Register Allocation 1-0-1

Right shift algorithm:

```
Source ← Mem[Value1]
Param ← Mem[Value2]
Res ← 0

Smask ← 1
Dmask ← 1

While Param > 0:
      Smask ← Smask << 1
      Param ← Param – 1

While Smask /= 0:
      Comp ← Source & Smask
      If Comp /= 0:
            Res += Dmask
      Smask ← Smask << 1
      Dmask ← Dmask << 1

Mem[Result] ← Res
```

“Naive” allocation: one register per variable

```
R0: Source
R1: Param
R2: Res
R3: Smask
R4: Dmask
R5: Comp
```

Problem: It’s too much! We can’t use R5.

Solution: The easy way to liveness analysis. Build a table with variables – and fill a column from the first point we need a variable to the last. Allocate the first available register which is not in use by any other variable.

Rule for loops: a variable touched inside a loop extends at least until the end of this loop.