Alternative Language Support in Smalltalk Image

Adrien Hopkins & Dave Mason

2026-07-09

□□□□□□

Zag Research

- Creating a Smalltalk compiler in Zig
- Stores code as ASTs

https://github.com/Zag-Research/

Image sources:
https://github.com/Zag-Research/
https://brand.github.com/foundations/logo
https://github.com/ziglang/logo

How might we translate code between Python and Smalltalk?

About Programming Language Translation

Why translate programming languages?

- Smalltalk is not too common
- More useful libraries
- AI code generation
- Multilingual collaboration

```
f = float(input("Enter temp in °F: "))
c = (f - 32) / 1.8
print("Temp in °C is", round(c))
```

```
| f c |
f := (UIManager default request:
'Enter temp in °F') asNumber.
c := f - 32 / 1.8.
Transcript show: 'Temp in °C is ',c
rounded asString.
```

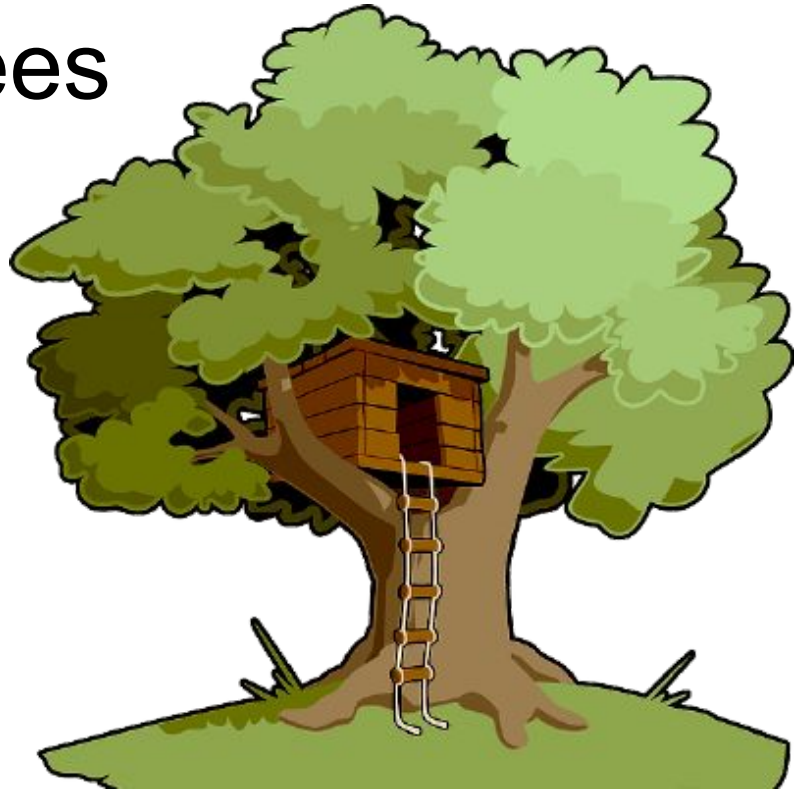
How have others done it?

- Manually create transformation rules
- Use machine learning

Transpiler Design: Python to Smalltalk

Convert by walking trees

- Tree-sitter converts Python to a Python AST
- Recursively walk AST



https://tree-sitter.github.io/

Dealing with Truthiness

```
>>> x = 1
>>> if x:
...     print('x is truthy')
... else:
...     print('x is falsy')
...
x is truthy
```

Python logo source:
https://www.python.org/community/logos/

Dealing with Array Indexing Differences

- Python indexing starts from 0, Smalltalk from 1
- Python allows negative indices for end of list
- Added & used custom extension messages for zero-indexed wrapping operations

Python	0	1	2	3	4	5
	-6	-5	-4	-3	-2	-1
Smalltalk	1	2	3	4	5	6

A Custom Switch Statement

```
{
    x > 0 -> 'x is positive'.
    x < 0 -> 'x is negative'.
    x = 0 -> 'x is zero'.
    true -> 'x is not a number?'
} switch
```

```
switch
    ^self detect: [ :each | each key value ] ifFound: [ :each | each value value ] ifNone: nil.
```

Run as Pharo

- Zag is not complete yet
- Made another translator to change code to Pharo
- Then I can run my code as Pharo

Image source: https://pharo.org/

Transpiler Design: Smalltalk to Python

Translating Basic Features

- Use a visitor on the Zag AST
- Reverse Python-to-Smalltalk rules

Printing Expressions with Precedence

- Python has complex operator precedence
- Use tables & simple rules based on precedence integers

✗ `((a + b) * (c ** d))`

✓ `(a + b) * c ** d`

✗✗ `a + b * c ** d`

Translating Blocks

- Print blocks line-by-line
- Track indentation

```
cond ifTrue: [
   doTrue value
] ifFalse: [
   doFalse value
]
```

```
if cond:
   doTrue()
else:
   doFalse()
```

Evaluation

Zag Python Converter Demo

Let’s translate Python to Zag, then run it as Pharo!

Current Situation

- 81 unit tests, 2 full programs
- Only commit code that passes tests

Future Evaluation Plans

- Make Python & translate to Smalltalk & vice versa
- Compare to original Python & generated Smalltalk

Possible Evaluation Metrics

- Proportion of code that compiles
- Proportion of unit tests that pass
- Attempt to fix broken programs

Future Work

Additional Features

- Better locals
- Functions
- Classes
- and more!

Future Challenges

- Python has multiple inheritance
- Smalltalk blocks only support returns at the end
- Smalltalk allows multiple lines in a while condition

AI-Based Library Translation

- There are too many library calls to convert manually
- Train/use an LLM to convert Python calls to Zag code & vice versa

https://github.com/Zag-Research/Zag-Python-Converter

Any questions?