Advanced Topics in Compilers

Forest of loops

Simone Campanoni
simone.campanoni@northwestern.edu

Outline

- Forest of loops with NOELLE
- A tree of loops with NOELLE
- Modifying the forest
- Forest between functions

Get loops of a function with NOELLE

```
/*
 * Fetch the entry point.
 */
auto fm = noelle.getFunctionsManager();
auto mainF = fm->getEntryFunction();
```

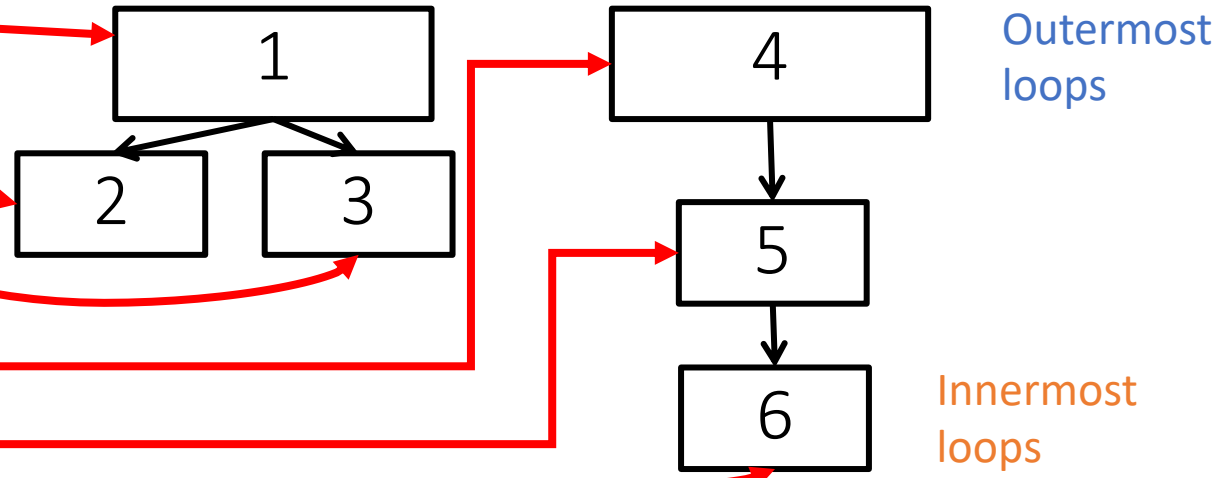
```
/*
 * Fetch the loops with only the loop structure abstraction.
 */
auto loopStructures = noelle.getLoopStructures(mainF);
```

Each loop is an instance of llvm::noelle::LoopStructure
Flat representation of the loops

But we know there is a nesting relation between some loops

Loop nesting forest

```
void myFunction (){
1: while (…){
2:    while (…){ … }
3:    while (…) {… }
   }
   …
4: for (…){
5:    do {
6:       while(…) {…}
      } while (…)
   }
}
```



Loop forest with NOELLE

```
/*
 * Fetch the entry point.
 */
auto fm = noelle.getFunctionsManager();
auto mainF = fm->getEntryFunction();
```

```
/*
 * Fetch the loops with only the loop structure abstraction.
 */
auto loopStructures = noelle.getLoopStructures(mainF);
```

```
/*
 * Fetch the loop forest.
 */
auto loopForest = noelle.organizeLoopsInTheirNestingForest(*loopStructures);
```

llvm::noelle::LoopForest *

noelle/core/LoopForest.hpp

Using LoopForest

```
/*
 * Check the loop that contains each instruction of main.
 */
errs() << "Check loops that contain instructions in main\n";
for (auto &inst : instructions(mainF)){
  errs() << "  Instruction: " << inst << "\n";

  /*
   * Fetch the loop.
   */
  auto loop = loopForest->getInnermostLoopThatContains(&inst);
  if (loop == nullptr){
    errs() <<     The instruction does not belong in any loop\n";
    continue ;
  }

  errs() << "    The instruction belongs to a loop\n";
}
```

llvm::noelle::LoopTree *

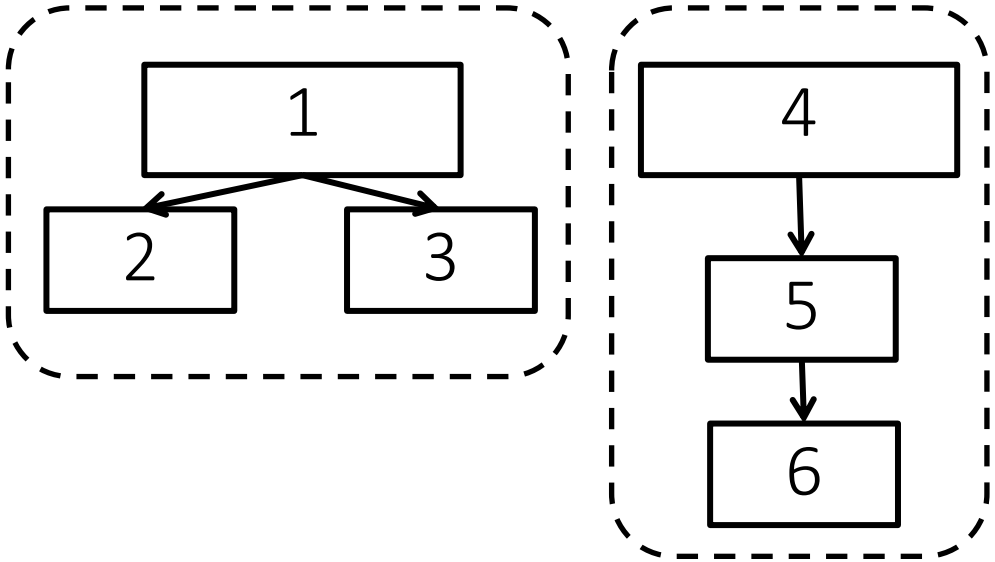
Traversing loop forest with NOELLE

```
/*
 * Iterate over the trees that compose the forest.
 */
errs() << "Printing the loop forest\n";
for (auto loopTree : loopForest->getTrees()) {

 8 lines: Fetch the root of the current tree.------
}
```

llvm::noelle::LoopTree *

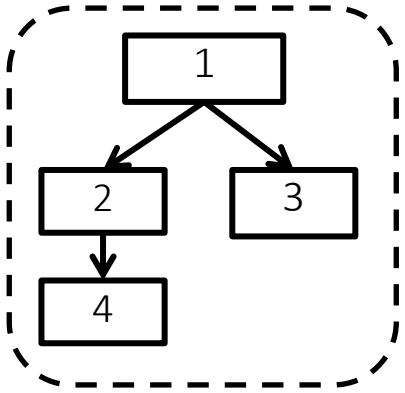
```
void myFunction (){
1: while (…){
2:    while (…){ … }
3:    while (…) {… }
   }
   …
4: for (…){
5:    do {
6:             while(…) {…}
      } while (…)
   }
}
```

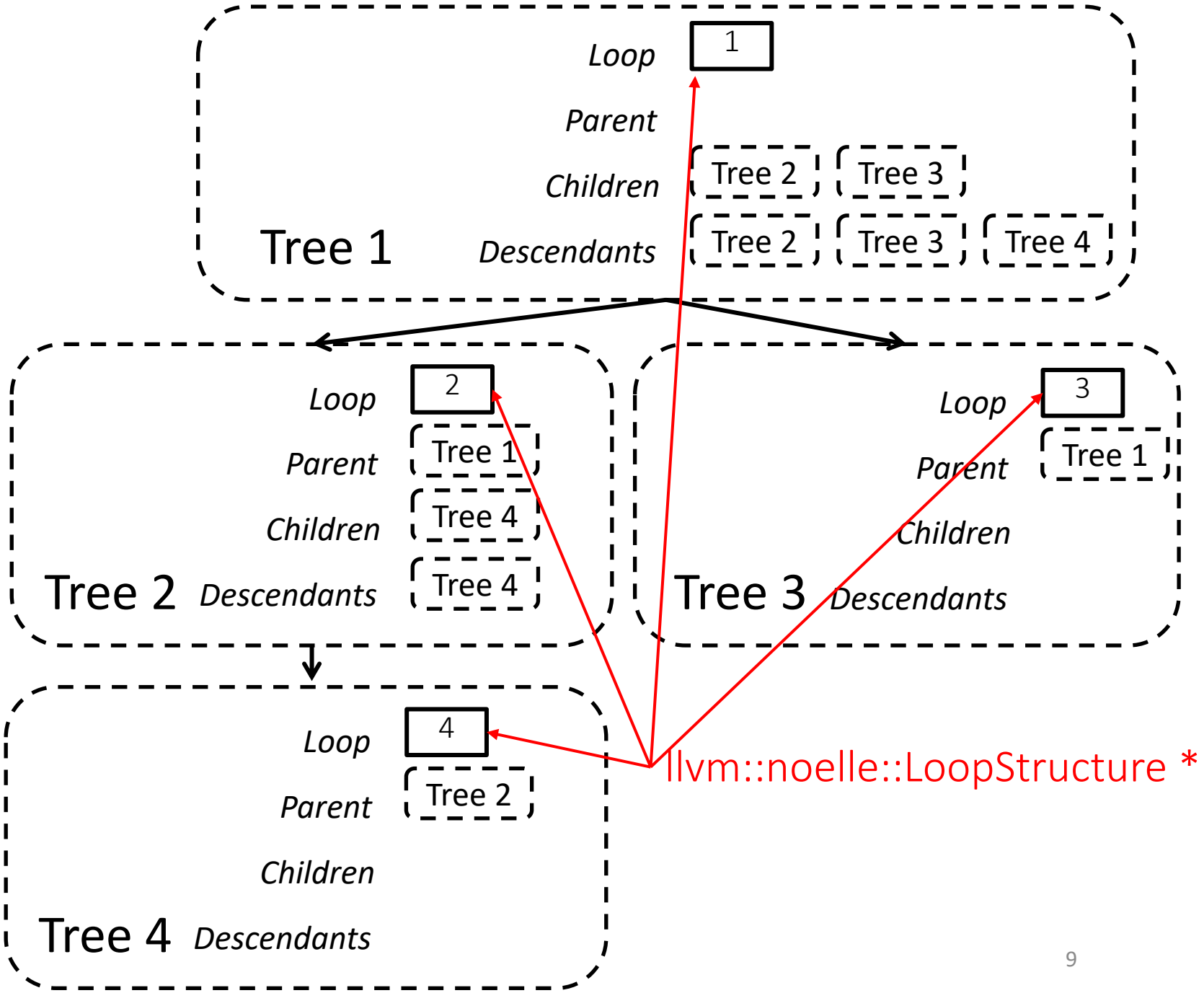


Outline

- Forest of loops with NOELLE
- A tree of loops with NOELLE
- Modifying the forest
- Forest between functions

LoopTree


1
2
3
4


Loop
1
Parent
Children
Tree 2
Tree 3
Tree 1
Descendants
Tree 2
Tree 3
Tree 4
Loop
2
Parent
Tree 1
Children
Tree 4
Tree 2
Descendants
Tree 4
Loop
3
Parent
Tree 1
Children
Tree 3
Descendants
Loop
4
Parent
Tree 2
Children
Tree 4
Descendants
llvm::noelle::LoopStructure *

Traversing loop forest with NOELLE

```
void printTree(LoopTree *n) {

  /*
   * Print the current node.
   */
  auto l = n->getLoop();
  for (auto i = 1; i < l->getNestingLevel(); i++) {
    errs() << "-";
  }
  errs() << "-> ";
  errs() << "[ " << l->getFunction()->getName() << " ] "
         << *l->getEntryInstruction() << "\n";

  /*
   * Print the children
   */
  for (auto c : n->getDescendants()) {
    this->printTree(c);
  }

  return;
}
```

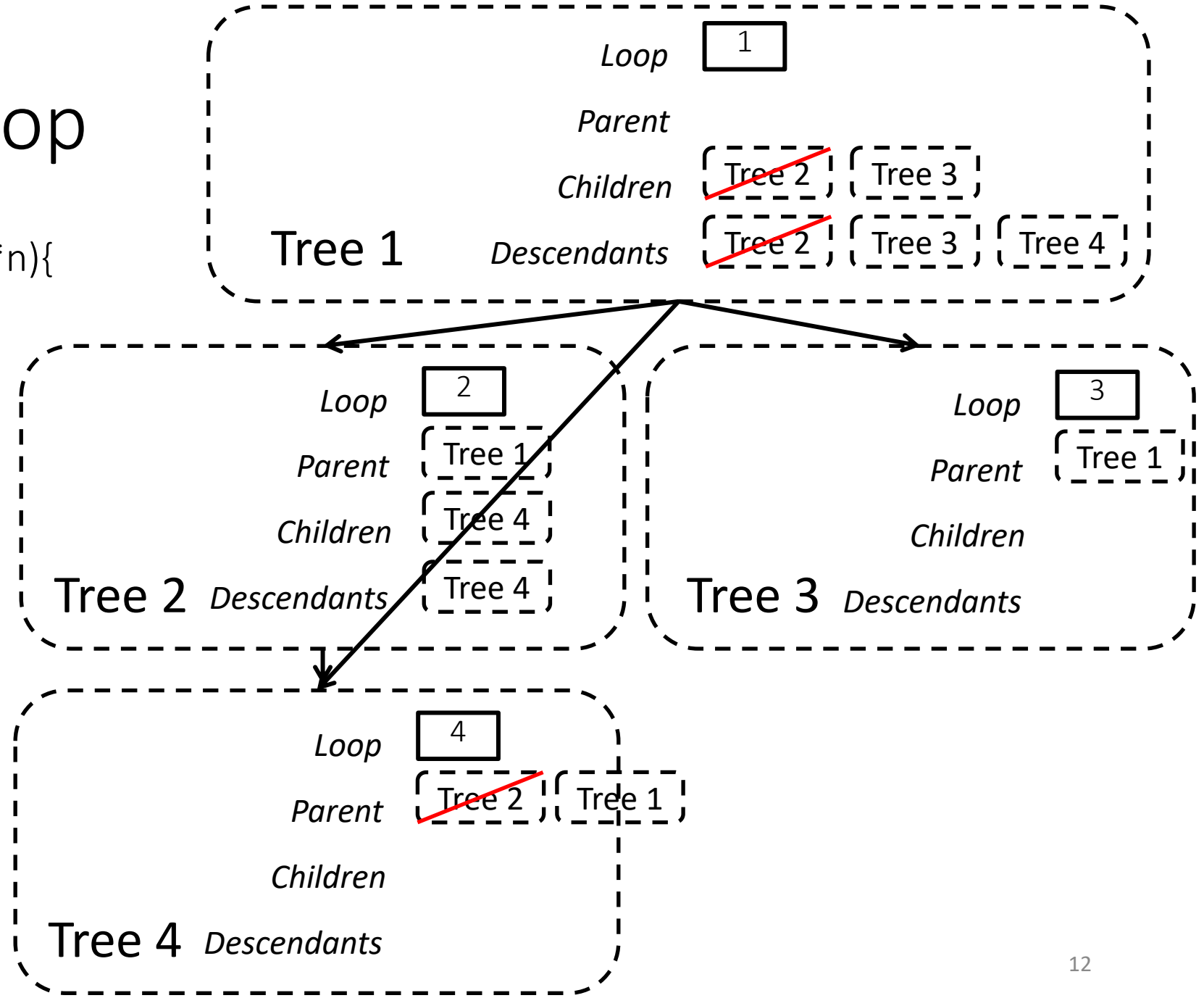
llvm::noelle::LoopStructure *

Outline

- Forest of loops with NOELLE
- A tree of loops with NOELLE
- Modifying the forest
- Forest between functions

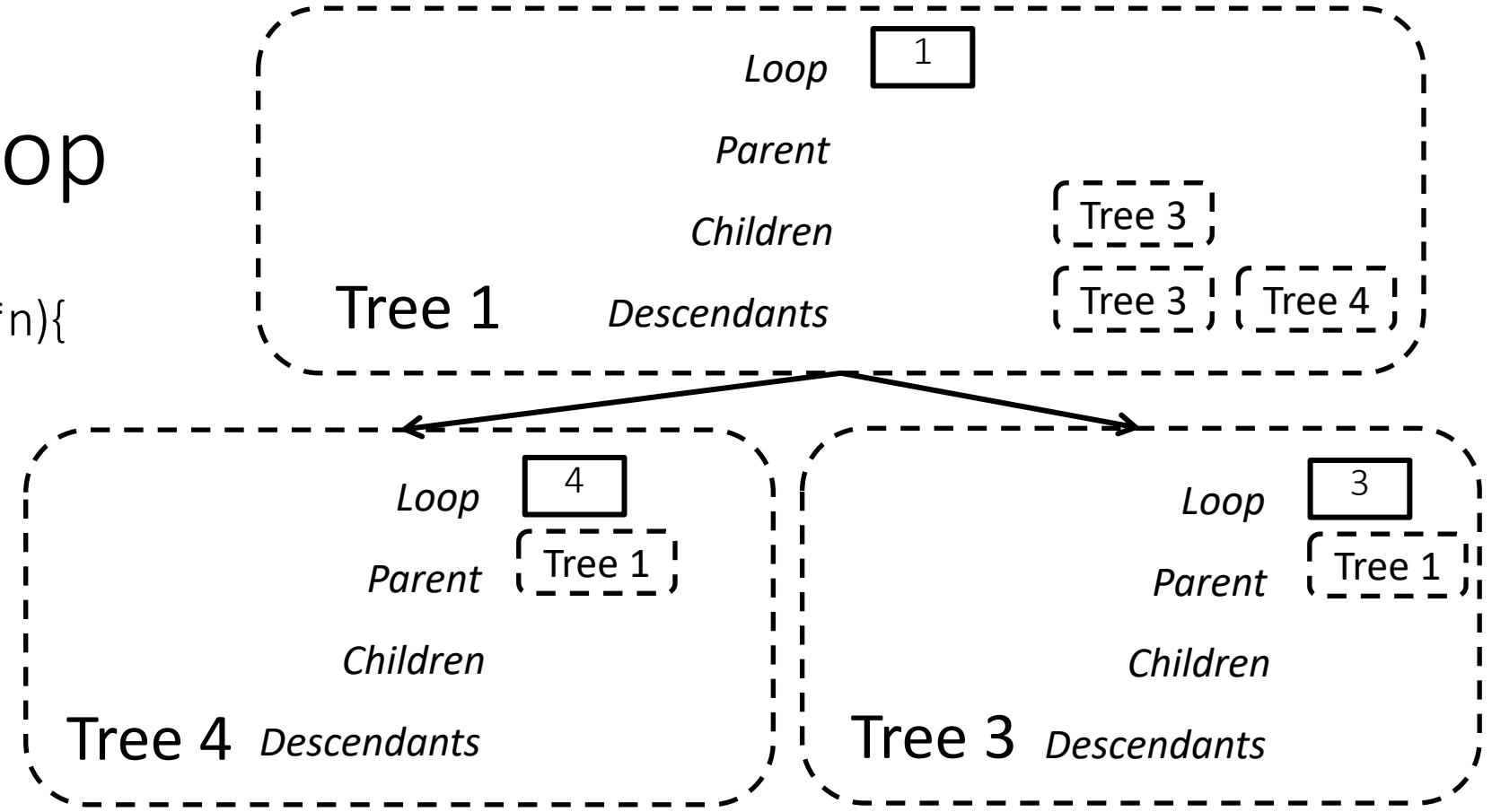
Removing a loop

```
void anExample (LoopTree *n){
   delete n;
}
```



Removing a loop

```
void anExample (LoopTree *n){
  delete n;
}
```



Their nesting level stored in llvm::noelle::LoopStructure didn't change

Outline

- Forest of loops with NOELLE
- A tree of loops with NOELLE
- Modifying the forest
- Forest between functions

Get all program loops with NOELLE

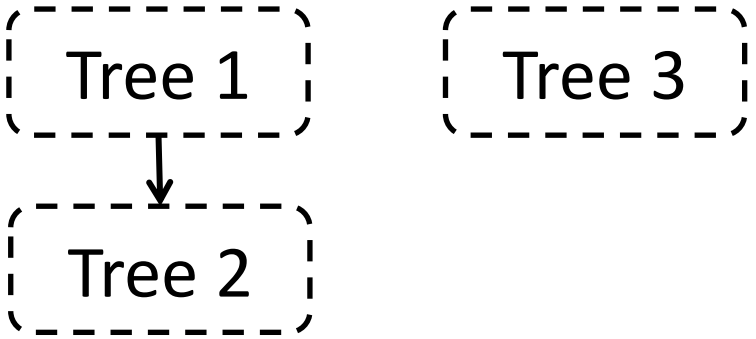
```
/*
 * Fetch the loops with only the loop structure abstraction.
 */
auto loopStructures = noelle.getLoopStructures(mainF);
```

```
/*
 * Fetch the loops with only the loop structure abstraction.
 */
auto loopStructures = noelle.getLoopStructures();
```

No nesting between functions

```
void foo (void){
   1: for (…){
   2:    while (…){
            bar();
         }
      }
}

void bar (void){
   3: for (…) {…}
}
```



Always have faith in your ability

Success will come your way eventually

Best of luck!