Partially Path-sensitive Maximum Fix Point Solution

by

Komal Pathade
(TCS Research, India)
and
Uday Khedker
(IIT Bombay, India)

Feb 24, 2018

Outline

- Background: Data Flow Analysis
- Maximum Fix Point (MFP) Solution in Data Flow Analysis
- Effect of Infeasible Control Flow Path on the MFP solution
- Partially Path Sensitive MFP
- Experimental Evaluation
- Future work

Data Flow Analysis

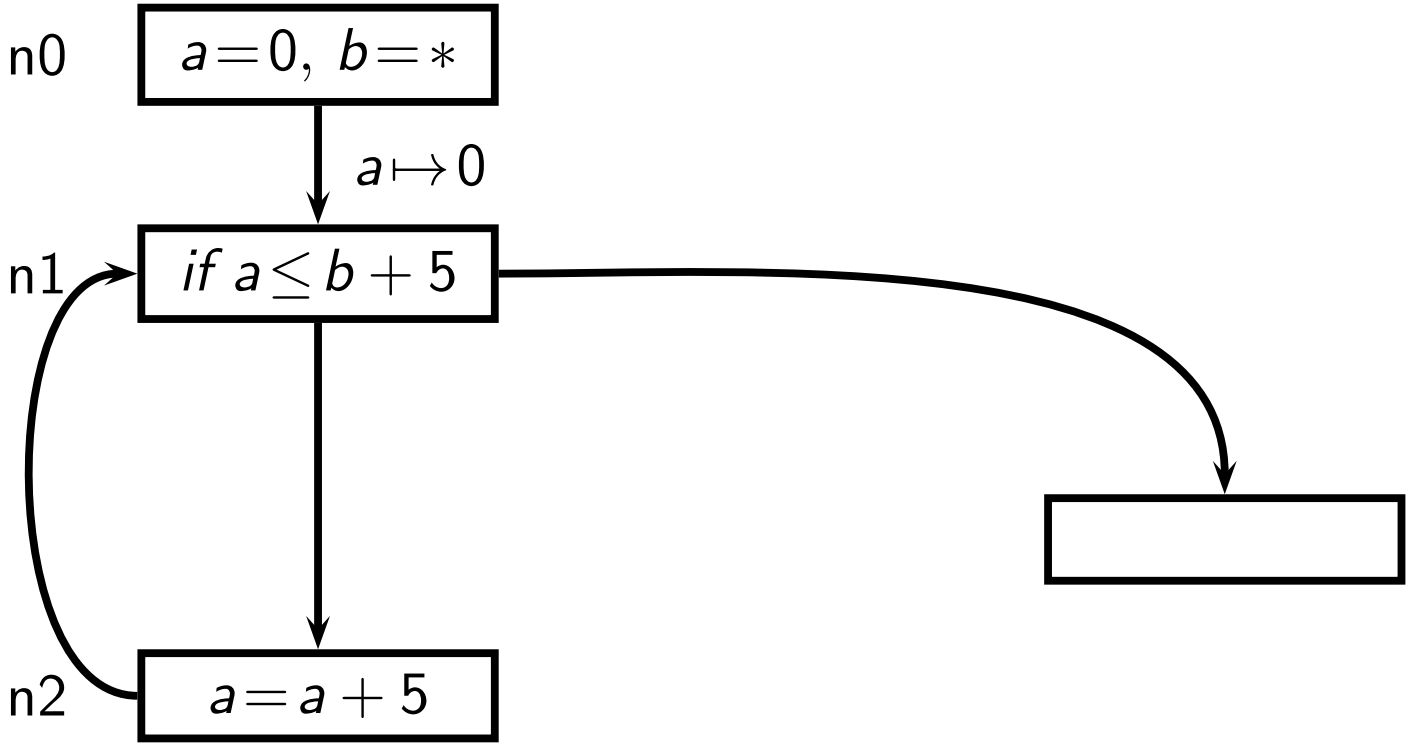
Captures code level information that can be later used for

- Program verification,
- Code understanding,
- Code optimization etc

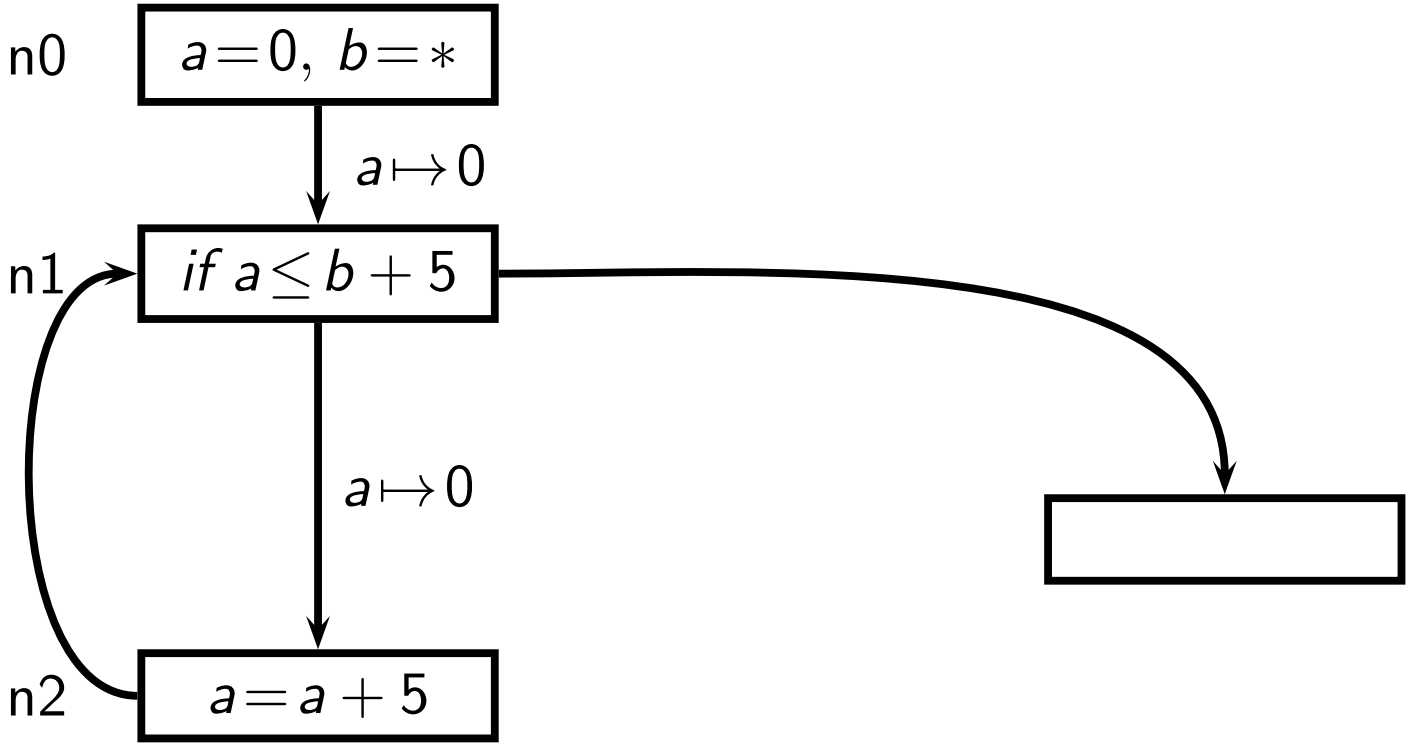
Solutions of Data Flow Analysis

Solution	Traverses Infeasible CFPs	Edge/Path Based Specification	Scalable	Precise
Meet Over Feasible Paths	No	Path	Least ⇓ Most	Most ⇓ Least
Meet Over all Paths	Yes	Path		
Maximum Fix Point	Yes	Edge		

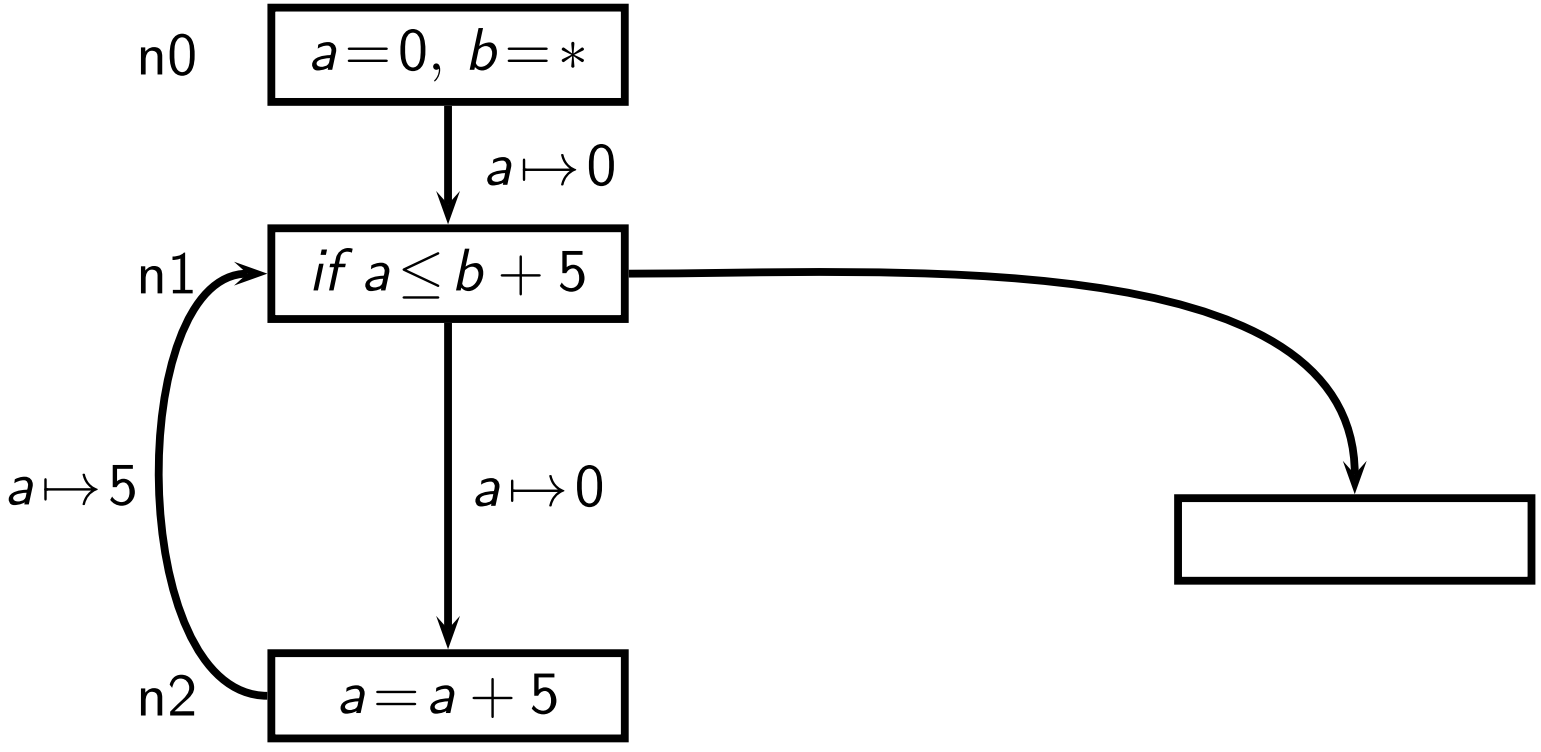
Example: The MFP Solution



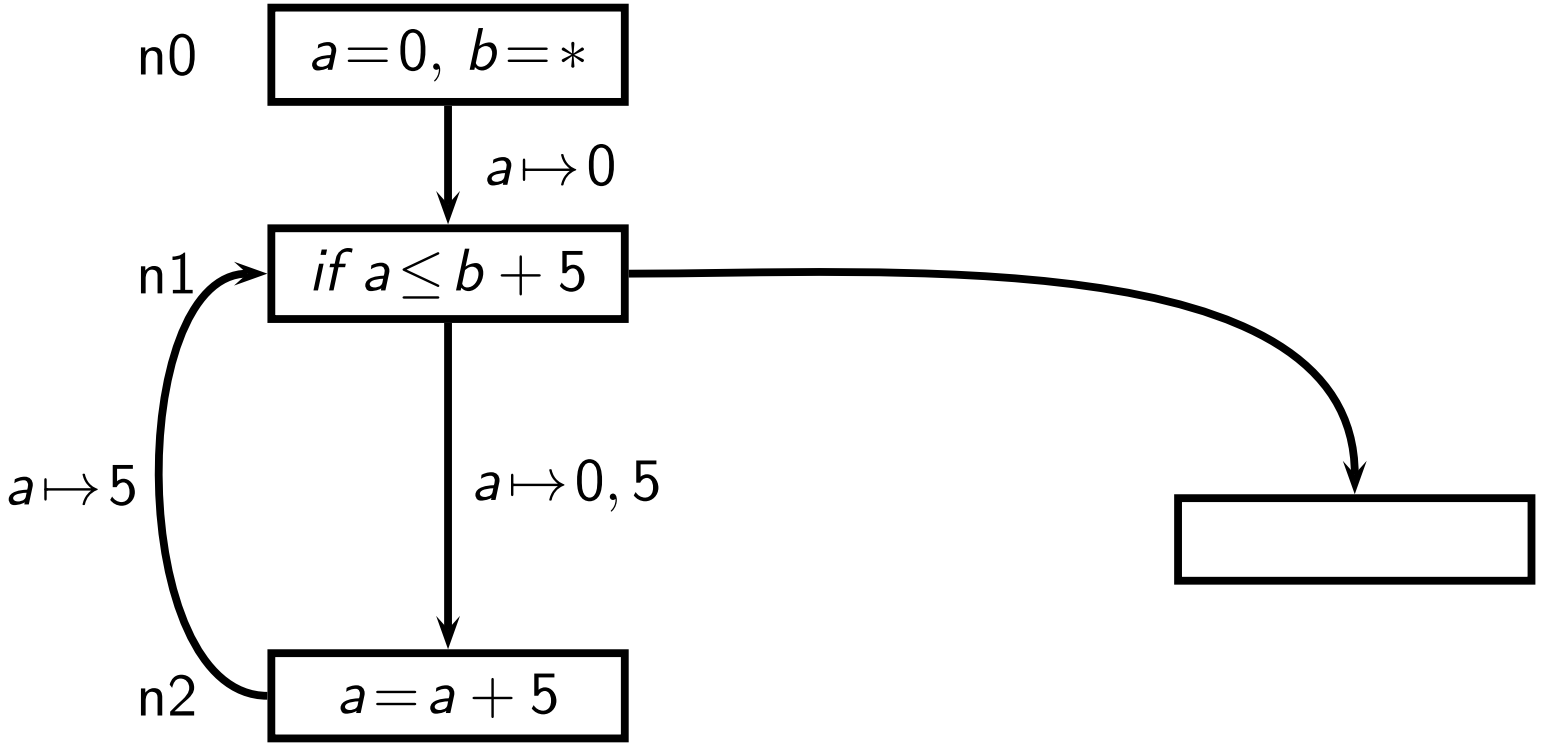
Example: The MFP Solution



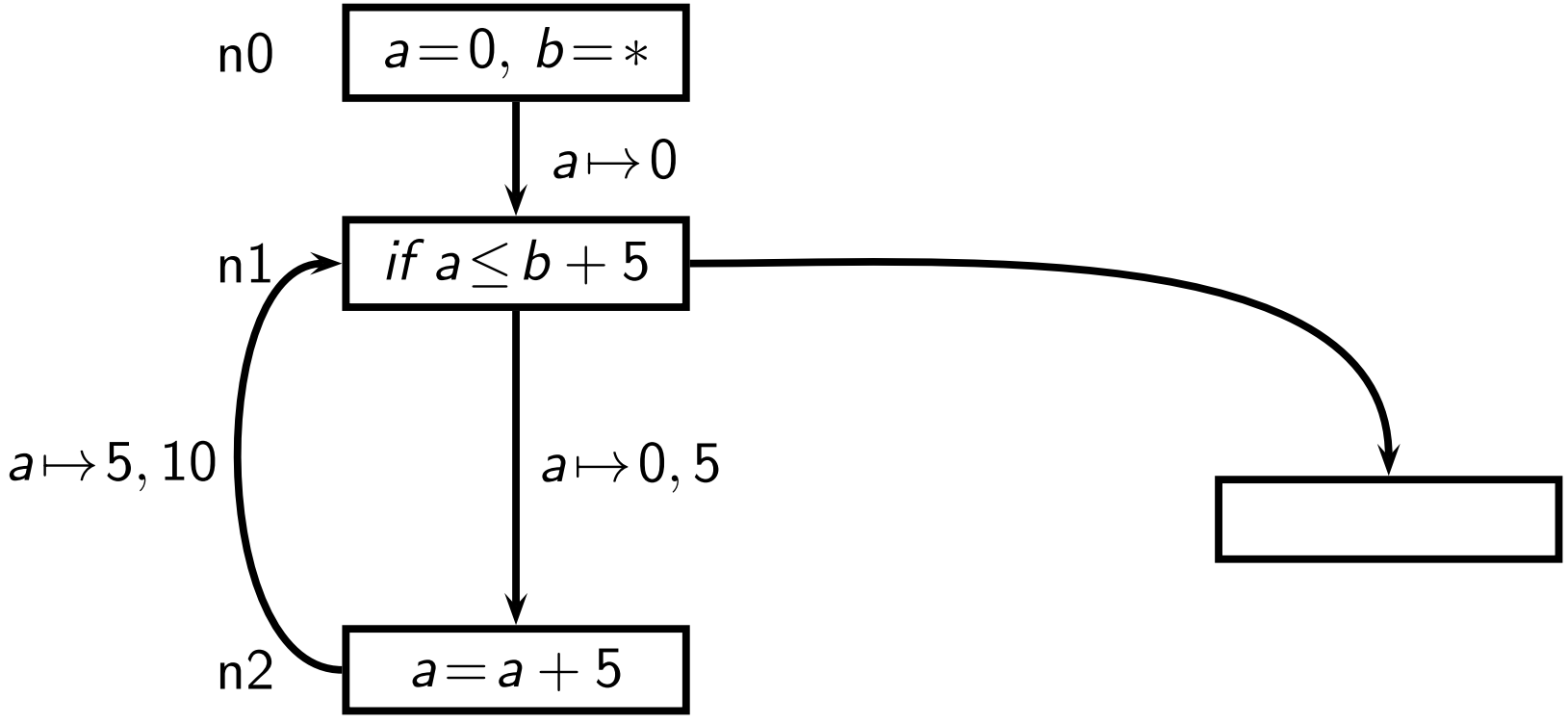
Example: The MFP Solution



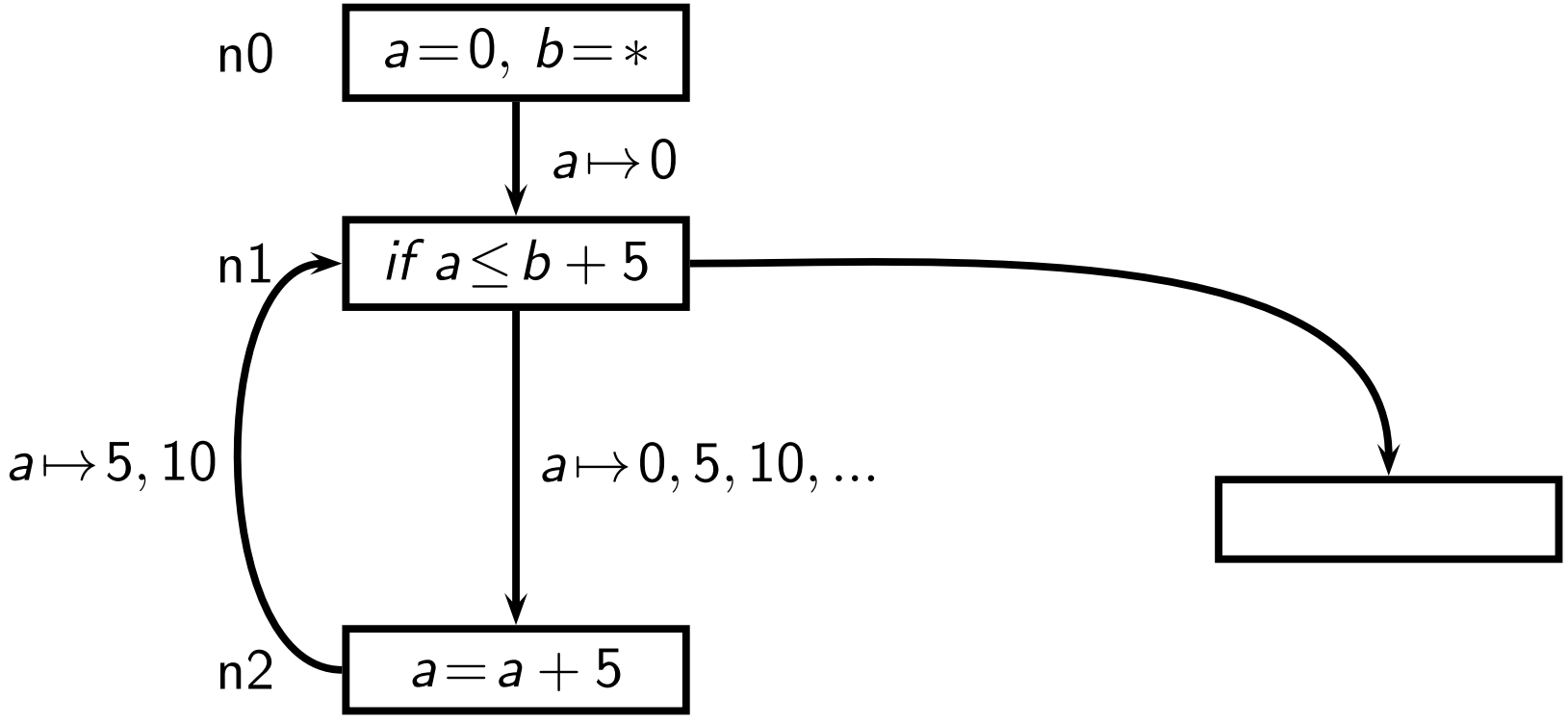
Example: The MFP Solution



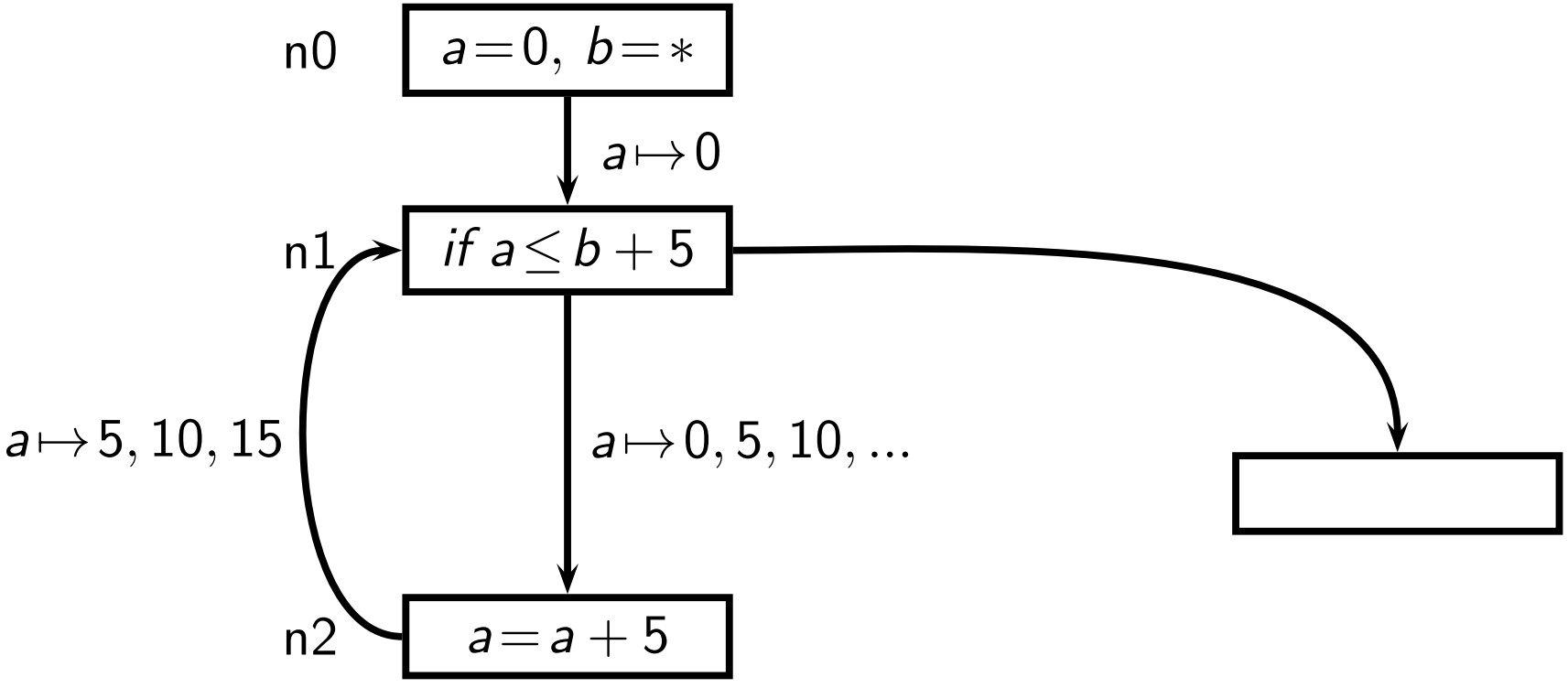
Example: The MFP Solution



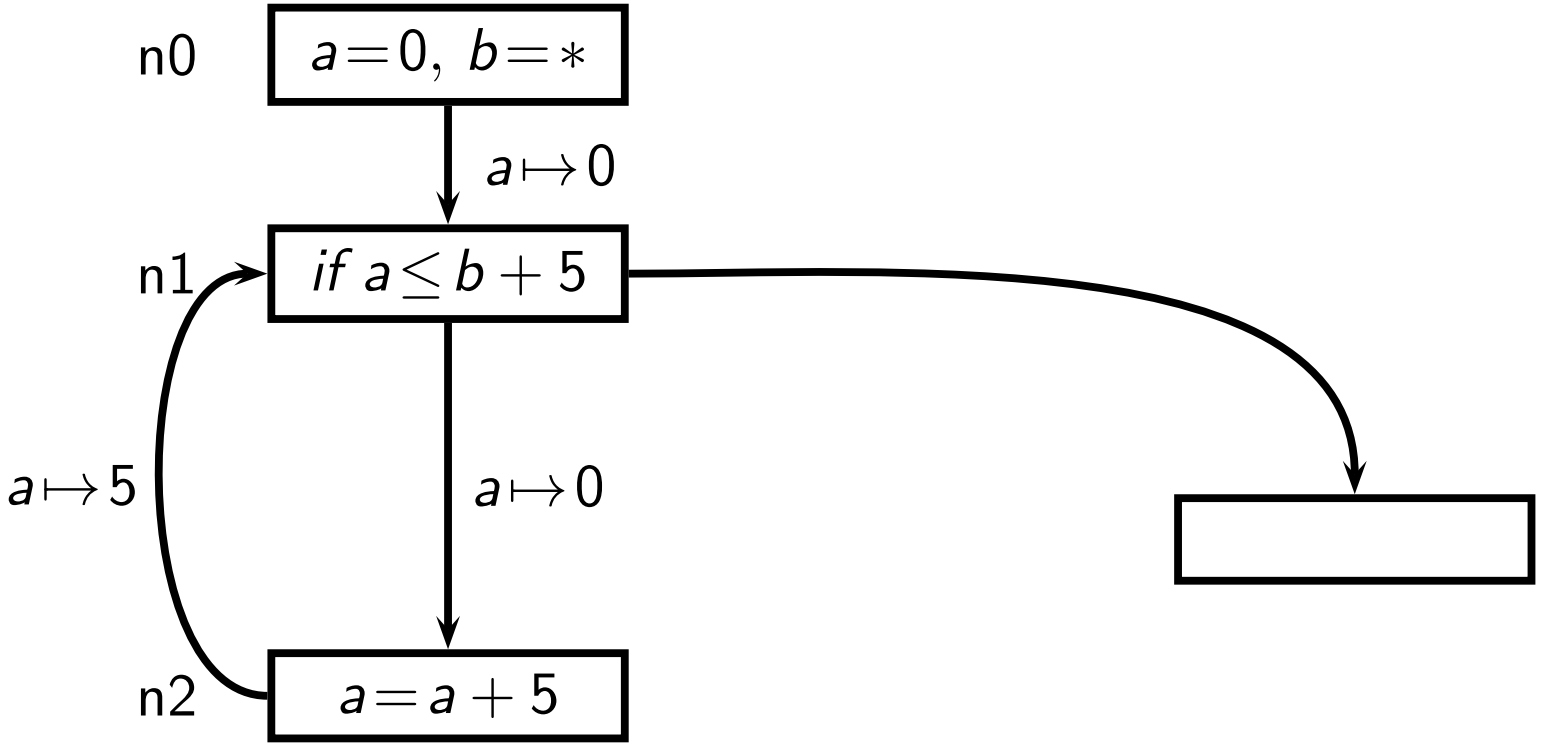
Example: The MFP Solution



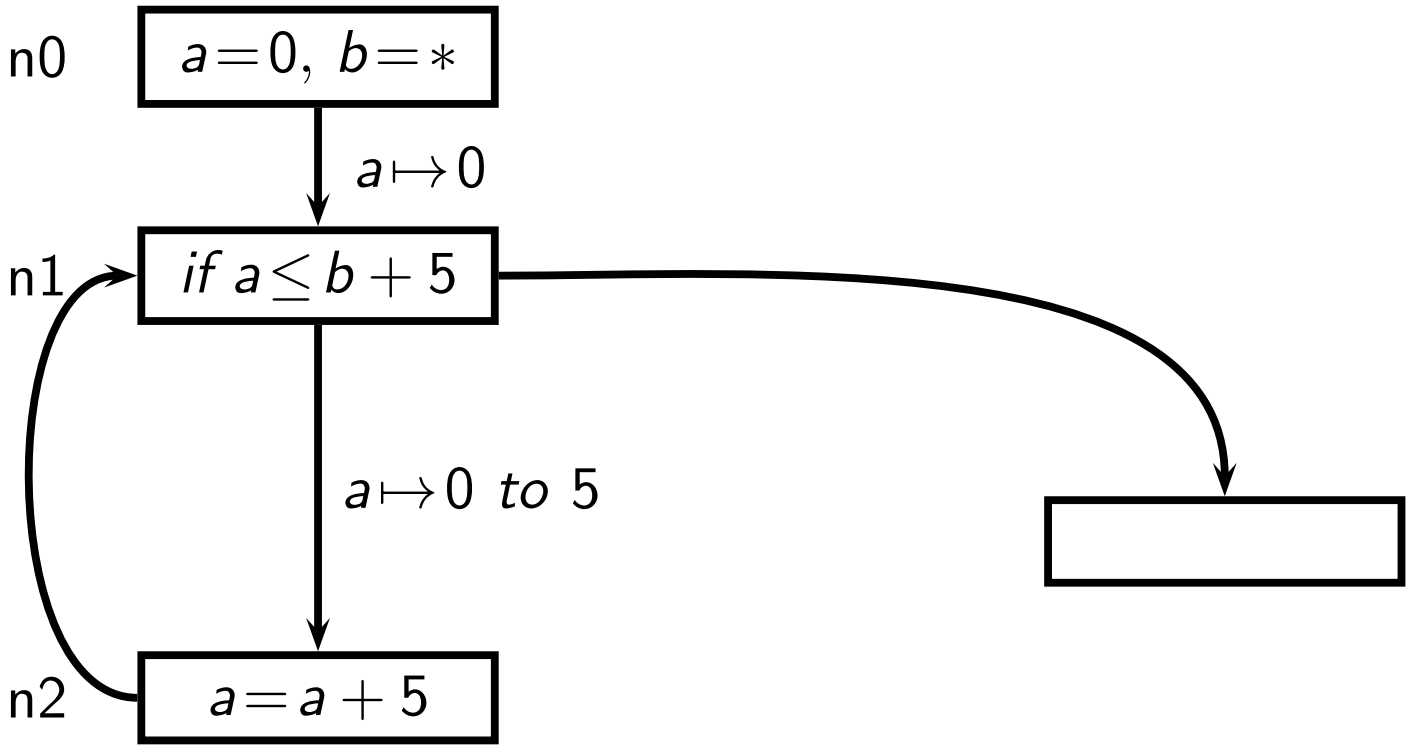
Example: The MFP Solution



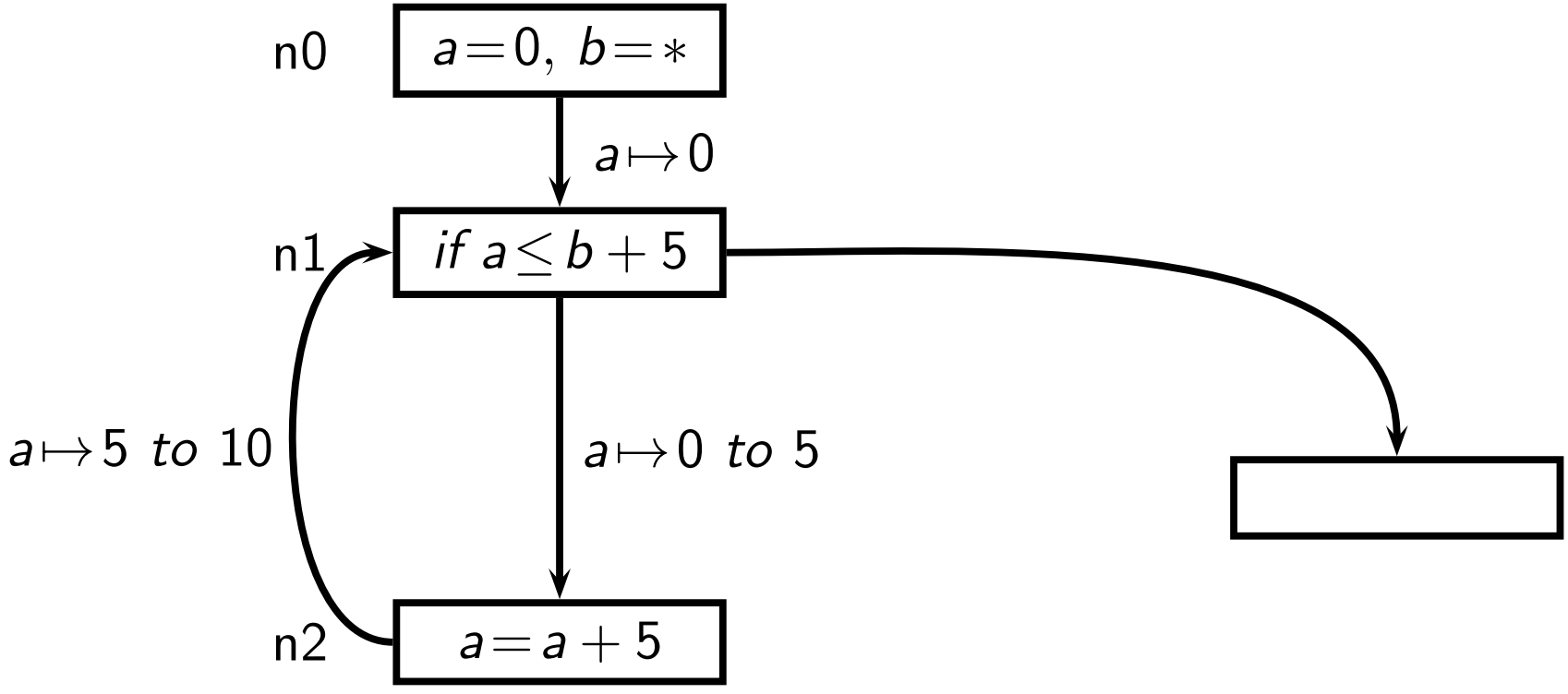
Example: The MFP Solution



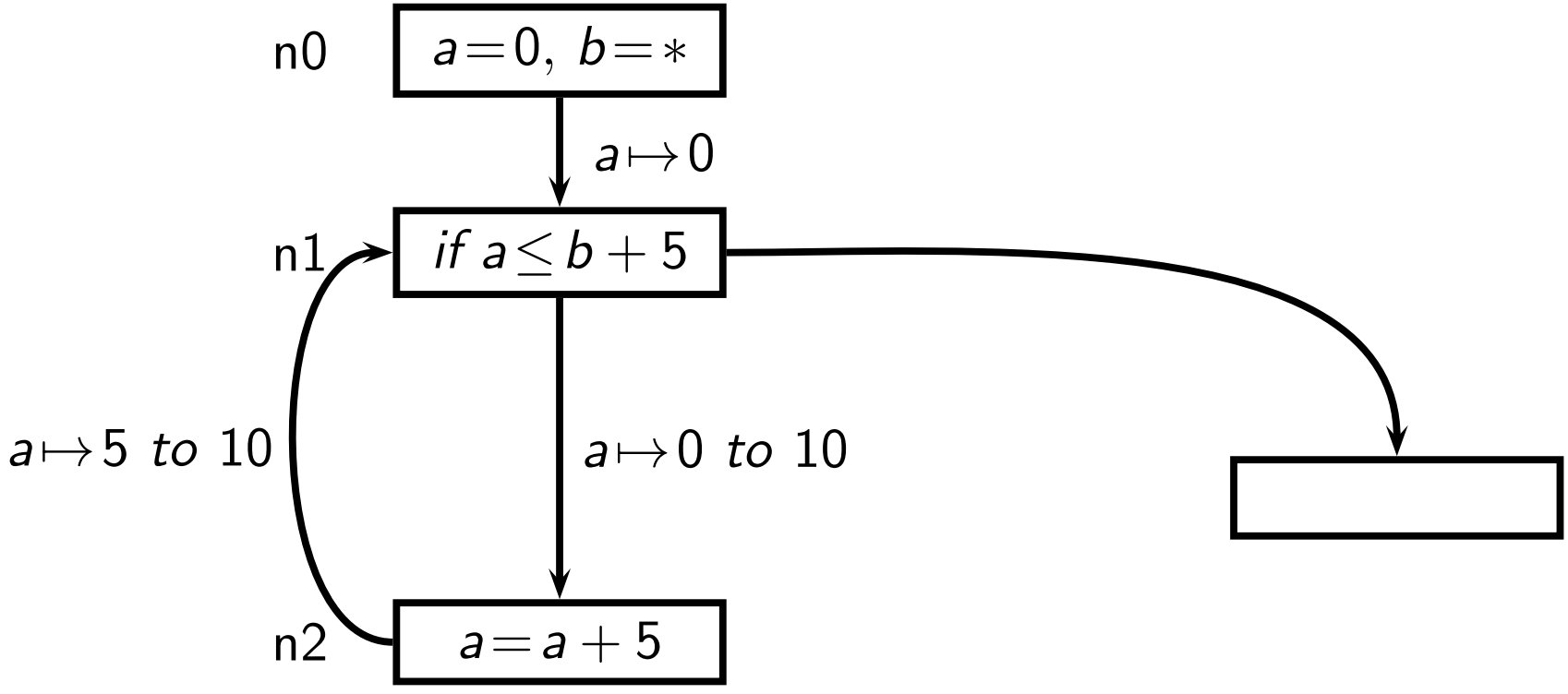
Example: The MFP Solution



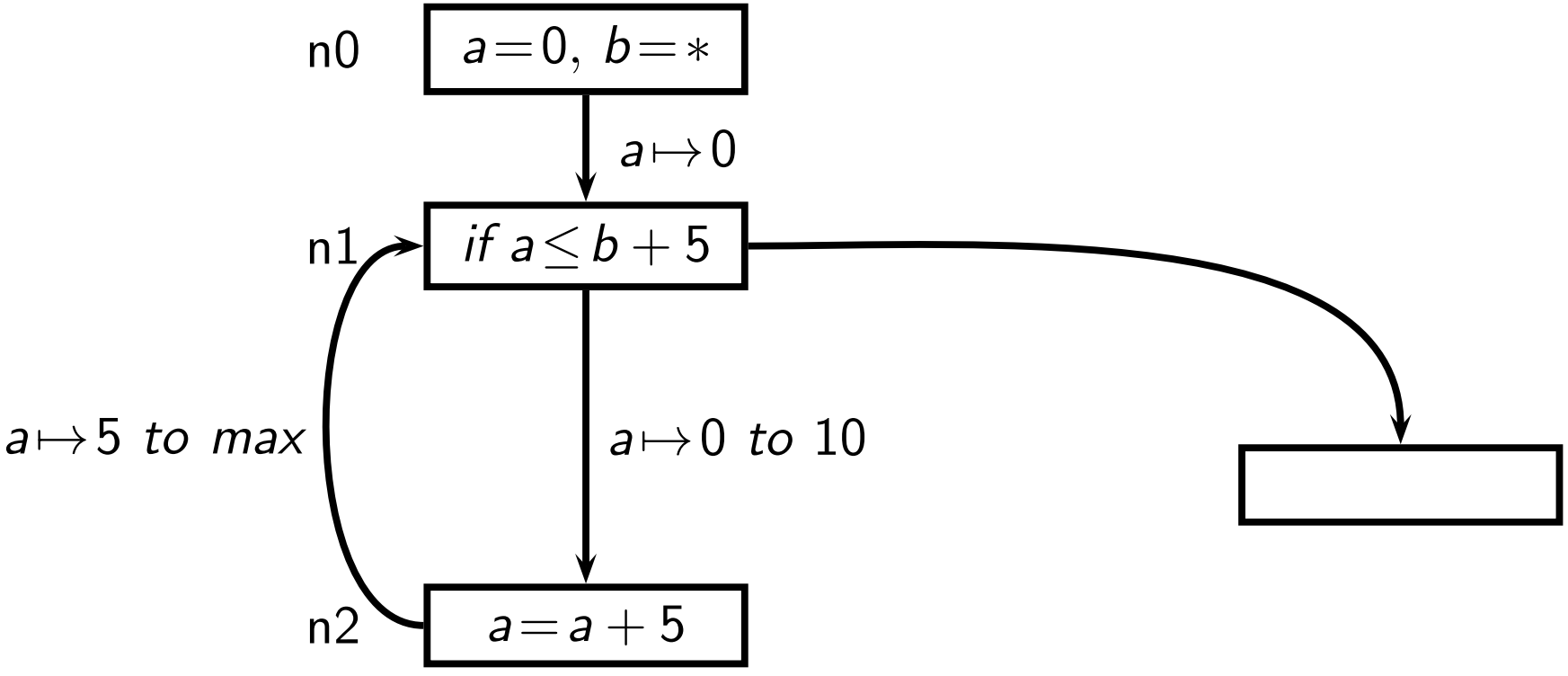
Example: The MFP Solution



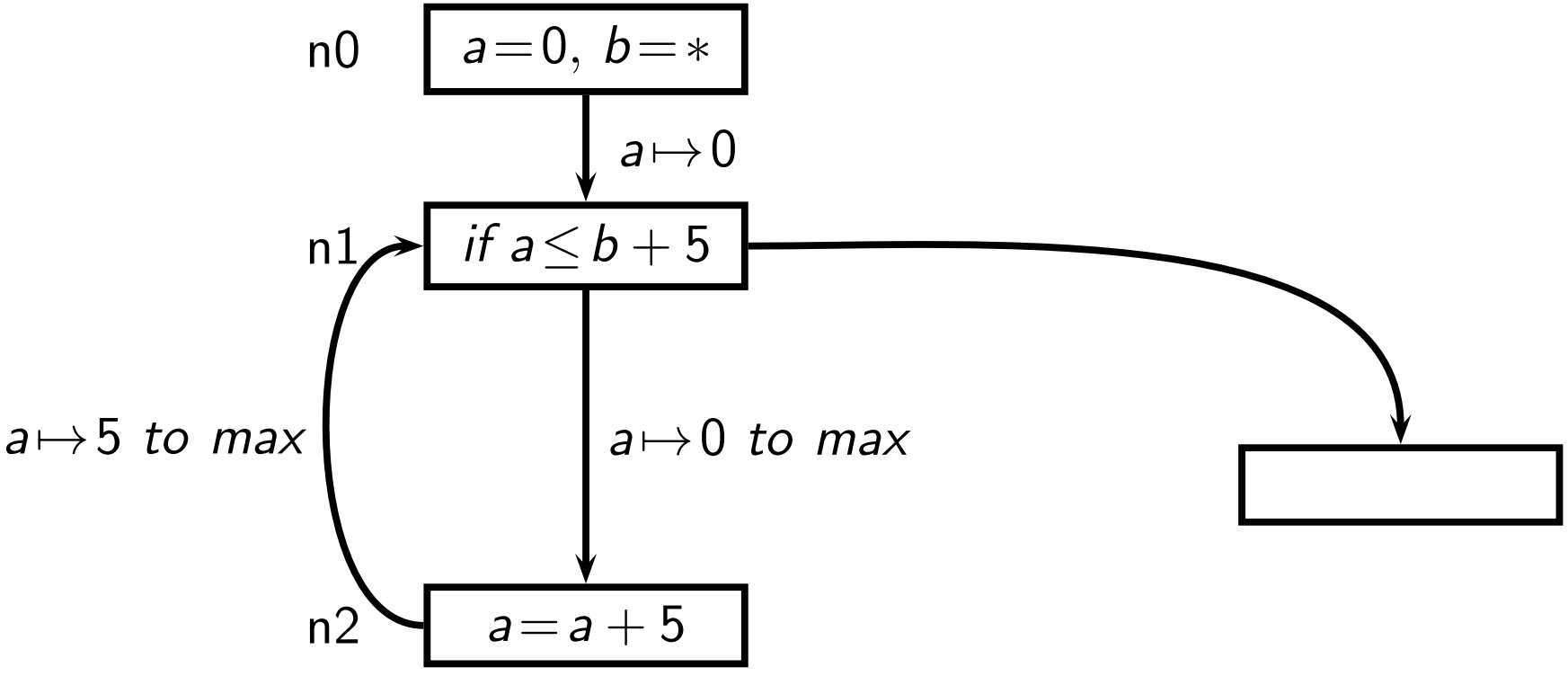
Example: The MFP Solution



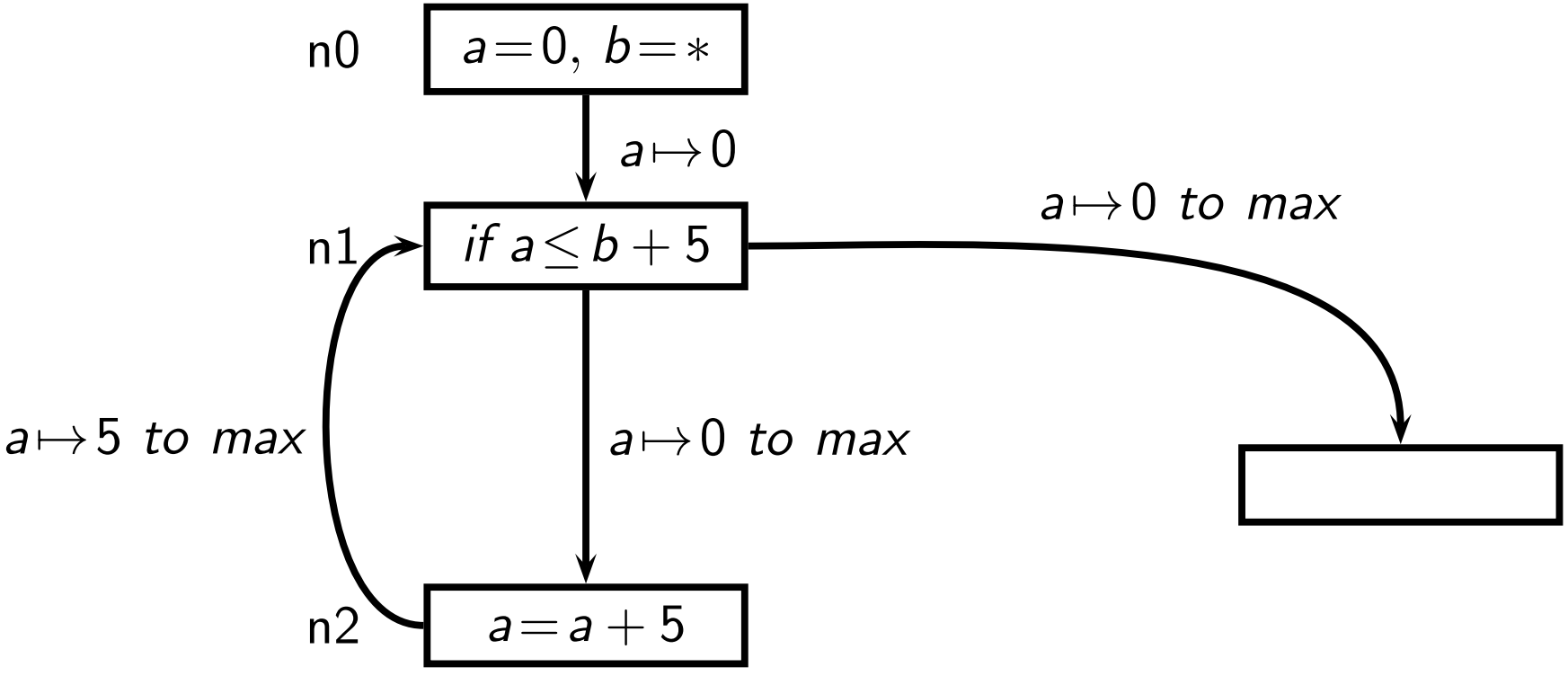
Example: The MFP Solution



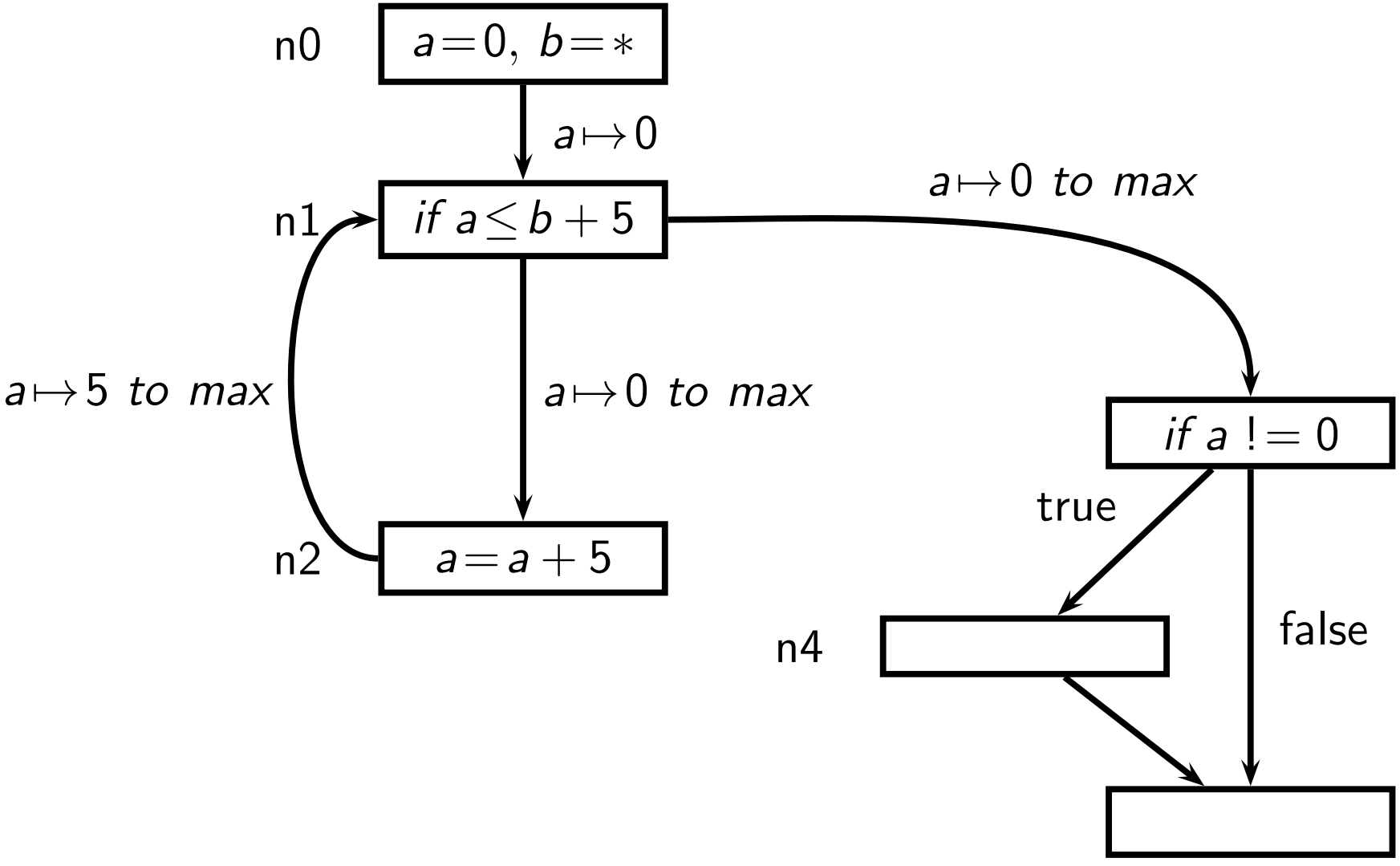
Example: The MFP Solution



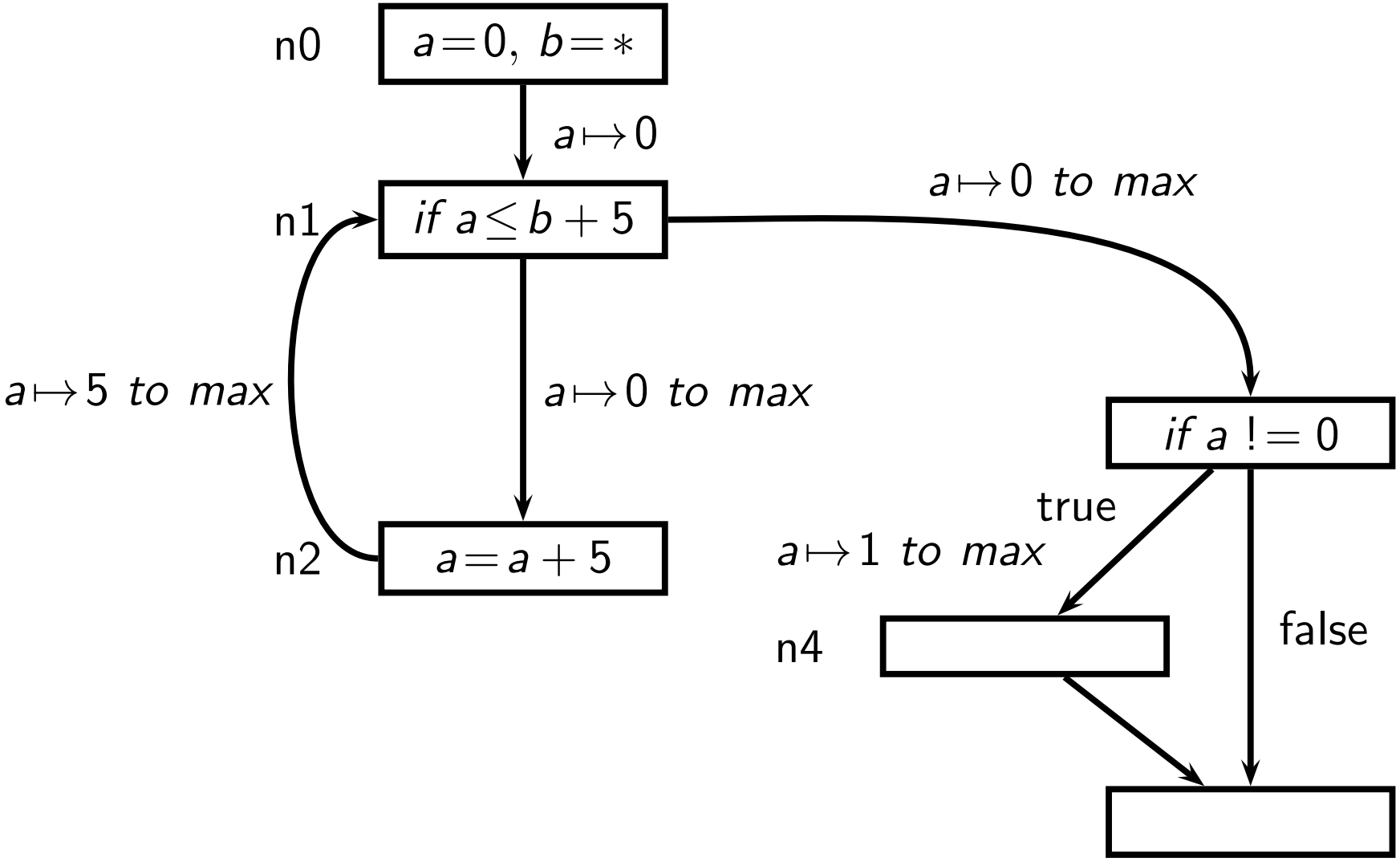
Example: The MFP Solution



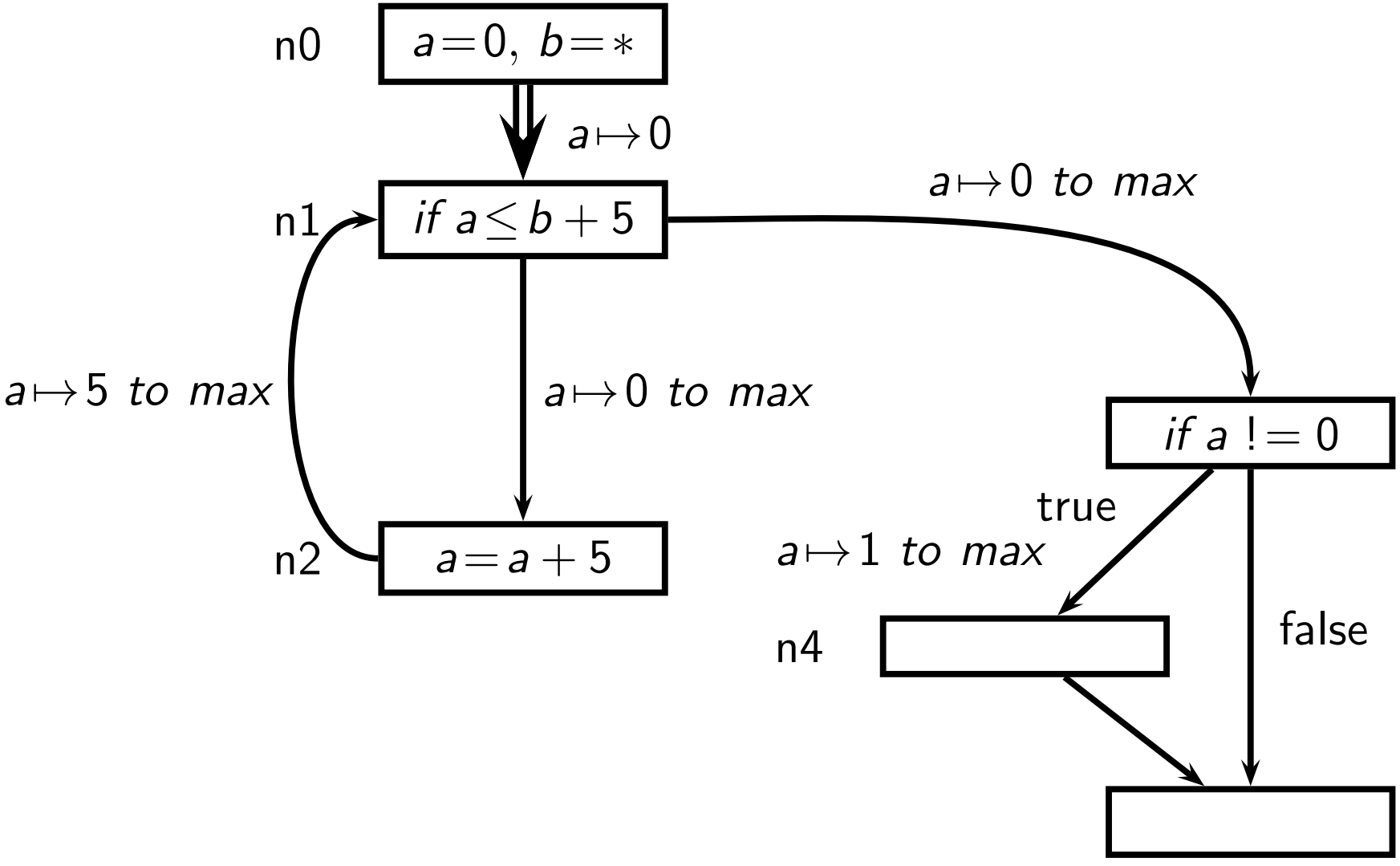
Example: The MFP Solution



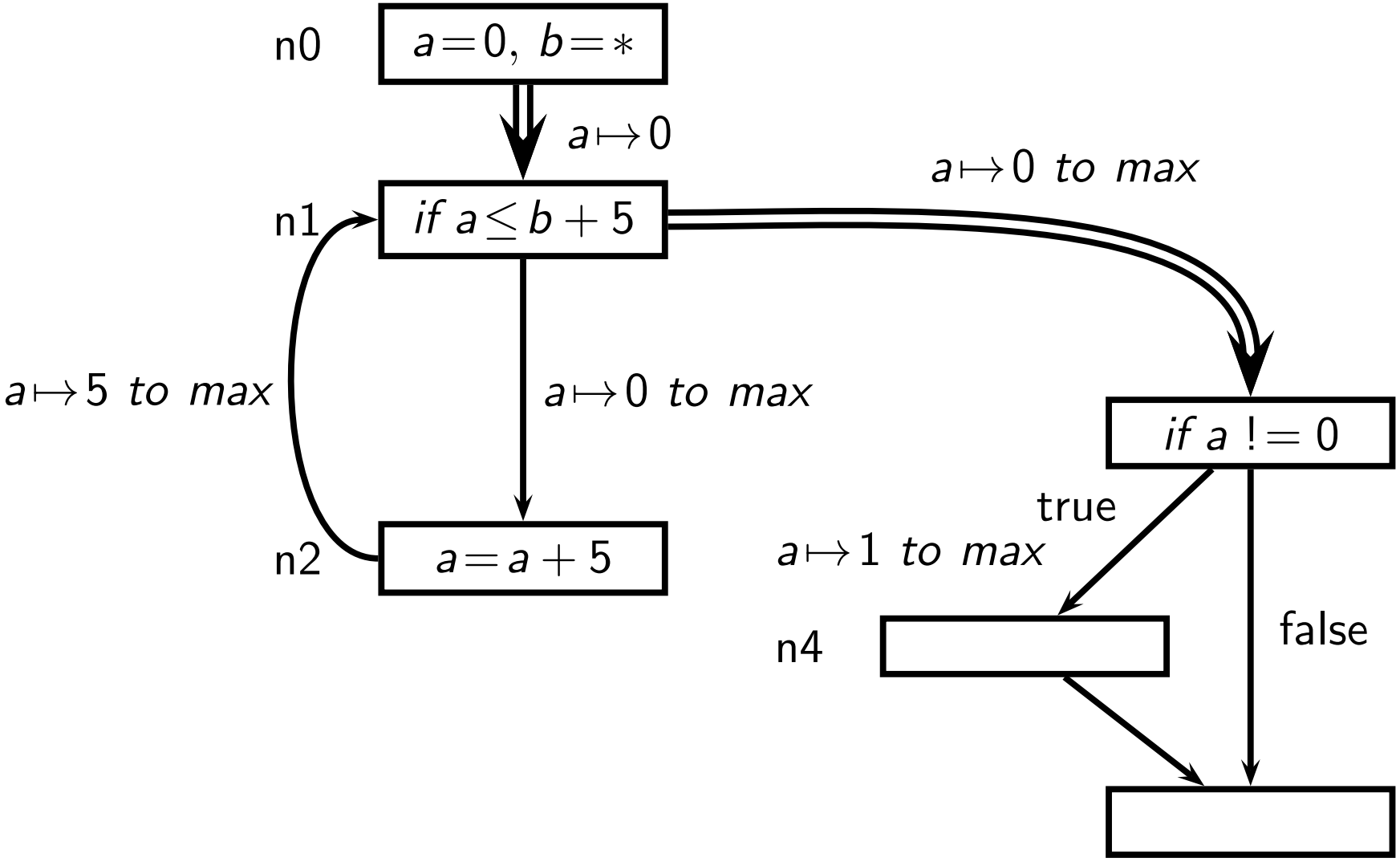
Example: The MFP Solution



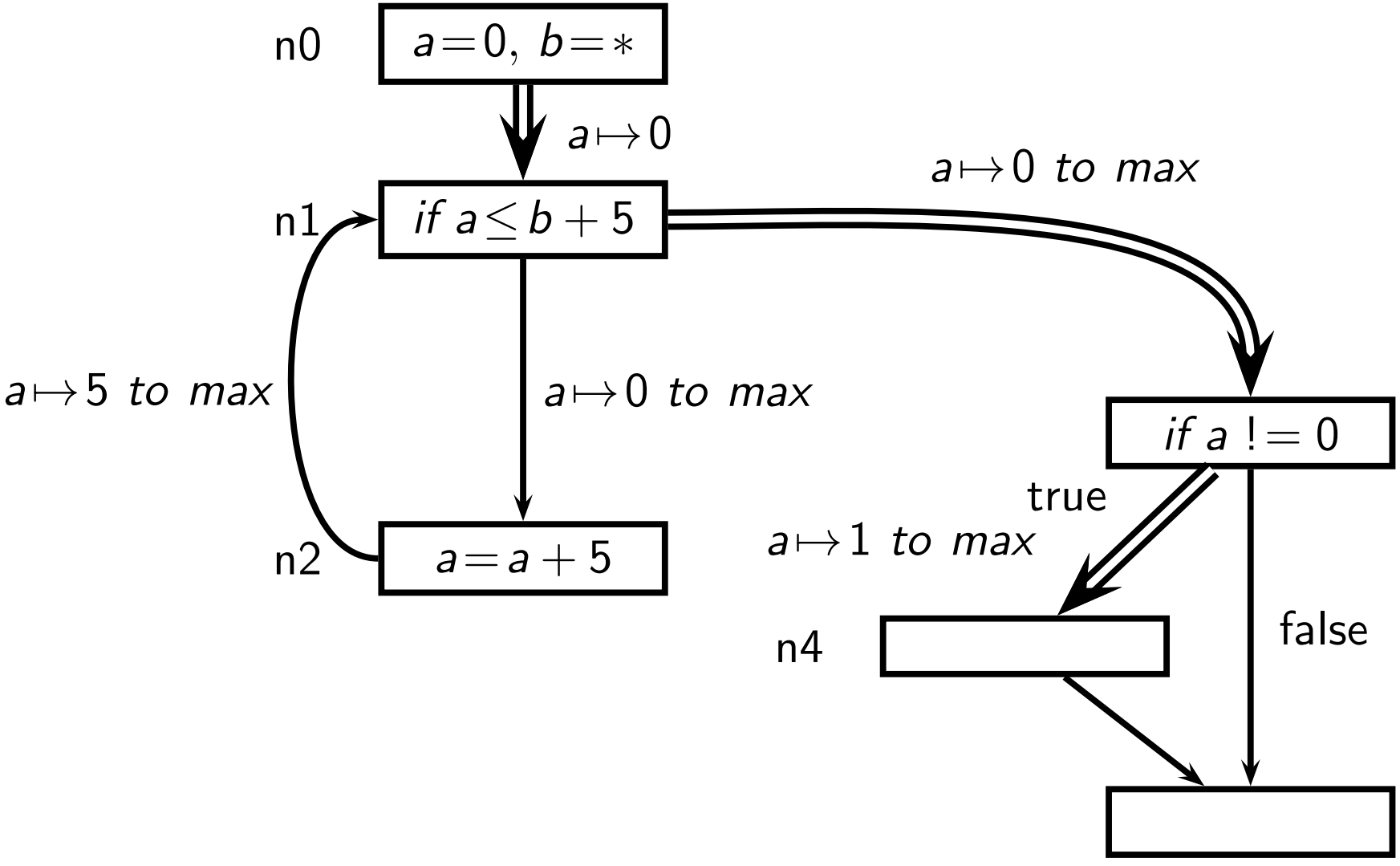
Example: The MFP Solution



Example: The MFP Solution



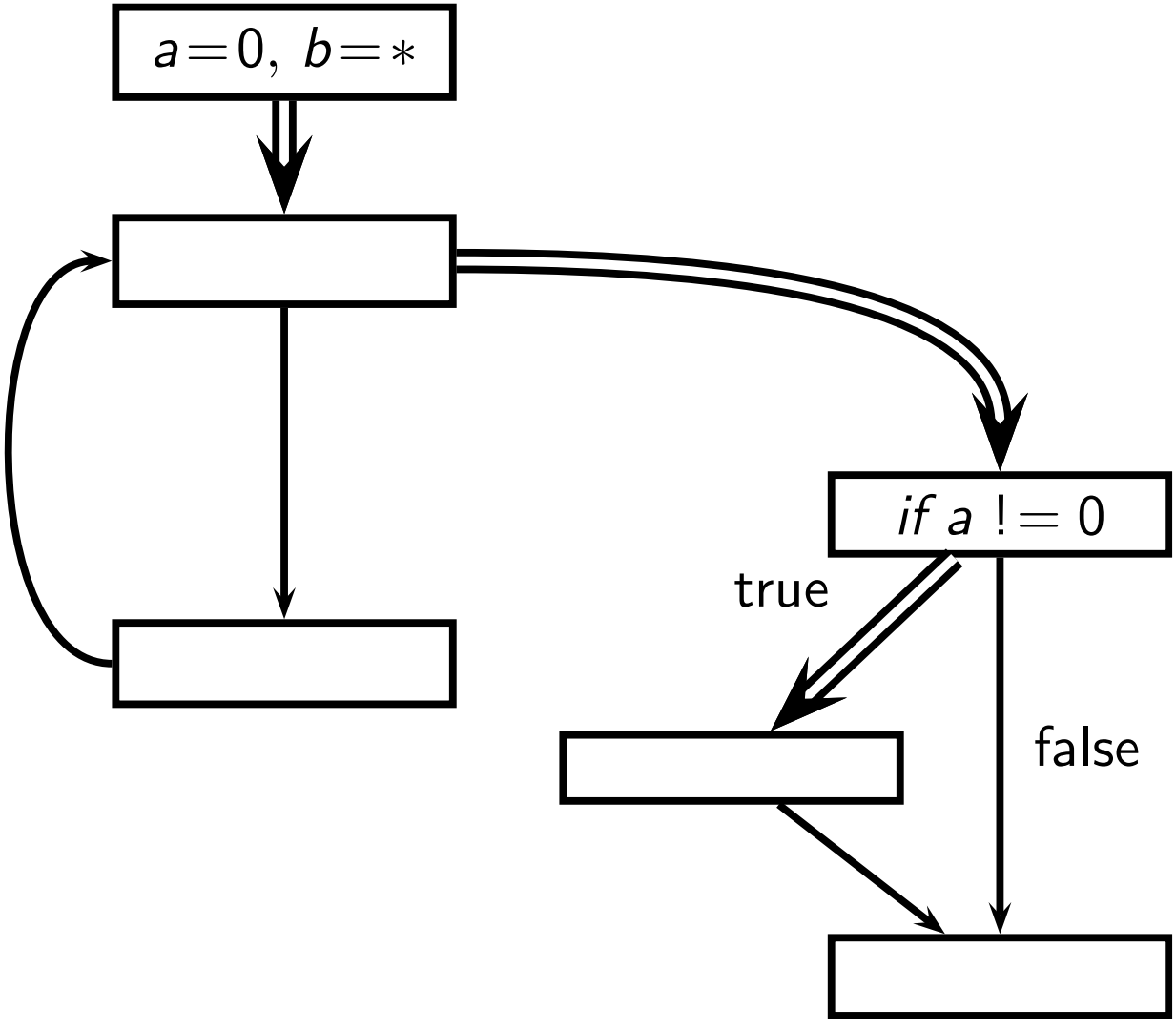
Example: The MFP Solution



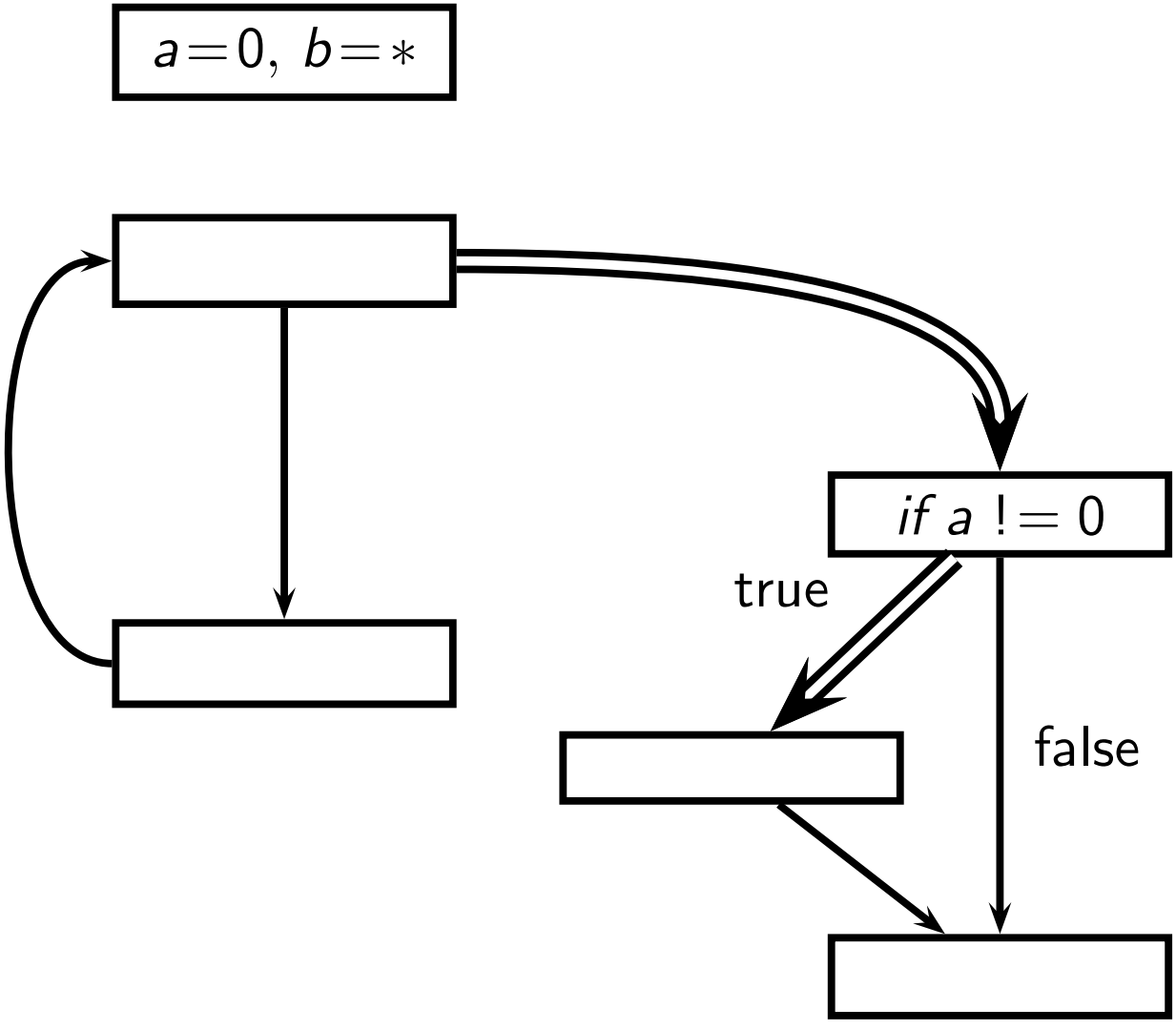
To Improve the MFP Solution

We need to *separate feasible and infeasible control flow paths*.

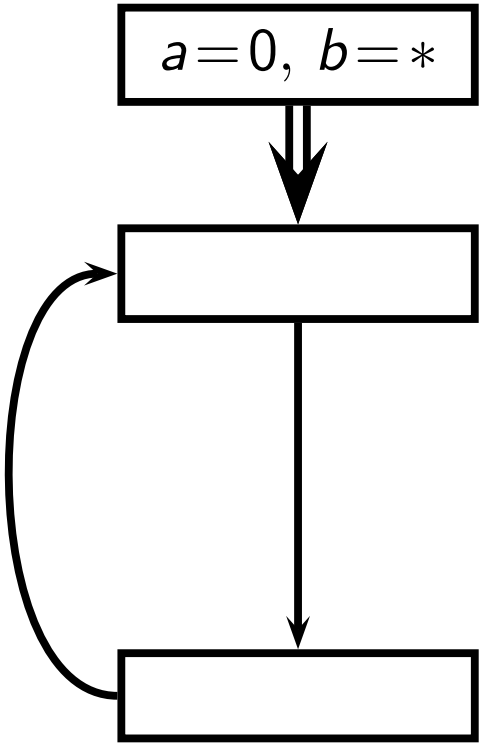
Issue 1: Program Representation Admits Infeasible Paths

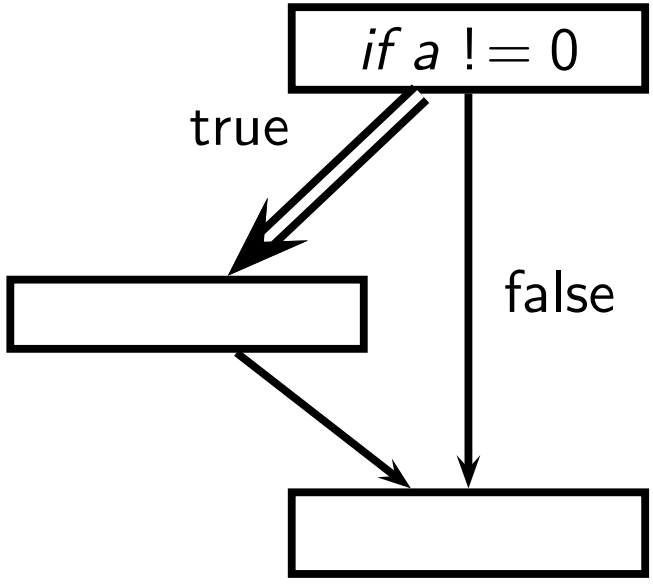


Issue 2: Edge Removal Affects Feasible Control Flow Paths

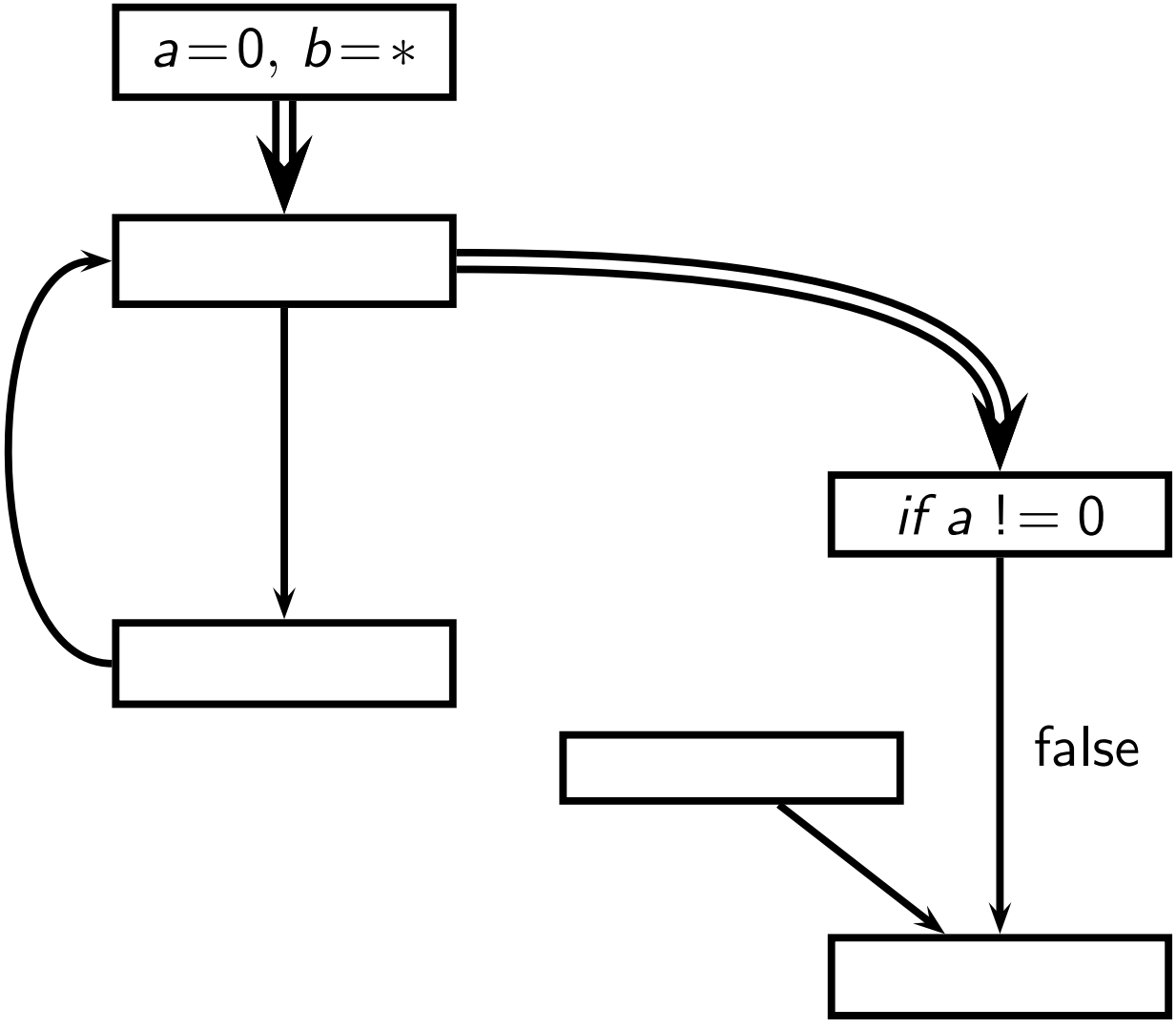


Issue 2: Edge Removal Affects Feasible Control Flow Paths

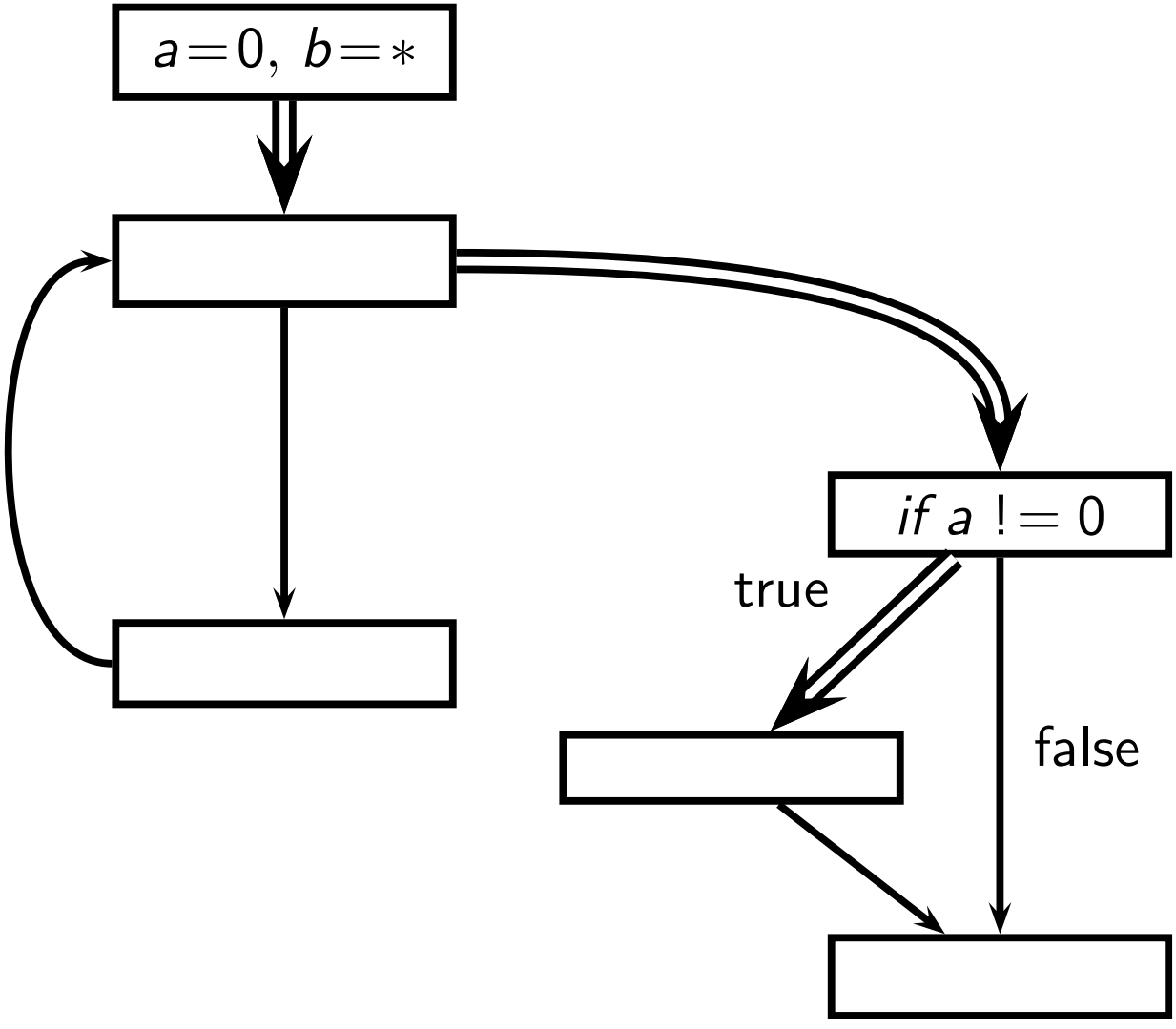




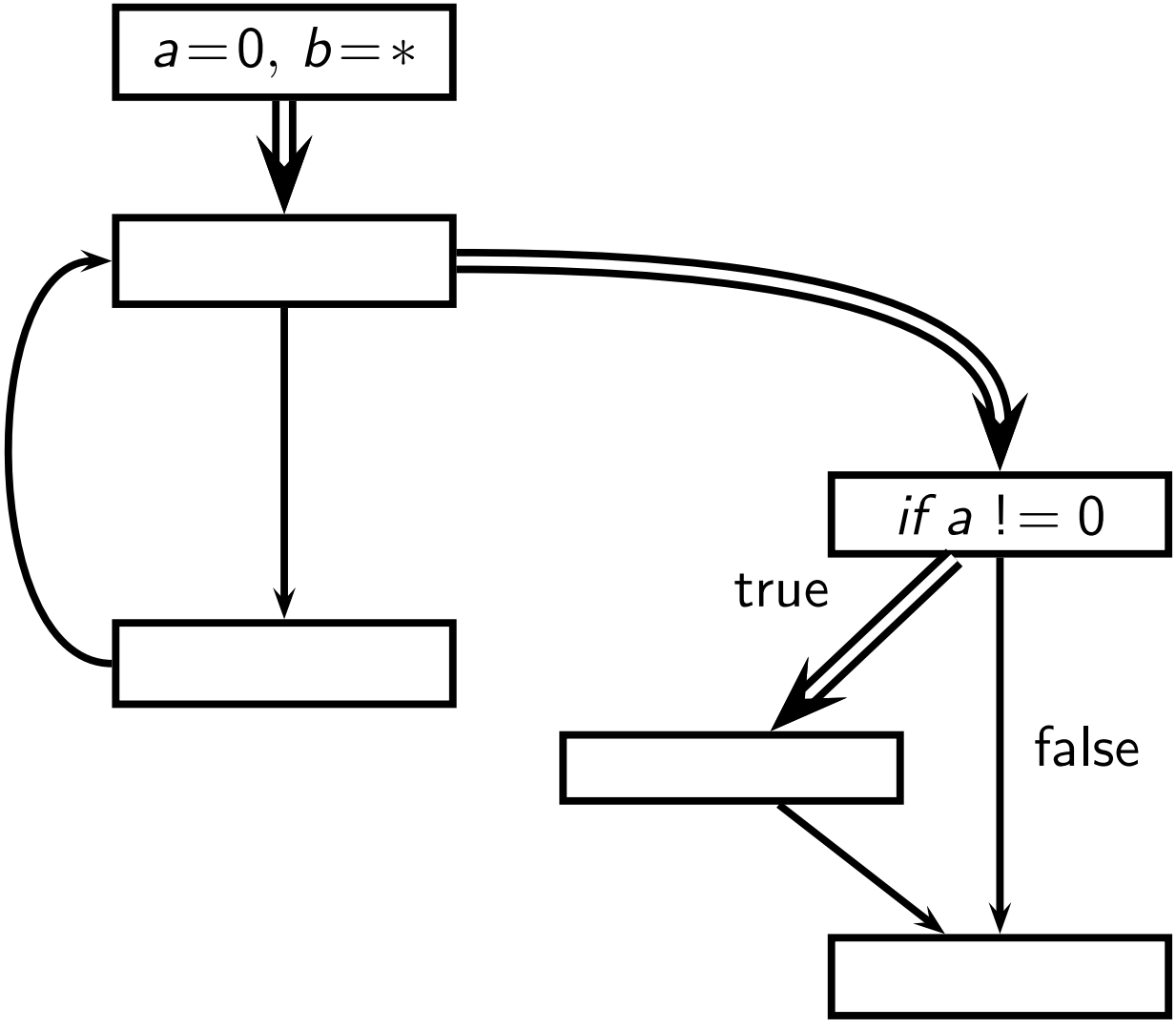
Issue 2: Edge Removal Affects Feasible Control Flow Paths



Issue 2: Edge Removal Affects Feasible Control Flow Paths



Issue 3: Graph Transformation Cost is Exponential



Issues in Removing infeasible Control Flow Paths

1. Static representation of programs admits control flow paths that are infeasible
2. Deletion of an edge could remove the feasible paths also
3. Restructuring control flow graph can cause exponential blow up in size of graph
4. Static analysis suffers imprecision because it is not possible to separate infeasible paths from feasible ones.

Can we remove the imprecision without changing CFG ?

Main Idea

The effect of

- separating the feasible and the infeasible control flow paths in CFG

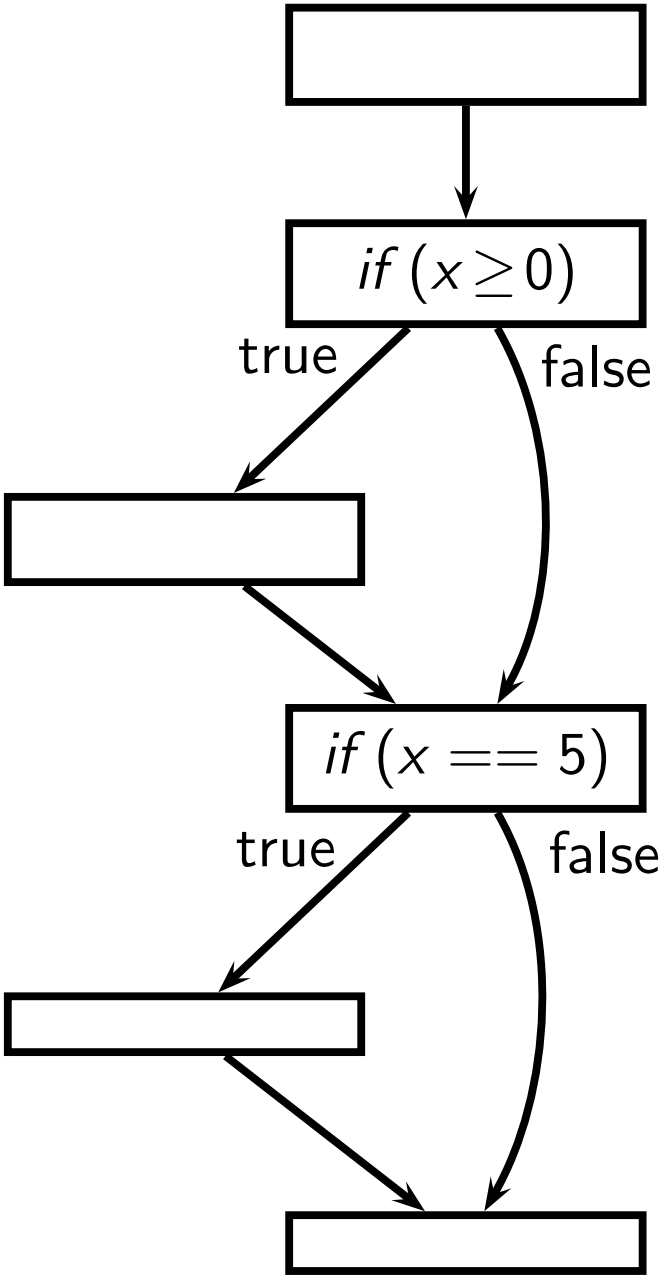
  could be achieved by just

- separating the information flowing along the feasible and infeasible control flow paths

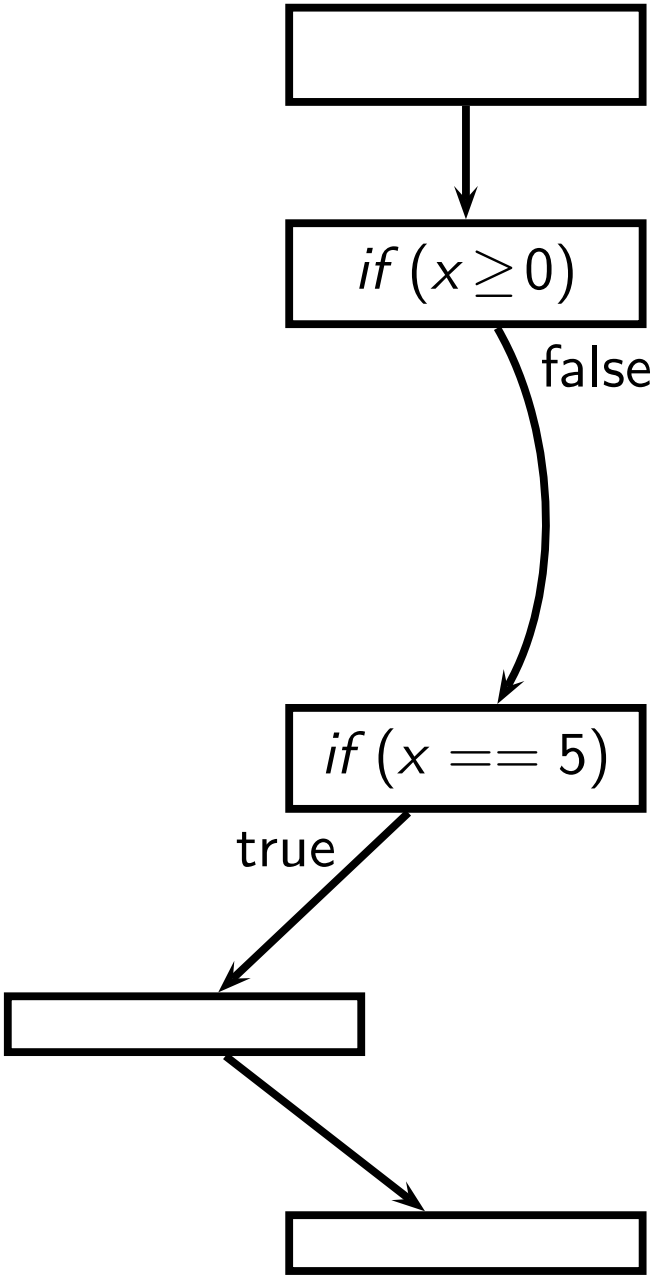
Identifying Infeasible Control Flow Paths (Bodik et.al)

- Bodik et.al [1999] proposed a method to detect *minimal infeasible path segments*
- We use this work to achieve desired separation between information flowing along feasible and infeasible control flow paths.

Minimal Infeasible Path Segment (MIPS) (Bodik et.al)

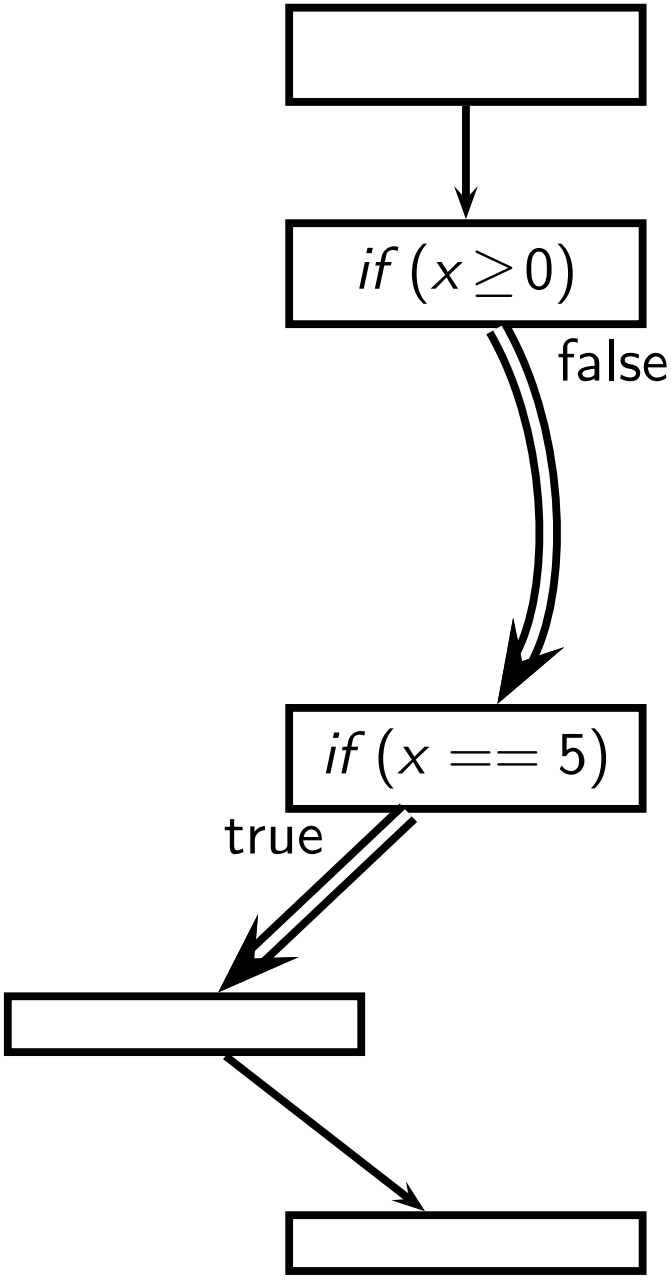


Minimal Infeasible Path Segment (MIPS) (Bodik et.al)



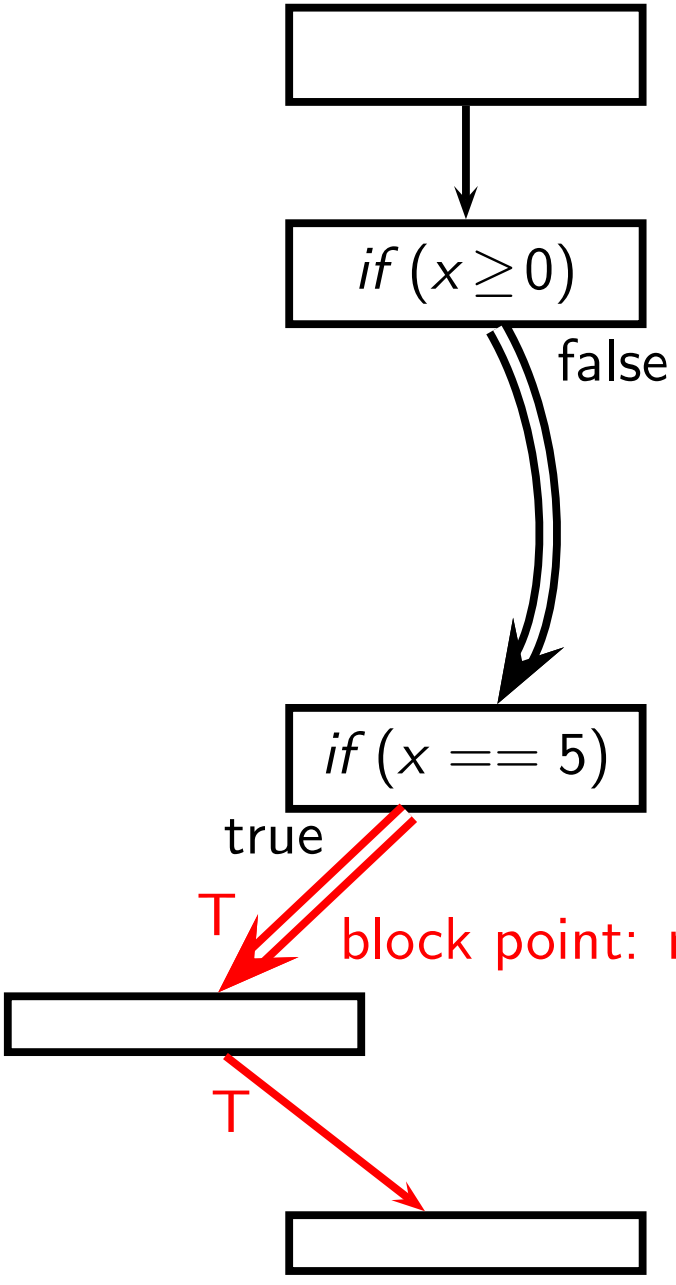
- Infeasible Control Flow Path

Minimal Infeasible Path Segment (MIPS) (Bodik et.al)



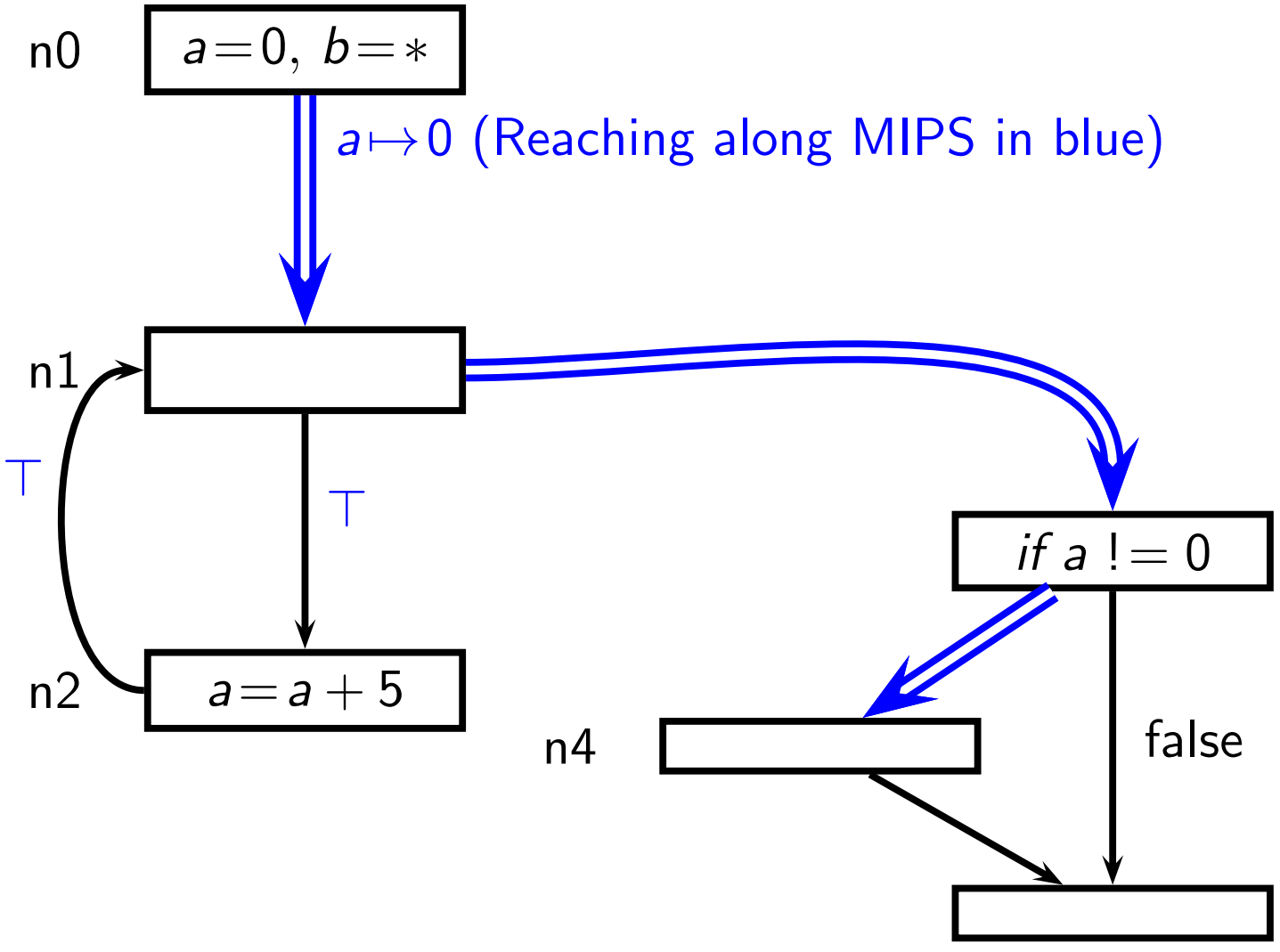
- Infeasible Control Flow Path
- Minimal Infeasible Control Flow Path
  - Any super path is infeasible
  - Any sub path is feasible

Data Flow Through MIPS (Bodik et.al)

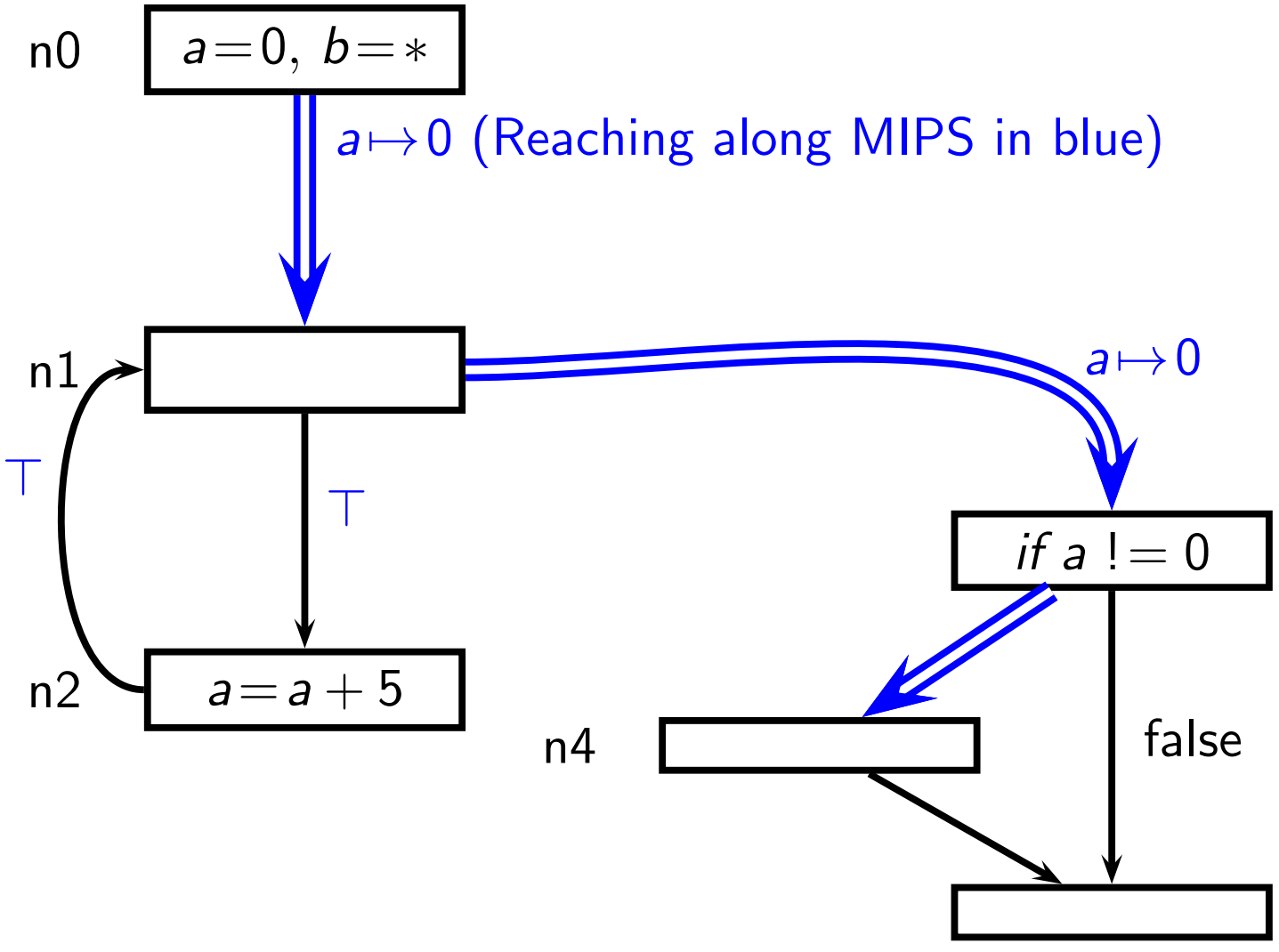


block point: no data flow from this point onwards in MIPS

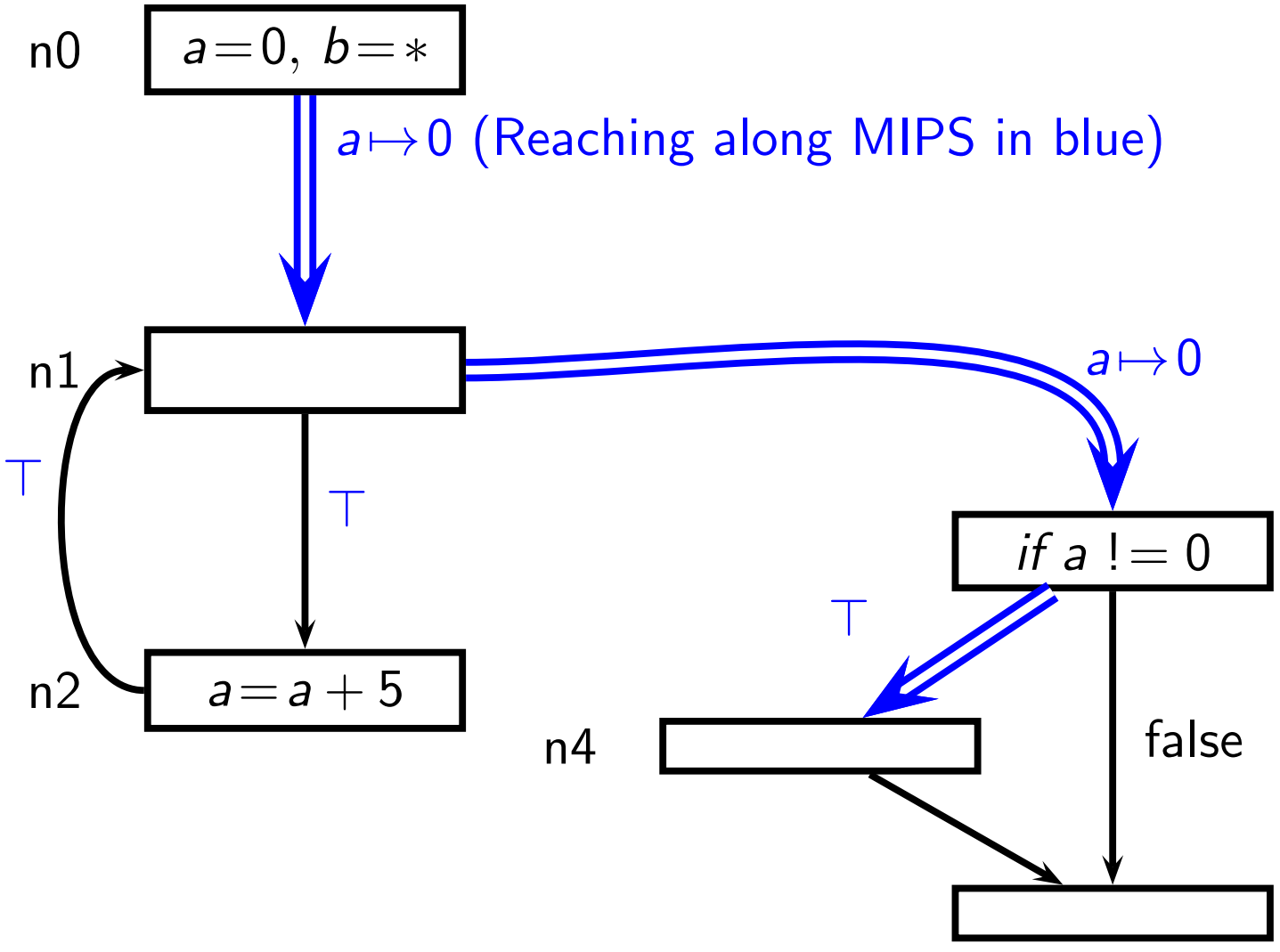
Separating Information Flowing along MIPS



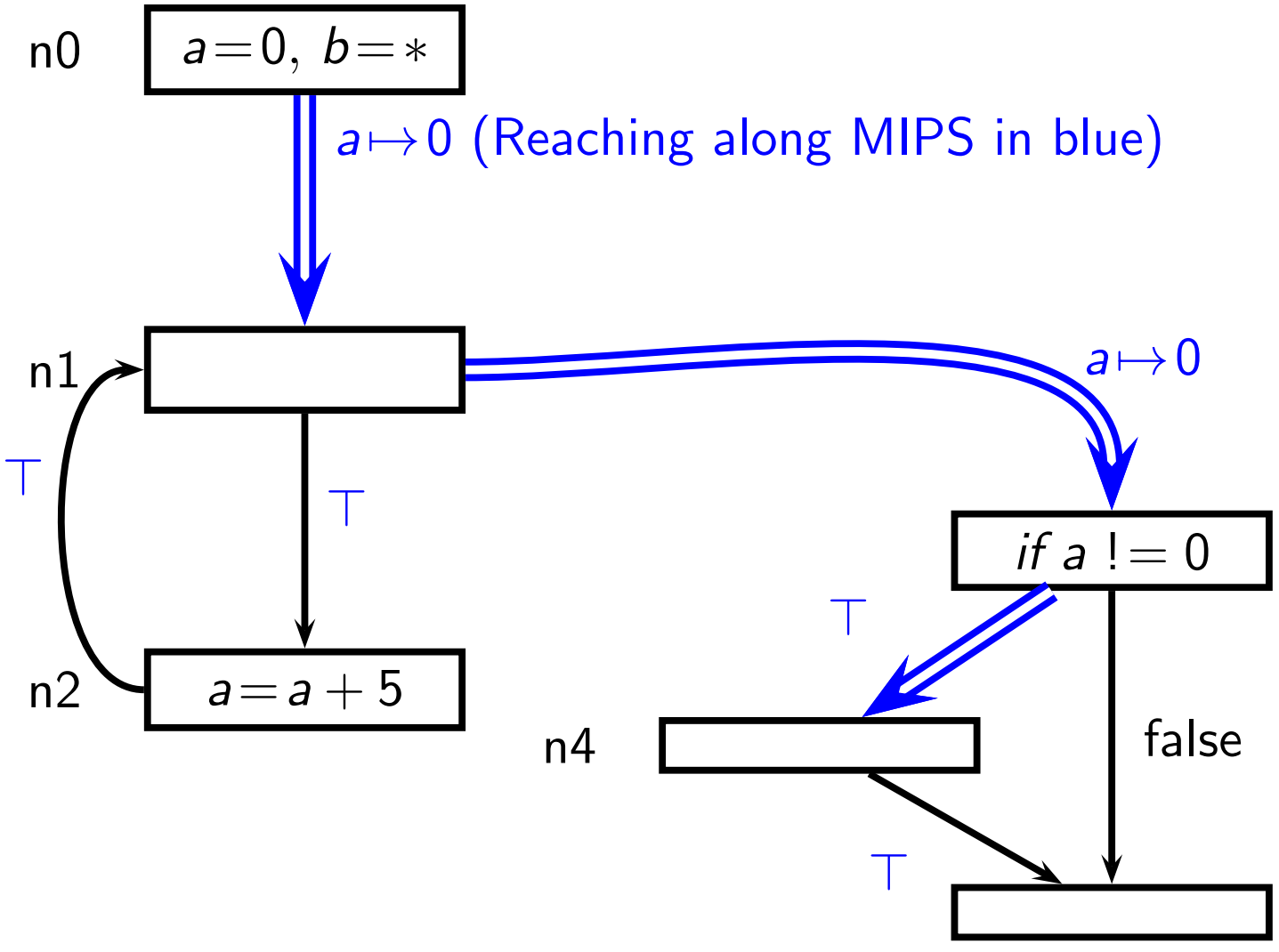
Separating Information Flowing along MIPS



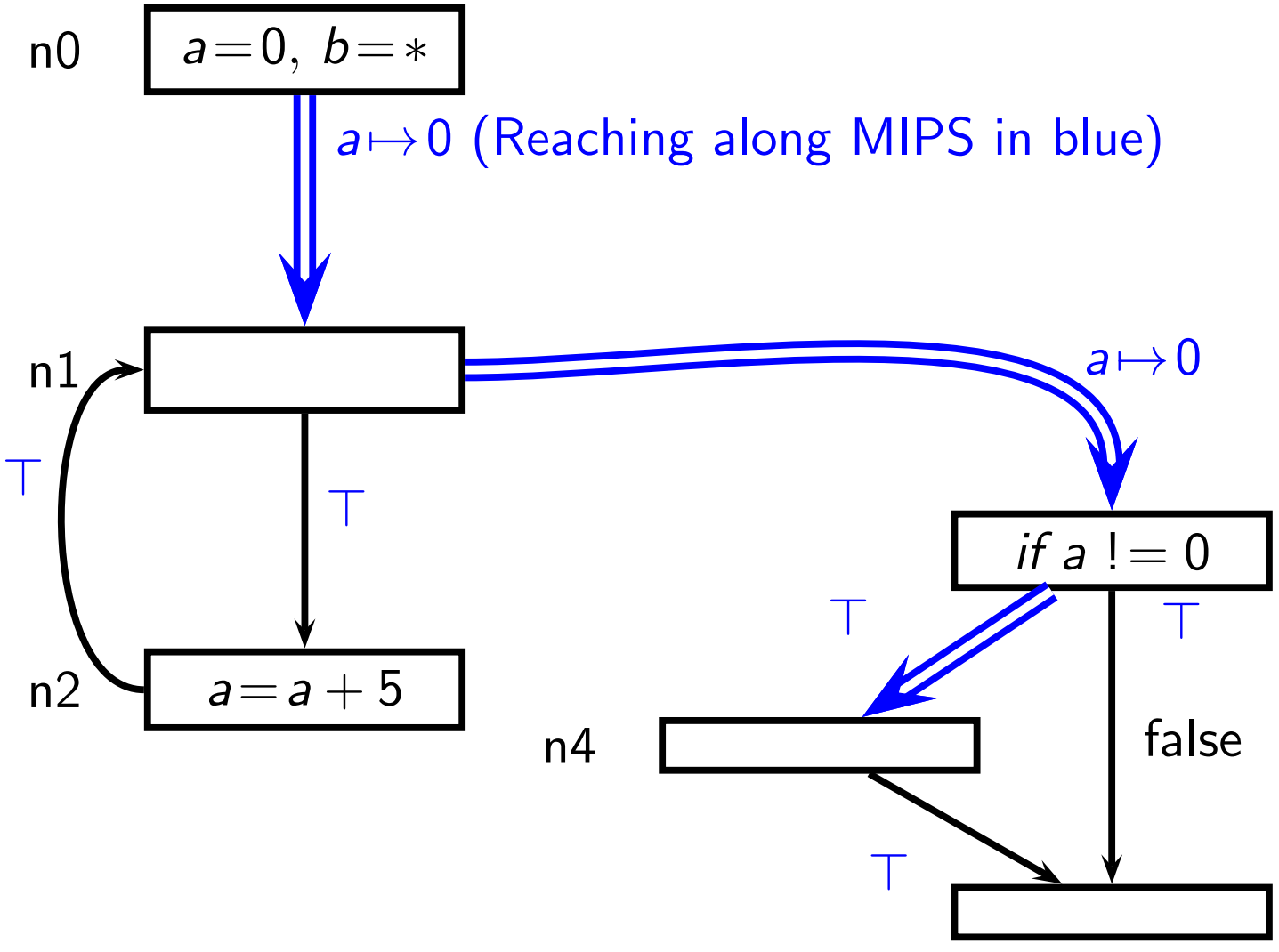
Separating Information Flowing along MIPS



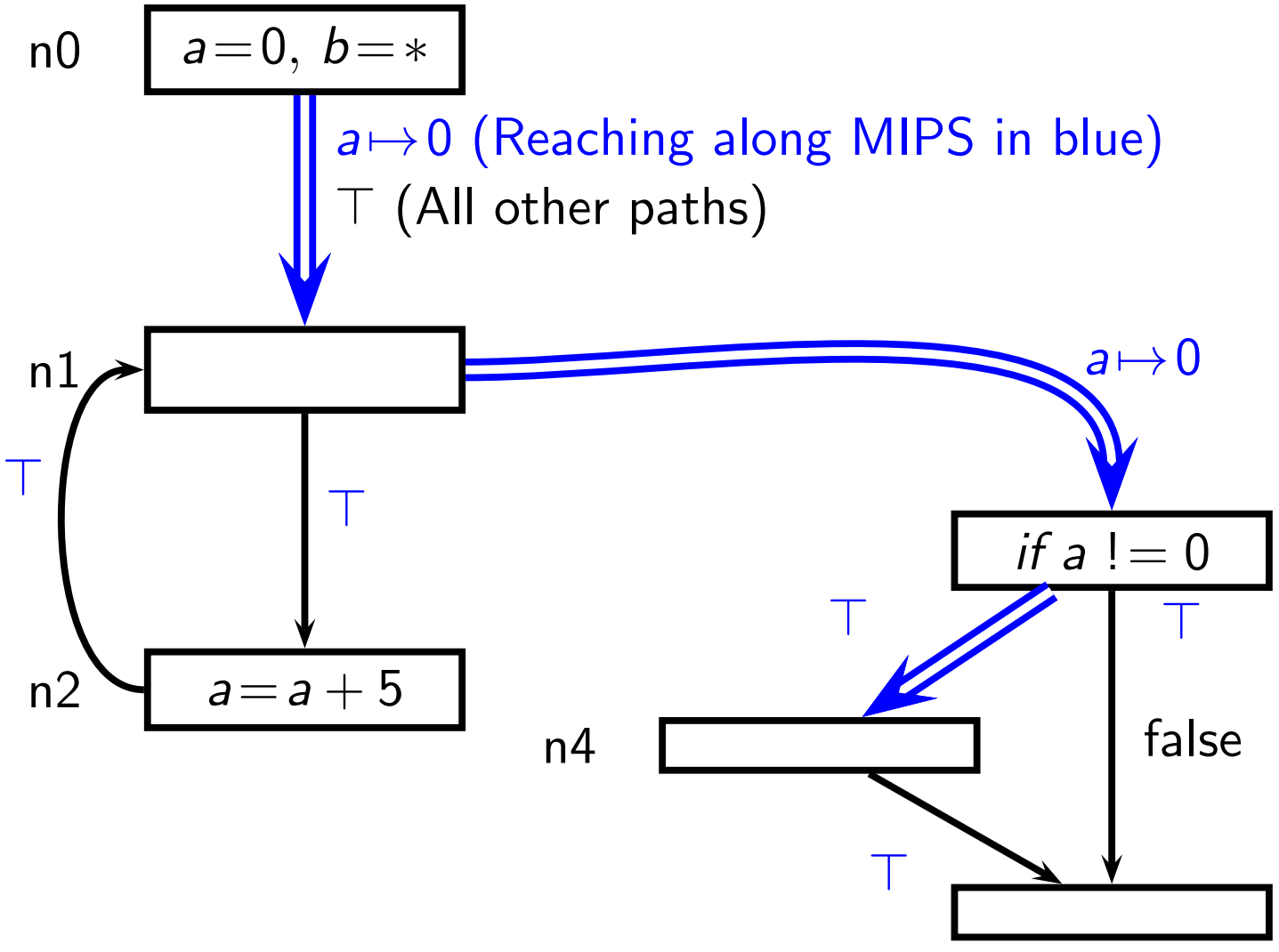
Separating Information Flowing along MIPS



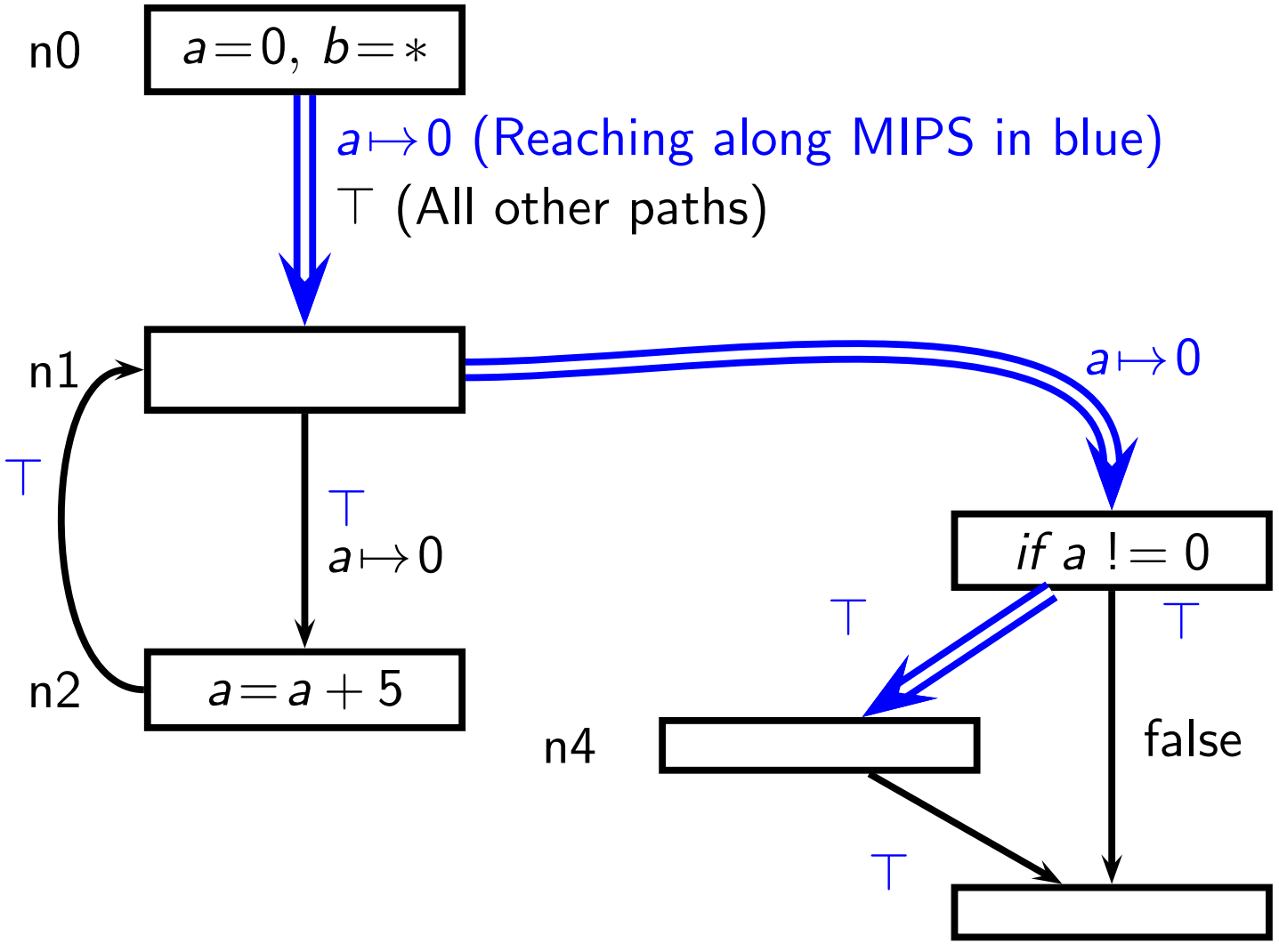
Separating Information Flowing along MIPS



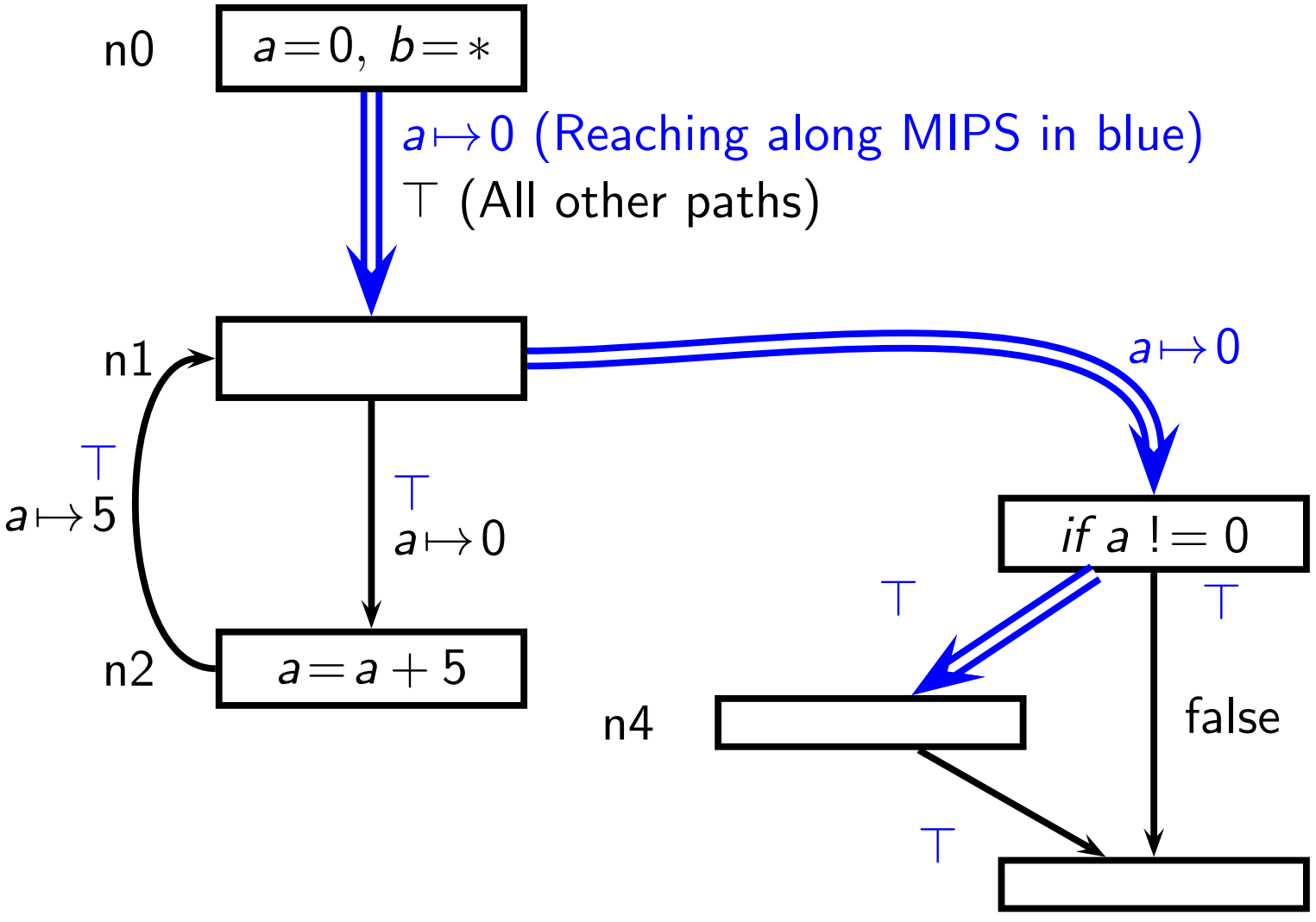
Separating Information Flowing along MIPS



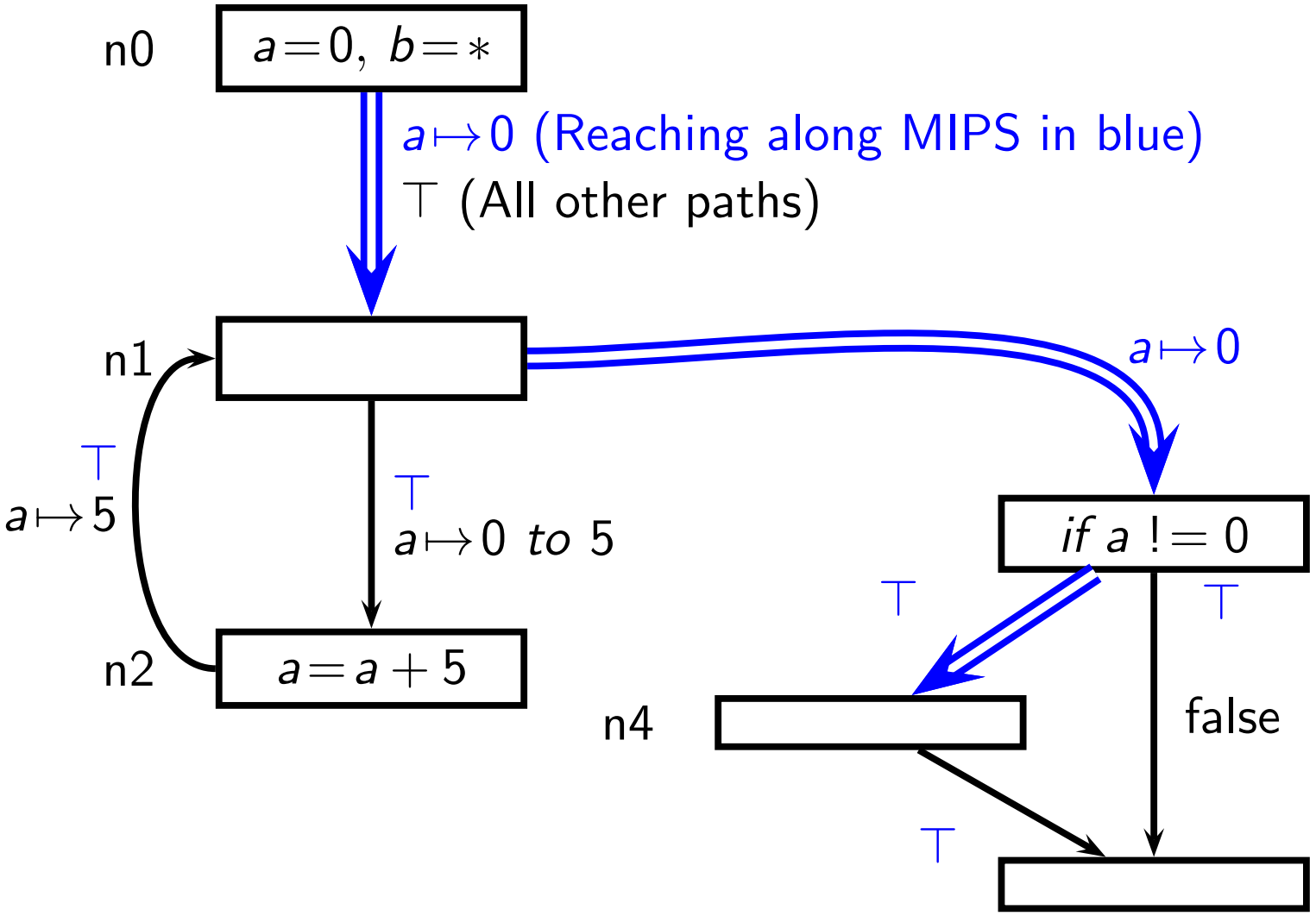
Separating Information Flowing along MIPS



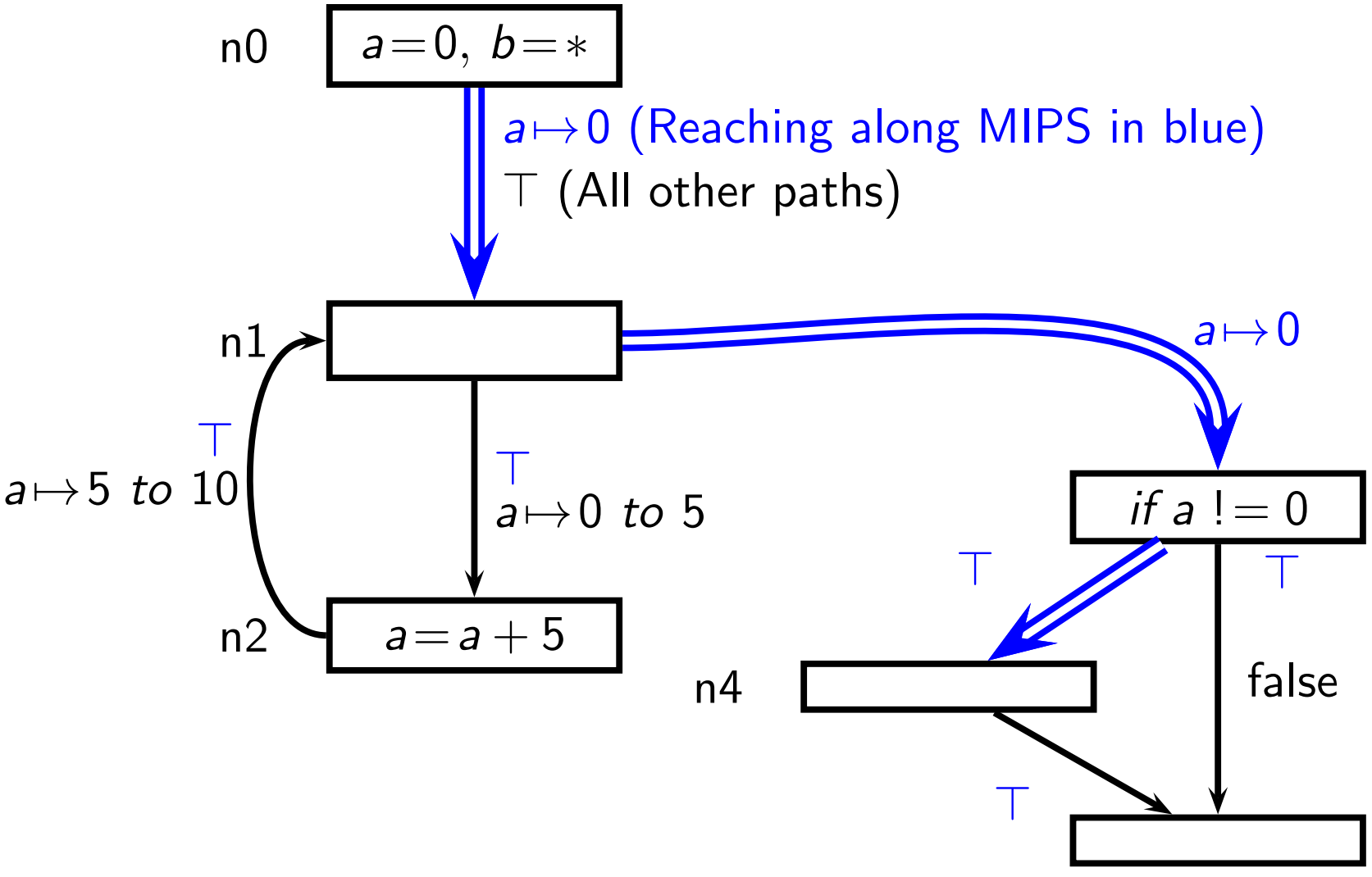
Separating Information Flowing along MIPS



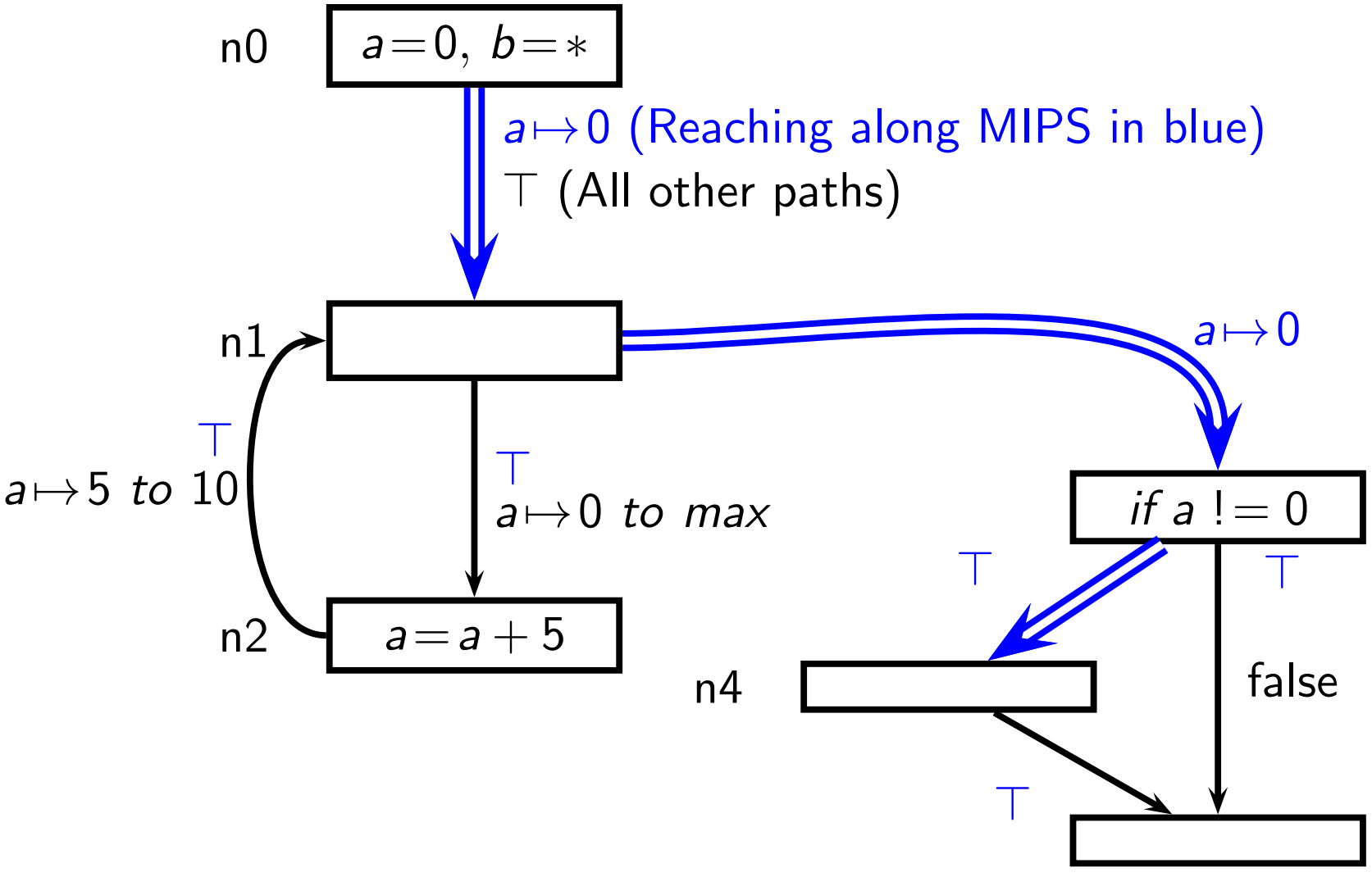
Separating Information Flowing along MIPS



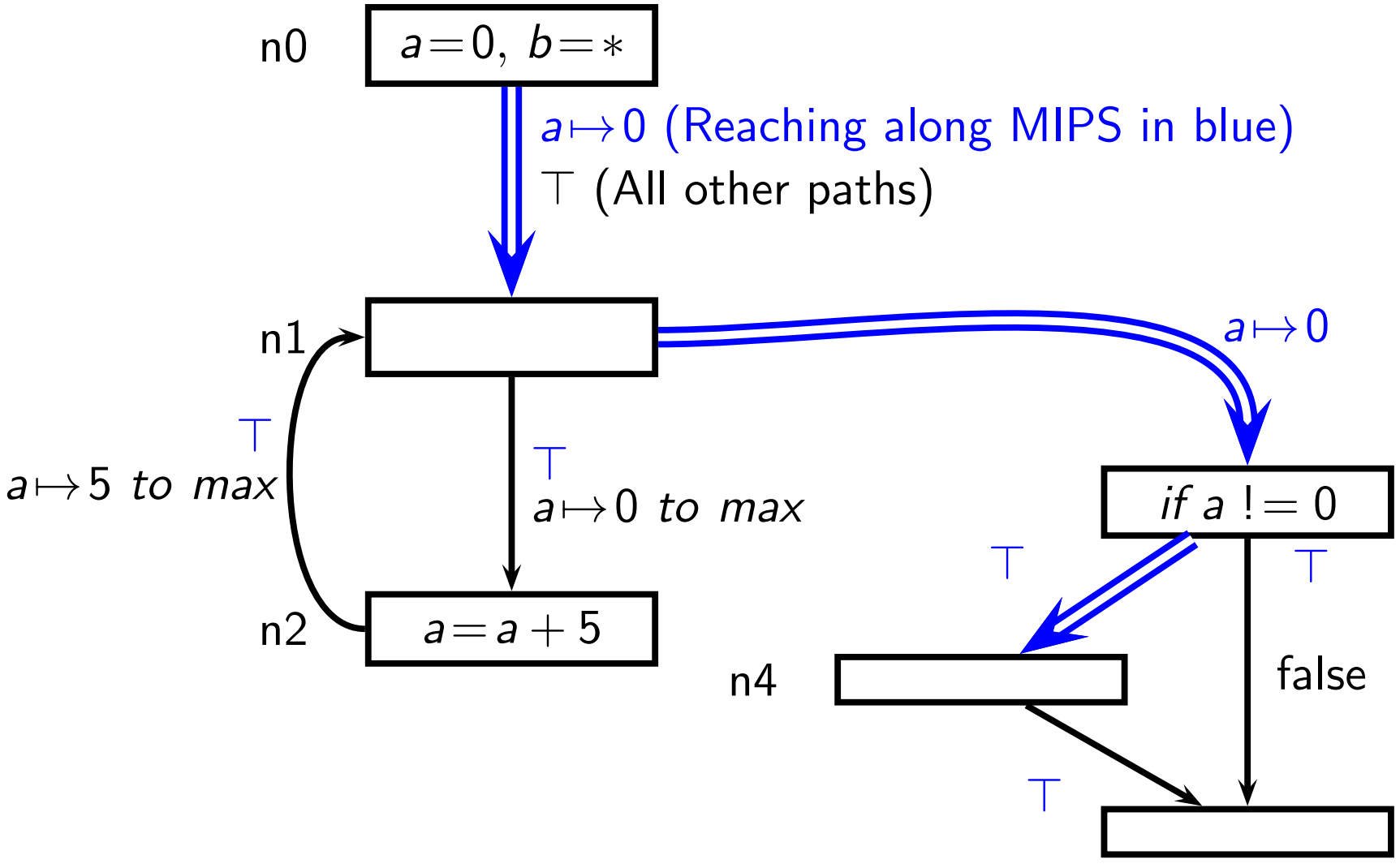
Separating Information Flowing along MIPS



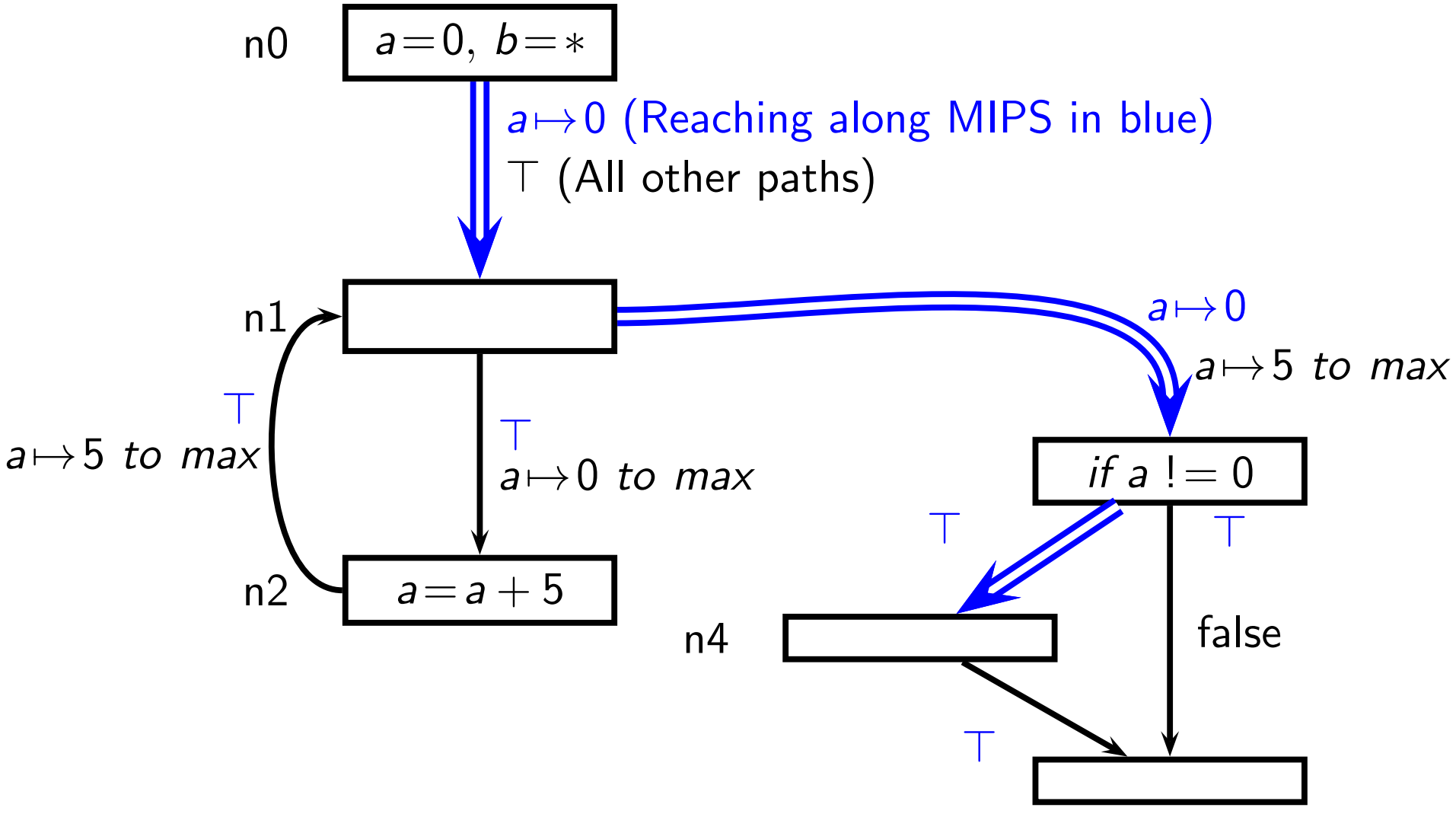
Separating Information Flowing along MIPS



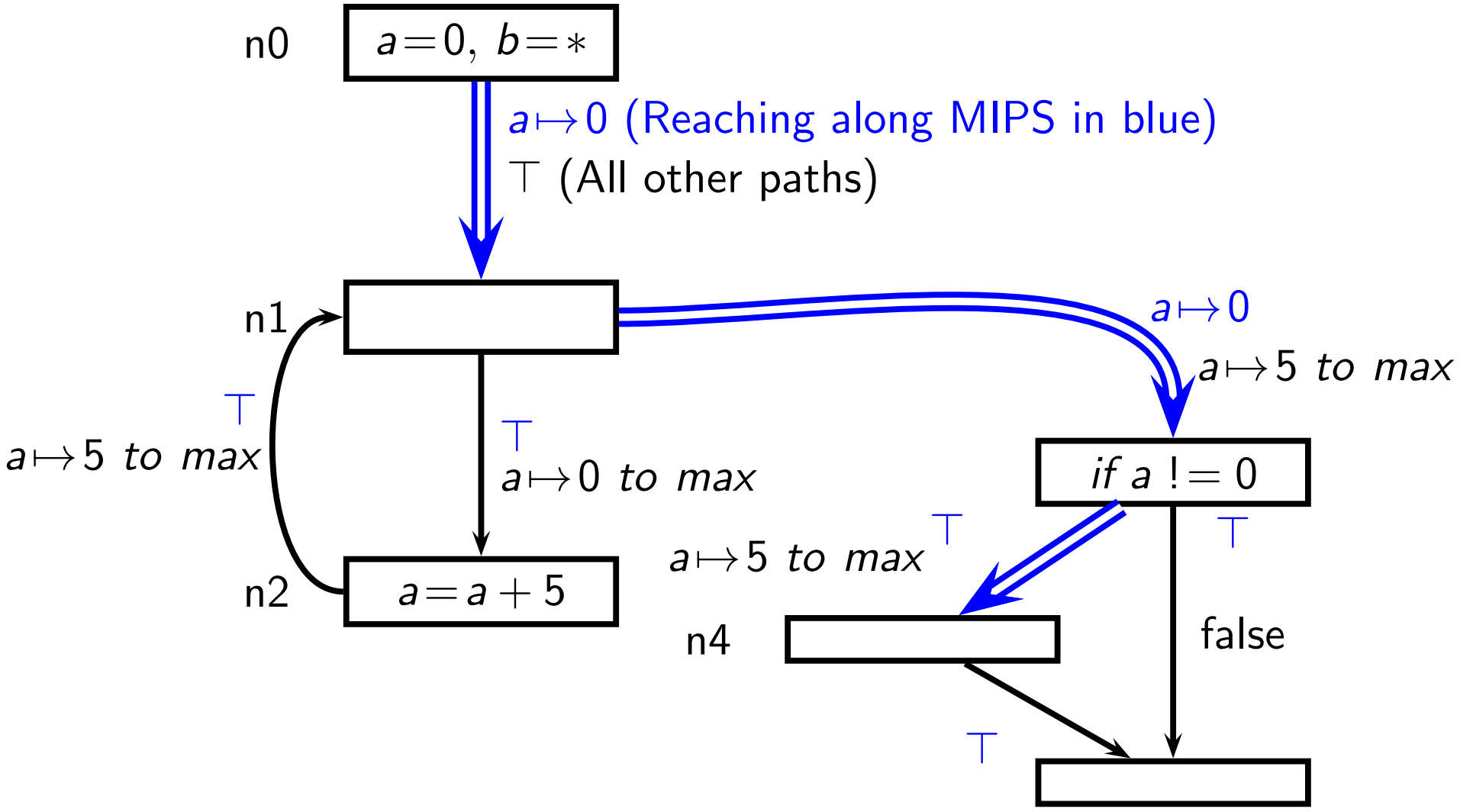
Separating Information Flowing along MIPS



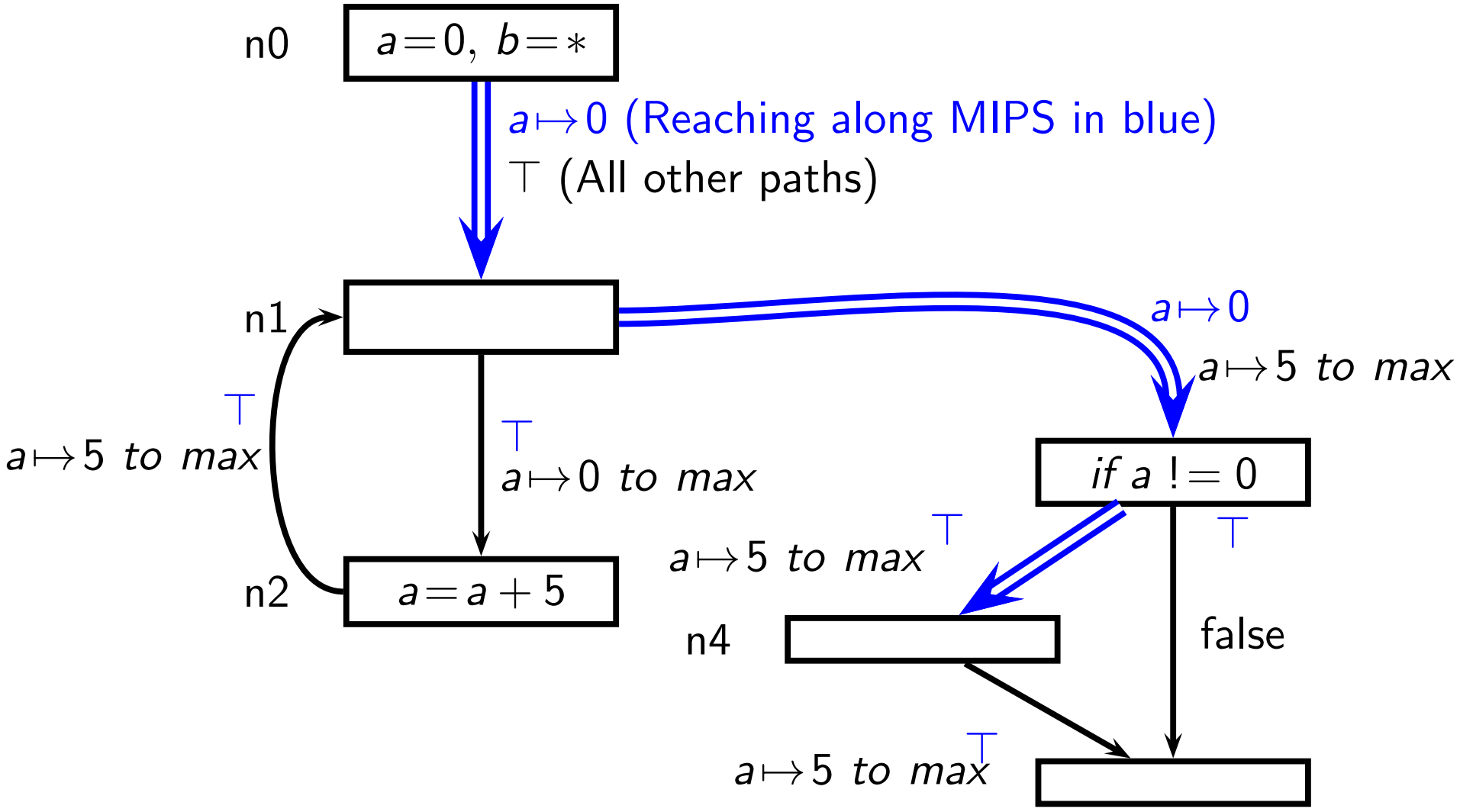
Separating Information Flowing along MIPS



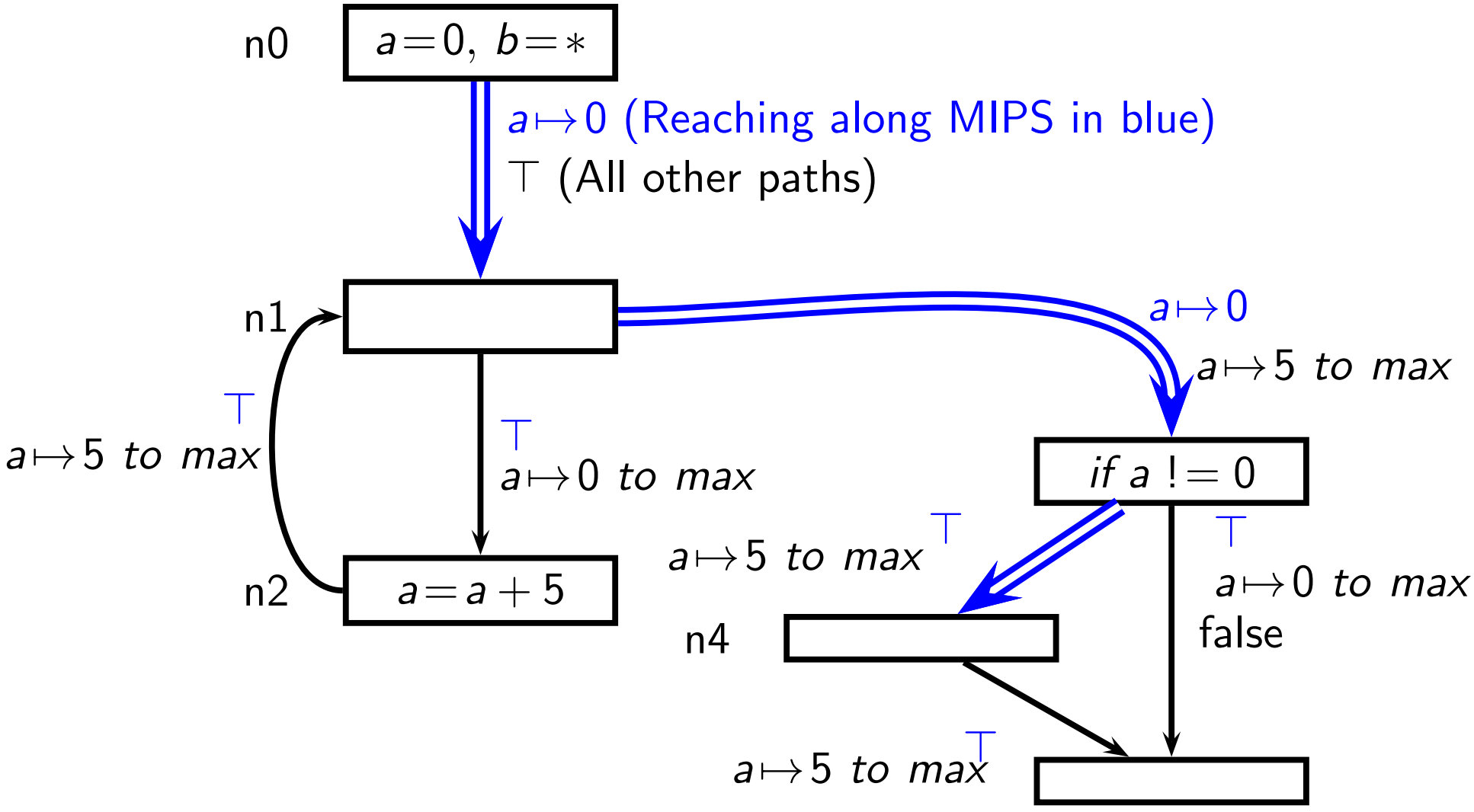
Separating Information Flowing along MIPS



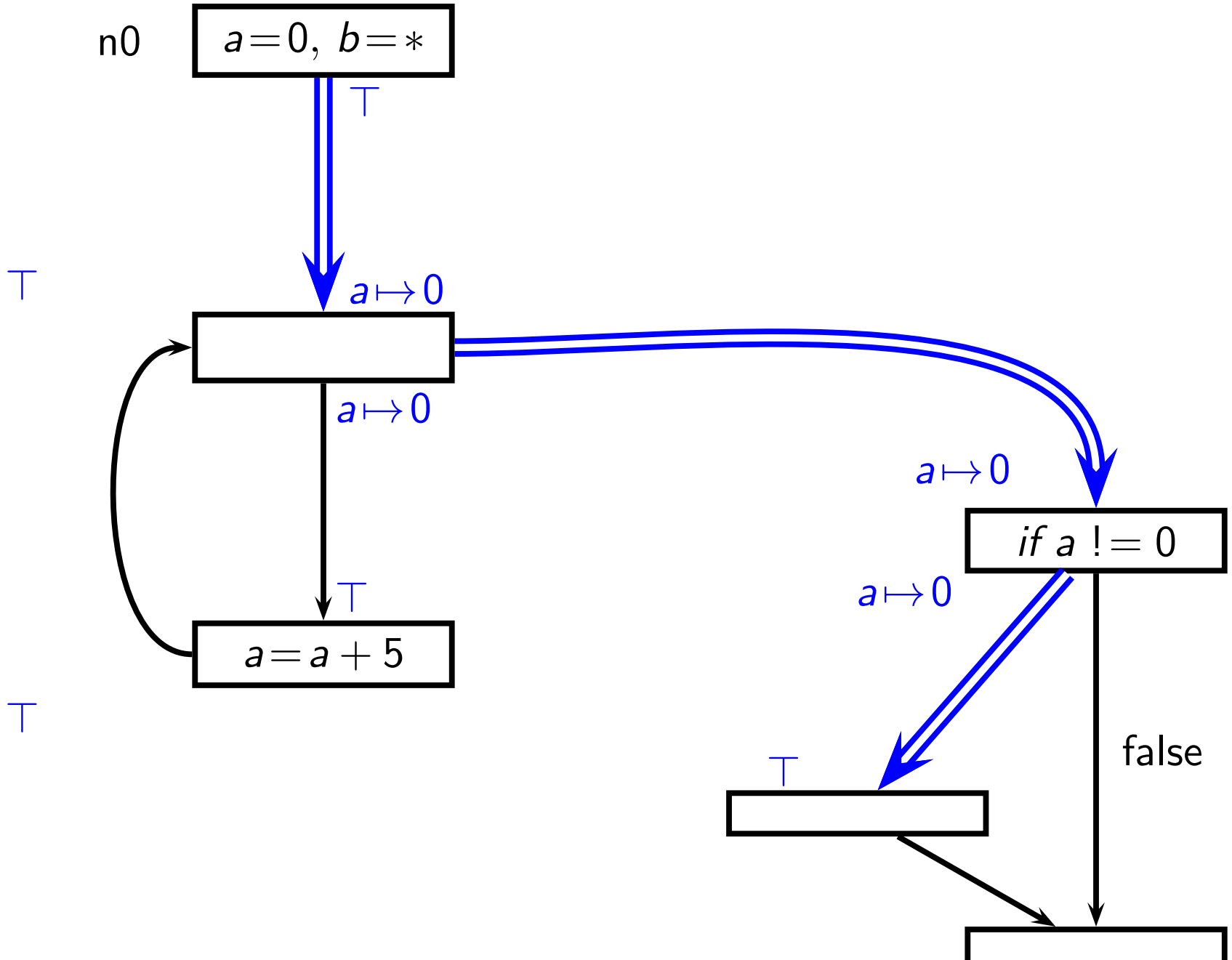
Separating Information Flowing along MIPS



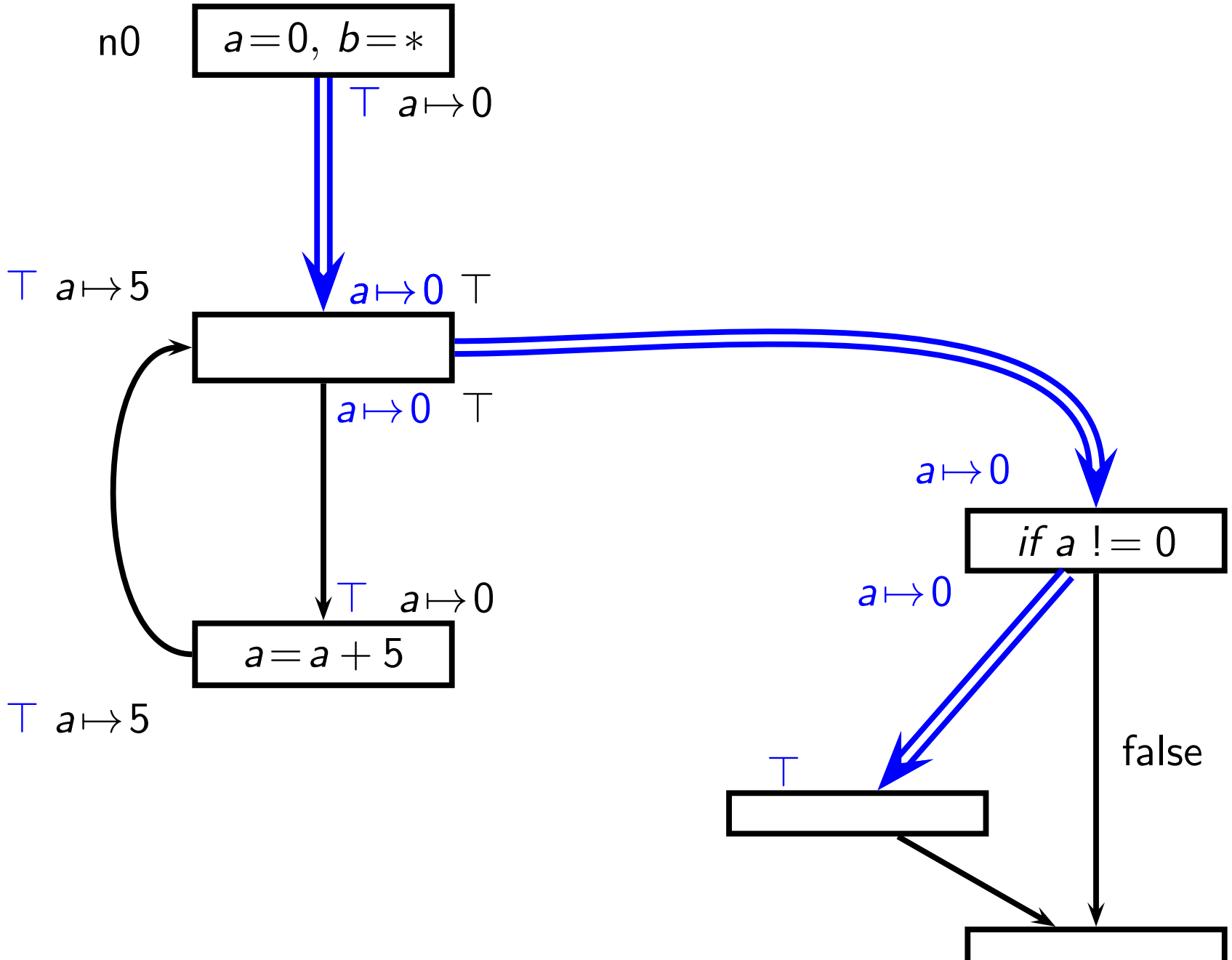
Separating Information Flowing along MIPS



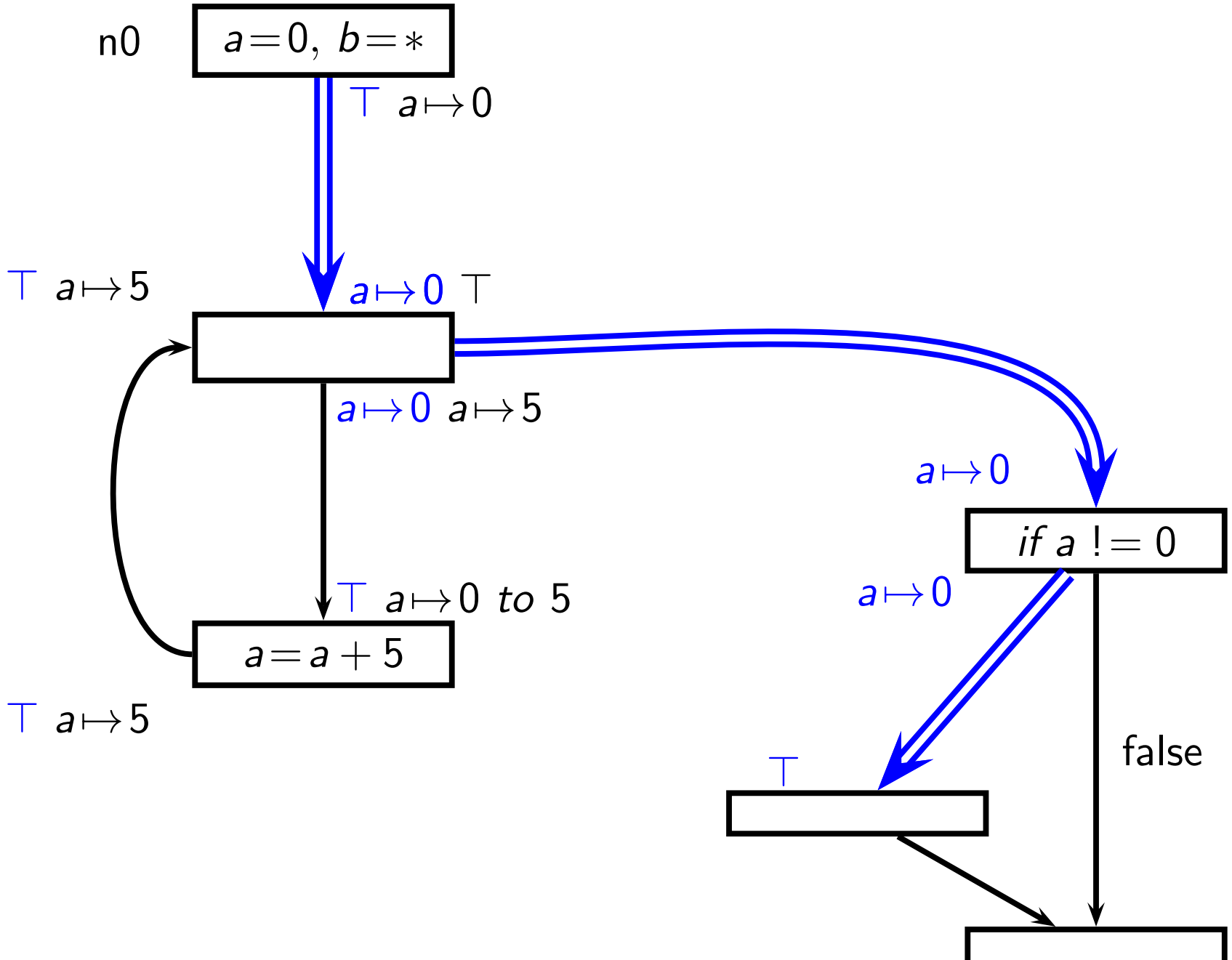
Separating Information Flowing along MIPS



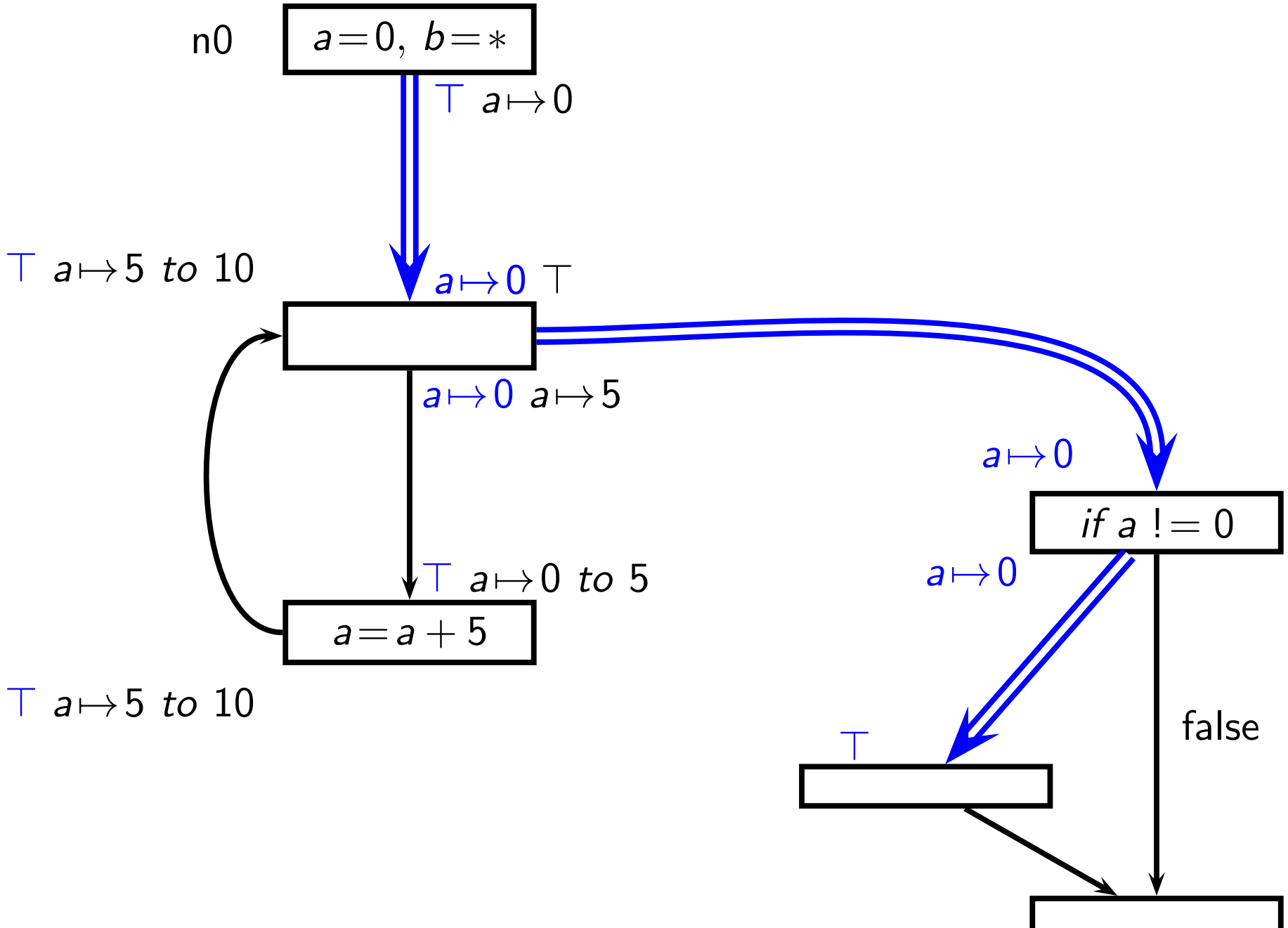
Separating Information Flowing along MIPS



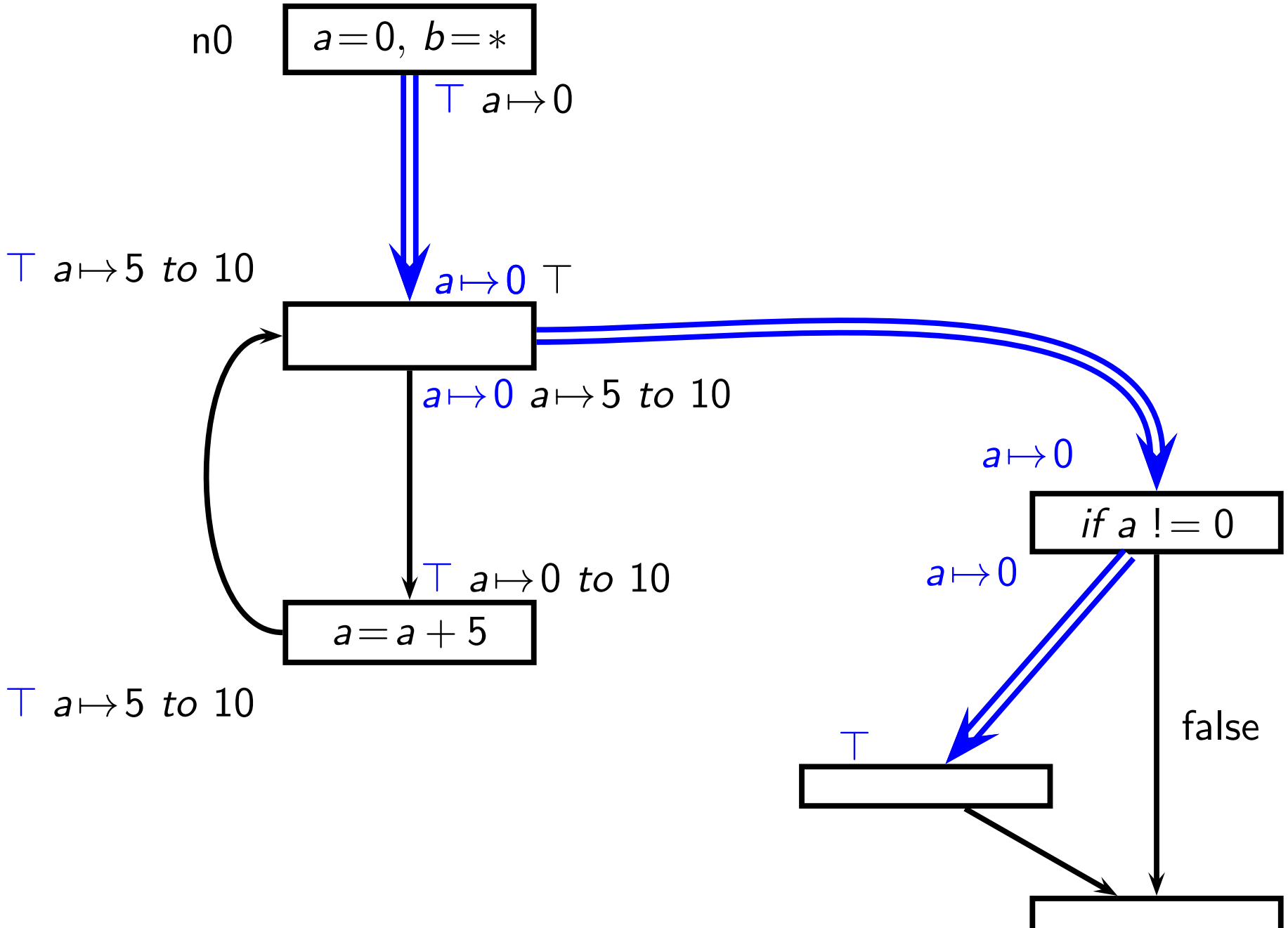
Separating Information Flowing along MIPS



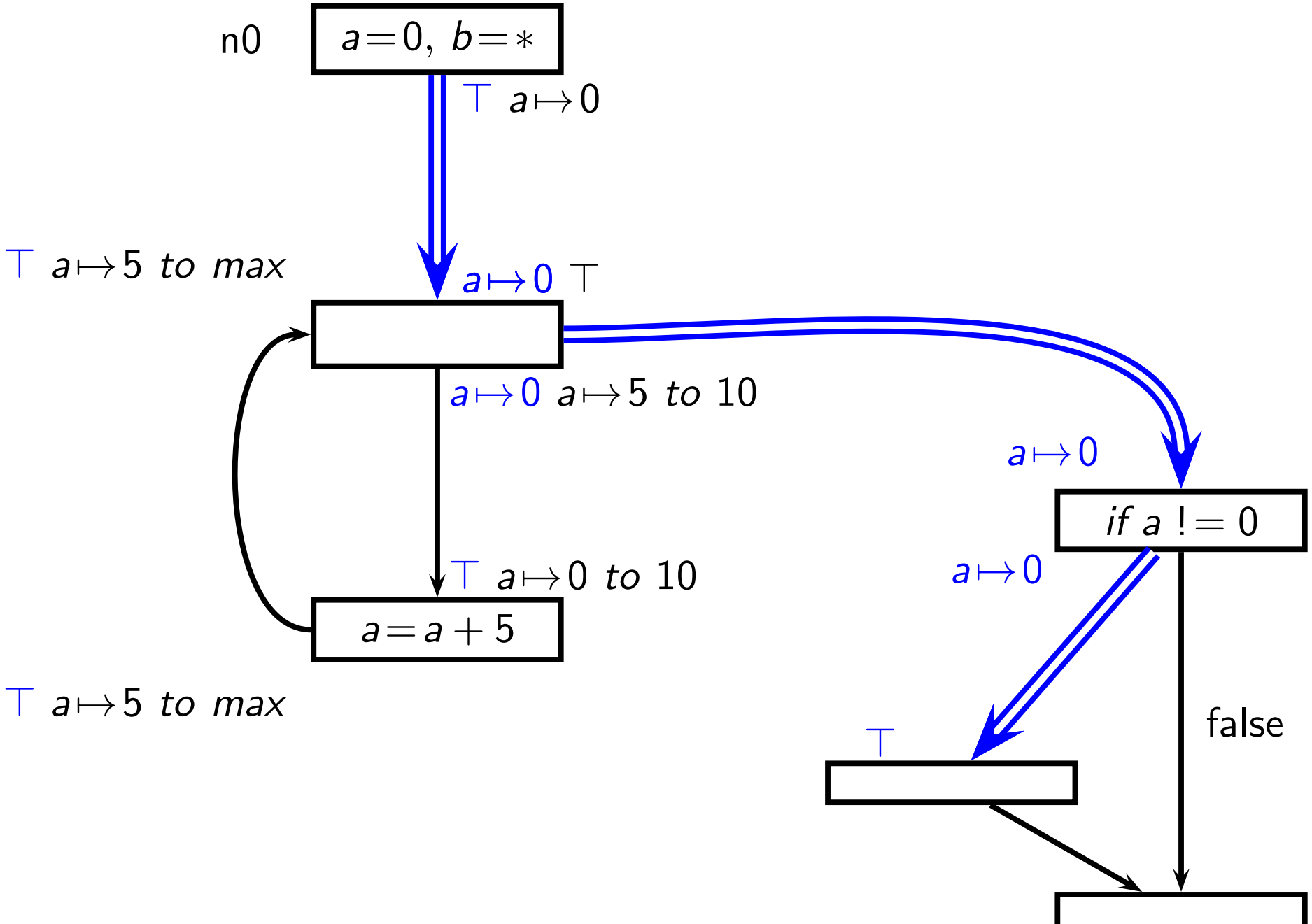
Separating Information Flowing along MIPS



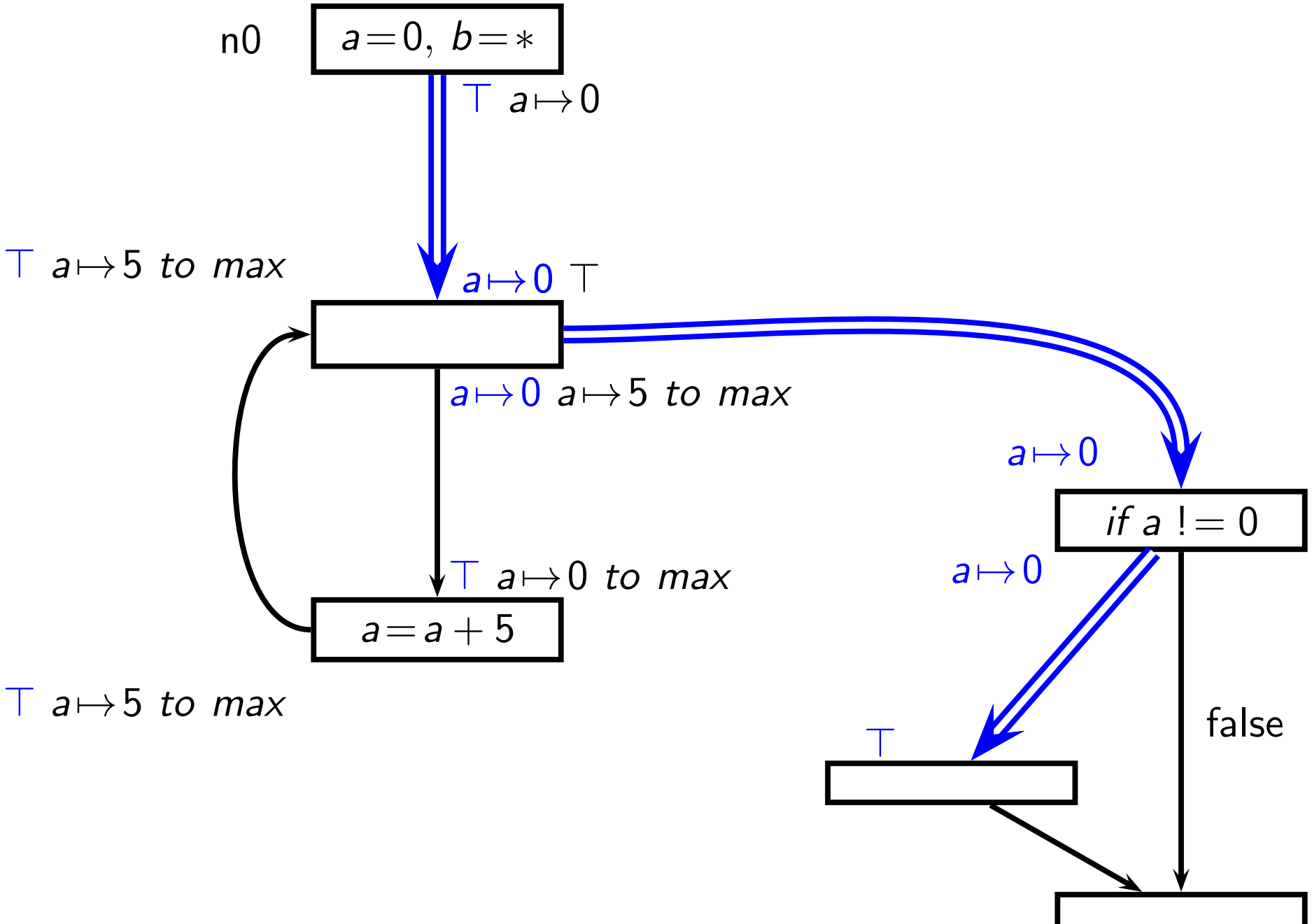
Separating Information Flowing along MIPS



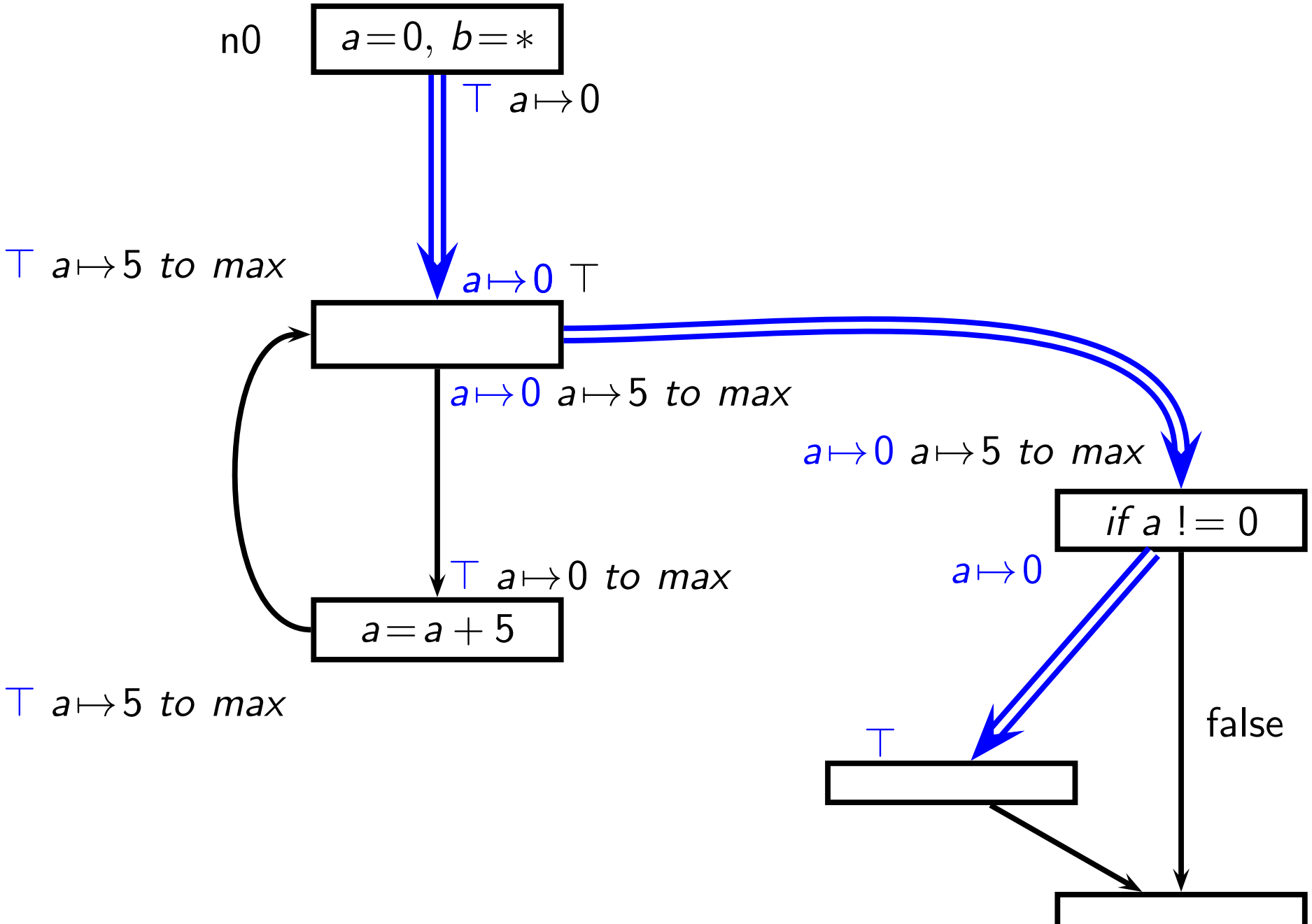
Separating Information Flowing along MIPS



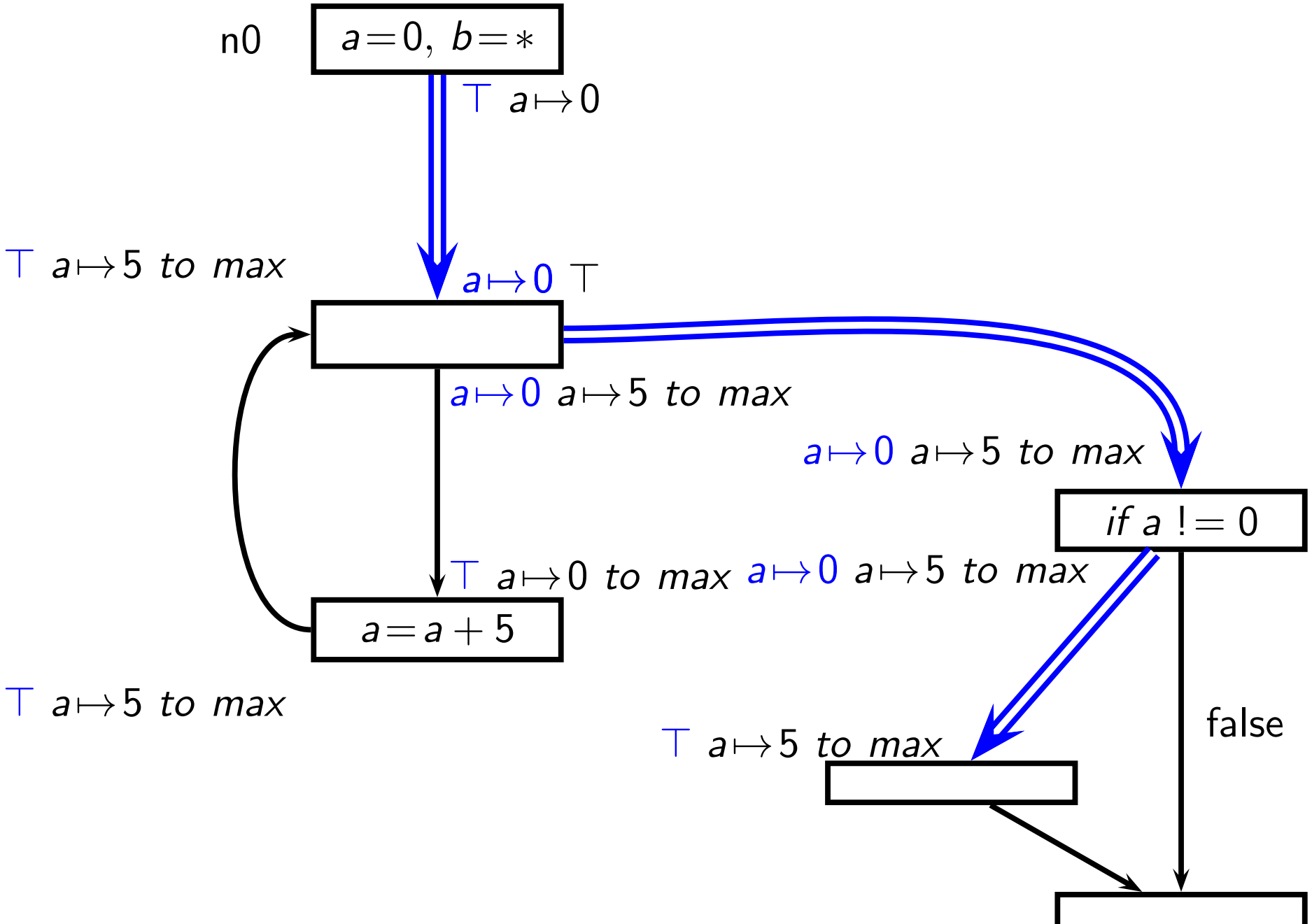
Separating Information Flowing along MIPS



Separating Information Flowing along MIPS



Separating Information Flowing along MIPS



Exchange of Information

- Information along each MIPS is kept in a separate bucket
- An edge flow function distributes the information depending upon whether
  - edge is a part of MIPS, or
  - edge is not a part of MIPS

Generalization to Program Containing k MIPS

At Node n

- Original Value: D
- New Values: $< D_0, D_1, ..., D_k >$

$$D \sqsubseteq (D_0 \sqcap D_1 \sqcap \ldots \sqcap D_k)$$

where

D_0 is information not flowing through any MIPS

For other i, D_i is information flowing through i^{th} MIPS

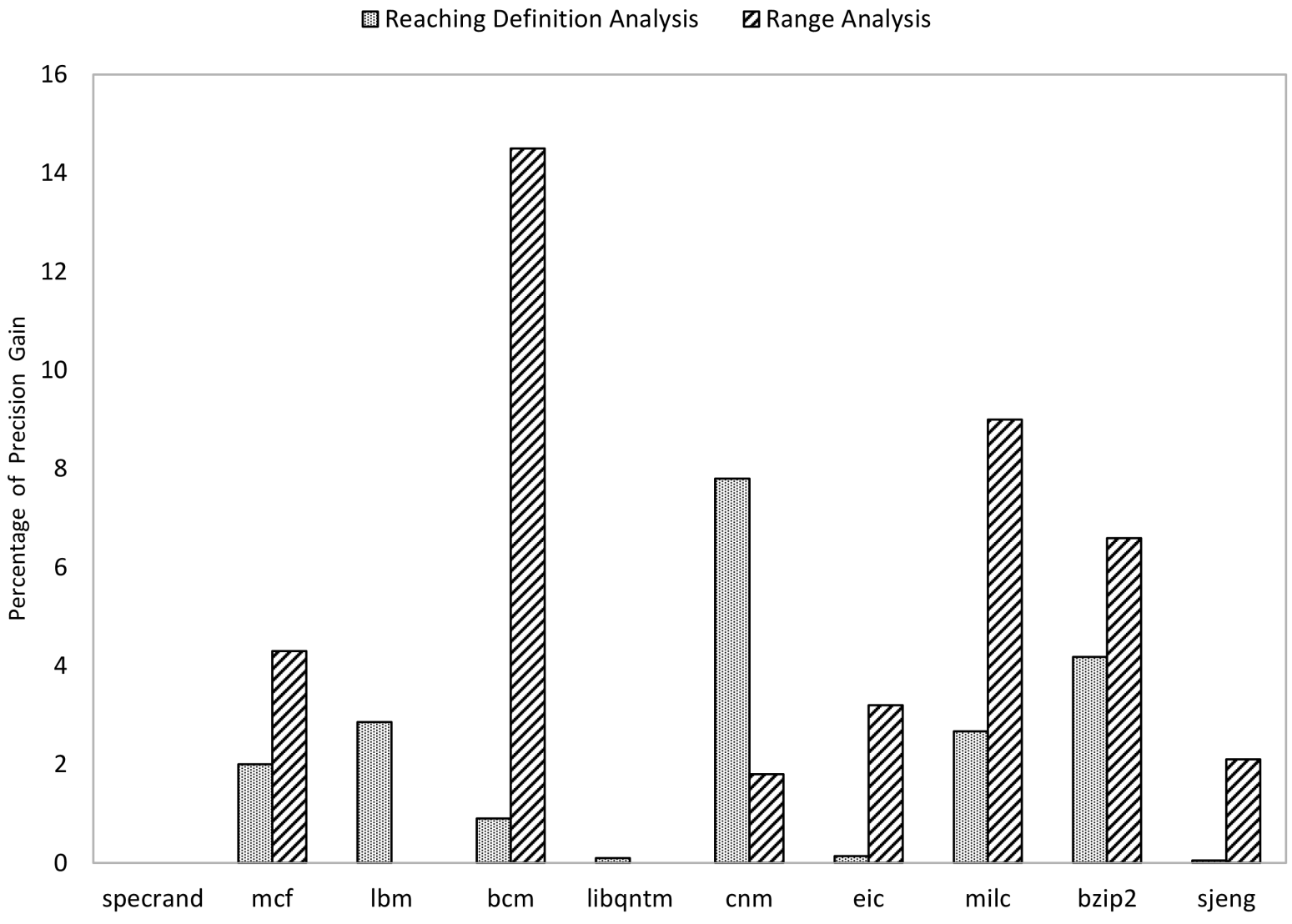
Complexity

- Number of MIPS: k (end disjoint MIPS)
- MFP complexity: N^2, N=Nodes in CFG
- Our complexity: k * N^2

  (Practically, the analysis is much more efficient.)

Experimental Evaluation

Results


Reaching Definition Analysis
Range Analysis
Percentage of Precision Gain
16
14
12
10
8
6
4
2
0
specrand
mcf
lbm
bcm
libqntm
cnm
eic
milc
bzip2
sjeng

Other Results

- Def-Use Pairs: upto 3% reduction
- Dead Code: 14, 124 and 216 program statements on different benchmarks.
- No improvement was found in *division by zero* property verification.

Performance

- Time: upto 3 times increase
- Memory: upto 4 times increase

Related Work

Path Sensitivity in Data Flow

- Control Flow Graph Restructuring [Bodik et.al, 1999]
  - Removes MIPS from control flow graph
  - Causes exponential blow up in graph size whereas we do not
- Trace Partitioning
  - Partitions program traces in equivalence classes using control flow criteria
  - We combine the trace partitioning and MIPS.

Future Work

- Interprocedural Partially Path Sensitive MFP.

Thank You

Questions?