Tailoring Programs for Static Analysis via Program Transformation

Rijnard van Tonder and Claire Le Goues

rvt@cs.cmu.edu clegoues@cs.cmu.edu

Carnegie Mellon University

institute for SOFTWARE RESEARCH

spotbugs / **spotbugs**

facebook / **infer**

uber / **NullAway**

Static analyzers approximate

spotbugs / **spotbugs**

facebook / **infer**

uber / **NullAway**

Static analyzers approximate

Theoretically

- Decidability
- Soundness

spotbugs / spotbugs

facebook / infer

uber / NullAway

Static analyzers approximate

Theoretically

- Decidability
- Soundness

Practicality

- Incomplete language support or models

spotbugs / spotbugs

facebook / infer

uber / NullAway

Static analyzers approximate

Theoretically

- Decidability
- Soundness

Practicality

- Incomplete language support or models

Specify `fclose` closes a file

Developers change their code to work around analyzer limitations

Developers change their code to work around analyzer limitations

rsyslog

```
+ // clang static analyzer work-around
+ const char *const const_END = “END”;
+ if (strcmp(rectype, const_END))
-  if (strcmp(rectype, “END”))
```

Developers change their code to work around analyzer limitations

rsyslog

```
+ // clang static analyzer work-around
+ const char *const const_END = “END”;
+ if (strcmp(rectype, const_END))
-  if (strcmp(rectype, “END”))
```

Pull out constant

Developers change their code to work around analyzer limitations

rsyslog

```
+ // clang static analyzer work-around
+ const char *const const_END = “END”;
+ if (strcmp(rectype, const_END))
-  if (strcmp(rectype, “END”))
```

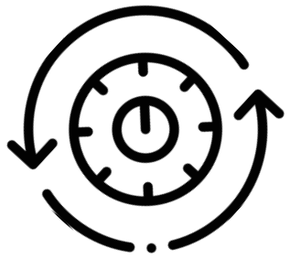
Prevents macro expansion

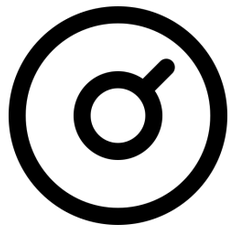
Pull out constant

→ No False Positive

Why fix an analyzer issue in this way?

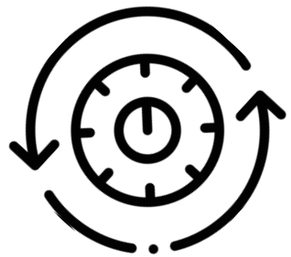
Why fix an analyzer issue in this way?

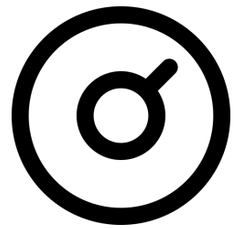




Icons made by Smashicon and Freepik from www.flaticon.com

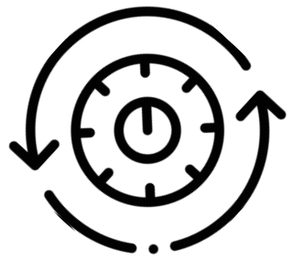
Why fix an analyzer issue in this way?

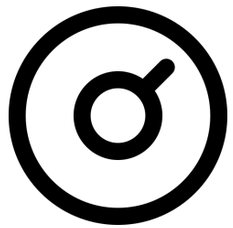




Analysis knobs are insufficient

Why fix an analyzer issue in this way?





Analysis knobs are insufficient

- Command line flags
- Supress with code comments

Why fix an analyzer issue in this way?

Analysis authors implement these

Analysis knobs are insufficient

- Command line flags
- Supress with code comments

Icons made by Smashicon and Freepik from www.flaticon.com

Why fix an analyzer issue in this way?

rsyslog

```
+ // clang static analyzer work-around
+ const char *const const_END = “END”;
+ if (strcmp(rectype, const_END))
-  if (strcmp(rectype, “END”))
```

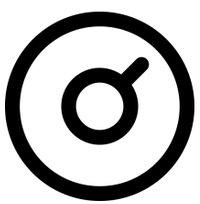
Why fix an analyzer issue in this way?

rsyslog

```
+ // clang static analyzer work-around
+ const char *const const_END = “END”;
+ if (strcmp(rectype, const_END))
-  if (strcmp(rectype, “END”))
```

Analysis *users* can modify programs to change analyzer behavior

Why fix an analyzer issue in this way?



rsyslog

```
+ // clang static analyzer work-around
+ const char *const const_END = “END”;
+ if (strcmp(rectype, const_END))
-  if (strcmp(rectype, “END”))
```

Analysis *users* can modify programs to change analyzer behavior

Our approach: Tailor analysis via program transformation

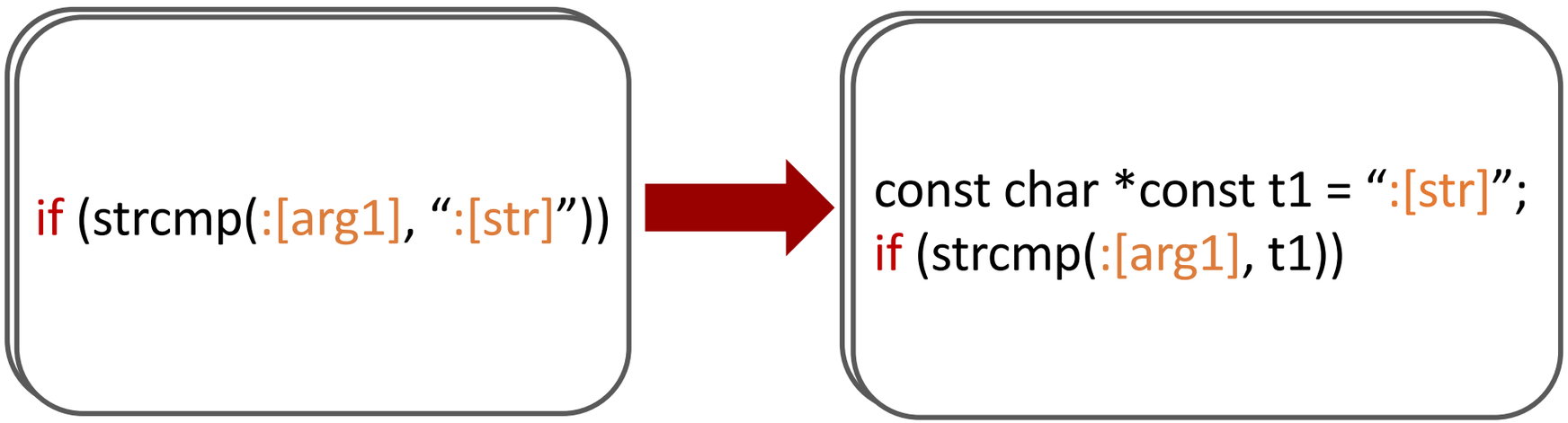
- Syntax rewriting with declarative templates

Our approach: Tailor analysis via program transformation

```
+ // clang static analyzer work-around
+ const char *const const_END = “END”;
+ if (strcmp(rectype, const_END))
-  if (strcmp(rectype, “END”))
```

Our approach: Tailor analysis via program transformation

```
+ // clang static analyzer work-around
+ const char *const const_END = “END”;
+ if (strcmp(rectype, const_END))
-  if (strcmp(rectype, “END”))
```



Our approach: Tailor analysis via program transformation

```
+ // clang static analyzer work-around
+ const char *const const_END = “END”;
+ if (strcmp(rectype, const_END))
-  if (strcmp(rectype, “END”))
```

Tooling:

https://comby.dev 

```
if (strcmp(:[arg1], “:[str]”))
```

```
const char *const t1 = “:[str]”;
if (strcmp(:[arg1], t1))
```

Our approach: Tailor analysis via program transformation

- Lightweight
- Many languages
- Fast

Tooling:

https://comby.dev 

```
if (strcmp(:[arg1], ":[str]"))
```

```
const char *const t1 = ":[str]";
if (strcmp(:[arg1], t1))
```

Our approach: Tailor analysis via program transformation

- Syntax rewriting with declarative templates

Our approach: Tailor analysis via program transformation

- Syntax rewriting with declarative templates

- Temporary modification

Evaluation: Showing that tailoring programs improve static analysis

Evaluation: Showing that tailoring programs improve static analysis

- Five analyzers, three languages

  PHP, Java, C

  PHPStan, Infer, SpotBugs, Clang Static Analyzer, CodeSonar

Evaluation: Showing that tailoring programs improve static analysis

- Five analyzers, three languages

  PHP, Java, C
  PHPStan, Infer, SpotBugs, Clang Static Analyzer, CodeSonar

- Longstanding open false positive issues on GitHub

Evaluation: Showing that tailoring programs improve static analysis

- Five analyzers, three languages

  PHP, Java, C
  PHPStan, Infer, SpotBugs, Clang Static Analyzer, CodeSonar

- Longstanding open false positive issues on GitHub
  - Developer-reported issues
  - Commit search

Results

Analyzer	Lang	Proj	KLOC	Anlyz	Rewr	Δ FP	# R
PHPStan	PHP	WordPress	5.7	2.5s	0.3s	-44	16
PHPStan	PHP	Codesniffer	1.6	1.1s	0.1s	-3	3
Infer	Java	Drift	50.7	2m47s	1.0s	-1	1
Infer	Java	Presto	813.0	39m20s	3.3s	-1	1
Infer	C	OpenSSL	402.9	17m19s	1.6s	-6	3,735
Clang SA	C	rsyslog	145.0	16m20s	2.9s	-3	10
CodeSonar	C	swoole-src	96.8	6m40s	0.6s	-2	2
CodeSonar	C	Ioping	1.2	14s	0.2s	-2	2
SpotBugs	Java	hazelcast-jet	103.8	1m48s	1.3s	-1	45
SpotBugs	Java	Santulator	11.1	44s	0.5s	-2	8
⋮						⋮	

9 templates remove 111 false positives across 15 projects

Results

Analyzer	Lang	Proj	KLOC	Anlyz	Rewr	Δ FP	# R
PHPStan	PHP	WordPress	5.7	2.5s	0.3s	-44	16
PHPStan	PHP	Codesniffer	1.6	1.1s	0.1s	-3	3
Infer	Java	Drift	50.7	2m47s	1.0s	-1	1
Infer	Java	Presto	813.0	39m20s	3.3s	-1	1
Infer	C	OpenSSL	402.9	17m19s	1.6s	-6	3,735
Clang SA	C	rsyslog	145.0	16m20s	2.9s	-3	10
CodeSonar	C	swoole-src	96.8	6m40s	0.6s	-2	2
CodeSonar	C	Ioping	1.2	14s	0.2s	-2	2
SpotBugs	Java	hazelcast-jet	103.8	1m48s	1.3s	-1	45
SpotBugs	Java	Santulator	11.1	44s	0.5s	-2	8

Large Projects

Results

Analyzer	Lang	Proj	KLOC	Anlyz	Rewr	Δ FP	# R
PHPStan	PHP	WordPress	5.7	2.5s	0.3s	-44	16
PHPStan	PHP	Codesniffer	1.6	1.1s	0.1s	-3	3
Infer	Java	Drift	50.7	2m47s	1.0s	-1	1
Infer	Java	Presto	813.0	39m20s	3.3s	-1	1
Infer	C	OpenSSL	402.9	17m19s	1.6s	-6	3,735
Clang SA	C	rsyslog	145.0	16m20s	2.9s	-3	10
CodeSonar	C	swoole-src	96.8	6m40s	0.6s	-2	2
CodeSonar	C	Ioping	1.2	14s	0.2s	-2	2
SpotBugs	Java	hazelcast-jet	103.8	1m48s	1.3s	-1	45
SpotBugs	Java	Santulator	11.1	44s	0.5s	-2	8

Efficient

Results

Analyzer	Lang	Proj	KLOC	Anlyz	Rewr	Δ FP	# R
PHPStan	PHP	WordPress	5.7	2.5s	0.3s	-44	16
PHPStan	PHP	Codesniffer	1.6	1.1s	0.1s	-3	3
Infer	Java	Drift	50.7	2m47s	1.0s	-1	1
Infer	Java	Presto	813.0	39m20s	3.3s	-1	1
Infer	C	OpenSSL	402.9	17m19s	1.6s	-6	3,735
Clang SA	C	rsyslog	145.0	16m20s	2	-3	10
CodeSonar	C	swoole-src	96.8	6m40s		-2	2
CodeSonar	C	Ioping	1.2	14s		-2	2
SpotBugs	Java	hazelcast-jet	103.8	1m		-1	45
SpotBugs	Java	Santulator	11.1			-2	8

False positive due to socket not closed by function

Results

Analyzer	Lang	Proj	KLOC	Anlyz	Rewr	Δ FP	# R
PHPStan	PHP	WordPress	5.7	2.5s	0.3s	-44	16
PHPStan	PHP	Codesniffer	1.6	1.1s	0.1s	-3	3
Infer	Java	Drift	50.7	2m47s	1.0s	-1	1
Infer	Java	Presto	813.0	39m20s	3.3s	-1	1
Infer	C	OpenSSL	402.9	17m19s	1.6s	6	3,735
Clang SA	C	rsyslog	145.0	16m20s	2.9s	-3	10
CodeSonar	C	swoole-src	96.8	6m40s		-2	2
CodeSonar	C	Ioping	1.2	14s		-2	2
SpotBugs	Java	hazelcast-jet	103.8	1m4		-1	45
SpotBugs	Java	Santulator	11.1			-2	8

Pattern reused and found in another project

Results

Analyzer	Lang	Proj	KLOC	Anlyz	Rewr	Δ FP	# R
PHPStan	PHP	WordPress	5.7	2.5s	0.3s	-44	16
PHPStan	PHP	Codesniffer	1.6	1.1s	0.1s	-3	3
Infer	Java	Drift	50.7	2m47s	1.0s	-1	1
Infer	Java	Presto	813.0	39m20s	3.3s	-1	1
Infer	C	OpenSSL	402.9	17m19s	1.6s	-6	3,735
Clang SA	C	rsyslog	145.0	16m20s	2.9s	-3	10
CodeSonar	C	swoole-src	96.8	6m40s	0.6s	-2	2
CodeSonar	C	Ioping	1.2	14s	0.2s	-2	2

Local, targeted transformations can improve analysis without adverse effects

Static analyzers approximate

Theoretically
- Decidability
- Soundness

Practicality
- Incomplete language support or models

SpotBugs — spotbugs / spotbugs

facebook / infer

UBER — uber / NullAway

Developers change their code to work around analyzer limitations

rsyslog

```
+ // clang static analyzer work-around
+ const char *const const_END = “END”;
+ if (strcmp(rectype, const_END))
-  if (strcmp(rectype, “END”))
```

Prevents macro expansion

Pull out constant

→ No False Positive

Our approach: tailor analysis via program transformation

```
+ // clang static analyzer work-around
+ const char *const const_END = “END”;
+ if (strcmp(rectype, const_END))
-  if (strcmp(rectype, “END”))
```

Tooling:

https://comby.dev

```
if (strcmp(:[arg1], “:[str]”))
```

→

```
const char *const t1 = “:[str]”;
if (strcmp(:[arg1], t1))
```

Analyzer	Lang	Proj	KLOC	Anlyz	Rewr	Δ FP	# R
PHPStan	PHP	WordPress	5.7	2.5s	0.3s	-44	16
PHPStan	PHP	Codesniffer	1.6	1.1s	0.1s	-3	3
Infer	Java	Drift	50.7	2m47s	1.0s	-1	1
Infer	Java	Presto	813.0	39m20s	3.3s	-1	1
Infer	C	OpenSSL	402.9	17m19s	1.6s	-25	3,583
Clang SA	C	rsyslog	145.0	16m20s	2.9s	-3	10
CodeSonar	C	swoole-src	96.8	6m40s	0.6s	-2	2

Local, targeted transformations can improve analysis without adverse effects

@rvtond @clegoues