LLVM code:

```
; ModuleID = 'TestModule'

define void @f() {
L0:
  %t0 = alloca i32
  %t1 = alloca i32
  %t2 = alloca i32
  %t3 = load i32* %t0
  %t4 = icmp eq i32 %t3, 7
  br i1 %t4, label %L2, label %L1

L1:
  %t5 = load i32* %t0
  %t6 = add i32 %t5, 1
  store i32 %t6, i32* %t1
  br label %L3

L2:
  %t7 = load i32* %t2
  %t8 = icmp eq i32 %t7, 7
  br i1 %t8, label %L4, label %L5

L3:
  store i32 1, i32* %t2
  ret void

L4:
  store i32 0, i32* %t1
  br label %L5

L5:
  br label %L3
}
```

Command:

```
cat test.ll | opt -load ./reaching-def.so -FuriosA
```

Result:

```
Pass on function f
GEN for L5 = { }
GEN for L4 = { }
GEN for L3 = { }
GEN for L2 = { t7 t8 }
GEN for L1 = { t5 t6 }
GEN for L0 = { t0 t1 t2 t3 t4 }
Iterations: 3
Basic Block L5:
IN = { t0 t1 t2 t3 t4 t7 t8 }
OUT = { t0 t1 t2 t3 t4 t7 t8 }
Basic Block L4:
IN = { t0 t1 t2 t3 t4 t7 t8 }
OUT = { t0 t1 t2 t3 t4 t7 t8 }
Basic Block L3:
IN = { t0 t1 t2 t3 t4 t5 t6 t7 t8 }
OUT = { t0 t1 t2 t3 t4 t5 t6 t7 t8 }
Basic Block L2:
IN = { t0 t1 t2 t3 t4 }
OUT = { t0 t1 t2 t3 t4 t7 t8 }
Basic Block L1:
IN = { t0 t1 t2 t3 t4 }
OUT = { t0 t1 t2 t3 t4 t5 t6 }
Basic Block L0:
IN = { }
OUT = { t0 t1 t2 t3 t4 }
```

I got the GEN & IN & OUT of 5 Basic Blocks with function of 3 iterations.