

analyzer Nullable
analysis direction:forward
lattice element type:Map<node<>,NullableState>
use instructions fromNullableInstructions
construction parameters
<< ... >>

```
initial(program)->Map<node<>,NullableState>{  
    Map<node<>,NullableState>result = newHashMap<node<>,NullableState>();  
    result;  
}
```

```
merge(program, input)->Map<node<>,NullableState> {  
    Map<node<>, NullableState>result=newHashMap<node<>,NullableState>();  
    for (Map<node<>, NullableState> inputElement:input) {  
        for (Map.Entry<node<>,NullableState> entry:inputElement.entrySet()) {  
            node<> expr = entry.getKey();  
            NullableState value = entry.getValue();  
            NullableState resValue = result.get(expr);  
            if (resValue == null) {  
                resValue = NullableState.NOT_INIT;  
            }  
            result.put(expr, resValue.merge(value));  
        }  
    }  
    result;  
}
```

```
fun(state, input, stateValues)->Map<node<>,NullableState> {  
    Map<node<>,NullableState> result = input;  
    Instruction instruction = state.getInstruction();  
    NullableState nullableState = NullableState.UNKNOWN;  
    if (instruction instanceof GeneratedInstruction) {  
        node<> node = (node<>)(((GeneratedInstruction)instruction).getParameter());  
        if (instruction is notNull) {  
            nullableState=NullableState.NOTNULL;  
        }  
        if (instruction is nullable) {  
            nullableState = NullableState.NULLABLE;  
        }  
        if (instruction is null) {  
            nullableState = NullableState.NULL;  
        }  
        if (node.isInstanceOf(VariableReference)) {  
            node = node:VariableReference.variableDeclaration;  
        }  
        if (node != null) {  
            result.put(node,nullableState);  
        }  
    }  
    if (instruction instanceof WriteInstruction) {  
        WriteInstruction write = (WriteInstruction)instruction;  
        node<>value = (node<Expression>)write.getValue();  
        if (value.isInstanceOf(VariableReference)) {  
            value = value:VariableReference.variableDeclaration;  
        }  
        NullableState valueState = result.get(value);  
        if (valueState == null){  
            valueState = NullableState.UNKNOWN;  
        }  
    }
```

```
        result.put((node<>)write.getVariable(),valueState);
    }
    result;
}
```