

# Project 6

CMSC 22620/32620, Spring 2007

*Assigned: May 21, 2005*

*Due: May 29, 2005*

## 1 Introduction

You are to write another simple optimizer for *closed* ANF to perform a simple form of *common subexpression elimination* (CSE) called *value numbering*.

Value numbering was discussed in class and is also explained in Appel's textbook in Chapter 17, Algorithm 17.7.

The tarball for this project contains a template `value-numbering.sml` with numerous comments that should help you in designing your solution. The basic logic is very similar to that of Project 5, and you should not have to use more than an average of 3-4 lines per case of the `Closed.exp` datatype, plus some helper routines.

## 2 Instructions

### 2.1 Files

Download the file `project6.tgz` from the course web page. This is a compressed tarball with the (almost) complete sources of the MiniML compiler. File `value-numbering.sml` contains a skeleton of the code you are supposed to write. File `anf-opt.sml` is still a skeleton. Substitute your solution to Project 5 here.

### 2.2 Building the compiler

To compile the sources, run `CM.make "miniml.cm";` from the source directory. Once this goes through without error, you can build a stand-alone heap image for the compiler using the shell command

```
ml-build miniml.cm Main.main miniml
```

If this completes without error, you can invoke the MiniML compiler from the shell command line using, e.g.,

```
./minimlc -S Tests/mergesort3.sml
```

The resulting assembly code file will be `Tests/mergesort3.s`.

## 2.3 Compiling to executable code

To compile to an executable, you first have to build the MiniML runtime system:

```
cd rt
make
cd ..
```

You can now omit the `-S` flag when you invoke `./minimlc`. The result of compiling, e.g., `Tests/mergesort3.sml` will then be an executable named `Tests/mergesort3`.

## 2.4 Testing

To see the closed ANF cluster code before value numbering, you can provide the command-line option `-PC` to the `minimlc` command. To see the cluster code after value numbering, you need to edit `compile.sml` and add appropriate pretty-printing code. (Using the `-PT` command line option you can also see the `BBTree` code that is generated right after value numbering.)

## 3 Handing it in

You should only have to make changes to file `value-numbering.sml`. To hand in your solution, send this file as an e-mail attachment to the instructor using the following e-mail address:

```
instructor | blume (at) tti (hyphen) c (dot) org
```

If you make other changes (which should really not be necessary!), then bundle all your files as a tarball and attach that to your e-mail.