TheDan64 / inkwell Public

Code Issues 73 Pull requests 2 Actions Projects Security Ins

Edit New issue Jump to bottom

Undocumented Unconventional Order of BR Instruction Label Operands #399

Open **wu-benjamin** opened this issue last week · 0 comments

wu-benjamin commented last week

Describe the Bug
The convention adopted by the syntax for the conditional 'br' instruction in the LLVM language reference manual (https://llvm.org/docs/LangRef.html#i-br) is:
`br i1 <cond>, label <iftrue>, label <iffalse>`

In Inkwell, an `InstructionValue` `instr` with opcode `instr.get_opcode() == InstructionOpcode::Br` and `instr.get_num_operands() == 3` has the indices of the `<iftrue>` and `<iffalse>` operands swapped.

That is, `instr.get_operand(1)` represents the `<iffalse>` basic block and `instr.get_operand(2)` represents the `<iftrue>` basic block.

Expected Behavior
I expected the behavior to match the syntax for the conditional 'br' instruction in the LLVM language reference manual: `instr.get_operand(1)` represents the `<iftrue>` basic block and `instr.get_operand(2)` represents the `<iffalse>` basic block.

I don't think changing this to match the syntax for the conditional 'br' instruction in the LLVM language reference manual is feasible due to such a change being breaking. Instead, I propose documentation for Inkwell explicitly declares the semantics of each operand.

LLVM Version (please complete the following information):

- LLVM Version: 13.0.1
- Inkwell Branch Used: master

Desktop (please complete the following information):

- OS: MacOS (Ventura) 13.2.1 (22D68)

wu-benjamin mentioned this issue last week

Notify inkwell of order of conditional branch operands JustinReiter/wombat-symx#10

Closed

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant