KØBENHAVNS UNIVERSITET, DATALOGISK INSTITUT

A Compiler for the FASTO Language

Allan Nielsen, Christian Nielsen, Troels Kamp Leskes

December 21, 2014

CONTENTS

1 MULTIPLICATION, DIVISION, BOOLEAN OPERATORS AND LITERALS

Implementing multiplication and division was a simple matter, when having the already implemented code for addition and subtraction to look at. They served as a great way of getting to know the fasto compiler, and how things operate.
Negation is implemeted using the Mips.SUB instruction, where we pass the original argument to the operator, and subtracts this from zero.
So the instruction looks like:

```
Mips.SUB(place, "0", t1),
```

where t1 register that holds the argument x for $\sim x$, and place is the register in which we place the result.
Not was more complicated than the previous ones, given that this requires more than one instruction to execute. However, the pattern learned here, proved to be useful for implementing and and or as well.

```
[ Mips.LI (place,"0")
, Mips.BNE (b,"0",falseLabel)
, Mips.LI (place,"1")
, Mips.LABEL falseLabel ]
```

Place is the register in which we want to store our result. We start by putting 0 into place, we then check if our argument b, actually is 0. Since Mips.BNE branches if its arguments are not equal, we will jump to falseLabel if and only if our argument b is 1, thus ending with a 0 in the place register, given we never execute Mips.LI(place, "1").

HAR INGEN IDE OM BOOL LITS

2 FILTER AND SCAN

3 λ-EXPRESSIONS IN SOACS

4 COPY PROPAGATION AND CONSTANT FOLDING