SimpliPy: A notional machine for learning Python

International Insitute of Information Technology Hyderabad

Venkatesh Choppella, Moida Praneeth Jain, Gnaneswar Kulindala, Prabhav Shetty, Anushka Srikanth

Notional Machine for Sequential Python

Expressions

An expression is

- A **literal** like 35, True or “hello”
- An **identifier** like x, y, z
- A **compound expression** built using arithmetic and logical operators like
  - 1 + 2 + 3 * 4
  - x + 4 * y
  - x > 2

The Result Datatype

- Expressions are **Evaluated**.

- The result of evaluation is either a value or an error.

$$\mathrm{Res} = \mathrm{Val} + \mathrm{Error}$$

Expression Assignment

$a = 10$

$b =$ "Hello"

$c = x$

$d = -x$

$e = 5 + 10 * x - \frac{4}{2}$

Environments

Expressions are evaluated **in environments**

An environment is a finite function
from identifiers to values

$$\mathrm{Env} = \mathrm{Id} \rightharpoonup \mathrm{Val}$$

- $\{a \mapsto 6, b \mapsto 4, c \mapsto \mathrm{True}\}$
- ϕ denotes the empty environment

Lookup in an environment

$$\text{lookup} : \text{Id} \times \text{Env} \to \text{Res}$$

$$e = \{x \mapsto 20, y \mapsto 5\}$$

$$e : x = 20$$

$$e : y = 5$$

$$e : z = \text{error}$$

Updating an environment

$$e = \{x \mapsto 20, y \mapsto 5\}$$

$$e' = e[x := 10] = \{x \mapsto 10, y \mapsto 5\}$$

$$e'' = e'[z := 11] = \{x \mapsto 10, y \mapsto 5, z \mapsto 11\}$$

Expressions are evaluated in an Environment

$$\text{eval} : \text{Exp} \times \text{Env} \to \text{Res}$$

$$\text{eval}(x + y, \{x \mapsto 10, y \mapsto 5\}) = 15$$

$$\text{eval}(10, \{x \mapsto 100\}) = 10$$

$$\text{eval}(x + y, \{x \mapsto 10\}) = \text{error}$$

$$\text{eval}(x + y, \{x \mapsto 10, y \mapsto \text{"hello"}\}) = \text{error}$$

$$\text{eval}(x/y, \{x \mapsto 10, y \mapsto 0\}) = \text{error}$$

Instruction

An instruction is either

- An expression assignment, or
- `pass`

Program

- Location datatype

$$\mathrm{Loc} = [0..N]$$

- A program is a sequence of instruction

$$P = [0..N-1] \rightarrow \mathrm{Instr}$$

```
0  x = 5
1  pass
2  y = x + 10
3  y = 2
4
```

Static Analysis

- Certain properties of the program can be determined statically (without execution)

- Control Transfer Functions

- Control Flow Graphs

Control Transfer Functions: `next` and `err`

Partial function that maps a location to another location

- Control transfers to `next(i)` in case of **no** error.
- Control transfers to `err(i)` in case of error.

```
0
    x = 5
1
    pass
2
    y = x + 10
3
    y = 2
4
```

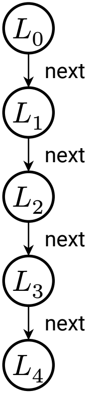
Loc	next	err
0	1	4
1	2	-
2	3	4
3	4	4
4	-	-

Control Flow Graph

- The set of control transfer functions represented as a graph.

```
x = 5
pass
y = x + 10
y = 2

```



Note that the `err` edges are omitted from the diagram for brevity.

Executing the program

- Static artefacts used in executing the program
- State
- Action (tick)
- Transitions (dynamics)

State of the Machine

- The machine is parametrized on the program P.

$$\mathrm{State} = \mathrm{Loc} \times \mathrm{Env}$$

Tranisitions of the Machine

$$(i, e) \xrightarrow{\text{tick}} (i', e')$$

Pass Transition

$$(i, e) \xrightarrow{\text{tick}} (\text{next}(i), e)$$

$$\text{if}$$

$$P_i ::= \text{pass}$$

Expression Assignment Transition

$$(i, e) \xrightarrow{\text{tick}} \begin{cases} (\text{next}(i), e') & \text{if res} \neq \text{error} \\ (\text{err}(i), e) & \text{if res} = \text{error} \end{cases}$$

$$\text{if}$$

$$P_i \Coloneqq \text{id} = \text{exp}$$

$$\text{where}$$

$$\text{res} = \text{eval}(\text{exp}, e)$$

$$e' = e[\text{id} := \text{res}]$$

Run of the Machine

```
x = 5
pass
y = x + 10
y = 2

```

Execution Diagram

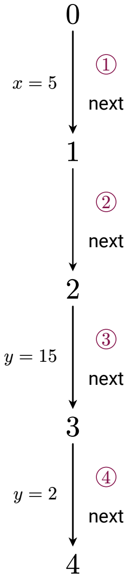
```python
x = 5
pass
y = x + 10
y = 2

```

Execution Diagram

```
0   x = 5
1   pass
2   y = x + 10
3   y = 2
4
```

$$e = \begin{cases} x \mapsto 5 & \textcircled{1} \\ \cancel{y \mapsto 15} & \textcircled{3} \\ y \mapsto 2 & \textcircled{4} \end{cases}$$



Example with error: Control Transfer Functions

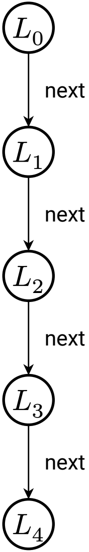
```
0  x = "Hello"
1  pass
2  y = x + 2
3  z = 5
4
```

Loc	next	err
0	1	4
1	2	-
2	3	4
3	4	4
4	-	-

Example with error: Control Flow Graph

```
x = "Hello"
pass
y = x + 2
z = 5

```



Example with error: Run of the machine

```
x = "Hello"
pass
y = x + 2
z = 5
```

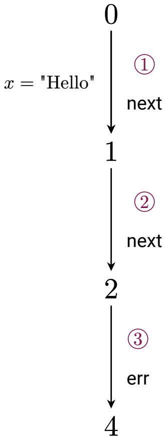
Example with error: Execution Diagram

```
x = "Hello"
pass
y = x + 2
z = 5

```

Example with error: Execution Diagram

```
0   x = "Hello"
1   pass
2   y = x + 2
3   z = 5
4
```



$$e = \{x \mapsto \text{"Hello"}$$ ①

Summary

- Expressions
- $\mathrm{Res} = \mathrm{Val} + \mathrm{Error}$
- Environments
- Evaluation
- Static Analysis
  - Control Transfer Functions (`next` and `err`)
  - Control Flow Graph
- State of the machine: (i, e)
- pass transition
- assignment expression transition
- Run of the machine
- Execution diagram