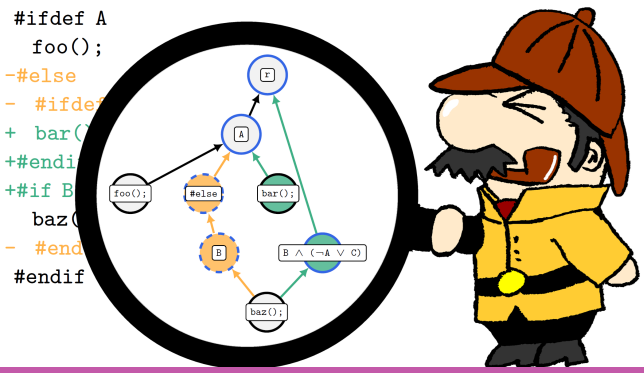


Variability-Aware Differencing with DiffDetective

Paul Bittner, Alexander Schultheiß, Benjamin Moosherr, Christof Tinnes, Sören Viegener, Timo Kehrer, Thomas Thüm | April 12, 2024

PADERBORN UNIVERSITY

FOSD 2023, Ulm

Pokémon images are trademarks of their respective owners, used here for educational purposes under fair use.

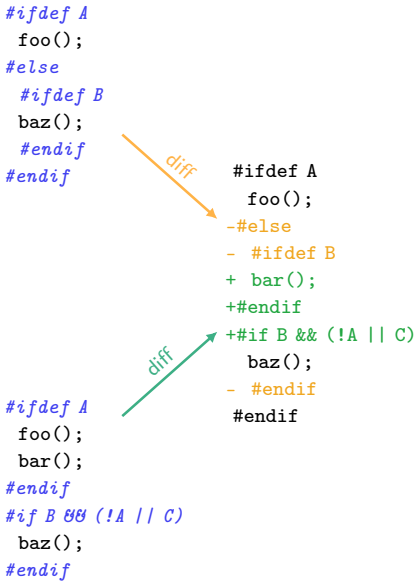
FOSD 2023, Ulm

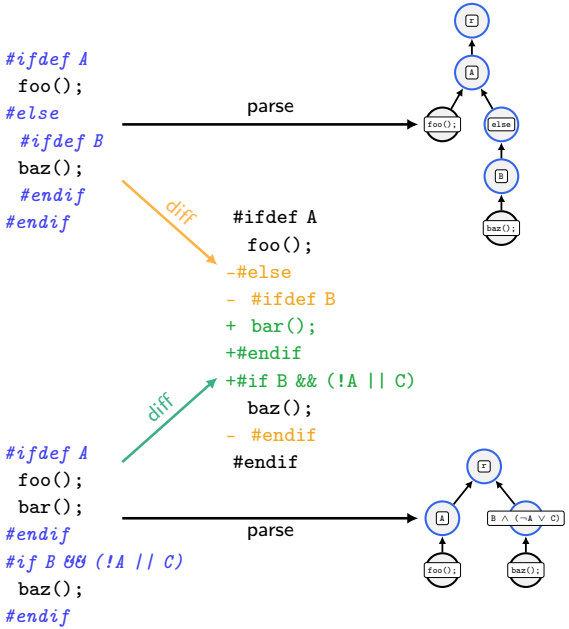
FOSD 2024, Eindhoven

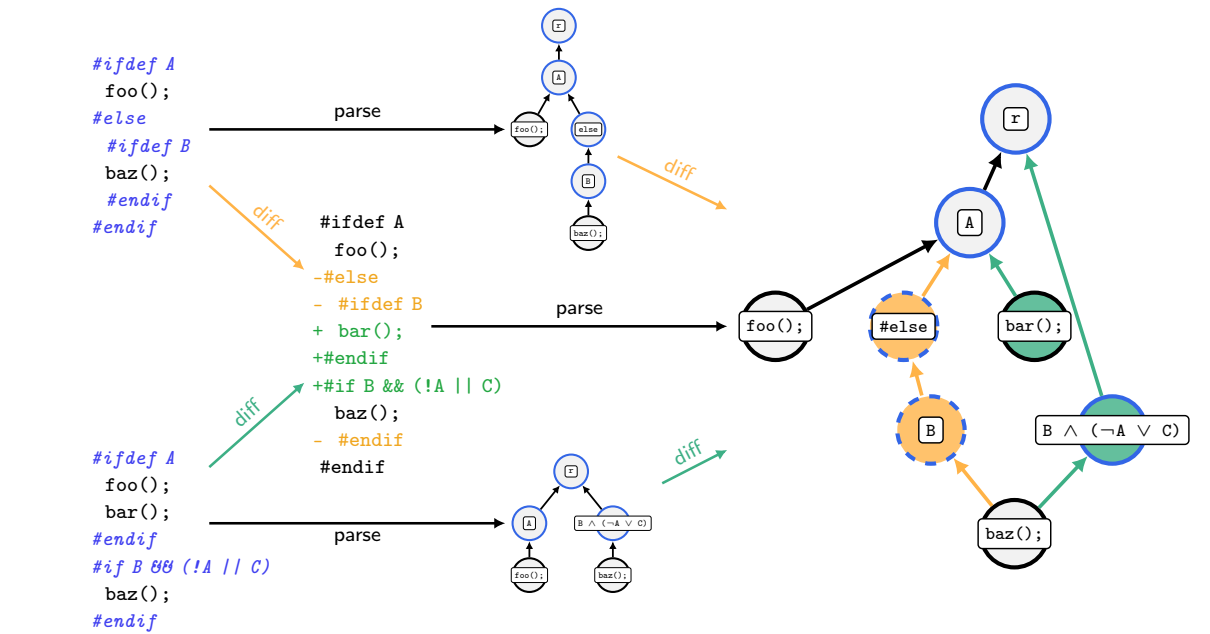
detectivepikachu.pokemon.com

```
#ifdef A
 foo();
#else
 #ifdef B
 baz();
 #endif
#endif
```

```
#ifdef A
 foo();
 bar();
#endif
#if B && (!A || C)
 baz();
#endif
```


#ifdef A
foo();
#else
#ifdef B
baz();
#endif
#endif
diff
#ifdef A
foo();
-#else
- #ifdef B
+ bar();
+#endif
+#if B && (!A || C)
baz();
- #endif
#endif
diff
#ifdef A
foo();
bar();
#endif
#if B && (!A || C)
baz();
#endif


#ifdef A
foo();
#else
#ifdef B
baz();
#endif
#endif
parse
true
A
foo();
else
B
baz();
diff
#ifdef A
foo();
-#else
- #ifdef B
+ bar();
+#endif
+#if B && (!A || C)
baz();
- #endif
#endif
diff
#ifdef A
foo();
bar();
#endif
#if B && (!A || C)
baz();
#endif
parse
true
A
B ∧ (¬A ∨ C)
foo();
baz();


#ifdef A
foo();
#else
#ifdef B
baz();
#endif
#endif
parse
r
A
foo();
else
B
baz();
diff
#ifdef A
foo();
-#else
- #ifdef B
+ bar();
+#endif
+#if B && (!A || C)
baz();
- #endif
#endif
parse
foo();
#else
bar();
A
B
B ∧ (¬A ∨ C)
baz();
r
diff
#ifdef A
foo();
bar();
#endif
#if B && (!A || C)
baz();
#endif
parse
r
A
B ∧ (¬A ∨ C)
foo();
baz();
diff

Why is this interesting?

Why is this interesting?

Operation Lifting [Bittner'23]

Lift operations on SPLs to operations on *edits* to SPLs (e.g., partial configuration, feature tracing), Potential use cases:

- feature-aware commit untangling
- patch backporting
- review merge requests
- ...

Why is this interesting?

Operation Lifting [Bittner'23]

Lift operations on SPLs to operations on *edits* to SPLs (e.g., partial configuration, feature tracing), Potential use cases:

- feature-aware commit untangling
- patch backporting
- review merge requests
- ...

Edit Explanations [Bittner'22, Güthing'24]

Does this line of code appear in more variants now?

Why is this interesting?

Operation Lifting [Bittner'23]

Lift operations on SPLs to operations on *edits* to SPLs (e.g., partial configuration, feature tracing), Potential use cases:

- feature-aware commit untangling
- patch backporting
- review merge requests
- ...

Simulate Clone-and-Own [Schult.'22,'23]

Generate evolution histories of single variants. (see talk of Alex)

Edit Explanations [Bittner'22, Güthing'24]

Does this line of code appear in more variants now?

Why is this interesting?

Operation Lifting [Bittner'23]

Lift operations on SPLs to operations on *edits* to SPLs (e.g., partial configuration, feature tracing), Potential use cases:

- feature-aware commit untangling
- patch backporting
- review merge requests
- ...

Simulate Clone-and-Own [Schult.'22,'23]

Generate evolution histories of single variants. (see talk of Alex)

Edit Explanations [Bittner'22, Güthing'24]

Does this line of code appear in more variants now?

Benchmarking Differencers [Moosherr'23]

- line-based differencing (Myers/Histogram)
- vs. GumTree [Falleri'14]
- (vs. TrueDiff [Erdweg'21])

Why is this interesting?

Operation Lifting [Bittner'23]

Lift operations on SPLs to operations on *edits* to SPLs (e.g., partial configuration, feature tracing), Potential use cases:

- feature-aware commit untangling
- patch backporting
- review merge requests
- ...

Simulate Clone-and-Own [Schult.'22,'23]

Generate evolution histories of single variants. (see talk of Alex)

Edit Explanations [Bittner'22, Güthing'24]

Does this line of code appear in more variants now?

Benchmarking Differencers [Moosherr'23]

- line-based differencing (Myers/Histogram)
- vs. GumTree [Falleri'14]
- (vs. TrueDiff [Erdweg'21])

Higher-Order Tree Differencing [TODO]

Diffs of Diffs of Diffs ...

DiffDetective

https://variantsync.github.io/DiffDetective

DiffDetective Demo

https://github.com/VariantSync/DiffDetective-Demo

References

Bittner'22 *Classifying Edits to Variability in Source Code, ESEC/FSE'22*

Bittner'23 *Views on Edits to Variational Software, SPLC'23*

Güthing'24 *Explaining Edits to Variability Annotations in Evolving Software Product Lines, VaMoS'24*

Schult.'22 *Simulating the Evolution of Clone-and-Own Projects with VEVOS, EASE'22*

Schult.'23 *Benchmark Generation with VEVOS: A Coverage Analysis of Evolution Scenarios in Variant-Rich Systems, VaMoS'23*

Moosherr'23 *Constructing Variation Diffs Using Tree Diffing Algorithms, Bachelor's Thesis, Benjamin Moosherr, Ulm'23*