1 The Pyret Language

Pyret is a programming language under development at Brown intended for educational use. It encourages programming in a functional style and has both typed and untyped versions. The type system enables a greater number of static guarantees on program execution but requires a greater number of *type annotations* in order to type-check a Pyret program.

2 Check-based Inference

One of Pyret’s education-oriented features is a purpose-built construct for test cases: specific *check* blocks contain specifications for program behavior. We exploit the fact that both types and test cases provide specifications for a program in order to *infer function types from test cases*.

```pyret
fun string-times(s, n):
    if n <= 0:
        ""
    else:
        s + string-times(s, n - 1)
    end
end

check:
    string-times("hi", 0) is ""   end
    string-times("hi", 3) is "hihihi" end
end

fun foobar(b):
    cases (Foo) b:
        bar(x :: Number) => bar(4)
        foo => bar(4)
    end
end

check:
    foobar(foo) is bar(4)
end
```

Figure 1: These programs type-check despite the lack of explicit annotations because the test cases cover the possible input types.

3 Further Applications

- Inference for polymorphic type variables
- Improving Type error messages