

What are Formal Methods for?

- ▶ Everything!
- ▶ Design
 - ▶ specification
 - ▶ animation
- ▶ Verification
 - ▶ finding bugs
 - ▶ proof of correctness
- ▶ Implementation
 - ▶ formal-methods based implementation languages
 - ▶ verifying/verified compilers

What are Formal Methods for?

- ▶ **Everything!**
- ▶ Design
 - ▶ specification
 - ▶ animation
- ▶ Verification
 - ▶ finding bugs
 - ▶ proof of correctness
- ▶ Implementation
 - ▶ formal-methods based implementation languages
 - ▶ verifying/verified compilers

What are Formal Methods for?

- ▶ Everything!
- ▶ Design
 - ▶ specification
 - ▶ animation
- ▶ Verification
 - ▶ finding bugs
 - ▶ proof of correctness
- ▶ Implementation
 - ▶ formal-methods based implementation languages
 - ▶ verifying/verified compilers

The Demise of Formal Methods

- ▶ Shankar's premise on the demise of formal methods:
 - ▶ out of the mainstream of computing
 - ▶ best students no longer flocking to it
 - ▶ not connected to mainstream
 - ▶ only modest industrial uptake
- ▶ Formal methods have a different flavour in Europe
 - ▶ USA:
 - ▶ static analysis, property-checking
 - ▶ SMT, model checking
 - ▶ Europe:
 - ▶ formal design: Z, VDM, B, SPARK
 - ▶ interactive theorem proving: various HOLs, Coq
 - ▶ Not clearcut: Mona, FDR (Europe), PVS, Nuprl, TLA (USA)
- ▶ Europeans not so worried about health of formal methods

The Demise of Formal Methods

- ▶ Shankar's premise on the demise of formal methods:
 - ▶ out of the mainstream of computing
 - ▶ best students no longer flocking to it
 - ▶ not connected to mainstream
 - ▶ only modest industrial uptake
- ▶ Formal methods have a different flavour in Europe
 - ▶ USA:
 - ▶ static analysis, property-checking
 - ▶ SMT, model checking
 - ▶ Europe:
 - ▶ formal design: Z, VDM, B, SPARK
 - ▶ interactive theorem proving: various HOLs, Coq
 - ▶ Not clearcut: Mona, FDR (Europe), PVS, Nuprl, TLA (USA)
- ▶ Europeans not so worried about health of formal methods

The Demise of Formal Methods

- ▶ Shankar's premise on the demise of formal methods:
 - ▶ out of the mainstream of computing
 - ▶ best students no longer flocking to it
 - ▶ not connected to mainstream
 - ▶ only modest industrial uptake
- ▶ Formal methods have a different flavour in Europe
 - ▶ USA:
 - ▶ static analysis, property-checking
 - ▶ SMT, model checking
 - ▶ Europe:
 - ▶ formal design: Z, VDM, B, SPARK
 - ▶ interactive theorem proving: various HOLs, Coq
 - ▶ Not clearcut: Mona, FDR (Europe), PVS, Nuprl, TLA (USA)
- ▶ Europeans not so worried about health of formal methods

The Demise of Formal Methods

- ▶ Shankar's premise on the demise of formal methods:
 - ▶ out of the mainstream of computing
 - ▶ best students no longer flocking to it
 - ▶ not connected to mainstream
 - ▶ only modest industrial uptake
- ▶ Formal methods have a different flavour in Europe
 - ▶ USA:
 - ▶ static analysis, property-checking
 - ▶ SMT, model checking
 - ▶ Europe:
 - ▶ formal design: Z, VDM, B, SPARK
 - ▶ interactive theorem proving: various HOLs, Coq
 - ▶ Not clearcut: Mona, FDR (Europe), PVS, Nuprl, TLA (USA)
- ▶ Europeans not so worried about health of formal methods

The Demise of Formal Methods

- ▶ Shankar's premise on the demise of formal methods:
 - ▶ out of the mainstream of computing
 - ▶ best students no longer flocking to it
 - ▶ not connected to mainstream
 - ▶ only modest industrial uptake
- ▶ Formal methods have a different flavour in Europe
 - ▶ USA:
 - ▶ static analysis, property-checking
 - ▶ SMT, model checking
 - ▶ Europe:
 - ▶ formal design: Z, VDM, B, SPARK
 - ▶ interactive theorem proving: various HOLs, Coq
 - ▶ Not clearcut: Mona, FDR (Europe), PVS, Nuprl, TLA (USA)
- ▶ Europeans not so worried about health of formal methods

The Demise of Formal Methods

- ▶ Shankar's premise on the demise of formal methods:
 - ▶ out of the mainstream of computing
 - ▶ best students no longer flocking to it
 - ▶ not connected to mainstream
 - ▶ only modest industrial uptake
- ▶ Formal methods have a different flavour in Europe
 - ▶ USA:
 - ▶ static analysis, property-checking
 - ▶ SMT, model checking
 - ▶ Europe:
 - ▶ formal design: Z, VDM, B, SPARK
 - ▶ interactive theorem proving: various HOLs, Coq
 - ▶ Not clearcut: Mona, FDR (Europe), PVS, Nuprl, TLA (USA)
- ▶ Europeans not so worried about health of formal methods

Comments from Europe

Although CS university enrollments are worryingly down, formal methods seems quite healthy:

"amount of activity is huge ...
by no means all academic ...
I don't see that a ``relaunch'' is required"

.....
"I teach two formal methods courses: one on Z, and one on unifying theories of programming.

The latter is very theoretical, but goes down brilliantly with the small number of students who sign up for it: it's an optional course.

The Z course is compulsory, and it is disliked by all but a small handful of students. I'll be teaching it again this year, entirely through case studies, as that seems the most palatable way.

Things really take off on the Z course when students get to use the tools."

Comments from Europe

Although CS university enrollments are worryingly down, formal methods seems quite healthy:

"amount of activity is huge ...
by no means all academic ...
I don't see that a ``relaunch'' is required"

.....
"I teach two formal methods courses: one on Z, and one on unifying theories of programming.

The latter is very theoretical, but goes down brilliantly with the small number of students who sign up for it: it's an optional course.

The Z course is compulsory, and it is disliked by all but a small handful of students. I'll be teaching it again this year, entirely through case studies, as that seems the most palatable way.

Things really take off on the Z course when students get to use the tools."

Are solutions to demise of CS in UK also good for FM?

- ▶ Plummeting applications to CS (even at Cambridge)
 - ▶ clever kids go into math/science/economics
 - ▶ CS perceived only for the second rate
(know this from talking to my son)
- ▶ UK response: CS is being relaunched for kids
 - ▶ attempt to change examination syllabus
 - ▶ spread information via YouTube competition
 - ▶ cool websites
- ▶ So ... **Does Formal Methods need a Relaunch?**
 - ▶ show it is a key scientific foundation
 - ▶ fun: *Secret Ninja Formal Methods*
 - ▶ exciting scientific challenges (not just verification)
 - ▶ fantastic employment potential

Are solutions to demise of CS in UK also good for FM?

- ▶ Plummeting applications to CS (even at Cambridge)
 - ▶ clever kids go into math/science/economics
 - ▶ CS perceived only for the second rate
(know this from talking to my son)
- ▶ UK response: CS is being relaunched for kids
 - ▶ attempt to change examination syllabus
 - ▶ spread information via YouTube competition
 - ▶ cool websites
- ▶ So ... **Does Formal Methods need a Relaunch?**
 - ▶ show it is a key scientific foundation
 - ▶ fun: *Secret Ninja Formal Methods*
 - ▶ exciting scientific challenges (not just verification)
 - ▶ fantastic employment potential

Are solutions to demise of CS in UK also good for FM?

- ▶ Plummeting applications to CS (even at Cambridge)
 - ▶ clever kids go into math/science/economics
 - ▶ CS perceived only for the second rate
(know this from talking to my son)
- ▶ UK response: CS is being relaunched for kids
 - ▶ attempt to change examination syllabus
 - ▶ spread information via YouTube competition
 - ▶ cool websites
- ▶ So ... Does Formal Methods need a Relaunch?
 - ▶ show it is a key scientific foundation
 - ▶ fun: *Secret Ninja Formal Methods*
 - ▶ exciting scientific challenges (not just verification)
 - ▶ fantastic employment potential

A Golden Age for Formal Methods

- ▶ New application areas
 - ▶ finance
 - ▶ biology
 - ▶ semantic web
 - ▶ saving the world
- ▶ New deduction-based tools:
 - ▶ static analysis
 - ▶ verifying synthesis
 - ▶ formal implementation languages
- ▶ Needs marketing
 - ▶ generate a 'buzz'
 - ▶ highlight awards (e.g. Turing winners)
 - ▶ influence hiring committees
 - ▶ make funders aware of potential

A Golden Age for Formal Methods

- ▶ New application areas
 - ▶ finance
 - ▶ biology
 - ▶ semantic web
 - ▶ saving the world
- ▶ New deduction-based tools:
 - ▶ static analysis
 - ▶ verifying synthesis
 - ▶ formal implementation languages
- ▶ Needs marketing
 - ▶ generate a 'buzz'
 - ▶ highlight awards (e.g. Turing winners)
 - ▶ influence hiring committees
 - ▶ make funders aware of potential

A Golden Age for Formal Methods

- ▶ New application areas
 - ▶ finance
 - ▶ biology
 - ▶ semantic web
 - ▶ saving the world
- ▶ New deduction-based tools:
 - ▶ static analysis
 - ▶ verifying synthesis
 - ▶ formal implementation languages
- ▶ Needs marketing
 - ▶ generate a 'buzz'
 - ▶ highlight awards (e.g. Turing winners)
 - ▶ influence hiring committees
 - ▶ make funders aware of potential

A Golden Age for Formal Methods

- ▶ New application areas
 - ▶ finance
 - ▶ biology
 - ▶ semantic web
 - ▶ saving the world
- ▶ New deduction-based tools:
 - ▶ static analysis
 - ▶ verifying synthesis
 - ▶ formal implementation languages
- ▶ Needs marketing
 - ▶ generate a 'buzz'
 - ▶ highlight awards (e.g. Turing winners)
 - ▶ influence hiring committees
 - ▶ make funders aware of potential

My particular interests

- ▶ High accuracy models
 - ▶ ISA as 'golden' semantics: ARM, IA-32, PowerPC
 - ▶ basis for verification
 - ▶ derive higher levels
- ▶ Theorem proving as an implementation method
 - ▶ software and hardware compilation by theorem proving
 - ▶ executing logic specifications (certified computation)
- ▶ Multiculturism
 - ▶ make different tools and formalisms get along
 - ▶ **easy**: HOL \leftrightarrow Isabelle/HOL; **hard**: HOL \leftrightarrow Coq
 - ▶ set theory as unifying framework

My particular interests

- ▶ High accuracy models
 - ▶ ISA as 'golden' semantics: ARM, IA-32, PowerPC
 - ▶ basis for verification
 - ▶ derive higher levels

- ▶ Theorem proving as an implementation method
 - ▶ software and hardware compilation by theorem proving
 - ▶ executing logic specifications (certified computation)

- ▶ Multiculturism
 - ▶ make different tools and formalisms get along
 - ▶ **easy** HOL \leftrightarrow Isabelle/HOL; **hard** HOL \leftrightarrow Coq
 - ▶ set theory as unifying framework

My particular interests

- ▶ High accuracy models
 - ▶ ISA as 'golden' semantics: ARM, IA-32, PowerPC
 - ▶ basis for verification
 - ▶ derive higher levels

- ▶ Theorem proving as an implementation method
 - ▶ software and hardware compilation by theorem proving
 - ▶ executing logic specifications (certified computation)

- ▶ Multiculturism
 - ▶ make different tools and formalisms get along
 - ▶ **easy** HOL \leftrightarrow Isabelle/HOL; **hard** HOL \leftrightarrow Coq
 - ▶ set theory as unifying framework

My particular interests

- ▶ High accuracy models
 - ▶ ISA as 'golden' semantics: ARM, IA-32, PowerPC
 - ▶ basis for verification
 - ▶ derive higher levels

- ▶ Theorem proving as an implementation method
 - ▶ software and hardware compilation by theorem proving
 - ▶ executing logic specifications (certified computation)

- ▶ Multiculturism
 - ▶ make different tools and formalisms get along
 - ▶ **easy:** HOL \leftrightarrow Isabelle/HOL; **hard:** HOL \leftrightarrow Coq
 - ▶ set theory as unifying framework