LapsPython

Extend LAPS to synthesize Python/R

Christopher Brückner

29.08.2022

Important Links

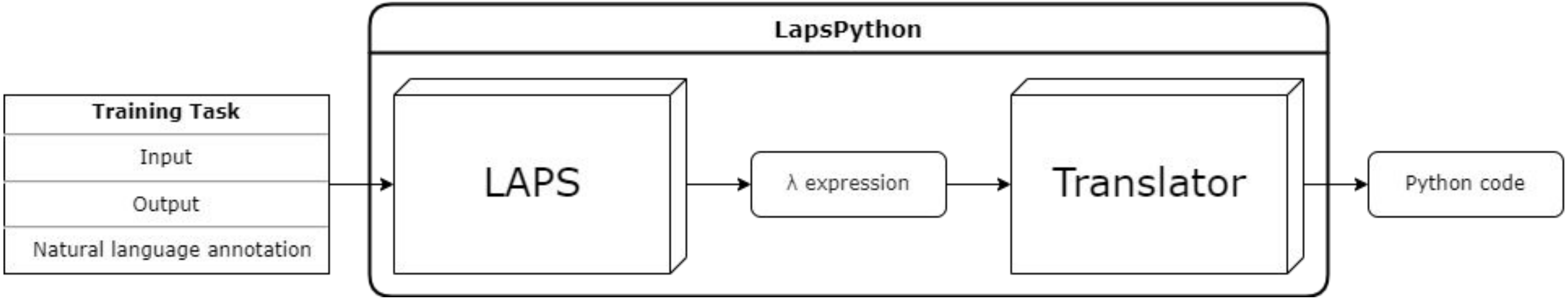
- Git Repository:
  https://github.com/pvs-hd-tea/LapsPython

- Documentation:
  https://pvs-hd-tea.github.io/LapsPython

  Including developer guide, API reference, accounting, metrics, presentations.

Goals & Results

Goals



- Create a rule-based translator from λ-calculus to Python or R
- Target useful domains, e.g., data processing, string processing
- Maybe address different target languages

Goals: What did I achieve?

- Translator: Working, but sometimes small bugs in translations
    - Example: Slide 6

- Target Domain: re2
    - Pre-implemented
    - Combines string processing (regex) and list processing

- Target Languages: Python, R
    - Python is obligatory for LAPS to work
    - Extended by manually translating the Python primitives to R

Evaluation Metrics

- Small checkpoint: 3 iterations of LAPS
  - 75 synthesized programs in 18 tasks
  - 100% bug-free translated by LapsPython

- Large checkpoint: Provided by the authors
  - 1646 synthesized programs in 346 tasks
  - 27 tasks (8%) solved with bug-free translations

- Bugs typically easy to detect and fix
  - Usually calls to invented primitives
  - Example: Next slide

Example: Flawed Translation

Task: if there is any letter, add v after that

Program: `(λ (f8 $0 (λ (_rconcat $0 _v))))`

Translation:

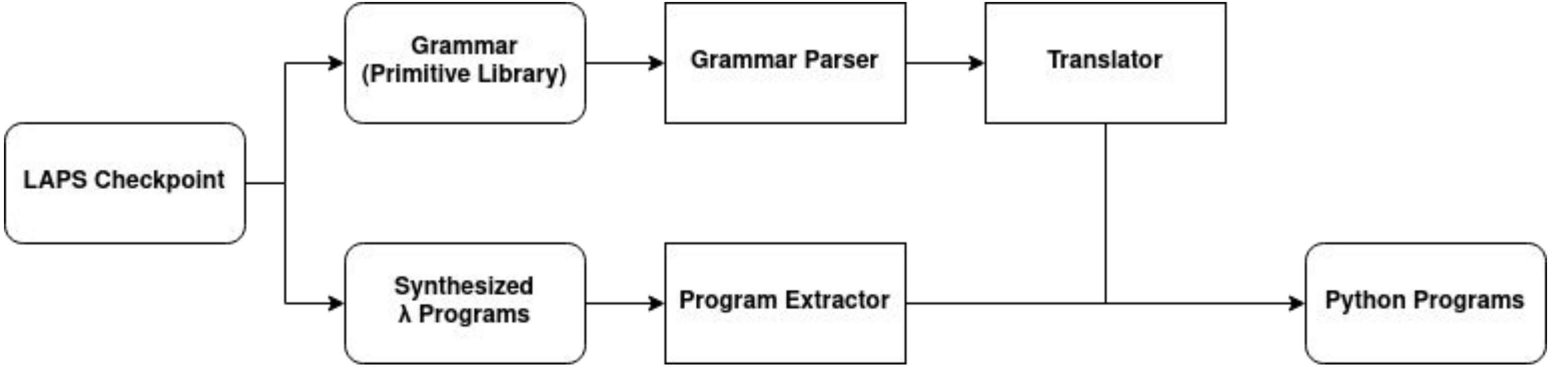
```
def f8(arg1, arg2):
    rsplit_1 = __regex_split('.', arg2)
    map_1 = list(map(arg1, rsplit_1))
    return "".join(map_1)


def re2 train 41 if there is_any_letter_add_v_after_that(arg1):
    rconcat_1 = arg1 + 'v’
    return f8(f8_0, arg1, lambda lx: lx + 'v')
```

Fix: `return f8(lambda lx: lx + 'v', arg1)`

Architecture

LapsPython Pipeline



GrammarParser: Read Python/R implementations, parse arguments, reformat code

ProgramExtractor: Read tasks descriptions, programs, input/output examples

Translator: Traverse program tree, substitute primitives with Python/R code

Example: Code Reformatting

Primitive implemented by LAPS:

```
def _rsplit(s1) : return lambda s2: __regex_split(s1, s2)
```

□

Primitive used by LapsPython:

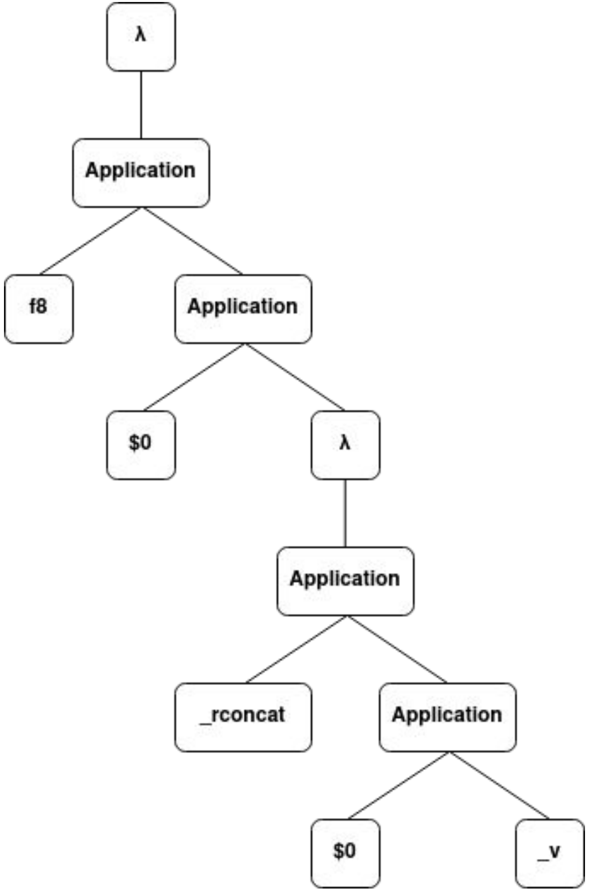
```
def _rsplit(s1, s2):
    return __regex_split(s1, s2)
```

Example: Program Tree

Task: if there is any letter, add v after that

Program: `(λ (f8 $0 (λ (_rconcat $0 _v))))`

Translation: `f8(lambda lx: lx + 'v', arg1)`



Code Quality & Statistics

Code Quality

- Continuous Integration Pipeline:
    - Unit Tests (pytest + pytest-cov)
    - Static Typechecks (mypy)
    - Linting (Flake8)

- Flake8 is used with extensions to
    - enforce stricter conventions in compliance with modern Python and Clean Code
    - simplify complex code
    - prevent bugs and security issues

- PEP-8 score according to Flake8: 100%

Code Statistics

- Lines of Code
  - Main Code: 775
  - Test Code: 430
  - Ratio: 1.8

- Test Coverage
  - Test Cases: 50
  - 94% coverage total, ranging from 88% to 100% per module
  - Out of 666 statements, 43 are missed (many unreachable)
    - Exception handling for fixed bugs
    - Handling of special cases not present in current dataset

Software Process

- Scrum: Sprints were very helpful
  - Helps with project planning
  - Regular insights into what is achievable and what isn’t
  - In general, I learned a lot about working with GitHub

- Team Work: Unfortunately zero

- Biggest Issue: Original project plan was designed for 2 people
  - Some interesting ideas had to be scrapped, e.g., more domains
  - Lack of different perspectives on problems
  - Huge workload

Special Difficulties

- LAPS is really difficult to get running
  - LapsPython will work anyway by using LAPS checkpoints ☺️
  - It can still be injected into the LAPS code

- Different approaches for similar problems required
  - Python primitives can be read from memory (`inspect` module)
  - R primitives need to be manually parsed from the source file

- Translation difficult and hard to debug
  - Program trees are not always simple to traverse
  - Lots of special cases to handle
  - LAPS does not give you insights into intermediate results
  - Translation module was rewritten twice and is still flawed

1-Click-Demo