



EUCLIDE :

automatic test data generation for critical C programs

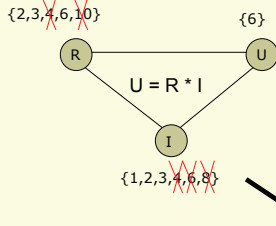


Supported by ANR CAVERN

<http://cavern.inria.fr>

Constraint Programming

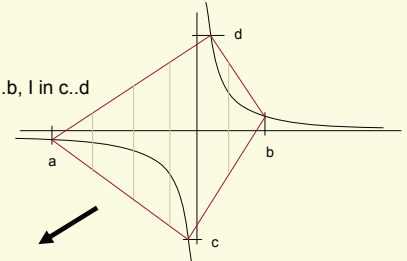
Exploit relations (constraints) to infer new informations on objects that represent unknowns (variables)



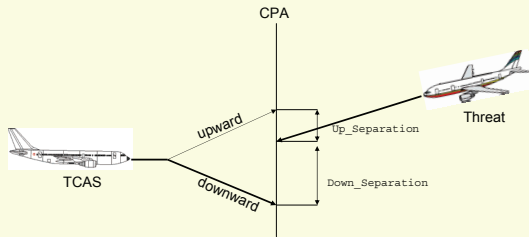
Abstractions

Over-approximate the computation of relations to benefit from powerful solving techniques (e.g. Linear Programming)

$$1 = R * I, \quad R \text{ in } a..b, I \text{ in } c..d$$



Automatic test data generation for critical C programs
Counter-example generation
Assertion verification



Are properties P1a, P1b, etc. satisfied by this implementation ?

```

int alt_sep_test()
{
  bool enabled, toas_equipped, intent_not_known;
  bool need_upward_RA, need_downward_RA;
  int alt_sep;

  enabled = HighConfidence && (Own_Tracked_Alt_Rate <= OLEV);
  toas_equipped = (Other_Capability == TCAS_RA);
  intent_not_known = (Two_of_Three_Reports_Valid && Other_RAC == NO_INTENT);

  alt_sep = UNRESOLVED;

  if (enabled && ((toas_equipped && intent_not_known) || !toas_equipped))
  {
    need_upward_RA = Non_Crossing_Biased_Climb() && Own_Below_Threat();
    need_downward_RA = Non_Crossing_Biased_Descend() && Own_Above_Threat();
    if (need_upward_RA && need_downward_RA)
      alt_sep = UNRESOLVED;
    // unreachable: Own_Below_Threat and Own_Above_Threat can't be both true
    else if (need_upward_RA)
      alt_sep = UPWARD_RA;
    else if (need_downward_RA)
      alt_sep = DOWNWARD_RA;
    else
      alt_sep = UNRESOLVED;
  }

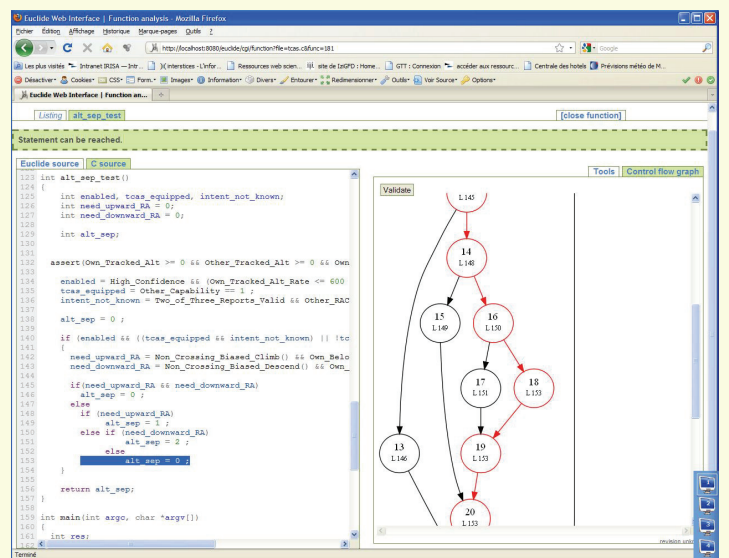
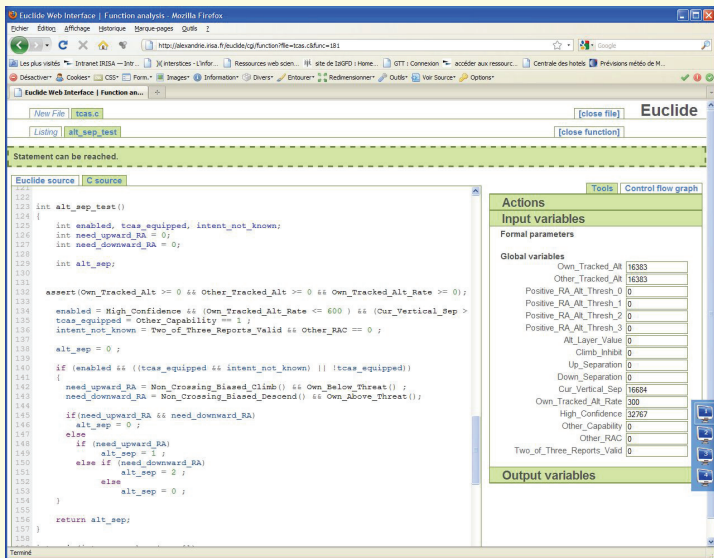
  return alt_sep;
}

```

Num.	Property	Explanation	ACSL specification
P1a	Safe advisory selection	An downward RA is never issued when an downward maneuver does not produce an adequate separation	assertion Up_Separation >= Positive_RA_Alt_Tresh && Down_Separation < Positive_RA_Alt_Tresh; ensure result != need_Downward_RA;
P1b	Safe advisory selection	An upward RA is never issued when an upward maneuver does not produce an adequate separation	assertion Up_Separation < Positive_RA_Alt_Tresh && Down_Separation >= Positive_RA_Alt_Tresh; ensure result != need_Upward_RA;
P1c	Safe advisory selection	A downward RA is never issued when neither climb or descend maneuver is available	assertion Up_Separation < Positive_RA_Alt_Tresh && Own_Below_Threat() && Own_Above_Threat();

EUCLIDE is a free open-source software

A web-based interface, accessible online <http://euclide.gforge.inria.fr/>



Euclide is a **C**onstraint **L**anguage based on **I**mperative **D**efinition

Contact: Arnaud.Gotlieb@inria.fr (INRIA Rennes)