The Future of Formal Software Verification in Avionics

Yannick Moy

Formal Methods 2012 Industry Day

AdaCore
The GNAT Pro Company

HI-LITE
Simplifying the use of formal methods

openDO

HI-LITE

Background

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)

HI-LITE Programming Contracts

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

logic contracts
for proofs

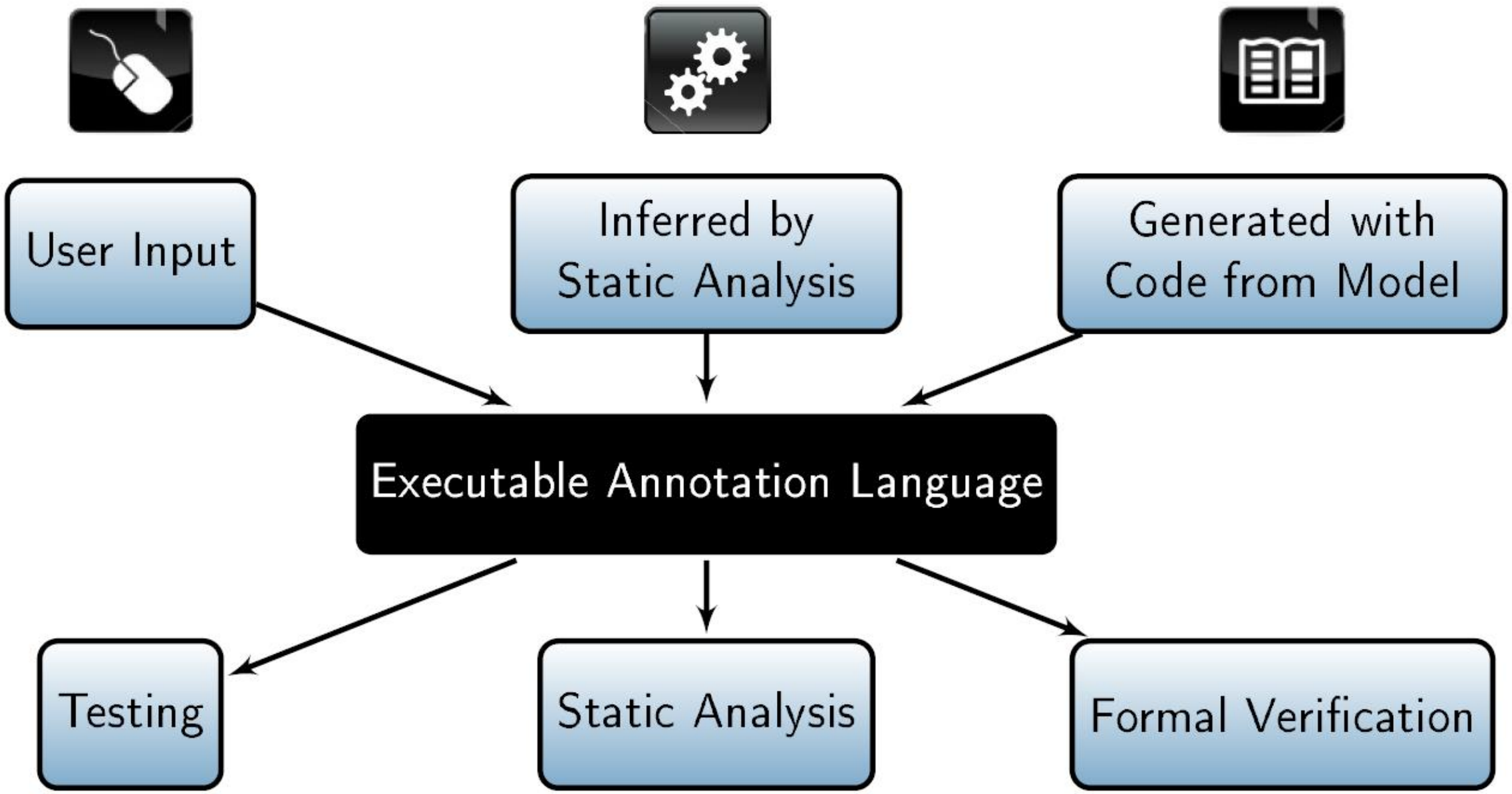
executable contracts
for tests

SPARK (1987)

Eiffel DbC (1986)

Hi-Lite: executable annotation language???

HI-LITE Project



HI-LITE Ada 2012

```
function One_Of (V, X, Y : in Int) return Boolean
is (V = X or else V = Y);

function Max (X, Y : in Int) return Int with
  Pre  => X /= Y,
  Post => Max'Result >= X and then
          Max'Result >= Y and then
          One_Of (Max'Result, X, Y);

function Max (X : in Int_Array) return Int with
  Post => (for all J in X'Range =>
             Max'Result >= X(J)) and then
          (for some J in X'Range =>
             Max'Result = X(J));
```

Example of contract

Example:

- ✓ A list of event detection statuses
- ✓ Request to reset the detection status for Event

```
procedure Reset_Event_Status (Event : in T_Event) with
Post =>
   not Event_Status(Event).Detection and
   (for all Other_Event in T_Event =>
    ( if Other_Event /= Event then
      Event_Status( Other_Event ) = Event_Status'Old( Other_Event ) ));
```

Post-condition

The detection of event is reset

For all other events

The detection status is unchanged

Event1	Event	Event3
Not detected	Not detected	Detected

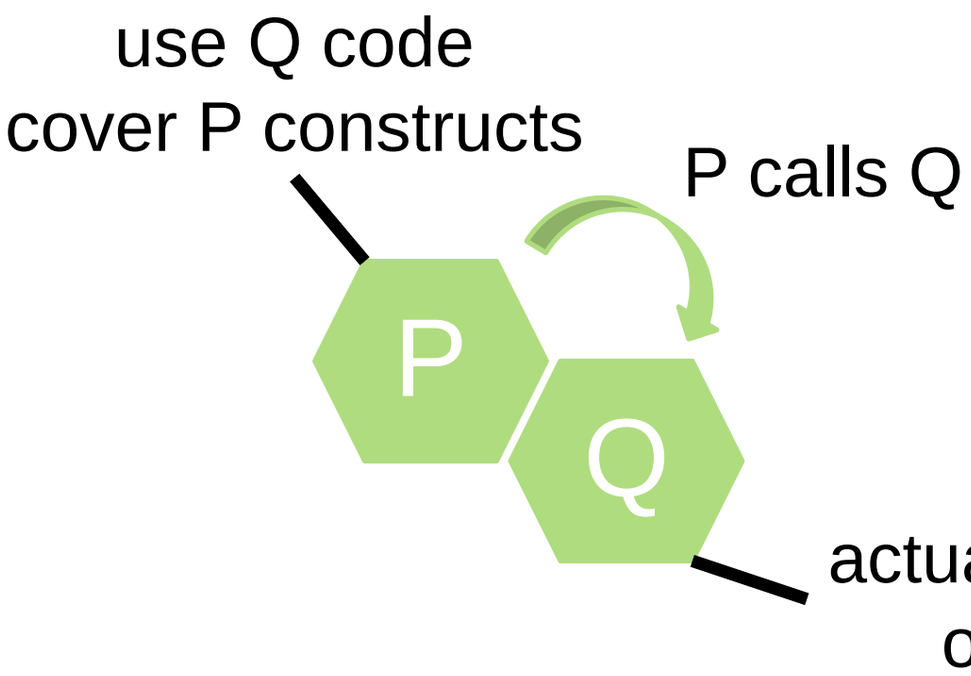
HI-LITE GNAT compiler support for Ada 2012

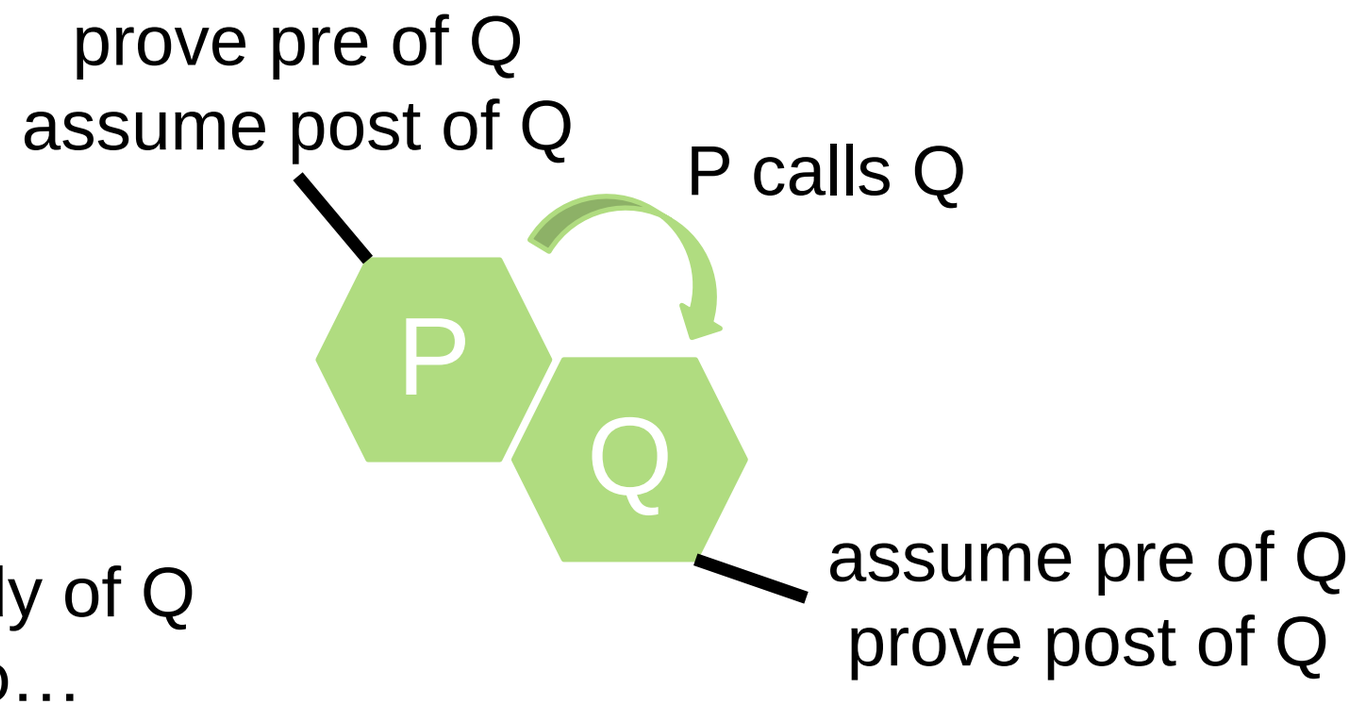
- Run-time checking of new assertions
- New aspect to formally specify test cases
- New switch to choose semantics of integers in assertions, e.g. unbounded integers or largest machine integers
- New library of containers adapted to formal verification (lists, sets, maps, vectors)
- New run-time checks for integrating tests and proofs

HI-LITE

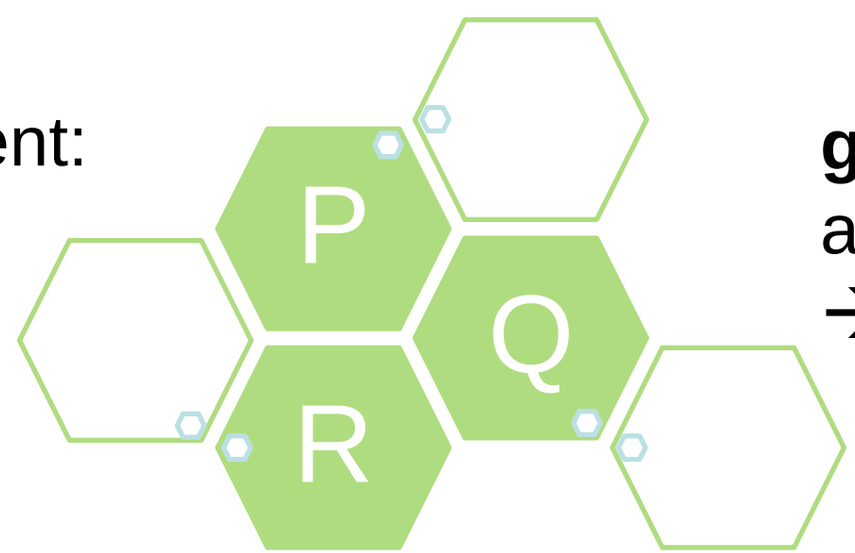
Proof + Test

Testing vs. Formal Verification



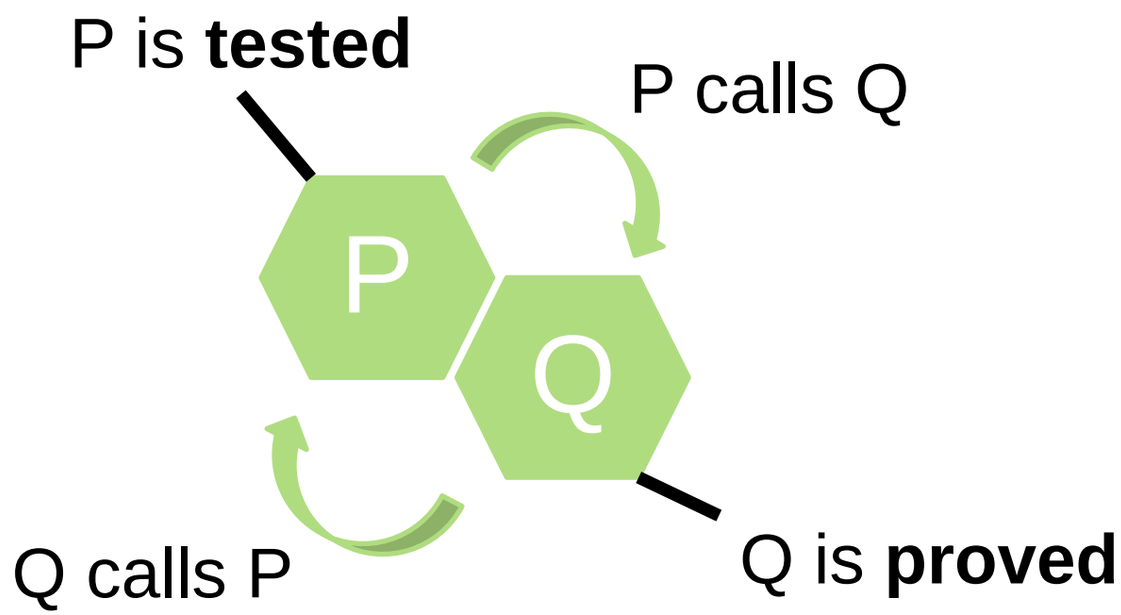


local exhaustivity argument:
each function covered
→ enough behaviors explored



global soundness argument:
all functions proved
→ all assumptions justified

HI-LITE Combining tests and proofs



How so we justify assumptions made during proof?

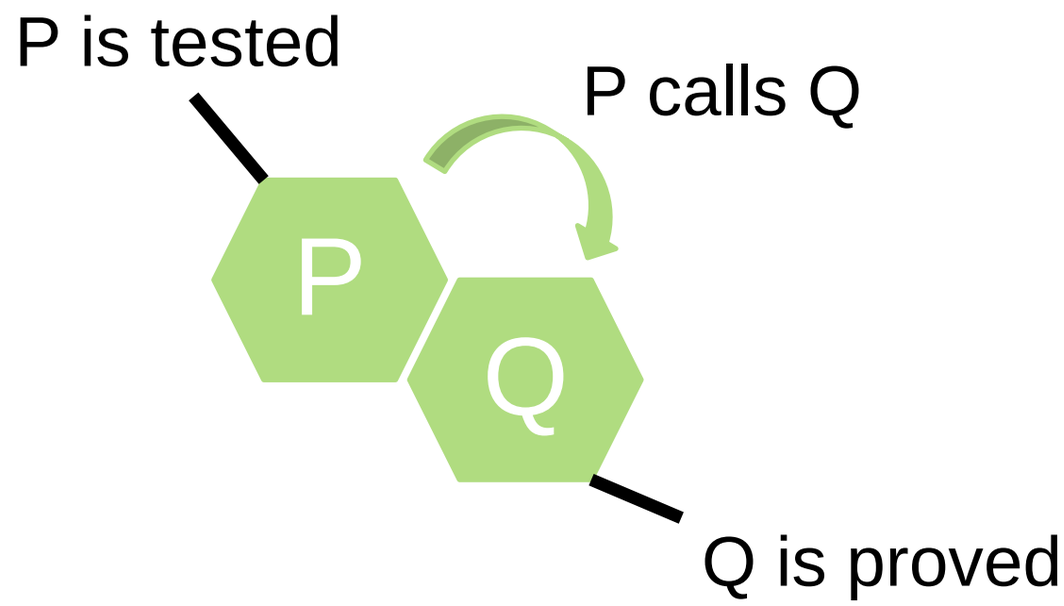
verification combining tests and proofs should be

AT LEAST AS GOOD AS

verification based on tests only

HI-LITE Combination 1: tested calls proved

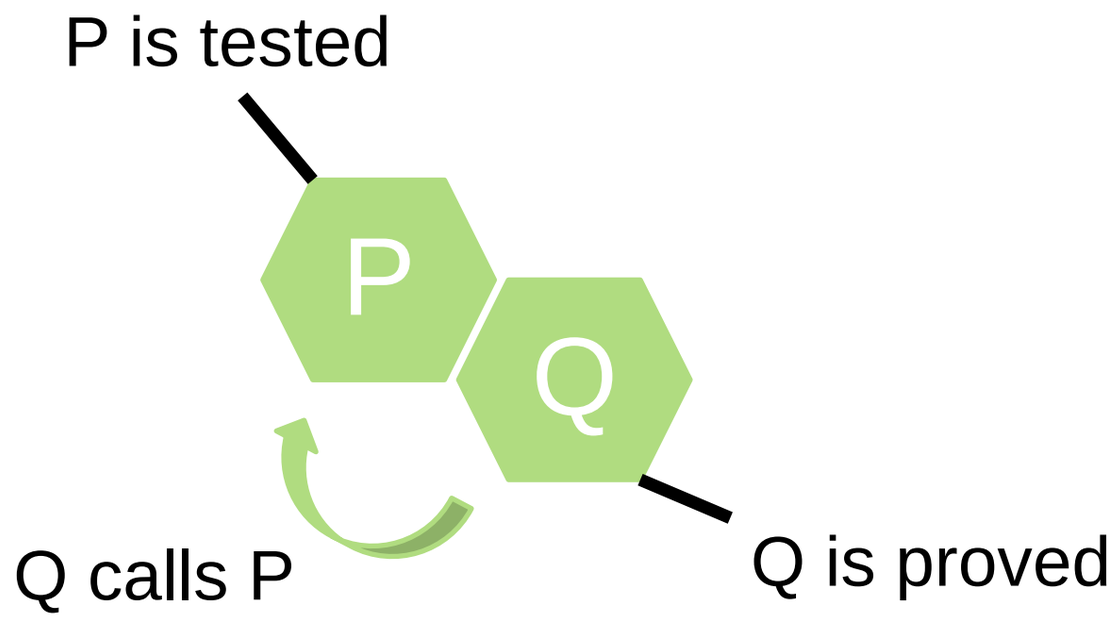
during testing:
check that
precondition of Q
is respected



assumption for proof:
precondition of Q
is respected

HI-LITE Combination 2: proved calls tested

during testing:
check that
postcondition of P
is respected



assumption for proof:
postcondition of P
is respected

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

strong typing

parameters not aliased

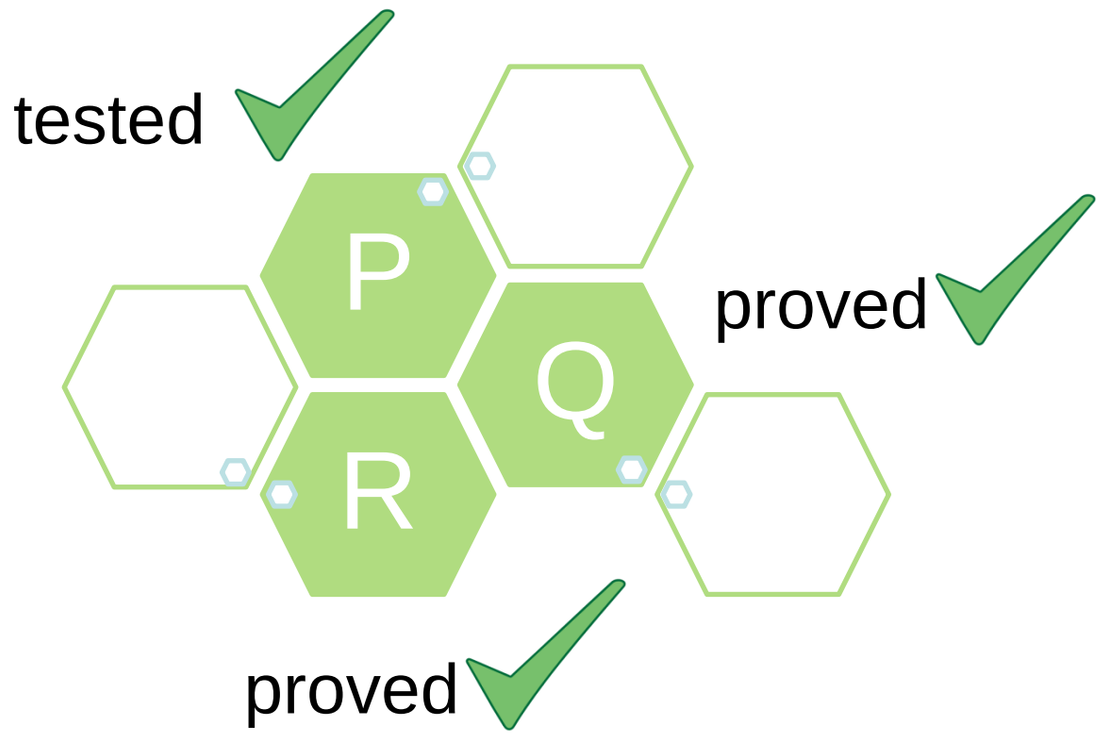
```
procedure Open
   (Customer : in      Identity.Name;
    Id       : in      Identity.Id;
    Cur      : in      Money.CUR;
    Account  :     out Account_Num)
with
   Pre  => not Max_Account_Reached,
   Post => Existing (Account)...
```

data dependences

parameters initialized

Testing must check additional properties
Done by compiler instrumentation

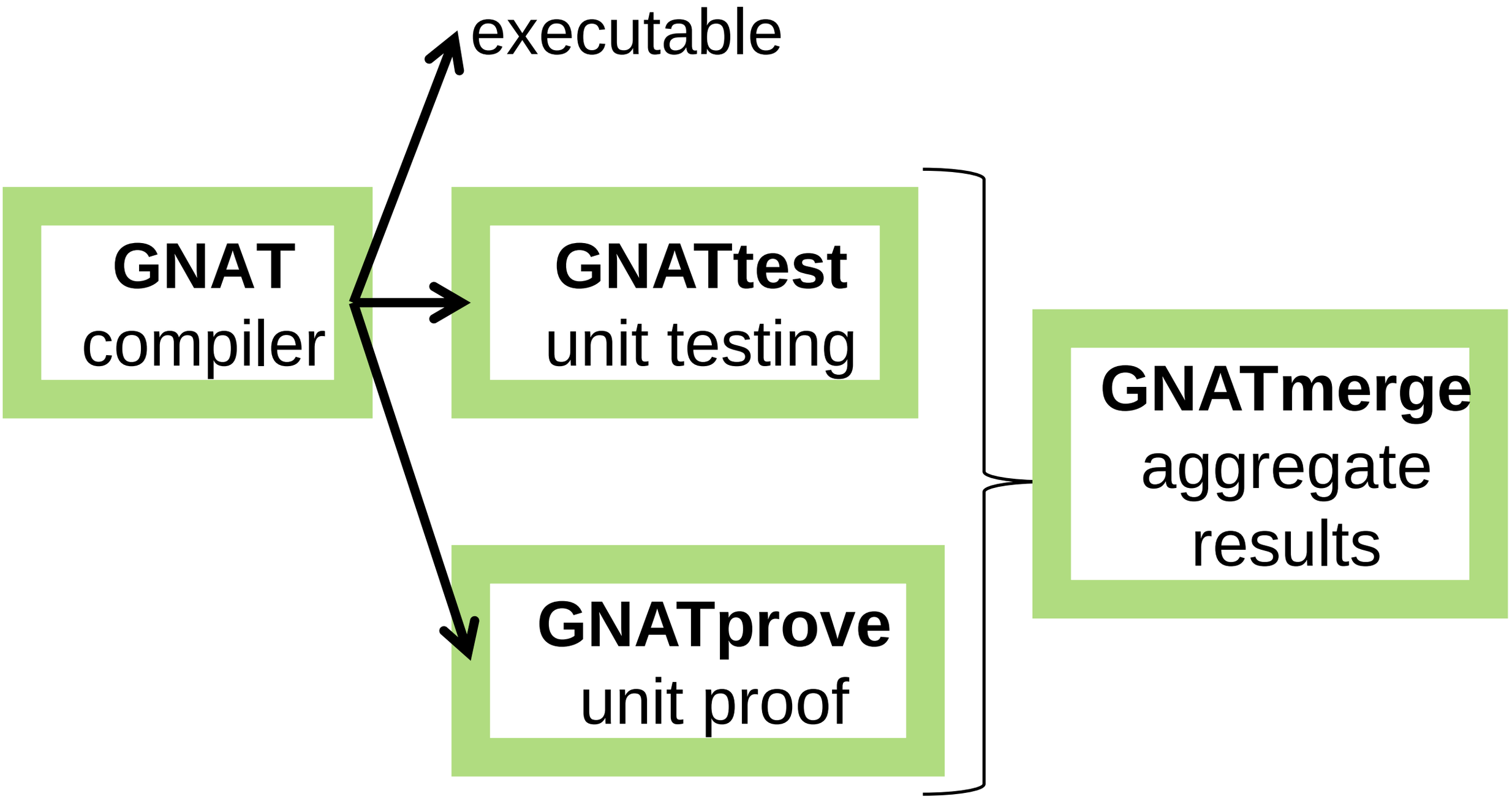
HI-LITE Testing + Formal Verification



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

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

HI-LITE GNAT toolsuite



HI-LITE GNATmerge result

Locations

- Builder results (4 items)
  - segway.ads (3 items)

```
16:4    goal not reached at Speed_Is_Valid (TEST FAILED)
22:4    info: goal reached at State_Update (TEST FAILED, PROVED)
28:4    info: goal reached at Execute (TEST PASSED)
```

  - segway.adb (1 item)

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 Project Partners

AdaCore
The GNAT Pro Company

ALTRAN PRAXIS

EADS astrium

cea list

INRIA

THALES

HI-LITE

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