artificial intelligence research koblenz

Testing a Verification Tool

http://userpages.uni-koblenz.de/~wagnermar/

Motivation

The goal of software verification is the **proof of correctness** of a program with respect to its specification. Normally, the tools used in a verification process are not verified, and their correctness is taken for granted. One approach to check the quality of these tools without verifying them is by testing.

In the BMBF-project "Verisoft XT", the Verifying C Compiler (VCC) from Microsoft Research is used for the verification of C code. **It is almost impossible to come up with a correctness proof of VCC's sound functioning and to keep it up-to-date** because of two reasons. First, VCC is being developed during the project, which results in frequent changes of the implementation and the verification formalism. Second, the compiler is used as a black box for non-experts.

… because the tool's correctness is taken for granted.

Goal

The goal of this thesis was to develop a systematic testing process for black-box verification systems. The theoretical part consists of the application of the concepts of black-box software testing in the context of verification systems. This includes the definition of the subject under test and the definition of a suitable test metric.

This process was applied to VCC, resulting in a specialized test suite and a test harness. The test harness runs and monitors the tests without user interaction, enabling the regression testing of both VCC and an arbitrary code base.

Our Approach

… we answer

What is to be tested?
How to test?
Origins of tests?
How to **assess** the tests?

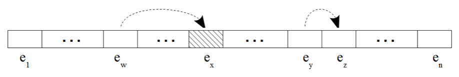
Origins of Test Cases

Source	Credibility	# of tests	Creation of tests
ISO/IEC 9899:201x	++	-	--
GNU C Compiler Collection 4.4.0	?	++	0
VCC, Spec#, Frama-C/Jessie	+	+/0/+	+

We use trustworthy sources to avoid buggy test cases, …

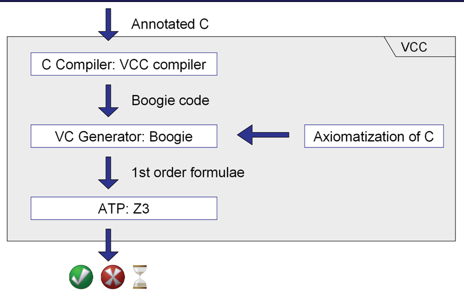
Test Assessment

Idea: determine a minimal subset of the axiomatization
(so that the expected output does not change)



Excursion Axiomatization
- Semantics of C and of specification constructs
- ~870 elements (arithmetics, memory model, …)

The verification tool under test: VCC



… and assess the tests with respect to the semantics of the C language.

Sample Test Case

Demonstrating the **iterative adaption** of the GCC test case files. The original test case file *990604-1.c* is shown on the left, its minimally annotated version in the center, and the version with some functional specification on the right. While VCC is able to verify both annotated versions, VCC's compiler returns an error for the original version because the function *abort()* is not defined in that context.

```c
// 990604-1.c original version

int b;

void f ()



{
  int i = 0;
  if (b == 0)
    do

    {
      b = i;
      i++;
    } while (i < 10);
}

int main ()




{
  f ();
  if (b != 9)
    abort ();
  return 0;
}
```

```c
// 990604-1.c minimal annotation
#include "vcc2.h"
int b;

void f ()
  writes(&b)


{
  int i = 0;
  if (b == 0)
    do

    {
      b = i;
      unchecked(i++);
    } while (i < 10);
}

int main ()
  writes(&b)



{
  f ();
  if (b != 9)
    return 1;
  return 0;
}
```

```c
// 990604-1.c incl functional spec.
#include "vcc2.h"
int b;

void f ()
  writes(&b)
  ensures(old(b) == 0 ==> b == 9)
  ensures(old(b) != 0 ==> b == old(b))
{
  int i = 0;
  if (b == 0)
    do
    invariant(0<=i && i<10)
    {
      b = i;
      i++;
    } while (i < 10) ;
}

int main ()
  writes(&b)
  ensures(old(b) == 0 ==> result == 0)
  ensures(old(b) != 0 && old(b) != 9 ==>
                                 result == 1)
{
  f ();
  if (b != 9)
    return 1;
  return 0;
}
```

Test Results

VCC		*Type*	*Const*	*Func.*	*Axiom*	*Var*	*Proc.*	*Total*
2.1.20731.0	total	19	43	387	378	1	30	**858**
	cov.	18	33	307	186	1	30	**575**
	%	95	77	79	49	100	100	**67**
2.1.20908.0	total	19	43	417	384	1	32	**896**
	cov.	18	32	292	139	1	27	**509**
	%	95	74	70	36	100	84	**57**

Two conclusions…

(1) The used part of the axiomatization seems to **correctly reflect the developers' assumptions** about how the verification methodology is supposed to work,
(2) **additional test cases** should be written to achieve a higher coverage.

Five Issues found…

Bug in axiomatization
VCC **deviated** from the C Standard
Tool related issues: VCC, Boogie, Z3 Model Viewer
(how many issues still hiding?)

artificial intelligence research koblenz
UNIVERSITÄT KOBLENZ · LANDAU

Markus Wagner
Department of Computer Science, University of Koblenz, Germany
wagnermar@uni-koblenz.de