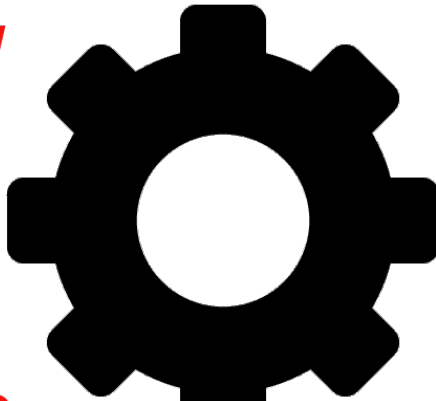


Advanced

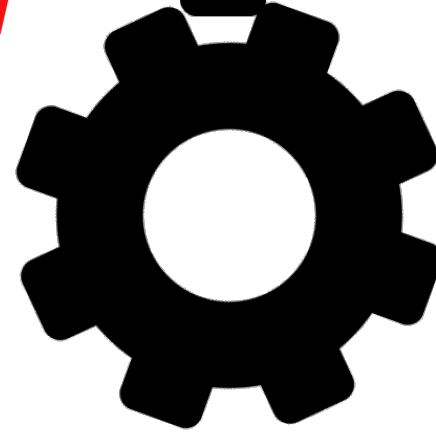
T



pics

in

C



mpilers



Simone Campanoni
simone.campanoni@northwestern.edu

Forest of loops



Multiple loops in NOELLE

- Sources:
src/core/loop_forest
- Main headers:
install/noelle/core/LoopForest.hpp
- Examples of passes using the abstraction:
examples/passes/loop_forest


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);
```

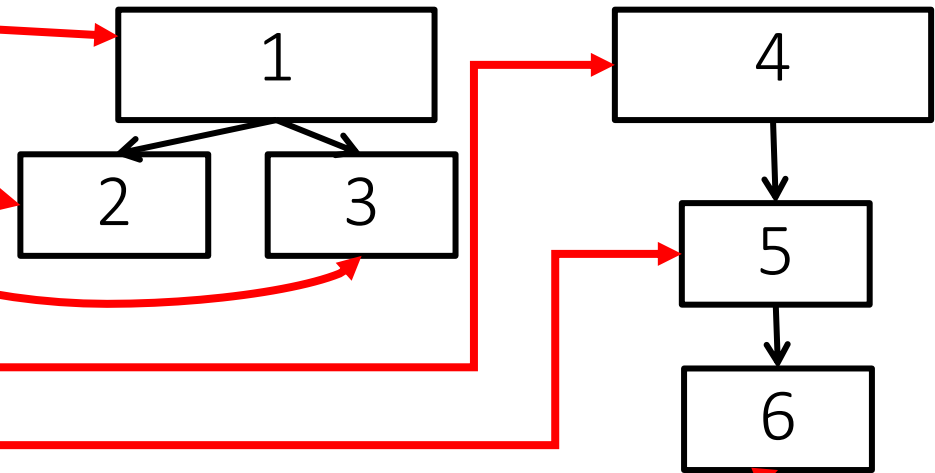


Each loop is an instance of `arcana::noelle::LoopStructure`
Flat representation of the loops

*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 (...)  
   }  
}
```



Outermost
loops

Innermost
loops

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);
```

[arcana::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";
}
```

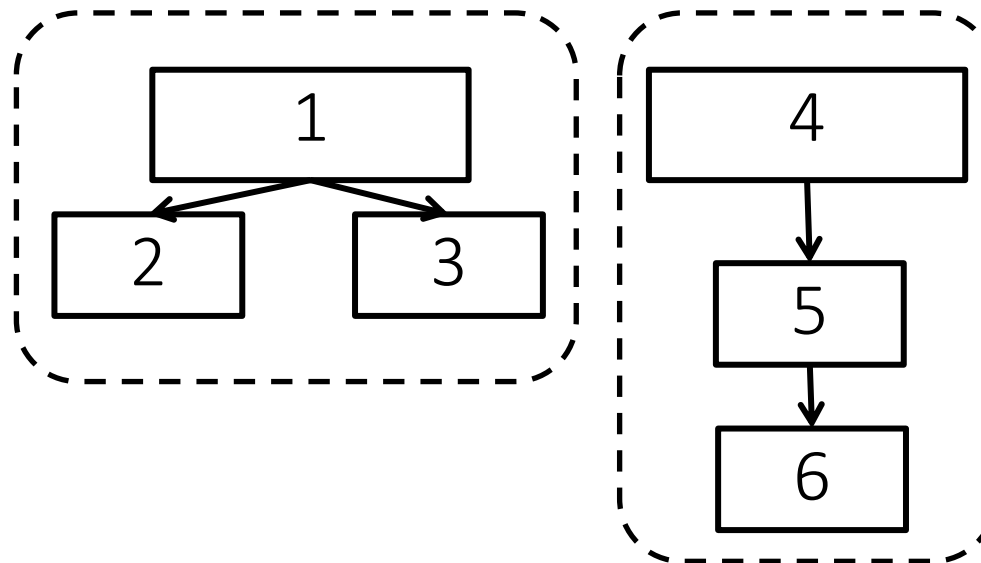
arcana::noelle::LoopTree *

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.-----  
}
```

arcana::noelle::LoopTree *

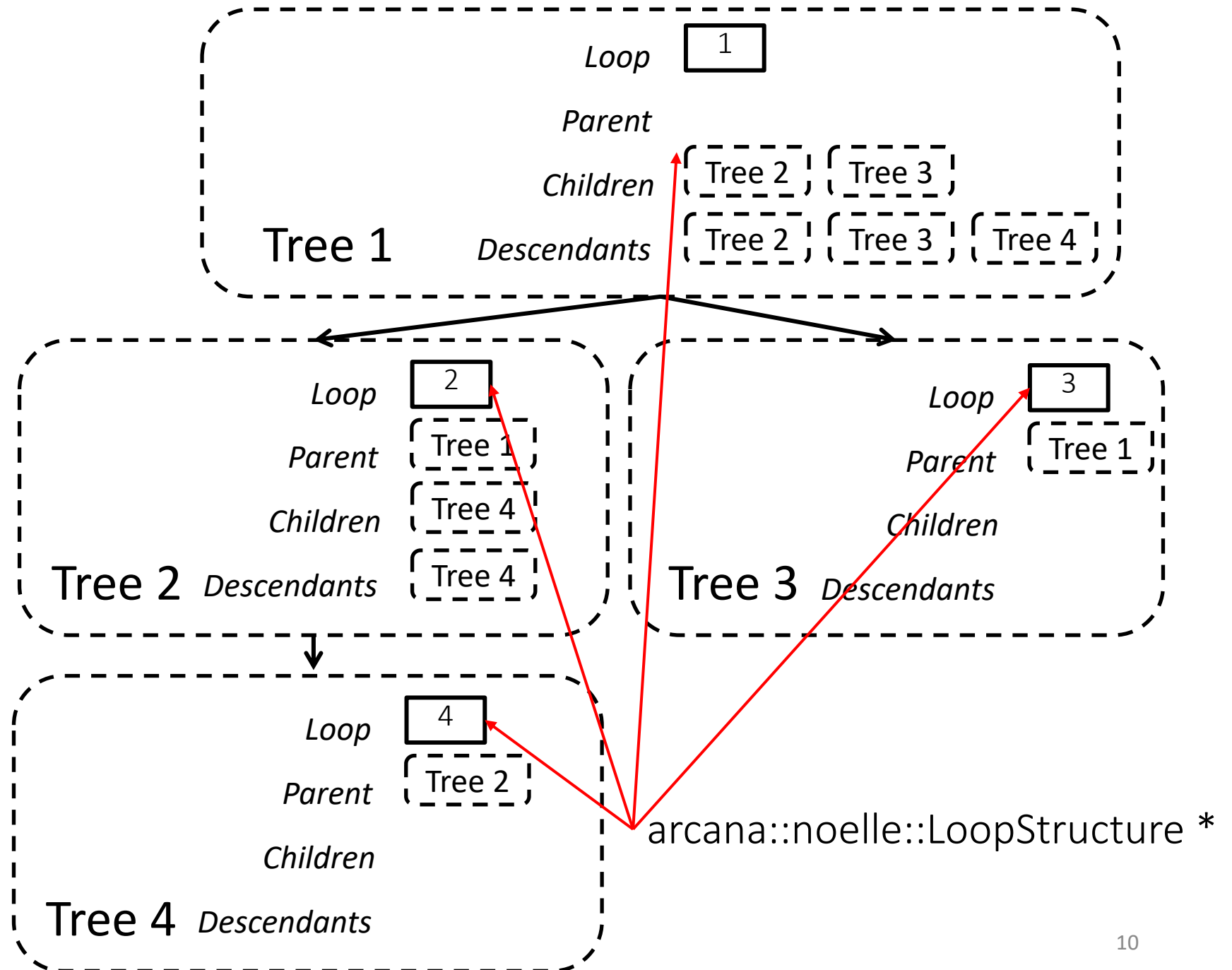
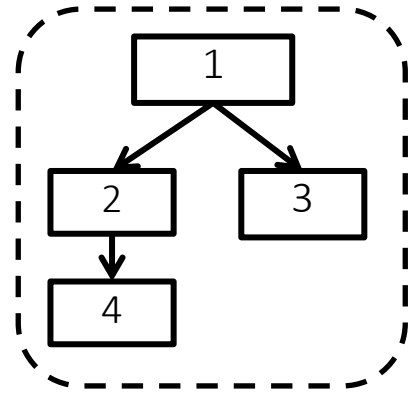
```
void myFunction ()  
{  
1: while (...){  
2:   while (...){ ... }  
3:   while (...){ ... }  
   }  
   ...  
4: for (...){  
5:   do {  
6:     while(...) {...}  
     } while (...)  
   }  
}
```



Outline

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


LoopTree



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;  
}
```

arcana::noelle::LoopStructure *

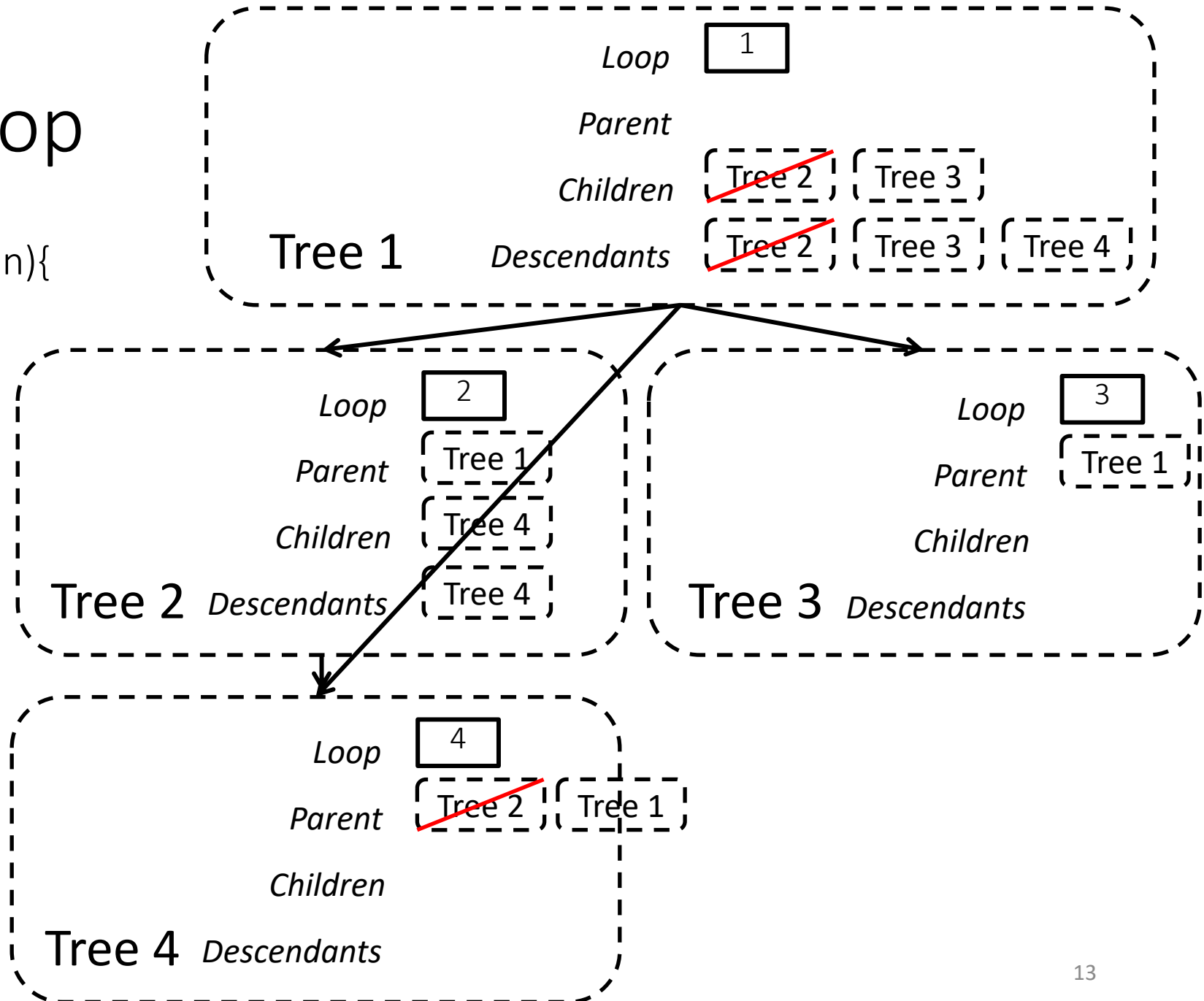


Outline

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

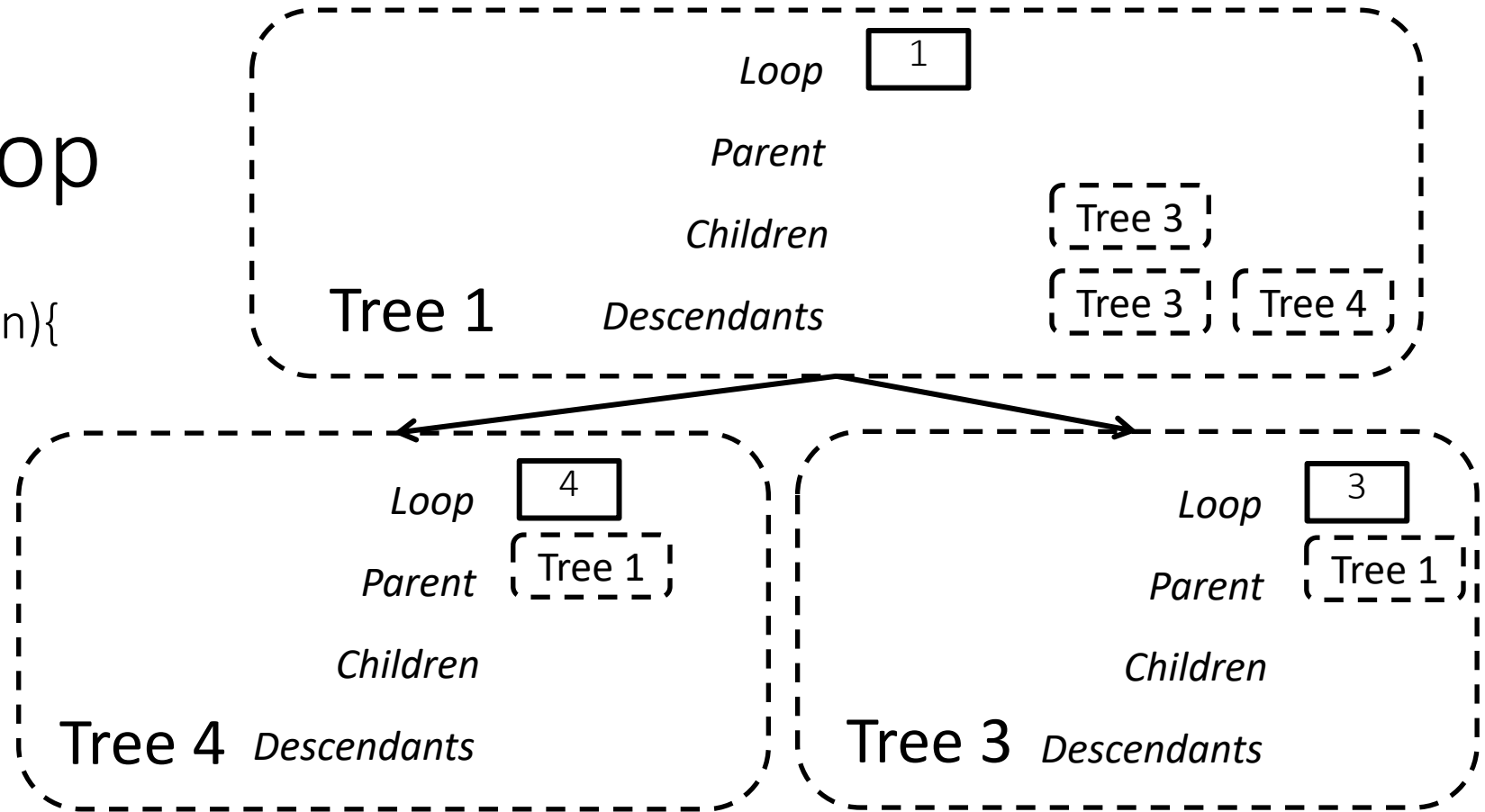
Removing a loop

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



Removing a loop

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



Their nesting level stored in `arcana::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

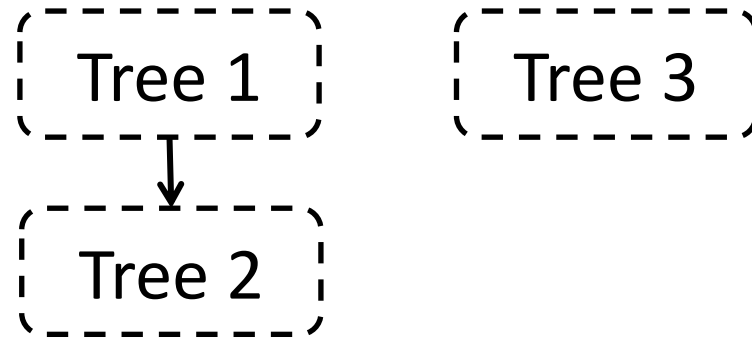
```
/*  
 * Fetch the loops with only the loop structure abstraction.  
 */  
auto loopStructures = noelle.getLoopStructures(mainF);
```

```
/*  
 * 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!