

The Typer Piper: Automating Data Structure Transformations Through Type Chaining

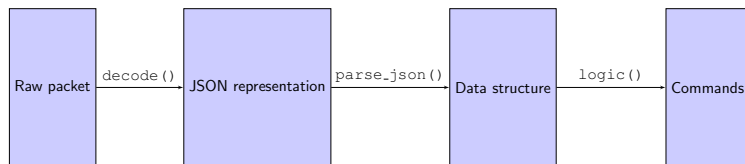
Josh Gruenstein Martin Schneider

May 7, 2019

Contents

- 1 Introduction
- 2 Type System
- 3 Search Engine
- 4 Demos!
- 5 Discussion

Motivation

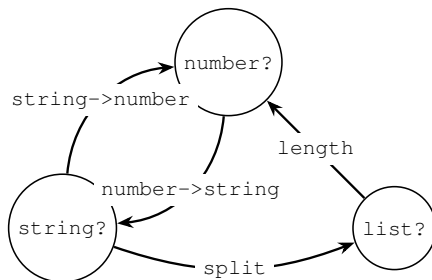


A type conversion diagram for a webserver.

Concept

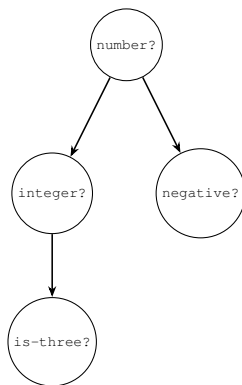
- Inference of type conversion flow
- Program superstructure writes itself
- Less code, less thinking, more good.

Predicate Transformation Graph



An example predicate conversion graph.

Subtyping and Supertyping



An example supertype tree.

Backtracking Search

- Start at input predicate, take predicate transforms on edges to explore.
- Backtrack if dead end, and don't revisit predicates.
- $(\text{string?} \quad \text{number?}) \xrightarrow{?} (\text{string?} \quad \text{string?})$

Compound Transformations

$$\begin{array}{lcl}
 & & \begin{array}{l} \text{(string? number?)} \\ \text{(string? string?)} \\ \text{(number? number?)} \\ \dots \\ \text{(list? string?)} \end{array} \\
 \text{(string? number?)} \rightarrow \begin{bmatrix} \text{string?} \\ \text{number?} \\ \text{list?} \end{bmatrix} \times \begin{bmatrix} \text{number?} \\ \text{string?} \end{bmatrix} \rightarrow
 \end{array}$$

Transformations of (string? number?) using the example graph.

Joiner Transformations

`person? $\xrightarrow{?}$ (person:first? person:last?)`

- 1 Find all possible compound targets using target and graph
- 2 Filter those that can reach target
- 3 Find path from input to each predicate in each compound

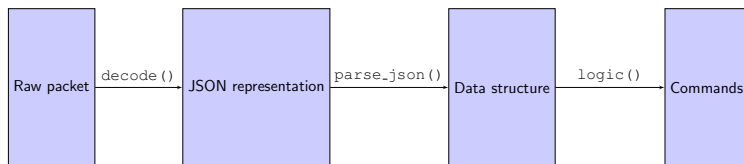
Simple example

```
(register-predicate! list?)  
(register-predicate! number?)  
(register-predicate! string?)  
  
(define (is-three? num) (eq? num 3))  
  
(register-predicate! is-three?)  
(register-super! is-three? number?)  
  
(register-type-transform! list? number? length)  
(register-type-transform! number? string? number->string)  
(register-type-transform! string? number? string->number)  
  
; (debug-get-transformations-values number? string? 1)  
; (debug-get-transformations-values list? string? '(1 2 3))  
; (debug-get-transformations-values is-three? string? 3)  
  
(debug-transform-to 3 string?)
```

Pause for live demos!

DEMOS.

Programming as type conversion



A type conversion diagram for a webserver.