Control Flow Graphs

what's a control flow graph?

- A *directed graph* G = (V, E) where:
  - V (vertices) are the basic blocks in the program
  - E (edges) are *control flow edges* between basic blocks
- A **control flow edge** shows that execution *may* proceed along that edge
  - It is possible (though not always guaranteed) that a program's execution can go from the source of the edge directly to the target of the edge

```
  ADD t7, t1, t2
Lab1:
  ADD t9, t1, t3
  SUB t2, t7, t9
  BNE t2, t1 Lab1
  ADD t2, t4, t7
```

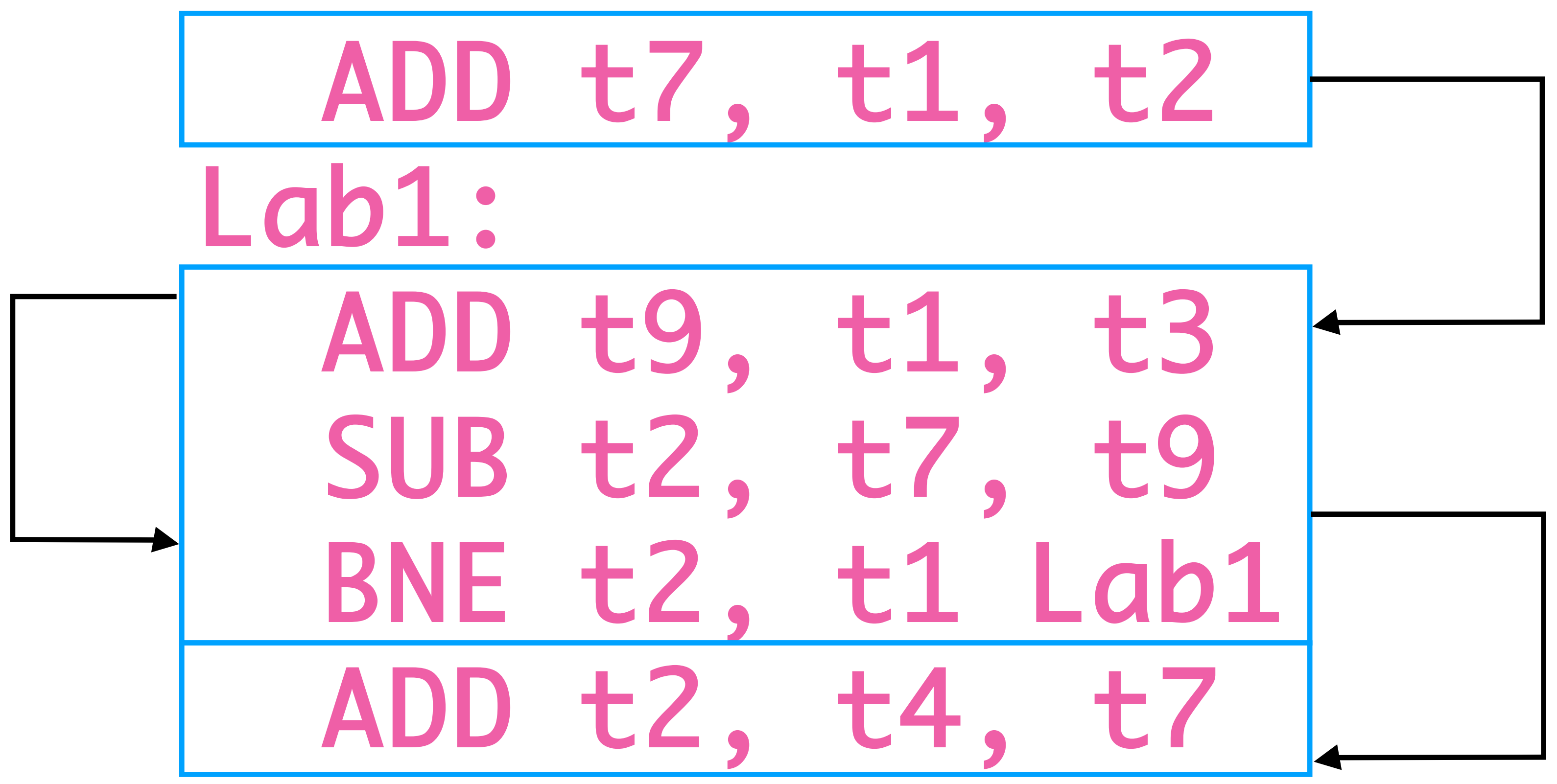
what's a control flow graph?

- A *directed graph* G = (V, E) where:
  - V (vertices) are the basic blocks in the program
  - E (edges) are *control flow edges* between basic blocks
- A **control flow edge** shows that execution *may* proceed along that edge
  - It is possible (though not always guaranteed) that a program's execution can go from the source of the edge directly to the target of the edge

```
  ADD t7, t1, t2
Lab1:
  ADD t9, t1, t3
  SUB t2, t7, t9
  BNE t2, t1 Lab1
  ADD t2, t4, t7
```

what's a control flow graph?

- A *directed graph* G = (V, E) where:
  - V (vertices) are the basic blocks in the program
  - E (edges) are *control flow edges* between basic blocks
- A **control flow edge** shows that execution *may* proceed along that edge
  - It is possible (though not always guaranteed) that a program's execution can go from the source of the edge directly to the target of the edge



adding control flow edges

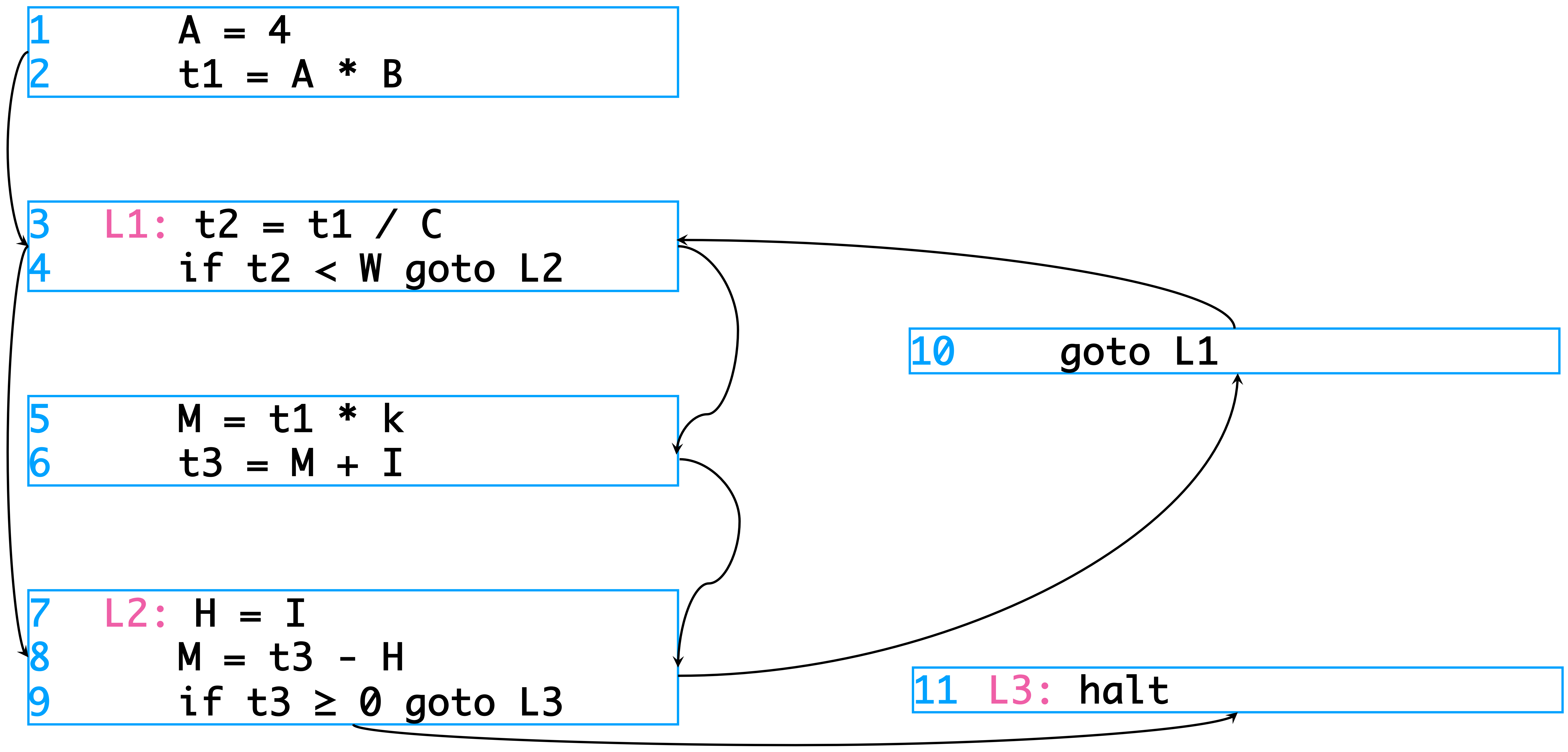
- There is a directed edge from B_1 to B_2 if
  - There is a branch from the last statement of B_1 to the first statement (leader) of B_2
  - B_2 immediately follows B_1 in program order and B_1 *does not* end with an unconditional branch
- Input: block, a sequence of basic blocks
- Output: The CFG

```
for i = 1 to |block|
  x = last statement of block(i)
  if stat(x) is a branch, then
    for each explicit target y of stat(x)
      create edge from block i to block y
    end for
  if stat(x) is not unconditional branch, then
    create edge from block i to block i+1
end for
```

example

```
    A = 4
    t1 = A * B
L1: t2 = t1 / C
    if t2 < W goto L2
    M = t1 * k
    t3 = M + I
L2: H = I
    M = t3 - H
    if t3 ≥ 0 goto L3
    goto L1
L3: halt
```

example


1 A = 4
2 t1 = A * B
3 L1: t2 = t1 / C
4 if t2 < W goto L2
5 M = t1 * k
6 t3 = M + I
7 L2: H = I
8 M = t3 - H
9 if t3 ≥ 0 goto L3
10 goto L1
11 L3: halt