class Pizza
  def title
    3 3 1
  end
end
```

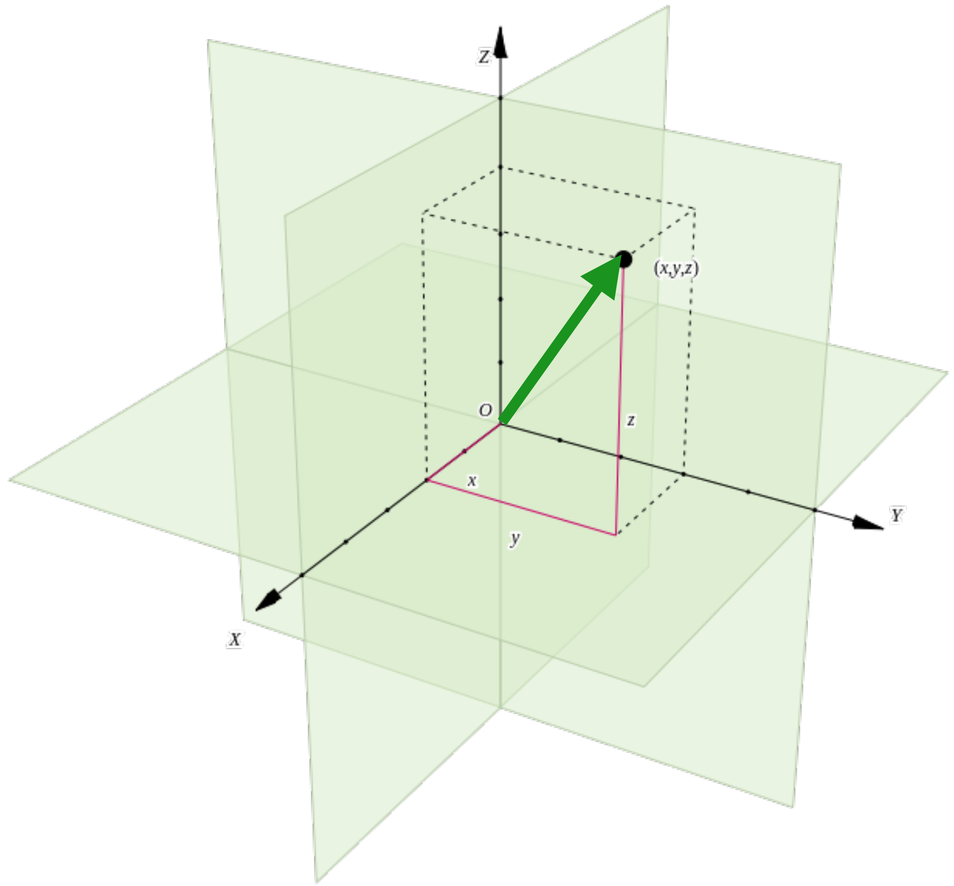
ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions



ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions



ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions

$$abc = \sqrt{(a^2 + b^2 + c^2)}$$

ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions

$$abc = \sqrt{(a^2 + b^2 + c^2)}$$

```
class Pizza
  def title
    3 3 1
  end
end
```

ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions

$$abc = \sqrt{(3^2 + 3^2 + 1^2)}$$

```
class Pizza
  def title
    3 3 1
  end
end
```

ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions

$$abc = \sqrt{(9 + 9 + 1)}$$

```
class Pizza
  def title
    3 3 1
  end
end
```

ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions

$$abc = \sqrt{19}$$

```
class Pizza
  def title
    3 3 1
  end
end
```

ABC Metric

Approximate complexity from AST by counting:
Assignments, Branches, Conditions

abc = 4.36

```
class Pizza
  def title
    4.36
  end
end
```

God Method

What is it?
A very large method (relatively speaking)

Why is it problematic?
Inhibits OO benefits: explanation, sharing, choosing

When does it arise?
👎🏼 Poor grasp of OO programming
👍🏼 ?

Obtuse Method

What is it?
A very complex method (relatively speaking)

Why is it problematic?
Inhibits OO benefits: explanation, sharing, choosing

When does it arise?
👎🏼 Poor grasp of OO programming
👍🏼 ?

Summary

Complex methods hinder habitability.

Cyclomatic (McCabe) Complexity and ABC metrics can identify complex methods.

ABC is less accurate than CC,
but much easier to compute.