Assume, Guarantee, or Repair

Hadar Frenkel, Orna Grumberg, Corina Pasareanu and Sarai Sheinvald

Computer Science Department, Technion – Israel Institute of Technology

Goal

- Modularly verify the correctness of a program against a safety specification, or repair the program, in case the verification fails.
- Using automata learning & abduction

Programs

1: pass = readInput;
2: while (pass ≤ 999)
3:     pass = readInput;
4: pass2 = encrypt(pass);
5: return pass2;

AGR Framework

Assume - Guarantee

Generate assumption $A_1$

(input: $M_1, M_2, P$

output: $A_i)$

model checking

(Step 1) $A_i \parallel M_1 = P$

false

true

false

spurious cex – weaken assumption

$cex$ analysis

$R_j$

$A_i$

(Step 2) $M_2 \times R_j = A_i$

$cex$ – strengthen assumption

Repair via Abduction

$\Sigma_j = \Sigma_{j-1} \cup \{c\}$

$\Sigma_j$

abduction on $t$

Generate repair $R_j$

to eliminate $t$

Repair to satisfy $A_i$

$cex$ analysis

Real cex $t$

Verification

Property

Does $M_1$ composed with $M_2$ satisfy the property?
The composition is too big - look for a small assumption $A$ such that $M_1$ composed with $A$ satisfy the property, and $A$ models $M_2$.

$M_1$ composed with $A$ satisfies the property!

But: $A$ does not model $M_2$ correctly.

A violating trace is:

$t = read(pass), pass > 999, enc(pass), pass2 := (pass - 11) \cdot 2, pass2 < 100000$ for the initial value $pass = 90000$.

We thus repair $M_2$.

We wish to learn a new constraint $C$ such that:

$C \land pass > 999 \land pass2 = (pass - 11) \cdot 2 \rightarrow pass2 < 100000$

$C$ is over the input variables of the system, here $- pass$.

$C := \forall pass2: pass > 999 \land pass2 = (pass - 11) \cdot 2 \rightarrow pass2 < 100000$

After quantifier elimination & simplification: $C = pass < 50012$.

Repaired $M_2$, constructed using automata learning:

1: pass = readInput;
2: while (pass ≤ 999 or pass ≥ 50012)
3:     pass = readInput;
4: pass2 = encrypt(pass);
5: return pass2;