Advanced Topics *in* Compilers

LoopEnvironment

Simone Campanoni
simone.campanoni@northwestern.edu

Outline

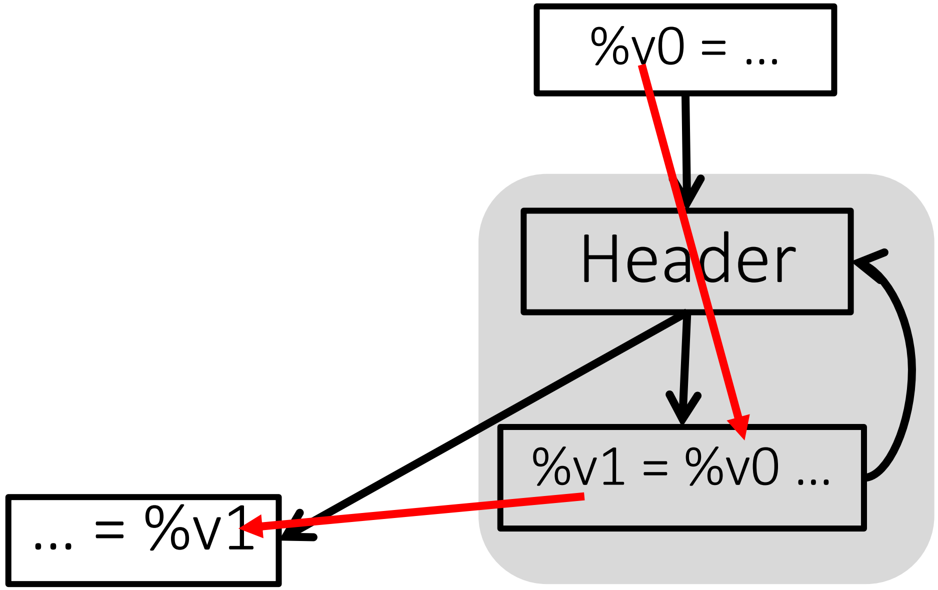
- What is it and why NOELLE provides it

- Live-In and Live-Out variables

- Producers and consumers

LoopEnvironment

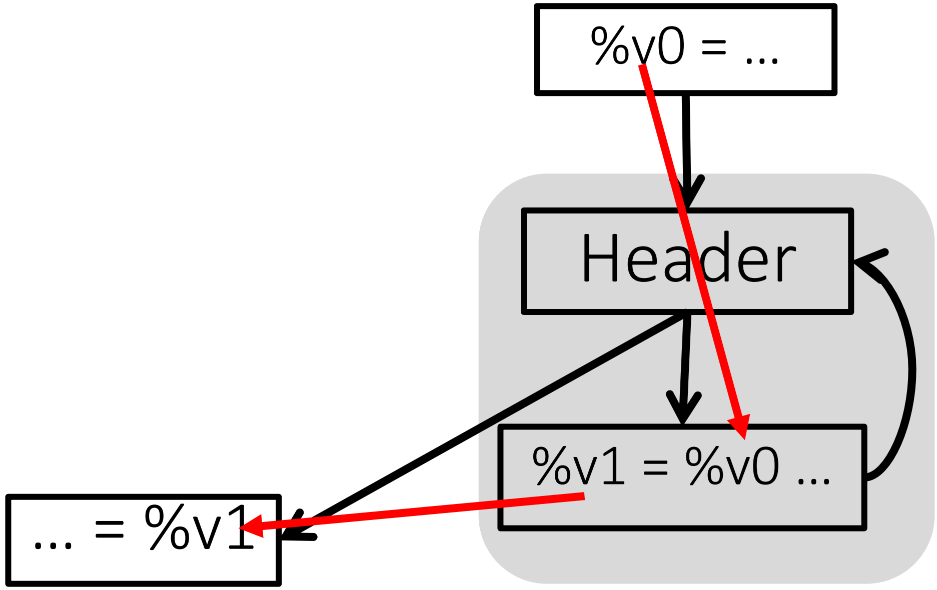
- It captures the data-flows generated by writing and reading variables to and from a loop



*It **only** captures data-flow triggered by accesses to variables, **not** memory*

LoopEnvironment

- It captures the data-flows generated by writing and reading variables to and from a loop



ID	Value	Type	Live-In ?
0	%v0	i64	True
1	%v1	I64	False

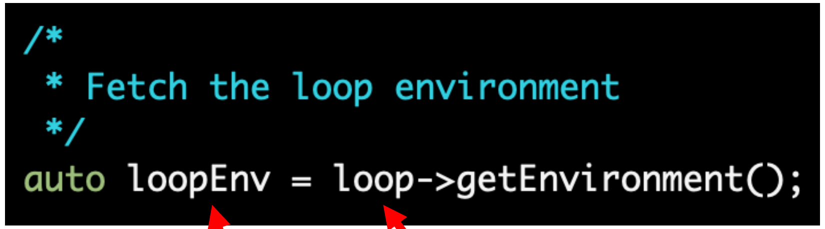
Outline

- What is it and why NOELLE provides it

- Live-In and Live-Out variables

- Producers and consumers

Live-In and Live-Out

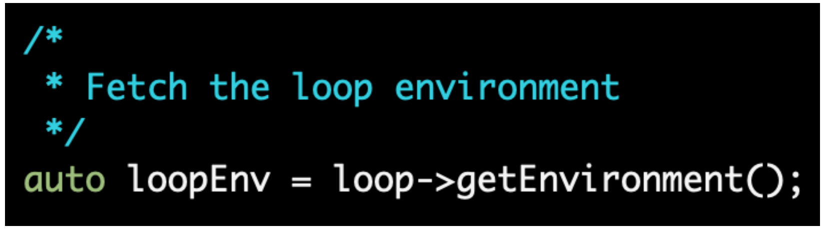


```
/*
 * Fetch the loop environment
 */
auto loopEnv = loop->getEnvironment();
```

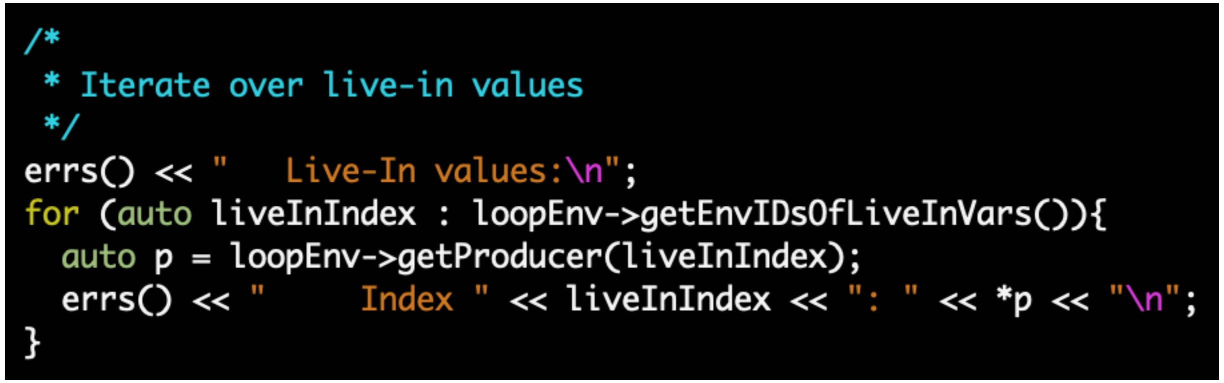
Instance of the class arcana::noelle::LoopDependenceInfo

Instance of the class arcana::noelle::LoopEnvironment

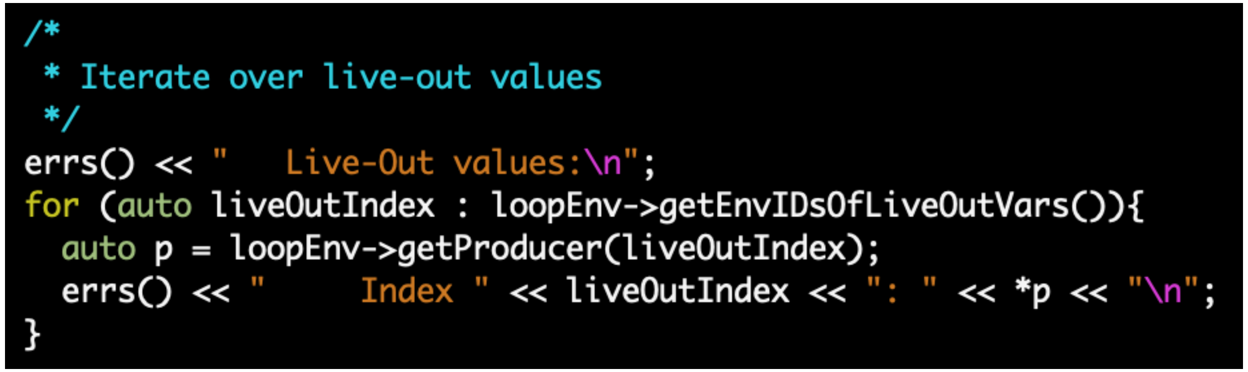
Live-In and Live-Out



```
/*
 * Fetch the loop environment
 */
auto loopEnv = loop->getEnvironment();
```



```
/*
 * Iterate over live-in values
 */
errs() << "   Live-In values:\n";
for (auto liveInIndex : loopEnv->getEnvIDsOfLiveInVars()){
  auto p = loopEnv->getProducer(liveInIndex);
  errs() << "     Index " << liveInIndex << ": " << *p << "\n";
}
```



```
/*
 * Iterate over live-out values
 */
errs() << "   Live-Out values:\n";
for (auto liveOutIndex : loopEnv->getEnvIDsOfLiveOutVars()){
  auto p = loopEnv->getProducer(liveOutIndex);
  errs() << "     Index " << liveOutIndex << ": " << *p << "\n";
}
```

Outline

- What is it and why NOELLE provides it

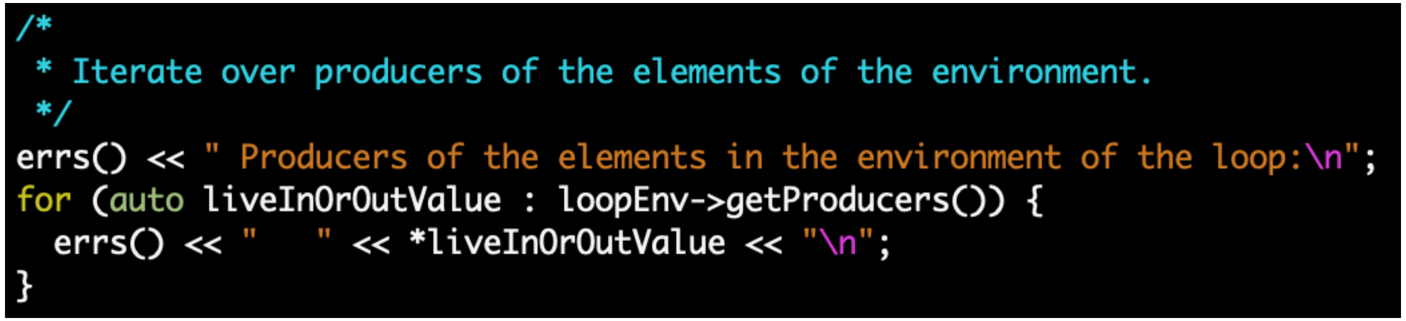
- Live-In and Live-Out variables

- Producers and consumers

Producers and consumers

Elements that compose the environment have producers and consumers

- Producers: instructions that define variables that compose the environment
- Consumers: instructions that use variables that compose the environment



```
/*
 * Iterate over producers of the elements of the environment.
 */
errs() << " Producers of the elements in the environment of the loop:\n";
for (auto liveInOrOutValue : loopEnv->getProducers()) {
  errs() << "    " << *liveInOrOutValue << "\n";
}
```

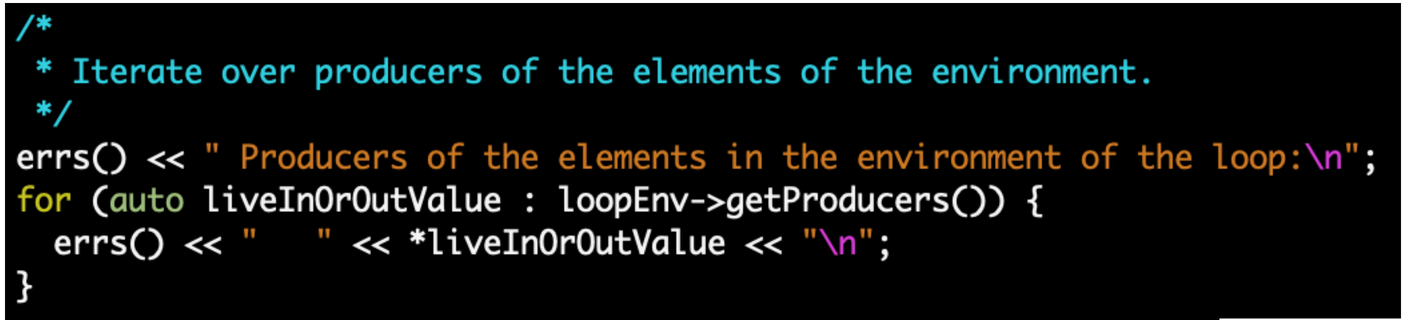
How many producers per environment element?

1 because the IR is in SSA

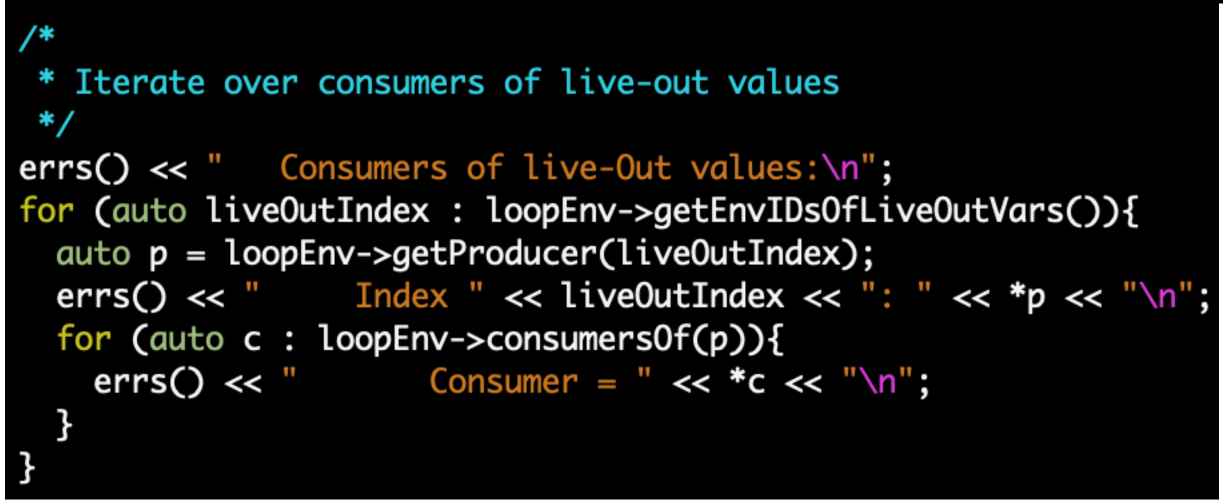
Producers and consumers

Elements that compose the environment have producers and consumers

- Producers: instructions that define variables that compose the environment
- Consumers: instructions that use variables that compose the environment



```
/*
 * Iterate over producers of the elements of the environment.
 */
errs() << " Producers of the elements in the environment of the loop:\n";
for (auto liveInOrOutValue : loopEnv->getProducers()) {
  errs() << "   " << *liveInOrOutValue << "\n";
}
```



```
/*
 * Iterate over consumers of live-out values
 */
errs() << "   Consumers of live-Out values:\n";
for (auto liveOutIndex : loopEnv->getEnvIDsOfLiveOutVars()){
  auto p = loopEnv->getProducer(liveOutIndex);
  errs() << "     Index " << liveOutIndex << ": " << *p << "\n";
  for (auto c : loopEnv->consumersOf(p)){
    errs() << "       Consumer = " << *c << "\n";
  }
}
```

Always have faith in your ability

Success will come your way eventually

Best of luck!