Integrating Formal Program Verification with Testing

Cyrille Comar, Johannes Kanig and **Yannick Moy**

Integrating Formal Program Verification with Testing

Cyrille Comar, Johannes Kanig and **Yannick Moy**

Testing Integrating Verification with Formal Progra

Cyrille Comar, Johannes Kanig and **Yannick Moy**

AdaCore
The GNAT Pro Company

HI-LITE
Simplifying the use of formal methods

openDO

HI-LITE

Motivation

DO-178C: formal methods can replace testing

Formal methods […] might be the primary source of evidence for the satisfaction of many of the objectives concerned with development and verification.

2011: Formal Methods Supplement (DO-333)

Myths of formal methods

- Myth 4: Formal methods require highly trained mathematicians
- Myth 5: Formal methods increase the cost of development
- Myth 6: Formal methods are unacceptable to users
- Myth 7: Formal methods are not used on real, large-scale software

(Anthony Hall, Praxis Systems, 1990)

Since 2001, Airbus has been integrating several tool supported formal verification techniques into the development process of avionics software products.

2009: *Formal Verification of Avionics Software Products*
(Souyris, Wiels, Delmas, Delseny)

HI-LITE | Proof + Test

HI-LITE Programming Contracts

{P}C{Q} Hoare logic (1969)

logic contracts
for proofs

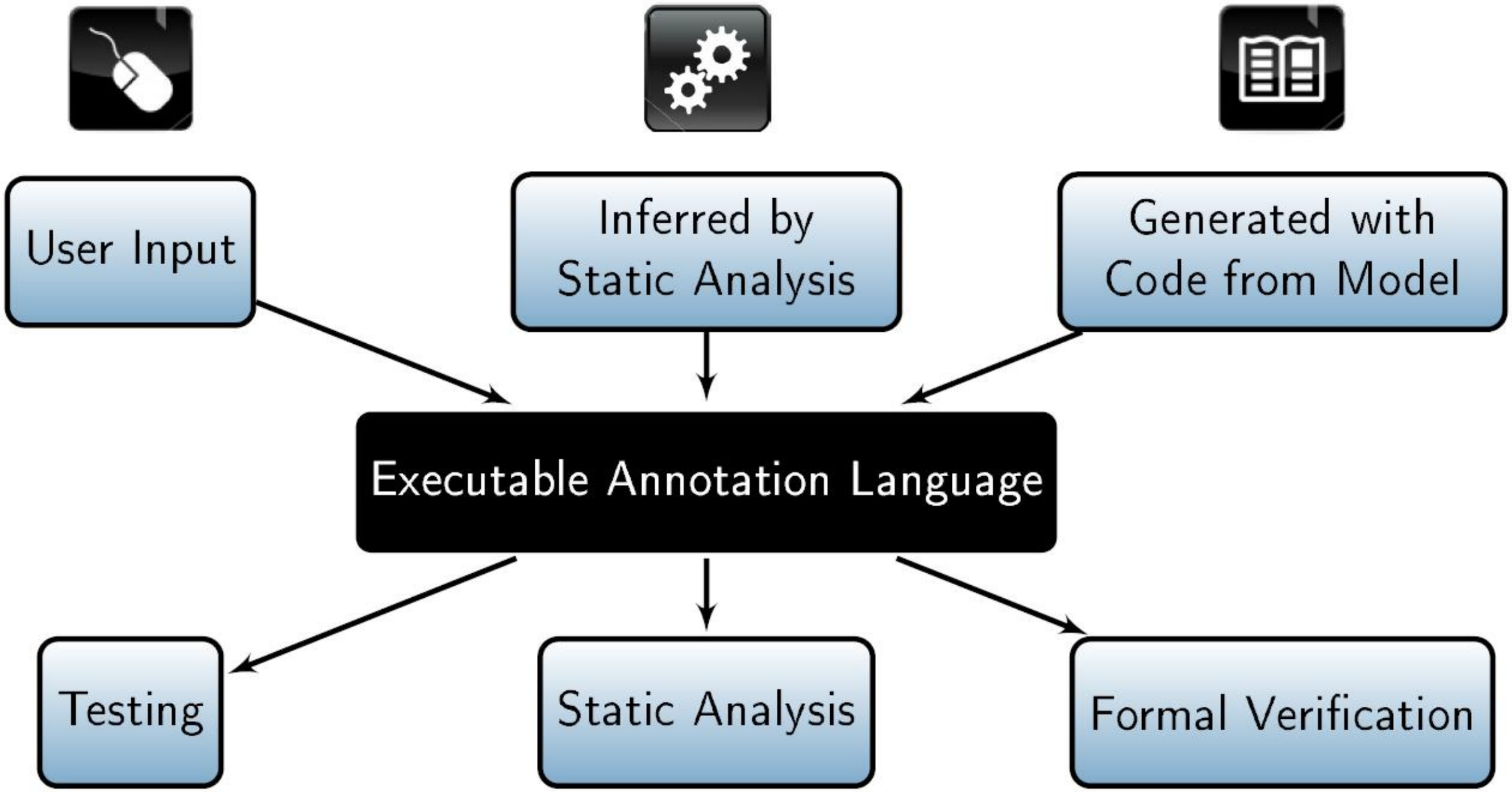
executable contracts
for tests

SPARK (1987)

Eiffel DbC (1986)

Hi-Lite: executable annotation language???

HI-LITE Project



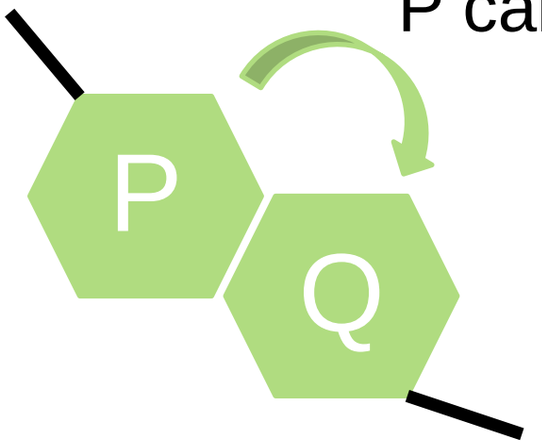
HI-LITE Ada 2012

```
1  function One_Of (V, X, Y : in Int) return Boolean
2  is (V = X or else V = Y);
3
4  function Max (X, Y : in Int) return Int with
5    Pre  => X /= Y,
6    Post => Max'Result >= X and then
7            Max'Result >= Y and then
8            One_Of (Max'Result, X, Y);
9
10 function Max (X : in Int_Array) return Int with
11   Post => (for all J in X'Range =>
12              Max'Result >= X(J)) and then
13           (for some J in X'Range =>
14              Max'Result = X(J));
```

HI-LITE Testing vs. Formal Verification

use Q code
cover P constructs

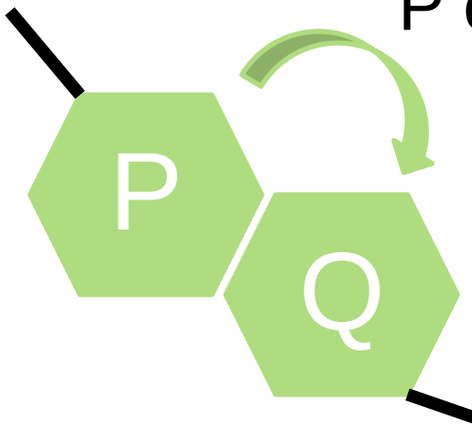
P calls Q



actual body of Q
or stub…

prove pre of Q
assume post of Q

P calls Q

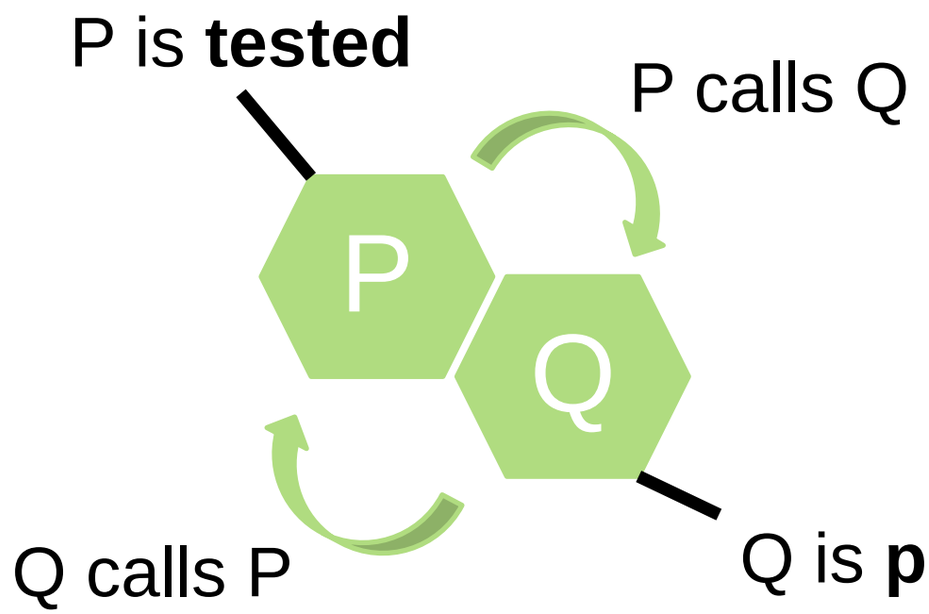


assume pre of Q
prove post of Q

local exhaustivity argument:
each function covered
→ enough behaviors explored

global soundness argument:
all functions proved
→ all assumptions justified

HI-LITE Combining tests and proofs



How so we justify assumptions made during proof?

verification combining tests and proofs should be

AT LEAST AS GOOD AS

verification based on tests only

HI-LITE Caution: contracts are not only pre/post!

strong typing

parameters not aliased

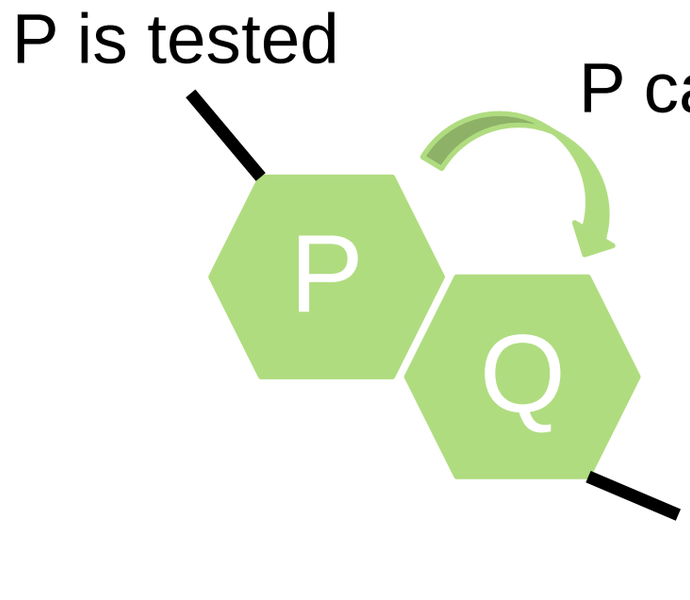
```
1  procedure Open
2    (Customer : in      Identity.Name;
3     Id       : in      Identity.Id;
4     Cur      : in      Money.CUR;
5     Account  :     out Account_Num)
6  with
7    Pre  => not Max_Account_Reached,
8    Post => Existing (Account)...
```

data dependences

parameters initialized

HI-LITE Combination 1: tested calls proved

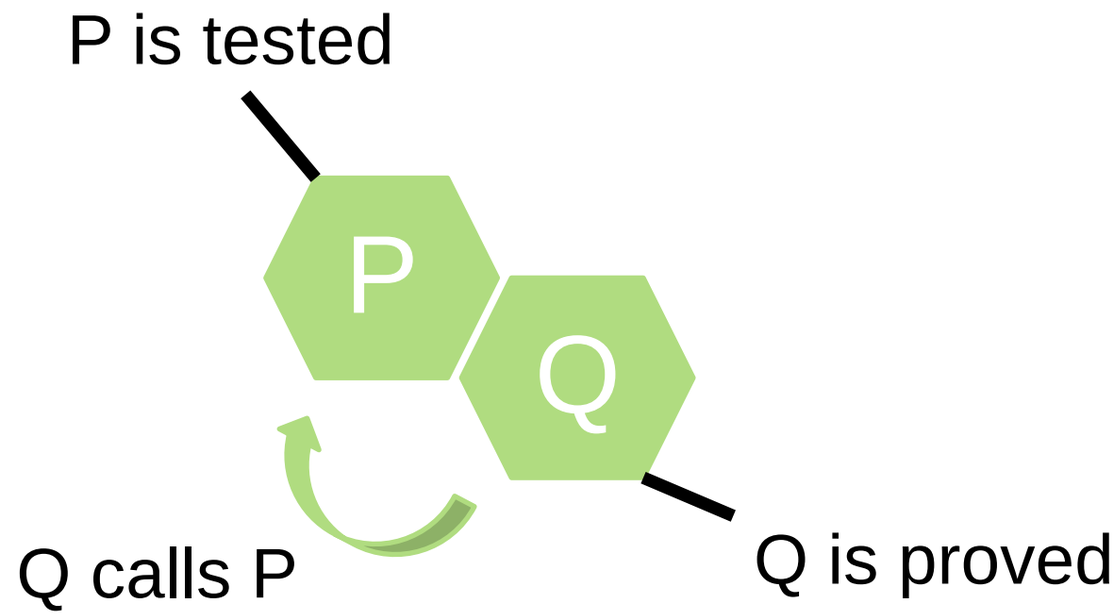
during testing:
check that
precondition of Q
is respected



assumption for proof:
precondition of Q
is respected

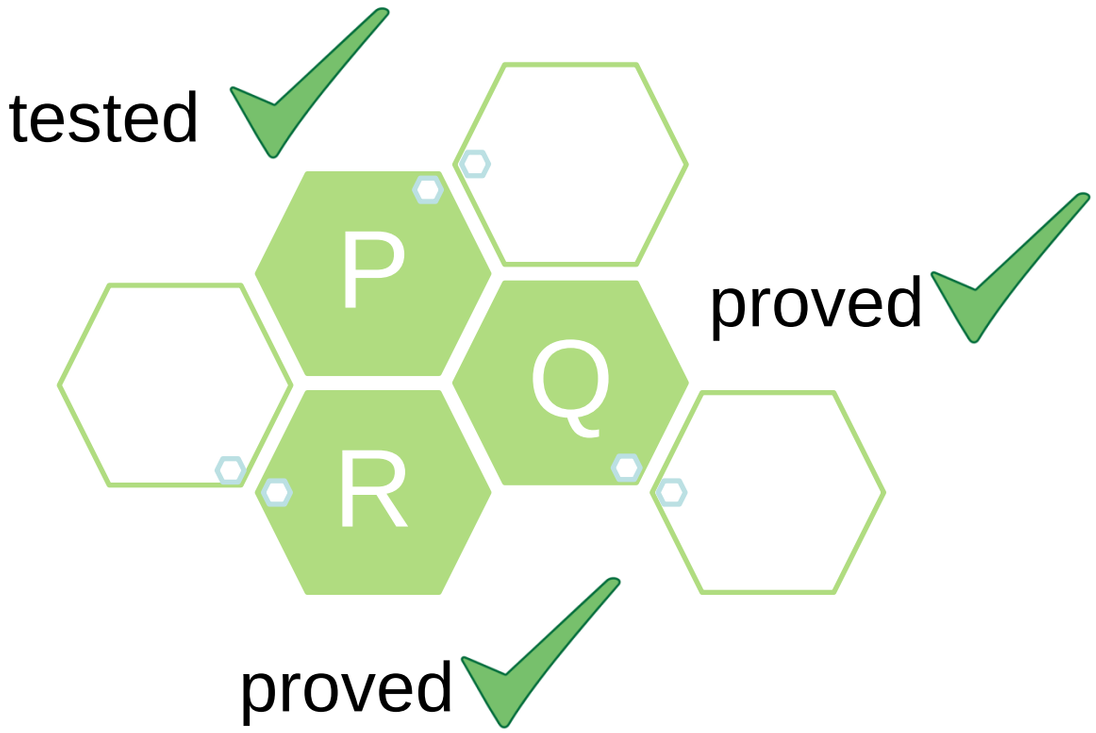
Combination 2: proved calls tested

during testing:
check that
postcondition of P
is respected



assumption for proof:
postcondition of P
is respected

HI-LITE Testing + Formal Verification

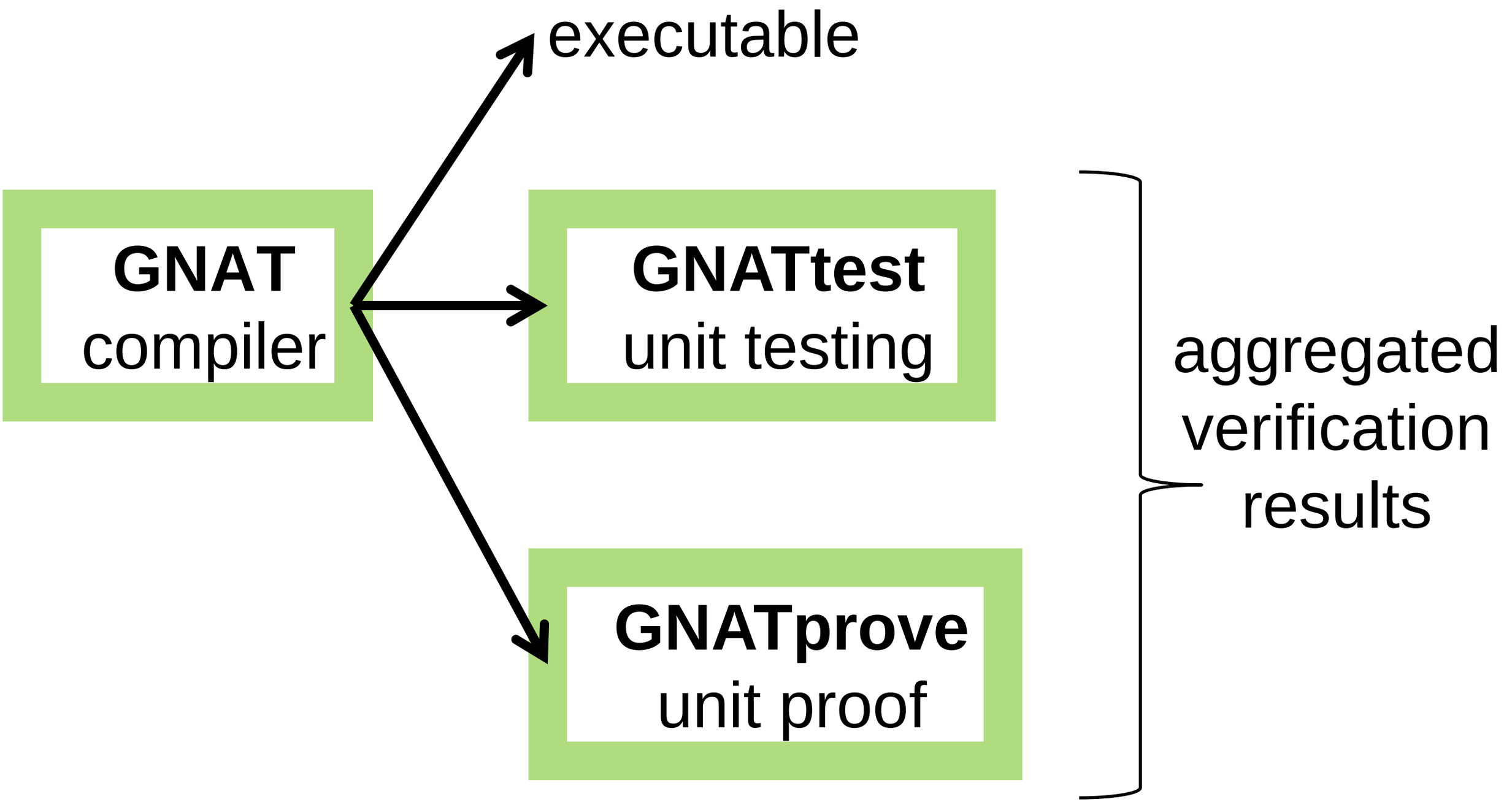


local exhaustivity argument:
- test: function covered
- proof: by nature of proof

global soundness argument:
- proof: assumptions proved
- test: assumptions tested

Testing must check additional properties
Done by compiler instrumentation

HI-LITE GNAT toolsuite



HI-LITE

Conclusion

HI-LITE Airbus 5 “must-have” of formal methods

- Soundness ✓
- Applicability to the code ✓
- Usability by normal engineers on normal computers ✓
- Improve on classical methods
- Certifiability

} current work

HI-LITE Benefits of openness

openDO.org

git

- announcements
- meeting slides
- articles / docs

- public:
  - ✓ meeting minutes
  - ✓ technical work
  - ✓ 69 members
- private:
  - ✓ management
  - ✓ partner code

- all code
- dev docs
- user docs

→ external collaborations with industry and academia

HI-LITE Project Partners

AdaCore
The GNAT Pro Company

ALTRAN
PRAXIS

EADS
astrium

cea
list

INRIA

THALES

HI-LITE

www.open-do.org/projects/hi-lite