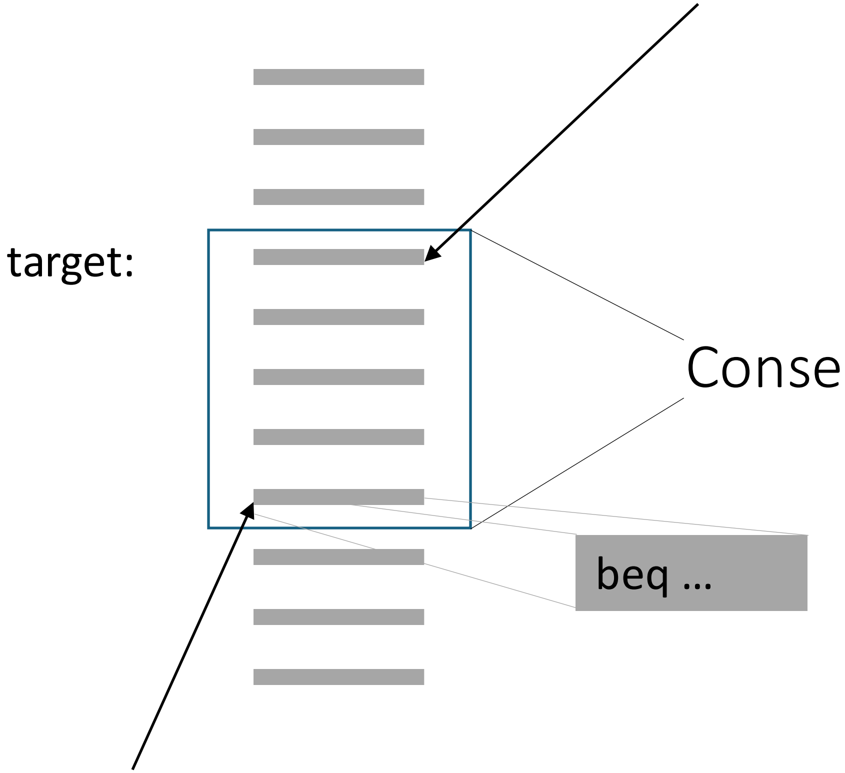
Basic Blocks

Basic Blocks

Only the first statement can be the target of a branch



Consecutive statements

If one statement in the basic block is executed, then all of them must be executed.

Only the last statement can be a branch

(AhoSethiUllman, pp. 529)

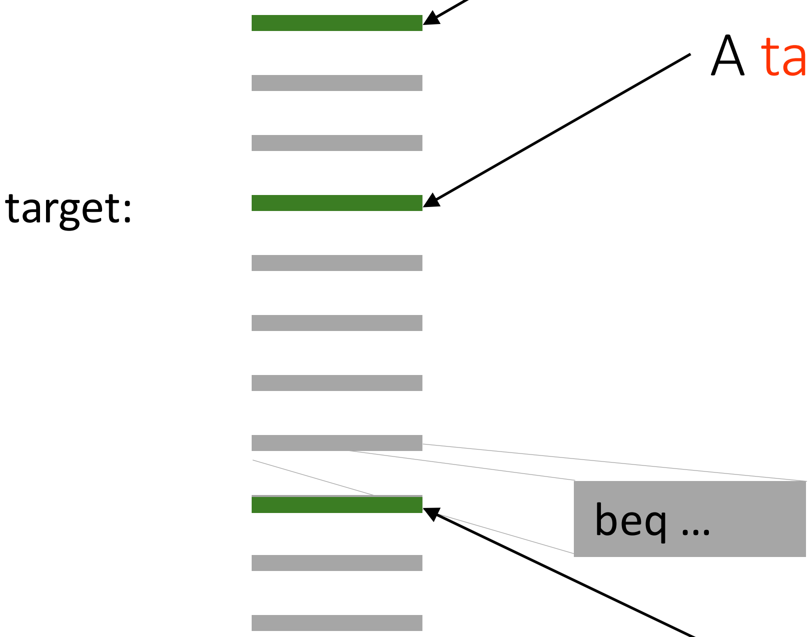
Basic Block Partitioning Algorithm

1. Identify leader statements (i.e. the first statements of basic blocks) by using the following rules:

(AhoSethiUllman, pp. 529)

The first statement in the program is a leader

A target of a branch is a leader



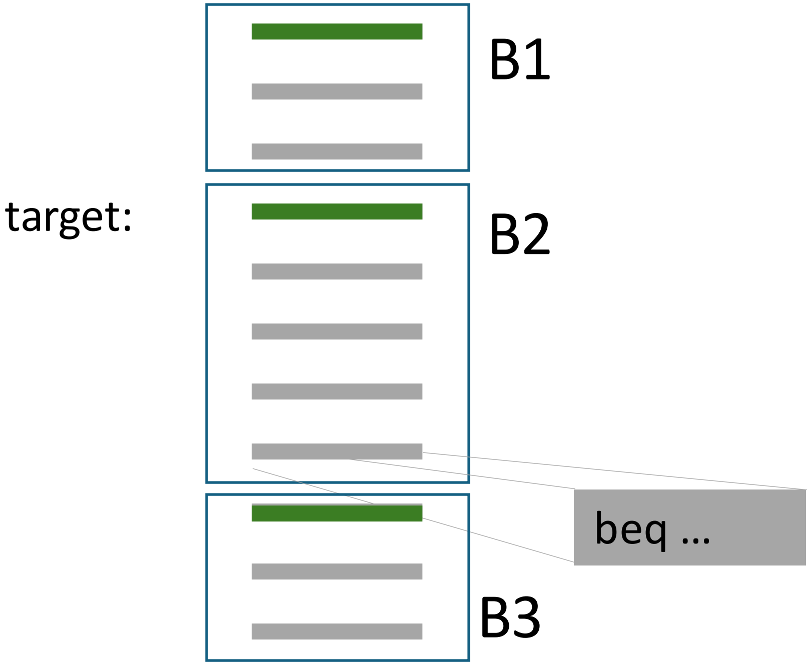
A statement that immediately follows a branch
or return statement is a leader

(AhoSethiUllman, pp. 529)

Basic Block
Partitioning Algorithm

2. A basic block is formed by a leader followed by all statements up to, but not including, the next leader or up to the end of the program.

(AhoSethiUllman, pp. 529)



(AhoSethiUllman, pp. 529)

Example: Finding Leaders

Inner product of two vectors:

```
begin
  prod = 0;
  i = 1;
  do begin
    prod = prod + a[i]*b[i];
    i = i+1;
    end
  while i <= 20
end
```

Source code

First statement

Branch target

```
(1)  prod = 0
(2)  i = 1
(3)  t1 = 4*i
(4)  t2 = a[t1]
(5)  t3 = 4*i
(6)  t4 = b[t3]
(7)  t5 = t2*t4
(8)  t6 = prod+t5
(9)  prod = t6
(10) t7 = i+1
(11) i = t7
(12) if i <= 20 goto (3)
(13)  ...
```

Follows a branch

Three-address code

(AhoSethiUllman, pp. 529)

Forming the Basic Blocks

2. **Basic block**: a leader plus all statements up to but not including the next leader or up to the end of the program.

Example: **Forming the Basic Blocks**

Basic Blocks:

B1

```
(1)  prod = 0
(2)  i = 1
```

B2

```
(3)  t1 = 4*i
(4)  t2 = a[t1]
(5)  t3 = 4*i
(6)  t4 = b[t3]
(7)  t5 = t2*t4
(8)  t6 = prod+t5
(9)  prod = t6
(10) t7 = i+1
(11) i = t7
(12) if i <= 20 goto (3)
```

B3

```
(13)  ...
```

Control Flow Graph (CFG)

- A directed multigraph where:
  - the nodes are basic blocks; and
  - the edges represent flow of control:
    - a branch
    - fall-through execution

Start node: the basic block whose leader is the first statement.

A CFG has no information about data flow.

An edge in the CFG means that the program may take that path.

Control Flow Graph (CFG)

There is a directed edge from basic block B1 to basic block B2 in the CFG if:

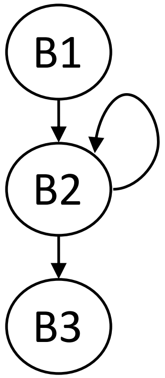
Branch Rule: There is a branch from the last statement of B1 to the first statement of B2, or

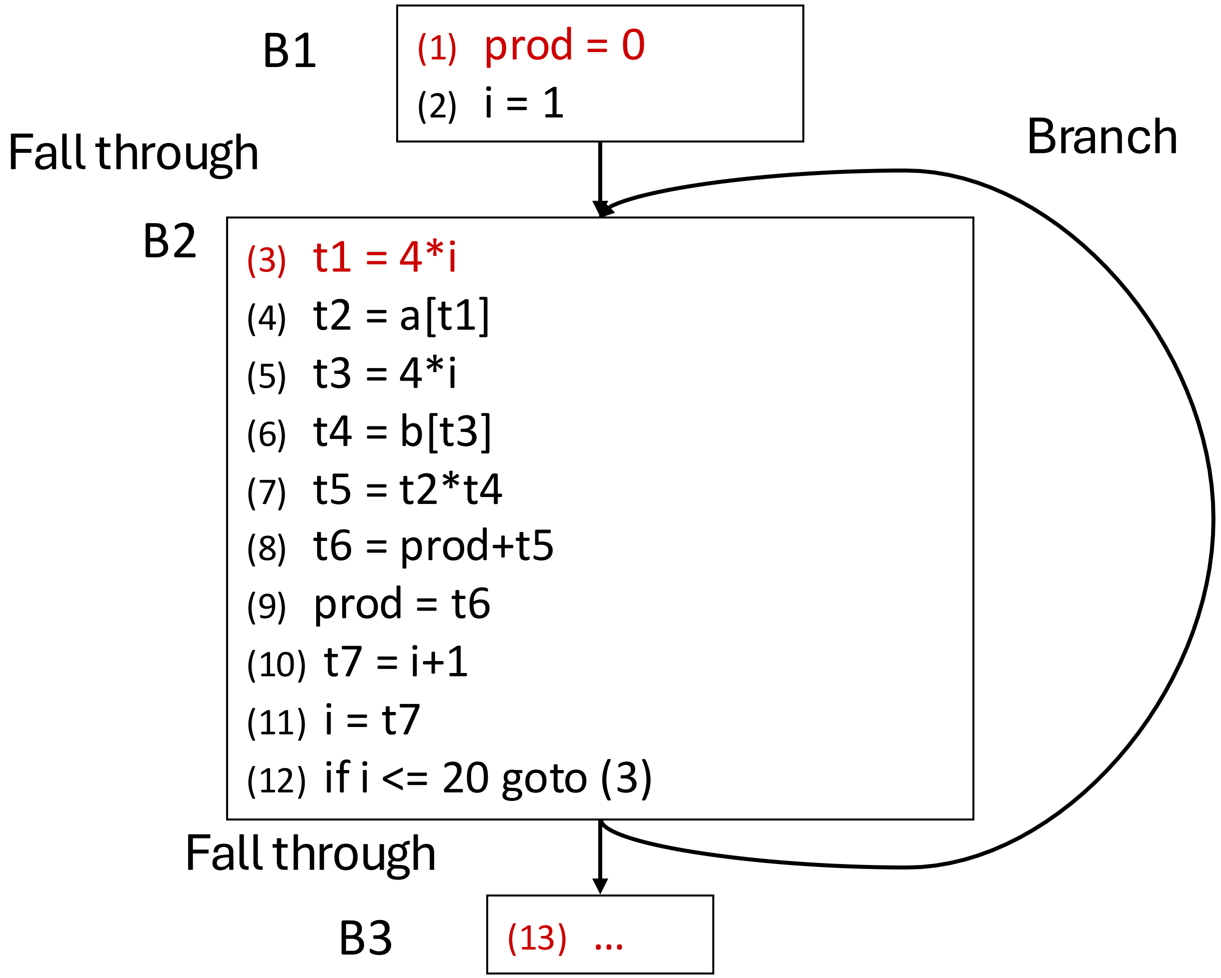
Fall-through Rule: Control flow can fall through from B1 to B2 because:

(i) B2 immediately follows B1, and

(ii) B1 does not end with an unconditional branch

Example : Control Flow Graph Formation





B1

```
(1) prod = 0
(2) i = 1
```

Fall through

Branch

B2

```
(3)  t1 = 4*i
(4)  t2 = a[t1]
(5)  t3 = 4*i
(6)  t4 = b[t3]
(7)  t5 = t2*t4
(8)  t6 = prod+t5
(9)  prod = t6
(10) t7 = i+1
(11) i = t7
(12) if i <= 20 goto (3)
```

Fall through

B3

```
(13) ...
```