Prevalence of Confusing Code in Software Projects

Atoms of Confusion in the Wild

Dan Gopstein
NYU

Hongwei Henry Zhou, Phyllis Frankl, Justin Cappos

AtomsOfConfusion.com

Atoms of Confusion in the Wild

```
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
    goto fail;
    goto fail;
```

Atoms of Confusion in the Wild

Apple’s Goto Fail bug

```
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)

    goto fail;

    goto fail;
```

Atoms of Confusion in the Wild

Apple’s Goto Fail bug

```
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
    goto fail;
    goto fail;
```

Two Atoms of Confusion:

- Assignment as Value
- Omitted Curly Brace

Atoms of Confusion in the Wild

Apple’s Goto Fail bug

```
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0) {
    goto fail;
}   goto fail;
```

Two Atoms of Confusion:

- Assignment as Value
- Omitted Curly Brace

Outline

Atoms of Confusion are ...

- **Confusing** - Both in the lab and in the wild
- **Prevalent** - Occurring frequently in practice
- **Buggy** - Causing or correlated with faults

Outline

Atoms of Confusion are ...

- **Confusing** - Both in the lab and in the wild
- **Prevalent** - Occurring frequently in practice
- **Buggy** - Causing or correlated with faults

Atoms of Confusion

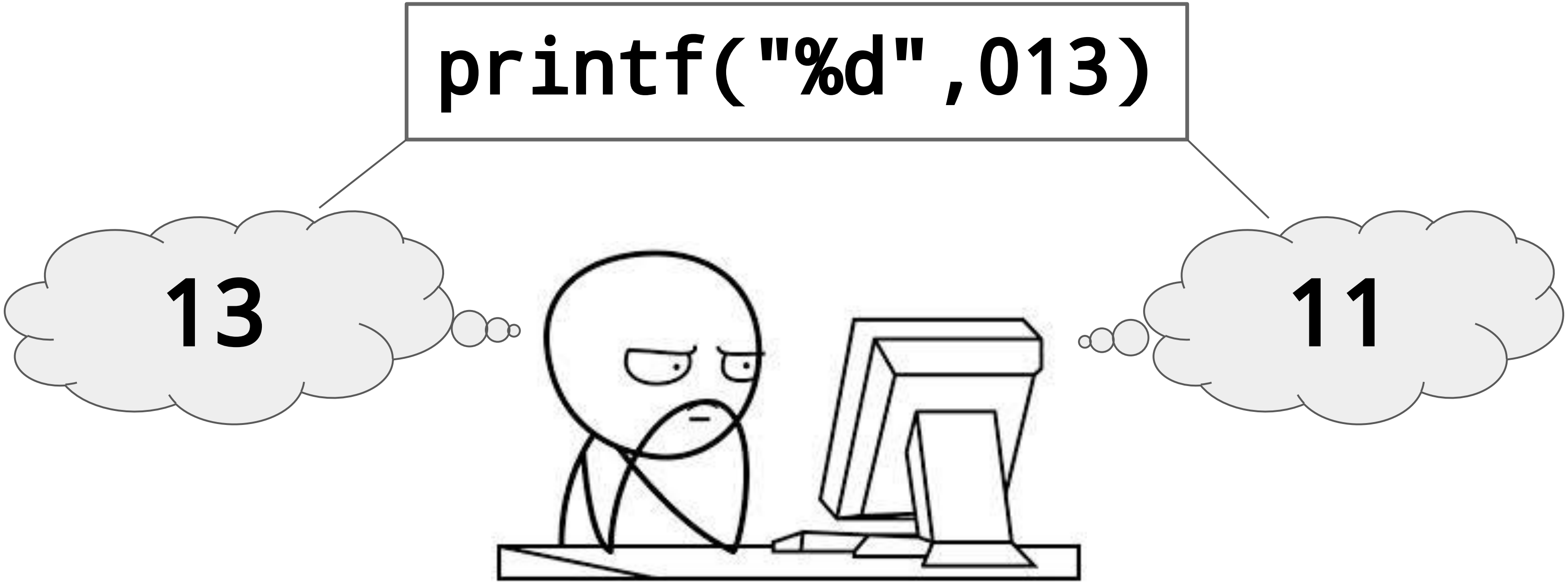
Understanding Misunderstandings in Source Code

D. Gopstein, J. Iannacone, Y. Yan, L. DeLong,
Y. Zhuang, M. Yeh, J. Cappos

ESEC/FSE 2017

Confusion

When a person and a machine read the same piece of code, yet come to different conclusions about its output.



```
printf("%d",013)
```

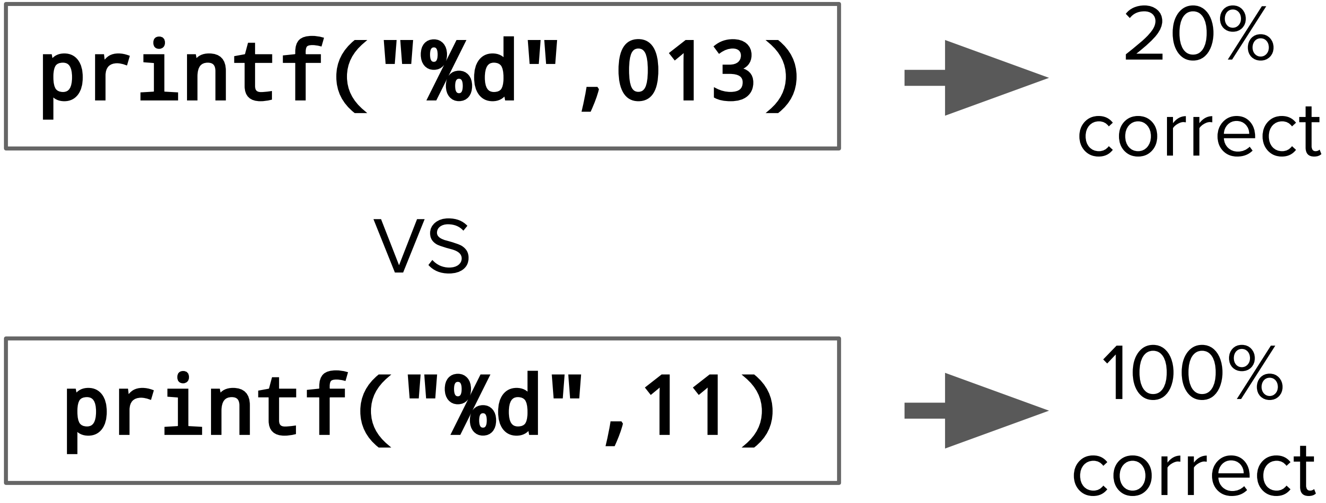
Measurable

```
printf("%d",013)
```

VS

```
printf("%d",11)
```

Measurable



```
printf("%d",013)
```

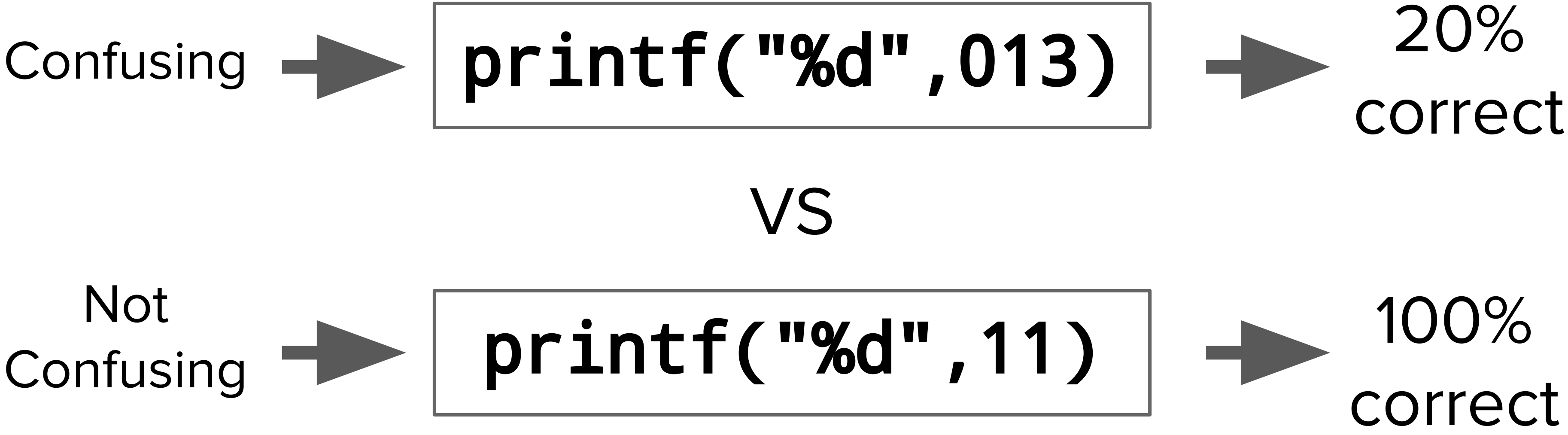
20% correct

VS

```
printf("%d",11)
```

100% correct

Measurable



Confusing

```
printf("%d",013)
```

20% correct

VS

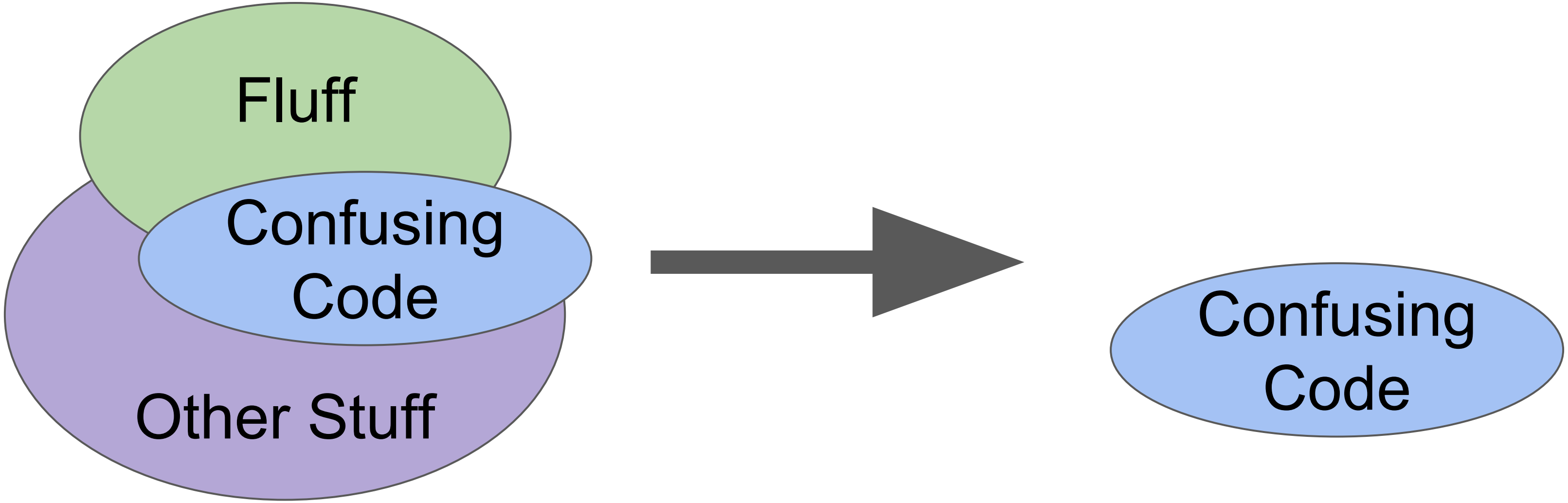
Not Confusing

```
printf("%d",11)
```

100% correct

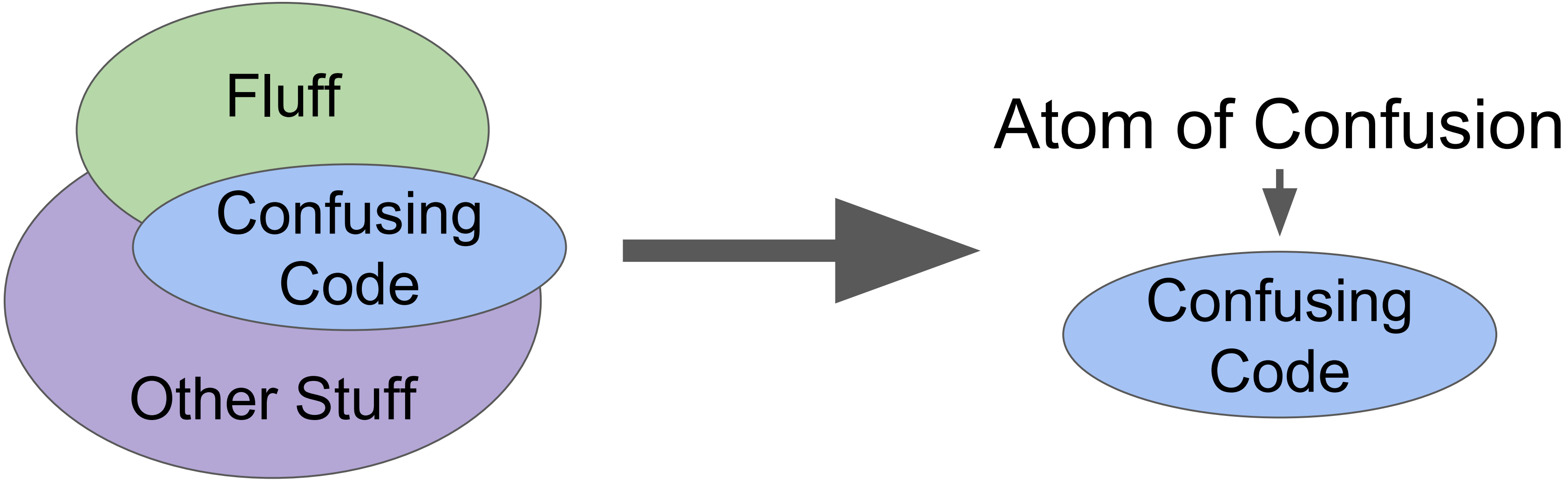
Precise

The smallest piece of code that can cause confusion

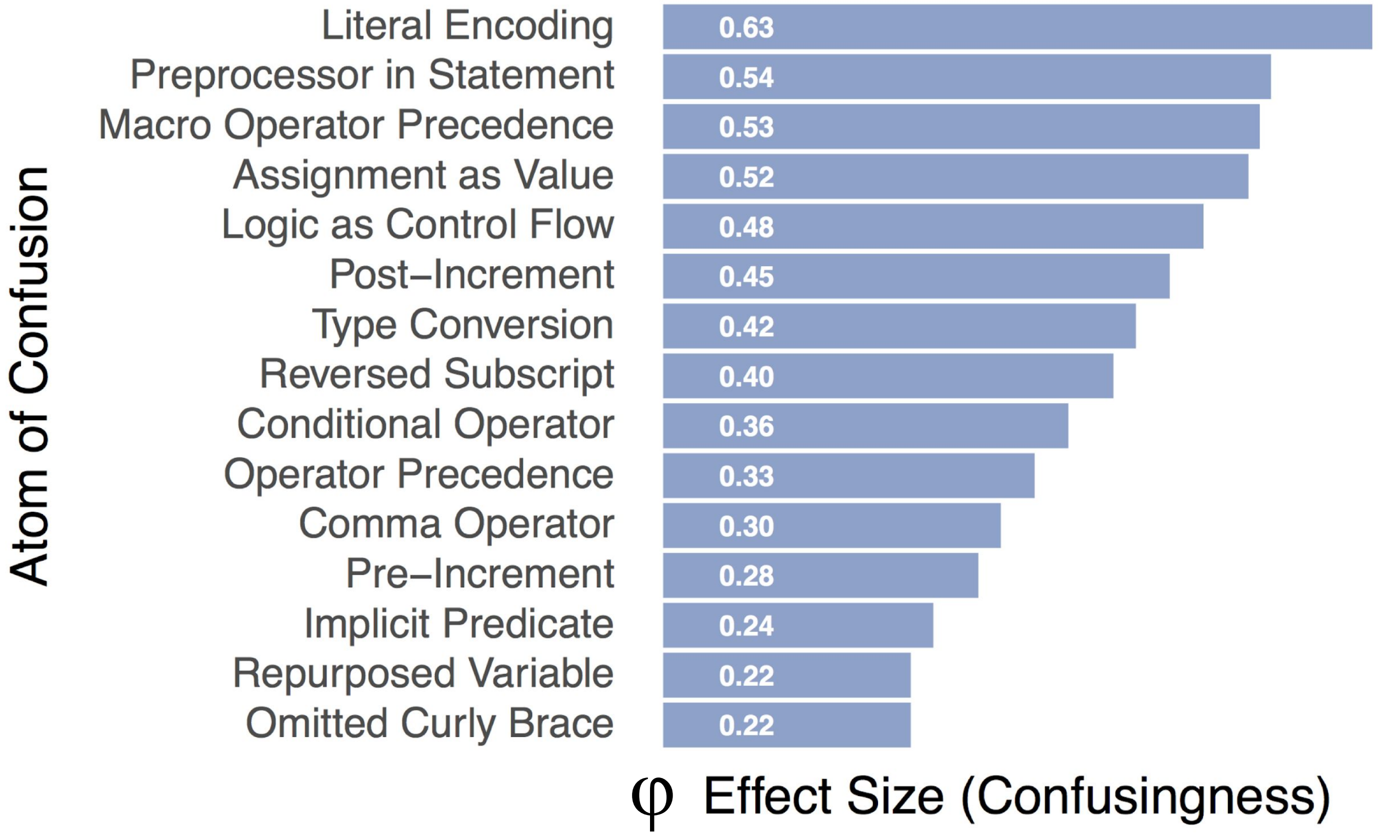


Precise

The smallest piece of code that can cause confusion



Identified Atoms



Atoms of Confusion

Literal Encoding φ = .63

```
printf("%d",013)
```

Logic as Control Flow φ = .48

```
V1 && F2()
```

Operator Precedence φ = .33

```
0 && 1 || 2
```

Pre-Increment φ = .28

```
V1 = ++V2;
```

Understanding Misunderstandings in Source Code
D. Gopstein, J. Iannacone, Y. Yan, L. DeLong, Y. Zhuang, M. Yeh, J. Cappos
ESEC/FSE 2017

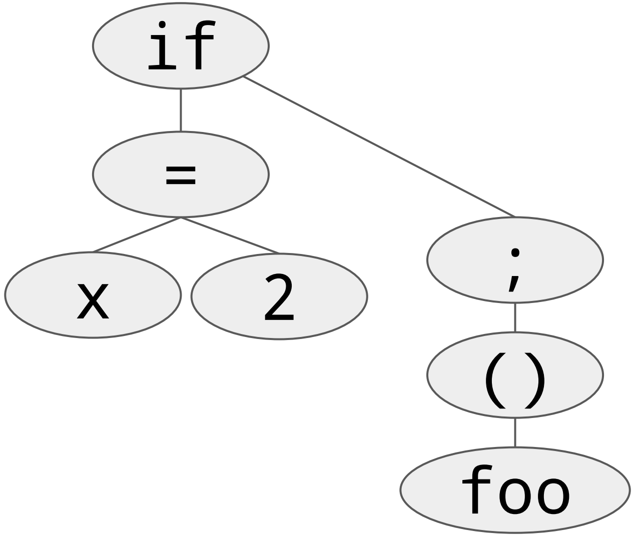
Outline

Atoms of Confusion are ...

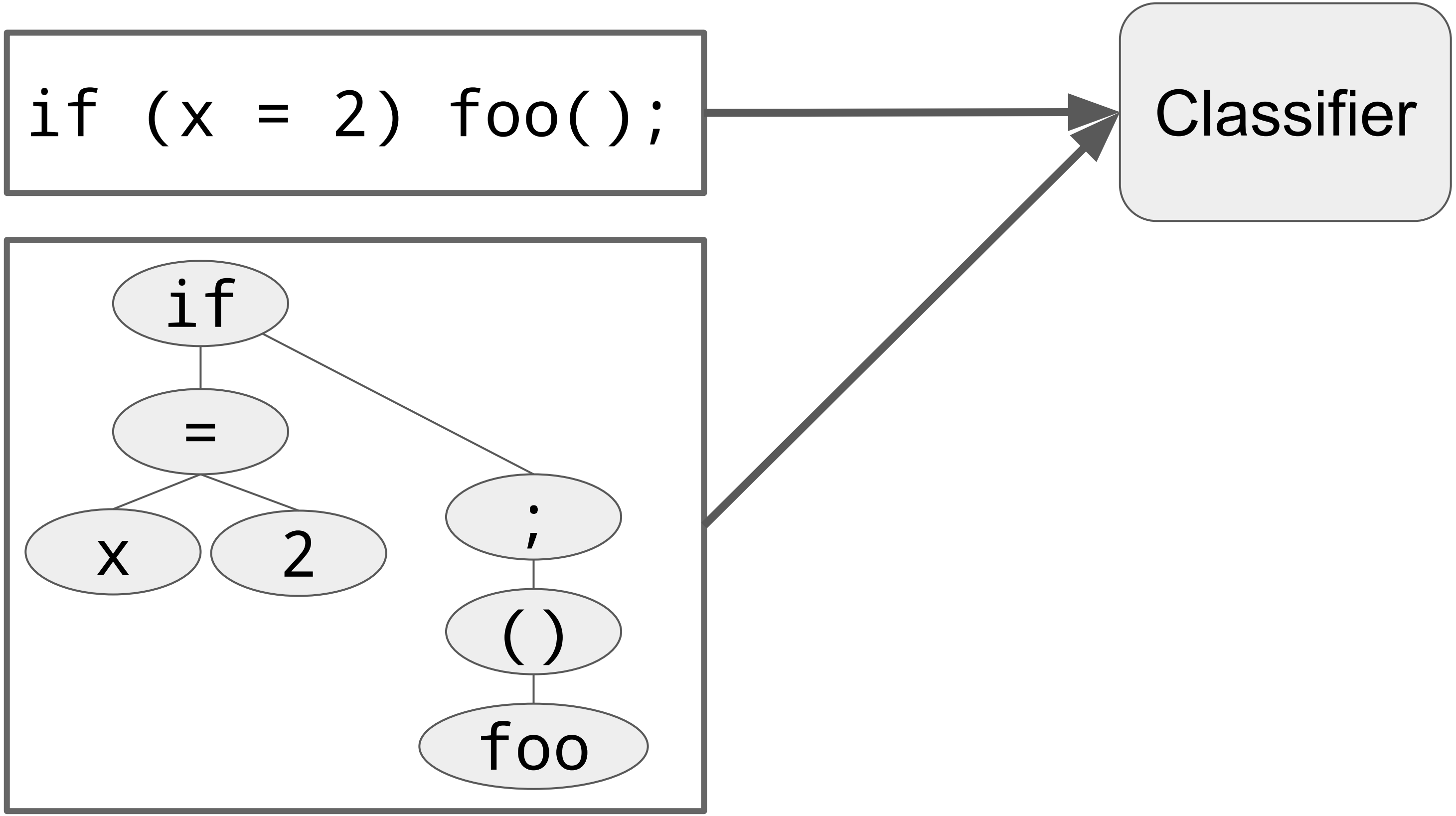
- **Confusing** - Both in the lab and in the wild
- **Prevalent** - Occurring frequently in practice
- **Buggy** - Causing or correlated with faults

Classifier

```
if (x = 2) foo();
```



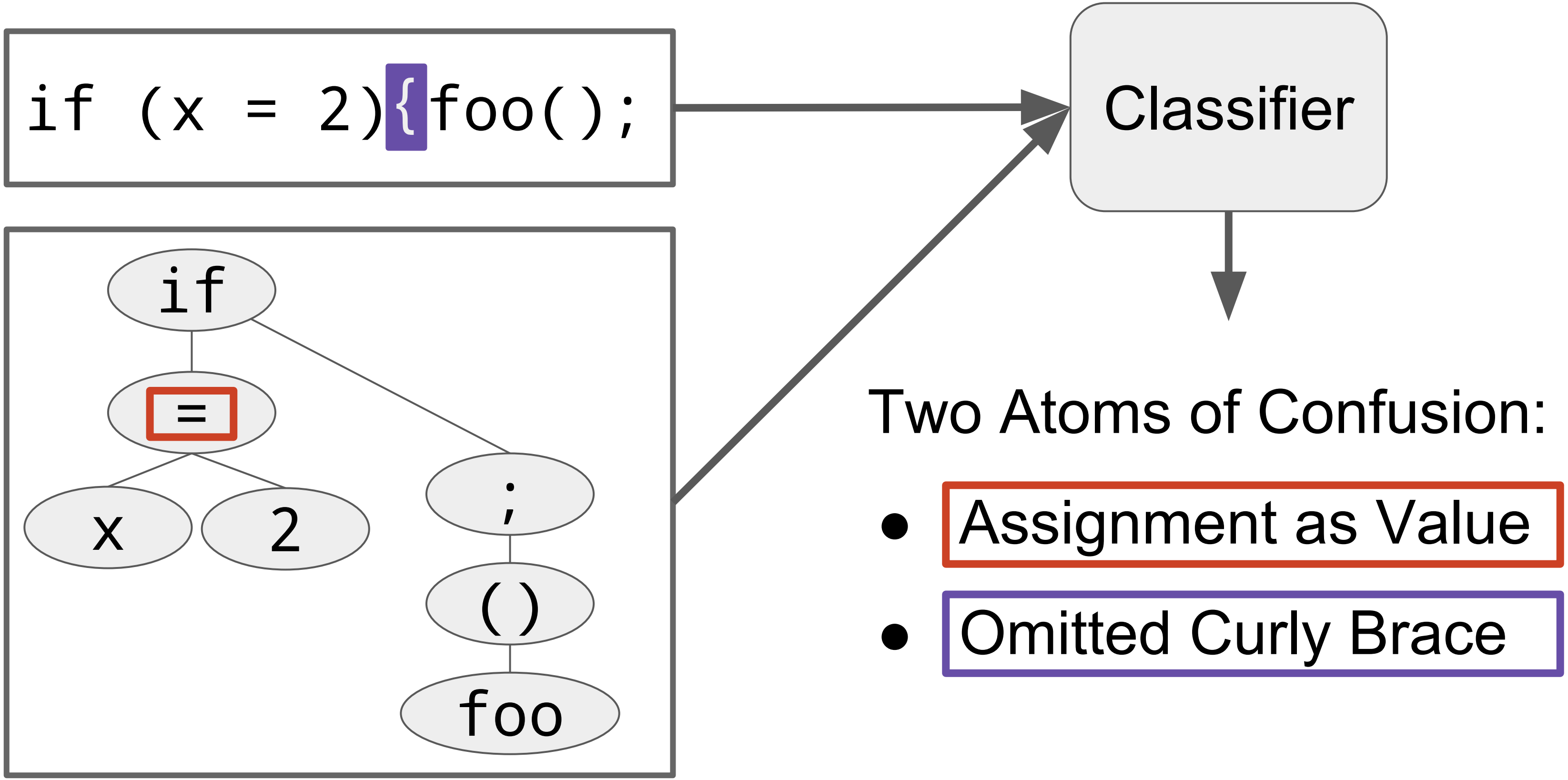
Classifier



```
if (x = 2) foo();
```

Classifier

Classifier

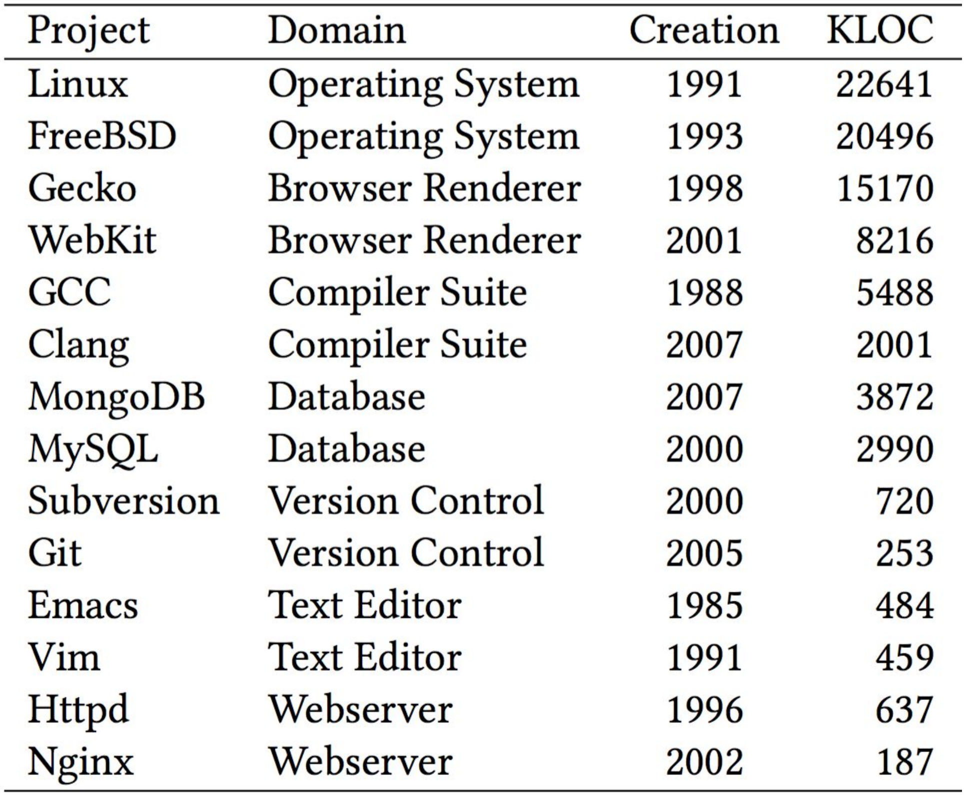


```
if (x = 2){foo();
```

Two Atoms of Confusion:

- Assignment as Value
- Omitted Curly Brace

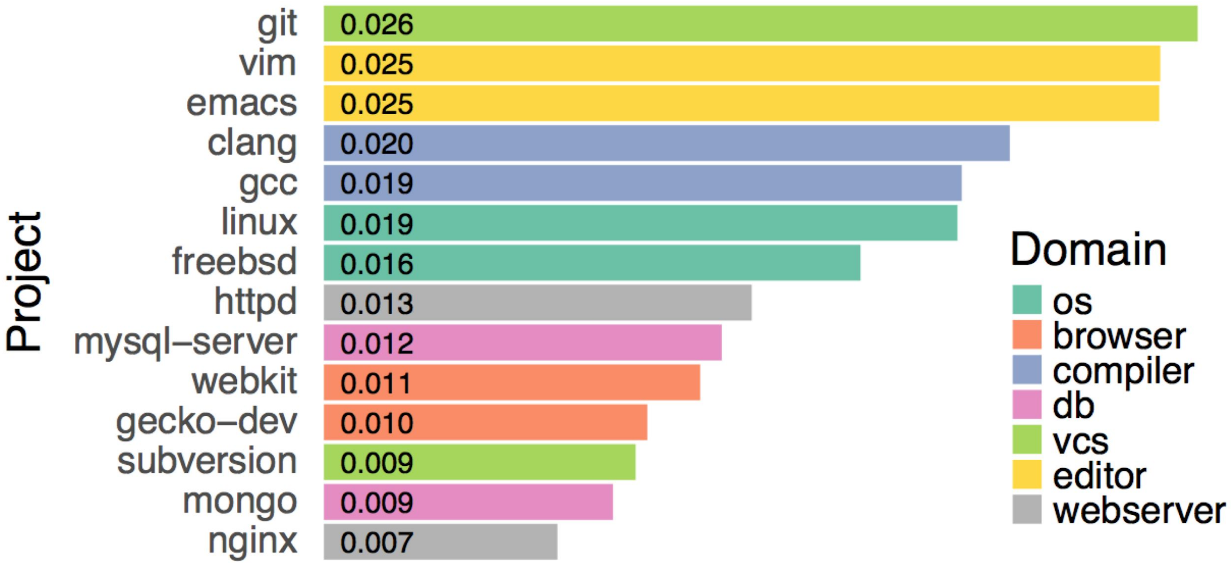
Corpus



Project	Domain	Creation	KLOC
Linux	Operating System	1991	22641
FreeBSD	Operating System	1993	20496
Gecko	Browser Renderer	1998	15170
WebKit	Browser Renderer	2001	8216
GCC	Compiler Suite	1988	5488
Clang	Compiler Suite	2007	2001
MongoDB	Database	2007	3872
MySQL	Database	2000	2990
Subversion	Version Control	2000	720
Git	Version Control	2005	253
Emacs	Text Editor	1985	484
Vim	Text Editor	1991	459
Httpd	Webserver	1996	637
Nginx	Webserver	2002	187

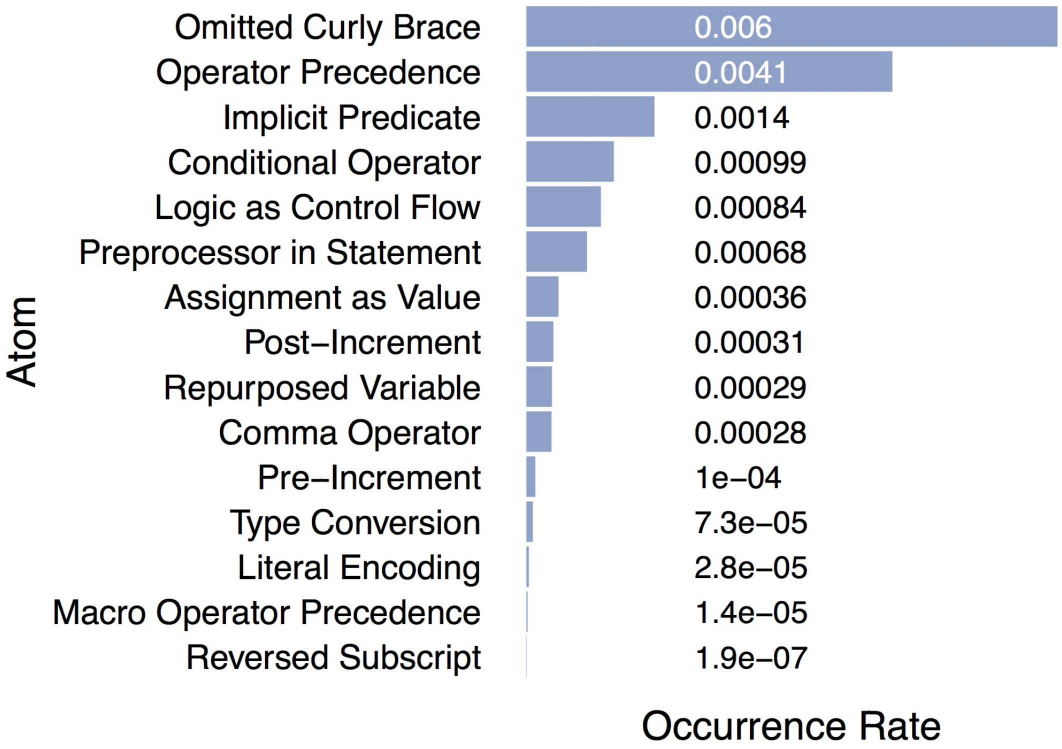
How Often do Atoms Occur?

1 atom every ~12 lines



1 atom every ~44 lines

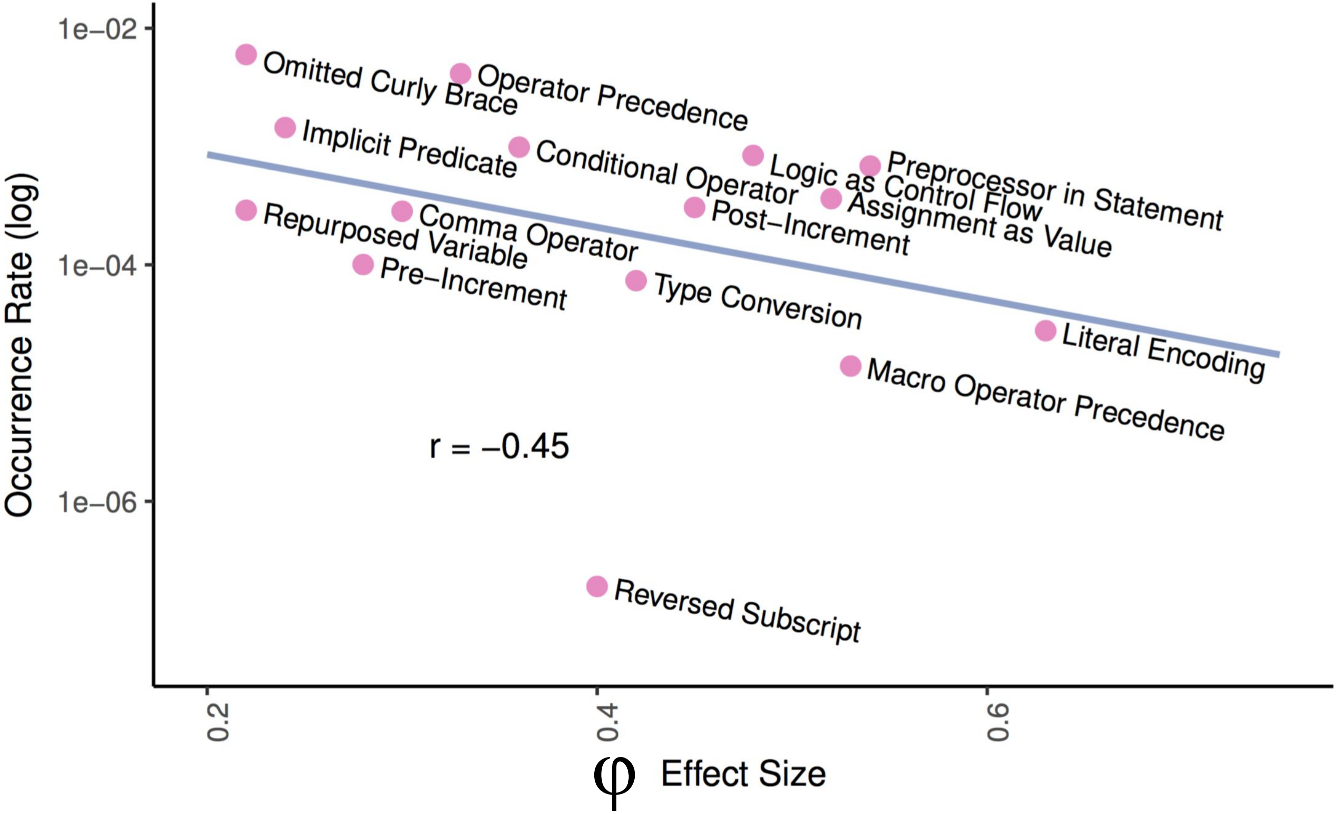
Which Atoms Occur Most Frequently?



1 every ~51 lines

1 every ~1.6 million

Are Confusing Patterns Less Common?



Prevalent

```
ulpmc->cmd = htobe32(V_ULPTX_CMD(ULP_TX_MEM_WRITE) |
    is_t4(sc) ? F_ULP_MEMIO_ORDER : F_T5_ULP_MEMIO_IMM);
```

https://github.com/freebsd/freebsd/blob/3c60e22da7d4460db7adb2b916f55e22b7d60e26/sys/dev/cxgbe/tom/t4_ddp.c#L766

Prevalent

```
ulpmc->cmd = htobe32(V_ULPTX_CMD(ULP_TX_MEM_WRITE) |
    is_t4(sc) ? F_ULP_MEMIO_ORDER : F_T5_ULP_MEMIO_IMM);
```

Contains:

- Operator Precedence
- Conditional Operator
- Implicit Predicate

https://github.com/freebsd/freebsd/blob/3c60e22da7d4460db7adb2b916f55e22b7d60e26/sys/dev/cxgbe/tom/t4_ddp.c#L766

Prevalent

```
ulpmc->cmd = htobe32(V_ULPTX_CMD(ULP_TX_MEM_WRITE) |
    is_t4(sc) ? F_ULP_MEMIO_ORDER : F_T5_ULP_MEMIO_IMM);
```

Contains:

- Operator Precedence
- Conditional Operator
- Implicit Predicate

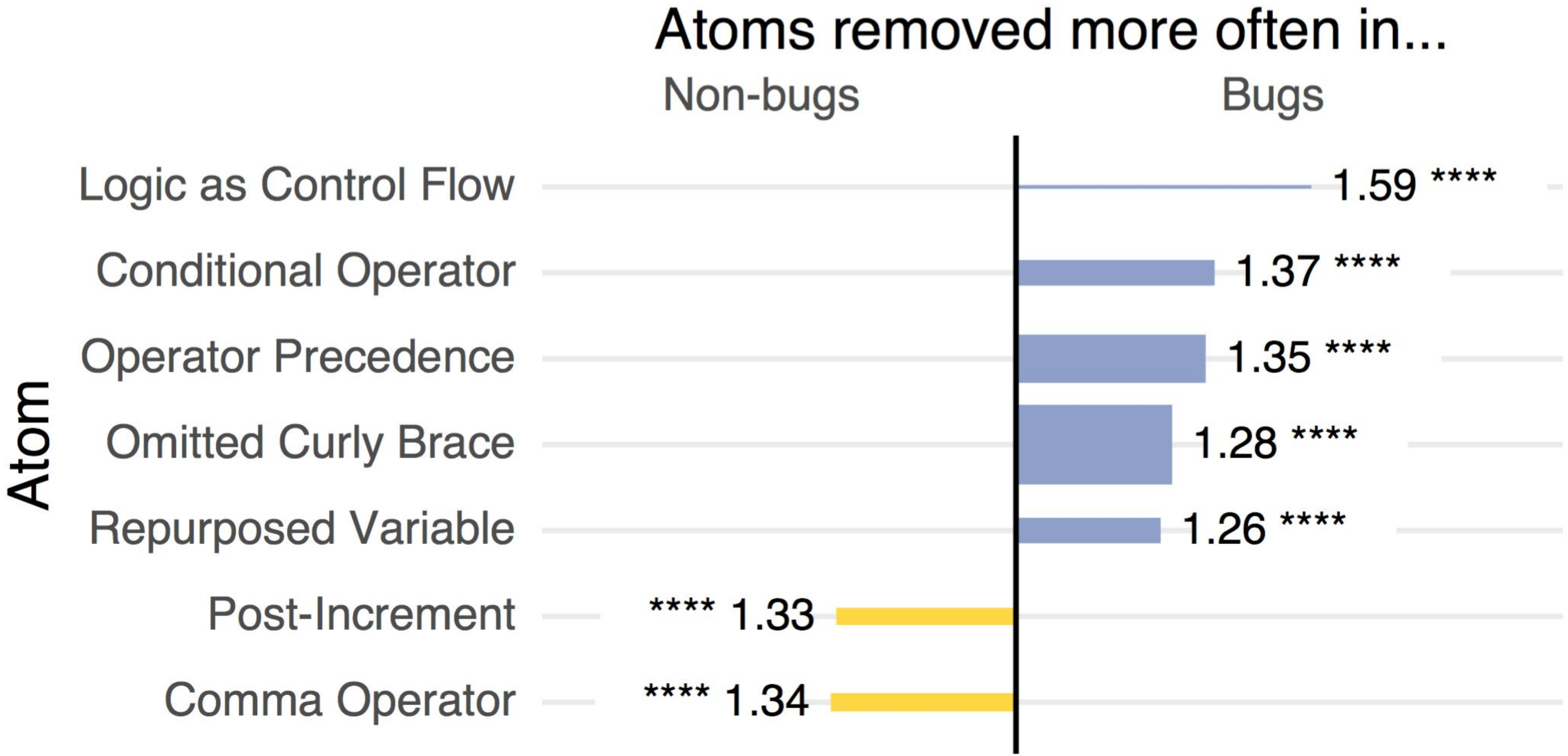
https://github.com/freebsd/freebsd/blob/3c60e22da7d4460db7adb2b916f55e22b7d60e26/sys/dev/cxgbe/tom/t4_ddp.c#L766

Outline

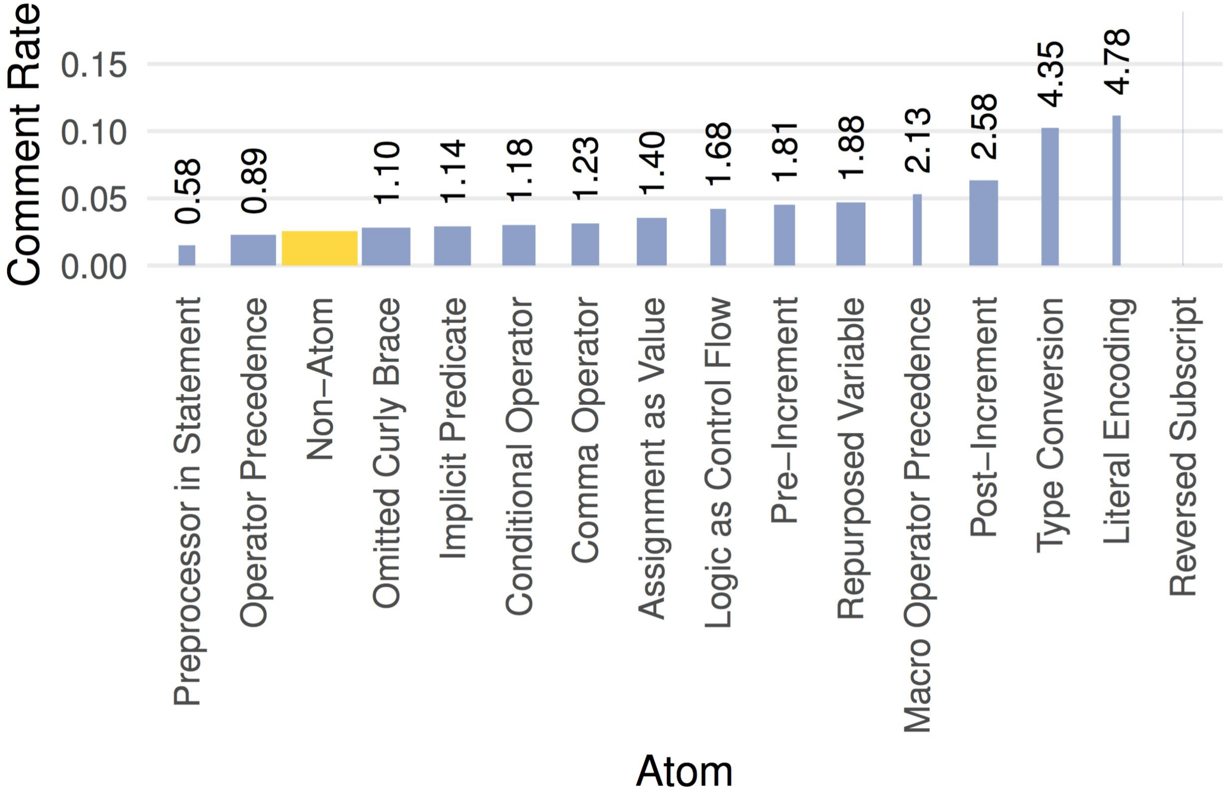
Atoms of Confusion are ...

- **Confusing** - Both in the lab and in the wild
- **Prevalent** - Occurring frequently in practice
- **Buggy** - Causing or correlated with faults

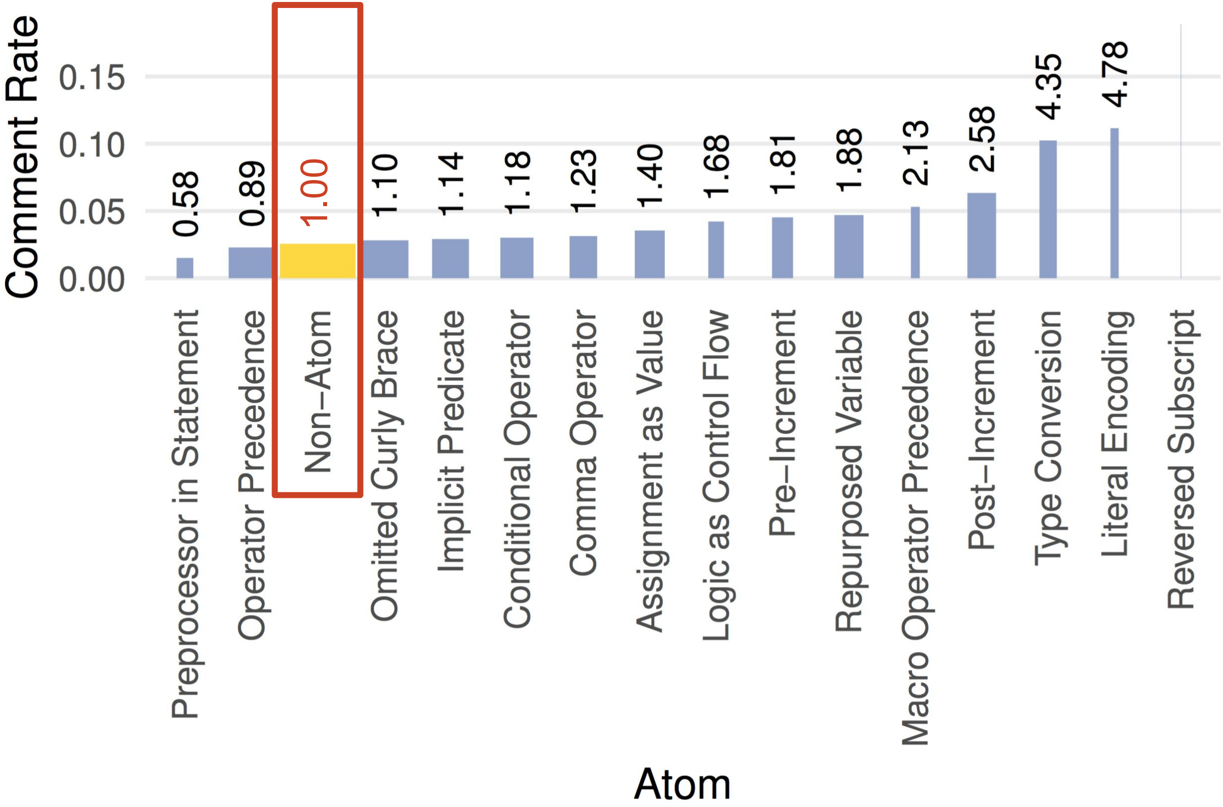
Are Atoms Removed More In Bug Fix Commits?



Are Atoms Commented More Often?



Are Atoms Commented More Often?



Buggy

```c
#define ABS(x) ((x) < 0 ? (-x) : (x))
```

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))
```

```
ABS(1)   => ???
```

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))
```

```
ABS(1)   =>   1
```

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))
```

```
ABS(1)   =>   1
ABS(-2)  => ???
```

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))
```

```
ABS(1)    =>   1
ABS(-2)   =>   2
```

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))
```

```
ABS(1)   =>   1
ABS(-2)  =>   2
ABS(1-2) => ???
```

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))
```

```
ABS(1)   =>   1
ABS(-2)  =>   2
ABS(1-2) =>   1
```

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))
```

ABS(1) => 1

ABS(-2) => 2

ABS(1-2) => ~~1~~ -3

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))
```

```
ABS(1-2)
```

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))
```

```
ABS(1-2)
```

```
(( x ) < 0 ? (- x ) : ( x ))
```

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))
```

```
ABS(1-2)
```

$$\texttt{((}\boxed{\texttt{x}}\texttt{) < 0 ? (-}\boxed{\texttt{x}}\texttt{) : (}\boxed{\texttt{x}}\texttt{))}$$

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))

ABS(1-2)

((1-2) < 0 ? (-1-2) : (1-2))
```

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))

ABS(1-2)

((1-2) < 0 ? (-1-2) : (1-2))
```

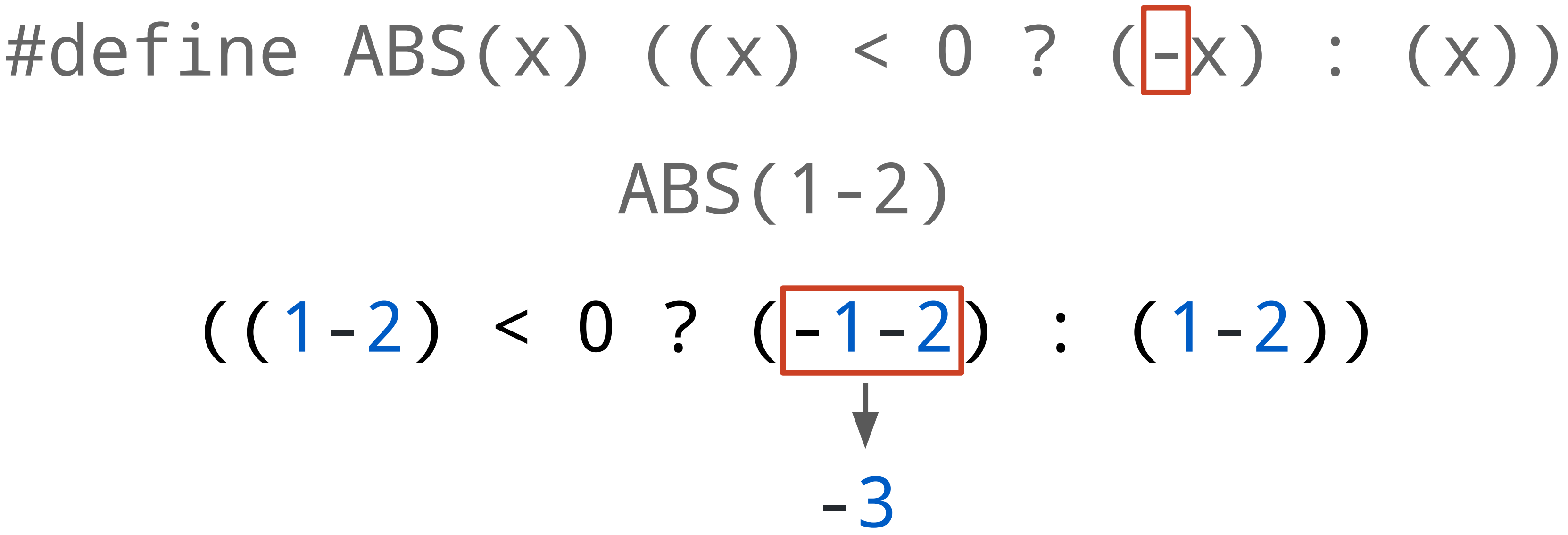
https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))

ABS(1-2)

((1-2) < 0 ? (-1-2) : (1-2))

-3
```

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy



```
#define ABS(x) ((x) < 0 ? (-x) : (x))

ABS(1-2)

((1-2) < 0 ? (-1-2) : (1-2))

-3
```

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

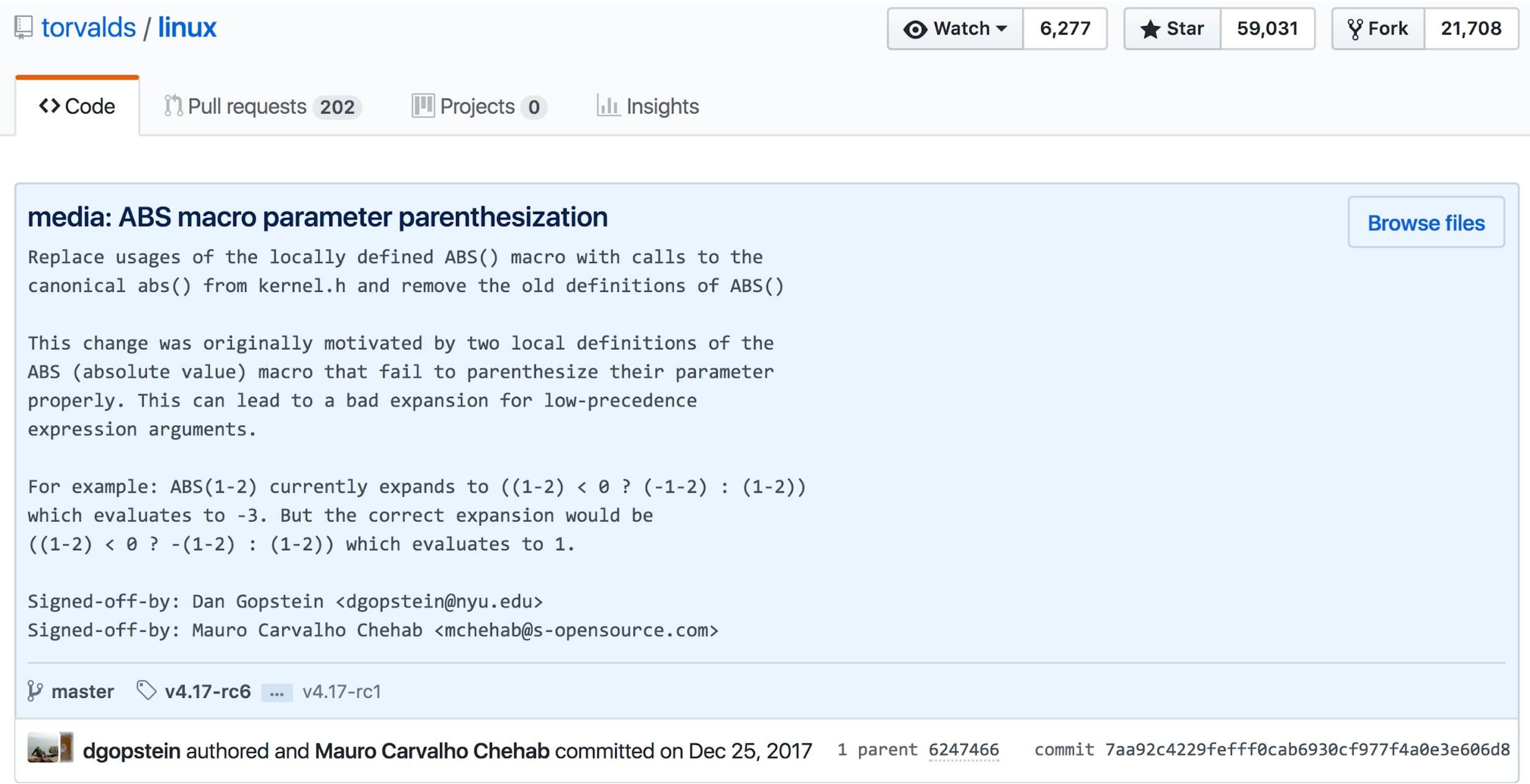
Buggy

```
#define ABS(x) ((x) < 0 ? (-x) : (x))
```

Macro Operator Precedence

https://github.com/torvalds/linux/commit/7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Buggy



media: ABS macro parameter parenthesization

```
Replace usages of the locally defined ABS() macro with calls to the
canonical abs() from kernel.h and remove the old definitions of ABS()

This change was originally motivated by two local definitions of the
ABS (absolute value) macro that fail to parenthesize their parameter
properly. This can lead to a bad expansion for low-precedence
expression arguments.

For example: ABS(1-2) currently expands to ((1-2) < 0 ? (-1-2) : (1-2))
which evaluates to -3. But the correct expansion would be
((1-2) < 0 ? -(1-2) : (1-2)) which evaluates to 1.

Signed-off-by: Dan Gopstein <dgopstein@nyu.edu>
Signed-off-by: Mauro Carvalho Chehab <mchehab@s-opensource.com>
```

master v4.17-rc6 ... v4.17-rc1

dgopstein authored and Mauro Carvalho Chehab committed on Dec 25, 2017 1 parent 6247466 commit 7aa92c4229fefff0cab6930cf977f4a0e3e606d8

Summary

Atoms of Confusion are ...

- **Confusing**
  - Atoms are statistically more confusing than other code in the lab
  - Atoms are 13% more likely to be commented than other code
- **Prevalent**
  - We found millions of examples in our corpus
  - 1 in ~23 lines of code has an atom
- **Buggy**
  - Bug-fix commits are 25% more likely remove atoms
  - We found and fixed a handful of bugs in Linux

Thank You

Prevalence of Confusing Code in Software Projects

Atoms of Confusion in the Wild

Dan Gopstein
NYU

Hongwei Henry Zhou, Phyllis Frankl, Justin Cappos

AtomsOfConfusion.com