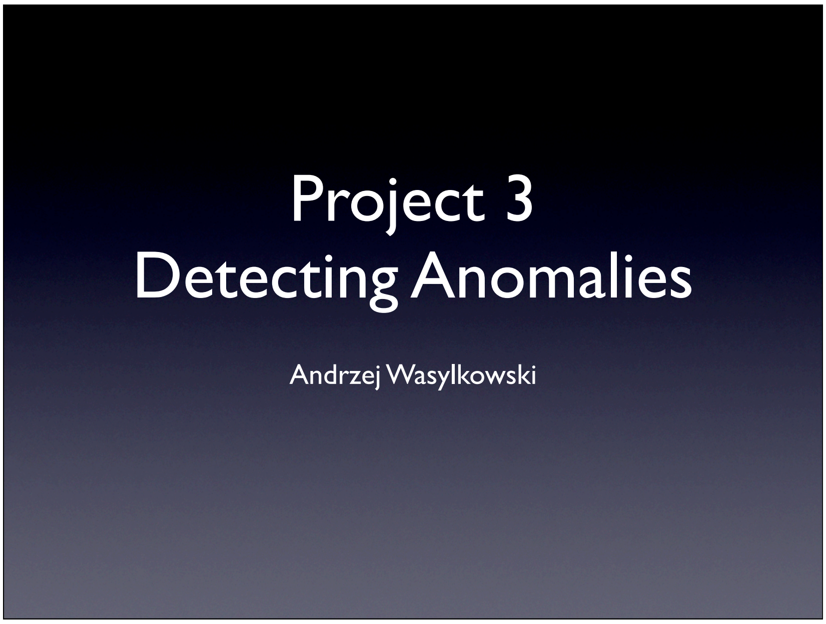
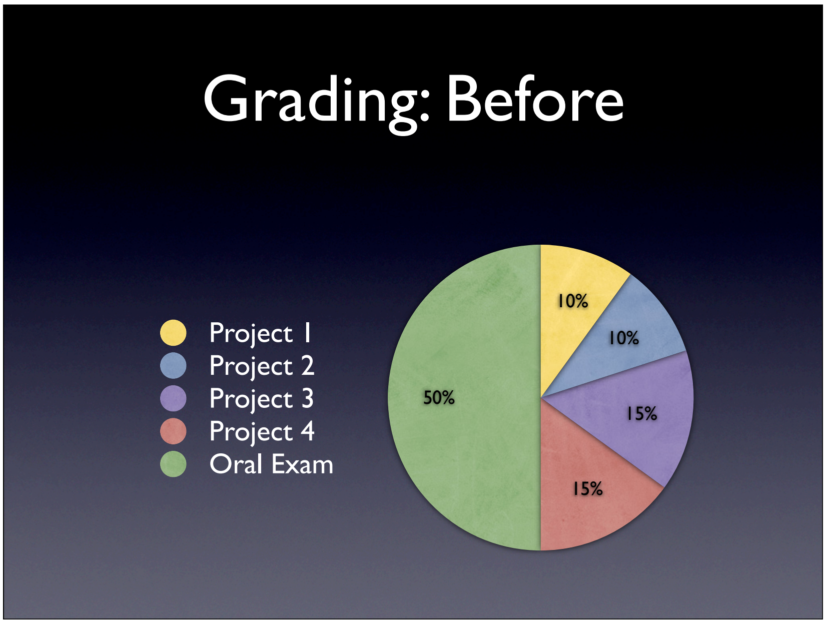
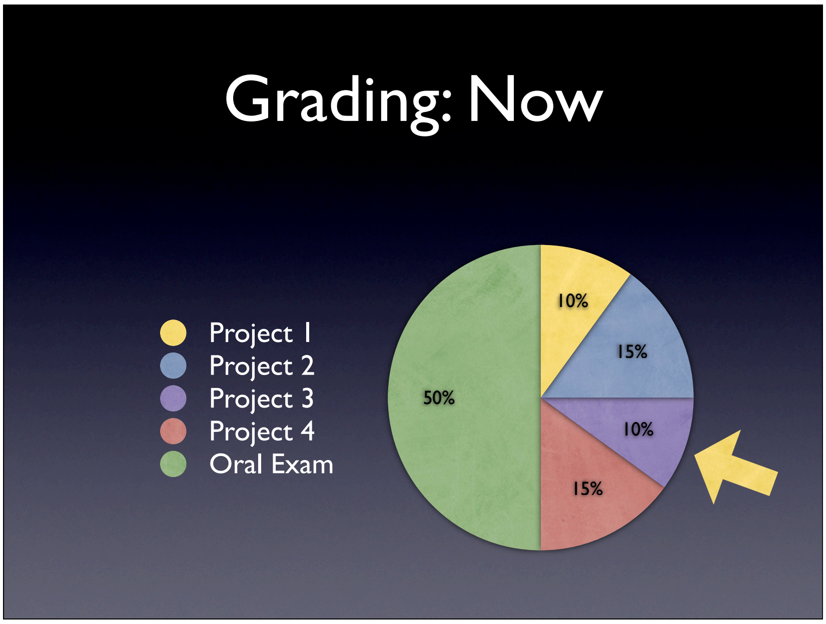


Project 3
Detecting Anomalies

Andrzej Wasylkowski



Grading: Before



Grading: Now

Your Task

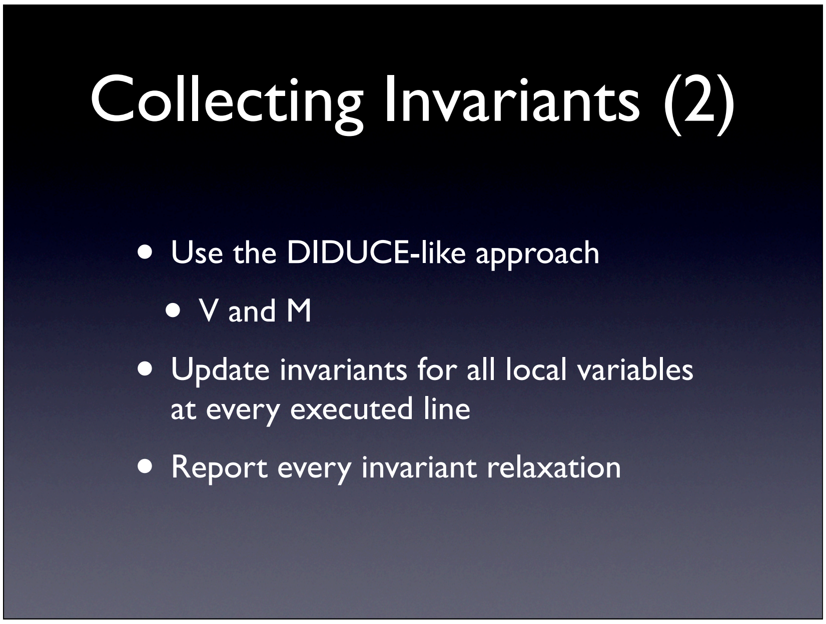
- Collect DIDUCE-like invariants
- Report invariants' violations
- Use invariants' violations to find the reason for the failure in XMLProc
- Implement extensions

Tracing Program Run

1. Set the tracing function
    - Trace all lines in the program
    - Ignore lines in the external libraries
2. Invoke the program to be analyzed
3. Output invariants' violations

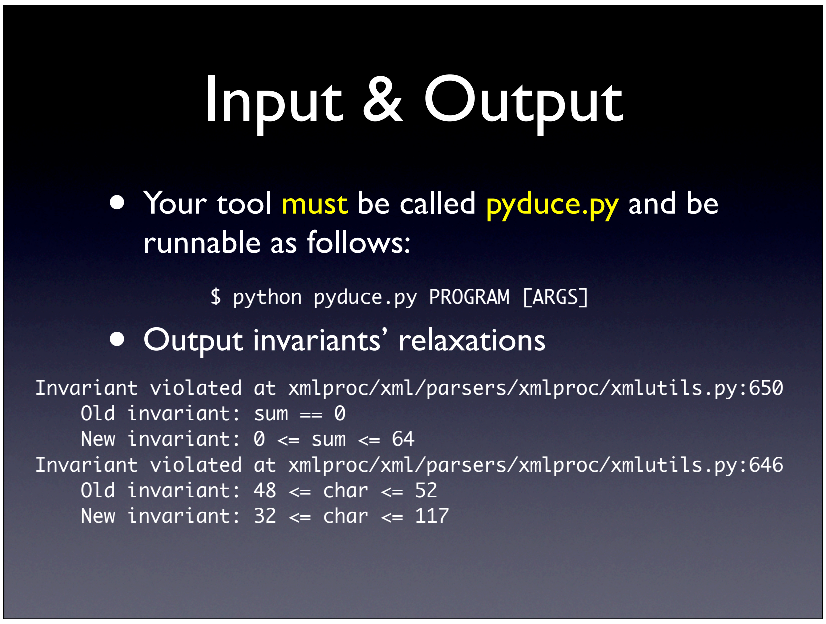
Collecting Invariants (1)

- Invariants associated with variables and source code locations
    - "x is always between 2 and 4 in file tooldir/main.py at line 8"
- Invariants for booleans, integers, longs, and instances of classes
    - Convert to integers; see the handout



Collecting Invariants (2)

- Use the DIDUCE-like approach
  - V and M
- Update invariants for all local variables at every executed line
- Report every invariant relaxation



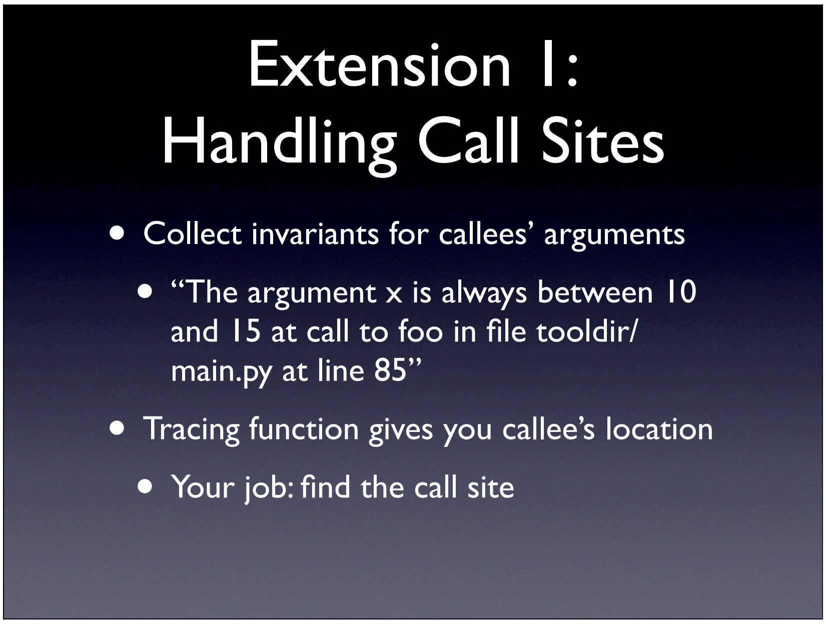
Input & Output

- Your tool must be called pyduce.py and be runnable as follows:

```
$ python pyduce.py PROGRAM [ARGS]
```

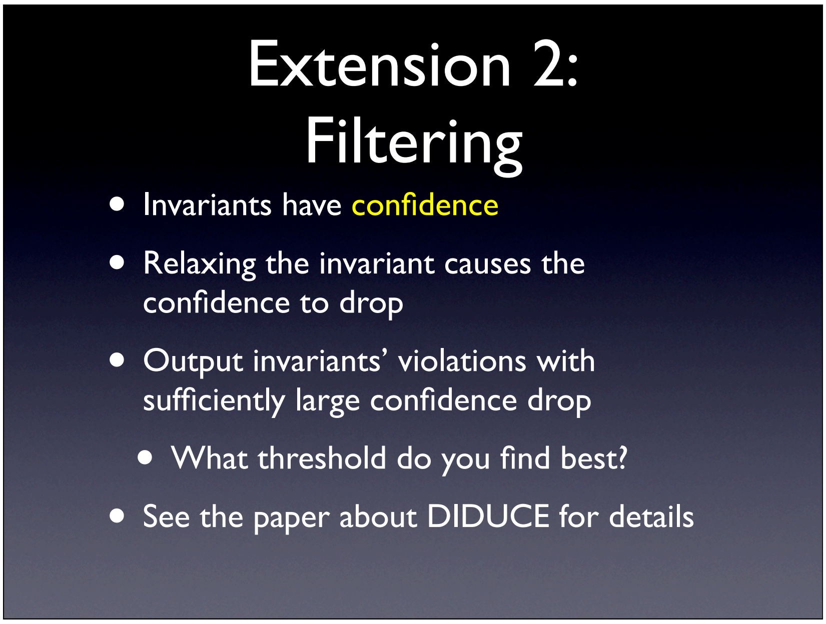
- Output invariants' relaxations

```
Invariant violated at xmlproc/xml/parsers/xmlproc/xmlutils.py:650
    Old invariant: sum == 0
    New invariant: 0 <= sum <= 64
Invariant violated at xmlproc/xml/parsers/xmlproc/xmlutils.py:646
    Old invariant: 48 <= char <= 52
    New invariant: 32 <= char <= 117
```



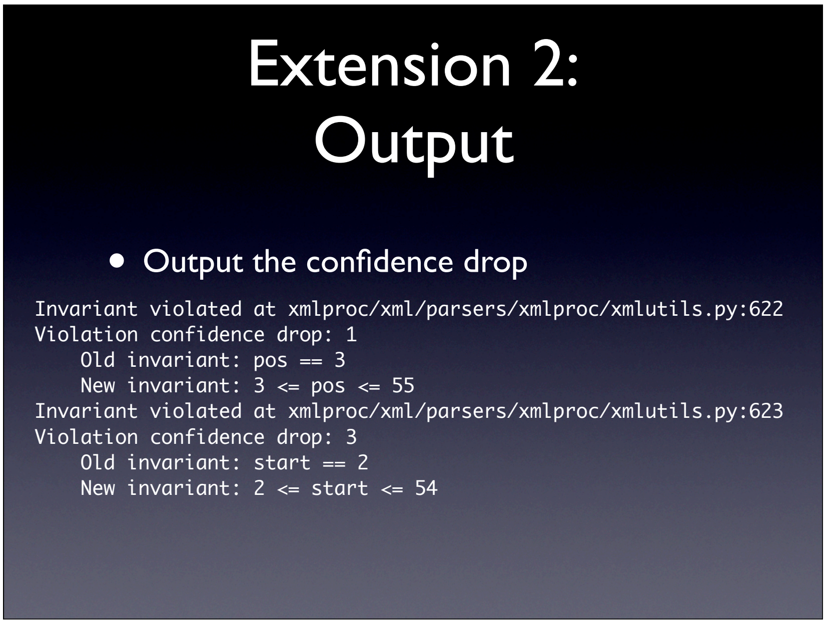
Extension 1: Handling Call Sites

- Collect invariants for callees' arguments
  - "The argument x is always between 10 and 15 at call to foo in file tooldir/main.py at line 85"
- Tracing function gives you callee's location
  - Your job: find the call site



Extension 2: Filtering

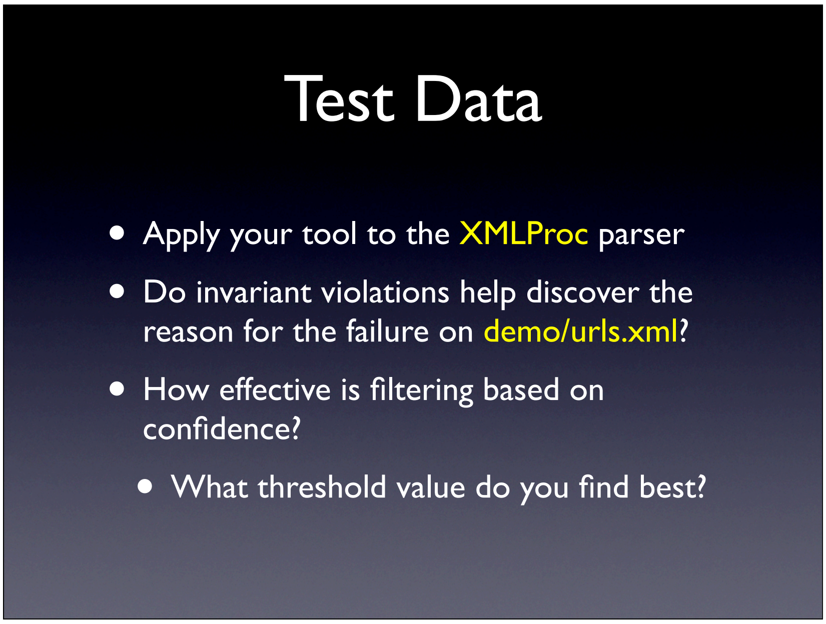
- Invariants have confidence
- Relaxing the invariant causes the confidence to drop
- Output invariants’ violations with sufficiently large confidence drop
  - What threshold do you find best?
- See the paper about DIDUCE for details



Extension 2: Output

- Output the confidence drop

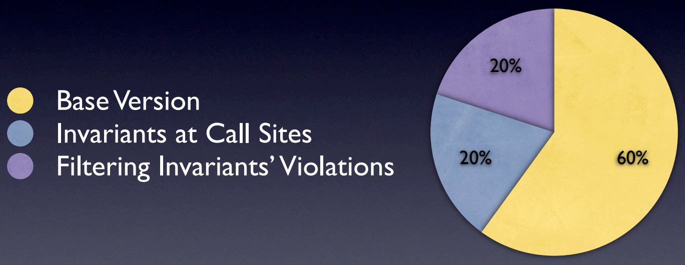
```
Invariant violated at xmlproc/xml/parsers/xmlproc/xmlutils.py:622
Violation confidence drop: 1
    Old invariant: pos == 3
    New invariant: 3 <= pos <= 55
Invariant violated at xmlproc/xml/parsers/xmlproc/xmlutils.py:623
Violation confidence drop: 3
    Old invariant: start == 2
    New invariant: 2 <= start <= 54
```



Test Data

- Apply your tool to the XMLProc parser
- Do invariant violations help discover the reason for the failure on demo/urls.xml?
- How effective is filtering based on confidence?
  - What threshold value do you find best?

Project Grading



Submission

- 2009-01-09 23:59
- Send .zip archive to: wasylkowski@.cs.uni-saarland.de
  - Subject should start with [Project 3]
  - Input and output exactly as prescribed
  - Source code should be documented