The SnuPL/1 Intermediate Representation

The SnuPL/1 IR is implemented in `ir.cpp/h` and largely follows the textbook. The class hierarchy is illustrated below:

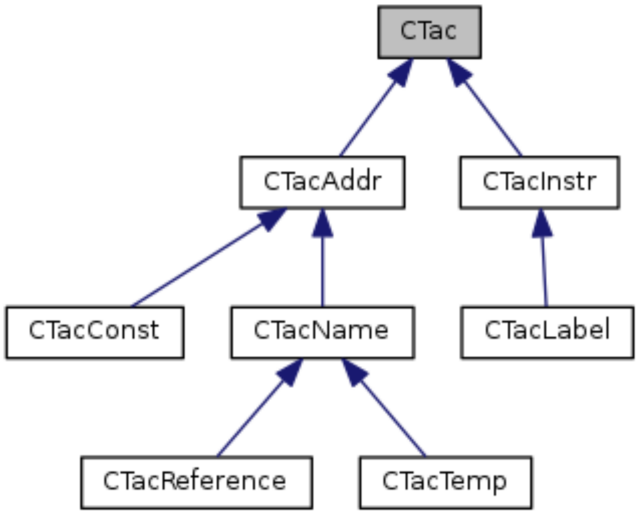

Illustration 1: Three-address code class hierarchy

`CTacAddr` and subclasses represent symbols, temporaries, and constant values. `CTacAddr` and its subclasses form the operands of `CTacInstr` instructions.

Operations are implemented using `CTacInstr`. `CTacLabel` is a special instruction that simply serves as a label and does not actually execute any code. `CTacLabel` can be used as an operand for branching operations (goto, if relop goto..., see below). Different operations require different operands, both in type and number; refer to Table 1 below.

The `CCodeBlock` class manages the list of instructions, and is also responsible to generate (unique) temporary values and labels. The relevant methods are:

```
CTacTemp* CCodeBlock::CreateTemp(const CType *type);
CTacLabel* CCodeBlock::CreateLabel(const char *hint=NULL);
CTacInstr* CCodeBlock::AddInstr(CTacInstr *instr);
```

`CScope` and its subclasses, finally, represent the module and procedures/functions of the program.

SnuPL/1 IR

Opcode	Dst	Src1	Src2	Description
opAdd	result	$operand_1$	$operand_2$	result := $operand_1$ + $operand_2$
opSub	result	$operand_1$	$operand_2$	result := $operand_1$ - $operand_2$
opMul	result	$operand_1$	$operand_2$	result := $operand_1$ * $operand_2$
opDiv	result	$operand_1$	$operand_2$	result := $operand_1$ / $operand_2$
opAnd	result	$operand_1$	$operand_2$	result := $operand_1$ && $operand_2$
opOr	result	$operand_1$	$operand_2$	result := $operand_1$ || $operand_2$
opNeg	result	operand		result := -operand
opPos	result	operand		result := +operand
opNot	result	operand		result := ~operand
opEqual	target	$operand_1$	$operand_2$	if $operand_1$ = $operand_2$ goto target
opNotEqual	target	$operand_1$	$operand_2$	if $operand_1$ # $operand_2$ goto target
opLessThan	target	$operand_1$	$operand_2$	if $operand_1$ < $operand_2$ goto target
opLessEqual	target	$operand_1$	$operand_2$	if $operand_1$ <= $operand_2$ goto target
opBiggerThan	target	$operand_1$	$operand_2$	if $operand_1$ > $operand_2$ goto target
opBiggerEqual	target	$operand_1$	$operand_2$	if $operand_1$ >= $operand_2$ goto target
opAssign	LHS	RHS		LHS := RHS
opAddress	result	operand		result := &operand
opDeref	result	operand		result := *operand
opCast	result	operand		result := (type)operand
opGoto	target			goto target
opCall	result	target		result := call target
opReturn		operand		return operand
opParam	index	operand		index-th parameter := operand
opLabel				jump target
opNop				no operation

Table 1: SnuPL/1 intermediate representation