Kraken Programming Guide

Jack SPARROW

October 18, 2015

1 Compiling

Kraken compilation currently only supports building the compiler from source. You can clone the repository from a terminal using:

```
git clone https://github.com/Limvot/kraken.git
```

Once you have the repository, run the following commands:

```
mkdir build %Create a build directory
cd build
cmake ..    %Requires cmake to build the compiler
make        %Create the compiler executable
```

This will create a kraken executable, which is how we will call the compiler. Kraken supports several ways of invoking the compiler. These include:

```
kraken source.krak
kraken source.krak outputExe
kraken grammarFile.kgm source.krak outputExe
```

The grammar file is a file specific to the compiler, and should be included in the github repository. When you run the compile command, a new directory with the name of the outputExe you specified will be created. In this directory is a shell script, which will compile the created C file into a binary executable. This binary exectuable can then be run as a normal C executable.

2 Variables

2.1 Variable Declaration

```
int main(){
  std::cout << ”Hello␣World” << std::endl;
  return 0;
}
```

2.2 Primitive Types

primitive types

3 Functions

Section func

4 Classes

Section class

5 Templates

Section T

6 Standard Library

Section STL

7 Understanding Kraken Errors

Section error

8 Answers to Definitions

a. Item 1

b. The *units of atomic weight* are two-fold, with an identical numerical value. They are g/mole of atoms (or just g/mol) or amu/atom.

c. *Percentage discrepancy* between an accepted (literature) value and an experimental value is

$$a = 4$$