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
LoopEnvironment

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

<table>
<thead>
<tr>
<th>Index</th>
<th>Value</th>
<th>Type</th>
<th>Live-In ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>%v0</td>
<td>i64</td>
<td>True</td>
</tr>
<tr>
<td>1</td>
<td>%v1</td>
<td>i64</td>
<td>False</td>
</tr>
</tbody>
</table>
Outline

• What is it and why NOELLE provides it

• Live-In and Live-Out variables

• Producers and consumers
Live-In and Live-Out

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

Instance of the class `llvm::noelle::LoopDependenceInfo`

Instance of the class `llvm::noelle::LoopEnvironment`
Live-In and Live-Out

```c++
/*
 * 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

```c++
/*
 * 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

```c
/*
 * 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!