LapsPython

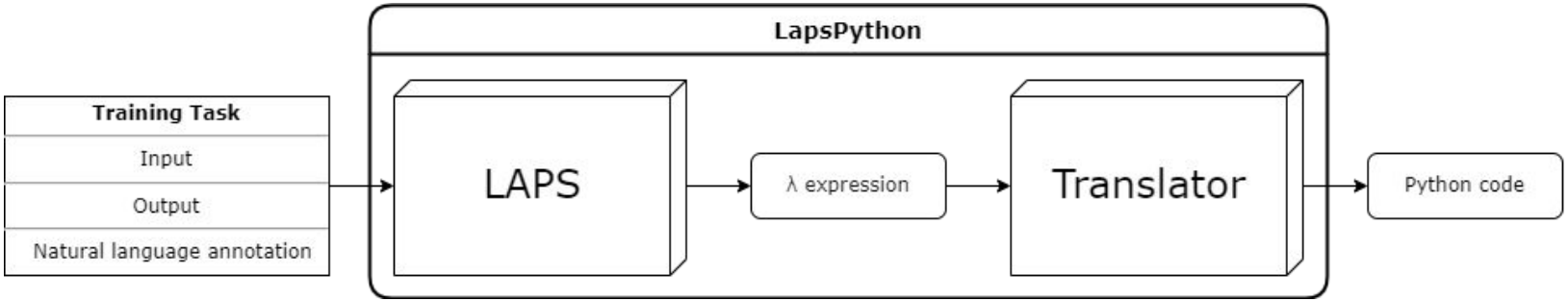
Extend LAPS to synthesize Python/R

Christopher Brückner & Enisa Sabo

20.06.2022

Objective

Extend LAPS to synthesize Python/R code from natural language



- Create rule-based translator from λ-calculus to Python code
- Define sets of primitives and tasks that target useful domains

LapsPython: Last Week

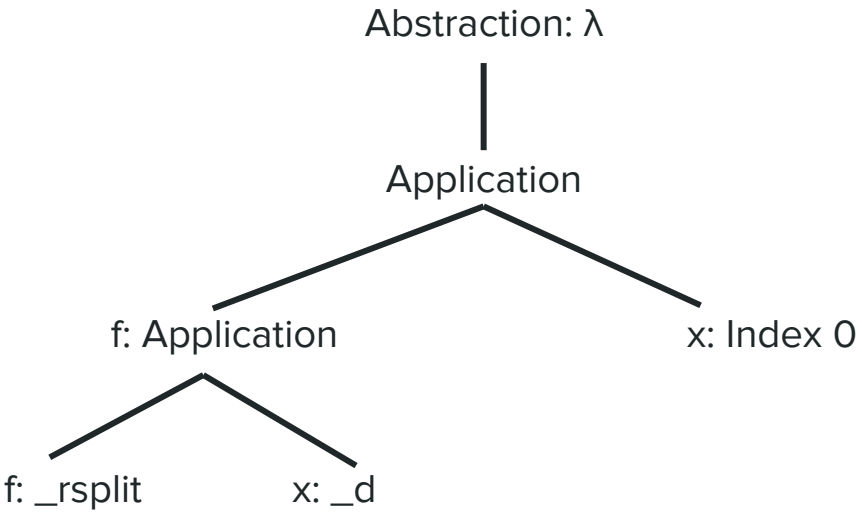
- LapsPython loads LAPS checkpoints and extracts:
  - Source codes of pre-implemented primitives in current library
  - Invented primitives in current library
  - All synthesized programs for each task
- Translation works if programs don’t contain invented primitives
- Little data to test the translation on

LapsPython: Today

- We found out: The authors provide their checkpoints
  - Lots of synthesized programs
  - Big library

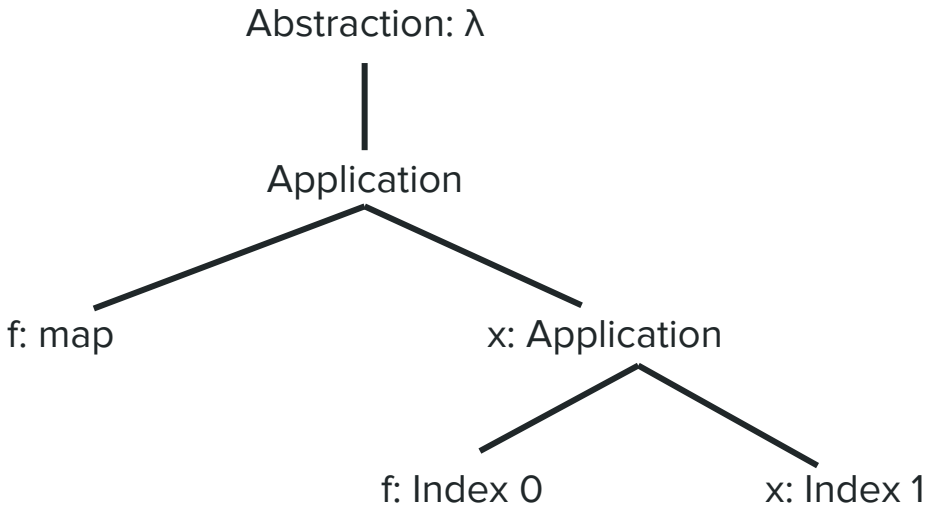
- We found out: Our translation did not work for more complex data
  - Variations in tree structure we did not encounter before
  - We need to handle 5 different classes whose meaning depends on their position
    - Abstractions (lambda function wrappers)
    - Applications f(x) with 2 child nodes f and x
    - Primitives
    - Invented Primitives
    - Indices (user-provided arguments)

Example: `(λ (_rsplit _d $0)))`



- Primitives (starting with _) can be function or constant
- Application in f node cannot be resolved: Index 0 is its 2nd argument

Example: `(λ (map $0 $1))`



- Incides (starting with $) can be lambda function or variable
- Application in x node cannot be resolved: map is the calling function

Backlog

- The translation code became quite complex and messy
  - We need to devote time to refactoring in Sprint 3

- Nested invented primitives are not handled yet
  - Example: `f(x) = 1 + g(x) + h(x)`
  - We want to translate f, but have no Python translation for g and h
  - Approach: Put translations for g and h on the stack
  - Experience: Sounds simpler than it will actually be

Sprint 3

- Original plan for Sprint 3:
  - Extension to list processing domain
  - Extension to data processing domain (pandas)

- pandas ist questionable:
  - Primitives must be implemented in Python and OCaml
  - Not sure if we can actually handle Python objects (data frames) with LAPS

- New plan for Sprint 3:
  - Finish backlog of Sprint 2
  - Modify list processing domain with annotations and appropriate tasks
  - Research LAPS + pandas

Sprint 4

- Old plan:
  - Extend data processing domain to R translation
  - Requirements:
    - Re-implementing primitives in R
    - Adapt new syntax where necessary
    - Modified code verification framework for R

- New plan:
  - Old plan, but different domain(s)
  - Python translation with pandas (if possible and we have time left)