Extend

Language Reference Manual

Ishaan Kolluri, Kevin Ye, Jared Samet, Nigel Schuster

October 18, 2016

Contents

1. Introduction to Extend

Extend is a domain-specific programming language used to designate ranges of cells as reusable functions. It abstracts dependencies between cells and builds a dependency graph during compilation. Extend compiles to LLVM in order to take advantage of its extensive garbage collection features.

Extend's syntax is meant to provide clear punctuation and specificity for cell ranges, while maintaining the look of modern functional programming languages. Given Extend's functionality resonates well with spreadsheets, it borrows syntactical elements from programs such as Microsoft Excel.

2. Types and Literals

2.1. Primitive Data Types

Extend's basic primitives are *integers*, *floats*, *char* literals, and *string* literals. They are all internally represented as numbers or a range of numbers. They are as follows:

A **char** literal is essentially a size 1 numerical range. At evaluation, the number in the range will be compared with its ASCII equivalent.

A **string** literal is a range of numbers of size n, where n is the length of the string. The string 'hello' can be represented internally as {42,23,18,27,-3}. A **integer** can be represented as a size 1 numerical range as well. However, it retains its numerical value upon evaluation.

TODO: Get a definition for float

Below is a snippet illustrating declarations for each of the above types.

```
/* Integer */

  num = 5;

/* Char */

chr = 'A'

/* String */

str = 'Hello'
```