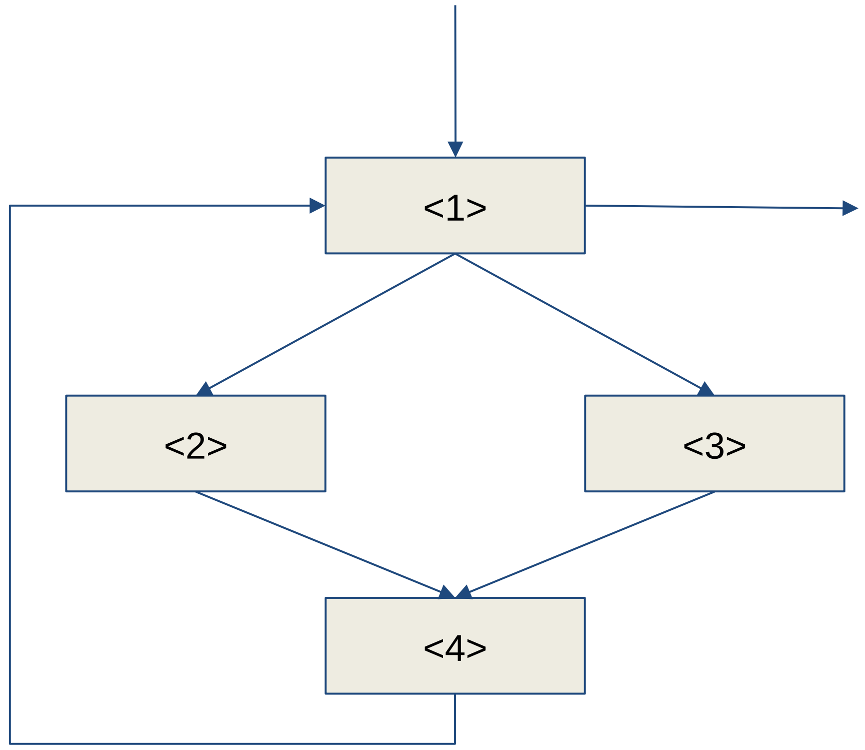
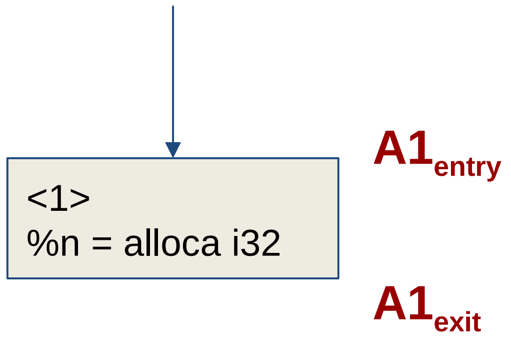
Abstract Interpretation

Demo

Example Program



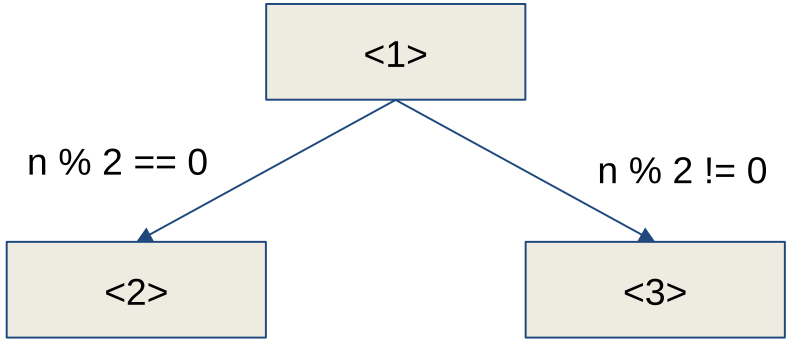
Example Program



$A1_{entry}$: { }

$A1_{exit}$: n => {ODD, EVEN}

Example Program

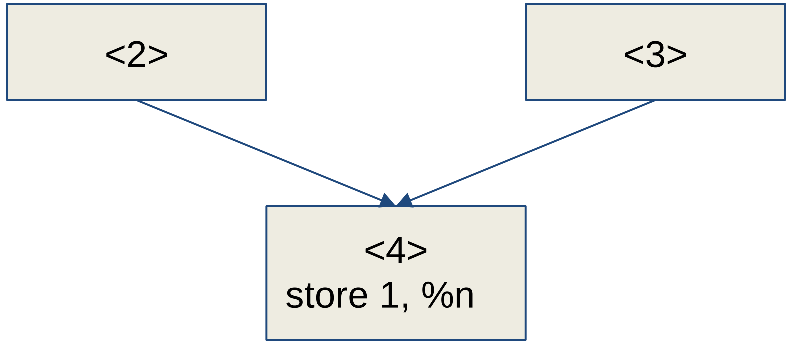


$A1_{exit}$: n => {ODD, EVEN}

$A2_{entry}$: n => {EVEN}

$A3_{entry}$: n => {ODD}

Example Program



$A2_{exit}$: n => {EVEN}

$A3_{exit}$: n => {ODD}

$A4_{entry}$: n => {ODD, EVEN} (A2 U A3)

$A4_{exit}$: n => {ODD}

Example Program

Round0:

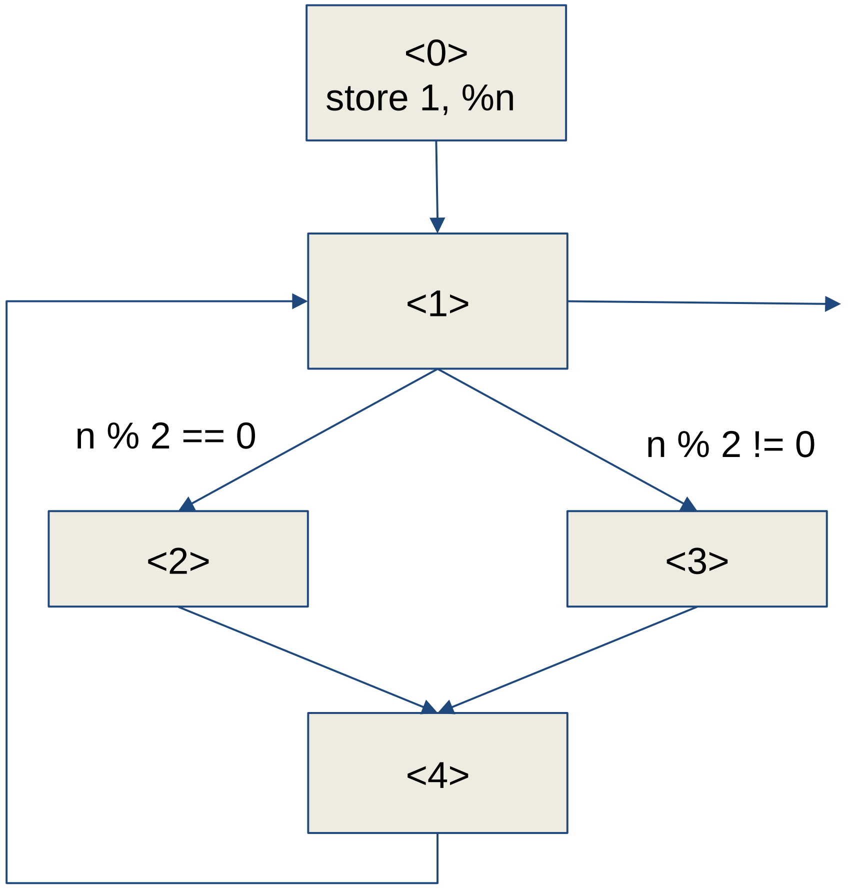
$A0_{exit}$: n => {}

$A1_{exit}$: n => {}

$A2_{exit}$: n => {}

$A3_{exit}$: n => {}

$A4_{exit}$: n => {}



Example Program

Round1:

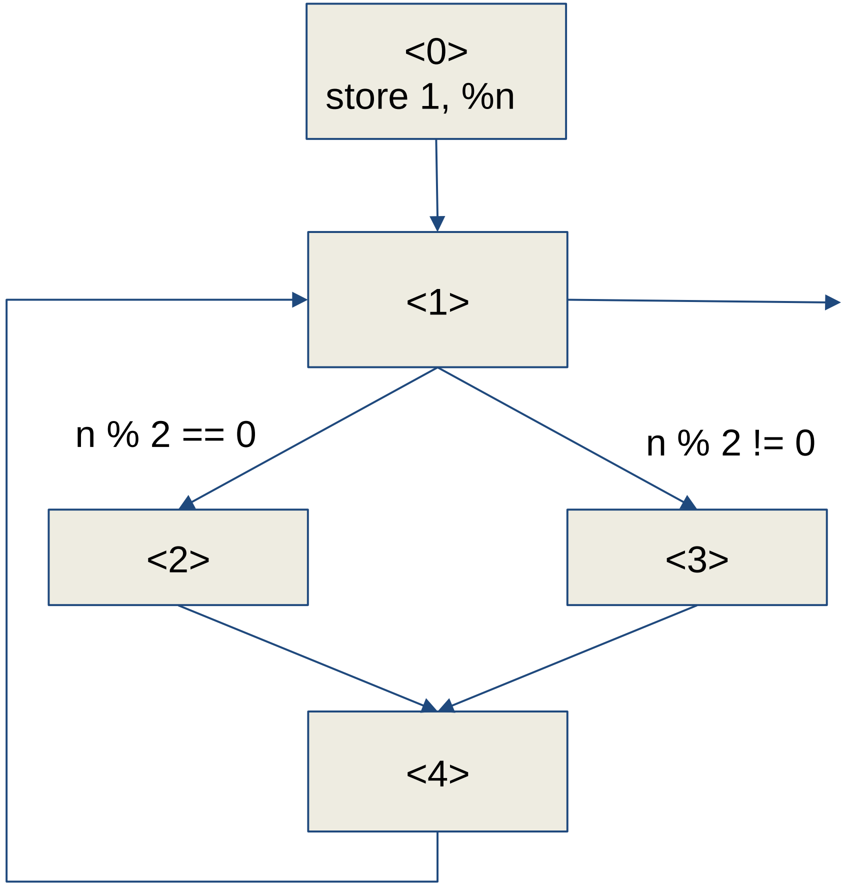
$A0_{exit}$: n => {ODD}

$A1_{exit}$: n => {ODD}

$A2_{exit}$: n => {EVEN}

$A3_{exit}$: n => {ODD}

$A4_{exit}$: n => {ODD,EVEN}



Example Program

Round2:

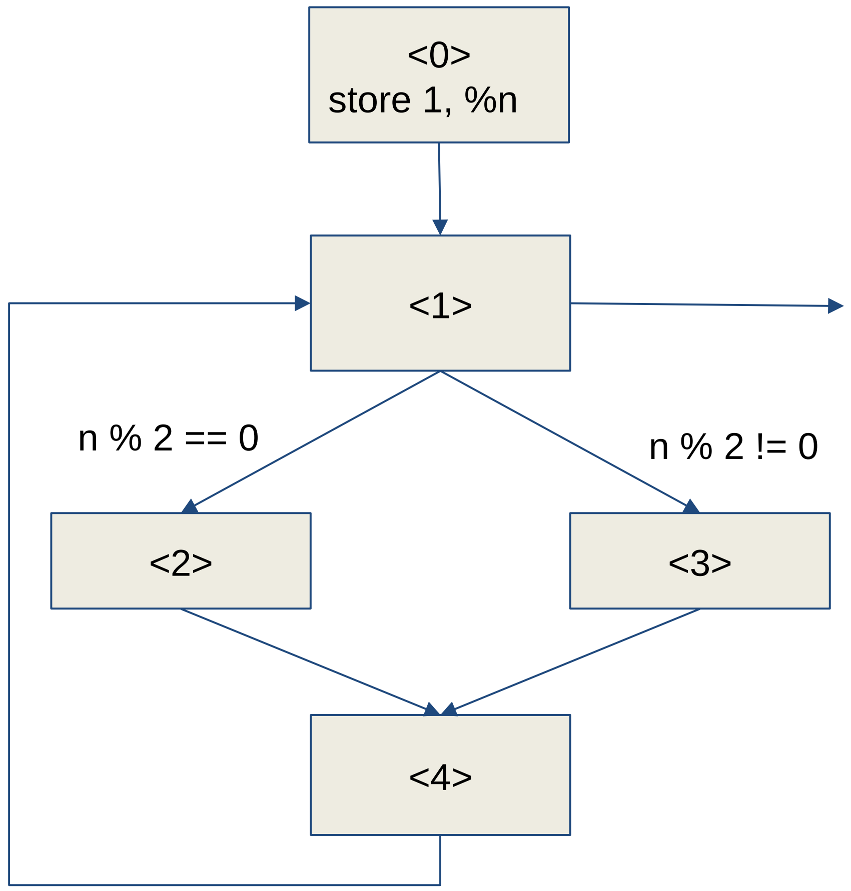
$A0_{exit}$: n => {ODD}

$A1_{exit}$: n => {ODD,EVEN}

$A2_{exit}$: n => {EVEN}

$A3_{exit}$: n => {ODD}

$A4_{exit}$: n => {ODD,EVEN}



General Algorithm

```
oldAnalysisMap = NULL
analysisMap = initialAnalysis( )

while( ! fixPointReached(analysisMap, oldAnalysisMap) )
{
    oldAnalysisMap = analysisMap
    updateGraphAnalysis( analysisMap,  Function)
}
```

Fixpoint Check

```
bool fixPointReached(analysisMap, oldAnalysisMap)
{
    if (oldAnalysisMap.empty() ) return false;
    for ( $I_1$ ∈ analysisMap & $I_2$ ∈ oldAnalysisMap)
        if( $I_1$ != $I_2$ )
            return false;
    return true;
}
```

Update Graph Analysis

```
void updateGraphAnalysis( analysisMap, Function ){
    for (BB ∈ Function){
        OldAnalysis = analysisMap [BB];
        EntryAnalysis = emptySet()
        // Load the stored analysis for predecessor nodes
        for (p ∈ pred_begin(BB) )
            PredSet = applyGuard ( analysisMap[ p ], p )
            EntryAnalysis = union_analysis( entryAnalysis, predSet )
        NewExitAnalysis = updateBBAnalysis(BB, EntryAnalysis)
        If ( OldExitAnalysis != NewExitAnalysis )
            analysisMap [BB] = union_analysis(OldExitAnalysis, NewExitAnalysis)
    }
}
```

Update Basic Block Analysis

```
Set updateBBAnalysis (BB, predSet){
    for (Ins ∈ BB){
        if (Ins is storeInst)   // Update predSet accordingly
        if (Ins is loadInst)    // Update predSet accordingly
        if (Ins is cmpInst)     // Update predSet accordingly
        if (Ins is binaryInst) // Update predSet accordingly
    }
    return predSet
}
```