

# Universal Language Transpiler

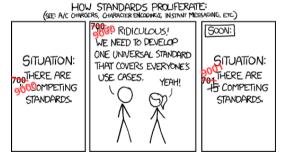
Marcel Ullrich

Saarland University

02.05.2022

#### Problem

- Migration from scripting to programming
- Rewriting of code
- Interface changes



https://xkcd.com/927/ (modified)

### Goal

- translation of code
- language agnostic
- minimal boilerplate
- readable

### Classification

- Machine languages
- Assembly languages
- High-level languages
- Scripting languages
- System languages
- Domain-specific languages
- Visual languages
- Esoteric languages
- other

## High-level languages

- imperative
- functional
- declarative
- array
- constraint
- other

## Examples

- Mandelbrot
- https://cs.lmu.edu/~ray/notes/pltypes/

#### Issues

- large difference between languages
- boilerplate around differences
- language specific features
- correct code generation
- interaction with the user
- scaling

### Ideas

- concept:
  - transpiler generator
  - transpiler that generates
- approach:
  - structure based
  - partial evaluation
  - translation
  - synthesis
  - + validation
  - common language

### Concept: Transpiler generator

- general language independent framework
- translation rules per language
- need synthesis of rules
- advantages:
  - modular
  - well suited for very similar languages
- disadvantages:
  - complicated rules
  - difficult for different ASTs
  - syntactical

### Concept: Generating Transpiler

- takes a fragment of code
- produces a fragment of code
- advantages:
  - general
  - allows structure changes
- disadvantages:
  - not modular
  - not necessarily structure preserving

### Idea: Structure based

- synthesize translation rules
- compare examples in the two languages
- same context ⇒ same semantics
- advantages:
  - modular
  - well suited for very similar languages
- disadvantages:
  - complicated rules
  - difficult for different ASTs
  - syntactical

### Idea: Partial evaluation

- interpreter of A in B
- partial evaluation of B
- symbolic evaluation ⇒ partial eval, interpreter
- advantages:
  - reuses written code
  - correctness transfer
  - one manual transpilation
- disadvantages:
  - needs complex components
  - work for each new language

#### Idea: Translation

- text to text translation
- advantages:
  - works for natural languages
  - classical machine learning problem
- disadvantages:
  - no correctness guarantees
  - complicated structural constraints
  - needs suitable representation
  - no insight

Idea: Synthesis

TODO

Idea: Synthesis + Validation

TODO

## Idea: Common language

TODO

## **Applications**

- Similar to Pandoc, Rosetta, LLVM
- API changes
- virtual API
- Migration to new code (maintainance)
  - COBOL → C(++)
  - PHP → Javascript/Python
  - $\bullet \ \, \mathsf{X} \to \mathsf{Typescript}$

- Language features
  - verification
  - libraries
  - speed
  - knowledge
  - safety guarantees
  - optimizations
  - tooling environment
- competing frameworks