

Static approximation of error propagation in program variables

Greg Bronevetsky Sigmund Cherem Radu Popovici

CS 717

17th December 2004

Error propagation

- Bit flips in memory
- Bits vs. variable model
 - Bit level handles error masking
 - Variable level is more efficient
- Static vs. dynamic
 - Estimating probability of taking a conditional branch
 - Estimating values lifespan (ϵ over time)
- Data vs. code
 - Data subject to bit flips
 - Possible erroneous control flow decisions
 - Code corruption, too hard

Our approach

Design decisions

- Variable model
- Assume no code corruption
- Confidence on CFG
- 1 statement \approx 1 time unit
- No profiling (yet) decisions
 - β_i probability of taking true branch
 - Example: 50% for if statements, 80% for loops
- Core language: variables, arithmetic operations, unstructured control flow.

Variable model

- Events x^{l_i} : x has an error when executing l_i
- Probability $P(x^{l_i}) = \llbracket x \rrbracket_i$
- Correct executions
 - c_i : probability of executing l_i correctly
 - $\llbracket x \rrbracket_i^c$: probability of error in x on the right path
- Wrong executions
 - w_i : probability of executing l_i wrongfully
 - $\llbracket x \rrbracket_i^w$: probability of error in x on a wrong path
- Finally

$$\llbracket x \rrbracket_i = c_i \cdot \llbracket x \rrbracket_i^c + w_i \cdot \llbracket x \rrbracket_i^w$$

Correct executions

- Transfer functions for error flow on each statement
- Example: assignments

$$l_1: \quad a = b + c$$

$$\llbracket a \rrbracket_2^c = \llbracket b | \neg c \rrbracket_1^c \cdot (1 - \llbracket c \rrbracket_1^c) + \llbracket c | \neg b \rrbracket_1^c \cdot (1 - \llbracket b \rrbracket_1^c) + \llbracket b | c \rrbracket_1^c \cdot \llbracket c \rrbracket_1^c$$

When B^h and C^h are independent:

$$\begin{aligned} \llbracket a \rrbracket_2^c &= \llbracket b \rrbracket_1^c \cdot (1 - \llbracket c \rrbracket_1^c) + \llbracket c \rrbracket_1^c \cdot (1 - \llbracket b \rrbracket_1^c) + \llbracket b \rrbracket_1^c \cdot \llbracket c \rrbracket_1^c \\ &= \llbracket b \rrbracket_1^c + \llbracket c \rrbracket_1^c - \llbracket b \rrbracket_1^c \llbracket c \rrbracket_1^c \end{aligned}$$

Error correlation

Correlation example

$$l_1: a = x + y;$$

$$l_2: b = y + z;$$

$$l_3: c = a + b;$$

	x	y	z	a	b	c
x	1			$\frac{[x]_3^c}{[a]_3^c}$		
y		1		$\frac{[y]_3^c}{[a]_3^c}$	$\frac{[y]_3^c}{[b]_3^c}$	
z			1		$\frac{[z]_3^c}{[b]_3^c}$	
a	1	1		1	$\frac{([y]_3^c)^2}{[a]_3^c([b]_3^c)^2}$	
b		1	1	$\frac{([y]_3^c)^2}{[b]_3^c([a]_3^c)^2}$	1	
c						1

Random flips over time

Aging function

- Non-decreasing function in probability of error

$$\rho_{\varepsilon}(p) = p + \varepsilon - p \cdot \varepsilon$$

Wrong executions

- δ_b : probability of taking the wrong branch on condition b
- $\beta_b \bar{\delta}_b$: probability of taking true branch correctly
- $\bar{\beta}_b \delta_b$: probability of taking true branch incorrectly

Example

```
l0:  if (a == 2)
```

```
l1:      b = 3;
```

```
      else
```

```
l2:      c = 2;
```

$$c_1 = \beta_0 \bar{\delta}_0 \quad w_1 = \bar{\beta}_0 \delta_0$$

$$[[b]]_1^c = 0 \quad [[b]]_1^w = 1 \quad [[b]] = w_1$$

Wrong executions

- δ_b : probability of taking the wrong branch on condition b
- $\beta_b \bar{\delta}_b$: probability of taking true branch correctly
- $\bar{\beta}_b \delta_b$: probability of taking true branch incorrectly

Example

```
l0:  if (a == 2)
```

```
l1:      b = 3;
```

```
      else
```

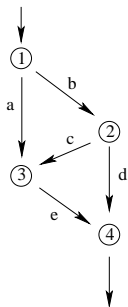
```
l2:      c = 2;
```

$$c_1 = \beta_0 \bar{\delta}_0 \quad w_1 = \bar{\beta}_0 \delta_0$$

$$[[a]]_1^c = \rho_\epsilon \circ [[a]]_0^c \quad [[a]]_1^w = \rho_\epsilon \circ [[a]]_0$$

Alternative basic blocks

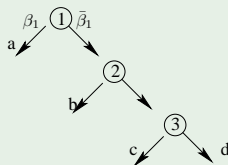
- Let A be a basic block, basic block B is alternative to A if:
 - $A \neq B$
 - There is no post-dominator of A, that dominates B



a alternatives are
b,c and d, but not
e

Correct and incorrect choices

Example



w_d : probability of reaching d incorrectly is

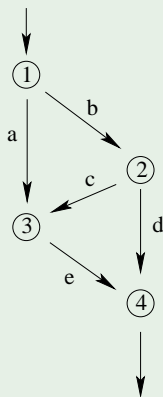
$$w_d = \beta_1 \delta_1 + \bar{\beta}_1 \bar{\delta}_1 \beta_2 \delta_2 + \bar{\beta}_1 \bar{\delta}_1 \bar{\beta}_2 \bar{\delta}_2 \beta_3 \delta_3$$

c_d : probability of reaching d correctly is

$$c_d = \bar{\beta}_1 \bar{\delta}_1 \bar{\beta}_2 \bar{\delta}_2 \bar{\beta}_3 \bar{\delta}_3$$

Propagating choices

Example



Intuition

- Choice errors inside component are incorporated in variable error probabilities
- Any choice (right or wrong) inside component, leads to 4
 - $\Rightarrow c_4 = c_0$
 - $\Rightarrow w_4 = w_0$

Incorrect values

- A : set of paths alternative (to d) starting from a
- A path $p \in A$ is a contiguous sequence of blocks from a to the post-dominator of a and d .
- π_p : weight on path p

$$\pi_p = \prod_{i \in p} \begin{cases} \beta_{c_i} & \text{if } i \text{ is true choice of } c_i \\ \bar{\beta}_{c_i} & \text{o.w.} \end{cases}$$

Incorrect values

- $ndef_A(x)$: probability of not assigning x in any path of A
- $def_i(x)$: Variable x is defined in basic block i
- $def_i^l(x)$: Variable x is defined in basic block i on or before instruction l .

$$ndef_A(x) = \sum_{p \in A} \pi_p \left(\prod_{i \in p} \bar{def}_i(x) \right)$$

Intuition

- If x is assigned a value, then $\llbracket x \rrbracket_j^w = 1$
- If x is not assigned a value, then
 - If there is an alternative path assigning x , then $\llbracket x \rrbracket_j^w > 0$
 - If no alternative path assigns x , then $\llbracket x \rrbracket_j^w$ includes just aging of the value.

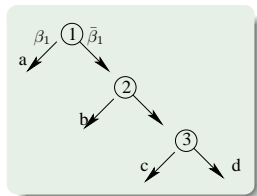
Incorrect values example

Q_d : The probability of having no assignments to x on alternative paths

$$Q_d = \frac{\beta_1 \delta_1 \text{ndef}_A(x) + \bar{\beta}_1 \bar{\delta}_1 \beta_2 \delta_2 \text{ndef}_B(x) + \bar{\beta}_1 \bar{\delta}_1 \bar{\beta}_2 \bar{\delta}_2 \beta_3 \delta_3 \text{ndef}_C(x)}{W_d}$$

Q'_d : The probability of having no assignments to x both on alternative paths and until statement l starting from d

$$Q'_d = Q_d \bar{\text{def}}'_d(x)$$



Finally

The probability of error on x , on a wrong execution of d is

$$\llbracket x \rrbracket_l^w = (1 - Q_d^l) + Q_d^l \cdot \rho_\varepsilon \circ \llbracket x \rrbracket_{l-1}^w$$

Merge points and loops

Merge operation

- Let a, b be the last program point of incoming branches
- α_a : probability of reaching the merge point from a

$$\llbracket x \rrbracket_{a \sqcup b}^c = \alpha_a \llbracket x \rrbracket_a^c + \alpha_b \llbracket x \rrbracket_b^c$$

$$\llbracket x \rrbracket_{a \sqcup b}^w = \alpha_a \llbracket x \rrbracket_a^w + \alpha_b \llbracket x \rrbracket_b^w$$

Convergence guarantees

- Transfer functions are monotonic
- Probability space is bounded by 1

Reducing vulnerability

Local replication

- Include assertions to model
- Insert new assertions (replicated code) to reduce error probability

Checkpointing

- Assume values to be valid
- Automatic placement of checkpoints
- Runtime decision on potential checkpoints according to error probability

Private memory

- Move error prone variables to secure memory
- Private variables don't age