Induction Variables

Region-Based Analysis

Meet and Closure of Transfer Functions

Affine Functions of Reference Variables

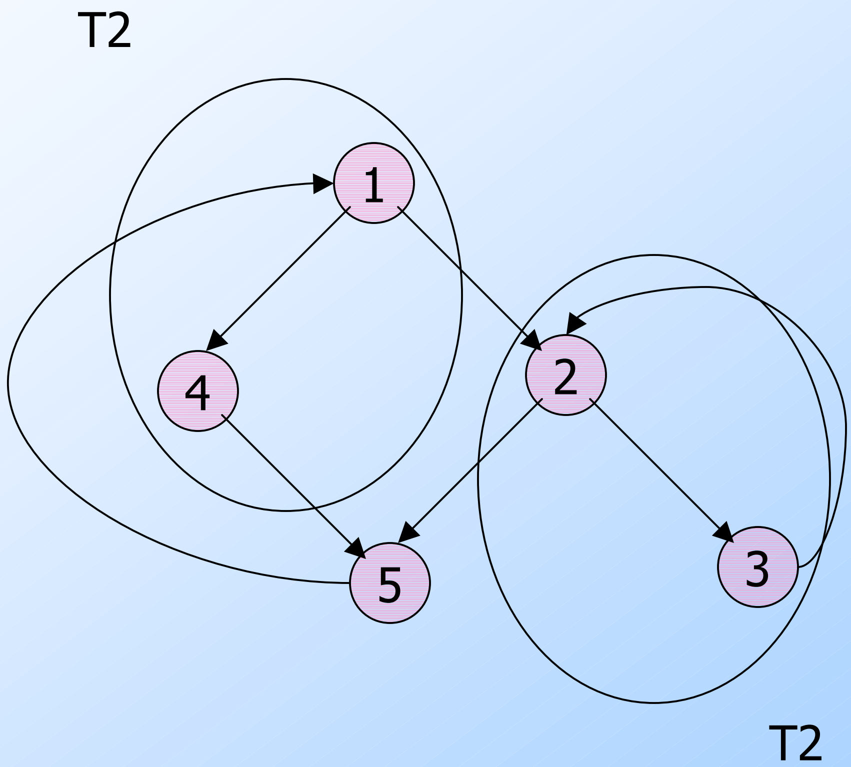
Regions

- A set of nodes N and edges E is a *region* if:
  1. There is a header h in N that dominates all nodes in N.
  2. If n $\neq$ h is in N, then all predecessors of n are also in N.
  3. E consists of all edges between nodes in N, possibly excluding nodes that enter h.

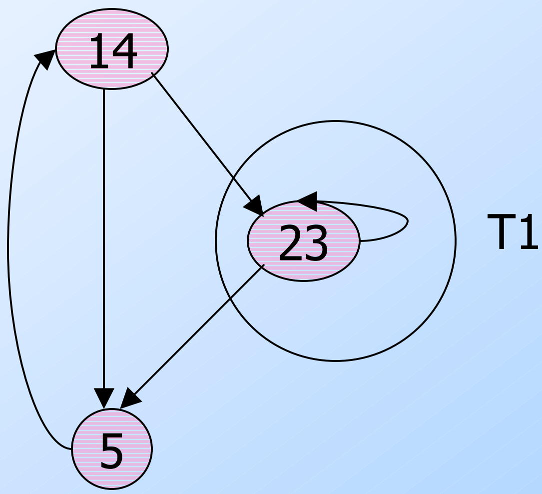
T1-T2 Reduction

- For reducible flow graphs, we can "reduce" the graph by two region-creating transformations.
  - T1: Remove a loop from a node.
  - T2: combine two nodes n and m such that m's only predecessor is n, and m is not the entry.

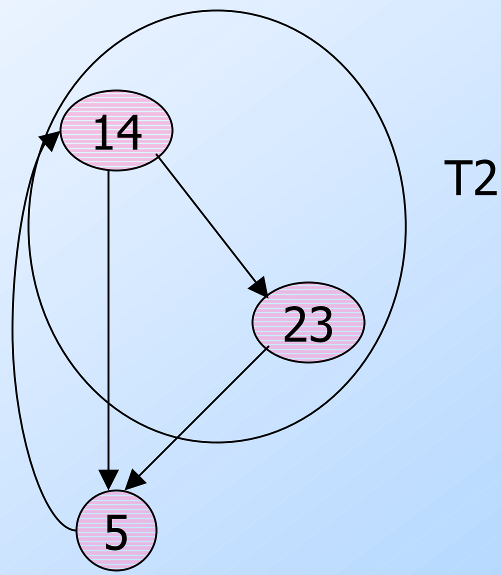
Example: T1-T2 Reduction



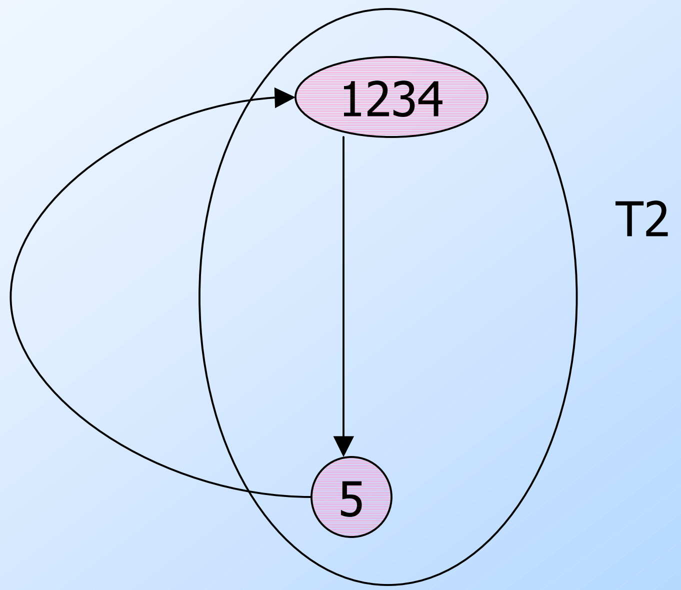
Example: T1-T2 Reduction



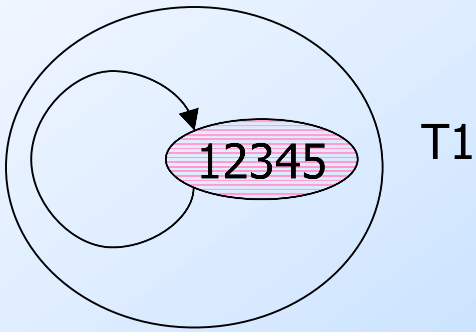
Example: T1-T2 Reduction



Example: T1-T2 Reduction



Example: T1-T2 Reduction



Example: T1-T2 Reduction

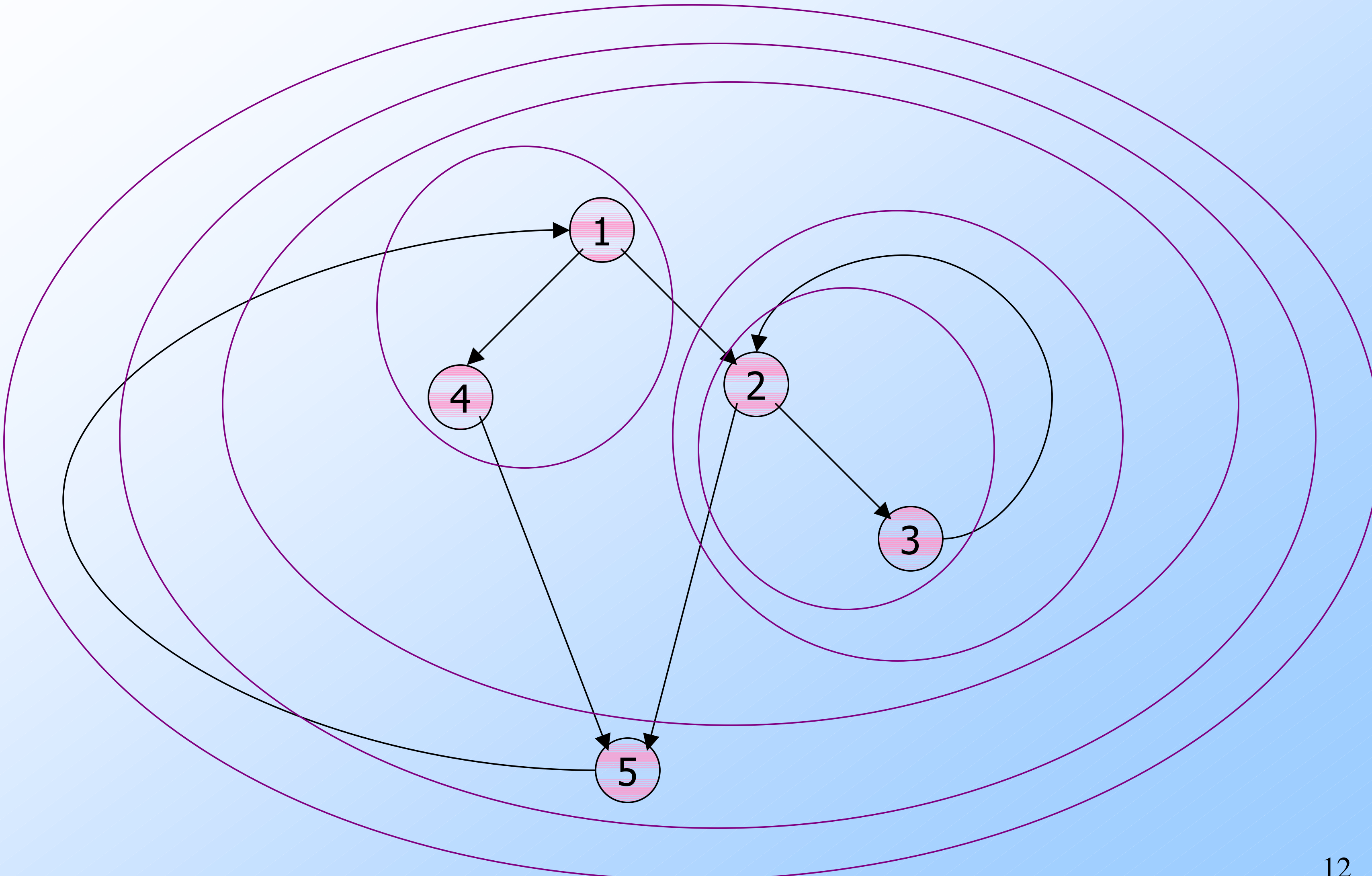
Regions Constructed During T1-T2 Reduction

- Each node represents a set of nodes and edges of the original flow graph.
- Each edge represents one or more edges of the original flow graph.

Regions Constructed During T1-T2 Reduction --- (2)

- T2: Take the union of the two sets of nodes and edges. Then add edges represented by the edge between the two combined nodes.
- T1: Add edges represented by the loop edge removed.

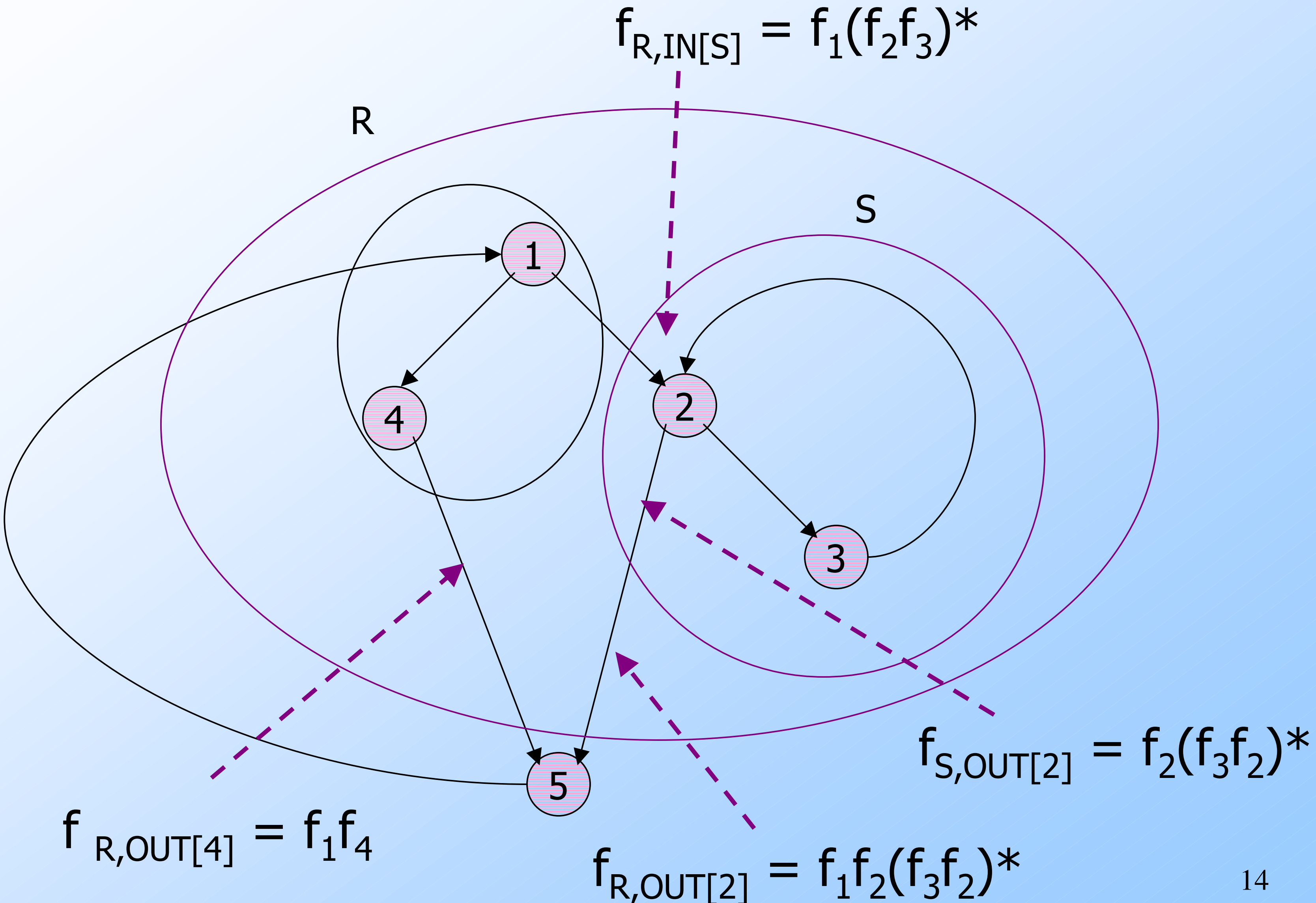
Example: T1-T2 Reduction



Region-Based Analysis --- (1)

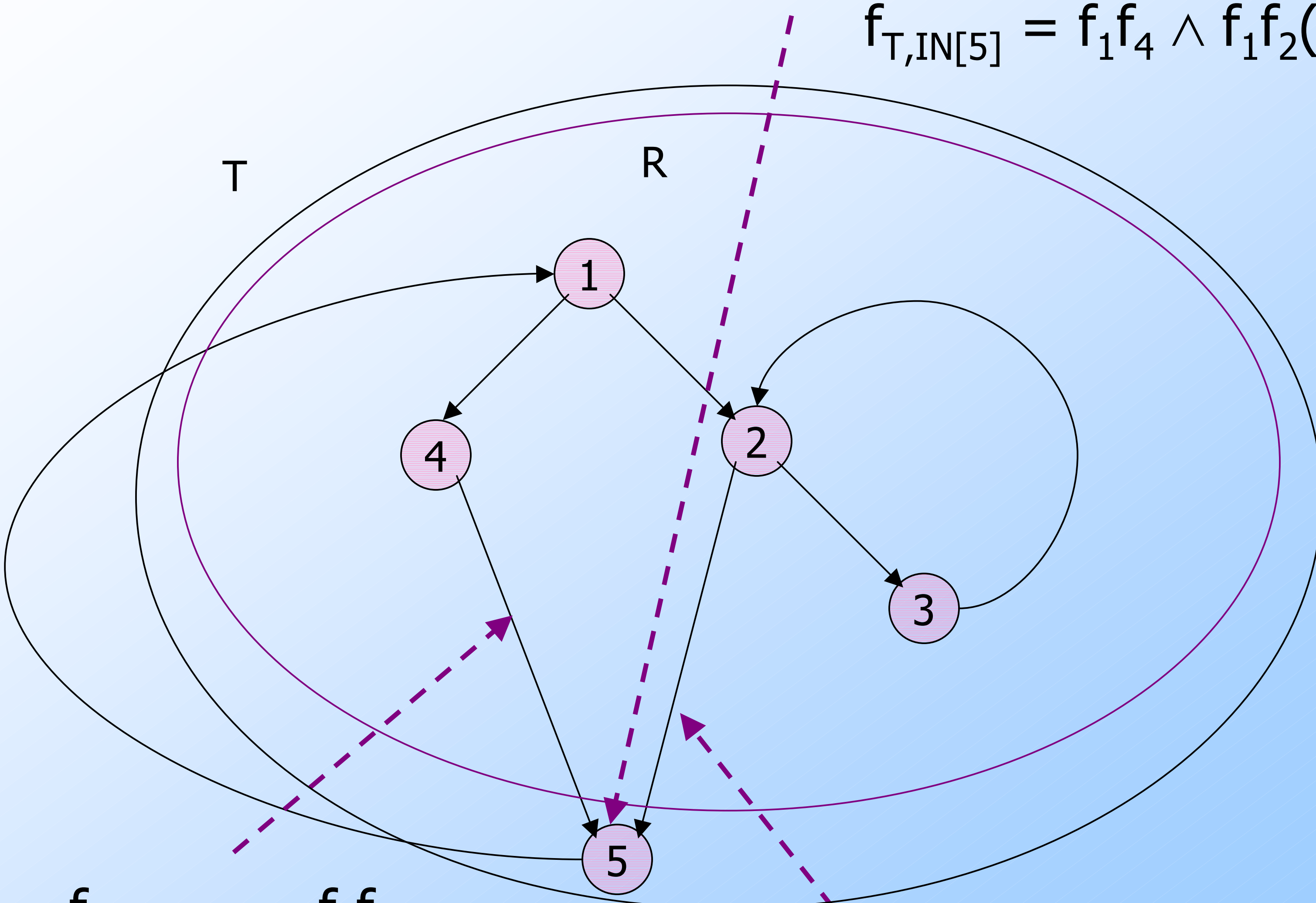
- First, compute for each region, smallest-to-largest, some transfer functions:
  1. For region R with subregion S: $f_{R,IN[S]}$ = meet over paths from the header of R to the header of S, staying within R.
  2. For each region R with *exit block* (= has a successor outside R) B, $f_{R,OUT[B]}$ = meet over paths within R from header of R to end of B.

Example: Transfer Functions



Example: Meet

$f_{T,IN[5]} = f_1 f_4 \wedge f_1 f_2 (f_3 f_2)^*$



$f_{R,OUT[4]} = f_1 f_4$

$f_{R,OUT[2]} = f_1 f_2 (f_3 f_2)^*$

Additional Assumptions

- We can take the meet of transfer functions.
  - $[f \wedge g](x) = f(x) \wedge g(x)$.
- There is a *closure* f^* for each transfer function f.
  - $f^* = f^0 \wedge f^1 \wedge f^2 \wedge \ldots$
    - Note f^0 = identity, $f^1 = f$.

Example: RD's

- Meet of transfer functions $f(x) = (x - K) \cup G$ and $g(x) = (x - K') \cup G'$ corresponds to the paths of f and g in parallel.
- Thus, $[f \cup g](x) = (x - (K \cap K') \cup (G \cup G'))$.
- $f^*(x) = x \cup f(x) \cup f(f(x)) \cup \ldots =$ $x \cup (x - K) \cup G \cup (x - K \cup G) - K \cup G \ldots =$ $x \cup G$.

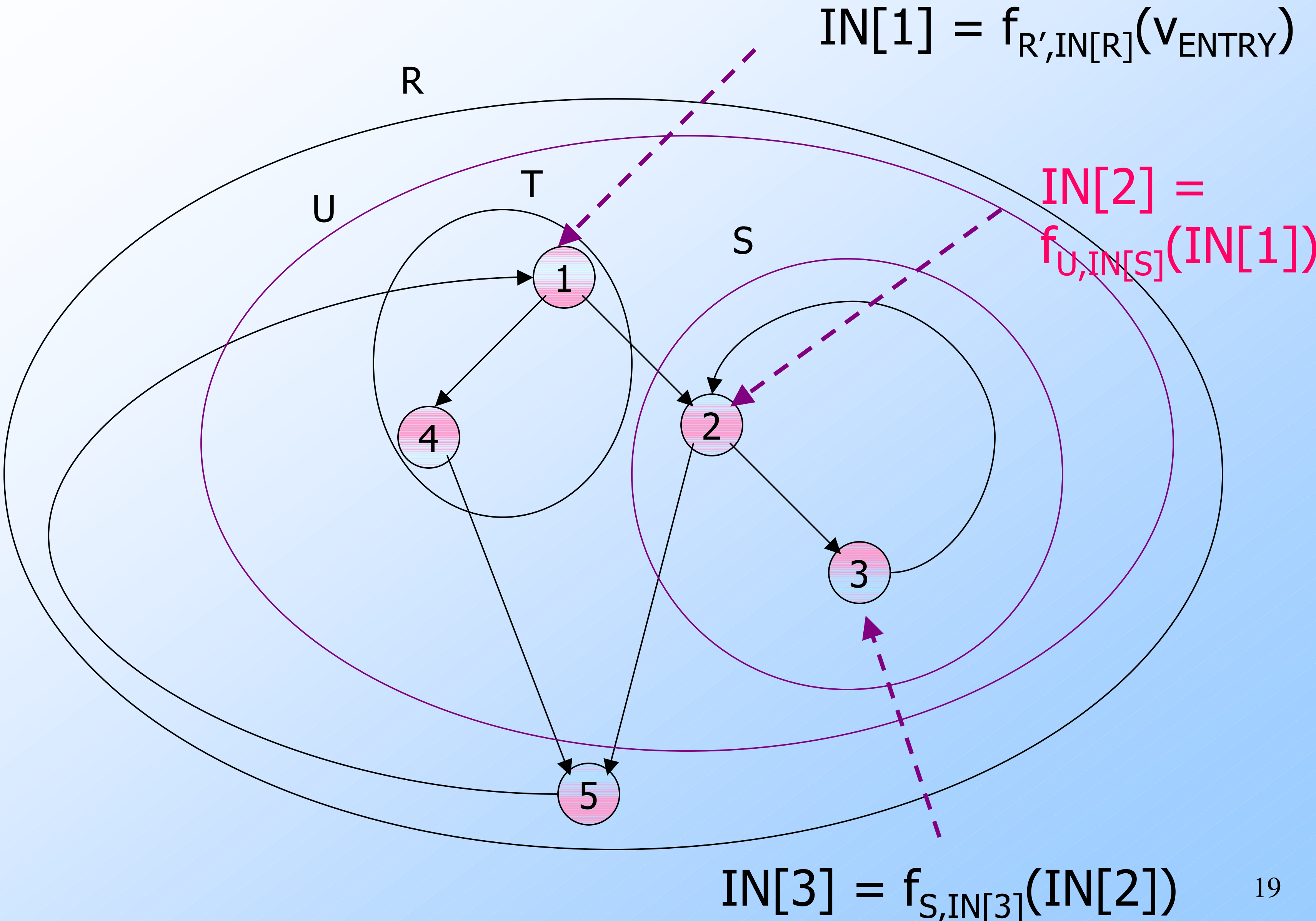
f(x) f(f(x))

Region-Based Analysis --- (2)

- Finally, having computed $f_{R',IN[R]}$
  - R = entire flow graph; R′ = R plus dummy entry

  proceed largest-to-smallest, computing IN[h], for the header h of each subregion.
- Since every node is the header of some region, we have all the IN’s, from which we can compute the OUT’s.

Example: Top-Down Finale



New Topic: Symbolic Analysis

- Values V are mappings from variables to expressions of *reference variables*.
- Example: m(a) = 2i; m(b) = i+j.
  - i and j are the reference variables.

Application to Induction Variables

- Reference variables are *basic induction variables* = counts of the number of times around some loop.
- V consists of *affine mappings* = each variable is mapped to either:
  1. A linear function of the reference variables, or
  2. NAA = "Not an affine" mapping, or
  3. UNDEF = top element = "nothing known."

Example: Affine Mapping

$m(a) = 2i + 3j + 4$

$m(b) = NAA$

$m(c) = 4i + 1$

$m(d) = UNDEF$

Induction Variable Discovery

- Natural application of region-based analysis.
- Each loop is a region, and its induction variables can be discovered by a framework based on affine mappings.
- In region-based analysis, you don't even need the top element, UNDEF.

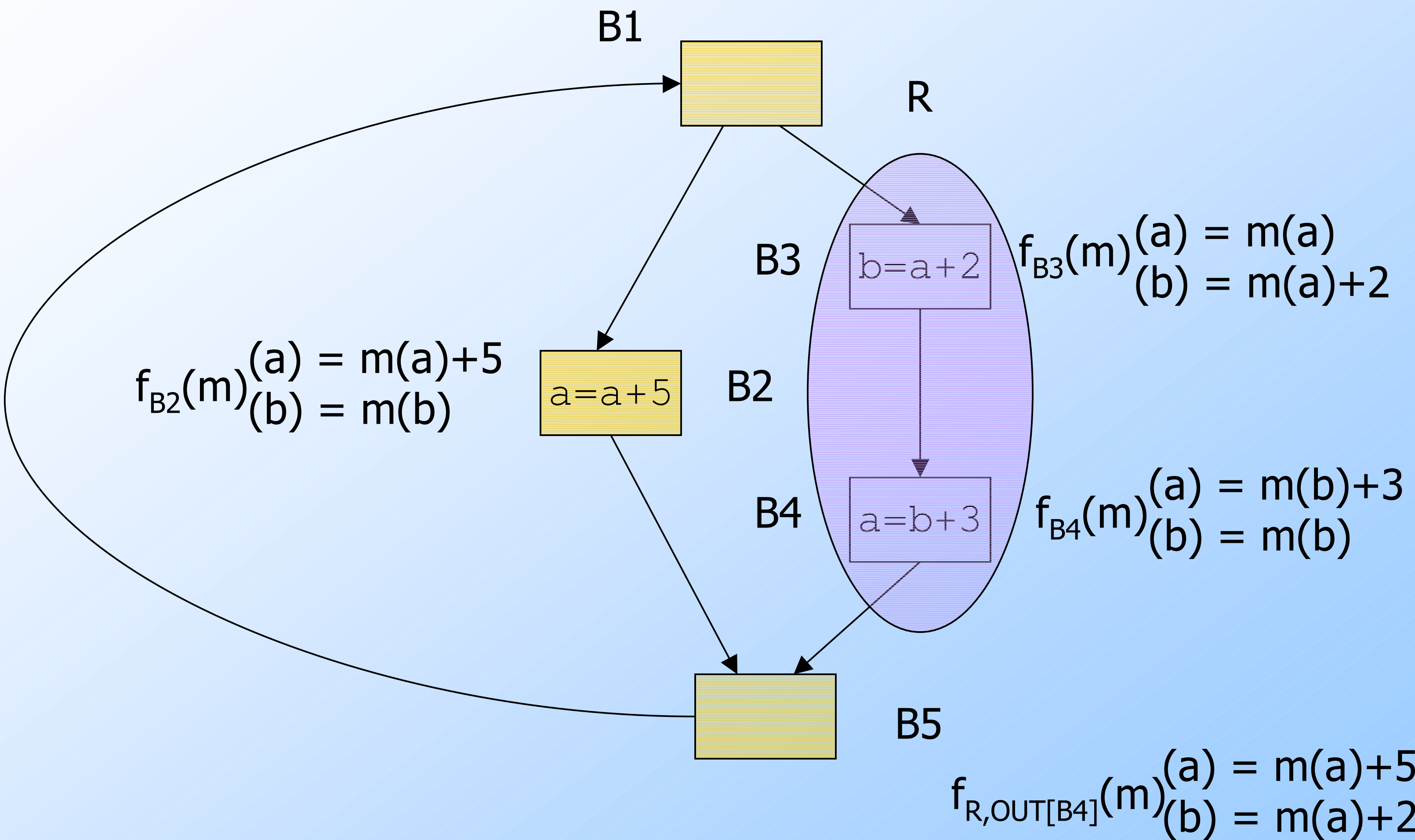
Example: A Transfer Function

- Let f be the transfer function associated with a block containing only `a = a+10`.
- Let $f(m) = m'$.
  - $m'(a) = m(a) + 10$.
  - $m'(x) = m(x)$ for all $x \neq a$.
- Book uses the notation $f(m)(a) = m(a)$, etc., to avoid having to name $f(m)$.

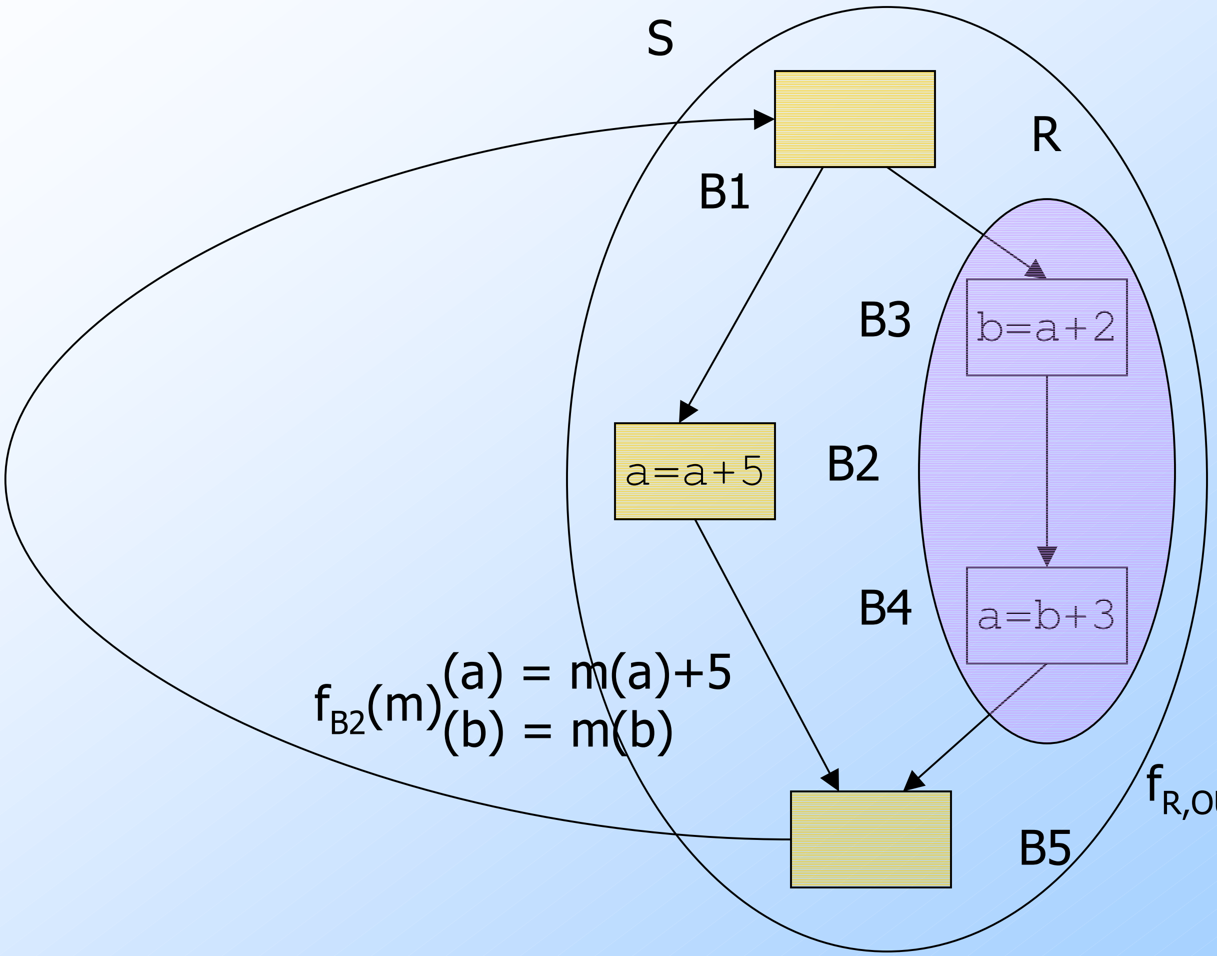
Example: Meet

- $(f \wedge g)(m)(x) =$
  - $f(m)(x)$ if $f(m)(x) = g(m)(x)$.
  - NAA otherwise.
- Note: above assumes no occurrences of UNDEF. How would you treat the case where $f(m)(x)$ or $g(m)(x)$, or both, are UNDEF?

Example: Some Analysis



Example: Meet



$$f_{B2}(m)\begin{matrix}(a) = m(a)+5 \\ (b) = m(b)\end{matrix}$$

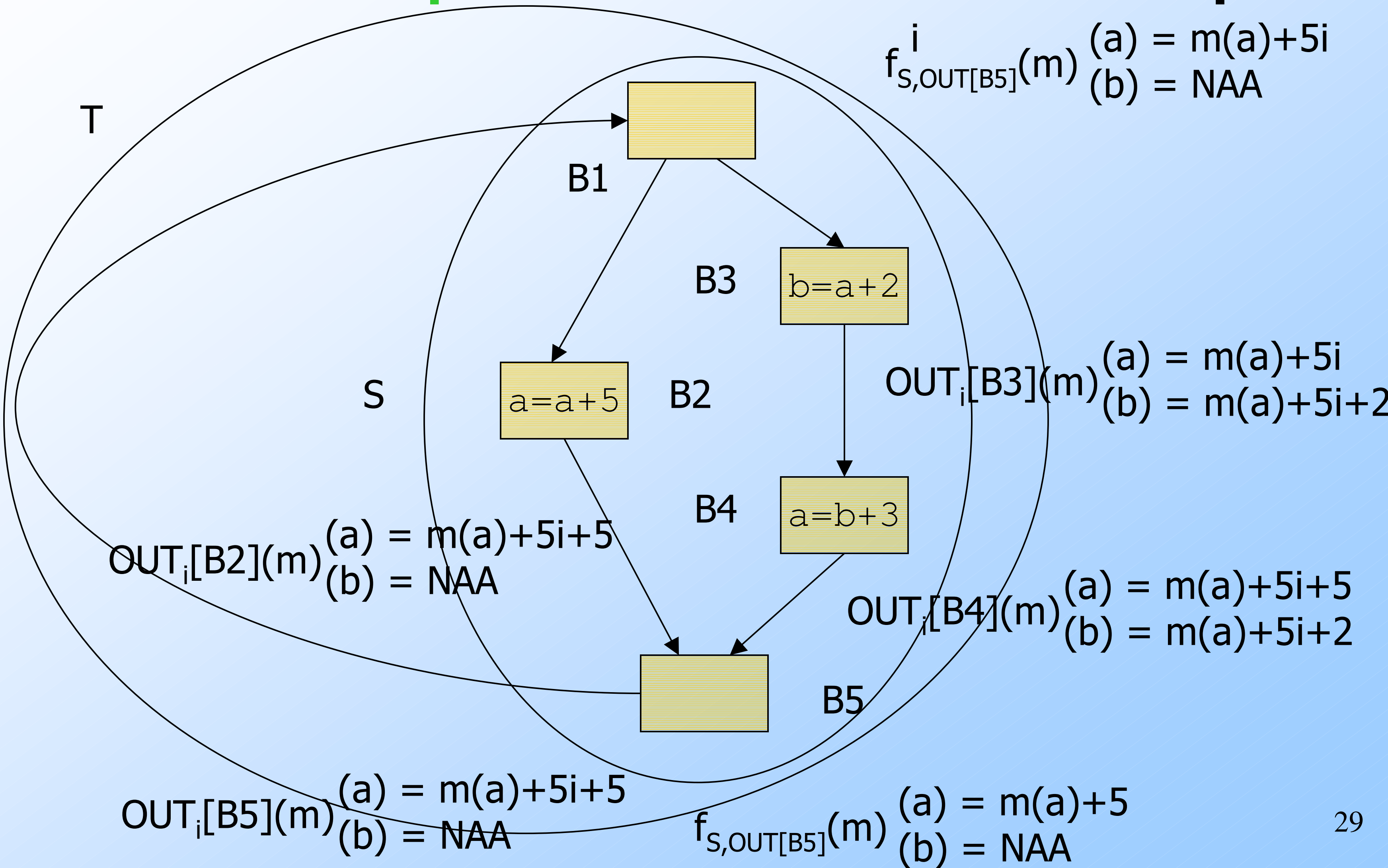
$$f_{R,OUT[B4]}(m)\begin{matrix}(a) = m(a)+5 \\ (b) = m(a)+2\end{matrix}$$

$$f_{S,OUT[B5]}(m)\begin{matrix}(a) = m(a)+5 \\ (b) = NAA\end{matrix}$$

Handling Loop Regions

- Treat the iteration count i as a basic induction variable.
- If $f(m)(x) = m(x)+c$, then $f^i(m)(x) = m(x)+ci$.
- Some other cases, e.g., where m(x) is an affine function of basic induction variables, in book.

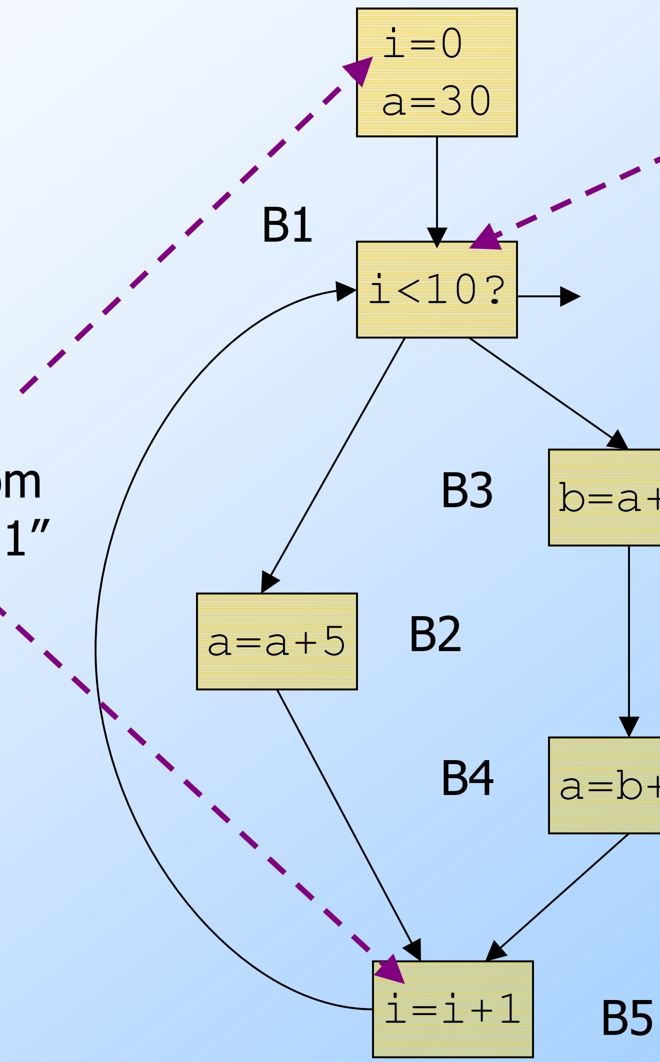
Example: The Entire Loop



Taking Advantage of Affine Expressions

- Replace the loop counter variable by one of the induction variables (variables that are mapped to an affine expression of the loop count at the point were the loop count is tested).

Example: Assume i is Loop Counter



Here, the value of a is m(a)+5i, where m(a) is the value of a before entering the loop, and i is the number of iterations of the loop so far.

If i is dead on exit from the loop, delete "i=i+1" and "i=0".

Since a=30 when i=0, replace "i<10?" by "a<80?".

Example: Revised Code