

bug.c

```
double bug(double z[], int n) {
    int i, j;
    i = 0;
    for (j = 0; j < n; j++) {
        ____i = i + j + 1;
        _____i</pre>
```

```
z[i] = z[i] * (z[0] + 1.0);
```

```
return z[n];
```

```
٢
```



What do we do now? We can follow Platon and say: Hey, let's just verify this compiler, let's do more abstraction, let's do more of the same. (This is what I learned in school: The state of the art is bad, but if only people would do it our way, than the world would be a

Locating Errors

An error is a deviation from what is correct, right, or true:

- Input ("The URL must be well-formed")
- Variables ("link is zero")
- Statements ("even(2) must return true")

How do we know one of these is correct?

How can we say "The defect is here"?

4



http://www.aeroxp.org/2009/01/ lesson-on-infinite-loops/ http://www.youtube.com/watch? v=fYTJ9v2vsaE

5

Locating Causes

An aspect of the execution causes a failure if it can be altered such that the failure no longer occurs:

- Input ("1114")
- Variables ("argc = 2")
- Statements ("Line 37")

Note that a cause need not be an error!

Causality States of the product of causality is deeply linked to fundamental questions of philosophy: What is it that makes things happen? So an we predict the future from causes? If everything has a cause, what is the ultimate cause of events in the past?

9



Aristotle on Causality

Aristotle suggested four types of causes:

- The material of which things come
- The *form* which things have when they are perfected
- The moving cause or actual agent
- The purpose or function of such things

Example Creating a silver chalice for a religious ceremony

- Material cause the silver
- Formal cause the design of the chalice
- Efficient cause the silversmith
- Final cause the religious ceremony









Ockham on Causality

- The only way in which we can establish any causal connection between one thing and another is the observation that when one of these occurs, the other also occurs at the same time and at or near the same place.
- This is the only way to establish causality



13	

Hume on Causality

- When we see that two events always occur together, we tend to form an expectation that when the first occurs, the second will soon follow.
- This constant conjunction and the expectation thereof is all that we can know of causation, and all that our idea of causation can amount to.

14

Causality as Illusion

- Just because the sun has risen every day since the beginning of the Earth does not mean that it will rise again tomorrow.
- Bertrand Russell: "causation = superstition"

Counterfactuals

- We may define a *cause* to be an object followed by another, and where all the objects, similar to the first, are followed by objects similar to the second. Or, in other words, where, *if the first object had not been*, the second never had existed. (Hume, 1748)
- Hume never explored this alternative

Hume also gave an alternate definition of causality, though – a counterfactual one. "Counterfactual" means to reason about the opposite of the current fact (the cause)















More possible causes

invocation	me
electricity	oxygen
	invocation electricity





Lewis on Causation

- C → E means "If C had been the case, E would have been the case"
- C causes E if C \hookrightarrow E and \neg C \hookrightarrow \neg E hold.
- C → E holds if some C-world where E holds is closer to the actual world than is any C-world where E does not hold.



Possible Worlds

 $C \rightarrow E$ holds if some C-world where E holds is *closer to the actual world* than is any C-world where E does not hold.

- A world with an alternate GCC input is closer than a world without oxygen
- A world with GCC fixed may be closer than a world with an alternate GCC input



5		











Isolating Causes

double bug(double z[], int n) { int i, j;

}

}

Actual cause narrowed down

29

Isolating Causes double bug(double z[], int n) { int i, j; i = 0;for (j = 0; j < n; j++) { i = i + j + 1;z[i] = z[i] * (z[0] + 1.0);}





Isolating Causes

double bug(double z[], int n) {
 int i, j;

}

```
i = 0;
for (j = 0; j < n; j++) {
    i = i + j + 1;
    z[i] = z[i] * (z[0] + 1.0);
}
return z[n];
```

Actual cause of the GCC crash











Search Space The choice of an *initial set of differences* determines the search space for causes: the input (data, configuration, ...) the program state the program code Sets a common context between worlds

36

Search Space

Input	State	Code	
OS	Compiler	Processor	
FBI	E.T.	Them!	
	36		

Ockham's Razor

 Whenever you have competing theories for how some effect comes to be, pick the simplest.



37

Ockham's Razor

In our context:

• Whenever you have the choice between multiple causes, pick the one whose alternate world is closer.

38

39

Search Space

Input	State	Code	close
OS	Compiler	Processor	far away
FBI	E.T.	Them!	far out
	39		

Hanlon's Razor

 Never explain by malice which is adequately explained by stupidity



Napoleon, Goethe, Richard Feinman, Robert Heinlein

40

Verifying Causes

\$./psharp db.p#
.psharprc: 37: no such interpreter
.psharprc: 37: bailing out
Segmentation fault

Do we know the configuration in .psharprc causes the failure?

41

Causes and Effects

To prove causality, one must show that

- the effect occurs when the cause occurs
- the effect does *not* occur when the cause does not.

This is the only way to prove causality

Verifying Causes

\$ mv ~/.psharprc ~/.psharprc.orig
\$./psharp db.p#
Segmentation fault

So it wasn't the configuration after all

43

Verifying Causes

\$./psharp db.p#
.psharprc: 37: no such interpreter
.psharprc: 37: bailing out
Segmentation fault

Avoid post hoc ergo propter hoc fallacies

44









What's going on?

double a; a = compute_value(); printf("a = %f\n", a);

We have isolated the format "%d" as the actual failure cause

49

Preemption

Billy and Suzy throw rocks at a bottle. Suzy throws first so that her rock arrives first and shatters the glass.Without Suzy's throw, Billy's throw would have shattered the bottle.

• Does Suzy's throw cause the shattering?

50

Alteration

- C influences E if C can be altered to C' such that E' occurs instead of E (Lewis; 1999)
- If Suzy had not thrown the stone, the bottle would have shattered in a different manner
- Therefore, Suzy's throw *influenced* and caused the original shattering

What's the Failure?

- Every failure has some aspects that we consider relevant
- This choice influences the search for causes
- If the *entire state* of the program is part of the failure, we get very *detailed causes*
- If just one aspect is relevant, we get simpler causes sometimes too simple

52	



	_	
	_	
	_	
	-	
	_	
	_	
	-	
	-	
	-	
This work is licensed under the Creative Commons Attribution License. To view a copy of this license, visit	_	
http://creativecommons.org/licenses/by/1.0		
or send a letter to Creative Commons, 559 Abbott Way, Stanford, California 94305, USA. 54	54	