dots
A graph programming language
The Team

Hosanna Fuller: Manager
Rachel Gordon: Language Guru
Adam Incera: C Guru
Yumeng Liao: Tester
The Problem

Masters student at Columbia interested in Networks

Noticed that implementing graphs or graph operations are a high cost data structure to build

Wanted to try to implement Graph language to help with my research
Our Goal

Create a versatile language to allow simple manipulation of graphs and their data.
/*Types*/
num
string
bool
node
graph
list<type>
dict<key,val>

/*Declare Function*/
def num foo(num bar){
    return bar;
}

/*Operators*/
+
-
*
/
>
<

>=
<=
==
!=
||
&&
-->

/*Some Builtin Fns*/
enqueue(x)
dequeue()
node_var.ine()
node_var.oute()
min(x)
max(x)

/*Control Flow*/
for (x in y) {
    something;
}
if (x == y) {
    something;
} else {
    other;
}
while (x == y) {
}
/* This is a comment! Let’s declare and assign some variables. */
num w; string v = "hello world";
node y; node z("hello");
list<num> l = [1,2,3];
dict<num,string> d; d[3.3] = "foo"; d[10] = "bar";

/* Declare graphs with weighted edges easily */
graph g1 = {
    y -->[10] z
};

/* Alternatively, add nodes together to make a graph! */
graph g2 = y + z;
Demo
Test Suite

- Unit/Regression testing
- Vast majority of testing on generated code
- Python script that runs all tests in a folder by opening subprocesses
Workflow

1. Weekly Meetings
2. Biweekly Standups
3. Slack Channel
4. Github
5. Pair Programming
Lessons Learned

-Ambition is good, but don’t try to do everything ever in the universe

-Pair programming is awesome

-Get the minimum viable product working

-Ocaml has its magical moments
Excluding merges, 4 authors have pushed 169 commits to master and 171 commits to all branches. On master, 287 files have changed and there have been 4,710 additions and 1,119 deletions.