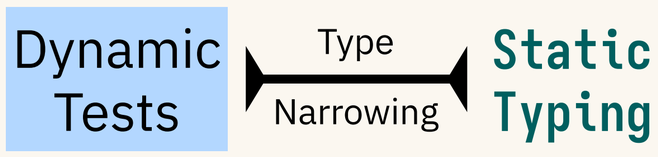
If-T: A Benchmark for Type Narrowing

Hanwen Guo (hanwen.guo@utah.edu)
Ben Greenman (benjamin.l.greenman@gmail.com)

Background: Type Narrowing

```
x :: T
if x is Number:
   x :: T ∩ Number
else:
   x :: T \ Number
```



Dynamic Languages

- Embed type tests into ordinary control flow

Gradual Typecheckers

- Attach static type system to dynamic languages

Type Narrowing

- Refine types along control flow paths

How to Compare Them?

Problem: Conflicting Implementations

Too Many Implementations

One predicate, Many syntaxes

```
function p(x): x is T
def p(x) -> TypsIs[T]
(: p (-> Any Boolean : T))
```

Expressive Power Differs

Not Expressible!

Core Benchmark

Basic Narrowing					
positive	•	•	•	•	•
…	…				
Compound Structures					
struct fields	•	•	•	•	•
tuple elements	•	•	•	•	•
tuple length	×	•	•	•	•
Advanced Control Flow					
alias	•	•	×	×	•
nesting condition	•	×	×	×	×
merge with union	•	•	•	×	•
Custom Predicates					
predicate 2-way	•	•	•	•	•
predicate 1-way	•	×	•	•	•
predicate checked	•	×	•	×	×

4 groups
13 narrowing features
(each with 2 example programs)
5 typecheckers

An open source benchmark, check it out!