Improving the Quality of Static Analysis Bug Reports

Kristóf Umann <szelethus@inf.elte.hu>
Zoltán Porkoláb <gsd@inf.elte.hu>

Eötvös Loránd University, Faculty of Informatics, Department of Compilers

Bolyai Collage, 2024. March. 20., Budapest, Hungary

Static Analysis is hard…

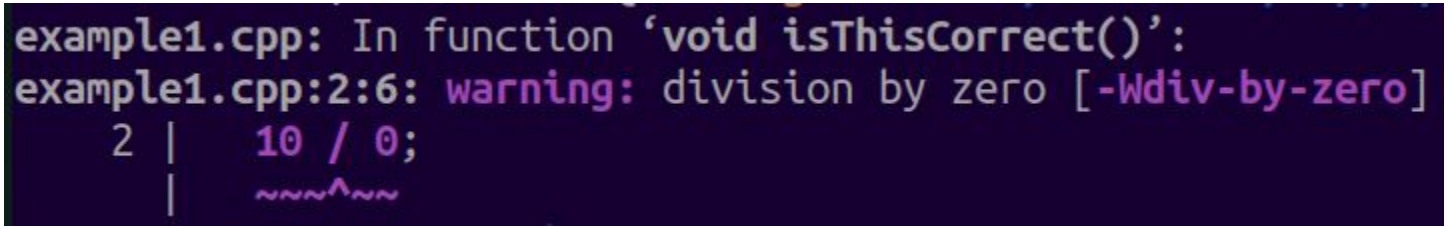
- Static analysis is a great tool for early feedback on software quality
- Detecting complex, deep-rooted problems requires complex analysis

```
void isThisCorrect() {
  *NULL = 5;
  10 / 0;
  int *notMyProblem = malloc(4);
}
```

```
int handle(struct A *a) {
  struct Obj o = conjure(a);
  int i = extract(o);
  return 10 / i;
}
```

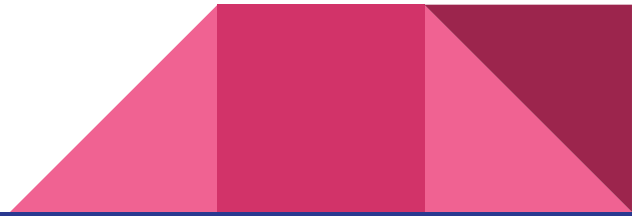
…great bug report generation is harder!

- Simple analyses require simple bug reporting techniques



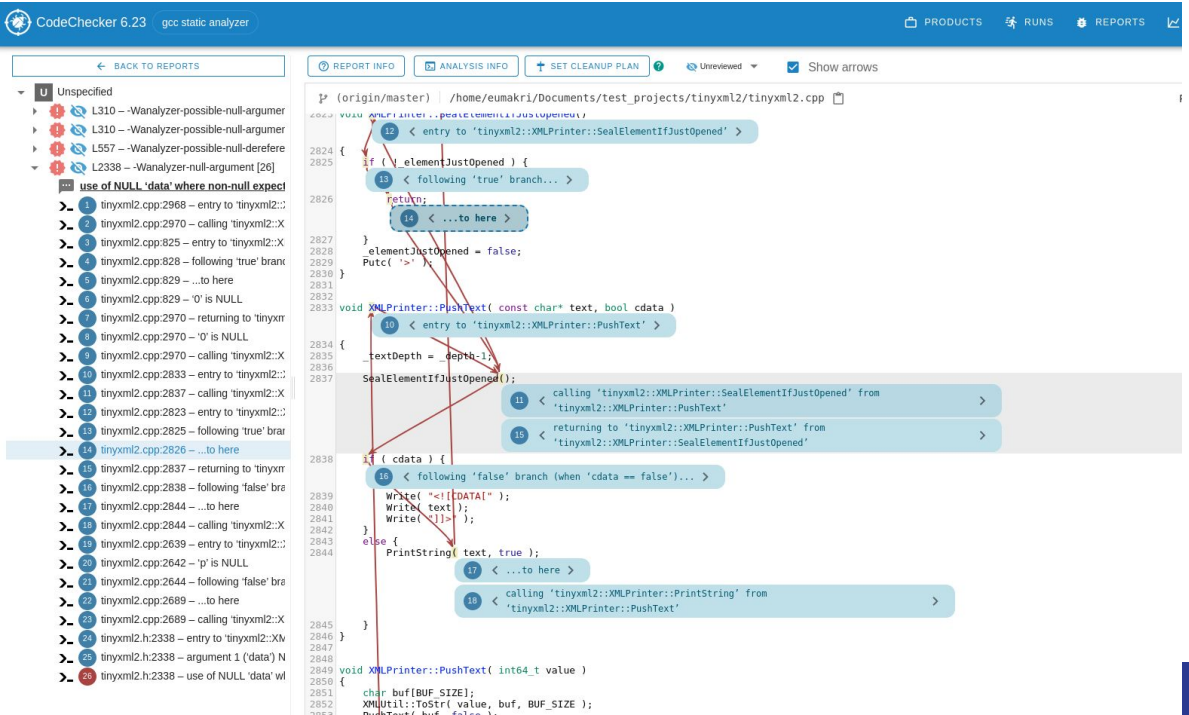
…great bug report generation is harder!

- Complex analyses require complex bug reporting techniques



…great bug report generation is harder!

- Complex analyses require complex bug reporting techniques



BACK TO REPORTS

REPORT INFO ANALYSIS INFO SET CLEANUP PLAN Unreviewed Show arrows BLAME VIEW COMMENTS (0)

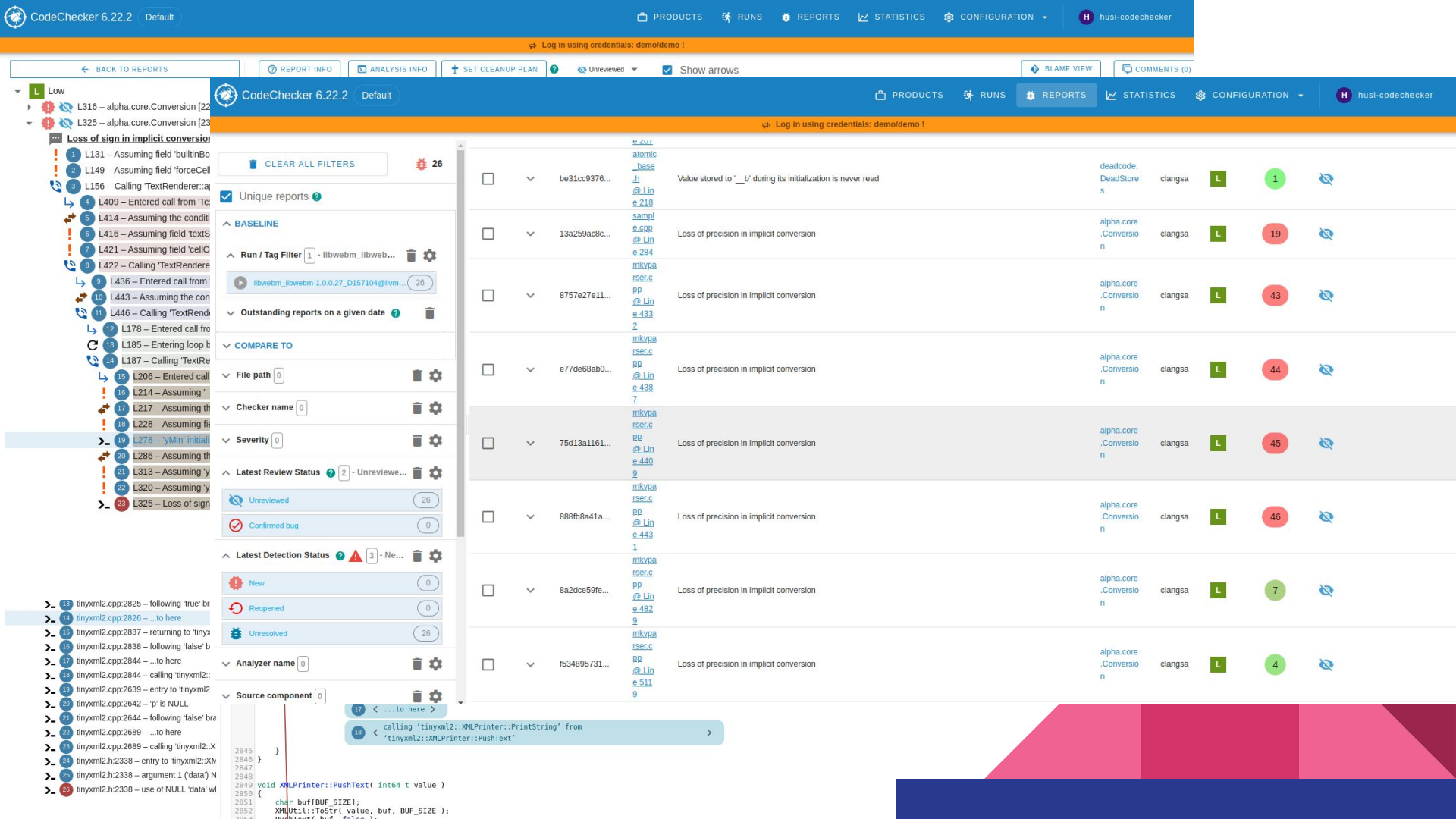
Found in: contour_v0.2.0.173_D157104@llvm/main_new:TextRenderer.cpp:L325 23

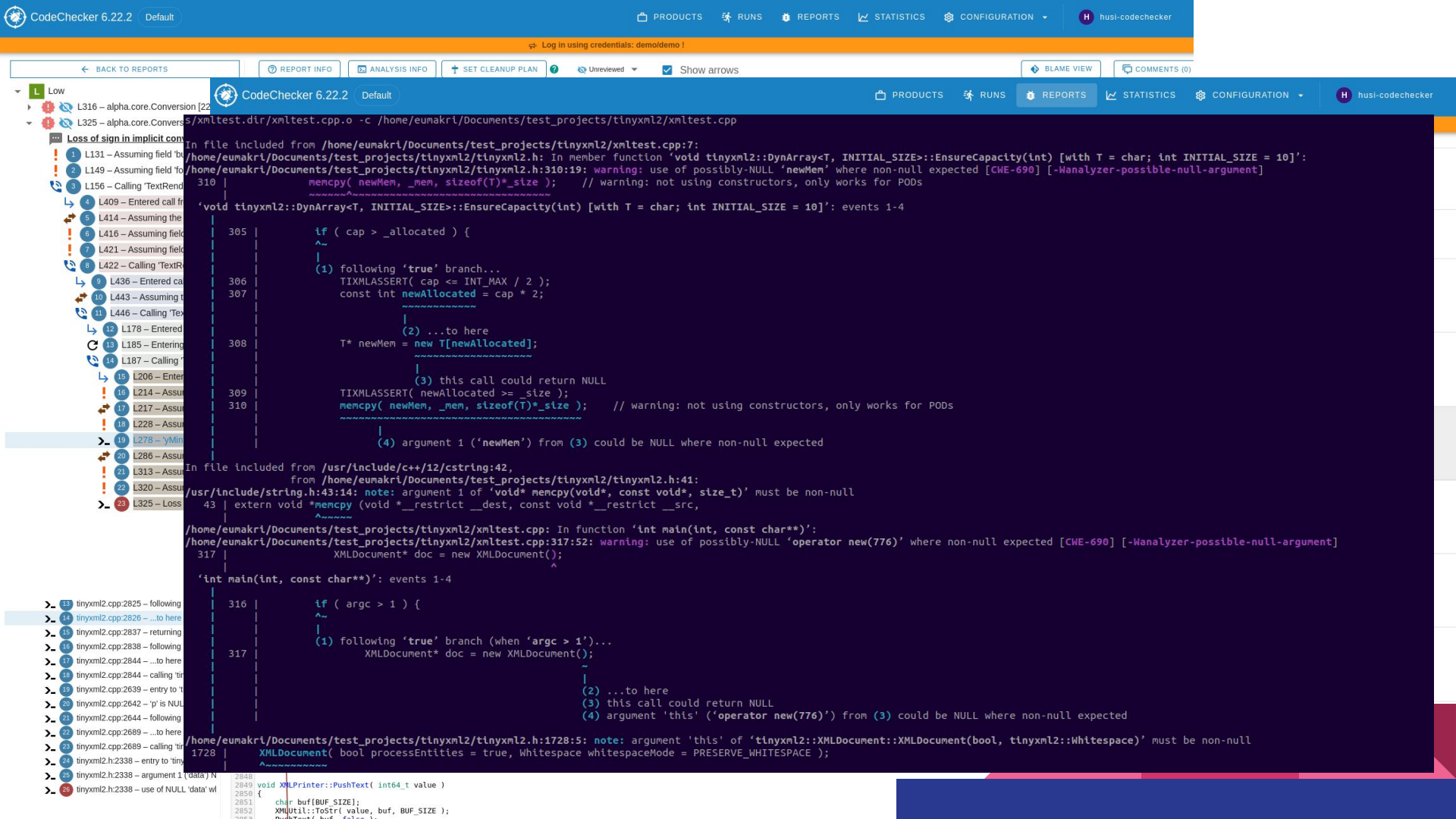
- Low
  - L316 – alpha.core.Conversion [22]
  - L325 – alpha.core.Conversion [23]
    - Loss of sign in implicit conversion
      - 1 L131 – Assuming field 'builtinBoxDrawir
      - 2 L149 – Assuming field 'forceCellGroupS
      - 3 L156 – Calling 'TextRenderer::appendC
        - 4 L409 – Entered call from 'TextRende
        - 5 L414 – Assuming the condition is fal
        - 6 L416 – Assuming field 'textStartFou
        - 7 L421 – Assuming field 'cellCount_' is
        - 8 L422 – Calling 'TextRenderer::endS
          - 9 L436 – Entered call from 'TextRe
          - 10 L443 – Assuming the condition is
          - 11 L446 – Calling 'TextRenderer::rer
            - 12 L178 – Entered call from 'Text
            - 13 L185 – Entering loop body
            - 14 L187 – Calling 'TextRenderer::
              - 15 L206 – Entered call from 'T
              - 16 L214 – Assuming '_present
              - 17 L217 – Assuming the condi
              - 18 L228 – Assuming field 'form
              - 19 L278 – 'yMin' initialized her
              - 20 L286 – Assuming the condi
              - 21 L313 – Assuming 'yOverflo
              - 22 L320 – Assuming 'yMin' is <
              - 23 L325 – Loss of sign in impli

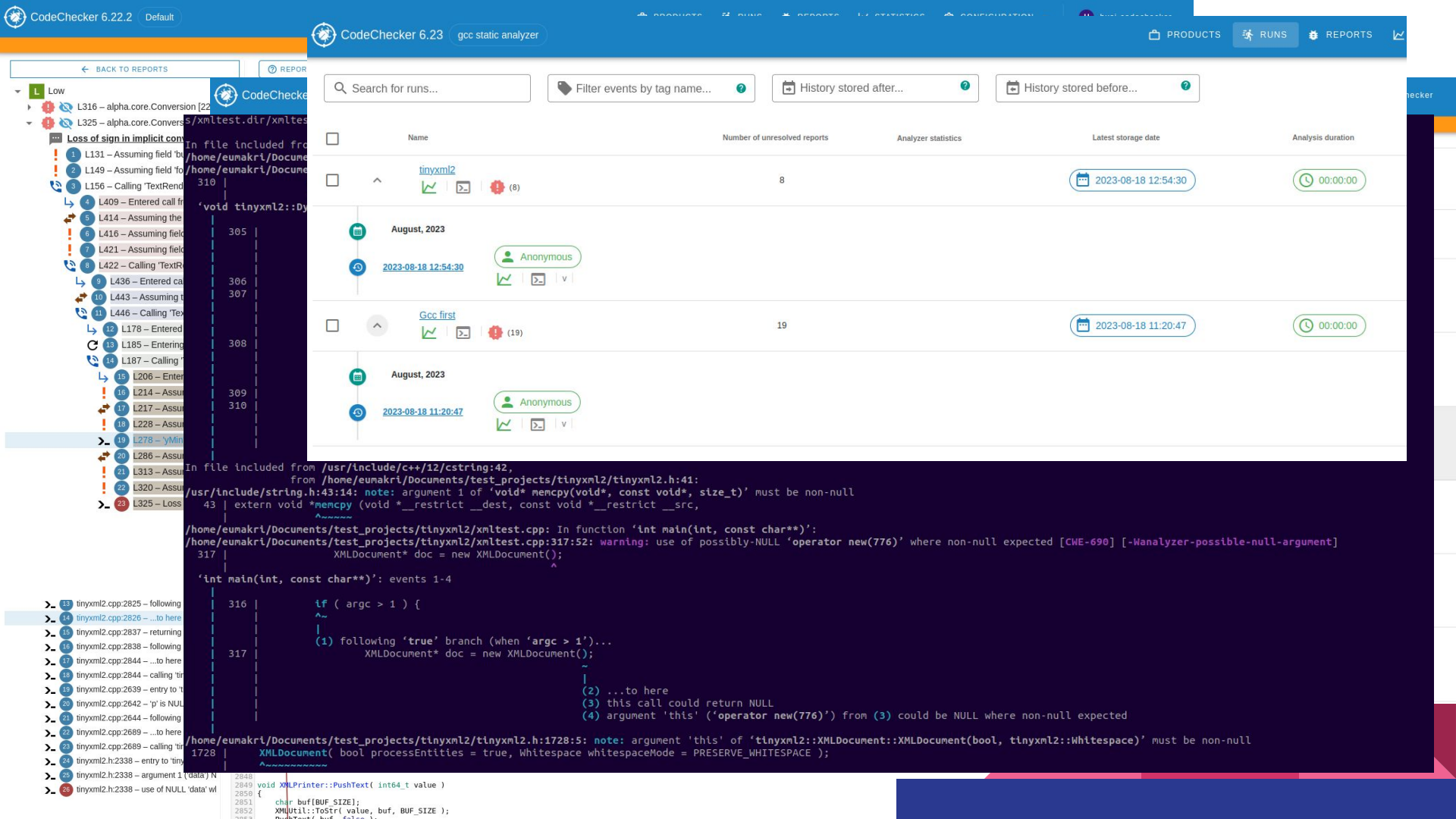
```
271             // if (rightEdge != std::numeric_limits<int>::max())
272             //     debuglog(TextRendererTag).write("right edge found. {} < {}.", rightEdge+1, glyph.bitmap.width);
273         }
274     }
275     // }}}
276
277     auto const yMax = gridMetrics_.baseline + glyph.position.y;
278     auto const yMin = yMax - glyph.size.height.as<int>();
        (19) < 'yMin' initialized here >
279
280     auto const ratio = !colored
281                      ? 1.0f
282                      : max(float(gridMetrics_.cellSize.width.as<int>() * numCells) / float(glyph.size.width.as<int>()),
283                            float(gridMetrics_.cellSize.height.as<int>()) / float(glyph.size.height.as<int>()));
284
285     auto const yOverflow = gridMetrics_.cellSize.height.as<int>() - yMax;
286     if (crispy::debugtag::enabled(TextRendererTag))
            (20) < Assuming the condition is false >
287         debuglog(TextRendererTag).write("insert glyph {}: {}; ratio:{}; yOverflow({}, {}); {}",
288                                         _id.index,
289                                         colored ? "emoji" : "text",
290                                         ratio,
291                                         yOverflow < 0 ? yOverflow : 0,
292                                         yMin < 0 ? yMin : 0,
293                                         glyph);
294
295     auto && [userFormat, targetAtlas] = [this](bool _colorFont, text::bitmap_format _glyphFormat) -> pair<int, TextureAtlas&> { // {{{
296         // this format ID is used by the fragment shader to select the right texture atlas
297         if (_colorFont)
298             return {1, *colorAtlas_};
299         switch (_glyphFormat)
300         {
301             case text::bitmap_format::rgba:
302                 return {1, *colorAtlas_};
303             case text::bitmap_format::rgb:
304                 return {2, *lcdAtlas_};
305             case text::bitmap_format::alpha_mask:
306                 return {0, *monochromeAtlas_};
307         }
308         return {0, *monochromeAtlas_};
309     }(colored, glyph.format); // }}}
310
311     glyphToTextureMapping_[_id] = glyph.format;
312
313     if (yOverflow < 0)
            (21) < Assuming 'yOverflow' is >= 0 >
314     {
315         debuglog(TextRendererTag).write("Cropping {} overflowing bitmap rows.", -yOverflow);
```

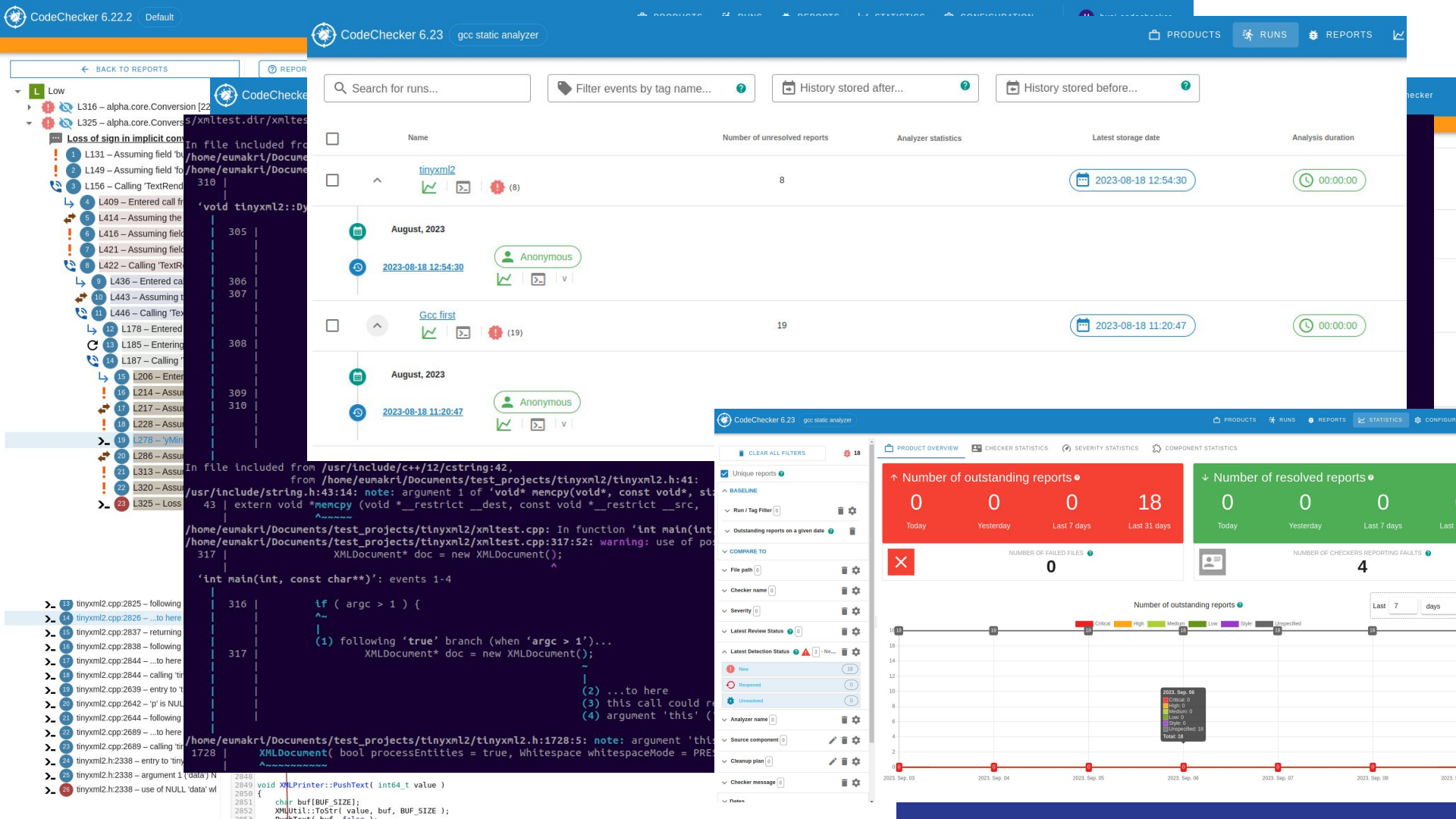
- 13 tinyxml2.cpp:2825 – following 'true' bran
- 14 tinyxml2.cpp:2826 – ...to here
- 15 tinyxml2.cpp:2837 – returning to 'tinyxm
- 16 tinyxml2.cpp:2838 – following 'false' bra
- 17 tinyxml2.cpp:2844 – ...to here
- 18 tinyxml2.cpp:2844 – calling 'tinyxml2::X
- 19 tinyxml2.cpp:2639 – entry to 'tinyxml2::;
- 20 tinyxml2.cpp:2642 – 'p' is NULL
- 21 tinyxml2.cpp:2644 – following 'false' bra
- 22 tinyxml2.cpp:2689 – ...to here
- 23 tinyxml2.cpp:2689 – calling 'tinyxml2::X
- 24 tinyxml2.h:2338 – entry to 'tinyxml2::XM
- 25 tinyxml2.h:2338 – argument 1 ('data') N
- 26 tinyxml2.h:2338 – use of NULL 'data' wl

```
        (15) < 'tinyxml2::XMLPrinter::SealElementIfJustOpened' >
2838    if ( cdata ) {
            (16) < following 'false' branch (when 'cdata == false')... >
2839        Write( "<![CDATA[" );
2840        Write( text );
2841        Write( "]]>" );
2842    }
2843    else {
2844        PrintString( text, true );
                (17) < ...to here >
                (18) < calling 'tinyxml2::XMLPrinter::PrintString' from 'tinyxml2::XMLPrinter::PushText' >
2845    }
2846}
2847
2848
2849void XMLPrinter::PushText( int64_t value )
2850{
2851    char buf[BUF_SIZE];
2852    XMLUtil::ToStr( value, buf, BUF_SIZE );
```


CodeChecker 6.22.2
Default
PRODUCTS
RUNS
REPORTS
STATISTICS
CONFIGURATION
husi-codechecker
Log in using credentials: demo/demo !
BACK TO REPORTS
REPORT INFO
ANALYSIS INFO
SET CLEANUP PLAN
Unreviewed
Show arrows
BLAME VIEW
COMMENTS (0)
Low
L316 – alpha.core.Conversion [22
L325 – alpha.core.Conversion [23
Loss of sign in implicit conversion
L131 – Assuming field 'builtinBo
L149 – Assuming field 'forceCell
L156 – Calling 'TextRenderer::a
L409 – Entered call from 'Te
L414 – Assuming the conditi
L416 – Assuming field 'textS
L421 – Assuming field 'cellC
L422 – Calling 'TextRendere
L436 – Entered call from '
L443 – Assuming the con
L446 – Calling 'TextRend
L178 – Entered call fro
L185 – Entering loop b
L187 – Calling 'TextRe
L206 – Entered call
L214 – Assuming '_
L217 – Assuming th
L228 – Assuming fi
L278 – 'yMin' initiali
L286 – Assuming th
L313 – Assuming 'y
L320 – Assuming 'y
L325 – Loss of sign
tinyxml2.cpp:2825 – following 'true' br
tinyxml2.cpp:2826 – ...to here
tinyxml2.cpp:2837 – returning to 'tinyx
tinyxml2.cpp:2838 – following 'false' b
tinyxml2.cpp:2844 – ...to here
tinyxml2.cpp:2844 – calling 'tinyxml2::
tinyxml2.cpp:2639 – entry to 'tinyxml2
tinyxml2.cpp:2642 – 'p' is NULL
tinyxml2.cpp:2644 – following 'false' bra
tinyxml2.cpp:2689 – ...to here
tinyxml2.cpp:2689 – calling 'tinyxml2::X
tinyxml2.h:2338 – entry to 'tinyxml2::XM
tinyxml2.h:2338 – argument 1 ('data') N
tinyxml2.h:2338 – use of NULL 'data' wl
CLEAR ALL FILTERS
26
Unique reports
BASELINE
Run / Tag Filter 1 - libwebm_libweb...
libwebm_libwebm-1.0.0.27_D157104@llvm...
26
Outstanding reports on a given date
COMPARE TO
File path 0
Checker name 0
Severity 0
Latest Review Status 2 - Unreviewe...
Unreviewed
26
Confirmed bug
0
Latest Detection Status 3 - Ne...
New
0
Reopened
0
Unresolved
26
Analyzer name 0
Source component 0
be31cc9376...
atomic_base.h @ Line 218
Value stored to '__b' during its initialization is never read
deadcode.DeadStores
clangsa
L
1
13a259ac8c...
sample.cpp @ Line 284
Loss of precision in implicit conversion
alpha.core.Conversion
clangsa
L
19
8757e27e11...
mkvparser.cpp @ Line 4332
Loss of precision in implicit conversion
alpha.core.Conversion
clangsa
L
43
e77de68ab0...
mkvparser.cpp @ Line 4387
Loss of precision in implicit conversion
alpha.core.Conversion
clangsa
L
44
75d13a1161...
mkvparser.cpp @ Line 4409
Loss of precision in implicit conversion
alpha.core.Conversion
clangsa
L
45
888fb8a41a...
mkvparser.cpp @ Line 4431
Loss of precision in implicit conversion
alpha.core.Conversion
clangsa
L
46
8a2dce59fe...
mkvparser.cpp @ Line 4829
Loss of precision in implicit conversion
alpha.core.Conversion
clangsa
L
7
f534895731...
mkvparser.cpp @ Line 5119
Loss of precision in implicit conversion
alpha.core.Conversion
clangsa
L
4
17 ...to here
18 calling 'tinyxml2::XMLPrinter::PrintString' from 'tinyxml2::XMLPrinter::PushText'
2845 }
2846 }
2847
2848
2849 void XMLPrinter::PushText(int64_t value)
2850 {
2851 char buf[BUF_SIZE];
2852 XMLUtil::ToStr(value, buf, BUF_SIZE);


CodeChecker 6.22.2
Default
PRODUCTS
RUNS
REPORTS
STATISTICS
CONFIGURATION
husi-codechecker
Log in using credentials: demo/demo !
BACK TO REPORTS
REPORT INFO
ANALYSIS INFO
SET CLEANUP PLAN
Unreviewed
Show arrows
BLAME VIEW
COMMENTS (0)
Low
L316 – alpha.core.Conversion [22
L325 – alpha.core.Conversi
Loss of sign in implicit con
L131 – Assuming field 'bi
L149 – Assuming field 'fo
L156 – Calling 'TextRend
L409 – Entered call fr
L414 – Assuming the
L416 – Assuming field
L421 – Assuming field
L422 – Calling 'TextR
L436 – Entered ca
L443 – Assuming t
L446 – Calling 'Tex
L178 – Entered
L185 – Entering
L187 – Calling '
L206 – Enter
L214 – Assu
L217 – Assu
L228 – Assu
L278 – 'yMin
L286 – Assu
L313 – Assu
L320 – Assu
L325 – Loss
tinyxml2.cpp:2825 – following
tinyxml2.cpp:2826 – ...to here
tinyxml2.cpp:2837 – returning
tinyxml2.cpp:2838 – following
tinyxml2.cpp:2844 – ...to here
tinyxml2.cpp:2844 – calling 'tir
tinyxml2.cpp:2639 – entry to 't
tinyxml2.cpp:2642 – 'p' is NUL
tinyxml2.cpp:2644 – following
tinyxml2.cpp:2689 – ...to here
tinyxml2.cpp:2689 – calling 'tir
tinyxml2.h:2338 – entry to 'tiny
tinyxml2.h:2338 – argument 1 ('data') N
tinyxml2.h:2338 – use of NULL 'data' wl
CodeChecker 6.22.2
Default
PRODUCTS
RUNS
REPORTS
STATISTICS
CONFIGURATION
husi-codechecker
s/xmltest.dir/xmltest.cpp.o -c /home/eumakri/Documents/test_projects/tinyxml2/xmltest.cpp
In file included from /home/eumakri/Documents/test_projects/tinyxml2/xmltest.cpp:7:
/home/eumakri/Documents/test_projects/tinyxml2/tinyxml2.h: In member function 'void tinyxml2::DynArray<T, INITIAL_SIZE>::EnsureCapacity(int) [with T = char; int INITIAL_SIZE = 10]':
/home/eumakri/Documents/test_projects/tinyxml2/tinyxml2.h:310:19: warning: use of possibly-NULL 'newMem' where non-null expected [CWE-690] [-Wanalyzer-possible-null-argument]
310 | memcpy(newMem, _mem, sizeof(T)*_size); // warning: not using constructors, only works for PODs
'void tinyxml2::DynArray<T, INITIAL_SIZE>::EnsureCapacity(int) [with T = char; int INITIAL_SIZE = 10]': events 1-4
305 | if (cap > _allocated) {
(1) following 'true' branch...
306 | TIXMLASSERT(cap <= INT_MAX / 2);
307 | const int newAllocated = cap * 2;
(2) ...to here
308 | T* newMem = new T[newAllocated];
(3) this call could return NULL
309 | TIXMLASSERT(newAllocated >= _size);
310 | memcpy(newMem, _mem, sizeof(T)*_size); // warning: not using constructors, only works for PODs
(4) argument 1 ('newMem') from (3) could be NULL where non-null expected
In file included from /usr/include/c++/12/cstring:42,
from /home/eumakri/Documents/test_projects/tinyxml2/tinyxml2.h:41:
/usr/include/string.h:43:14: note: argument 1 of 'void* memcpy(void*, const void*, size_t)' must be non-null
43 | extern void *memcpy (void *__restrict __dest, const void *__restrict __src,
/home/eumakri/Documents/test_projects/tinyxml2/xmltest.cpp: In function 'int main(int, const char**)':
/home/eumakri/Documents/test_projects/tinyxml2/xmltest.cpp:317:52: warning: use of possibly-NULL 'operator new(776)' where non-null expected [CWE-690] [-Wanalyzer-possible-null-argument]
317 | XMLDocument* doc = new XMLDocument();
'int main(int, const char**)': events 1-4
316 | if (argc > 1) {
(1) following 'true' branch (when 'argc > 1')...
317 | XMLDocument* doc = new XMLDocument();
(2) ...to here
(3) this call could return NULL
(4) argument 'this' ('operator new(776)') from (3) could be NULL where non-null expected
/home/eumakri/Documents/test_projects/tinyxml2/tinyxml2.h:1728:5: note: argument 'this' of 'tinyxml2::XMLDocument::XMLDocument(bool, tinyxml2::Whitespace)' must be non-null
1728 | XMLDocument(bool processEntities = true, Whitespace whitespaceMode = PRESERVE_WHITESPACE);
2848
2849 void XMLPrinter::PushText(int64_t value)
2850 {
2851 char buf[BUF_SIZE];
2852 XMLUtil::ToStr(value, buf, BUF_SIZE);


CodeChecker 6.22.2
Default
BACK TO REPORTS
Low
L316 – alpha.core.Conversion [22
L325 – alpha.core.Converse
Loss of sign in implicit con
L131 – Assuming field 'b
L149 – Assuming field 'fo
L156 – Calling 'TextRend
L409 – Entered call fr
L414 – Assuming the
L416 – Assuming field
L421 – Assuming field
L422 – Calling 'TextR
L436 – Entered ca
L443 – Assuming t
L446 – Calling 'Tex
L178 – Entered
L185 – Entering
L187 – Calling '
L206 – Enter
L214 – Assu
L217 – Assu
L228 – Assu
L278 – 'yMin
L286 – Assu
L313 – Assu
L320 – Assu
L325 – Loss
tinyxml2.cpp:2825 – following
tinyxml2.cpp:2826 – ...to here
tinyxml2.cpp:2837 – returning
tinyxml2.cpp:2838 – following
tinyxml2.cpp:2844 – ...to here
tinyxml2.cpp:2844 – calling 'tir
tinyxml2.cpp:2639 – entry to 't
tinyxml2.cpp:2642 – 'p' is NUL
tinyxml2.cpp:2644 – following
tinyxml2.cpp:2689 – ...to here
tinyxml2.cpp:2689 – calling 'tir
tinyxml2.h:2338 – entry to 'tiny
tinyxml2.h:2338 – argument 1 ('data') N
tinyxml2.h:2338 – use of NULL 'data' wl
CodeChecke
/xmltest.dir/xmltes
In file included fro
/home/eumakri/Docume
/home/eumakri/Docume
310 |
'void tinyxml2::Dy
305 |
306 |
307 |
308 |
309 |
310 |
In file included from /usr/include/c++/12/cstring:42,
from /home/eumakri/Documents/test_projects/tinyxml2/tinyxml2.h:41:
/usr/include/string.h:43:14: note: argument 1 of 'void* memcpy(void*, const void*, size_t)' must be non-null
43 | extern void *memcpy (void *__restrict __dest, const void *__restrict __src,
^~~~~~
/home/eumakri/Documents/test_projects/tinyxml2/xmltest.cpp: In function 'int main(int, const char**)':
/home/eumakri/Documents/test_projects/tinyxml2/xmltest.cpp:317:52: warning: use of possibly-NULL 'operator new(776)' where non-null expected [CWE-690] [-Wanalyzer-possible-null-argument]
317 | XMLDocument* doc = new XMLDocument();
^
'int main(int, const char**)': events 1-4
316 | if (argc > 1) {
^~
(1) following 'true' branch (when 'argc > 1')...
317 | XMLDocument* doc = new XMLDocument();
~
(2) ...to here
(3) this call could return NULL
(4) argument 'this' ('operator new(776)') from (3) could be NULL where non-null expected
/home/eumakri/Documents/test_projects/tinyxml2/tinyxml2.h:1728:5: note: argument 'this' of 'tinyxml2::XMLDocument::XMLDocument(bool, tinyxml2::Whitespace)' must be non-null
1728 | XMLDocument(bool processEntities = true, Whitespace whitespaceMode = PRESERVE_WHITESPACE);
^~~~~~~~~~~
2848
2849 void XMLPrinter::PushText(int64_t value)
2850 {
2851 char buf[BUF_SIZE];
2852 XMLUtil::ToStr(value, buf, BUF_SIZE);
CodeChecker 6.23
gcc static analyzer
PRODUCTS
RUNS
REPORTS
Search for runs...
Filter events by tag name...
History stored after...
History stored before...
Name
Number of unresolved reports
Analyzer statistics
Latest storage date
Analysis duration
tinyxml2
(8)
8
2023-08-18 12:54:30
00:00:00
August, 2023
2023-08-18 12:54:30
Anonymous
Gcc first
(19)
19
2023-08-18 11:20:47
00:00:00
August, 2023
2023-08-18 11:20:47
Anonymous


CodeChecker 6.22.2
Default
BACK TO REPORTS
Low
L316 – alpha.core.Conversion [22
L325 – alpha.core.Conver
Loss of sign in implicit con
L131 – Assuming field 'b
L149 – Assuming field 'fo
L156 – Calling 'TextRend
L409 – Entered call fr
L414 – Assuming the
L416 – Assuming field
L421 – Assuming field
L422 – Calling 'TextR
L436 – Entered ca
L443 – Assuming t
L446 – Calling 'Tex
L178 – Entered
L185 – Entering
L187 – Calling '
L206 – Enter
L214 – Assu
L217 – Assu
L228 – Assu
L278 – 'yMin
L286 – Assu
L313 – Assu
L320 – Assu
L325 – Loss
tinyxml2.cpp:2825 – following
tinyxml2.cpp:2826 – ...to here
tinyxml2.cpp:2837 – returning
tinyxml2.cpp:2838 – following
tinyxml2.cpp:2844 – ...to here
tinyxml2.cpp:2844 – calling 'tir
tinyxml2.cpp:2639 – entry to 't
tinyxml2.cpp:2642 – 'p' is NUL
tinyxml2.cpp:2644 – following
tinyxml2.cpp:2689 – ...to here
tinyxml2.cpp:2689 – calling 'tir
tinyxml2.h:2338 – entry to 'tiny
tinyxml2.h:2338 – argument 1 ('data') N
tinyxml2.h:2338 – use of NULL 'data' wl
CodeChecke
/xmltest.dir/xmltes
In file included fro
/home/eumakri/Docume
/home/eumakri/Docume
310 |
'void tinyxml2::Dy
305 |
306 |
307 |
308 |
309 |
310 |
In file included from /usr/include/c++/12/cstring:42,
from /home/eumakri/Documents/test_projects/tinyxml2/tinyxml2.h:41:
/usr/include/string.h:43:14: note: argument 1 of 'void* memcpy(void*, const void*, si
43 | extern void *memcpy (void *__restrict __dest, const void *__restrict __src,
^~~~~~
/home/eumakri/Documents/test_projects/tinyxml2/xmltest.cpp: In function 'int main(int
/home/eumakri/Documents/test_projects/tinyxml2/xmltest.cpp:317:52: warning: use of po
317 | XMLDocument* doc = new XMLDocument();
^
'int main(int, const char**)': events 1-4
316 | if (argc > 1) {
~~
(1) following 'true' branch (when 'argc > 1')...
317 | XMLDocument* doc = new XMLDocument();
~
(2) ...to here
(3) this call could re
(4) argument 'this' (
/home/eumakri/Documents/test_projects/tinyxml2/tinyxml2.h:1728:5: note: argument 'this
1728 | XMLDocument(bool processEntities = true, Whitespace whitespaceMode = PRES
^~~~~~~~~~~
2848
2849 void XMLPrinter::PushText(int64_t value)
2850 {
2851 char buf[BUF_SIZE];
2852 XMLUtil::ToStr(value, buf, BUF_SIZE);
CodeChecker 6.23
gcc static analyzer
PRODUCTS
RUNS
REPORTS
Search for runs...
Filter events by tag name...
History stored after...
History stored before...
Name
Number of unresolved reports
Analyzer statistics
Latest storage date
Analysis duration
tinyxml2
(8)
8
2023-08-18 12:54:30
00:00:00
August, 2023
2023-08-18 12:54:30
Anonymous
Gcc first
(19)
19
2023-08-18 11:20:47
00:00:00
August, 2023
2023-08-18 11:20:47
Anonymous
CodeChecker 6.23
gcc static analyzer
PRODUCTS
RUNS
REPORTS
STATISTICS
CONFIGUR
CLEAR ALL FILTERS
18
Unique reports
BASELINE
Run / Tag Filter
Outstanding reports on a given date
COMPARE TO
File path
Checker name
Severity
Latest Review Status
Latest Detection Status
New
Reopened
Unresolved
Analyzer name
Source component
Cleanup plan
Checker message
Dates
PRODUCT OVERVIEW
CHECKER STATISTICS
SEVERITY STATISTICS
COMPONENT STATISTICS
Number of outstanding reports
0
0
0
18
Today
Yesterday
Last 7 days
Last 31 days
Number of resolved reports
0
0
0
Today
Yesterday
Last 7 days
NUMBER OF FAILED FILES
0
NUMBER OF CHECKERS REPORTING FAULTS
4
Number of outstanding reports
Last
7
days
Critical
High
Medium
Low
Style
Unspecified
2023. Sep. 06
Critical: 0
High: 0
Medium: 0
Low: 0
Style: 0
Unspecified: 18
Total: 18
2023. Sep. 03
2023. Sep. 04
2023. Sep. 05
2023. Sep. 06
2023. Sep. 07
2023. Sep. 08

CodeChecker 6.22.2 Default

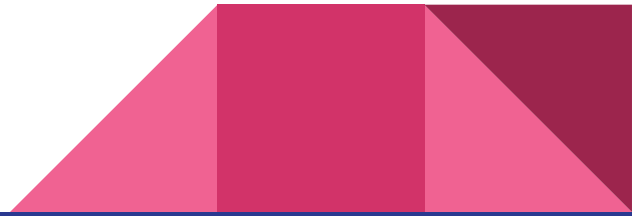
```
/home/eumakri/Documents/llvm-project/clang/test/Analysis/iterator-range.cpp:915:12: warning: Iterator incremented behind the past-the-end iterator [
.cplusplus.IteratorRange]
  (void)(i + 2); // expected-warning{{Iterator incremented behind the past-the-end iterator}}
         ~~^~~
/home/eumakri/Documents/llvm-project/clang/test/Analysis/iterator-range.cpp:915:12: note: Iterator incremented behind the past-the-end iterator
  (void)(i + 2); // expected-warning{{Iterator incremented behind the past-the-end iterator}}
         ~~^~~
/home/eumakri/Documents/llvm-project/clang/test/Analysis/iterator-range.cpp:921:5: warning: Iterator incremented behind the past-the-end iterator [a
cplusplus.IteratorRange]
  i += 2; // expected-warning{{Iterator incremented behind the past-the-end iterator}}
  ~~^~~~
/home/eumakri/Documents/llvm-project/clang/test/Analysis/iterator-range.cpp:921:5: note: Iterator incremented behind the past-the-end iterator
  i += 2; // expected-warning{{Iterator incremented behind the past-the-end iterator}}
  ~~^~~~
/home/eumakri/Documents/llvm-project/clang/test/Analysis/iterator-range.cpp:927:12: warning: Iterator decremented ahead of its valid range [alpha.cp
us.IteratorRange]
  (void)(i - 2); // expected-warning{{Iterator decremented ahead of its valid range}}
         ~~^~~
/home/eumakri/Documents/llvm-project/clang/test/Analysis/iterator-range.cpp:927:12: note: Iterator decremented ahead of its valid range
  (void)(i - 2); // expected-warning{{Iterator decremented ahead of its valid range}}
         ~~^~~
/home/eumakri/Documents/llvm-project/clang/test/Analysis/iterator-range.cpp:934:5: warning: Iterator decremented ahead of its valid range [alpha.cp
s.IteratorRange]
  i -= 2; // expected-warning{{Iterator decremented ahead of its valid range}}
  ~~^~~~
/home/eumakri/Documents/llvm-project/clang/test/Analysis/iterator-range.cpp:934:5: note: Iterator decremented ahead of its valid range
  i -= 2; // expected-warning{{Iterator decremented ahead of its valid range}}
  ~~^~~~
/home/eumakri/Documents/llvm-project/clang/test/Analysis/iterator-range.cpp:945:12: warning: The right operand of '-' is a garbage value [core.Undef
inaryOperatorResult]
  return n - uninit; // no-crash
           ^ ~~~~~~
/home/eumakri/Documents/llvm-project/clang/test/Analysis/iterator-range.cpp:944:3: note: 'uninit' declared without an initial value
  int uninit; // expected-note{{'uninit' declared without an initial value}}
  ^~~~~~~~~~
/home/eumakri/Documents/llvm-project/clang/test/Analysis/iterator-range.cpp:945:12: note: The right operand of '-' is a garbage value
  return n - uninit; // no-crash
```

```
2851    char buf[BUF_SIZE];
2852    XMLUtil::ToStr( value, buf, BUF_SIZE );
```

How do we measure this stuff?

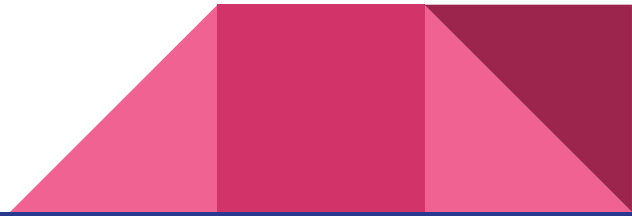
How do we measure this stuff?

- Whether something is explained well is inherently subjective
- We have great intuition on what's good
- Intuition is hard to define in concrete terms



Agenda

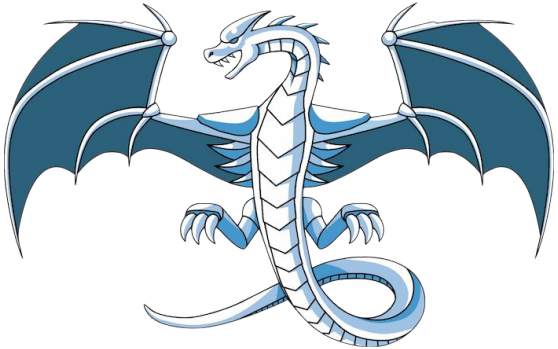
- Symbolic execution and the Clang Static Analyzer
  - How the analysis works
  - How bug report generation works
- Problem examples
- Previous (failed) survey
- Manual survey
- Latest survey



Symbolic execution and the Clang Static Analyzer

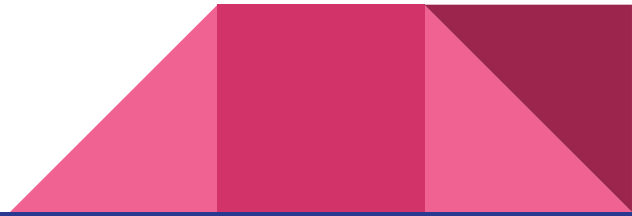
The Clang Static Analyzer

- Arguably the most popular and powerful C++ compiler today
- Part of LLVM
- The Clang Static Analyzer of a component of Clang
- Based on symbolic execution



Symbolic execution

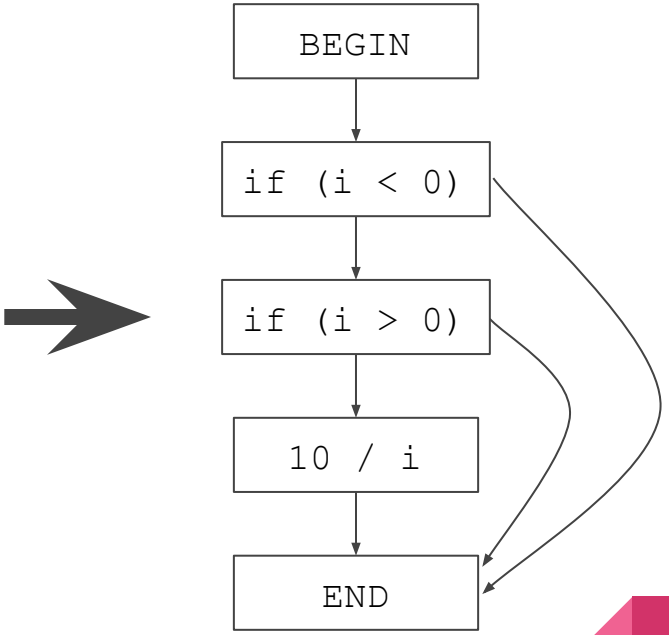
- Traverse the control flow graph (CFG) of a function
- Explored multiple path of execution
- On branches, explore a path on which the condition is true, and one on which its false



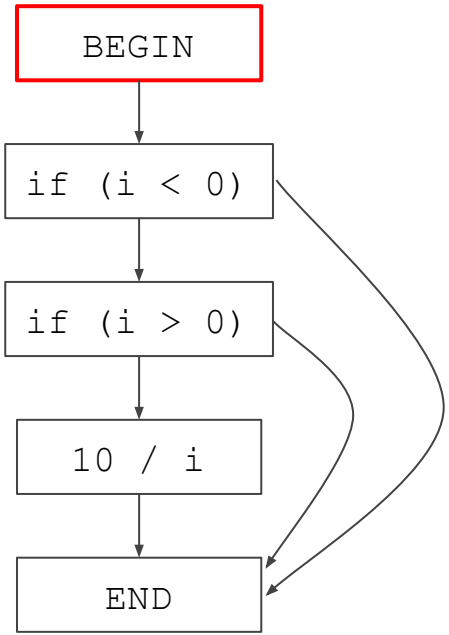
```
void a(int i) {
  if (i < 0)
    return;
  if (i > 0)
    return;
  10 / i;
}
```

Control Flow Graphs

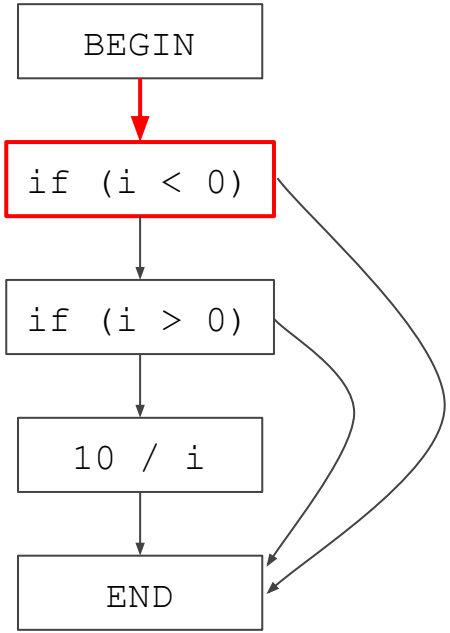
```
void a(int i) {
  if (i < 0)
    return;
  if (i > 0)
    return;
  10 / i;
}
```

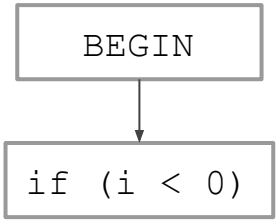


Exploded Graphs



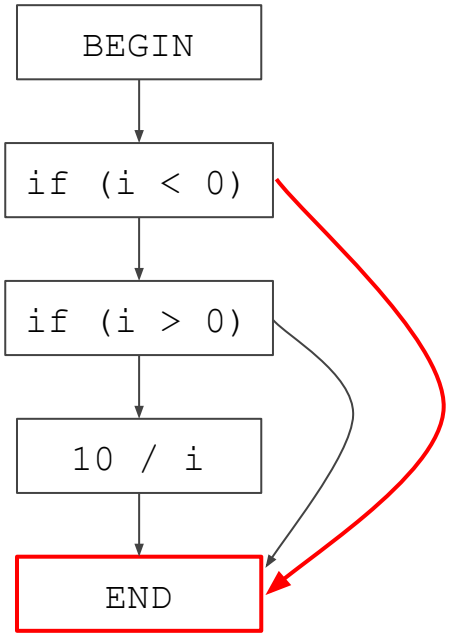
Exploded Graphs

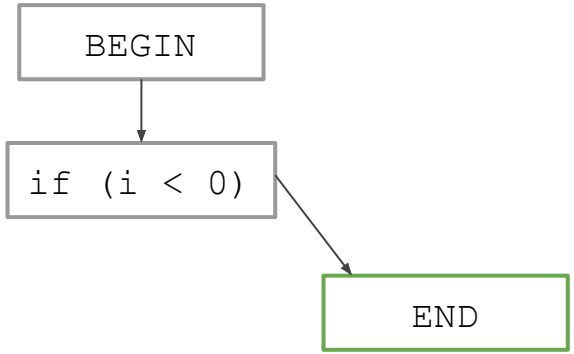




$i \in (-\infty, +\infty)$

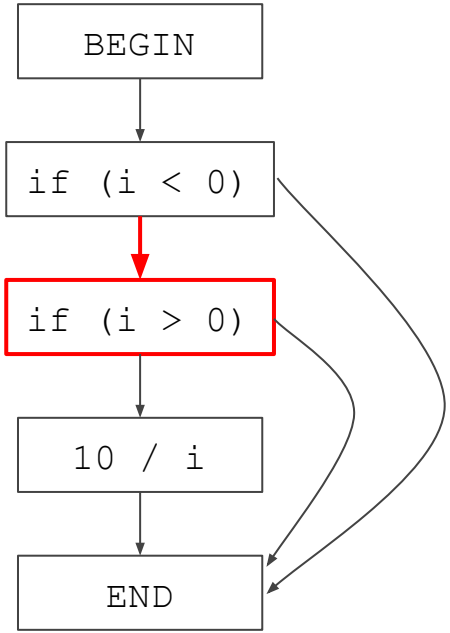
Exploded Graphs



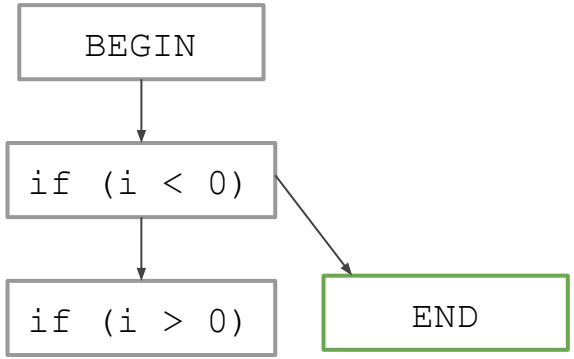


$i \in (-\infty, 0)$

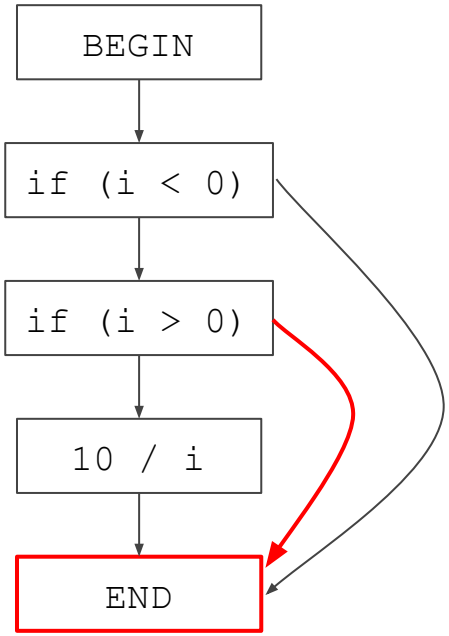
Exploded Graphs

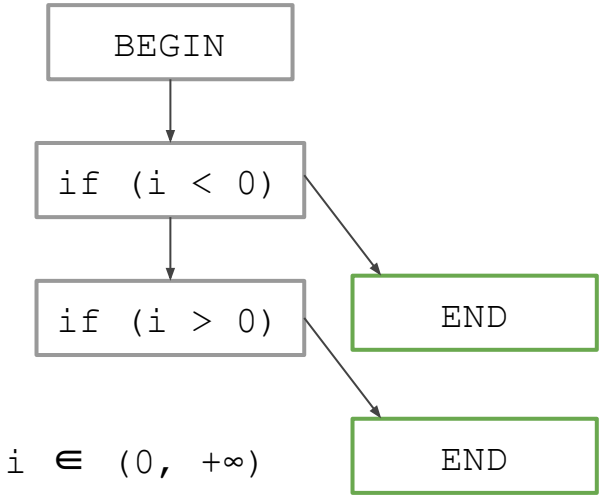


$i \in [0, +\infty)$

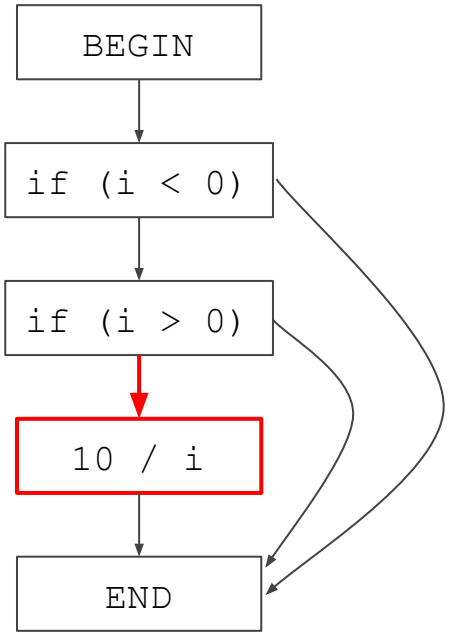


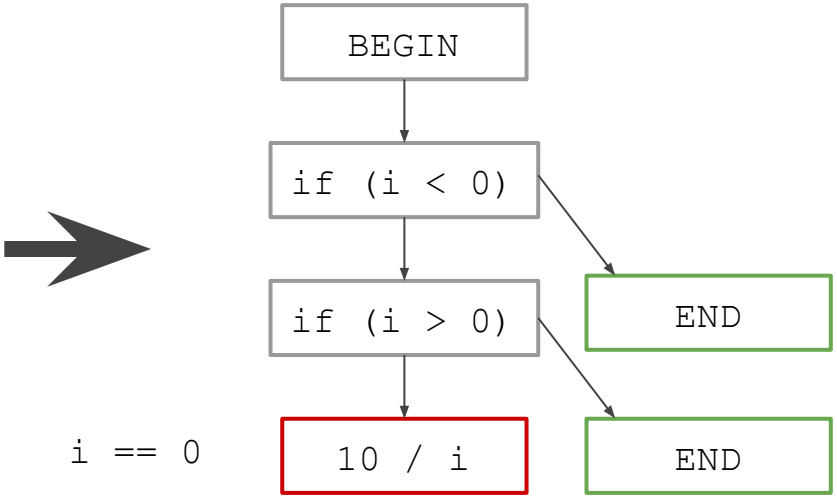
Exploded Graphs



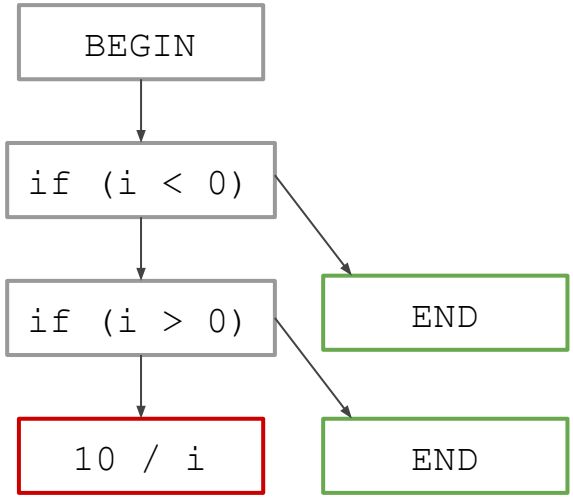


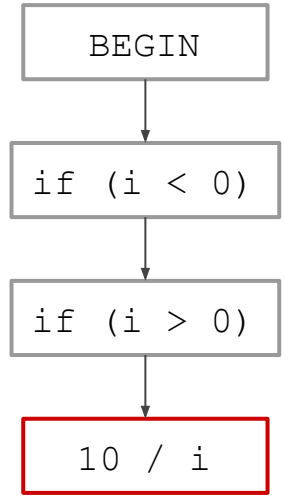
Exploded Graphs



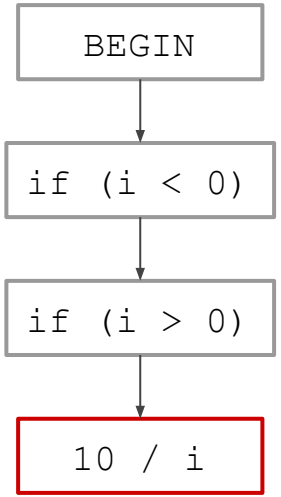


Report generation





Report generation



Assuming ‘i’ is >= 0

Assuming ‘i’ is <= 0

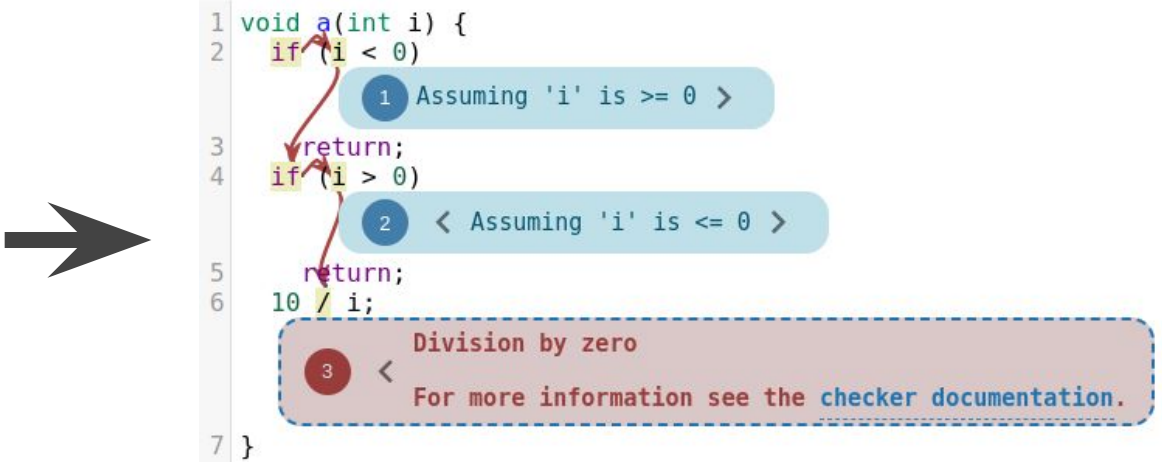
Division by zero

Report generation

Assuming ‘i’ is >= 0

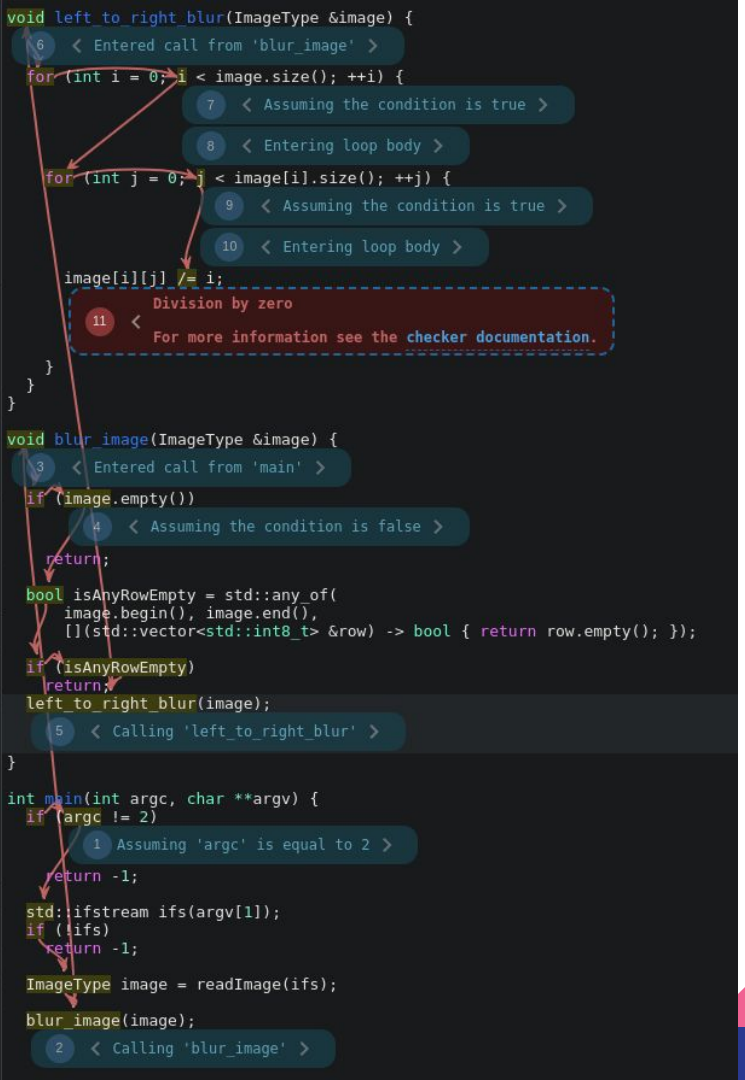
Assuming ‘i’ is <= 0

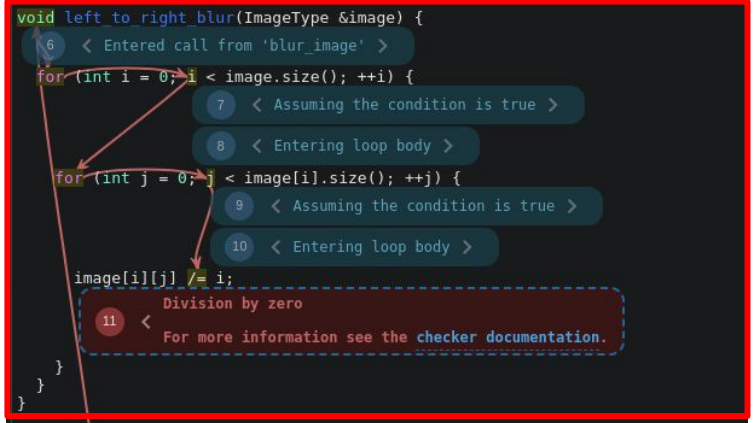
Division by zero

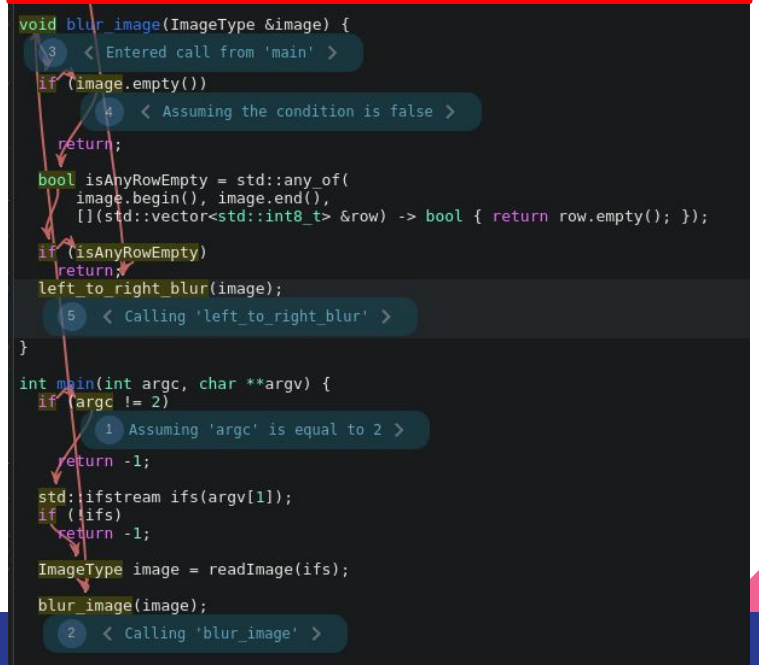


Problem example 1

```cpp
using ImageType = std::vector<std::vector<std::int8_t>>;

ImageType readImage(std::istream &input);
void writeImage(std::ostream &input, const ImageType &image);

void left_to_right_blur(ImageType &image) {
  for (int i = 0; i < image.size(); ++i) {
    for (int j = 0; j < image[i].size(); ++j) {
      image[i][j] /= i;
    }
  }
}

void blur_image(ImageType &image) {
  if (image.empty())
    return;

  bool isAnyRowEmpty = std::any_of(
      image.begin(), image.end(),
      [](std::vector<std::int8_t> &row) -> bool { return row.empty(); });

  if (isAnyRowEmpty)
    return;
  left_to_right_blur(image);
}

int main(int argc, char **argv) {
  if (argc != 2)
    return -1;

  std::ifstream ifs(argv[1]);
  if (!ifs)
    return -1;

  ImageType image = readImage(ifs);

  blur_image(image);

  std::ofstream ofs("output.jpg");
  if (!ofs)
    return -1;
  writeImage(ofs, image);
}
```

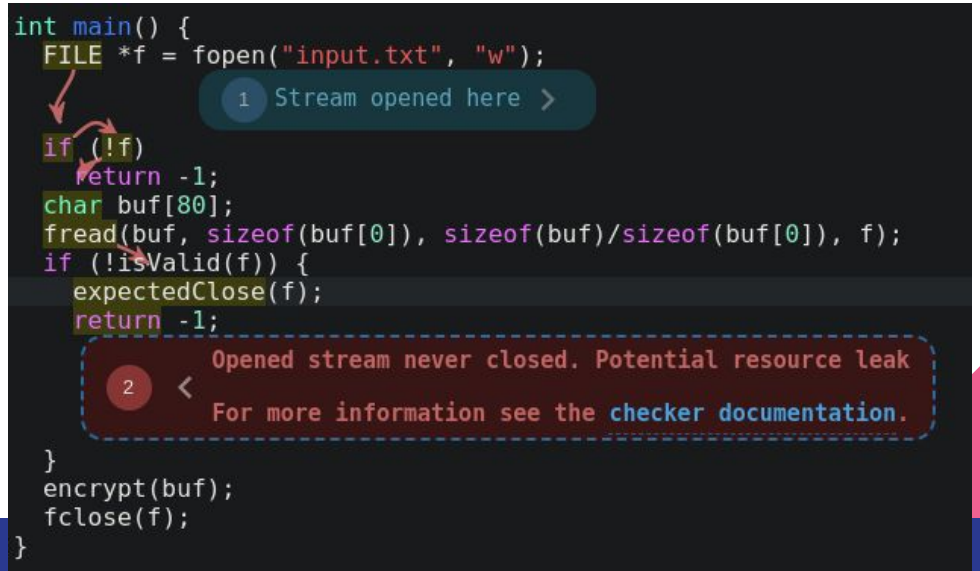

void left_to_right_blur(ImageType &image) {
6 Entered call from 'blur_image'
for (int i = 0; i < image.size(); ++i) {
7 Assuming the condition is true
8 Entering loop body
for (int j = 0; j < image[i].size(); ++j) {
9 Assuming the condition is true
10 Entering loop body
image[i][j] /= i;
11 Division by zero
For more information see the checker documentation.
}
}
}
void blur_image(ImageType &image) {
3 Entered call from 'main'
if (image.empty())
4 Assuming the condition is false
return;
bool isAnyRowEmpty = std::any_of(
image.begin(), image.end(),
[](std::vector<std::int8_t> &row) -> bool { return row.empty(); });
if (isAnyRowEmpty)
return;
left_to_right_blur(image);
5 Calling 'left_to_right_blur'
}
int main(int argc, char **argv) {
if (argc != 2)
1 Assuming 'argc' is equal to 2
return -1;
std::ifstream ifs(argv[1]);
if (!ifs)
return -1;
ImageType image = readImage(ifs);
blur_image(image);
2 Calling 'blur_image'

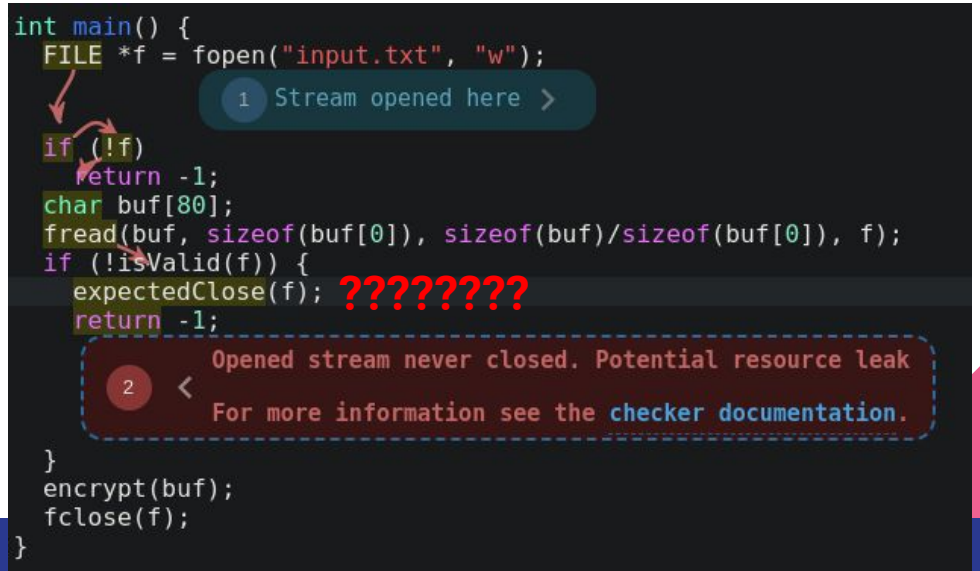

void left_to_right_blur(ImageType &image) {
6 Entered call from 'blur_image'
for (int i = 0; i < image.size(); ++i) {
7 Assuming the condition is true
8 Entering loop body
for (int j = 0; j < image[i].size(); ++j) {
9 Assuming the condition is true
10 Entering loop body
image[i][j] /= i;
11 Division by zero
For more information see the checker documentation.
}
}
}


void blur_image(ImageType &image) {
3 Entered call from 'main'
if (image.empty())
4 Assuming the condition is false
return;
bool isAnyRowEmpty = std::any_of(
image.begin(), image.end(),
[](std::vector<std::int8_t> &row) -> bool { return row.empty(); });
if (isAnyRowEmpty)
return;
left_to_right_blur(image);
5 Calling 'left_to_right_blur'
}
int main(int argc, char **argv) {
if (argc != 2)
1 Assuming 'argc' is equal to 2
return -1;
std::ifstream ifs(argv[1]);
if (!ifs)
return -1;
ImageType image = readImage(ifs);
blur_image(image);
2 Calling 'blur_image'

Problem example 2

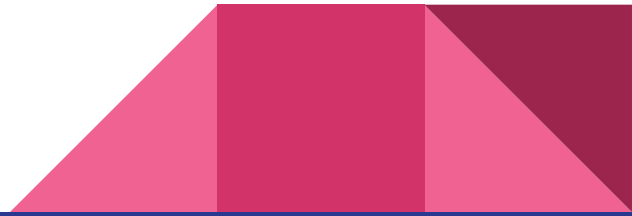
```
char *logDump();

void expectedClose(FILE *f) {
  if (char *log = logDump()) {
    printf("%s", log);
    fclose(f);

  }
}

void encrypt(std::string input) {
  std::string from = "aeiou";
  std::string to   = "eioua";
  for (char &c : input)
    for (int i = 0; i < from.size(); ++i)
      if (c == from[i])
        c = to[i];
}

bool isValid(FILE *F) {
  return !feof(F) && !ferror(F);
}


int main() {
  FILE *f = fopen("input.txt", "w");
  if (!f)
    return -1;
  char buf[80];
  fread(buf, sizeof(buf[0]), sizeof(buf)/sizeof(buf[0]), f);
  if (!isValid(f)) {
    expectedClose(f);
    return -1;
  }
  encrypt(buf);
  fclose(f);
}
```


int main() {
FILE *f = fopen("input.txt", "w");
1 Stream opened here
if (!f)
return -1;
char buf[80];
fread(buf, sizeof(buf[0]), sizeof(buf)/sizeof(buf[0]), f);
if (!isValid(f)) {
expectedClose(f);
return -1;
2
Opened stream never closed. Potential resource leak
For more information see the checker documentation.
}
encrypt(buf);
fclose(f);
}


int main() {
FILE *f = fopen("input.txt", "w");
1 Stream opened here
if (!f)
return -1;
char buf[80];
fread(buf, sizeof(buf[0]), sizeof(buf)/sizeof(buf[0]), f);
if (!isValid(f)) {
expectedClose(f);
????????
return -1;
2
Opened stream never closed. Potential resource leak
For more information see the checker documentation.
}
encrypt(buf);
fclose(f);
}

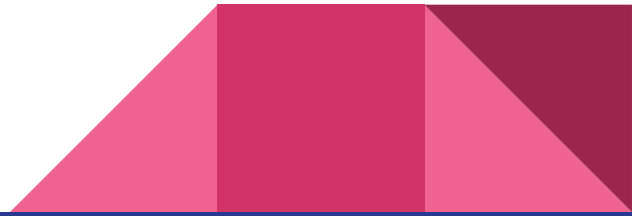
Bug reports have much to improve on

- Static analyzers are invaluable for detecting bugs early
- Experts need to evaluate the results manually
- Trust in the tool drops if the reports are poor



…but what to improve on?

- Whether a bug report is good is inherently subjective.
- We need a mathematical model to measure bug report quality on
- **That should be based on empirical data**

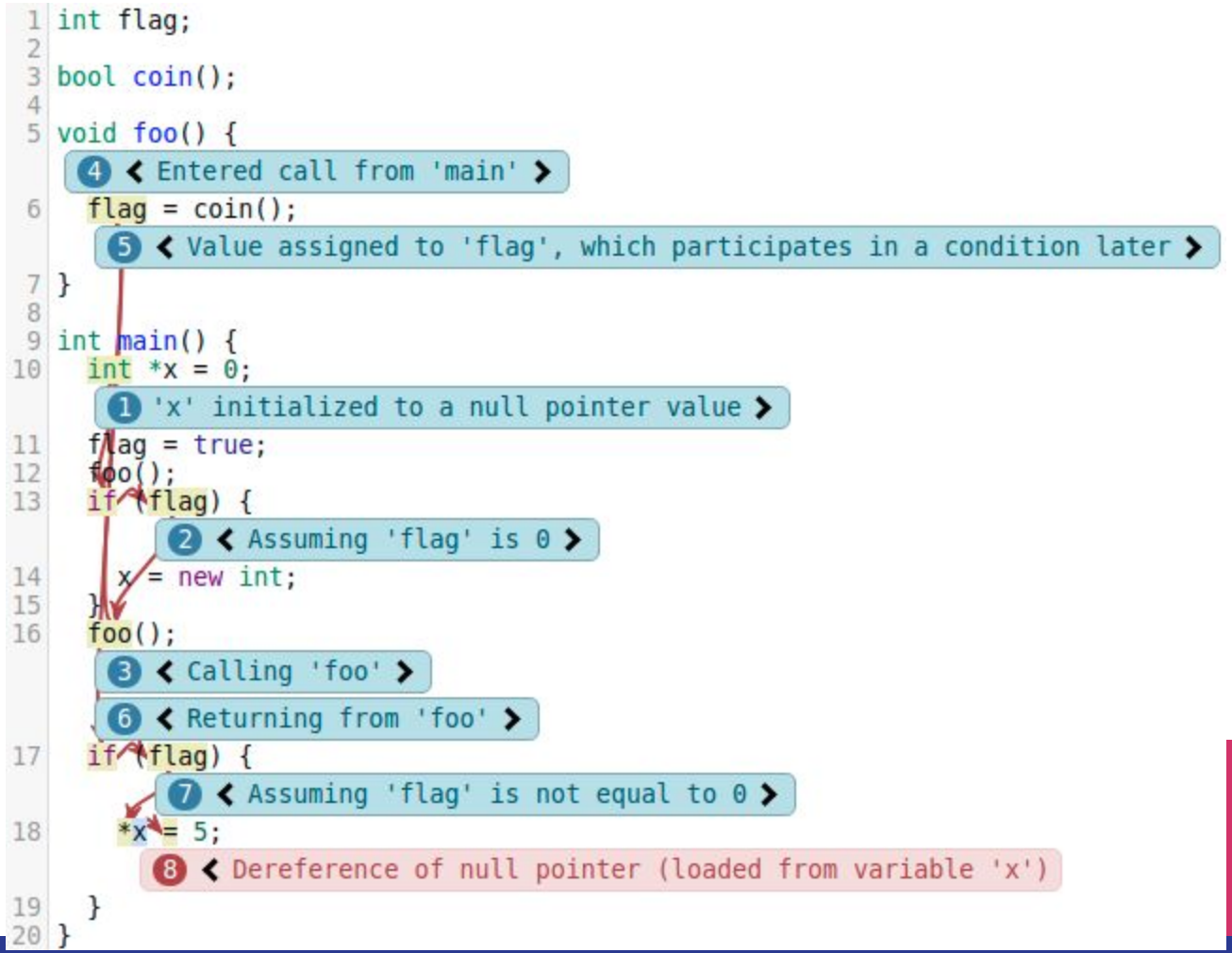


Previous (failed) survey

Addition of Control Dependency Analysis

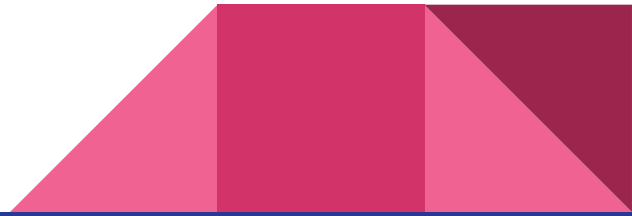
- CSA used to explain data dependency well
- Lack of understanding for control dependency
- Contribution as a part of GSoC’19


1 int flag;
2
3 bool coin();
4
5 void foo() {
6 flag = coin();
7 }
8
9 void bar() {
10 int *x = 0;
1 'x' initialized to a null pointer value
11 flag = true;
12 foo();
13 if (flag) {
2 Assuming 'flag' is 0
14 x = new int;
15 }
16 foo();
17 if (flag) {
3 Assuming 'flag' is not equal to 0
18 *x = 1;
4 Dereference of null pointer (loaded from variable 'x')
19 }
20 }

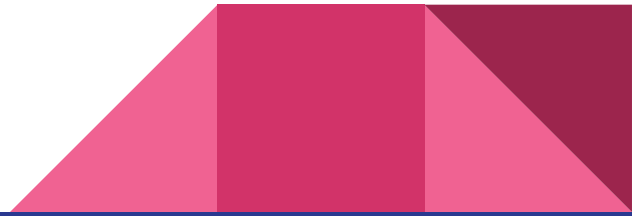

1 int flag;
2
3 bool coin();
4
5 void foo() {
4 Entered call from 'main'
6 flag = coin();
5 Value assigned to 'flag', which participates in a condition later
7 }
8
9 int main() {
10 int *x = 0;
1 'x' initialized to a null pointer value
11 flag = true;
12 foo();
13 if (flag) {
2 Assuming 'flag' is 0
14 x = new int;
15 }
16 foo();
3 Calling 'foo'
6 Returning from 'foo'
17 if (flag) {
7 Assuming 'flag' is not equal to 0
18 *x = 5;
8 Dereference of null pointer (loaded from variable 'x')
19 }
20 }

Survey

- 11 participants of the CodeChecker team
- Real bug reports on large, open source software
- Required domain specific knowledge
- No statistically relevant results



Manual survey



Measurement methodology

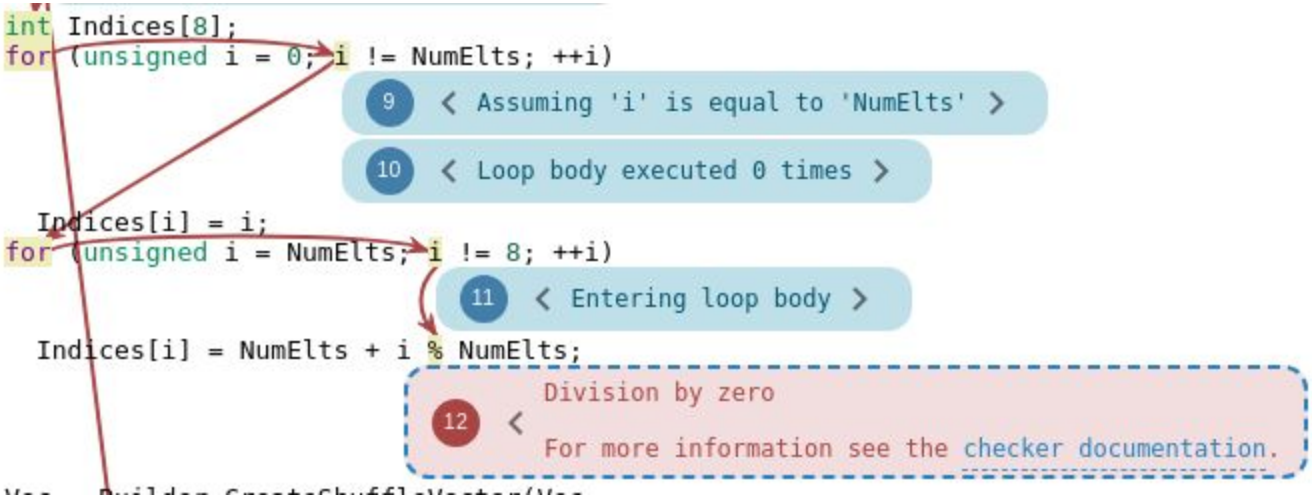
- Analyses on large, open source C/C++ projects
- Only results by core.DivByZero
- Manual inspection of all reports, up to 30 minutes on each

Categorization

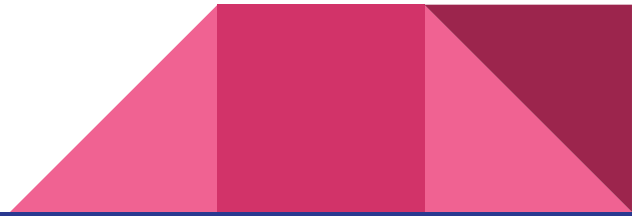
- **Acceptable**: It is possible to understand the report, and whether it stands, even if it could be improved.
- **Not enough info**: It is not possible understand the report, but it is possible to say which function calls / value changes the analyzer neglected to explain, and a domain expert may possess the missing information and judge whether the report stands.
- **Incomprehensible**: The entire bug report is incomprehensible, and its doubtful that even a domain expert can judge the report.

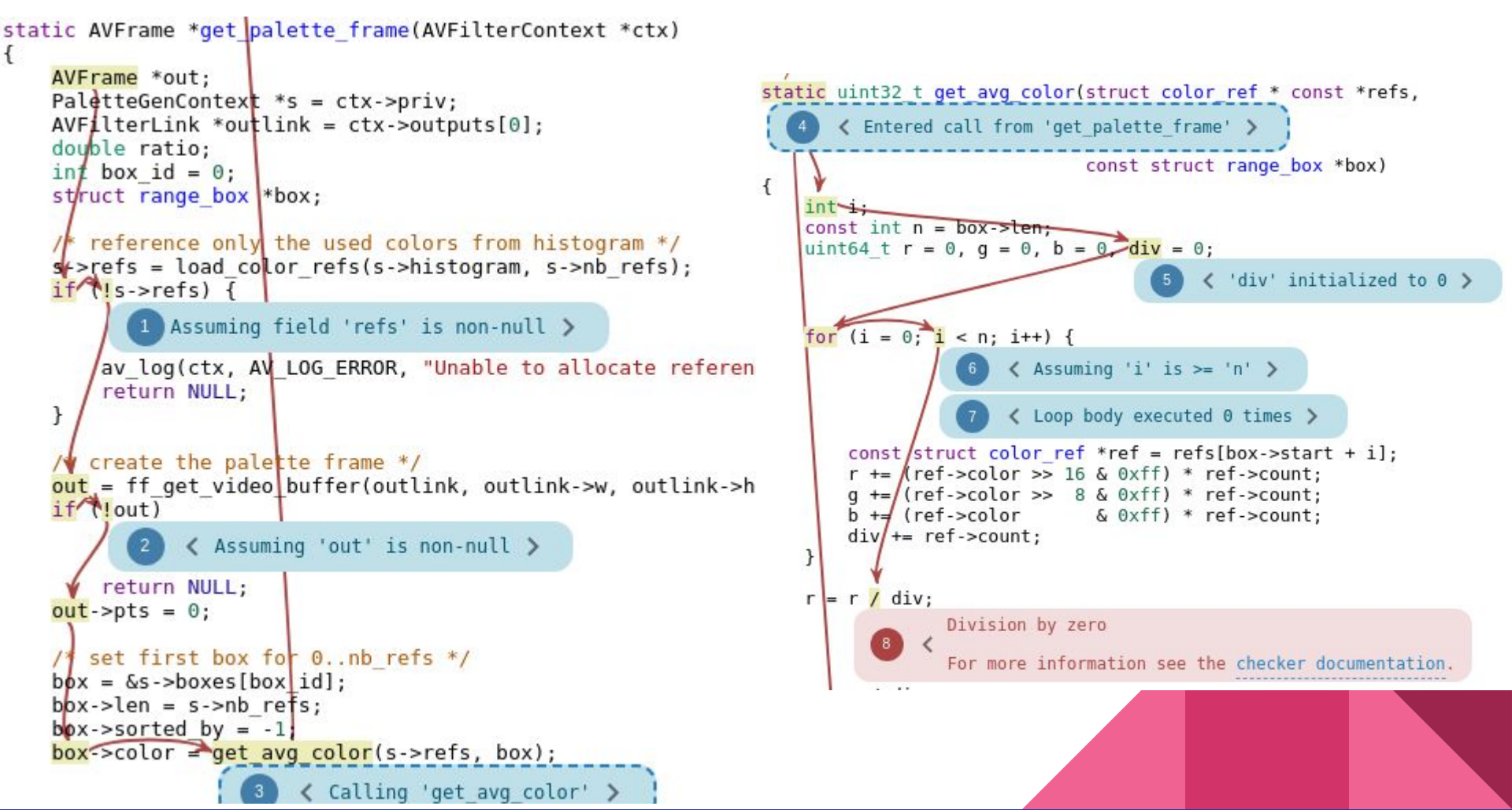
	Total reports	Acceptable	Not enough info	Incomprehensible
Acid	1	1	0	0
ffmpeg	6	5	1	0
LLVM + Clang	11	8	2	1
OpenSSL	1	1	0	0
postgres	2	1	0	1
QTBase	7	2	1	4
Vim	2	2	0	0
Xerces	1	0	1	0
Total	31	20	5	6

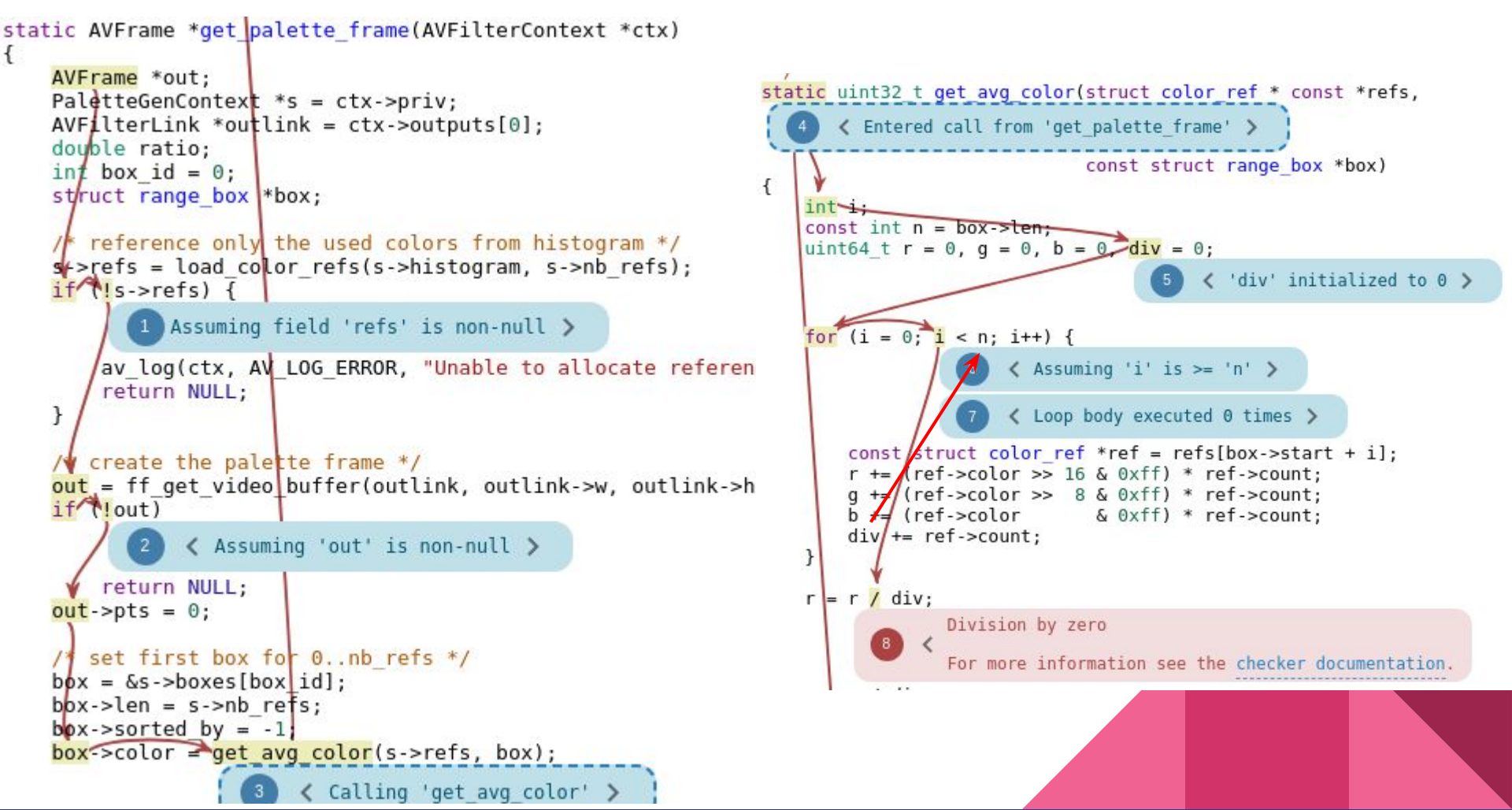
Acceptable (LLVM)

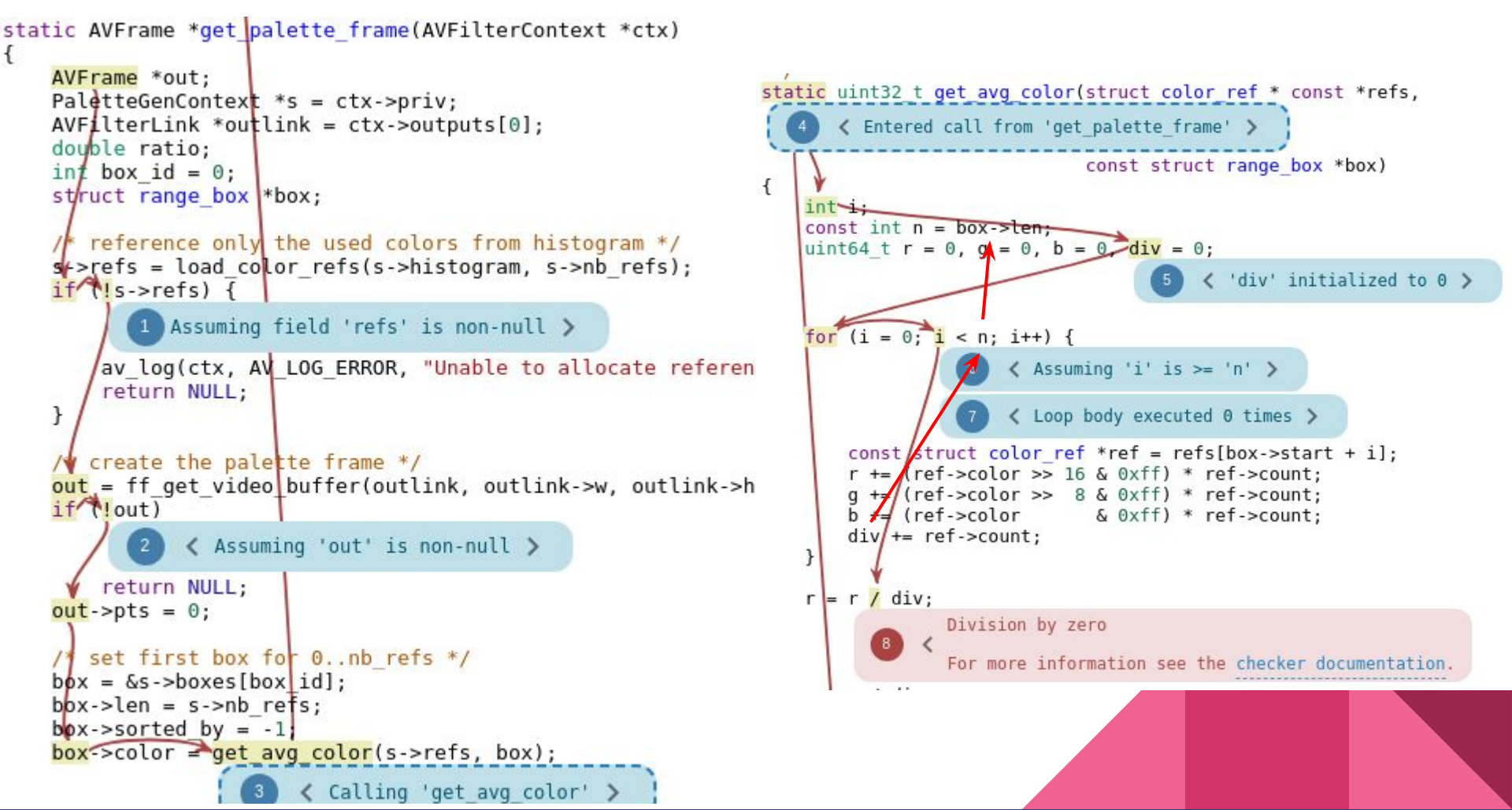


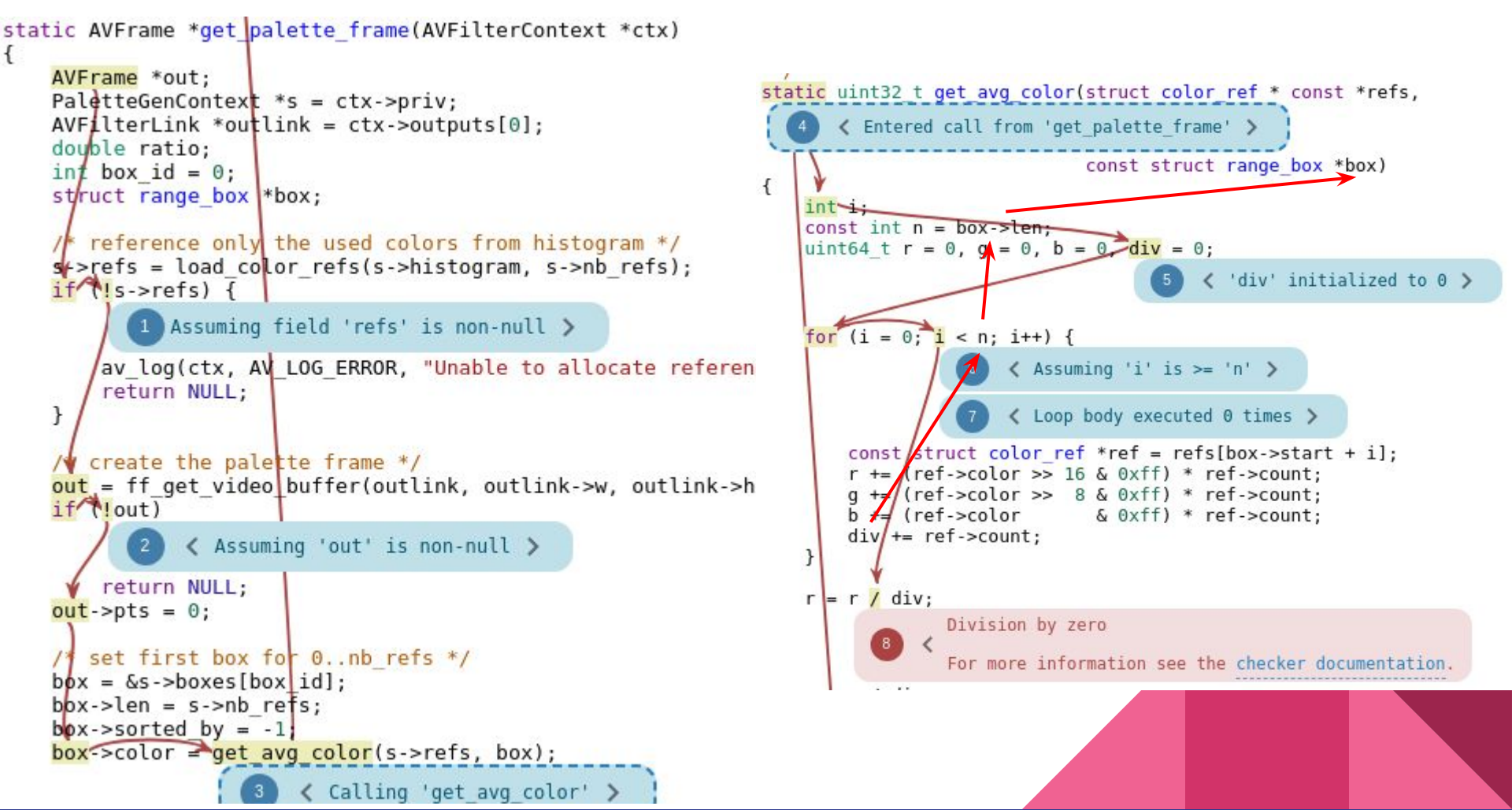
Not enough info (ffmpeg)

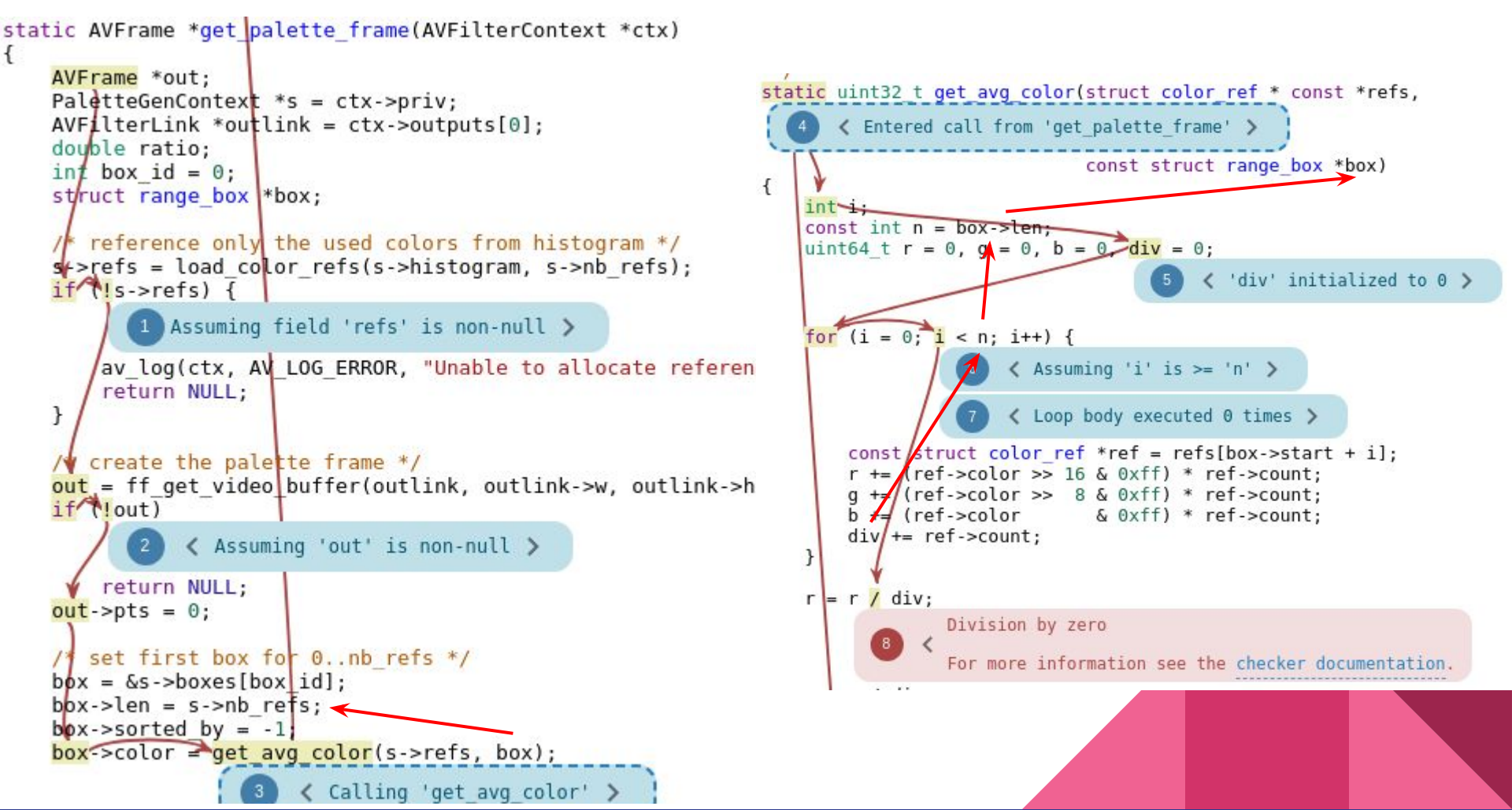


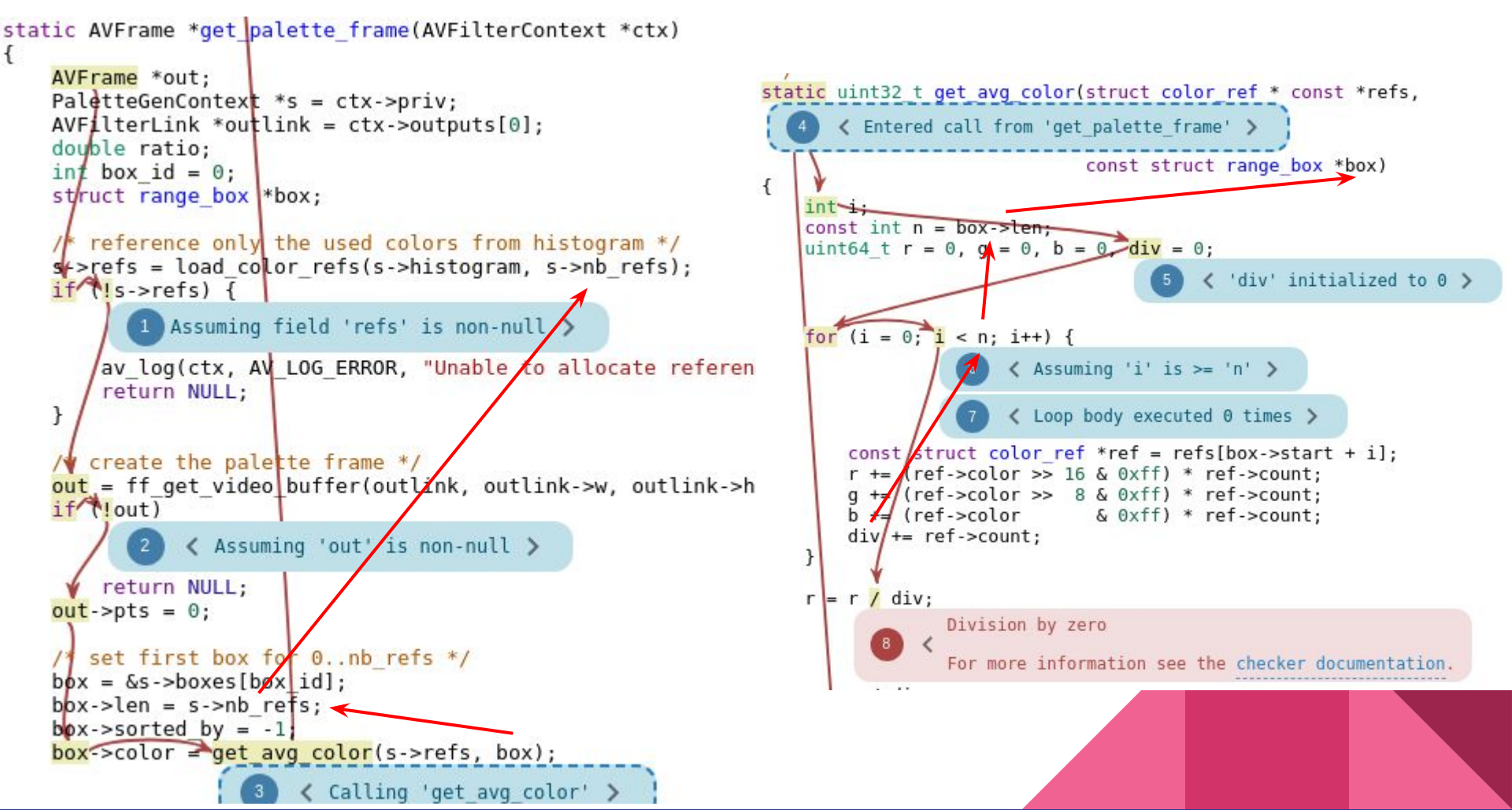

static AVFrame *get_palette_frame(AVFilterContext *ctx)
{
AVFrame *out;
PaletteGenContext *s = ctx->priv;
AVFilterLink *outlink = ctx->outputs[0];
double ratio;
int box_id = 0;
struct range_box *box;
/* reference only the used colors from histogram */
s->refs = load_color_refs(s->histogram, s->nb_refs);
if (!s->refs) {
1 Assuming field 'refs' is non-null
av_log(ctx, AV_LOG_ERROR, "Unable to allocate referen
return NULL;
}
/* create the palette frame */
out = ff_get_video_buffer(outlink, outlink->w, outlink->h
if (!out)
2 Assuming 'out' is non-null
return NULL;
out->pts = 0;
/* set first box for 0..nb_refs */
box = &s->boxes[box_id];
box->len = s->nb_refs;
box->sorted_by = -1;
box->color = get_avg_color(s->refs, box);
3 Calling 'get_avg_color'
static uint32_t get_avg_color(struct color_ref * const *refs,
4 Entered call from 'get_palette_frame'
const struct range_box *box)
{
int i;
const int n = box->len;
uint64_t r = 0, g = 0, b = 0, div = 0;
5 'div' initialized to 0
for (i = 0; i < n; i++) {
6 Assuming 'i' is >= 'n'
7 Loop body executed 0 times
const struct color_ref *ref = refs[box->start + i];
r += (ref->color >> 16 & 0xff) * ref->count;
g += (ref->color >> 8 & 0xff) * ref->count;
b += (ref->color & 0xff) * ref->count;
div += ref->count;
}
r = r / div;
8 Division by zero
For more information see the checker documentation.

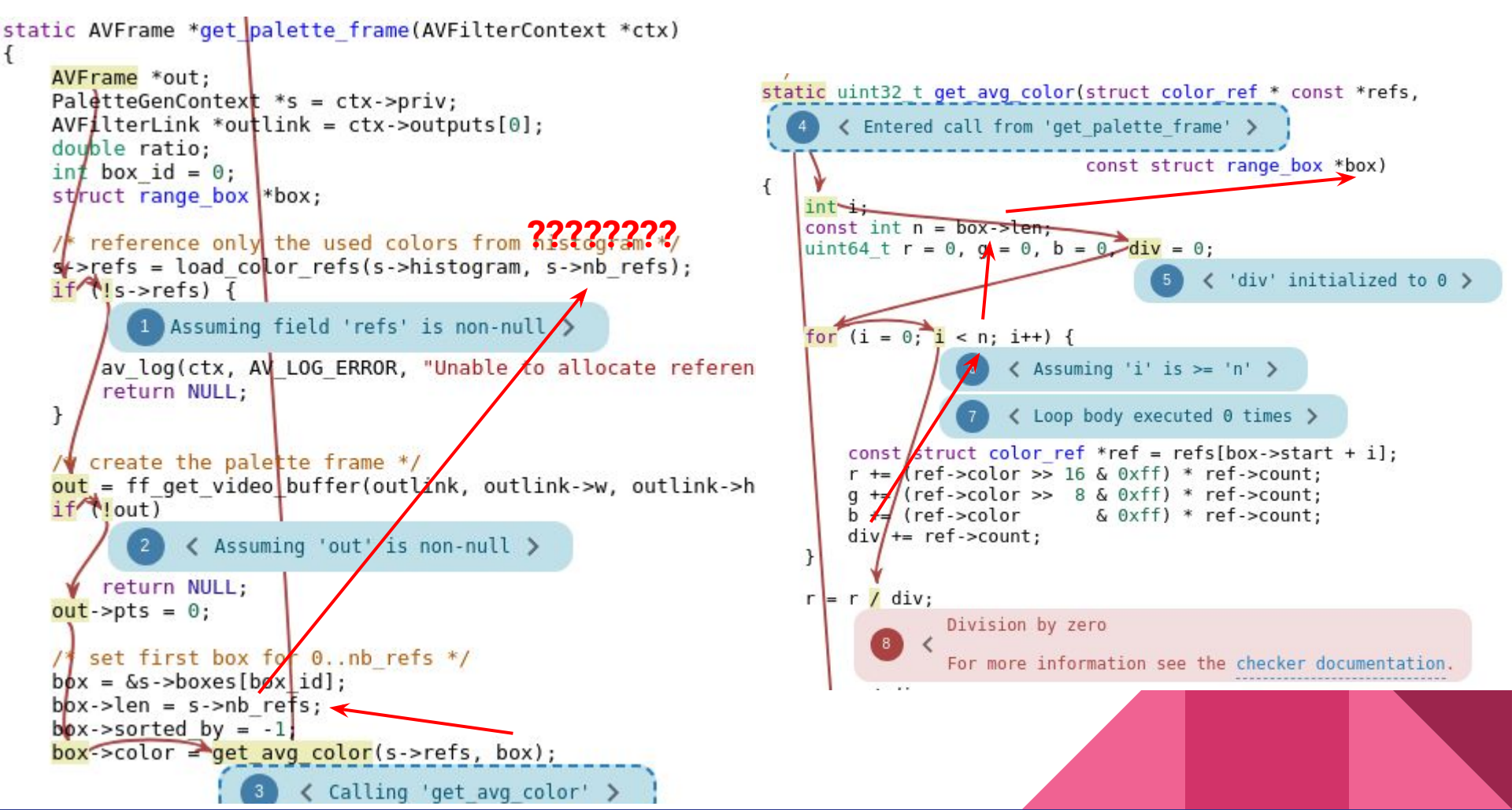

static AVFrame *get_palette_frame(AVFilterContext *ctx)
{
AVFrame *out;
PaletteGenContext *s = ctx->priv;
AVFilterLink *outlink = ctx->outputs[0];
double ratio;
int box_id = 0;
struct range_box *box;
/* reference only the used colors from histogram */
s->refs = load_color_refs(s->histogram, s->nb_refs);
if (!s->refs) {
1 Assuming field 'refs' is non-null
av_log(ctx, AV_LOG_ERROR, "Unable to allocate referen
return NULL;
}
/* create the palette frame */
out = ff_get_video_buffer(outlink, outlink->w, outlink->h
if (!out)
2 Assuming 'out' is non-null
return NULL;
out->pts = 0;
/* set first box for 0..nb_refs */
box = &s->boxes[box_id];
box->len = s->nb_refs;
box->sorted_by = -1;
box->color = get_avg_color(s->refs, box);
3 Calling 'get_avg_color'
static uint32_t get_avg_color(struct color_ref * const *refs,
4 Entered call from 'get_palette_frame'
const struct range_box *box)
{
int i;
const int n = box->len;
uint64_t r = 0, g = 0, b = 0, div = 0;
5 'div' initialized to 0
for (i = 0; i < n; i++) {
6 Assuming 'i' is >= 'n'
7 Loop body executed 0 times
const struct color_ref *ref = refs[box->start + i];
r += (ref->color >> 16 & 0xff) * ref->count;
g += (ref->color >> 8 & 0xff) * ref->count;
b += (ref->color & 0xff) * ref->count;
div += ref->count;
}
r = r / div;
8
Division by zero
For more information see the checker documentation.


static AVFrame *get_palette_frame(AVFilterContext *ctx)
{
AVFrame *out;
PaletteGenContext *s = ctx->priv;
AVFilterLink *outlink = ctx->outputs[0];
double ratio;
int box_id = 0;
struct range_box *box;
/* reference only the used colors from histogram */
s->refs = load_color_refs(s->histogram, s->nb_refs);
if (!s->refs) {
1 Assuming field 'refs' is non-null
av_log(ctx, AV_LOG_ERROR, "Unable to allocate referen
return NULL;
}
/* create the palette frame */
out = ff_get_video_buffer(outlink, outlink->w, outlink->h
if (!out)
2 Assuming 'out' is non-null
return NULL;
out->pts = 0;
/* set first box for 0..nb_refs */
box = &s->boxes[box_id];
box->len = s->nb_refs;
box->sorted_by = -1;
box->color = get_avg_color(s->refs, box);
3 Calling 'get_avg_color'
static uint32_t get_avg_color(struct color_ref * const *refs,
4 Entered call from 'get_palette_frame'
const struct range_box *box)
{
int i;
const int n = box->len;
uint64_t r = 0, g = 0, b = 0, div = 0;
5 'div' initialized to 0
for (i = 0; i < n; i++) {
6 Assuming 'i' is >= 'n'
7 Loop body executed 0 times
const struct color_ref *ref = refs[box->start + i];
r += (ref->color >> 16 & 0xff) * ref->count;
g += (ref->color >> 8 & 0xff) * ref->count;
b += (ref->color & 0xff) * ref->count;
div += ref->count;
}
r = r / div;
8
Division by zero
For more information see the checker documentation.

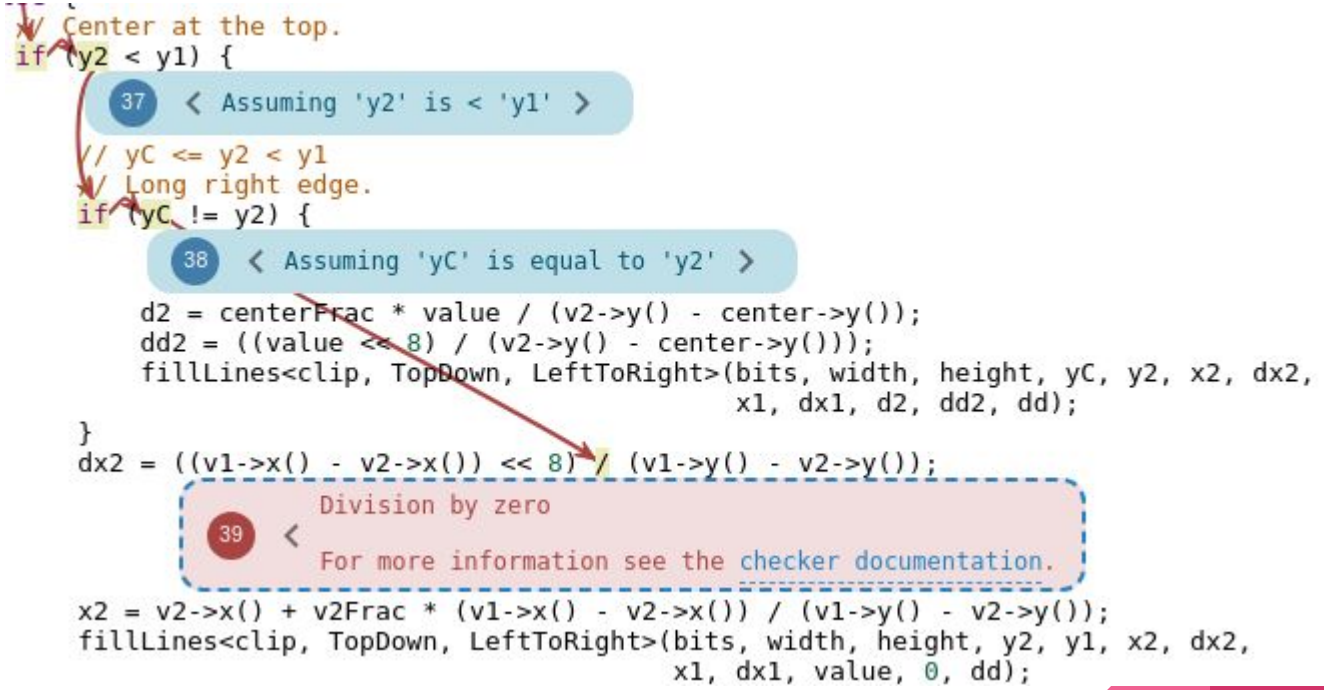

static AVFrame *get_palette_frame(AVFilterContext *ctx)
{
AVFrame *out;
PaletteGenContext *s = ctx->priv;
AVFilterLink *outlink = ctx->outputs[0];
double ratio;
int box_id = 0;
struct range_box *box;
/* reference only the used colors from histogram */
s->refs = load_color_refs(s->histogram, s->nb_refs);
if (!s->refs) {
1 Assuming field 'refs' is non-null
av_log(ctx, AV_LOG_ERROR, "Unable to allocate referen
return NULL;
}
/* create the palette frame */
out = ff_get_video_buffer(outlink, outlink->w, outlink->h
if (!out)
2 Assuming 'out' is non-null
return NULL;
out->pts = 0;
/* set first box for 0..nb_refs */
box = &s->boxes[box_id];
box->len = s->nb_refs;
box->sorted_by = -1;
box->color = get_avg_color(s->refs, box);
3 Calling 'get_avg_color'
static uint32_t get_avg_color(struct color_ref * const *refs,
4 Entered call from 'get_palette_frame'
const struct range_box *box)
{
int i;
const int n = box->len;
uint64_t r = 0, g = 0, b = 0, div = 0;
5 'div' initialized to 0
for (i = 0; i < n; i++) {
6 Assuming 'i' is >= 'n'
7 Loop body executed 0 times
const struct color_ref *ref = refs[box->start + i];
r += (ref->color >> 16 & 0xff) * ref->count;
g += (ref->color >> 8 & 0xff) * ref->count;
b += (ref->color & 0xff) * ref->count;
div += ref->count;
}
r = r / div;
8
Division by zero
For more information see the checker documentation.


static AVFrame *get_palette_frame(AVFilterContext *ctx)
{
AVFrame *out;
PaletteGenContext *s = ctx->priv;
AVFilterLink *outlink = ctx->outputs[0];
double ratio;
int box_id = 0;
struct range_box *box;
/* reference only the used colors from histogram */
s->refs = load_color_refs(s->histogram, s->nb_refs);
if (!s->refs) {
1 Assuming field 'refs' is non-null
av_log(ctx, AV_LOG_ERROR, "Unable to allocate referen
return NULL;
}
/* create the palette frame */
out = ff_get_video_buffer(outlink, outlink->w, outlink->h
if (!out)
2 Assuming 'out' is non-null
return NULL;
out->pts = 0;
/* set first box for 0..nb_refs */
box = &s->boxes[box_id];
box->len = s->nb_refs;
box->sorted_by = -1;
box->color = get_avg_color(s->refs, box);
3 Calling 'get_avg_color'
static uint32_t get_avg_color(struct color_ref * const *refs,
4 Entered call from 'get_palette_frame'
const struct range_box *box)
{
int i;
const int n = box->len;
uint64_t r = 0, g = 0, b = 0, div = 0;
5 'div' initialized to 0
for (i = 0; i < n; i++) {
6 Assuming 'i' is >= 'n'
7 Loop body executed 0 times
const struct color_ref *ref = refs[box->start + i];
r += (ref->color >> 16 & 0xff) * ref->count;
g += (ref->color >> 8 & 0xff) * ref->count;
b += (ref->color & 0xff) * ref->count;
div += ref->count;
}
r = r / div;
8
Division by zero
For more information see the checker documentation.


static AVFrame *get_palette_frame(AVFilterContext *ctx)
{
AVFrame *out;
PaletteGenContext *s = ctx->priv;
AVFilterLink *outlink = ctx->outputs[0];
double ratio;
int box_id = 0;
struct range_box *box;
/* reference only the used colors from histogram */
s->refs = load_color_refs(s->histogram, s->nb_refs);
if (!s->refs) {
1 Assuming field 'refs' is non-null
av_log(ctx, AV_LOG_ERROR, "Unable to allocate referen
return NULL;
}
/* create the palette frame */
out = ff_get_video_buffer(outlink, outlink->w, outlink->h
if (!out)
2 Assuming 'out' is non-null
return NULL;
out->pts = 0;
/* set first box for 0..nb_refs */
box = &s->boxes[box_id];
box->len = s->nb_refs;
box->sorted_by = -1;
box->color = get_avg_color(s->refs, box);
3 Calling 'get_avg_color'
static uint32_t get_avg_color(struct color_ref * const *refs,
4 Entered call from 'get_palette_frame'
const struct range_box *box)
{
int i;
const int n = box->len;
uint64_t r = 0, g = 0, b = 0, div = 0;
5 'div' initialized to 0
for (i = 0; i < n; i++) {
6 Assuming 'i' is >= 'n'
7 Loop body executed 0 times
const struct color_ref *ref = refs[box->start + i];
r += (ref->color >> 16 & 0xff) * ref->count;
g += (ref->color >> 8 & 0xff) * ref->count;
b += (ref->color & 0xff) * ref->count;
div += ref->count;
}
r = r / div;
8 Division by zero
For more information see the checker documentation.


static AVFrame *get_palette_frame(AVFilterContext *ctx)
{
AVFrame *out;
PaletteGenContext *s = ctx->priv;
AVFilterLink *outlink = ctx->outputs[0];
double ratio;
int box_id = 0;
struct range_box *box;
/* reference only the used colors from histogram */
s->refs = load_color_refs(s->histogram, s->nb_refs);
if (!s->refs) {
1 Assuming field 'refs' is non-null
av_log(ctx, AV_LOG_ERROR, "Unable to allocate referen
return NULL;
}
/* create the palette frame */
out = ff_get_video_buffer(outlink, outlink->w, outlink->h
if (!out)
2 Assuming 'out' is non-null
return NULL;
out->pts = 0;
/* set first box for 0..nb_refs */
box = &s->boxes[box_id];
box->len = s->nb_refs;
box->sorted_by = -1;
box->color = get_avg_color(s->refs, box);
3 Calling 'get_avg_color'
????????
static uint32_t get_avg_color(struct color_ref * const *refs,
4 Entered call from 'get_palette_frame'
const struct range_box *box)
{
int i;
const int n = box->len;
uint64_t r = 0, g = 0, b = 0, div = 0;
5 'div' initialized to 0
for (i = 0; i < n; i++) {
6 Assuming 'i' is >= 'n'
7 Loop body executed 0 times
const struct color_ref *ref = refs[box->start + i];
r += (ref->color >> 16 & 0xff) * ref->count;
g += (ref->color >> 8 & 0xff) * ref->count;
b += (ref->color & 0xff) * ref->count;
div += ref->count;
}
r = r / div;
8
Division by zero
For more information see the checker documentation.

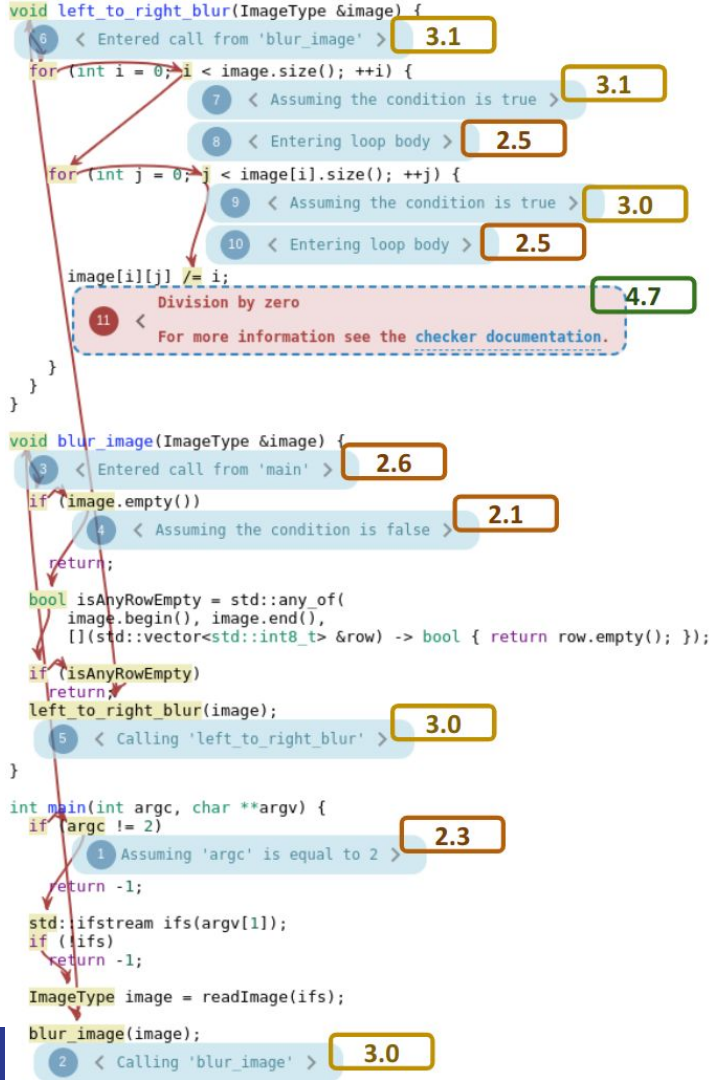
Incomprehensible (QTBase)

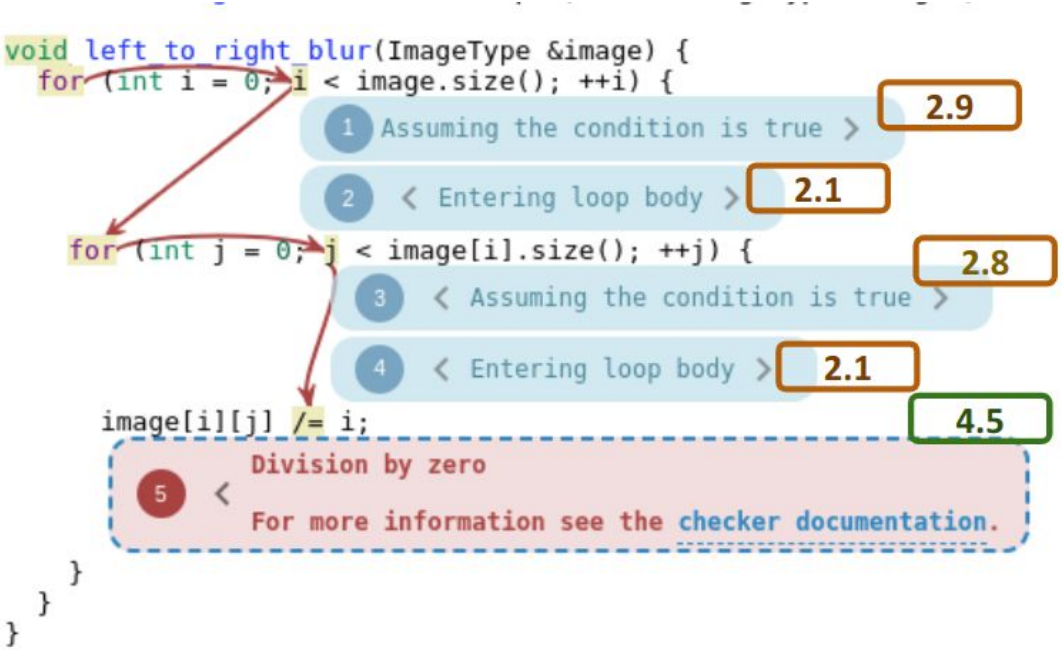


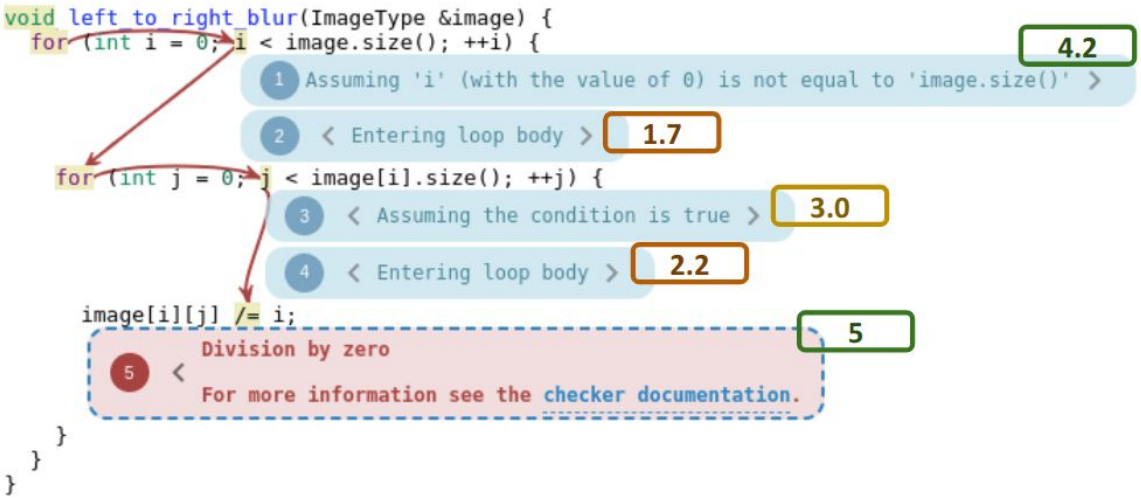
Latest survey

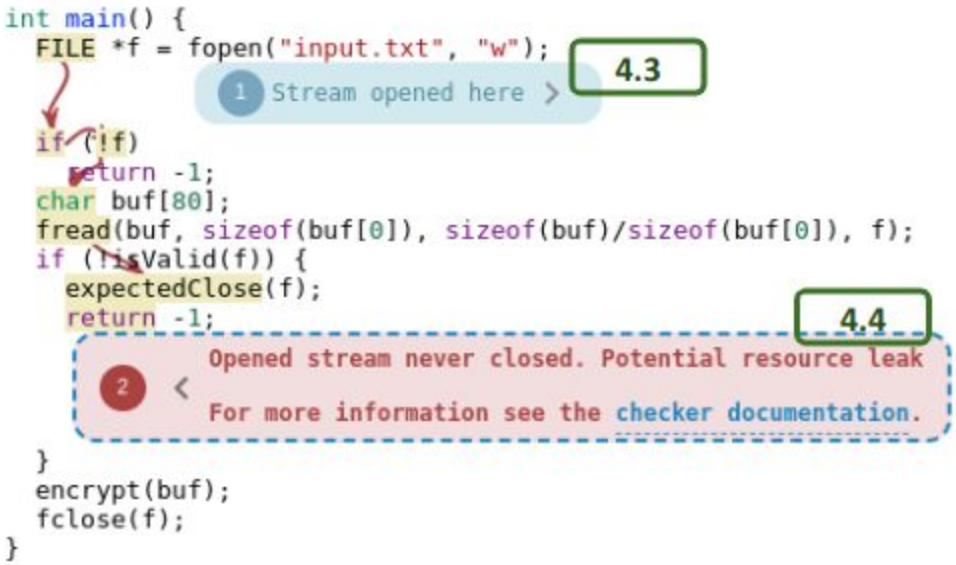
Measurement methodology

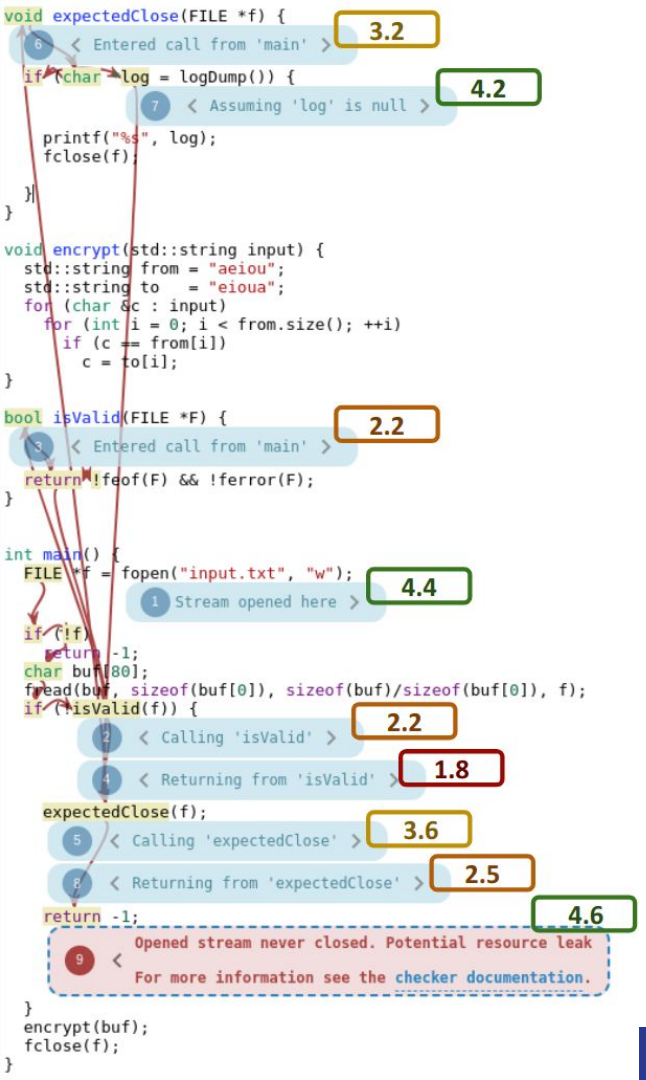
- 10 participants from the CodeChecker team
- 7 different bugs on synthetized examples
- For each bug, we generated 2-3 bug reports
  - Default
  - Verbose
  - “Ideal”
- Much less time spent untangling the source code
- Each note of the bug report needed to be ranked on a 1-5 scale

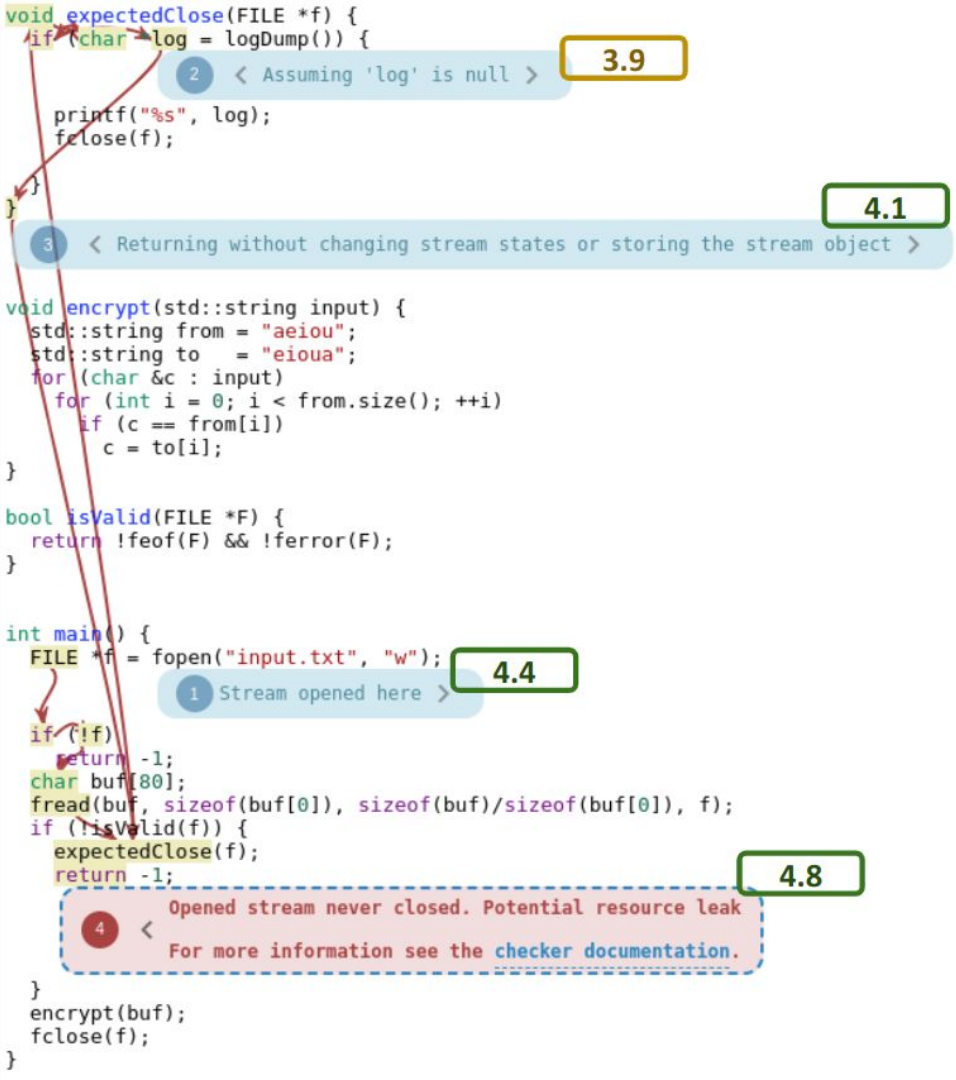

void left_to_right_blur(ImageType &image) {
6 < Entered call from 'blur_image' >
3.1
for (int i = 0; i < image.size(); ++i) {
7 < Assuming the condition is true >
3.1
8 < Entering loop body >
2.5
for (int j = 0; j < image[i].size(); ++j) {
9 < Assuming the condition is true >
3.0
10 < Entering loop body >
2.5
image[i][j] /= i;
11 < Division by zero
For more information see the checker documentation.
4.7
}
}
}
void blur_image(ImageType &image) {
3 < Entered call from 'main' >
2.6
if (image.empty())
4 < Assuming the condition is false >
2.1
return;
bool isAnyRowEmpty = std::any_of(
image.begin(), image.end(),
[](std::vector<std::int8_t> &row) -> bool { return row.empty(); });
if (isAnyRowEmpty)
return;
left_to_right_blur(image);
5 < Calling 'left_to_right_blur' >
3.0
}
int main(int argc, char **argv) {
if (argc != 2)
1 Assuming 'argc' is equal to 2 >
2.3
return -1;
std::ifstream ifs(argv[1]);
if (!ifs)
return -1;
ImageType image = readImage(ifs);
blur_image(image);
2 < Calling 'blur_image' >
3.0


void left_to_right_blur(ImageType &image) {
for (int i = 0; i < image.size(); ++i) {
1 Assuming the condition is true
2.9
2 Entering loop body
2.1
for (int j = 0; j < image[i].size(); ++j) {
2.8
3 Assuming the condition is true
4 Entering loop body
2.1
image[i][j] /= i;
4.5
5
Division by zero
For more information see the checker documentation.
}
}
}


void left_to_right_blur(ImageType &image) {
for (int i = 0; i < image.size(); ++i) {
1 Assuming 'i' (with the value of 0) is not equal to 'image.size()'
4.2
2 Entering loop body
1.7
for (int j = 0; j < image[i].size(); ++j) {
3 Assuming the condition is true
3.0
4 Entering loop body
2.2
image[i][j] /= i;
5
Division by zero
For more information see the checker documentation.
}
}
}


int main() {
FILE *f = fopen("input.txt", "w");
1 Stream opened here
4.3
if (!f)
return -1;
char buf[80];
fread(buf, sizeof(buf[0]), sizeof(buf)/sizeof(buf[0]), f);
if (!isValid(f)) {
expectedClose(f);
return -1;
4.4
2
Opened stream never closed. Potential resource leak
For more information see the checker documentation.
}
encrypt(buf);
fclose(f);
}


void expectedClose(FILE *f) {
6 < Entered call from 'main' >
3.2
if (char *log = logDump()) {
7 < Assuming 'log' is null >
4.2
printf("%s", log);
fclose(f);
}
}
void encrypt(std::string input) {
std::string from = "aeiou";
std::string to = "eioua";
for (char &c : input)
for (int i = 0; i < from.size(); ++i)
if (c == from[i])
c = to[i];
}
bool isValid(FILE *F) {
3 < Entered call from 'main' >
2.2
return !feof(F) && !ferror(F);
}
int main() {
FILE *f = fopen("input.txt", "w");
1 Stream opened here >
4.4
if (!f)
return -1;
char buf[80];
fread(buf, sizeof(buf[0]), sizeof(buf)/sizeof(buf[0]), f);
if (!isValid(f)) {
2 < Calling 'isValid' >
2.2
4 < Returning from 'isValid' >
1.8
expectedClose(f);
5 < Calling 'expectedClose' >
3.6
8 < Returning from 'expectedClose' >
2.5
return -1;
9 < Opened stream never closed. Potential resource leak
For more information see the checker documentation.
4.6
}
encrypt(buf);
fclose(f);
}


void expectedClose(FILE *f) {
if (char log = logDump()) {
2 Assuming 'log' is null
3.9
printf("%s", log);
fclose(f);
}
}
4.1
3 Returning without changing stream states or storing the stream object
void encrypt(std::string input) {
std::string from = "aeiou";
std::string to = "eioua";
for (char &c : input)
for (int i = 0; i < from.size(); ++i)
if (c == from[i])
c = to[i];
}
bool isValid(FILE *F) {
return !feof(F) && !ferror(F);
}
int main() {
FILE *f = fopen("input.txt", "w");
4.4
1 Stream opened here
if (!f)
return -1;
char buf[80];
fread(buf, sizeof(buf[0]), sizeof(buf)/sizeof(buf[0]), f);
if (!isValid(f)) {
expectedClose(f);
return -1;
4.8
4 Opened stream never closed. Potential resource leak
For more information see the checker documentation.
}
encrypt(buf);
fclose(f);
}

	Min score	Max score	Mean score
Leak "verbose" (Fig 1)	1.8	4.6	3.2
Leak "default" (Fig 2)	4.4	4.3	4.35
Leak "ideal" (Fig 3)	3.9	4.8	4.25
Map "verbose" (Fig 4)	1.8	5.0	3.2
Map "ideal" (Fig 5)	2.1	5.0	3.5
Blur "verbose" (Fig 6)	2.1	4.7	3.0
Blur "shortened" (Fig 7)	2.1	4.5	2.8
Blur "ideal" (Fig 8)	1.7	5.1	3.0
Complex "verbose" (Fig 9)	1.2	4.9	3.1
Complex "ideal" (Fig 10)	2.5	4.8	3.45

Table 1. A summary of the scores from our survey.

Conclusion

- Complex bugs -> complex analyses
- Complex analyses -> complex bug report generation
- Bug reports are the main interface in between the user and the analyzer
- In order to measurably improve bug reports, we need to measure quality
- 3 surveys:
  - Human experiments won't work unless the survey is very well thought out
  - Manual inspection can be a good basis for creating a well thought out survey
  - Make simple, targeted questions, ask for simple responses