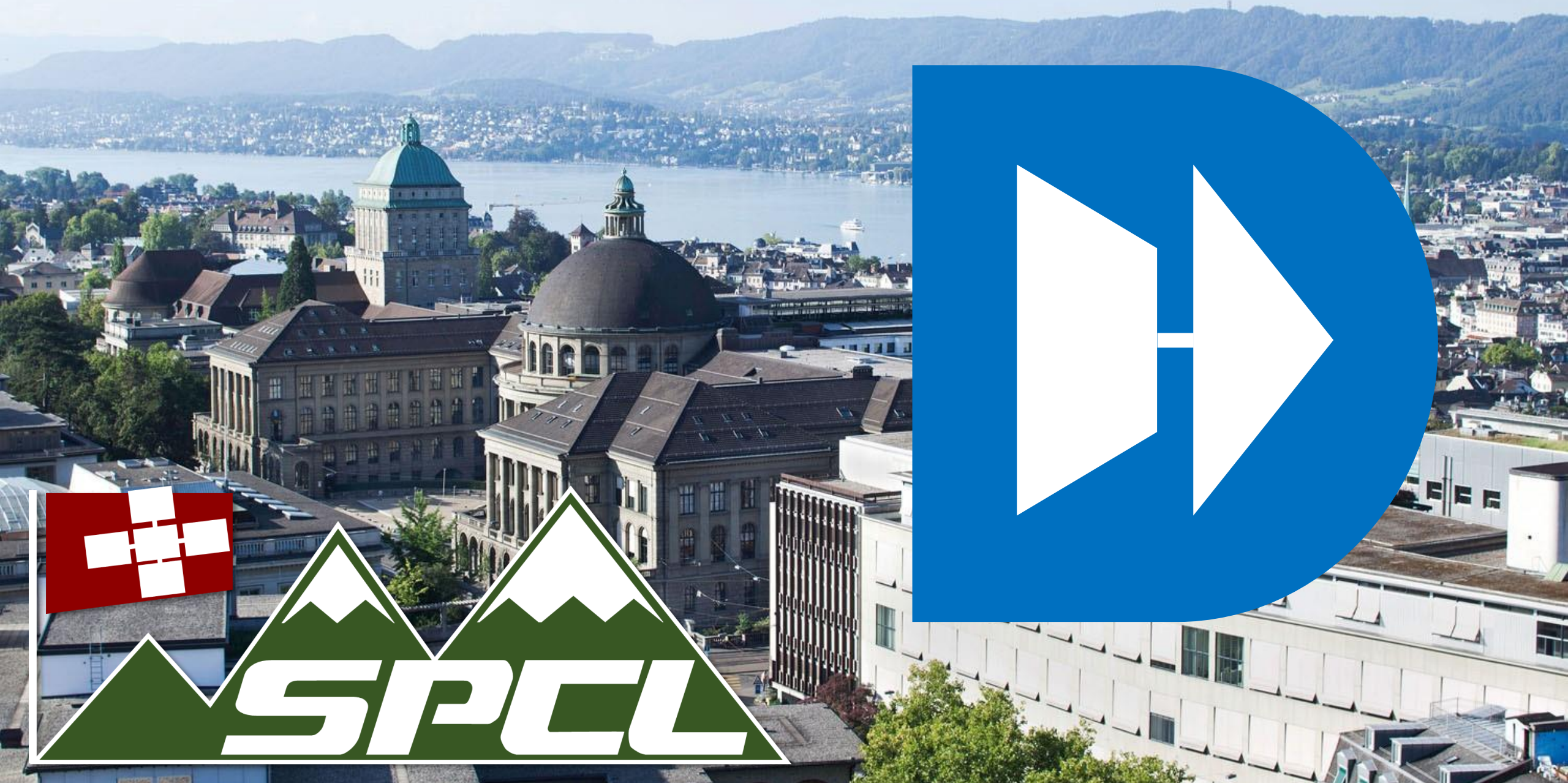
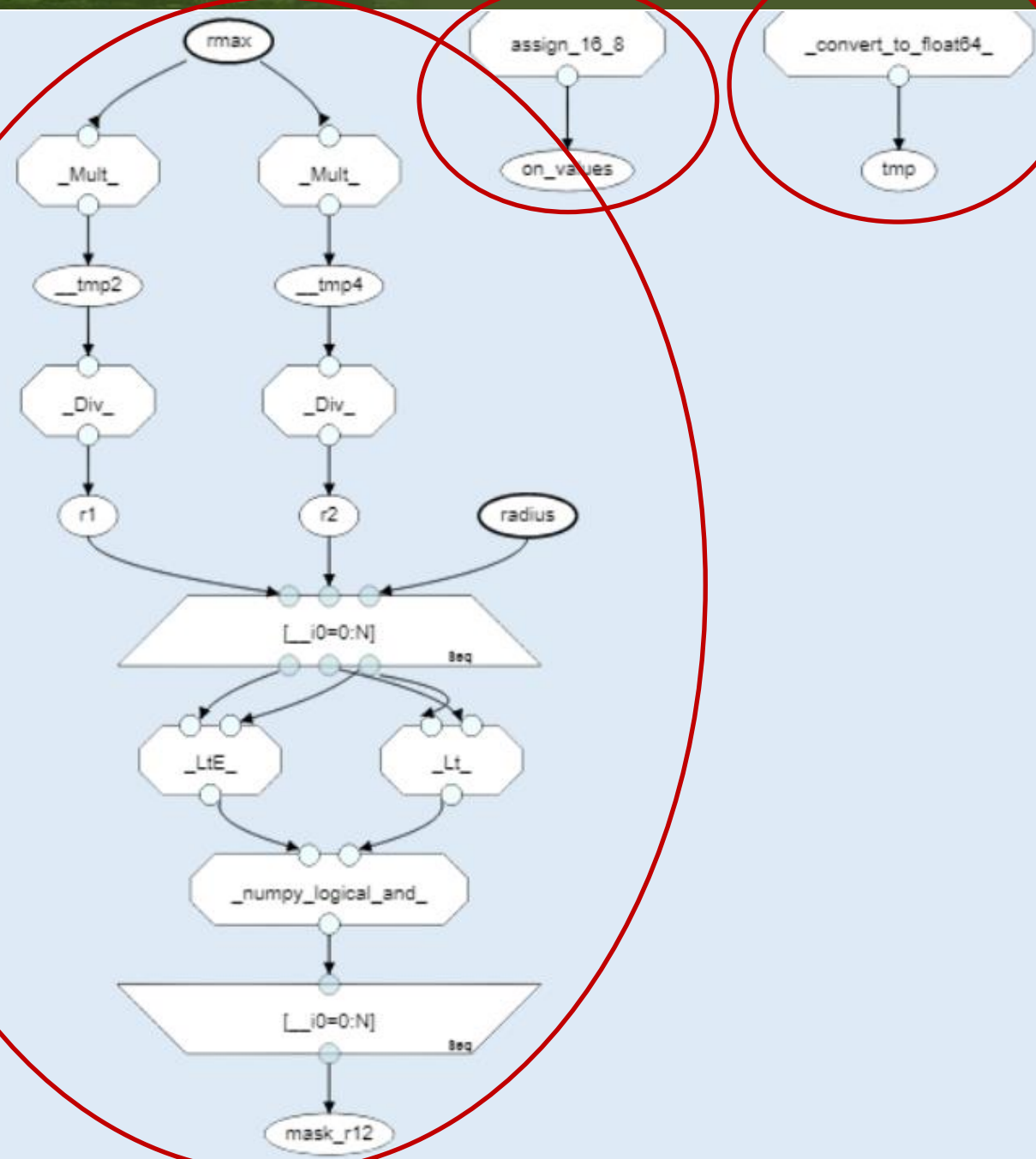


Alexandru Calotoiu



# Canonicalization of SDFGs

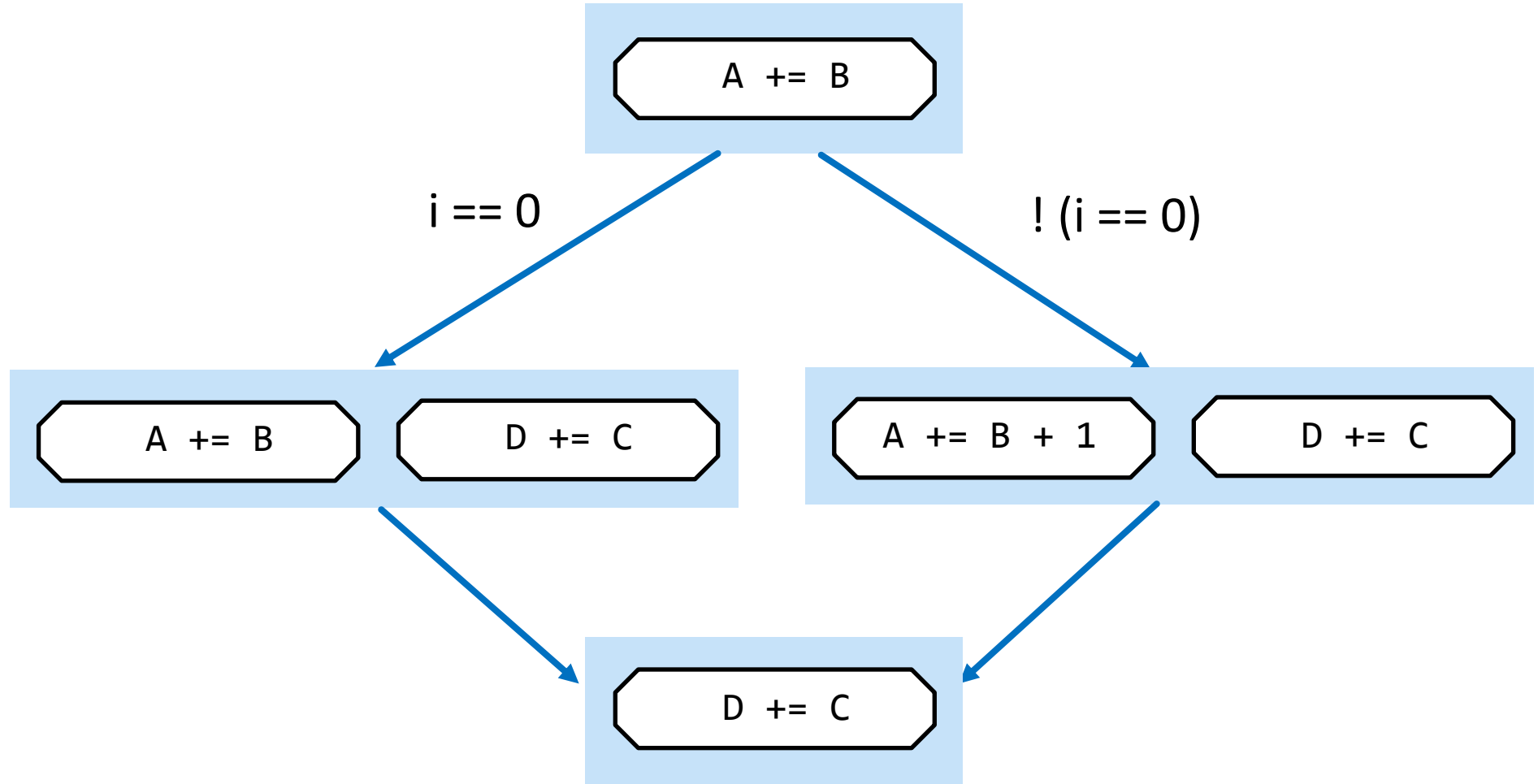
- What are Basic Block in SDFGs?
  - Proposal:  
*Components connected **only** by dataflow*
- States are obstacles
  - Increasing “fusibility” of SDFGs is always desirable (?)



# Irreducible control flow and optimization

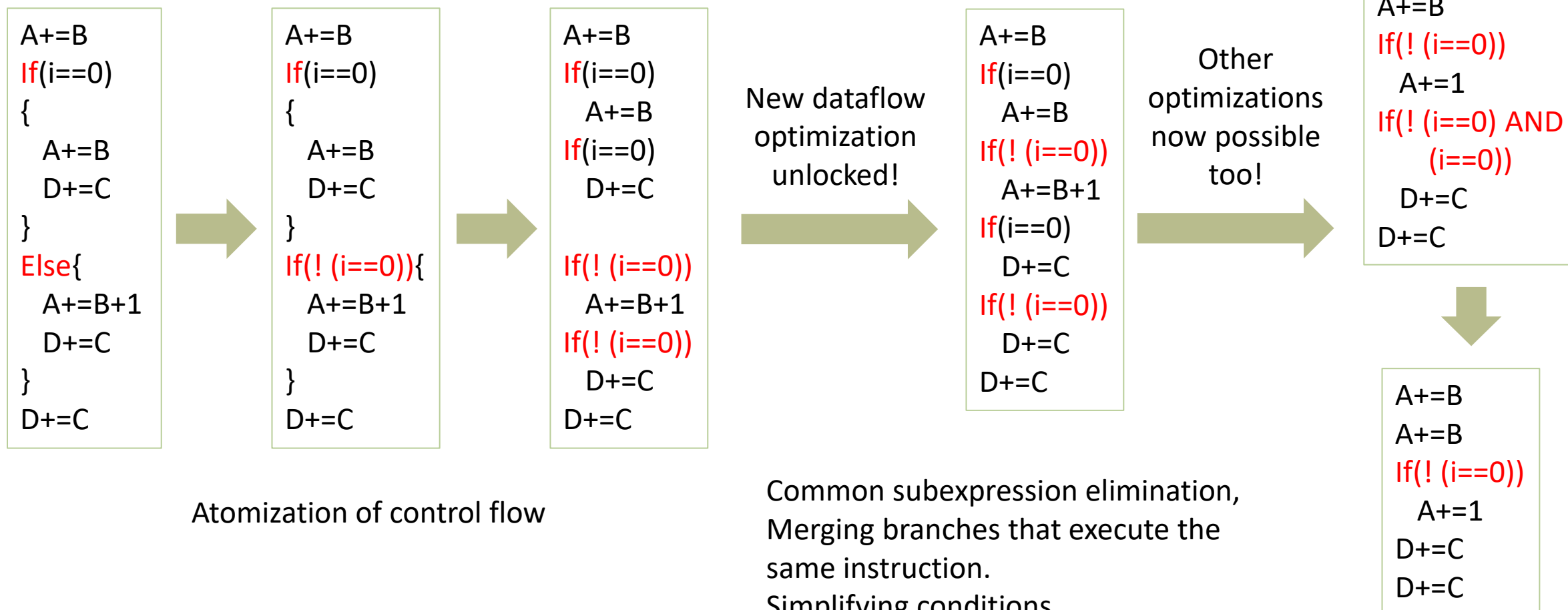
```

A+=B
If(i==0)
{
  A+=B
  D+=C
}
Else{
  A+=B+1
  D+=C
}
D+=C
  
```



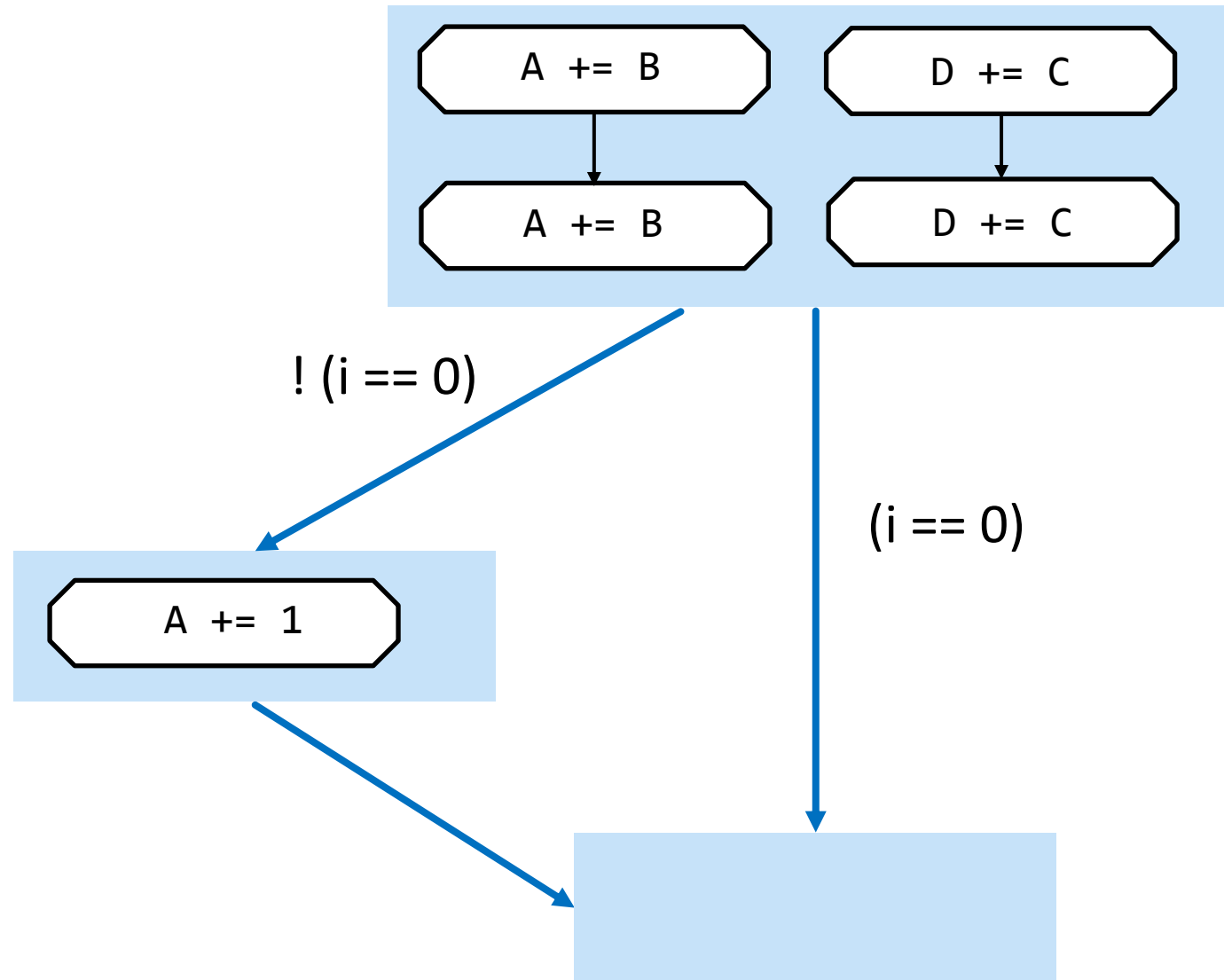


# Irreducible control flow and optimization



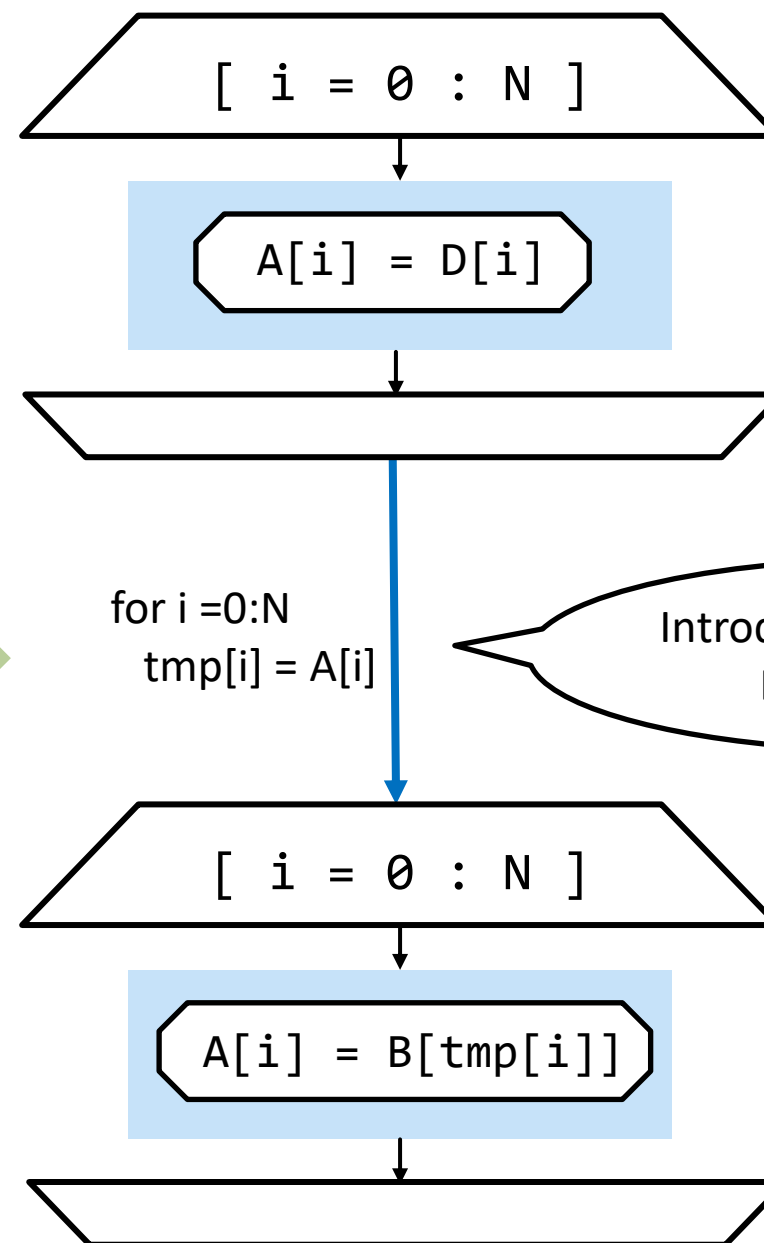
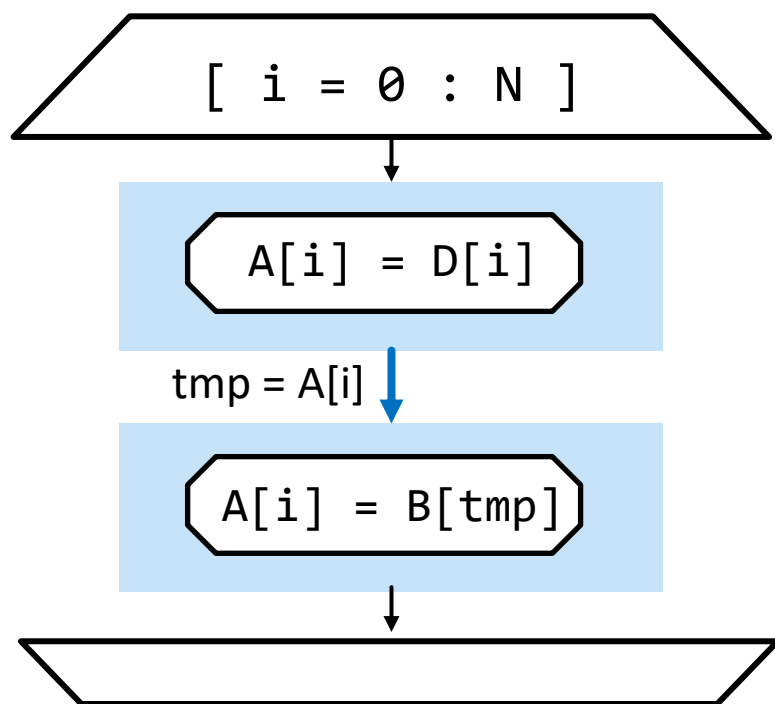
# The graph from before:

A+=B  
A+=B  
**If(! (i==0))**  
  A+=1  
D+=C  
D+=C



# Atomizing control flow

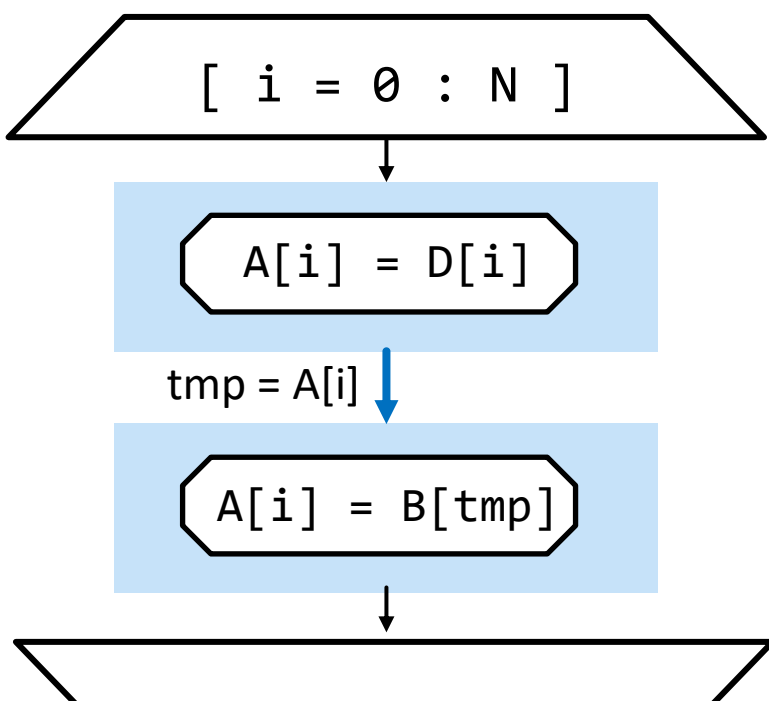
- If loops – including nested
- For loops – including nested
- Maps – including nested
  - Arrays of symbols questions – potential advantage to schedule tree analysis pass



for i = 0:N  
 tmp[i] = A[i]

Introduce symbol arrays?  
 Do not fission?

# Map fission in schedule tree



- **map i in [0:N]:**
  - A(i)=tasklet(D(i))
  - tmp=assign(A(i))
  - A(i)=tasklet(B(tmp))

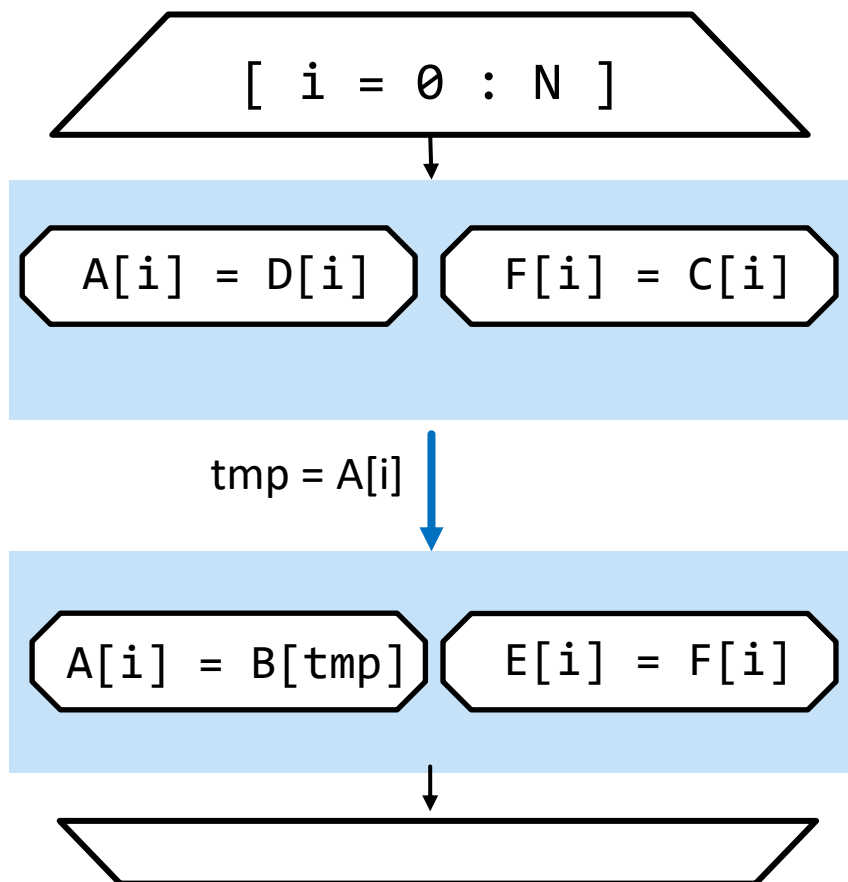
- **map i in [0:N]:**
  - A(i)=tasklet(D(i))
- **map i in [0:N]:**
  - tmp=assign(A(i))

- **map i in [0:N]:**
  - A(i)=tasklet(B(tmp))

Same issue! – But it's really a temporary view, with relaxed conditions!

So maybe we can just let the assignment as is, and add a property that we throw an error if it's not merged back at the end.

# Map fission in schedule tree – Why?



- **map i in [0:N]:**
  - A(i)=tasklet(D(i))
  - F(i)=tasklet(C(i))
  - tmp=assign(A(i))
  - A(i)=tasklet(B(tmp))
  - E(i)=tasklet(F(i))

- **map i in [0:N]:**
  - A(i)=tasklet(D(i))
- **map i in [0:N]:**
  - F(i)=tasklet(C(i))
- **map i in [0:N]:**
  - tmp=assign(A(i))
- **map i in [0:N]:**
  - A(i)=tasklet(B(tmp))
- **map i in [0:N]:**
  - E(i)=tasklet(F(i))

- **map i in [0:N]:**
  - A(i)=tasklet(D(i))
  - tmp=assign(A(i))
  - A(i)=tasklet(B(tmp))
- **map i in [0:N]:**
  - F(i)=tasklet(C(i))
  - E(i)=tasklet(F(i))