Formalna weryfikacja oprogramowania w lotnictwie

Bartłomiej Wołowiec

30 listopada 2010

- wysoka niezawodność,
- zmniejszenie kodu obsługi sytuacji wyjątkowych,
- DO178B

Wymagania stawiane nowemu podejściu:

- możliwość sprawdzania kodu C (nie modelu!),
- łatwe do zrozumienia i używania,
- szybki zwrot (zastąpienie testów),
- łatwość integracji,
- sprawdzanie właściwości funkcjonalnych, bezpieczeństwa, zależności czasowych
- sprawdzanie małych modułów, ale również całego systemu (do 1mln linii kodu),
- głównie proste algorytmy, proste pętle i ich niezmienniki.

ANSI/ISO C Specification Language (ACSL)

- stworzony pod koniec 2008 przez Commissariat à l'Énergie Atomique i INRIA,
- inspirowany JML,
- nastawiony na weryfikacje przez wnioskowanie (nie na sprawdzanie w czasie wykonania).

```
 /*@
requires \valid(p);
assigns *p;
ensures *p == \old(*p) + 1;
*/
void incrstar (int *p);
```

CIL - C Intermediate Language

- podzbiór języka C,
- narzędzia do tłumaczenia C na CIL (Ocaml, BSD),
- prosta analiza,
- zgodny z GCC (testy: kernel, gimp), MSC.

Różne opcje do włączenia:

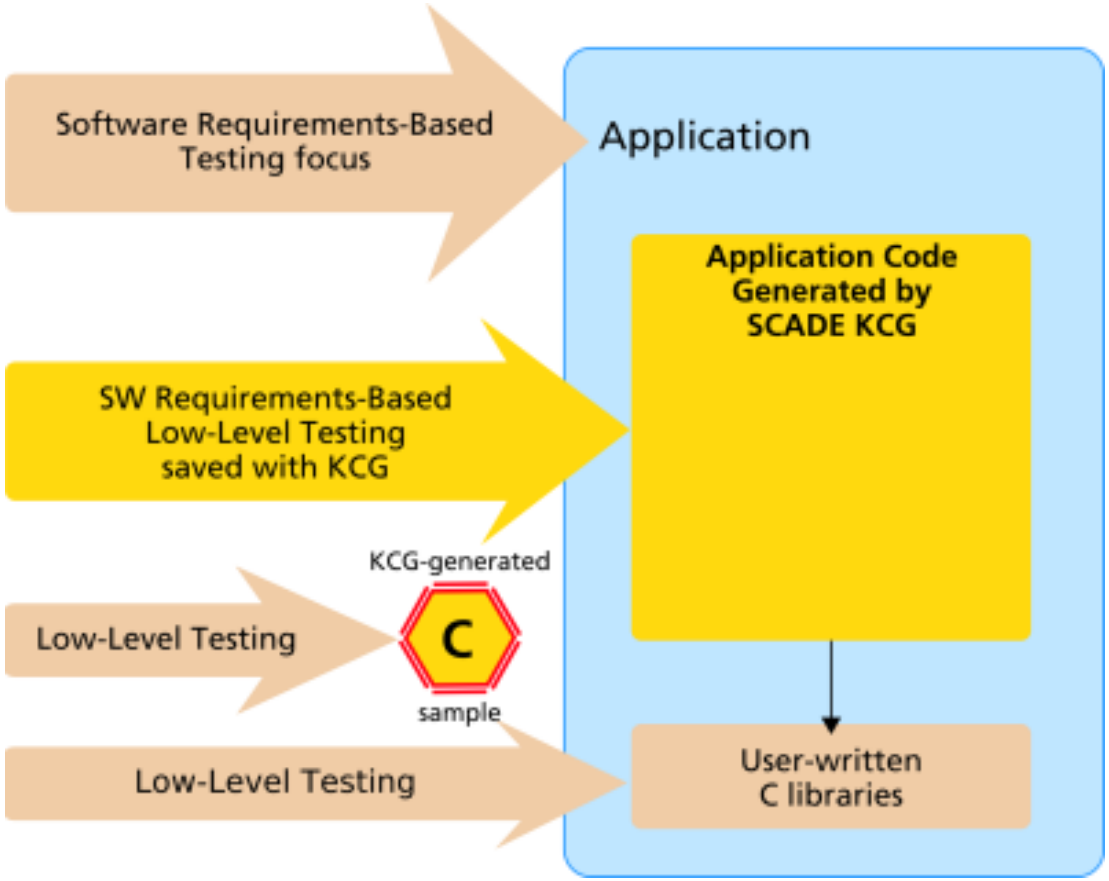
- unikalne identyfikatory, brak anonimowych typów itd,
- redukcja zagnieżdżonych struktur,
- tablice bez rozmiarów (int tab[] = 1,2,3;),
- brak zmiennych statycznych,
- pojedynczy return,
- brak goto, break, continue,
- brak operatora '_?_:_' ,
- itd...

Dodatkowe ograniczenia zakładane przez niektóre programy:

- brak rekurencji, nieograniczonych pętli,
- brak efektów ubocznych funkcji,
- brak wskaźników do funkcji,
- brak arytmetyki na wskaźnikach,

SCADE (Esterel, komercyjny)

- modelowanie całego systemu
- wstępne generowanie kodu



Dowodzenia własności na niskim poziomie, interpretując znaczenie instrukcji. Idea działania:

1. uproszczenie języka C do jego podzbioru (np CIL),
2. przedstawienie własności jako twierdzenia w formalnym języku,
3. udowodnienie twierdzenia, albo za pomocą automatycznego narzędzia (cvc3, z3, alt-ergo), albo z pomocą interaktywnego dowodzenia (coq, isabelle).

Caveat

- operuje na ograniczonym zbiorze języka C,
- własny język specyfikacji oparty na logice pierwszego rzędu,
- dwa cele:
  - analiza sterowania i przepływu danych,
  - dowodzenie własności zdefiniowanych przez użytkownika,
- brak dostępnych informacji poza publikacjami.

Frama-C (LGPL, BSD)

- framework,
- integracja z CIL,
- wspiera ACSL.

Wiele modułów, m.in:

- jessie - formalne dowodzenie (adnotacje, bezpieczeństwo operacji),
- value analysis - oblicza zbiór możliwych wartości zmiennej (lub jego przybliżenie),
- slicing - usuwanie zbędnego kodu z zachowaniem pewnych własności.

Astree

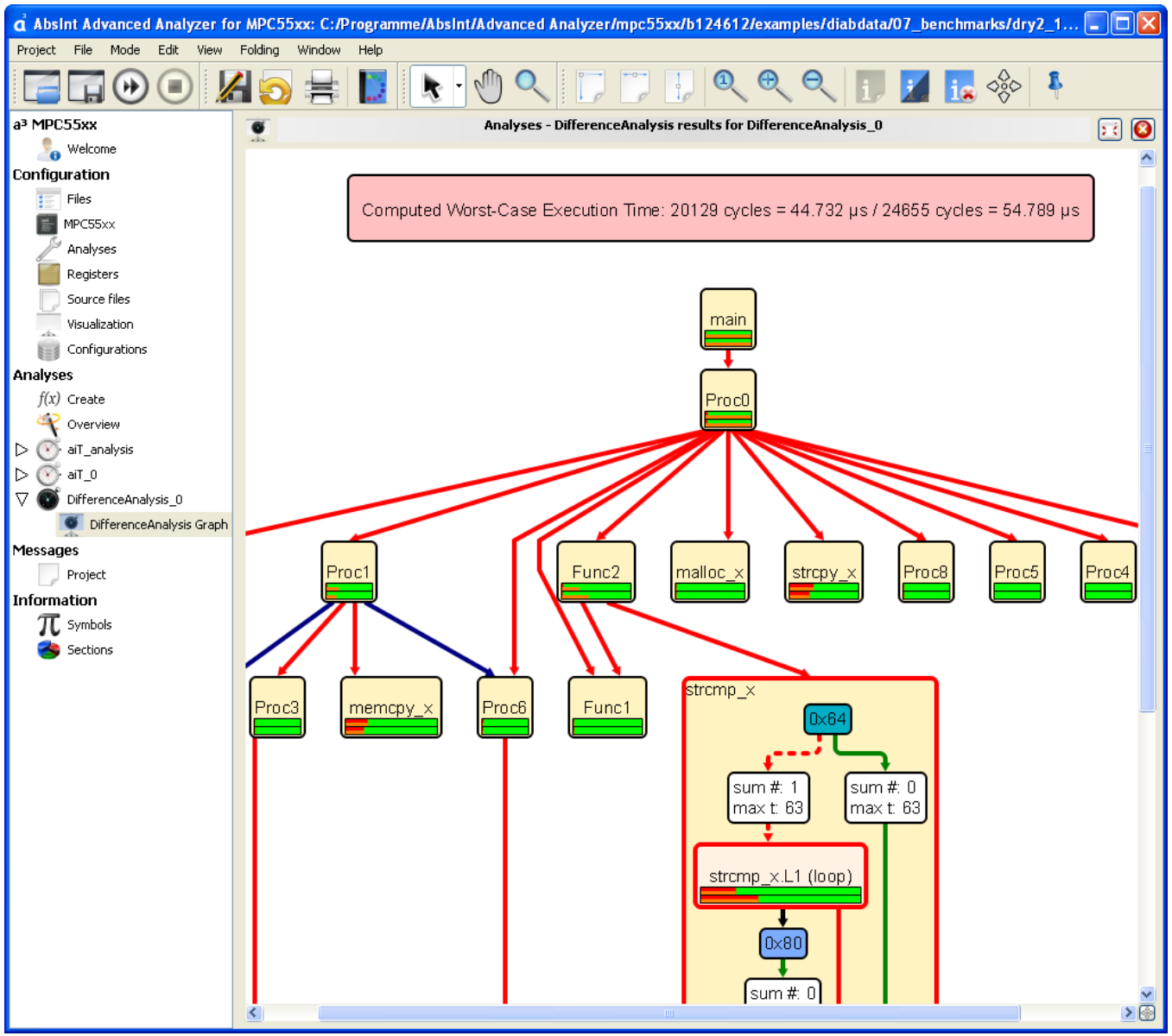
- operuje na podzbiorze języka C,
- sprawdzanie wystąpień RTE,
- bez rekurencji, bez dynamicznej alokacji pamięci,
- analiza głównie automatyczna,
- 1000 false alarms na 100 000 instrukcji (ile prawdziwych ?).

aiT

- analiza binarnego kodu (10 platform, m.in ARM7, PowerPC),
- obliczanie WCET (worst-case execution time),
- precyzyjna analiza łącznie z cache procesora,
- analiza abstrakcyjna, szacowanie z góry,
- przeanalizowana większość komponentów A380,
- umożliwia off-line scheduling.

Ograniczenia

- brak innych wątków, przerwań w tym samym czasie
- brak wyjątków procesora
- brak dynamicznej alokacji


AbsInt Advanced Analyzer for MPC55xx: C:/Programme/AbsInt/Advanced Analyzer/mpc55xx/b124612/examples/diabdata/07_benchmarks/dry2_1...
Project File Mode Edit View Folding Window Help
a³ MPC55xx
Welcome
Configuration
Files
MPC55xx
Analyses
Registers
Source files
Visualization
Configurations
Analyses
Create
Overview
aiT_analysis
aiT_0
DifferenceAnalysis_0
DifferenceAnalysis Graph
Messages
Project
Information
Symbols
Sections
Analyses - DifferenceAnalysis results for DifferenceAnalysis_0
Computed Worst-Case Execution Time: 20129 cycles = 44.732 µs / 24655 cycles = 54.789 µs
main
Proc0
Proc1
Func2
malloc_x
strcpy_x
Proc8
Proc5
Proc4
Proc3
memcpy_x
Proc6
Func1
strcmp_x
0x64
sum #: 1
max t: 63
sum #: 0
max t: 63
strcmp_x.L1 (loop)
0x80
sum #: 0

Stackanalyzer

- analiza binarnego kodu (10 platform, m.in ARM7, PowerPC),
- obliczanie maksymalnej wielkości stosu,
- analiza abstrakcyjna, szacowanie z góry,
- używany przez wszystkie zespoły w Airbus,
- umożliwia statyczny przydział stosu do zadania.

Fluctuat

- operuje na C lub asm,
- znajduje problemy z liczbami typu float, double,
- szacowanie górne i dolne wyniku.

Certyfikowana kompilacja.
Zapewnienie, że dowodzone własności w C dalej zachodzą w asm.
Przykład CompCert (2005, GPL / komercyjny).

- formalnie zweryfikowany (coq, 2008)
- kompilator CIL na powerPC.

- http://frama-c.com/
- http://www.first.fraunhofer.de/owx_download/acsl-by-example-5_1_0.pdf
- http://www.esterel-technologies.com/
- Formal Verification of Avionics Software Products, Jean Souyris, Virginie Wiels, David Delmas, and Herve Delseny
- Applying Formal Proof Techniques to Avionics Software: A Pragmatic Approach, Famantanantsoa Randimbivololona and Jean Souyris