Gradual Programming: Bridging the Semantic Gap

Bor-Yuh Evan Chang Amer Diwan Jeremy G. Siek
University of Colorado, Boulder

PLDI FIT 2009

Have you noticed a time where your program is not optimized where you expect?

Observation: A disconnect between programmer intent and program meaning

Problem: Tools (IDEs, checkers, optimizers) have no knowledge of what the programmer cares about

… hampering programmer productivity, software reliability, and execution efficiency

Example: Iteration Order

```
class OpenArray
    private Dou
    public boolean cont s(Object lookFor) {
        for (i = 0; i < data.length; i++) {
          if (data[i].equals(lookFor)) return true;
        }
        return false;
    }
}
```

Must specify an iteration order
even when it should not matter

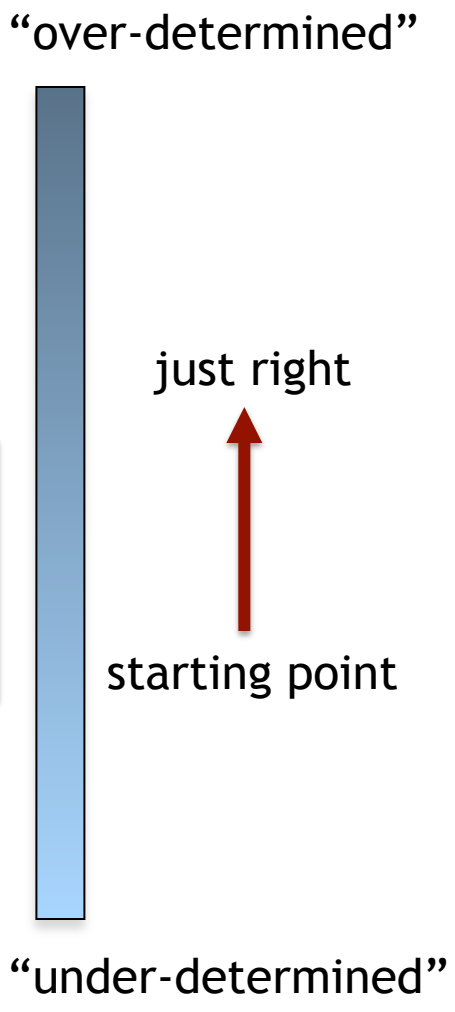
Compiler cannot choose a different iteration order (e.g., **parallel**)

Wild and Crazy Idea: Use Non-Determinism

- Programmer starts with a potentially non-deterministic program
- Analysis identifies instances of "under-determinedness"
- Programmer eliminates "under-determinednes

Question: What does this mean? Is it "under-determined"?
Response: Depends, is the iteration order important?

```
class OpenArray e
    private Doub
    public boole
        ∀ i ∈ 0 .. data.length-1 {
            if (data[i].equals(lookFor)) return true;
        }
        return false;
    }
}
```



Let's try a few program variants

```
public boolean contains(Object lookFor) {
  for (i = 0; i < data.length; i++) {
    if (data[i].equals(lookFor)) return true; }
  return false;
}
```

```
public boolean contains(Object lookFor) {
  for (i = data.length-1; i >= 0; i--) {
    if (data[i].equals(lookFor)) return true; }
  return false;
}
```

```
public boolean contains(Object lookFor) {
  parallel_for (0, data.length-1) i => {
    if (data[i].equals(lookFor)) return true; }
  return false;
}
```

Do they compute the same result?

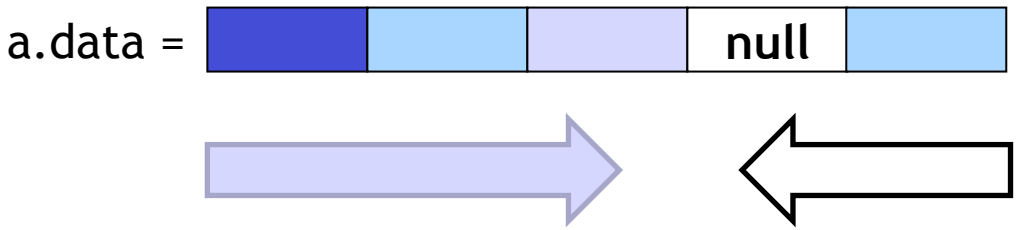
Approach: Try to verify equivalence of program variants up to a specification

Yes ➔ Pick any one
No ➔ Ask user

What about here?

Surprisingly, analysis says no. Why?

Exceptions!



a.contains( )

left-to-right iteration
returns true

right-to-left iteration
throws NullPointerException

Need user interaction to refine specification that captures programmer intent

Scary Proposal?

- "Fix semantics per program": Abstract constructs with many possible concrete implementations
- Apply program analysis to find inconsistent implementations
- Interact with the user to refine the specification
- Language designer role can enumerate the possible implementations

Bridging the Semantic Gap