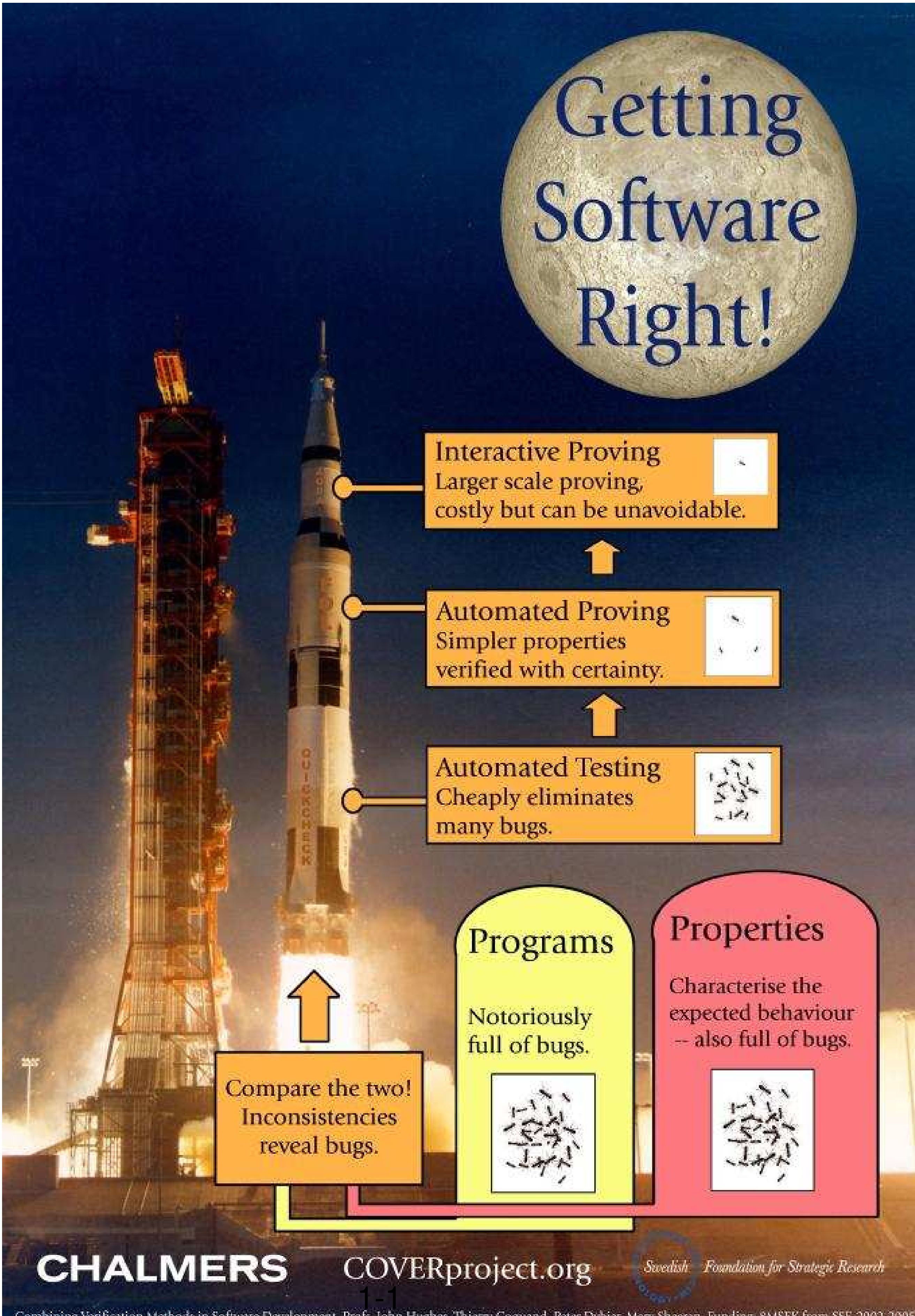
Cover Summary

Nils Anders Danielsson

Chalmers

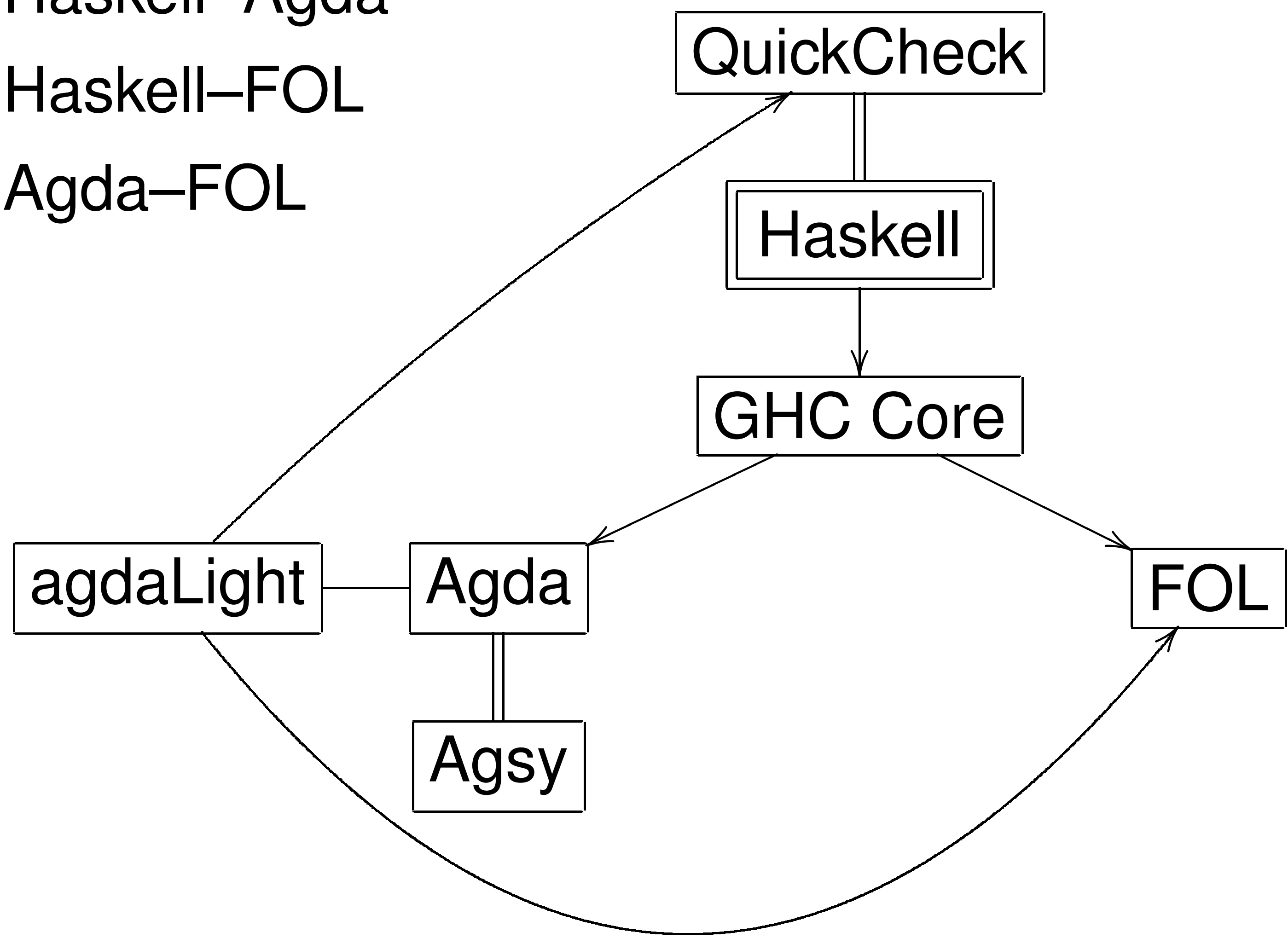

Getting Software Right!
Interactive Proving
Larger scale proving, costly but can be unavoidable.
Automated Proving
Simpler properties verified with certainty.
Automated Testing
Cheaply eliminates many bugs.
QUICKCHECK
Programs
Notoriously full of bugs.
Properties
Characterise the expected behaviour -- also full of bugs.
Compare the two! Inconsistencies reveal bugs.
CHALMERS
COVERproject.org
Swedish Foundation for Strategic Research
Combining Verification Methods in Software Development. Profs. John Hughes, Thierry Coquand, Peter Dybjer, Mary Sheeran. Funding: 8MSEK from SSF, 2002-2005

Base tools

- QuickCheck
  - Specification and randomised testing library for Haskell.
  - New version coming soon.
- Agda
  - Interactive theorem prover (dependent type theory).
  - Recently added: Classes, hidden arguments, primitive types, John Major equality.
- Agsy
  - Proof search plugin for Agda capable of some inductive arguments.

Current focus

- Haskell–Agda
- Haskell–FOL
- Agda–FOL



Haskell–Agda

- Monadic translation of GHC Core into Agda.
- $f\ 0 \mapsto f^* \ggg\!= \lambda g \to g\ (\mathit{return}\ 0)$.
- Identity monad or Maybe monad.
- Proof under way that total reasoning sometimes is meaningful.
- Some problems related to impredicativity. (GHC Core is a variant of F^ω.)
- General recursion cannot be handled.

Haskell–FOL

- Translation of GHC Core into (untyped) FOL.
- Santa: Tool for translating FOL formulas into many different formats.
- Example: $(P_2 P_1)^{-1} = P_1^{-1} P_2^{-1}$ in the revision control system Darcs.

$$
\begin{aligned}
&prop\_invert\_comp = \\
&\quad forAll \; \$ \; \lambda p_1 \rightarrow \\
&\quad forAll \; \$ \; \lambda p_2 \rightarrow \\
&\qquad invert \; (ComP \; [p_2, p_1]) {=}{=}{=} ComP \; [invert \; p_1, invert \; p_2]
\end{aligned}
$$

- Many things open: Types, totality, finiteness, bottoms, induction, coinduction…

Agda–FOL

- agdaLight, reimplementation of Agda without meta variables and inductive families.
- Plugins for QuickCheck and FOL.

Agda–FOL example

- $isRing$: Axiomatises a ring.
- $isBool :: (X :: Set) \to (X \to X \to X) \to Prop$
  $isBool\ X\ (*) = (x :: X) \to Id\ X\ (x * x)\ x$
- $axR1 :: isRing\ R\ (+)\ (*)\ minus\ Zero\ One$
  $axR2 :: isBool\ R\ (*)$
- $thm :: (x :: R) \to x + x \equiv Zero$
  $thm\ x = fol{-}plugin\ (axR1, axR2)$
- $thm1 :: (x :: R) \to (y :: R) \to x * y \equiv y * x$
  $thm1\ x\ y = fol{-}plugin\ (axR1, axR2)$