Are We There Yet?
20 Years of Formal Verification in Critical Software

Roderick Chapman
Principal Engineer, Altran UK
Agenda

• Our world…
• An opening thought…
• So why Formalize?
• Examples of FV in Software
• Encouraging signs…
• Homework
• A Closing Thought…
• Our world…
• An opening thought…
• So why Formalize?
• Examples of FV in Software
• Encouraging signs…
• Homework
• A Closing Thought…
Our World – Critical Software

No defects please!
Agenda

• Our world...
• An opening thought...
• So why Formalize?
• Examples of FV in Software
• Encouraging signs...
• Homework
• A Closing Thought...
Every software project uses Formal Methods...

Professor Martyn Thomas CBE
A Opening Thought...

Huh?
Agenda

• Our world...
• An opening thought…
• So why Formalize?
• Examples of FV in Software
• Encouraging signs…
• Homework
• A Closing Thought…
Why Bother with Formal Methods?

\( \Delta IDStation; \Delta RealWorld | \)
\( TISOpThenUpdate \)
\( \land \ latched = \text{locked} \land \ latched' = \text{unlocked} \)
\( \models \)
\( (\exists \ \text{ValidToken} \bullet \ \text{goodT}(\theta \text{ValidToken}) = \text{currentUserTok} \land \text{UserTokenOKNoCurrencyCheck} \land \text{FingerOK}) \)
\( \lor \)
\( (\exists \ \text{TokenWithValidAuth} \bullet \ \text{goodT}(\theta \text{TokenWithValidAuth}) \land \text{UserTokenWithOKAuthCertNoCurrencyCheck}) \)
\( \lor \)
\( (\exists \ \text{ValidToken} \bullet \ \text{goodT}(\theta \text{ValidToken}) = \text{currentAdminTok} \land \text{authCert} \neq \emptyset \land (\text{the authCert}).role = \text{guard}) \)

See: TISOpThenUpdate (p. 5), UserTokenOKNoCurrencyCheck (p. 5), UserTokenWithOKAuthCertNoCurrencyCheck (p. 5)
Why Bother with Formal Methods?
Thinking and Tooling exposes... 

Ambiguity...
Thinking and Tooling exposes...

Contradiction...
Thinking and Tooling exposes...

Incompleteness...
Thinking and Tooling enables...

#include "customer_call.h";
The Catch...

- What’s stopping us?
  - Fear of maths?
  - Snake-oil?
  - Oversold promises in the 1980s?
  - Something else?

- “Software is a fashion industry with delusions of grandeur”

  Prof. Les Hatton, Author of “Safer C”
The Catch...

• Big problem – most “notations” that we deal with (*mostly* Programming Languages)...
  • Are not *formal* or *unambiguous*...
  • Are poorly defined...
  • Contain hard-to-avoid features that are intrinsically hostile to *sound* and *fast* formal verification.
    • e.g. undefined behaviour, pointers, un-disciplined use of concurrency etc.
• Consider the following code in C or C++

```c
int i;
int a[10];
i = ... ; /* initialize i */
a = { ... }; /* initialize a fully */

i = a[i++]; /* ????????? */
```
Agenda

- Our world…
- An opening thought…
- So why Formalize?
- Examples of FV in Software
- Encouraging signs…
- Homework
- A Closing Thought…
Formal Languages?

• So are there any “properly formal” programming languages?
• Yes! An incomplete list…
  • OCAML
  • Scheme
  • SPARK (the Ada subset, not Apache–SPARK)
  • Eiffel
  • JVM Bytecode
  • All machine code (e.g. ARM ISA)
Formal Languages?

• What if we include specification or “modelling” languages? OK… then…
  • SCADE (Lustre)
  • B and Event–B
  • CSP
  • Subset(s) of MATLAB/Simulink
  • Escher Perfect
  • ..and many more…
Formally Verified Software?

• Here are some examples of systems using formal software verification…
Agenda

• Our world…
• An opening thought…
• So why Formalize?
• Examples of FV in Software

• Encouraging signs…
• Homework
• A Closing Thought…
Formal Methods tend to “disappear” as they become accepted... to the point where you don’t even know you’re using them...

For example...

- Compiler optimization.
- Basic Static Code Analysis/Verification.
- Bounded Model Checking and Constraint Solving for test data generation.
Encouraging Signs...

• *Security Changes Everything*...

• Against a malicious and capable attacker, a “test it lots” verification approach will *never* be good enough.

• *Finally*, people at realizing that *sound* formal verification can deal with this, because a sound verification *prevents all the bugs*...
Agenda

• Our world…
• An opening thought…
• So why Formalize?
• Examples of FV in Software
• Encouraging signs…
• Homework
• A Closing Thought…
Homework Assignment 1...
Homework Assignment 2...

- Check out www.fbinfer.com

- *Infer* – facebook’s static code analysis tool for C/C++/Java/Objective-C
• Our world…
• An opening thought…
• So why Formalize?
• Examples of FV in Software
• Encouraging signs…
• Homework

• A Closing Thought…
A Closing Thought...

Every software project uses Formal Methods...

Professor Martyn Thomas CBE
Here’s a Formal Language that you *all* use

```assembly
MOV R0, #1024
MOV R1, #0
MOV R2, #0x19000000
ORR R2, R2, #0x00990000
ORR R2, R2, #0x00009900
ORR R2, R2, #0x0000009A
MOV R3, #10
loop
UMULL R4, R5, R0, R2
UMULL R4, R6, R5, R3
SUB R4, R0, R4
ADD R1, R1, R4
MOVS R0, R5
BNE loop
```
A Closing Thought...

The big question is not *if* to use Formal Methods, but *when to start*...
References

- R. Chapman and F. Schanda
  “Are we there yet? 20 years of industrial theorem proving with SPARK.”

- Full details, data, and references for SHOLIS, C130J, Tokeneer, iFACTS projects, and how SPARK developed over the years.

- PDF available from me: 
  rod.chapman@altran.com
Questions...