Project #2

Static Buffer Overrun Analysis

Hagai Attias 036860682

Maayan Arbiv 037117132

Ari Zigler 036853141

General description

- **Static** buffer overrun analyzer for java programs, based on *Soot* infrastructure.
- Uses **Jimple** as the intermediate language.
- Implements **intraprocedural** analysis using **Interval** abstraction.
- A **May** problem, information flows **forwards**.

Implementation

- **Arithmetic** operators: OP+, OP-, OP*, OP/, Negation.
- **Boolean** operators: Pair<, Pair<=, Pair==, Pair!=.
- Uses Soot's `ForwardBranchedFlowAnalysis` to propagate different information for the **branch** and **fall-through** routes.
- Distinguishes between a **definite** illegal access and a **potential** one.
- Applying **Delayed widening** to improve precision.

3-Phase analysis

Pass #1

Compute for each program point the set of locals defined before this point, with their corresponding interval.

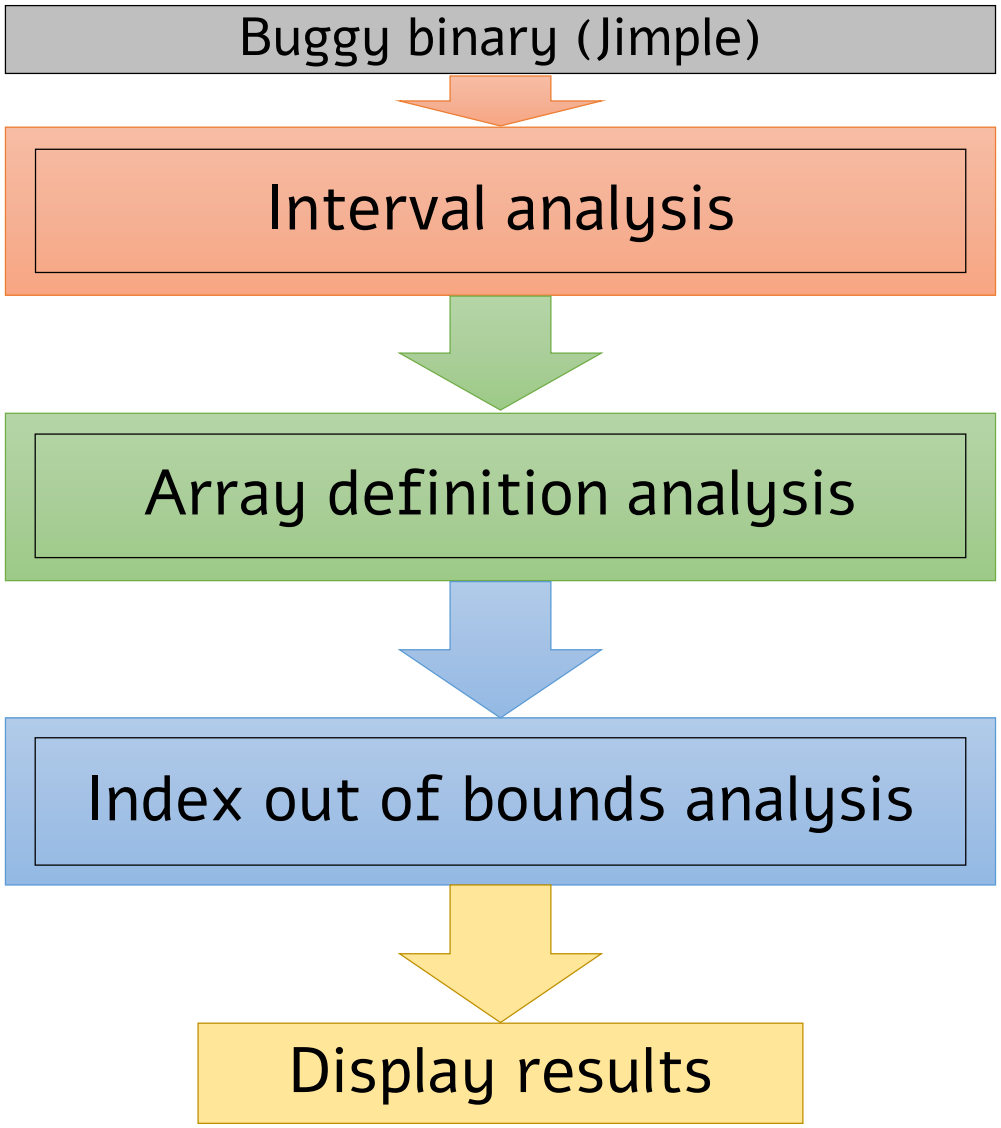
Pass #2

Compute for each program point the set of arrays defined before this point, and their corresponding size interval, determined from phase #1.

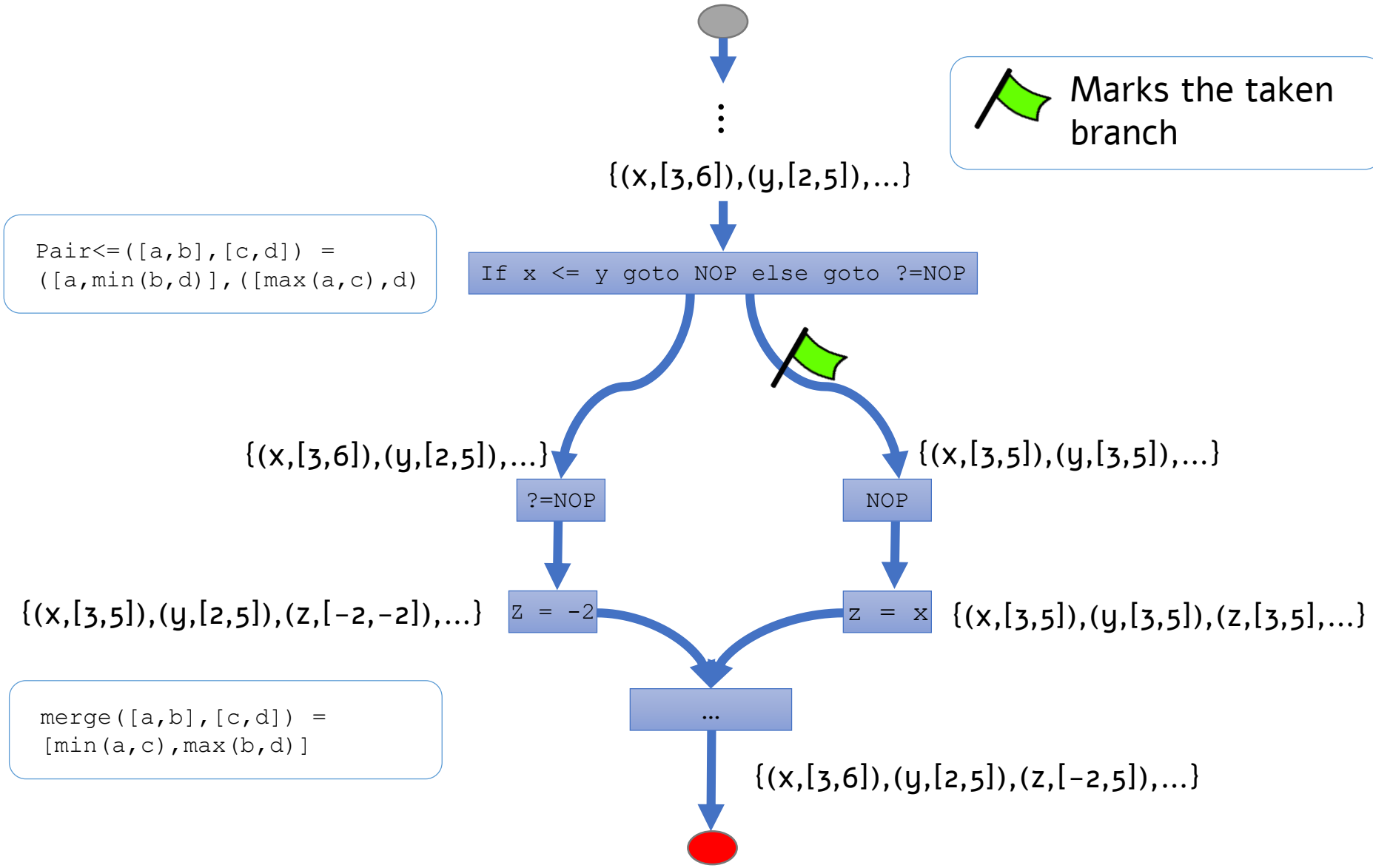
Pass #3

For each array access in the program, check whether it is a possible illegal access, based on the array size interval from phase #2 and index size interval from phase #1.

Output is displayed using the *Soot* plugin, displaying an ***SA*** mark for each illegal access line, along with the information message displayed as a tooltip.



Example: Enforcing Pair<= and merge



Questions?