Goanna Static Analysis Tool at SATE

Ansgar Fehnker

ansgar.fehnker@RedLizards.com

www.RedLizards.com

About Us

- R&D spin-out
- 5 years technology research
- Funded and backed by NICTA

NICTA

What We Do

Goanna Static Analysis for C/C++

Inspects code automatically for

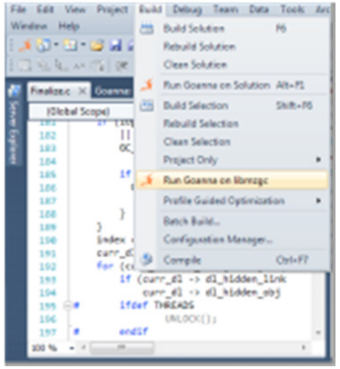
- memory corruption and leaks
- software quality issues
- security vulnerabilities
- API rule violation
- coding standards violations
- identifies >100 types of serious defects

Does not execute, but investigate code.

Products

Goanna Studio

Visual Studio PARTNER



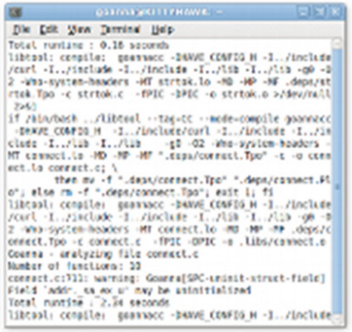
IDE integrated static analysis

- Visual Studio 2005-2010 on Windows
- Eclipse on Linux

Goanna Central

Server / command line version

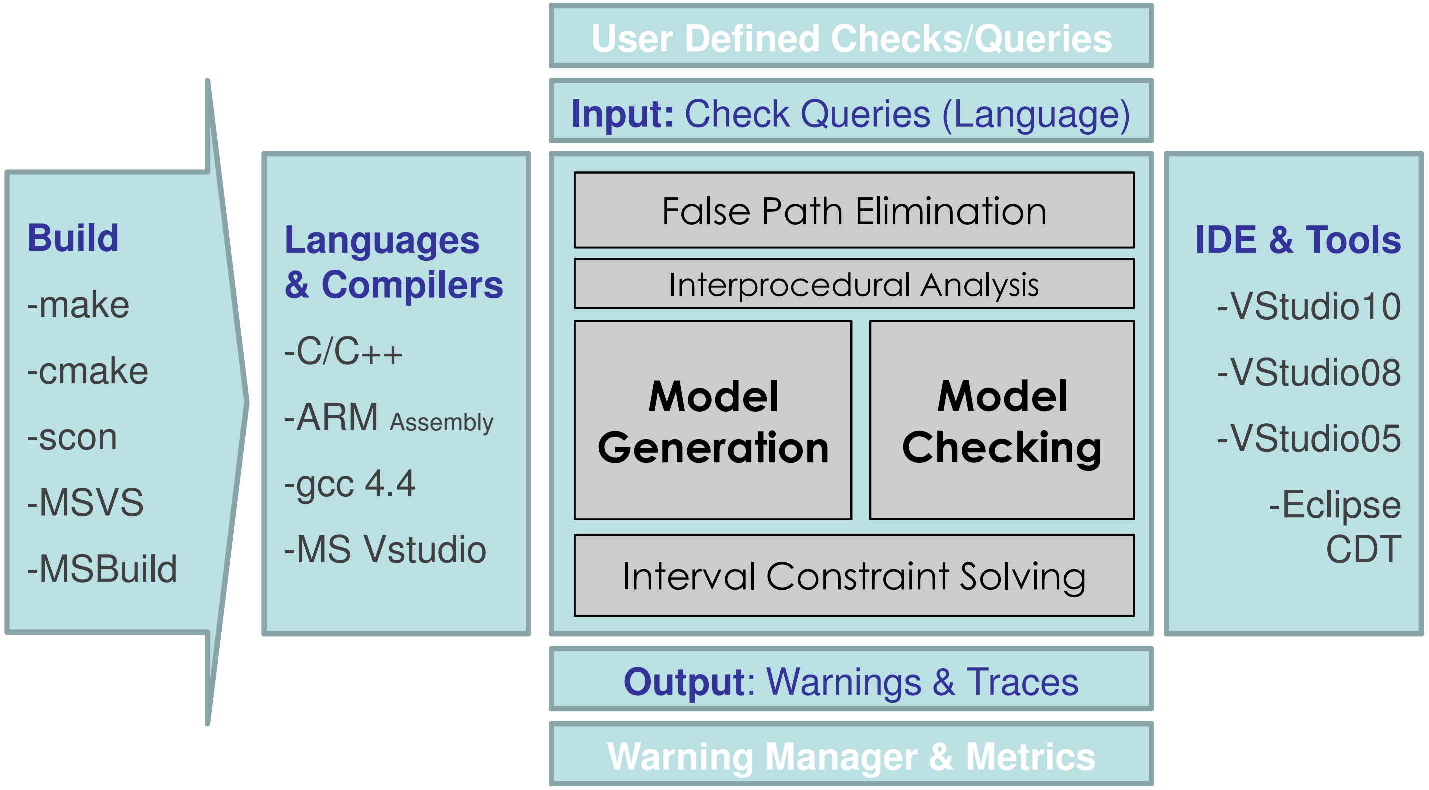
- Linux
- Windows/MSBuild (beta)



```
Total runtime : 0.16 seconds
libtool: compile:  goannacc -DHAVE_CONFIG_H -I../include
/curl -I../include -I../include -I../lib -I../lib -g0 -O
2 -Wno-system-headers -MT strtok.lo -MD -MP -MF .deps/st
rtok.Tpo -c strtok.c  -fPIC -DPIC -o strtok.o >/dev/null
 2>&1
if /bin/bash ../libtool --tag=CC --mode=compile goannacc
 -DHAVE_CONFIG_H  -I../include/curl -I../include -I../in
clude -I../lib -I../lib    -g0 -O2 -Wno-system-headers -
MT connect.lo -MD -MP -MF ".deps/connect.Tpo" -c -o conn
ect.lo connect.c; \
        then mv -f ".deps/connect.Tpo" ".deps/connect.Pl
o"; else rm -f ".deps/connect.Tpo"; exit 1; fi
libtool: compile:  goannacc -DHAVE_CONFIG_H -I../include
/curl -I../include -I../include -I../lib -I../lib -g0 -O
2 -Wno-system-headers -MT connect.lo -MD -MP -MF .deps/c
onnect.Tpo -c connect.c  -fPIC -DPIC -o .libs/connect.o
Goanna - analyzing file connect.c
Number of functions: 33
connect.c:711: warning: Goanna[SPC-uninit-struct-field]
Field `addr._sa_ex_u' may be uninitialized
Total runtime : 2.34 seconds
libtool: compile:  goannacc -DHAVE_CONFIG_H -I../include
```

Under The Hood

Goanna Architecture



User Defined Checks/Queries

Input: Check Queries (Language)

Build

- -make
- -cmake
- -scon
- -MSVS
- -MSBuild

Languages & Compilers

- -C/C++
- -ARM Assembly
- -gcc 4.4
- -MS Vstudio

False Path Elimination

Interprocedural Analysis

Model Generation

Model Checking

Interval Constraint Solving

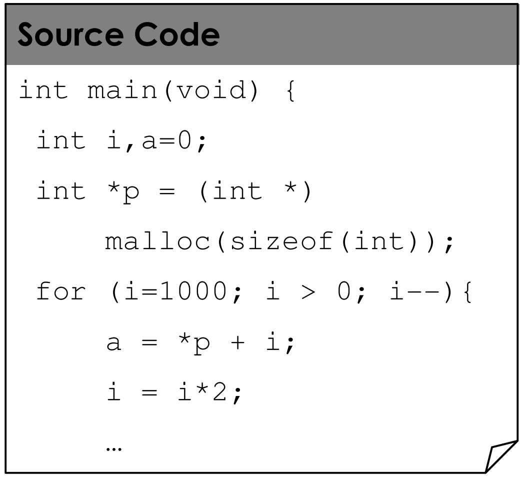
IDE & Tools

- -VStudio10
- -VStudio08
- -VStudio05
- -Eclipse CDT

Output: Warnings & Traces

Warning Manager & Metrics

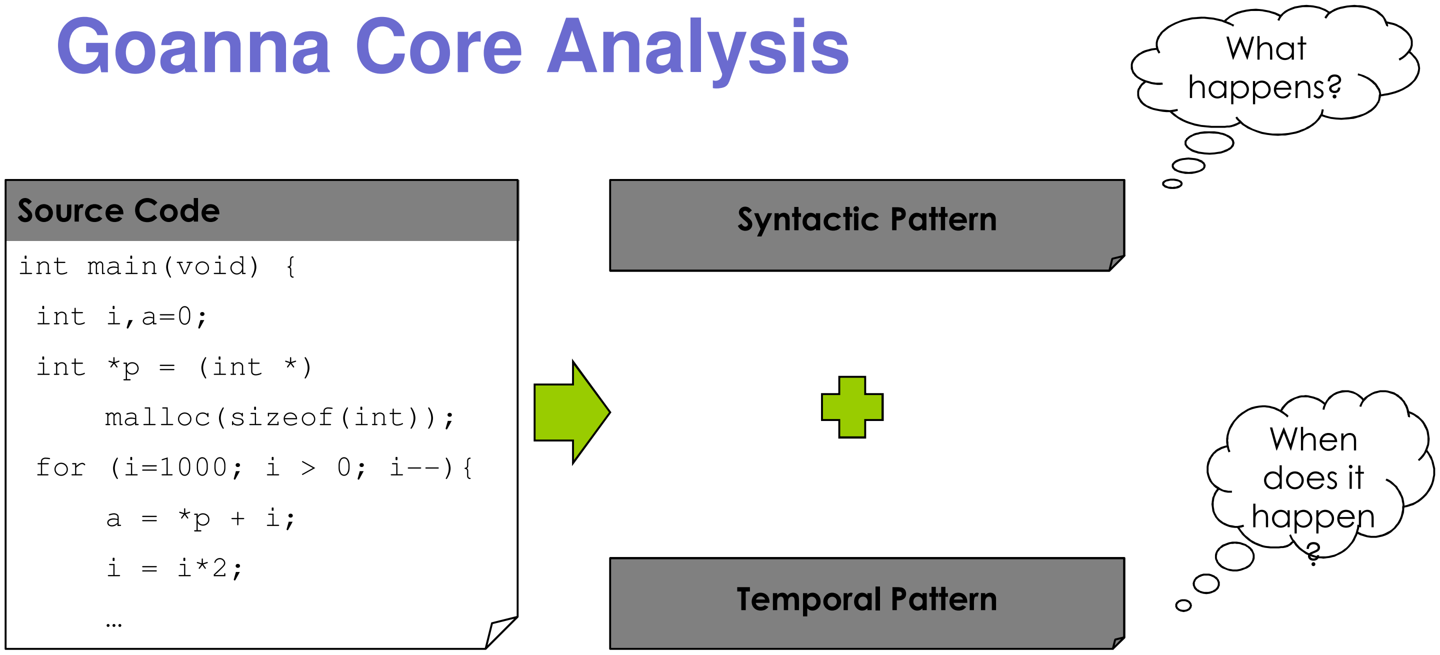
Goanna Core Analysis



Source Code

```
int main(void) {
 int i,a=0;
 int *p = (int *)
     malloc(sizeof(int));
 for (i=1000; i > 0; i--){
     a = *p + i;
     i = i*2;
     …
```

Goanna Core Analysis



What happens?

Source Code

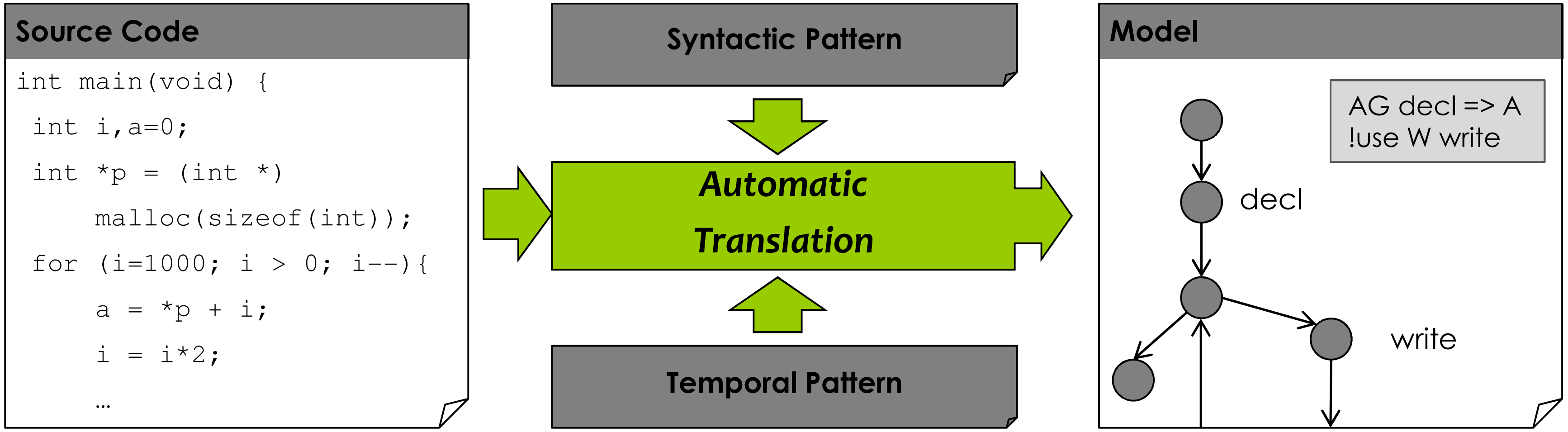
```
int main(void) {
 int i,a=0;
 int *p = (int *)
     malloc(sizeof(int));
 for (i=1000; i > 0; i--){
     a = *p + i;
     i = i*2;
     …
```

Syntactic Pattern

Temporal Pattern

When does it happen?

Goanna Core Analysis



Source Code

```
int main(void) {
 int i,a=0;
 int *p = (int *)
     malloc(sizeof(int));
 for (i=1000; i > 0; i--){
     a = *p + i;
     i = i*2;
     …
```

Syntactic Pattern

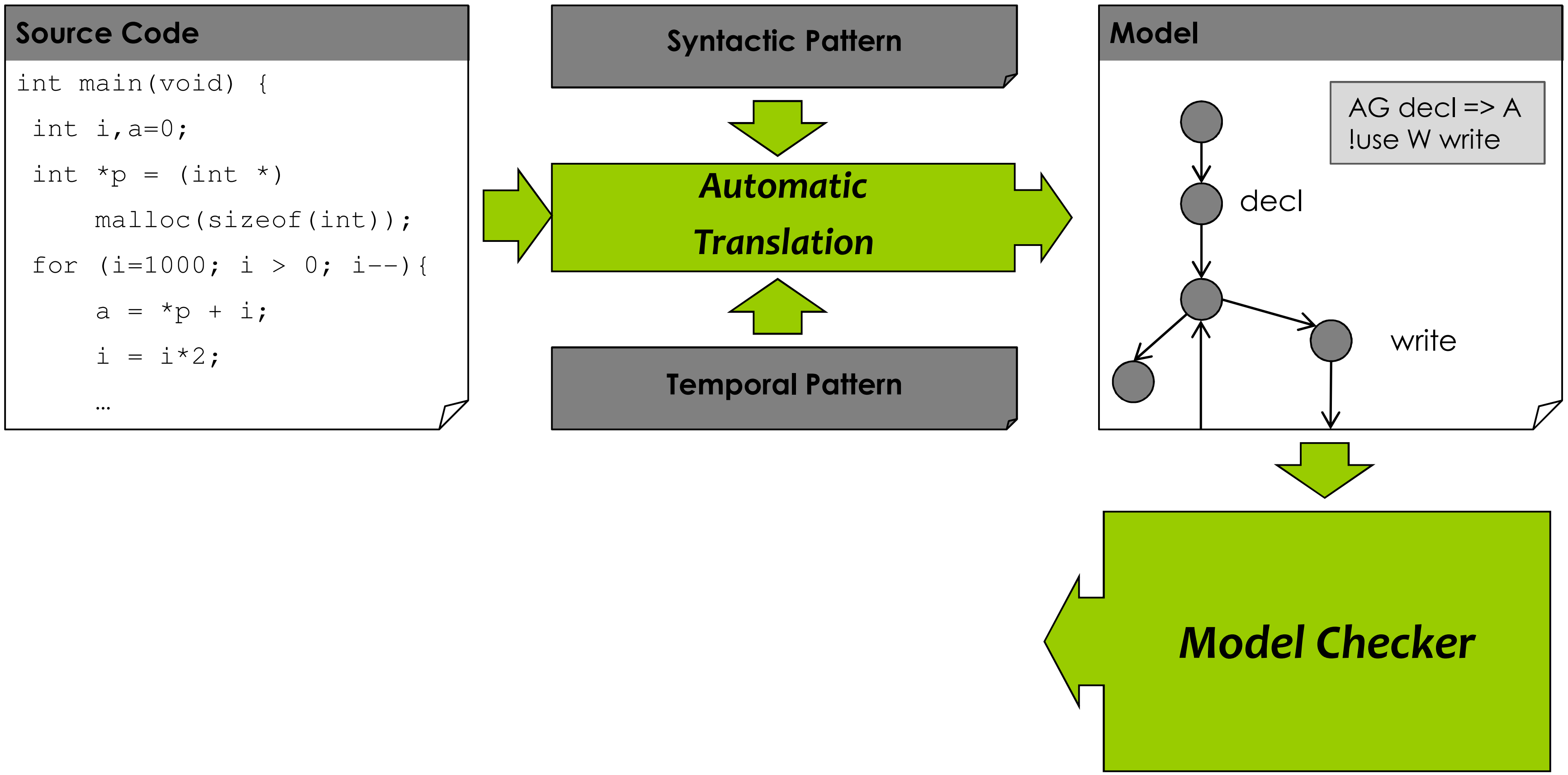
Automatic Translation

Temporal Pattern

Model

AG decl => A
!use W write

Goanna Core Analysis



Source Code

```
int main(void) {
 int i,a=0;
 int *p = (int *)
     malloc(sizeof(int));
 for (i=1000; i > 0; i--){
     a = *p + i;
     i = i*2;
     …
```

Syntactic Pattern

Automatic Translation

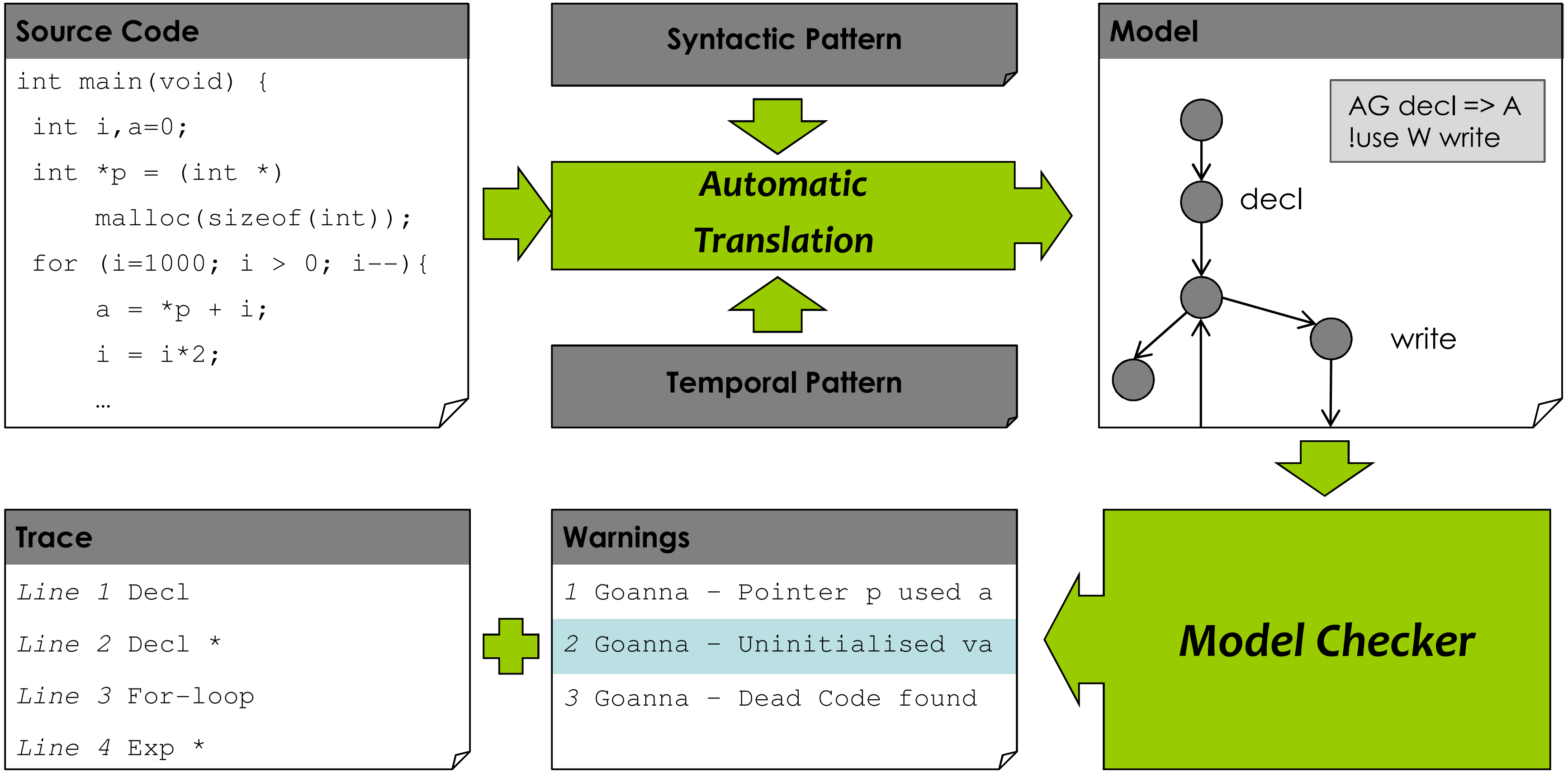
Temporal Pattern

Model

AG decl => A
!use W write

Model Checker

Goanna Core Analysis



Source Code

```
int main(void) {
 int i,a=0;
 int *p = (int *)
     malloc(sizeof(int));
 for (i=1000; i > 0; i--){
     a = *p + i;
     i = i*2;
     …
```

Syntactic Pattern

Automatic Translation

Temporal Pattern

Model

AG decl => A !use W write

Model Checker

Warnings

```
1 Goanna - Pointer p used a
2 Goanna - Uninitialised va
3 Goanna - Dead Code found
```

Trace

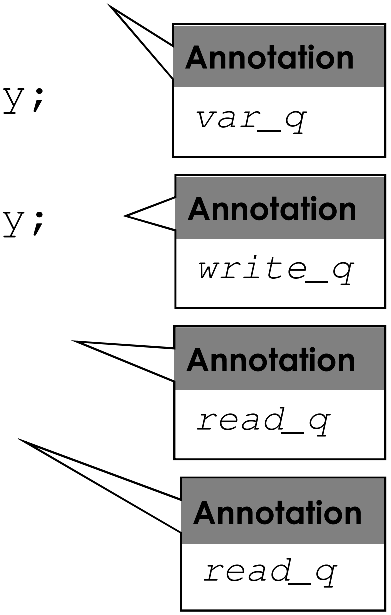
```
Line 1 Decl
Line 2 Decl *
Line 3 For-loop
Line 4 Exp *
```

Example: Uninitialized Variable

```
int foo(int n) {
 int x = 0, y = 1, q, i = 0;
 do {
      int oldy = y;
      y = x;
      q = x + oldy;
      x = q;
      i++;
  } while(i < n);
  return q;
}
```

Example: Uninitialized Variable

```
int foo(int n) {
 int x = 0, y = 1, q, i = 0;
 do {
      int oldy = y;
      y = x;
      q = x + oldy;
      x = q;
      i++;
  } while(i < n);
  return q;
}
```



Annotation

var_q

Annotation

write_q

Annotation

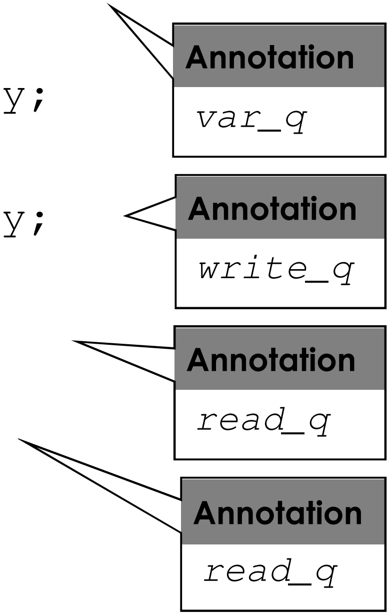
read_q

Annotation

read_q

Example: Uninitialized Variable

```
int foo(int n) {
 int x = 0, y = 1, q, i = 0;
 do {
      int oldy = y;
      y = x;
      q = x + oldy;
      x = q;
      i++;
  } while(i < n);
  return q;
}
```



Annotation

var_q

Annotation

write_q

Annotation

read_q

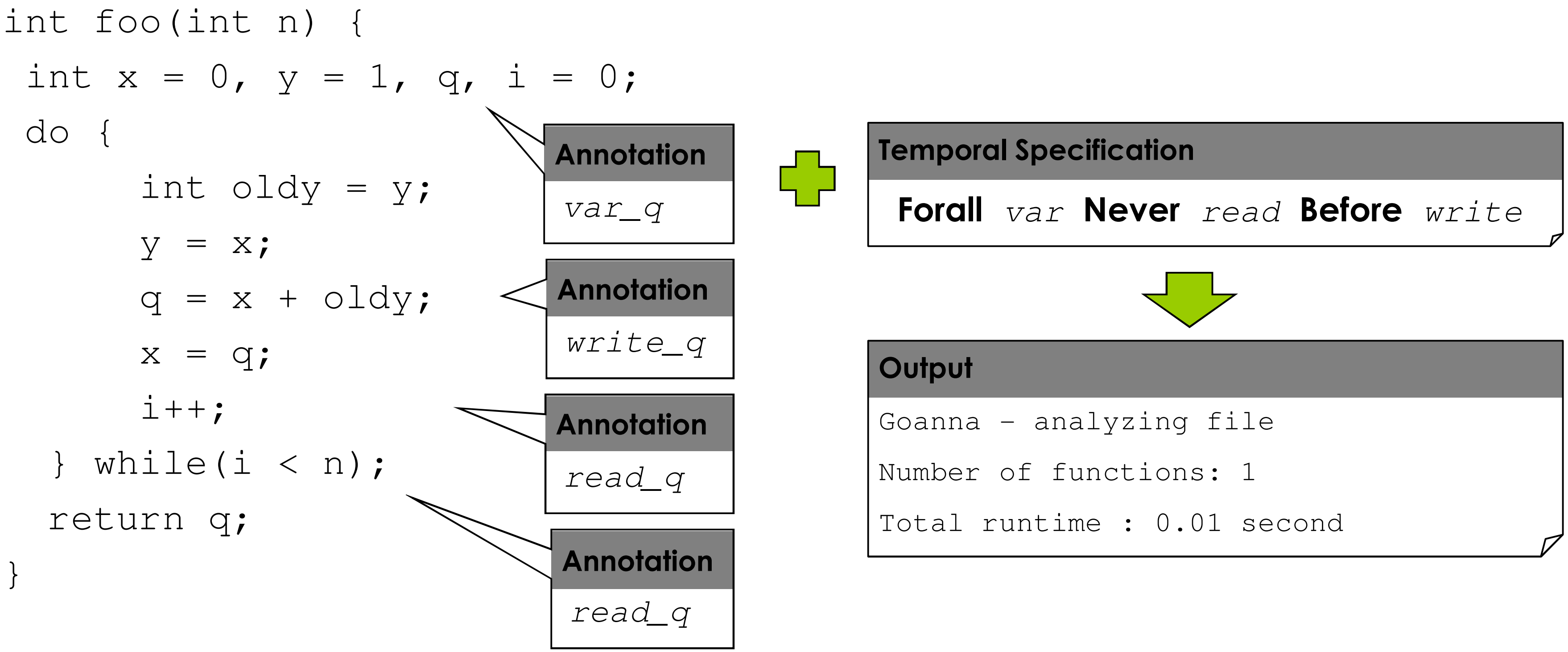
Annotation

read_q

Temporal Specification

Forall *var* **Never** *read* **Before** *write*

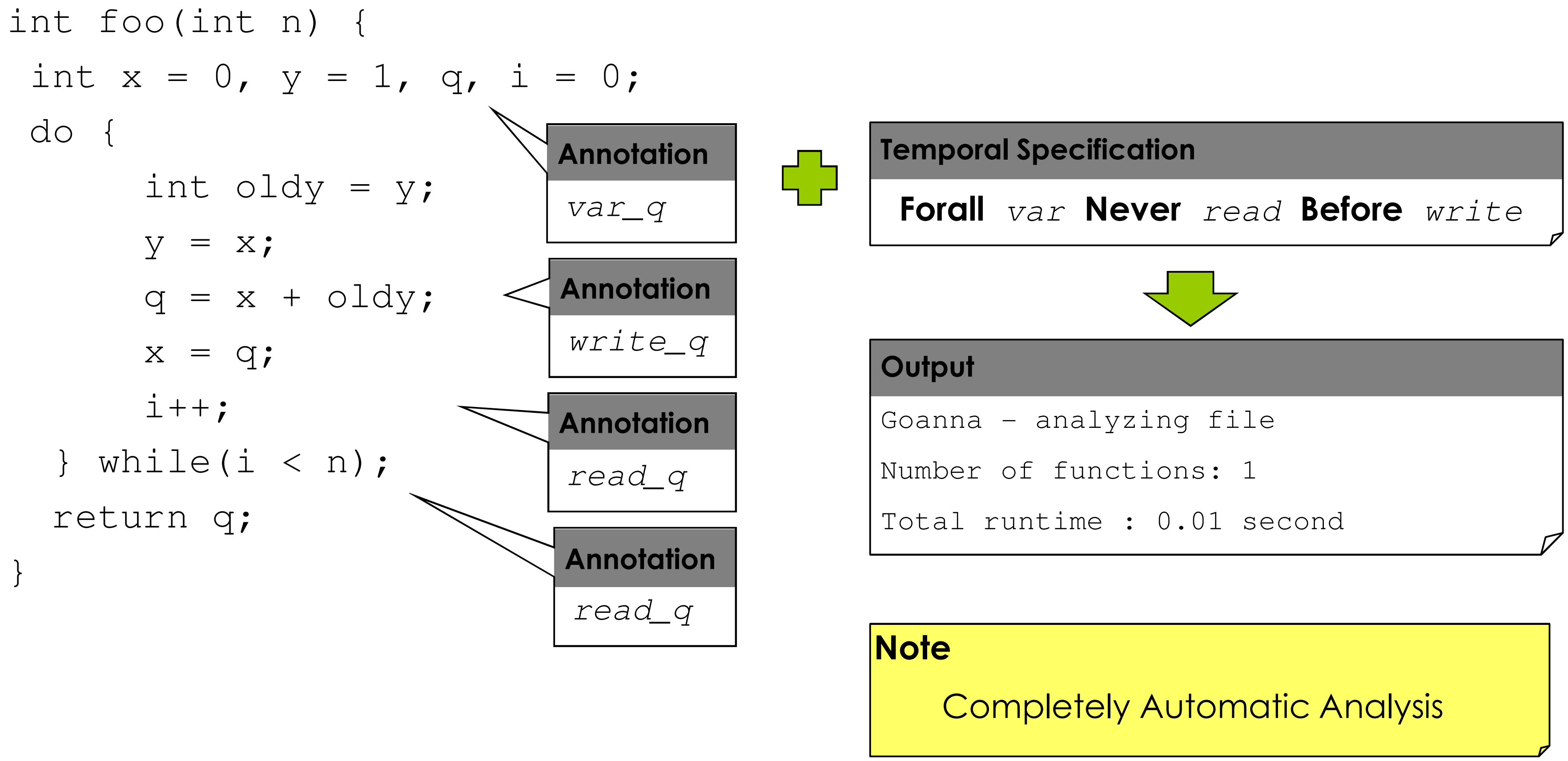
Example: Uninitialized Variable



```
int foo(int n) {
 int x = 0, y = 1, q, i = 0;
 do {
      int oldy = y;
      y = x;
      q = x + oldy;
      x = q;
      i++;
  } while(i < n);
  return q;
}
```

```
Goanna - analyzing file
Number of functions: 1
Total runtime : 0.01 second
```

Example: Uninitialized Variable



```
int foo(int n) {
 int x = 0, y = 1, q, i = 0;
 do {
      int oldy = y;
      y = x;
      q = x + oldy;
      x = q;
      i++;
  } while(i < n);
  return q;
}
```

Annotation

var_q

Annotation

write_q

Annotation

read_q

Annotation

read_q

Temporal Specification

Forall *var* **Never** *read* **Before** *write*

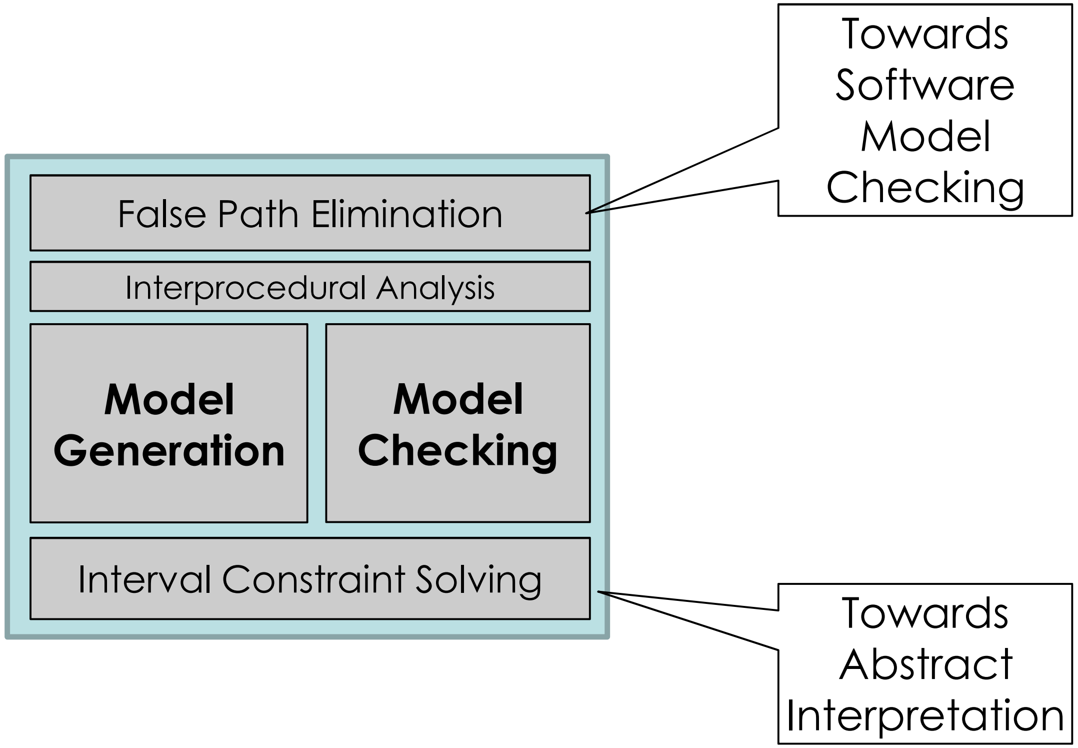
Output

```
Goanna - analyzing file
Number of functions: 1
Total runtime : 0.01 second
```

Note

Completely Automatic Analysis

Goanna Core


False Path Elimination
Interprocedural Analysis
Model Generation
Model Checking
Interval Constraint Solving
Towards Software Model Checking
Towards Abstract Interpretation

Goanna in SATE

Goanna setup for SATE

55 default checks for C/C++

- Geared towards quality issues
- Targeted at “Must Fix” and “Fix if time” issues.
- Omitted checks for “cosmetic issues”

```
ARR-inv-index
ARR-inv-index-pos
ARR-neg-index
ATH-cmp-unsign-neg
ATH-cmp-unsign-pos
ATH-div-0
ATH-div-0-aft-assign
ATH-div-0-aft-cmp
ATH-div-0-bef-cmp
ATH-div-0-interval
ATH-inc-bool
ATH-neg-check-nonneg
ATH-sizeof-by-sizeof
COP-assign-op-ret
COP-assign-op-self
COP-init-order
CPU-ctor-call-virt
CPU-dtor-call-virt
CPU-malloc-class
CPU-nonvirt-dtor
EXP-dangling-else
EXP-main-ret-int
FPT-arith
FPT-misuse
LIB-return-leak
MEM-delete-op
MEM-double-free
MEM-free-variable
MEM-malloc-arith
MEM-malloc-sizeof
MEM-malloc-sizeof-ptr
MEM-stack-global
MEM-stack-param
MEM-stack-param-ref
MEM-use-free-all
PTR-null-assign
PTR-null-assign-pos
PTR-null-cmp-aft
PTR-null-pos-assign
PTR-param-unchk-some
RED-case-reach
RED-cmp-always
RED-cmp-never
RED-const-assign-cond
RED-unused-val-ptr
RED-unused-var-all
SEM-const-call
SEM-const-global
SEM-pure-call
SEM-pure-global
SPC-order
SPC-uninit-struct
SPC-uninit-struct-field
SPC-uninit-var-all
SPC-uninit-var-some
```

Results Overall

Number of warnings

Chrome 375.54	*1079*
Chrome 375.70	*1173*
Dovecot	*180*
Wireshark 2.0	*534*
Wireshark 2.9	*532*

Top 10

PTR-null-pos-assign	952
RED-cmp-never	788
RED-cmp-always	448
PTR-null-cmp-aft	328
SPC-uninit-var-some	189
PTR-null-assign-fun-pos	144
RED-unused-val-ptr	124
PTR-param-unchk-some	94
RED-unused-var-all	67
RED-case-reach	46

PTR:	Pointer misuse
RED:	Redundant code
SPC:	Unspecified behavior

A Closer Look

SEM-const-call

unichar.c:193: warning: Goanna[SEM-const-call] Non-const function \`uint16_find' is called in const function \`uni_ucs4_to_titlecase'

```
unichar_t uni_ucs4_to_titlecase(unichar_t chr)
{  (…)
  if (!uint16_find(titlecase16_keys,
      N_ELEMENTS(titlecase16_keys), chr, &idx))
         return chr; (…)
```

- Semantic attributes are a GNU language extension
- *uni_ucs4_to_titlecase* has `__attribute__ ((const))` (see *unichar.h*)
- *uint16_find* has not
- GNU says: *“(...) a function that calls a non-const function usually must not be const“*

RED-cmp-never

director-connection.c:655: warning: Goanna[RED-cmp-never]
Comparison never holds

```
if (str_array_length(args) != 2 ||
    director_args_parse_ip_port(conn, args, &ip, &port) < 0) {
      i_error("director(%s): Invalid CONNECT args", conn->name);
      return FALSE;
}
```

- *director_args_parse_ip_port()* only returns *TRUE or FALSE*
- *director_args_parse_ip_port()<0* never true
- *ip* and *port* might not be assigned, but this failure is not detected

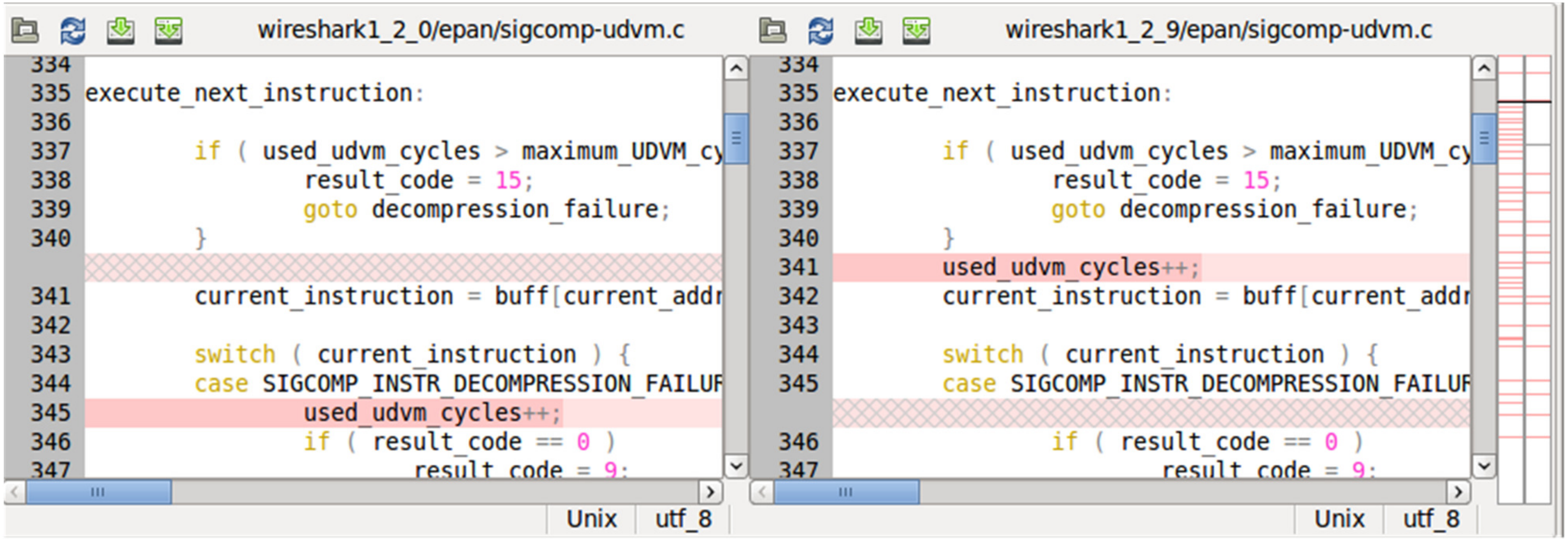
PTR-param-unchk-some

packet-smb.c:8211:64: warning: Goanna[PTR-param-unchk-some] Parameter `nti' is not checked against NULL before it is dereferenced on some paths, but on other paths it is.

```
case NT_TRANS_IOCTL: (…)
  dissect_smb2_ioctl_data(ioctl_tvb, pinfo, tree, top_tree,
  nti->ioctl_function, TRUE);
  (…)
case NT_TRANS_SSD:
  if(nti){switch(nti->fid_type){ (…)
```

- *nti* checked in one branch, but not the other
- pointer *nti* can be null and is passed to *dissect_smb2_ioctl_data()*
- related to CVE-2010-2283

Not Found: CVE-2010-2286



wireshark1_2_0/epan/sigcomp-udvm.c

```
334
335 execute_next_instruction:
336
337         if ( used_udvm_cycles > maximum_UDVM_cy
338                 result_code = 15;
339                 goto decompression_failure;
340         }

341         current_instruction = buff[current_addr
342
343         switch ( current_instruction ) {
344         case SIGCOMP_INSTR_DECOMPRESSION_FAILUR
345                 used_udvm_cycles++;
346                 if ( result_code == 0 )
347                         result_code = 9;
```

Unix utf_8

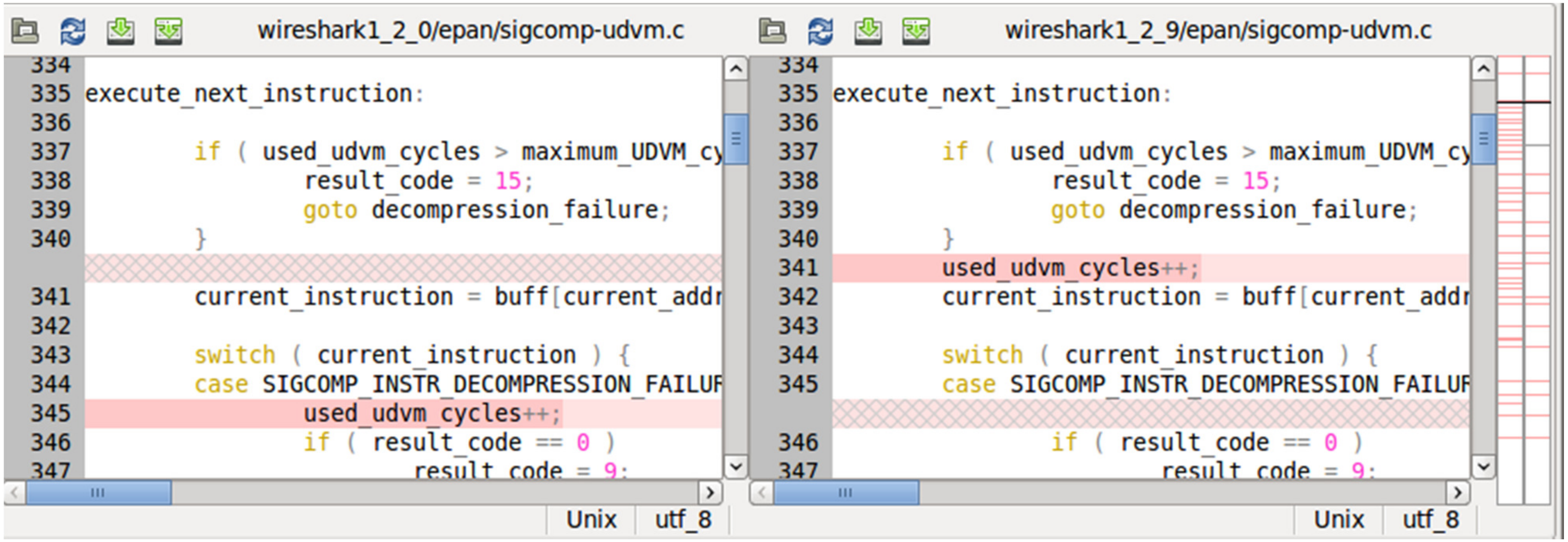
wireshark1_2_9/epan/sigcomp-udvm.c

```
334
335 execute_next_instruction:
336
337         if ( used_udvm_cycles > maximum_UDVM_cy
338                 result_code = 15;
339                 goto decompression_failure;
340         }
341         used_udvm_cycles++;
342         current_instruction = buff[current_addr
343
344         switch ( current_instruction ) {
345         case SIGCOMP_INSTR_DECOMPRESSION_FAILUR

346                 if ( result_code == 0 )
347                         result_code = 9;
```

Unix utf_8

- Label *execute_next_instruction* in line 335,
- *switch* from line 344 to 2750 with 36 cases,
- 35 *goto execute_next_instruction*
- 34 increments of *used_udvm_cycles*

Not Found: CVE-2010-2286



wireshark1_2_0/epan/sigcomp-udvm.c

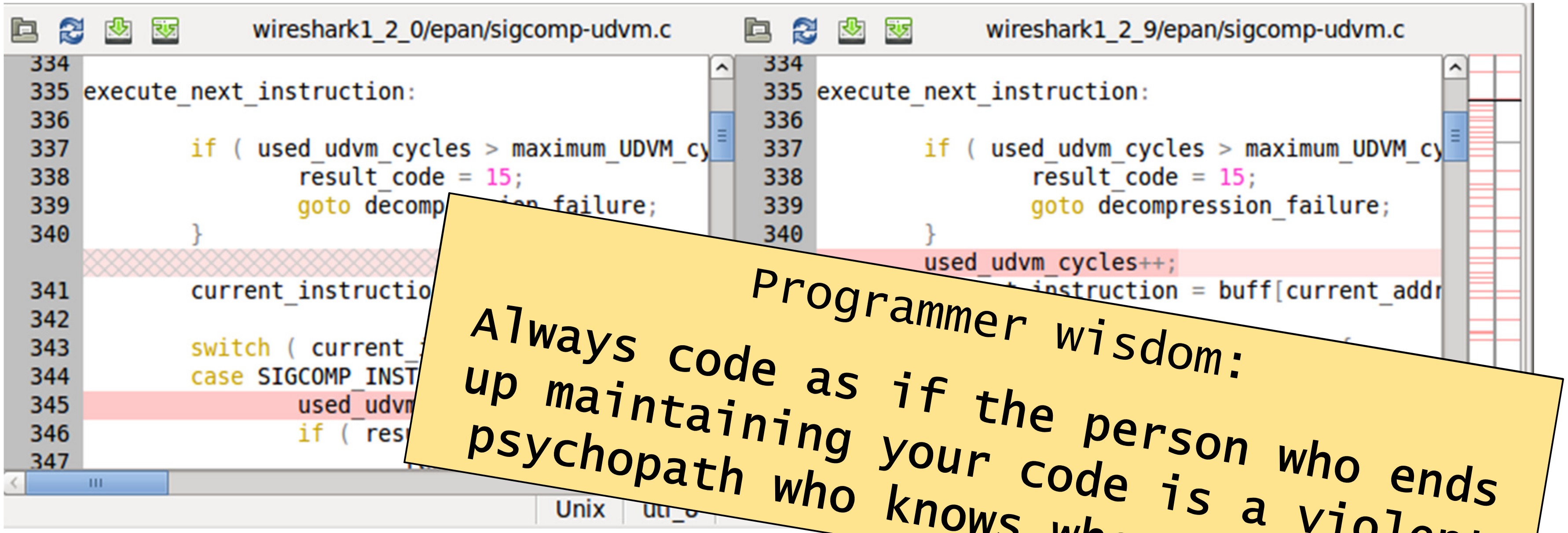
```
execute_next_instruction:

        if ( used_udvm_cycles > maximum_UDVM_cy
                result_code = 15;
                goto decompression_failure;
        }

        current_instruction = buff[current_addr

        switch ( current_instruction ) {
        case SIGCOMP_INSTR_DECOMPRESSION_FAILUR
                used_udvm_cycles++;
                if ( result_code == 0 )
                        result_code = 9;
```

wireshark1_2_9/epan/sigcomp-udvm.c

```
execute_next_instruction:

        if ( used_udvm_cycles > maximum_UDVM_cy
                result_code = 15;
                goto decompression_failure;
        }
        used_udvm_cycles++;
        current_instruction = buff[current_addr

        switch ( current_instruction ) {
        case SIGCOMP_INSTR_DECOMPRESSION_FAILUR

                if ( result_code == 0 )
                        result_code = 9;
```

- Problem: Infinite loop possible.
- Need: Show absence of loop-invariant for a goto-stucture
- Do we want to spend resources on find this?
- Or advise to use a proper for-loop.

Not Found: CVE-2010-2286


wireshark1_2_0/epan/sigcomp-udvm.c
wireshark1_2_9/epan/sigcomp-udvm.c
Unix

```
execute_next_instruction:

        if ( used_udvm_cycles > maximum_UDVM_cy
                result_code = 15;
                goto decomp      ion_failure;
        }

        current_instructio

        switch ( current_
        case SIGCOMP_INST
                used_udvm
                if ( res
```

```
execute_next_instruction:

        if ( used_udvm_cycles > maximum_UDVM_cy
                result_code = 15;
                goto decompression_failure;
        }
        used_udvm_cycles++;
                instruction = buff[current_addr
```

Programmer wisdom:
Always code as if the person who ends up maintaining your code is a violent psychopath who knows where you live.

- Problem: Infinite loop possible.
- Need: Show absence of loop-invariant for a goto-stucture
- Do we want to spend resources on find this?
- Or advise to use a proper for-loop.

Summary

- Goanna is a static analysis solution for C/C++
- Desktop and server version available at redlizards.com
- It uses a combination of model checking and static analysis to find serious bugs
- It did find serious bugs
- It is named after a bug-eating lizard