Modeling

How to complete an exercise successfully?

Follow the rules as described in the Lecture!

How to get additional information?

You are encouraged to discuss past and present exercise sheets with the teaching assistants. Either approach the teaching assistant during the exercise session, or visit us during the weekly office hours. We are also available through e-mail or on the StudIP forum. We try to reply as quickly as possible and in general, you should get a reply the next weekday, but we cannot guarantee this.

Installation

In this course, you will develop a compiler for a subset of C++. You will implement your compiler either in Haskell[1] or in C++[2].

We will use the following software:

- GHC (included in Haskell-Platform)[3]
- cabal (included in Haskell-Platform)[4]
- BNFC[5]
- llvm[6]
- make[7]
- alex (optional, for Haskell; included in Haskell-Platform)[8]
- happy (optional, for Haskell; included in Haskell-Platform)[9]
- llvm-general (optional, for Haskell)[10]
- clang (optional, for C++)[11]
- bison (optional, for C++)[12]
- flex (optional, for C++)[13]

[1] http://www.haskell.org
[2] http://isocpp.org/
[3] http://www.haskell.org/ghc
[4] http://www.haskell.org/cabal
[5] http://bnfc.digitalgrammars.com/
[6] http://www.llvm.org
[7] https://www.gnu.org/software/make/
[8] http://www.haskell.org/alex/
[9] http://www.haskell.org/happy/
[10] http://hackage.haskell.org/package/llvm-general, http://hackage.haskell.org/package/llvm-general-pure
[11] http://clang.llvm.org/
[12] https://www.gnu.org/software/bison/
[13] http://flex.sourceforge.net/

General Preparations

We will use the BNFC tool to generate a parser and code snippets for the language of your choice, and the make tool to control the compilation process. Please make sure you have the make tool installed before proceeding. The BNFC tool is written in the Haskell programming language and, therefore, you will need a working Haskell installation to compile BNFC. We recommend to install the Haskell Platform[14], preferably via your OS's package management system. The Haskell Platform contains GHC, cabal, as well as alex and happy. The missing BNFC can be installed using the command `cabal update` followed by `cabal install BNFC`. Please make sure that the `bnfc` executable, installed by cabal, is in your search path.

Just as the installation of the make tool, the installation of the remaining software is OS dependent. Please make sure you have it installed.

On Windows systems, we recommend installing a virtual machine, e.g. using Virtualbox[15], with a debian-based Linux distribution, e.g. Debian or Ubuntu[16]. On such Linux distributions, the above-mentioned software can be installed using the following sequence of commands:

As root:

```
apt-get update
apt-get install build-essential haskell-platform llvm clang llvm-3.5 libedit-dev
```

As user:

```
cabal update
cabal install BNFC
cabal install llvm-general --force-reinstalls
echo "export PATH=${HOME}/.cabal/bin:${PATH}" >> ${HOME}/.bashrc
source ${HOME}/.bashrc
```

Tasks

Exercise 0.1 Download the sample grammar "CPP.cf" from StudIP, and test your BNFC installation using `bnfc -m CPP.cf`. This command will generate Haskell code as well as a makefile that you can use to compile with the command `make`. Alternatively, you may also generate code for the language of your choice. Download the sample "foo.cc" file from StudIP. Demonstrate the application of the `TestCPP` executable to the sample file. **Note** that to pass, this time you have to demonstrate your working toolchain to the teaching assistant in class.

Exercise 0.2 Depending on your language of choice, download the corresponding llvm demo from StudIP. Demonstrate that you can compile and run the demo using `make test`.

[14] http://www.haskell.org/platform
[15] https://www.virtualbox.org
[16] Tested on Ubuntu 15.10 Desktop