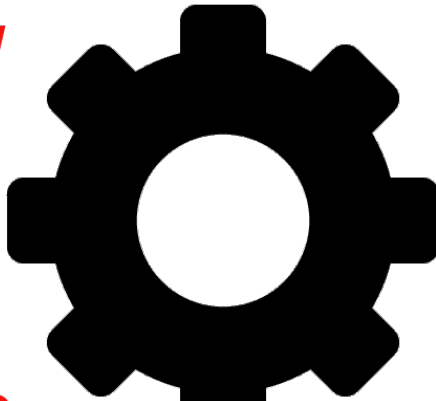


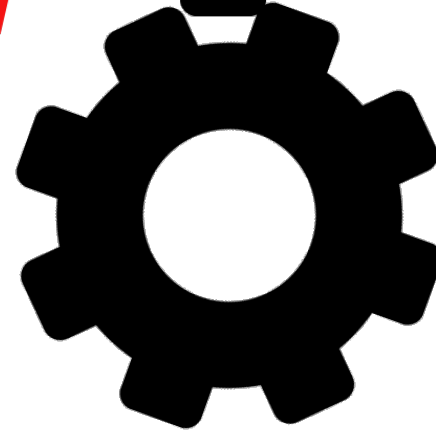
*Advanced*

T



pics

*in*



C

mpilers



# InvariantsManager

Simone Campanoni  
simone.campanoni@northwestern.edu

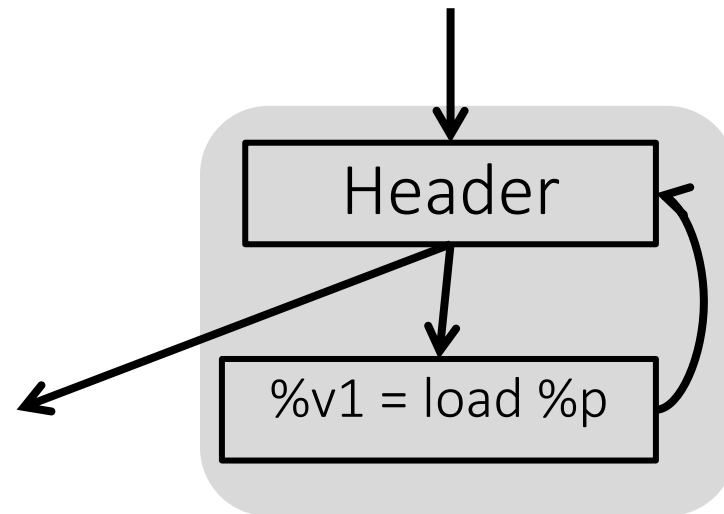


# Outline

- What is it and why NOELLE provides it
- Checking invariants

# InvariantsManager

- It captures the computation within a loop that are loop invariants for that loop



# Outline

- What is it and why NOELLE provides it
- Checking invariants

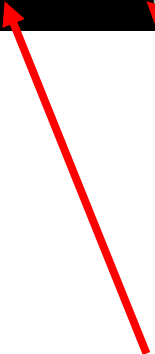
# Checking invariants

```
/*  
 * Invariants.  
 */  
errs() << " Invariants\n";  
auto IM = loop->getInvariantManager();
```

Instance of the class `arcana::noelle::LoopContent`



Instance of the class `arcana::noelle::InvariantManager`



# Checking invariants

```
/*  
 * Invariants.  
 */  
errs() << " Invariants\n";  
auto IM = loop->getInvariantManager();
```

```
/*  
 * Invariants.  
 */  
errs() << " Invariants (instruction granularity)\n";  
for (auto inv : IM->getLoopInstructionsThatAreLoopInvariants()) {  
    errs() << "    " << *inv << "\n";  
}
```

Instance of the class `llvm::Instruction` \*

# Checking invariants

```
/*  
 * Invariants.  
 */  
errs() << " Invariants\n";  
auto IM = loop->getInvariantManager();
```

```
/*  
 * Invariants.  
 */  
errs() << " Invariants (instruction granularity)\n";  
for (auto inv : IM->getLoopInstructionsThatAreLoopInvariants()) {  
    errs() << "    " << *inv << "\n";  
}
```

```
auto isInstructionInvariant = IM->isLoopInvariant(inst);
```

 Instance of the class `llvm::Instruction`

# Checking invariants

```
/*  
 * Invariants.  
 */  
errs() << " Invariants\n";  
auto IM = loop->getInvariantManager();
```

```
auto isInstructionInvariant = IM->isLoopInvariant(inst);
```

```
/*  
 * Check if the SCC is invariant  
 */  
auto isSCCInvariant = IM->isLoopInvariant(scc);
```

 Instance of the class `arcana::noelle::SCC`



Always have faith in your ability

Success will come your way eventually

**Best of luck!**