State of the Practice Code Search

How do I do X?

How have others done X?

Keywords:
- power function
- compute \( x^n \)
- ....

Search

Results

Which results are relevant???

How does this sample code behave?

Hasn't this problem been solved before???
**Proposed Code Search**

**Lightweight Specifications:**
- \( \text{in} = 3, 2 \)
- \( \text{out} = 9 \)
- or
- \( \text{in} = 4, 3 \)
- \( \text{out} = 64 \)
- ...

**Results**

```java
int addSixorSixty(int x, int y) {
  if (x == 3)
    return x + 6;
  else
    return x + 60;
}

double power(double x, double n) {
  double pow = Math.pow(x, n);
  return pow;
}
```

**Wow! These results all behave as specified!**

Sometimes there are **too many** results...

If there are no **exact** matches, some are close enough.
Research Contribution

- An approach for semantic search via lightweight specs
- Uses an SMT solver to solve the search
- Promotes reuse of repository code
- Benefits over state-of-the-art semantic search:
  - Cost of Query: I/O is easier to write than formal specifications
  - Cost of Search: candidate code is not executed
- Benefits over state-of-the-practice syntactic search:
  - Relevance of Results: all responses behave as specified, and close-enough matches can be identified
Challenges

• **Encoding** programs as constraints

• **Relaxing** constraints when no exact matches exist

• **Evaluating** the cost of our search vs. traditional search from two perspectives: efficiency and effectiveness