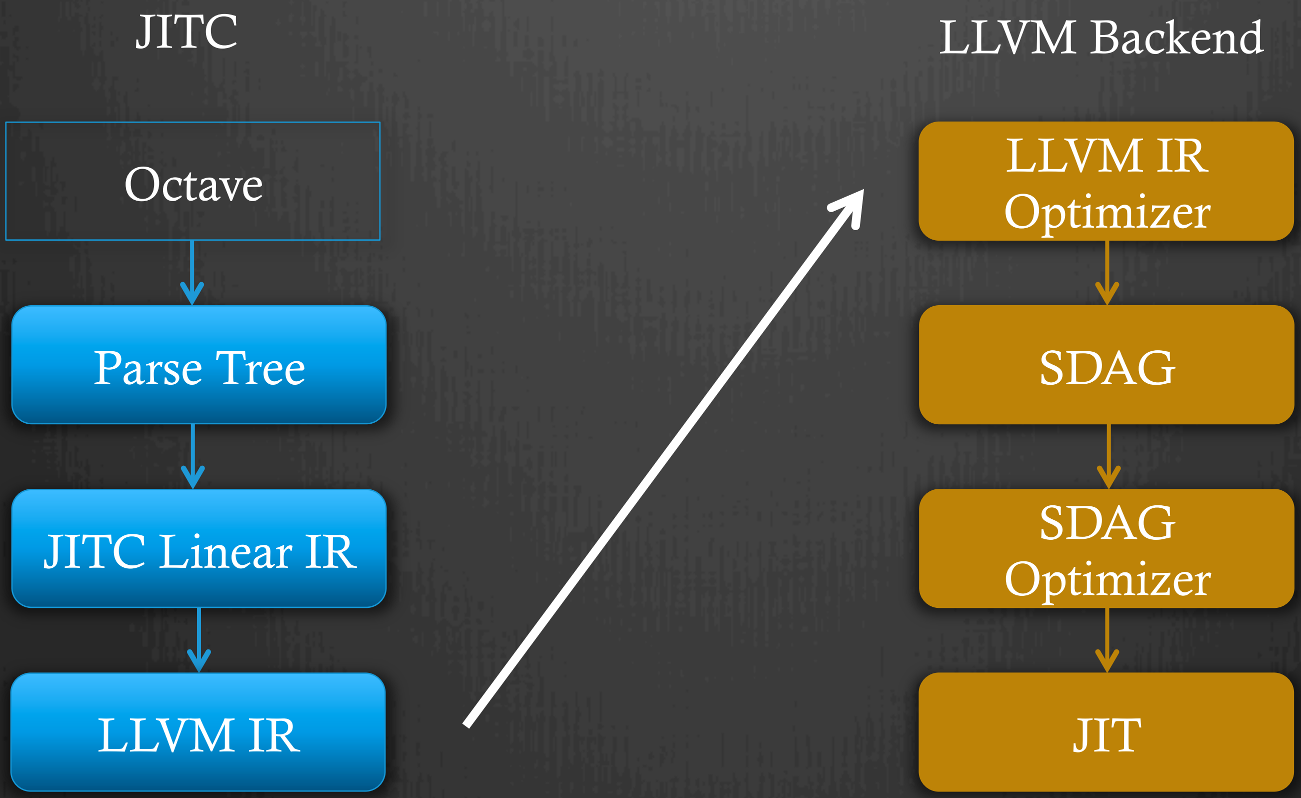
Improve JIT Compiling

L Y.H.

JITC

- Just In Time Compiler for Octave
- Google Summer of Code 2012
  - Max Brister
- Use LLVM as JIT engine
- ~9000 SLOC

How JITC Works



Midterm Goal

- Built-in functions support

```
function test1 ()
  x = 1.0;
  disp (x);
endfunction
```

Midterm Goal

- Built-in functions support
- Statements support
  - do - until
  - switch
  - …

```
i = 1;
x = 2;
do
  i++;
  x = x * 2;
until (i == 10)
```

Final Goal

- Statements support
- Functions support

```
function retval1 = fun1 ()
  retval1 = 2;
endfunction
```

JITted

```
function retval2 = fun2 ()
  retval2 = 2 * fun1 ();
endfunction
```

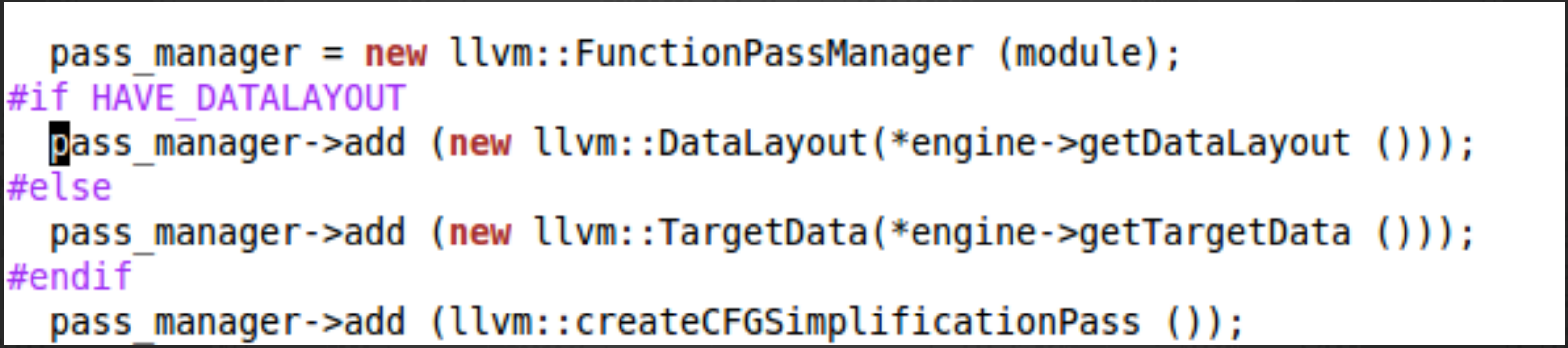
Not JITted

Some Issues

- LLVM now focus on MCJIT

Some Issues

- LLVM now focus on MCJIT
- LLVM API varies between different versions
  - LLVM IR has no change

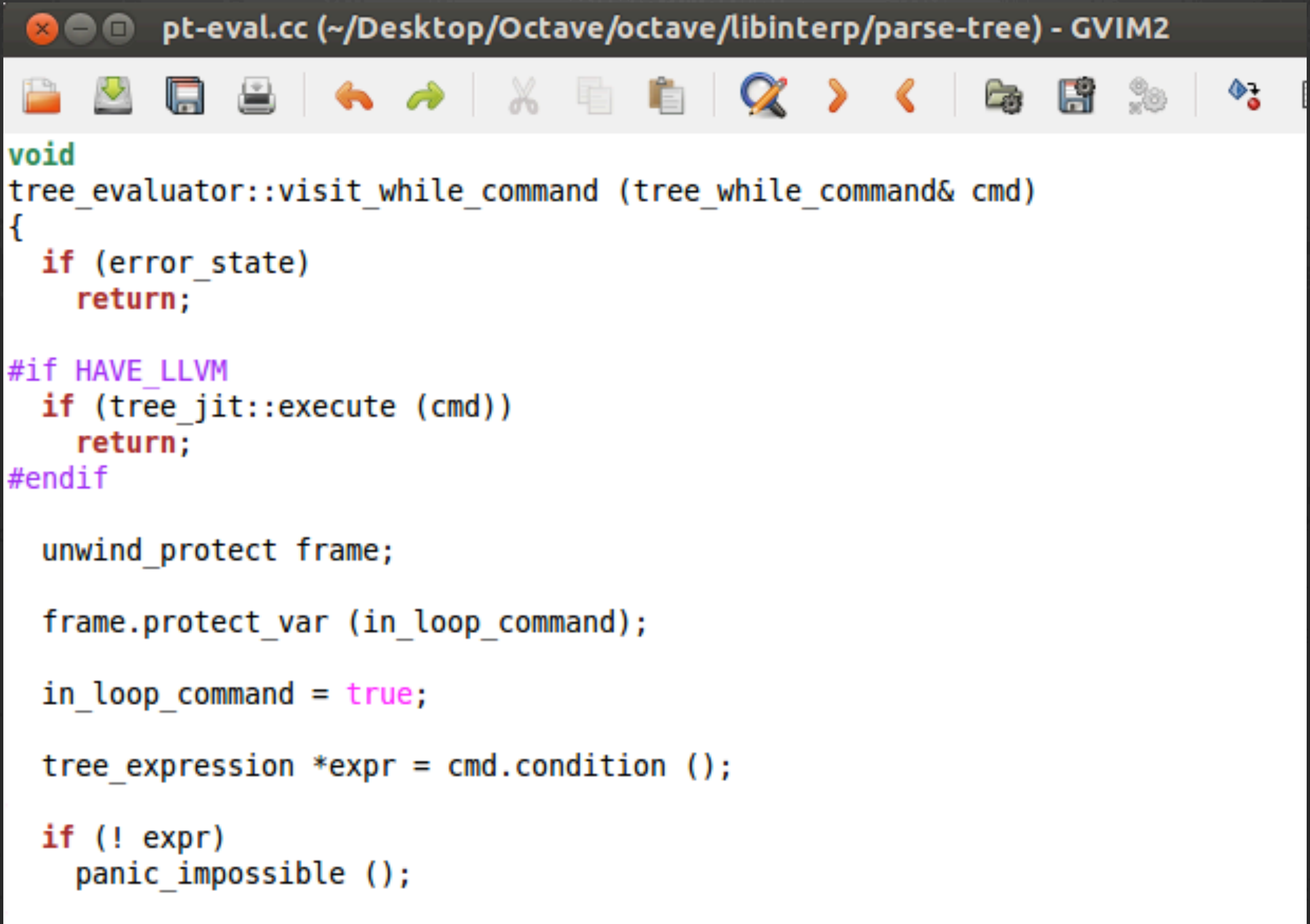


```
  pass_manager = new llvm::FunctionPassManager (module);
#if HAVE_DATALAYOUT
  pass_manager->add (new llvm::DataLayout(*engine->getDataLayout ()));
#else
  pass_manager->add (new llvm::TargetData(*engine->getTargetData ()));
#endif
  pass_manager->add (llvm::createCFGSimplificationPass ());
```

Some Issues

- LLVM now focus on MCJIT
- LLVM API varies between different versions
  - LLVM IR has no change
- JITC unaware code hotness

JITC unaware code hotness



pt-eval.cc (~/Desktop/Octave/octave/libinterp/parse-tree) - GVIM2

```
void
tree_evaluator::visit_while_command (tree_while_command& cmd)
{
  if (error_state)
    return;

#if HAVE_LLVM
  if (tree_jit::execute (cmd))
    return;
#endif

  unwind_protect frame;

  frame.protect_var (in_loop_command);

  in_loop_command = true;

  tree_expression *expr = cmd.condition ();

  if (! expr)
    panic_impossible ();
```

Thank you for your attention