

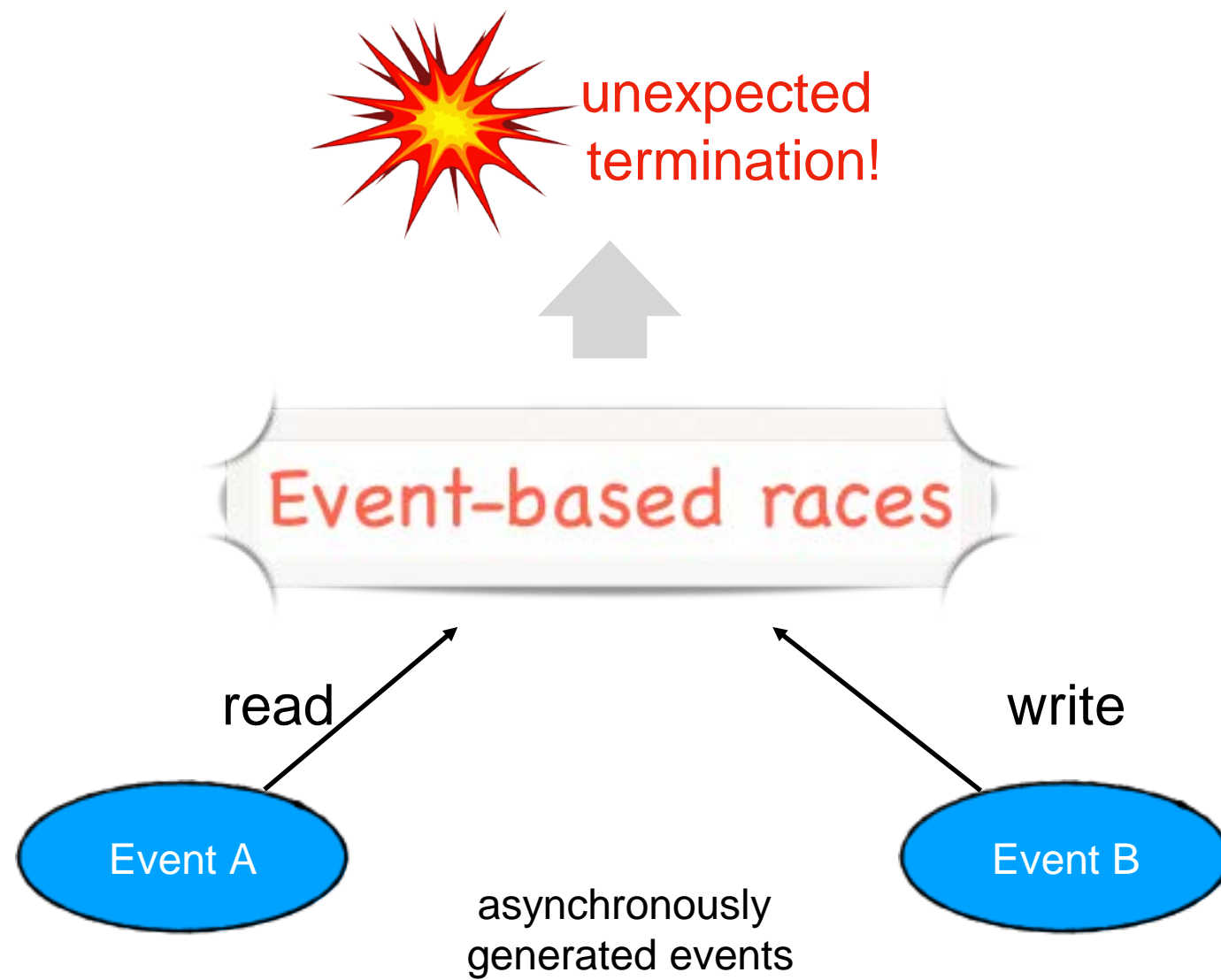
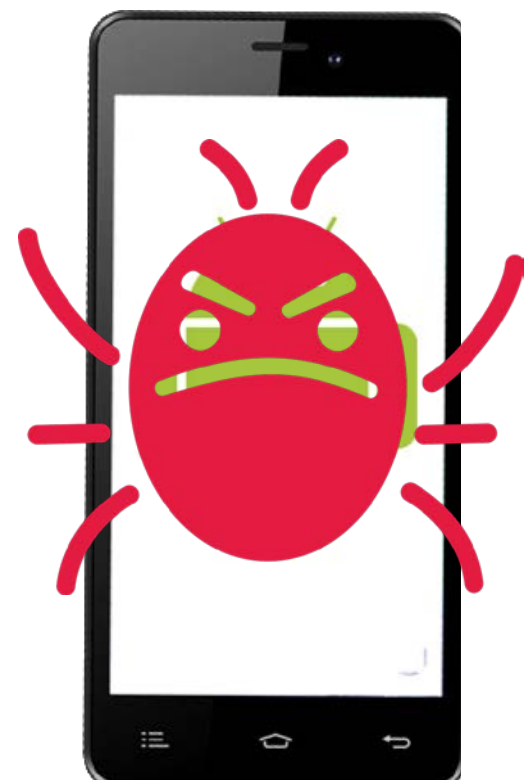
# Exposing Android Event-Based Races by Selective Branch Instrumentation

Diyu Wu, Dongjie He, Shiping Chen and Jingling Xue

The University of New South Wales, Australia

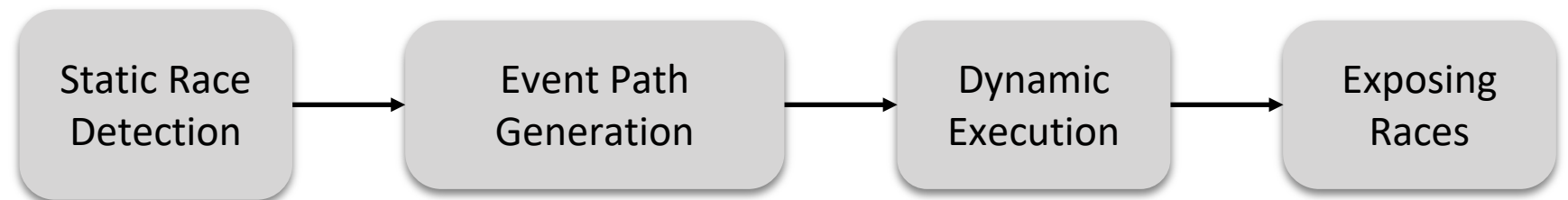


ISSRE 2020





Hybrid Analysis



**complicated conditionals!**

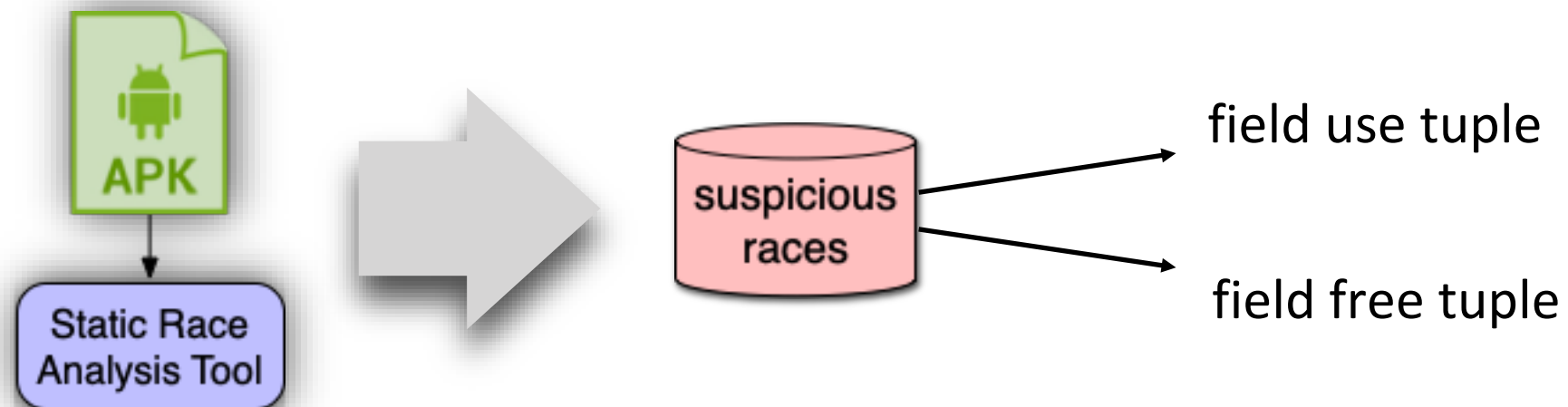
- Selective Branch Instrumentation:

`if(condition)`  $\longrightarrow$  `if(true/false)`

1. non-existent races

2. crashes

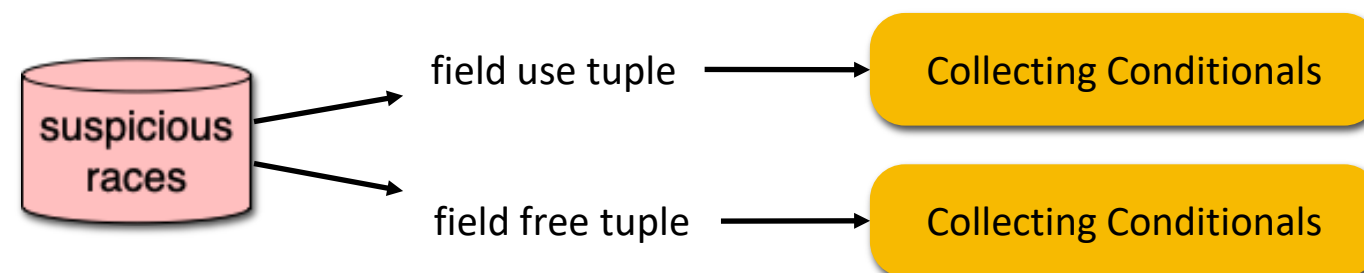
- Statically detect suspicious races.



a suspicious race:  $\langle \tau_u, \tau_f \rangle$

a tuple:  $\tau = \langle s, e, ctx \rangle$

- Collecting Conditionals

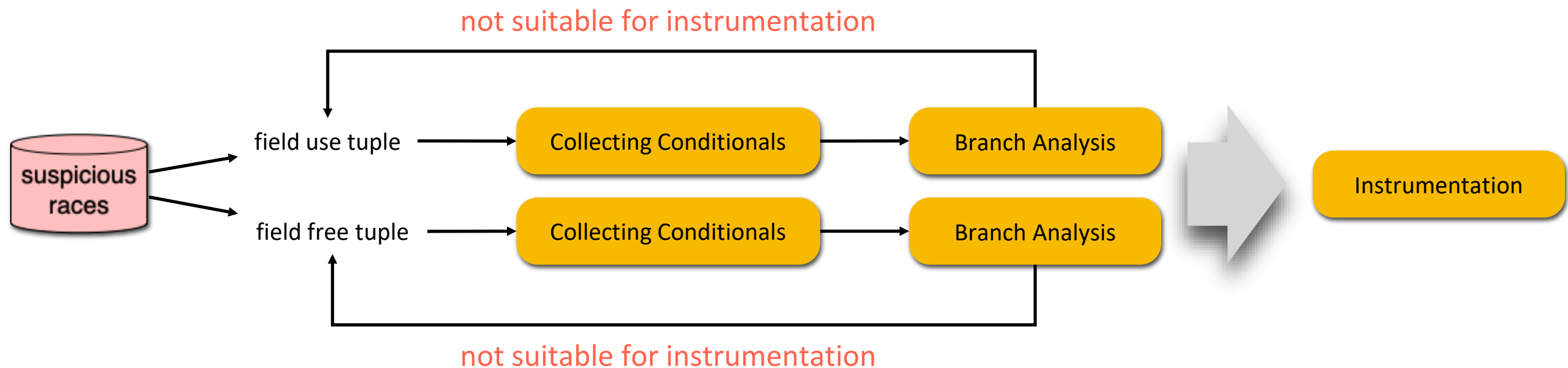


- Branch Analysis



- Definition tracing
- Value range analysis : possible values of variables.
- select conditionals : could be replaced with true/false.
  - Safety check
  - Satisfiability check

- Instrumentation



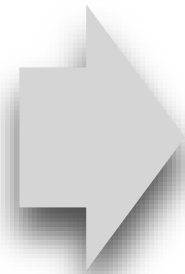
a tuple:  $\tau = \langle s, e, ctx \rangle$




- Dynamic Execution



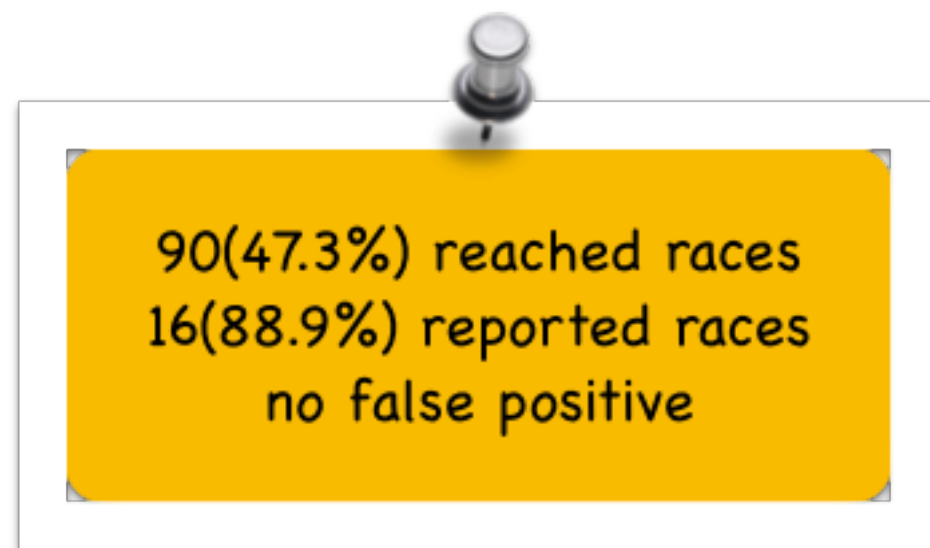
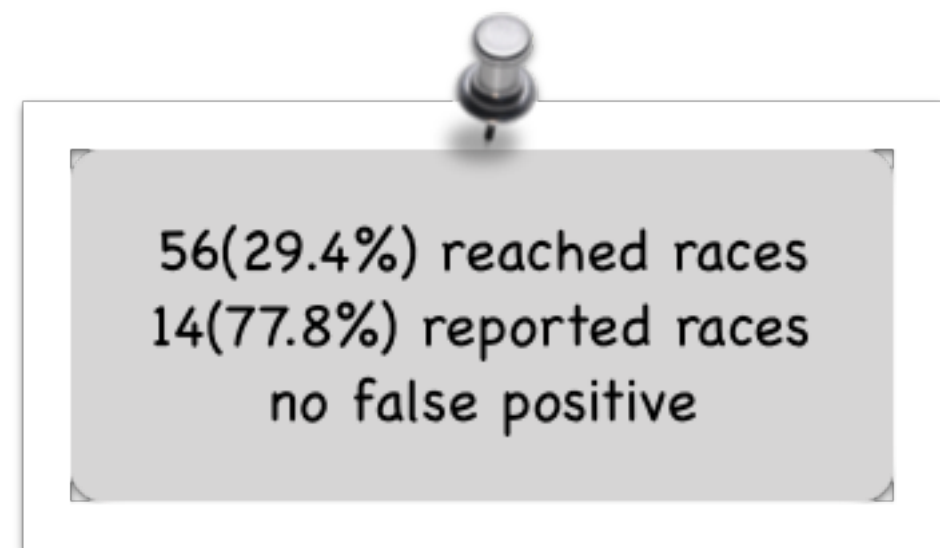
instrumented  
app



results

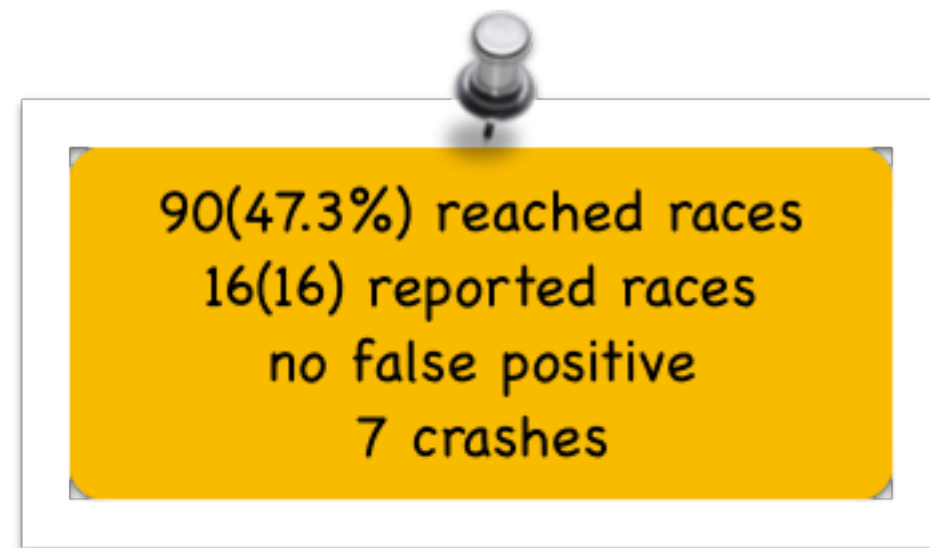
-  25 real world Android apps
  - 190 suspicious races
  - 18 true races
- Compare SIEVE with two baseline tools
  - Sieve-ZB: non-instrumentation
  - Sieve-FB: full-instrumentation

- Sieve vs Sieve-ZB (non-instrumentation)
  - Reached races: reached racy statements in order
  - Reported races: reached races with NPE

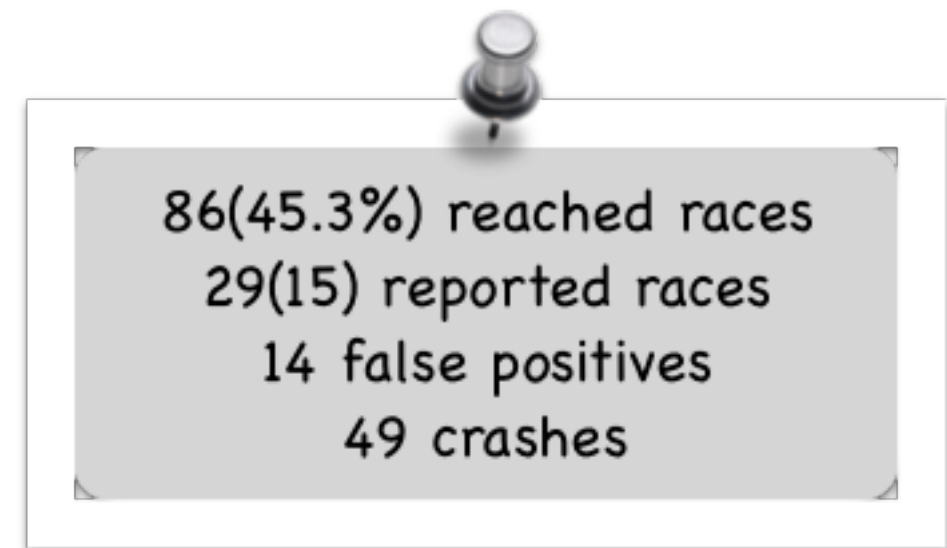
**Sieve****Sieve-ZB**

more effective!

- Sieve vs Sieve-FB (full-instrumentation)



**Sieve**



**Sieve-FB**

less false alarms and crashes!

- SIEVE: Selective branch instrumentation
  - ★ Expose event-based races more effectively
  - ★ Reduce negative ramifications of instrumentation

**THANK YOU!**