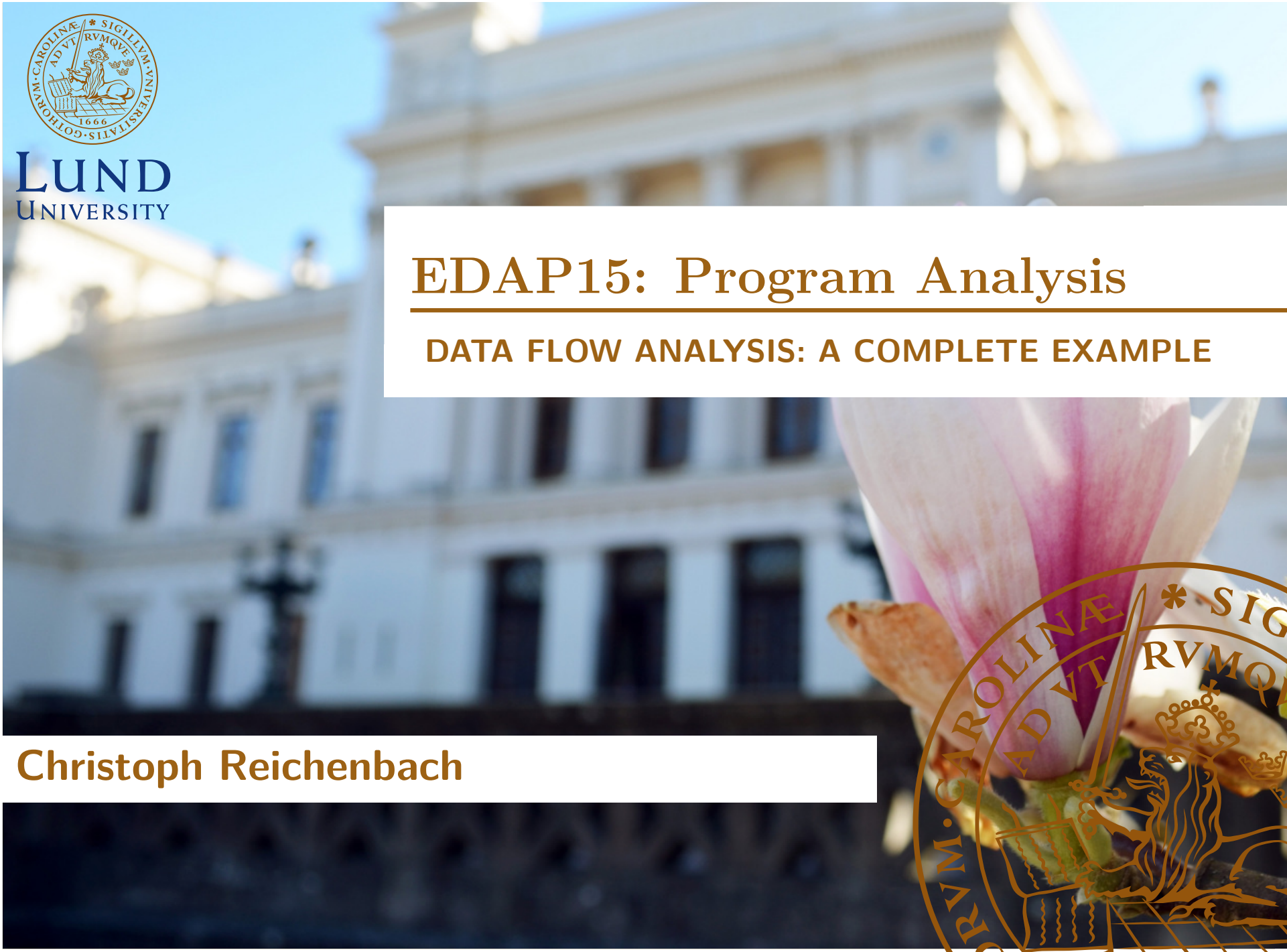


LUND UNIVERSITY

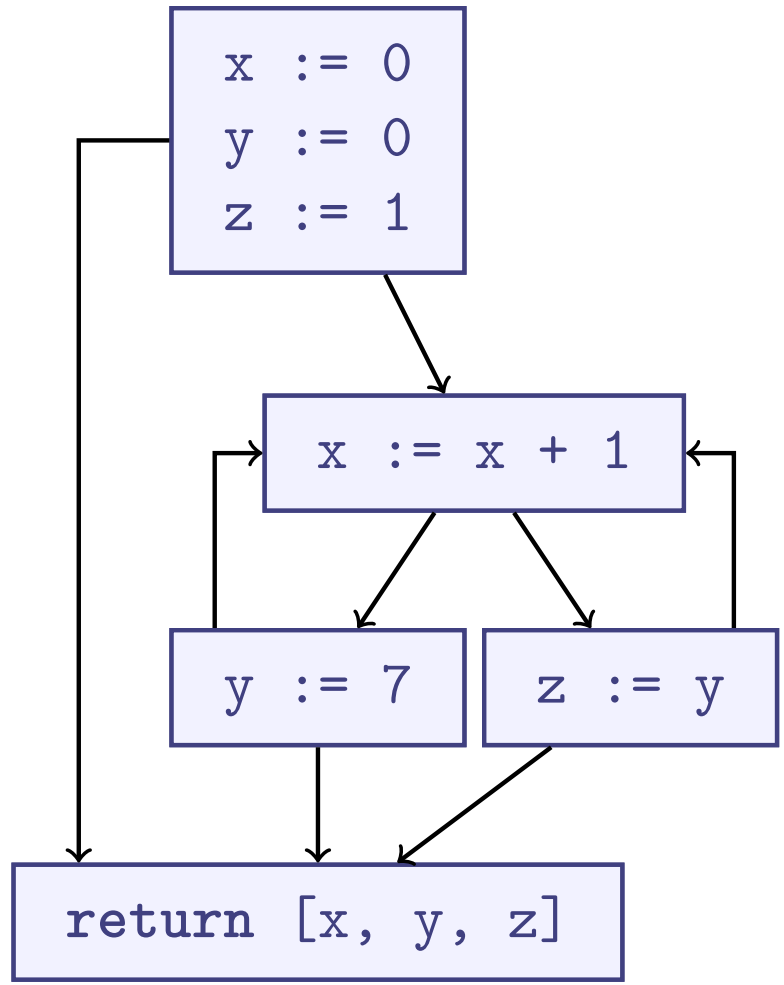
EDAP15: Program Analysis

DATA FLOW ANALYSIS: A COMPLETE EXAMPLE

Christoph Reichenbach

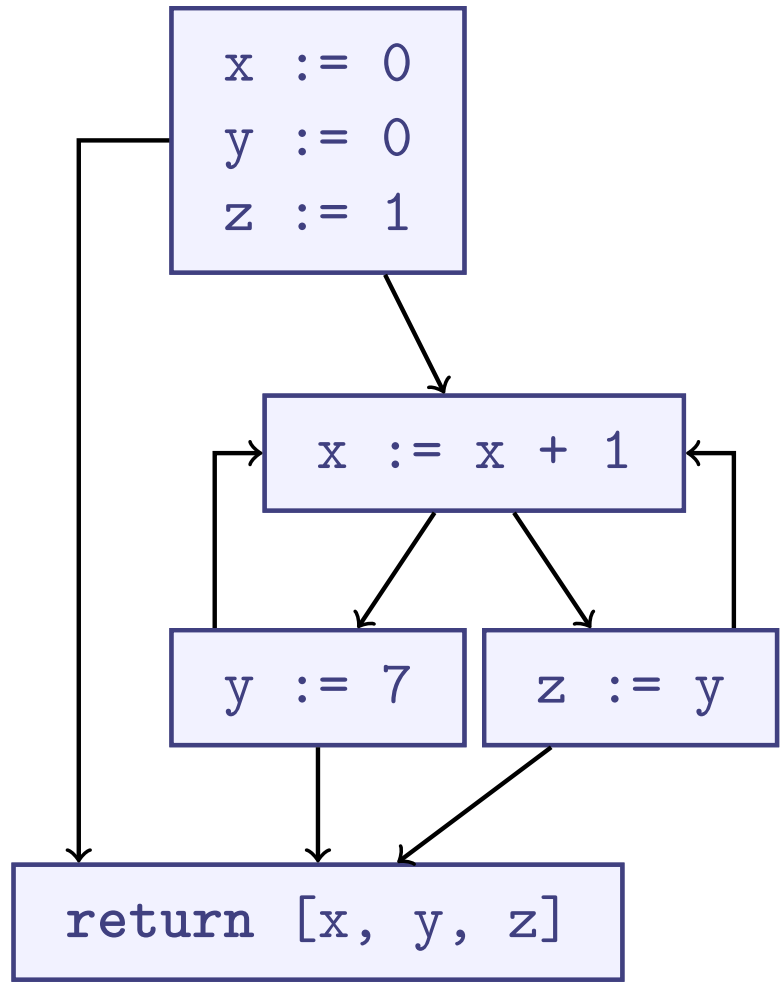
Example: Reaching Definitions

```
var x := 0;
var y := 0;
var z := 1;

while x < 5 {
  x := x + 1;
  if x >= 2 {
    y := 7;
  } else {
    z := y;
} }

return [x, y, z];
```



Example: Reaching Definitions

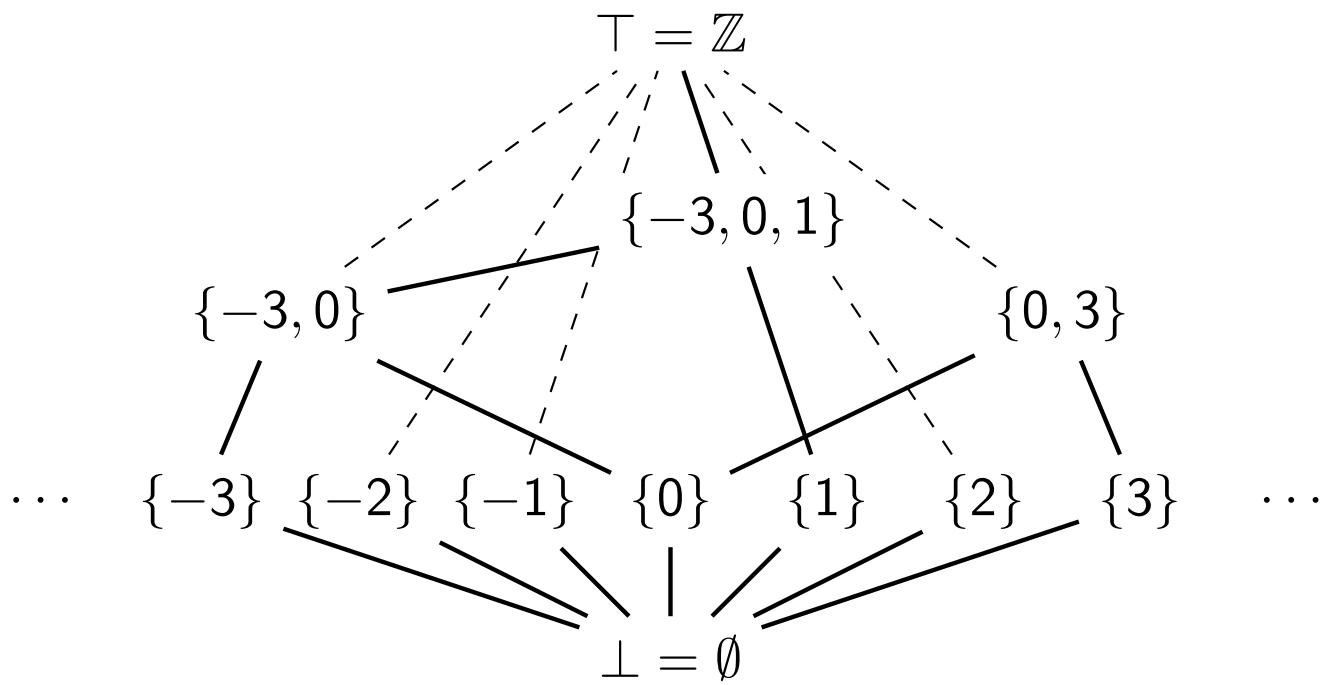
```
var x := 0;
var y := 0;
var z := 1;

while x < 5 {
  x := x + 1;
  if x >= 2 {
    y := 7;
  } else {
    z := y;
} }

return [x, y, z];
```



Reaching Definitions: What values are possible?

Example: Reaching Definitions

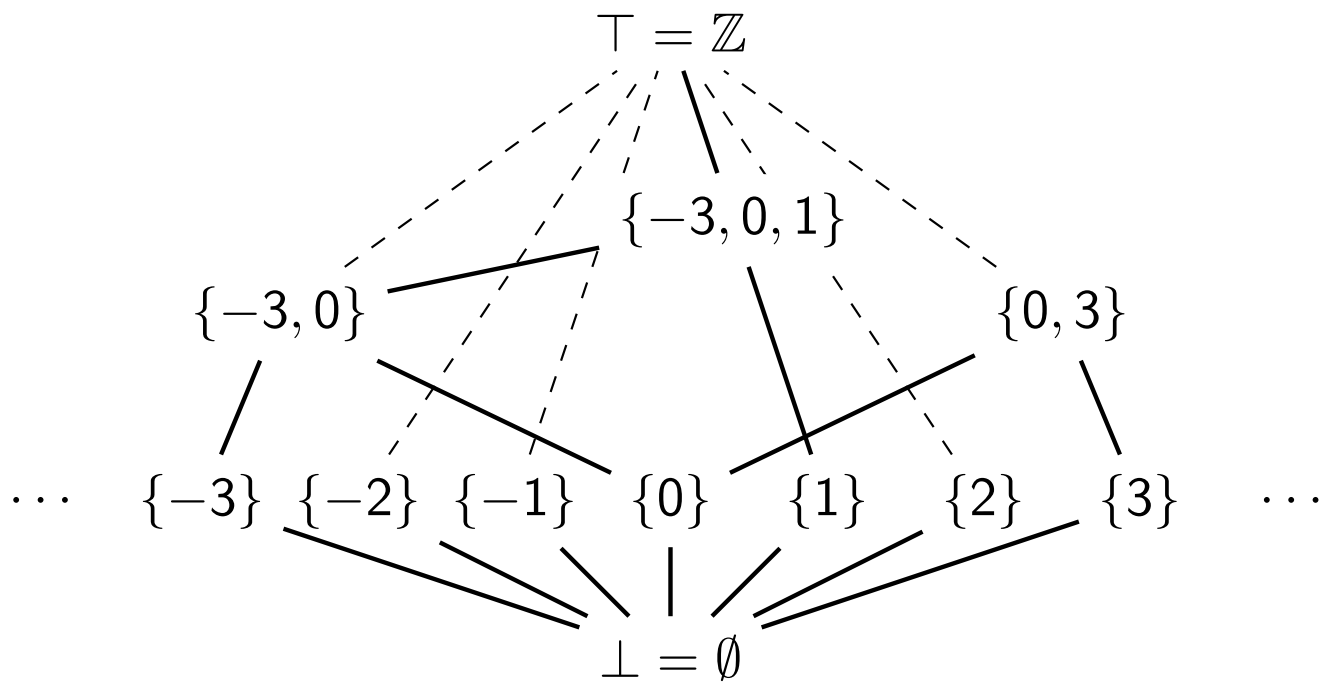
Designing our abstract domain:



- Capture sets of up to 3 possible numbers
- $\top$: More than 3 possible numbers
- $\bot$: $\emptyset$ (no possible numbers seen yet)

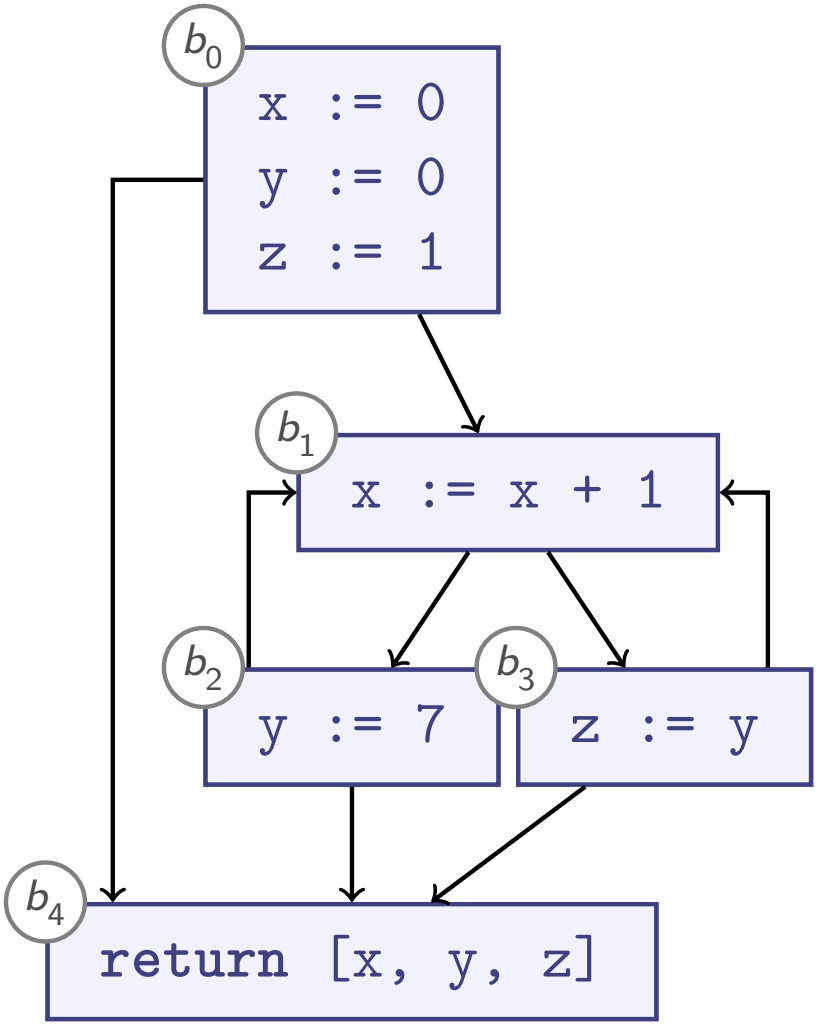
Example: Reaching Definitions

Designing our abstract domain:



- Capture sets of up to 3 possible numbers
- $\top$: More than 3 possible numbers
- $\bot$: $\emptyset$ (no possible numbers seen yet)
- Infinitely many elements, but finite height!

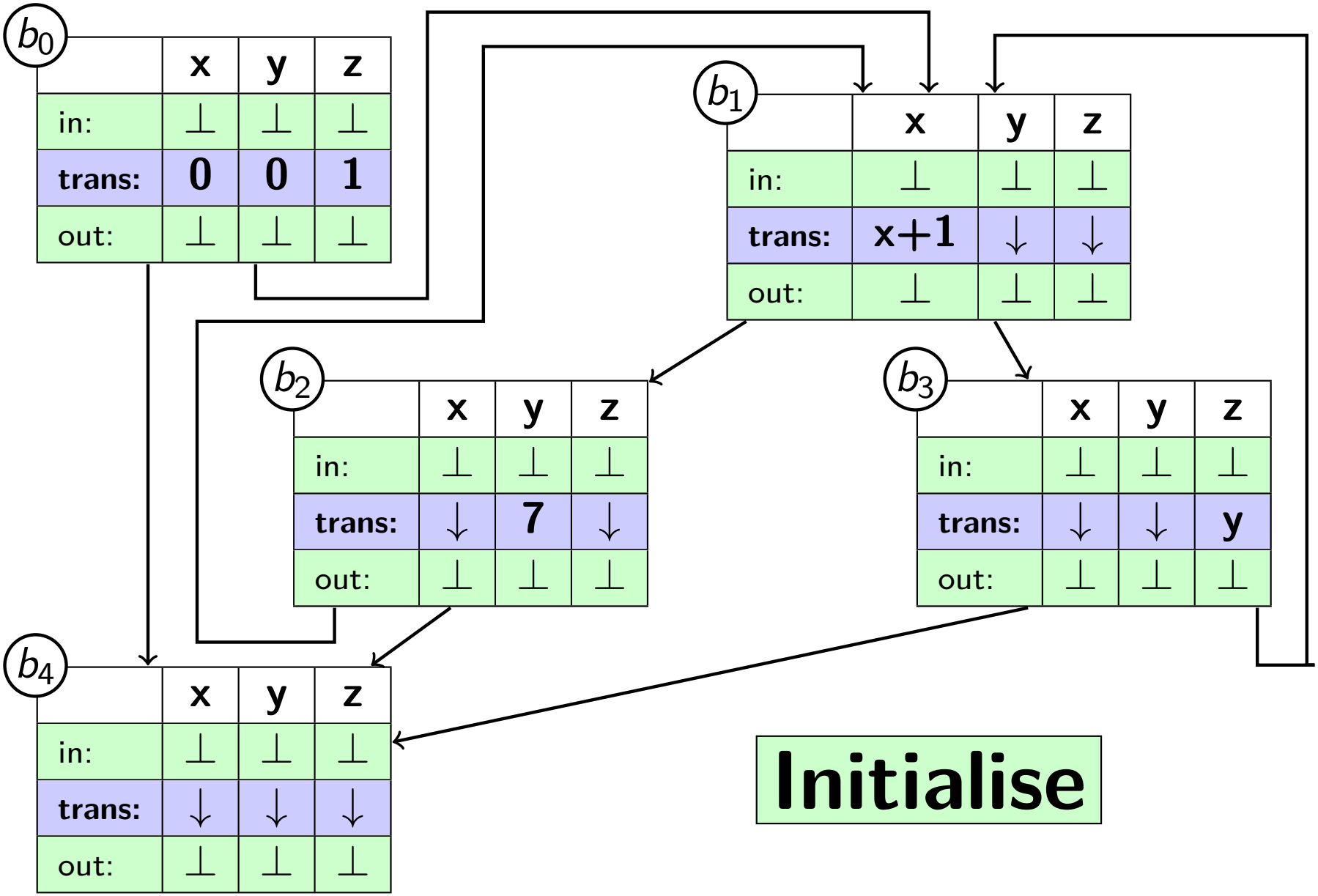
Example: Control-Flow Graph



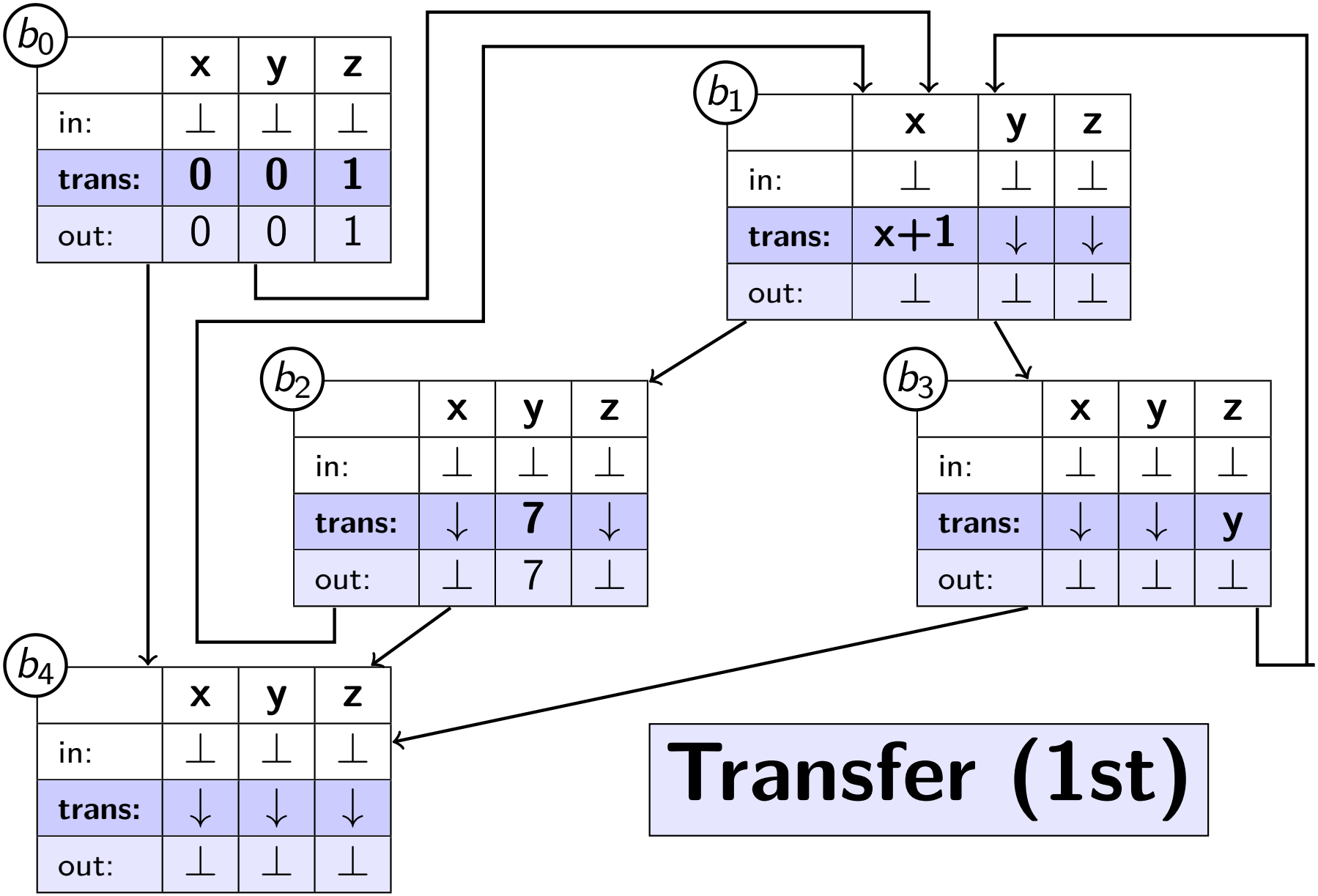
b	inputs	$trans_b$		
		x	y	z
b_0	$\emptyset$	0	0	1
b_1	$\{b_0, b_2, b_3\}$	$x+1$	y	z
b_2	$\{b_1\}$	x	7	z
b_3	$\{b_1\}$	x	y	y
b_4	$\{b_0, b_2, b_3\}$	x	y	z

$$join_b = \text{let} \quad j = \bigcup_{s \in \text{inputs}_b} s \quad \text{in} \quad \begin{cases} j & \iff \#j \leq 3 \\ \top & \iff \#j > 3 \end{cases}$$

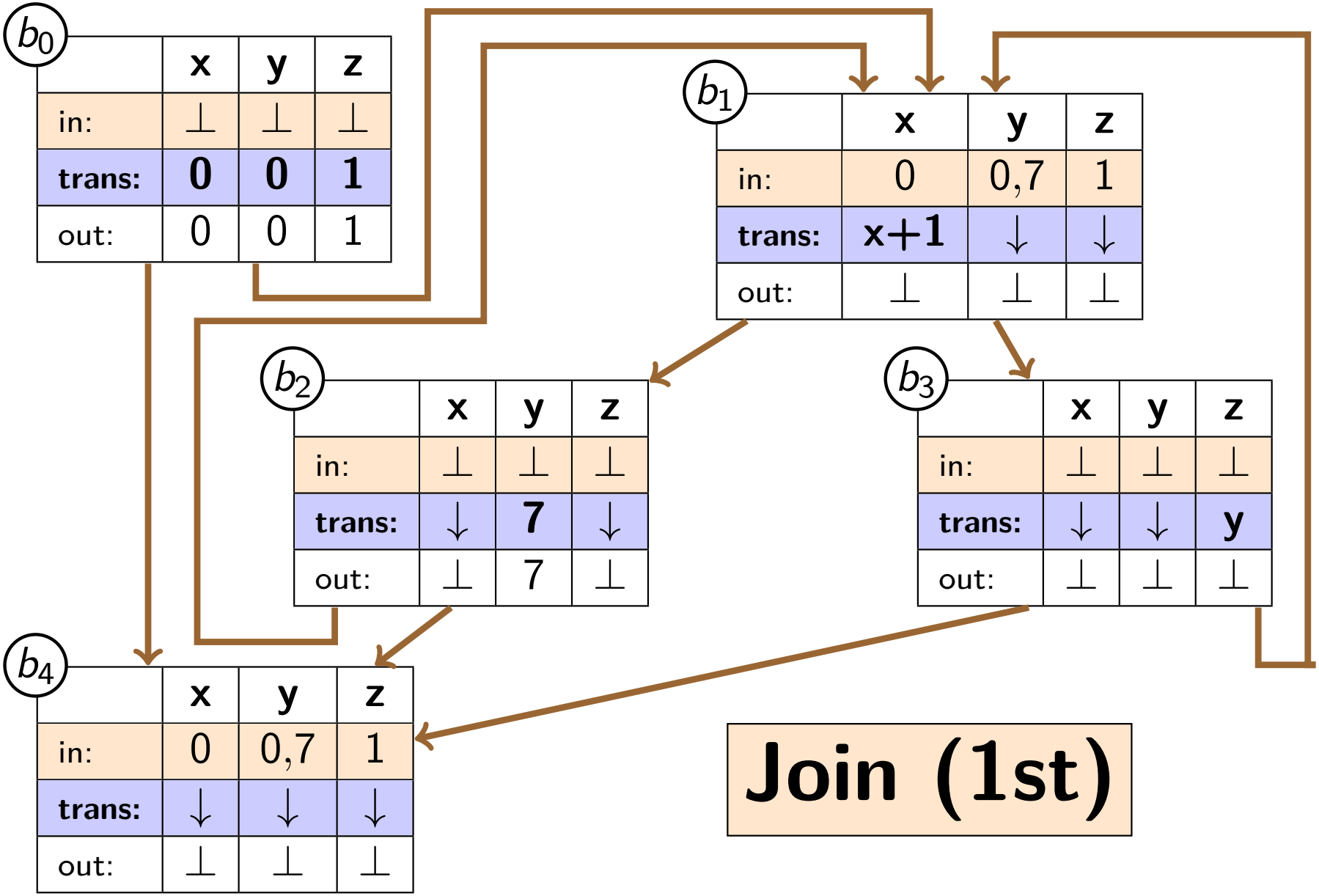
Example: Computing the Fixpoint



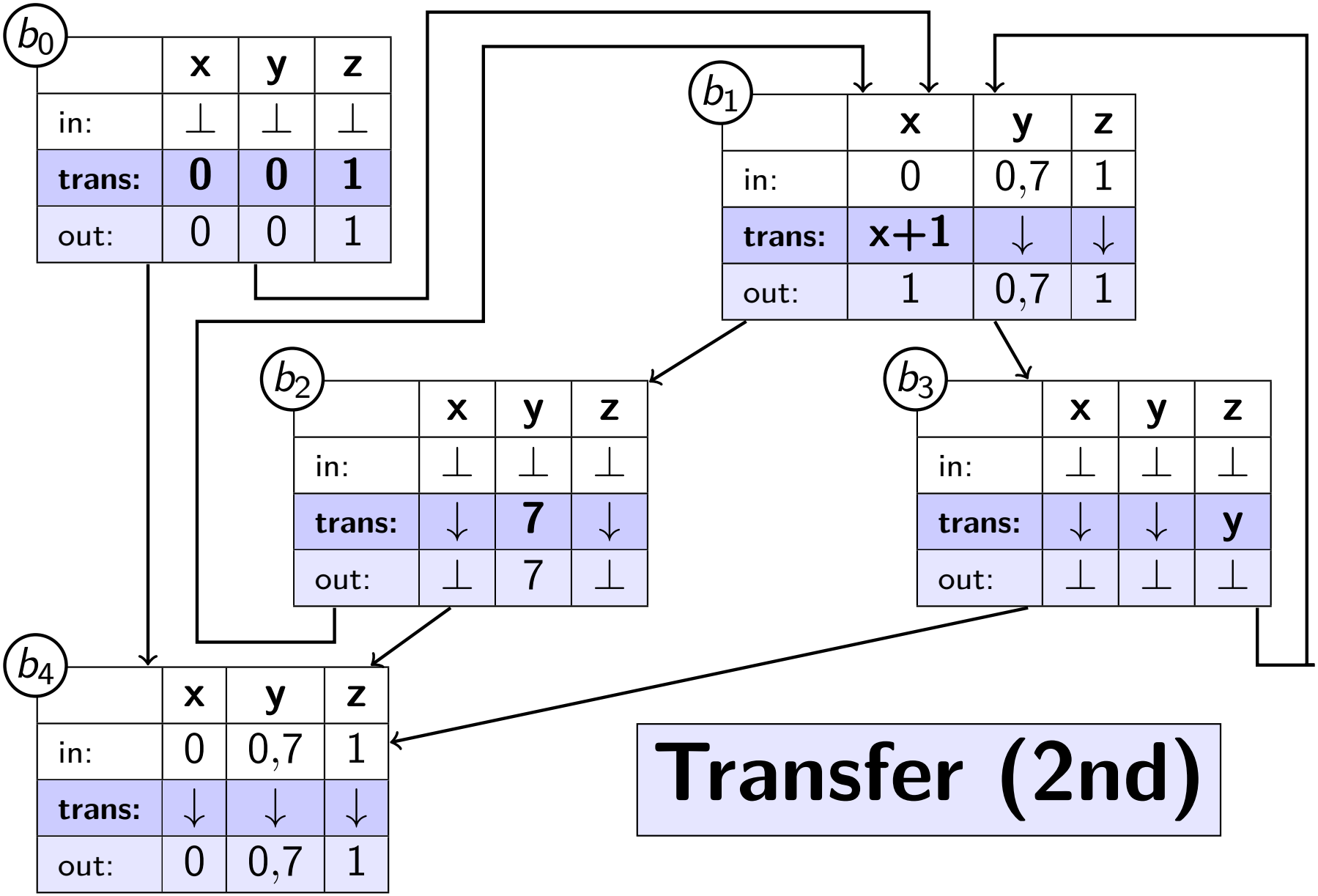
Example: Computing the Fixpoint



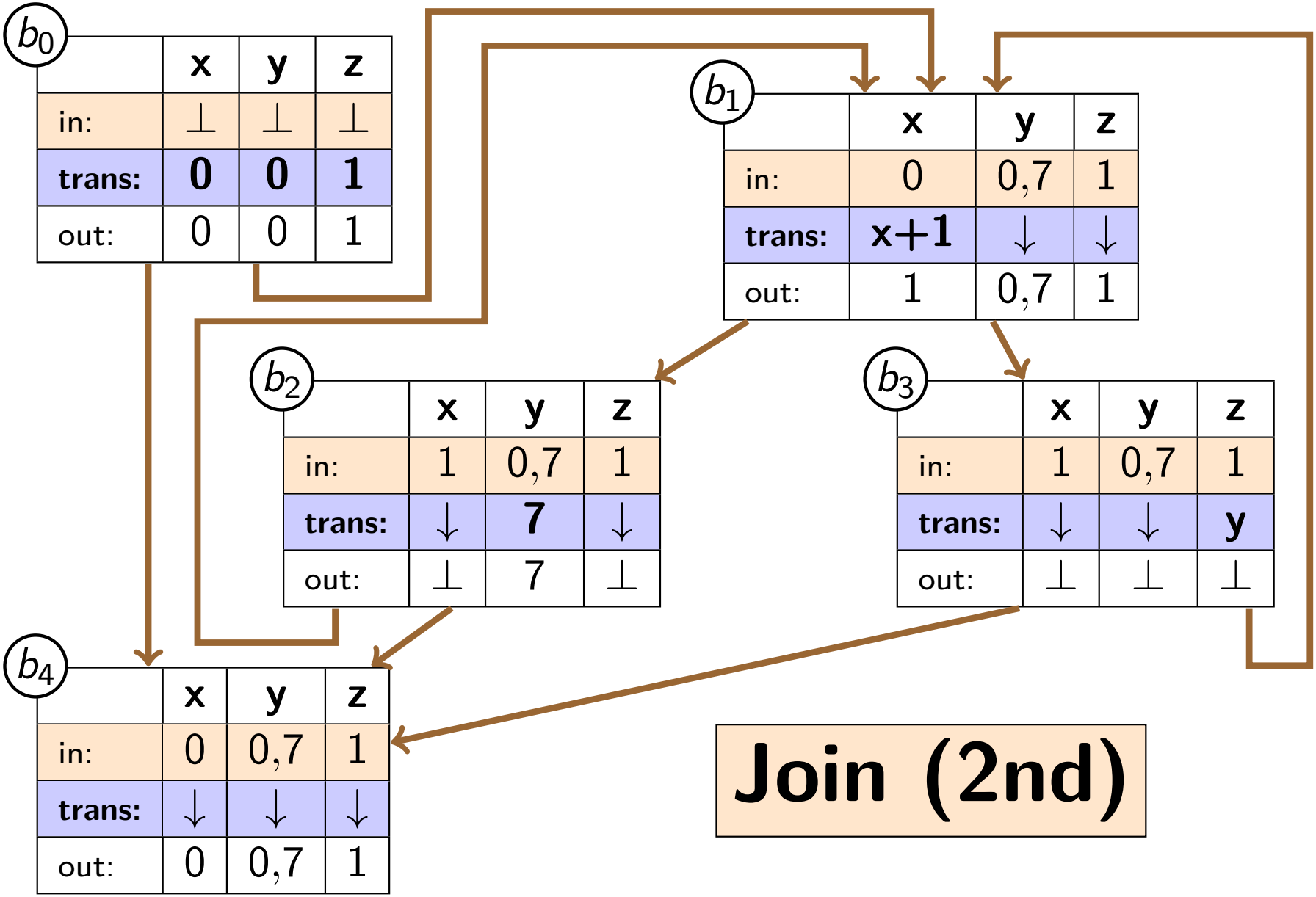
Example: Computing the Fixpoint



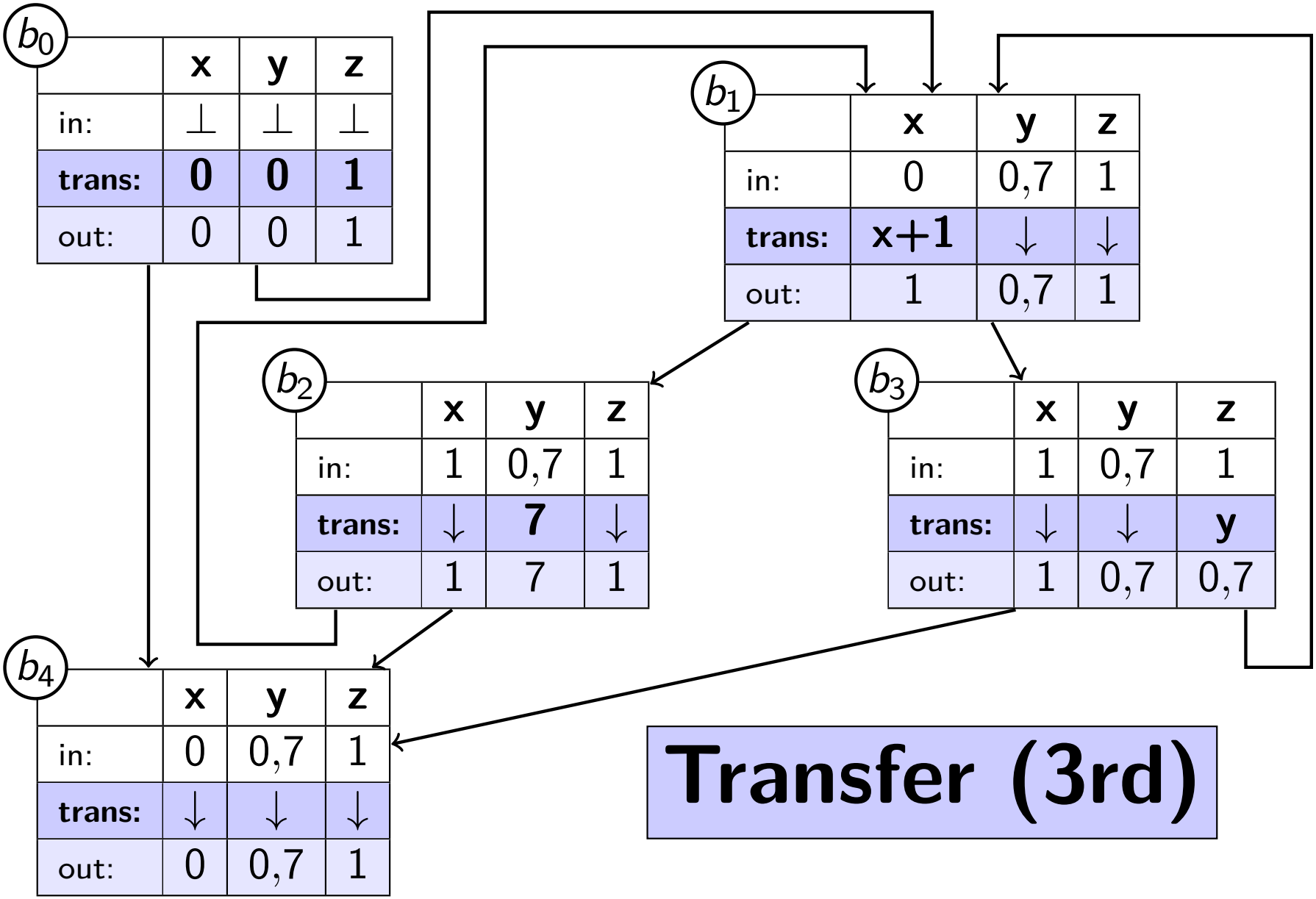
Example: Computing the Fixpoint



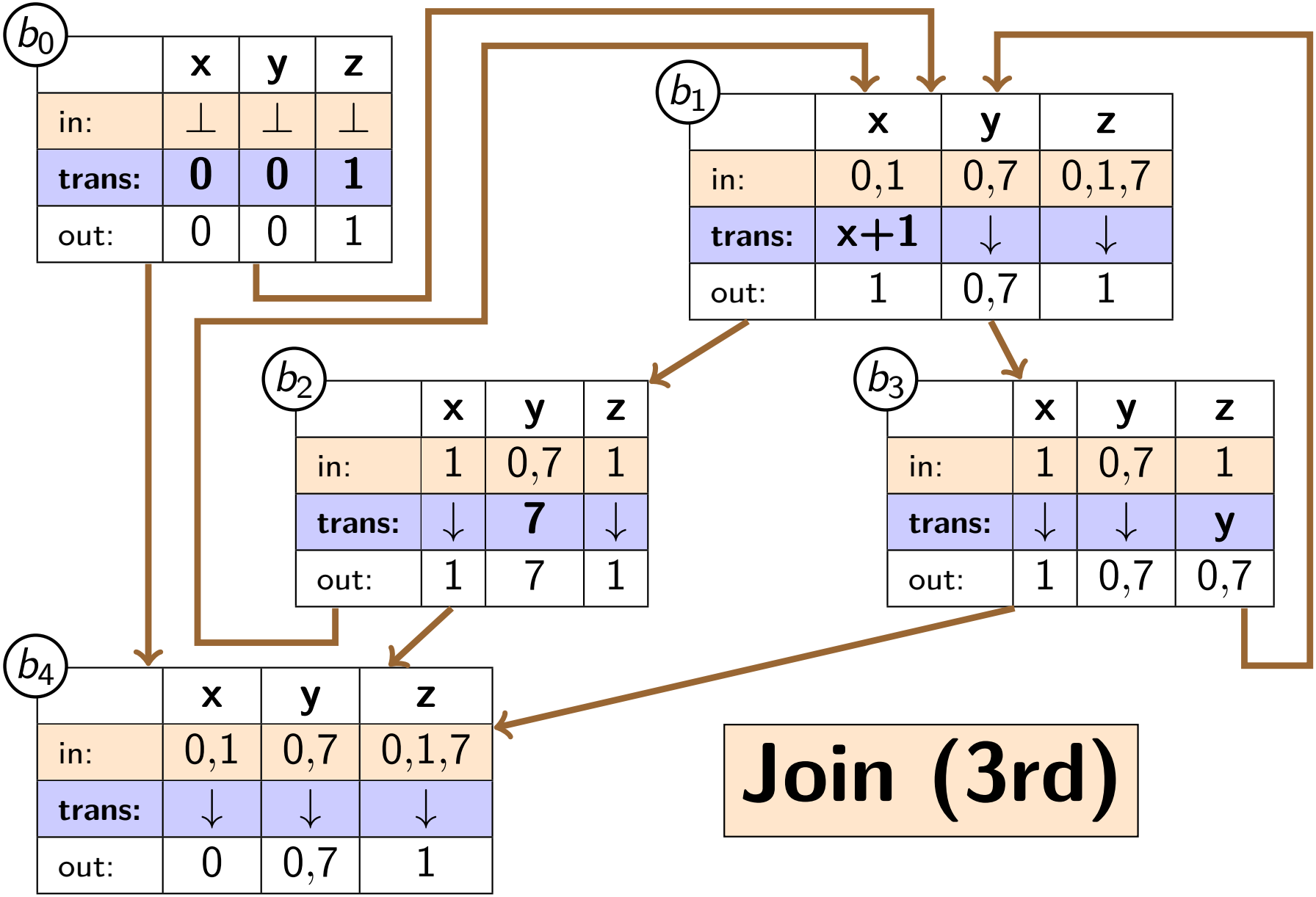
Example: Computing the Fixpoint



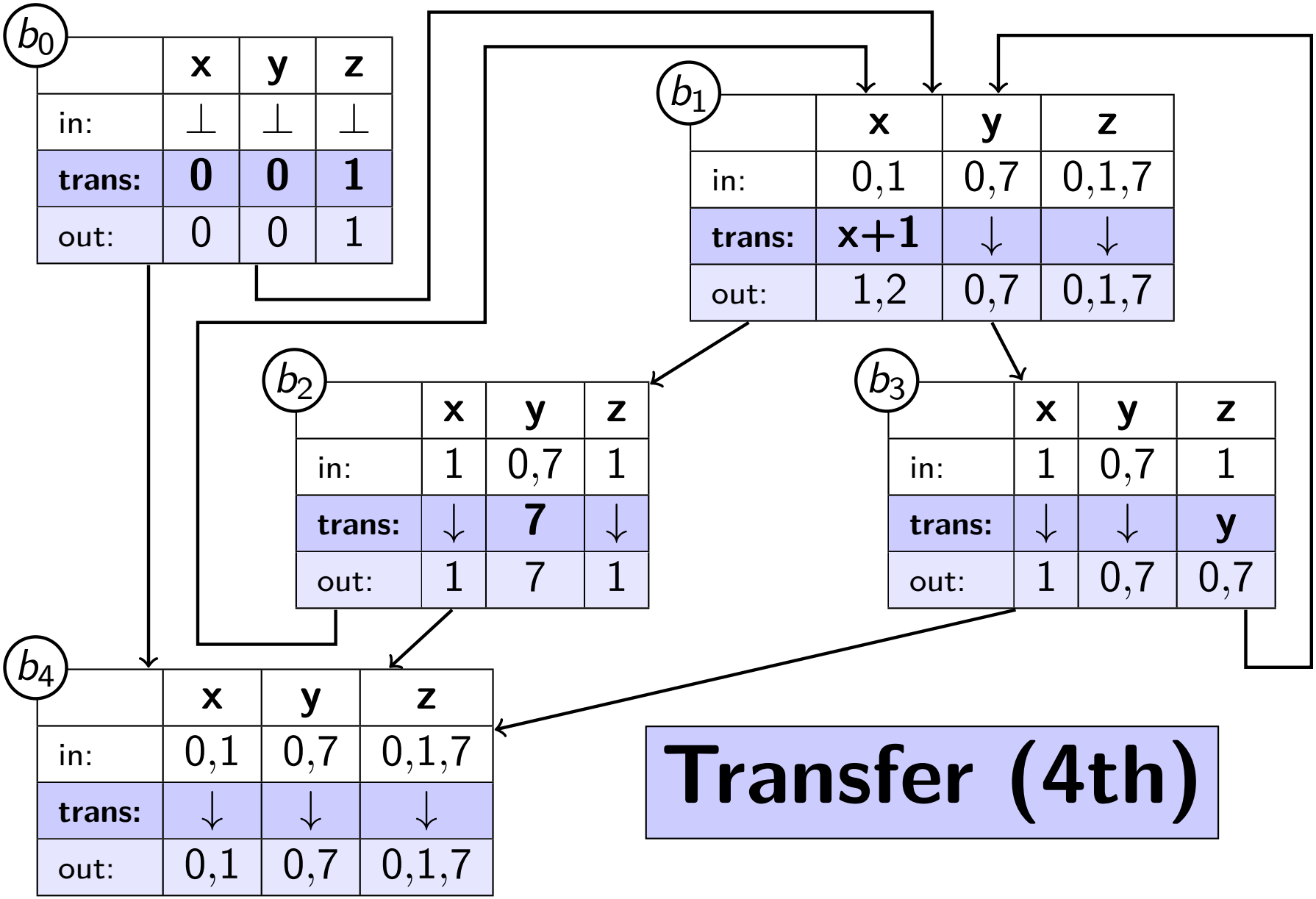
Example: Computing the Fixpoint



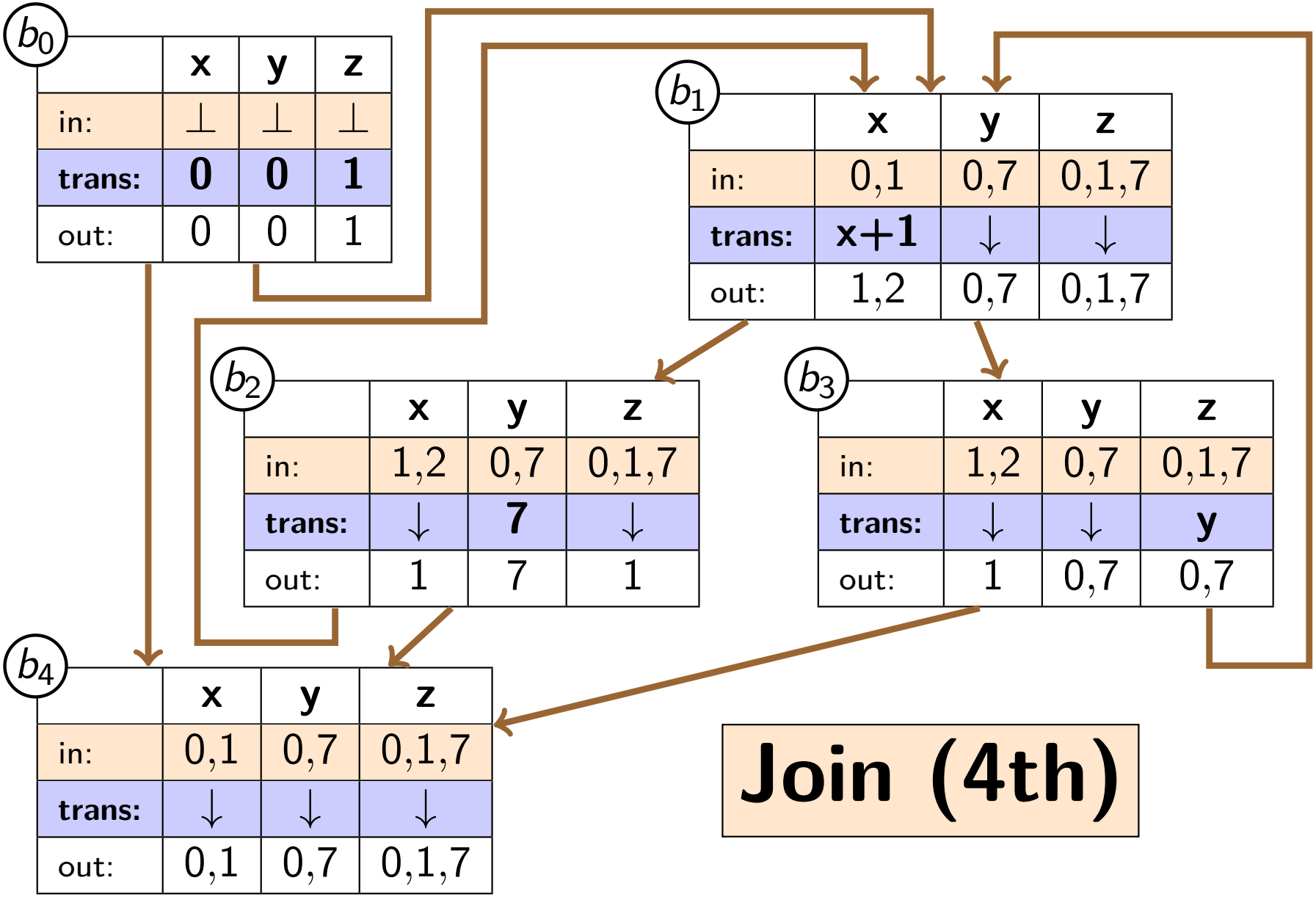
Example: Computing the Fixpoint



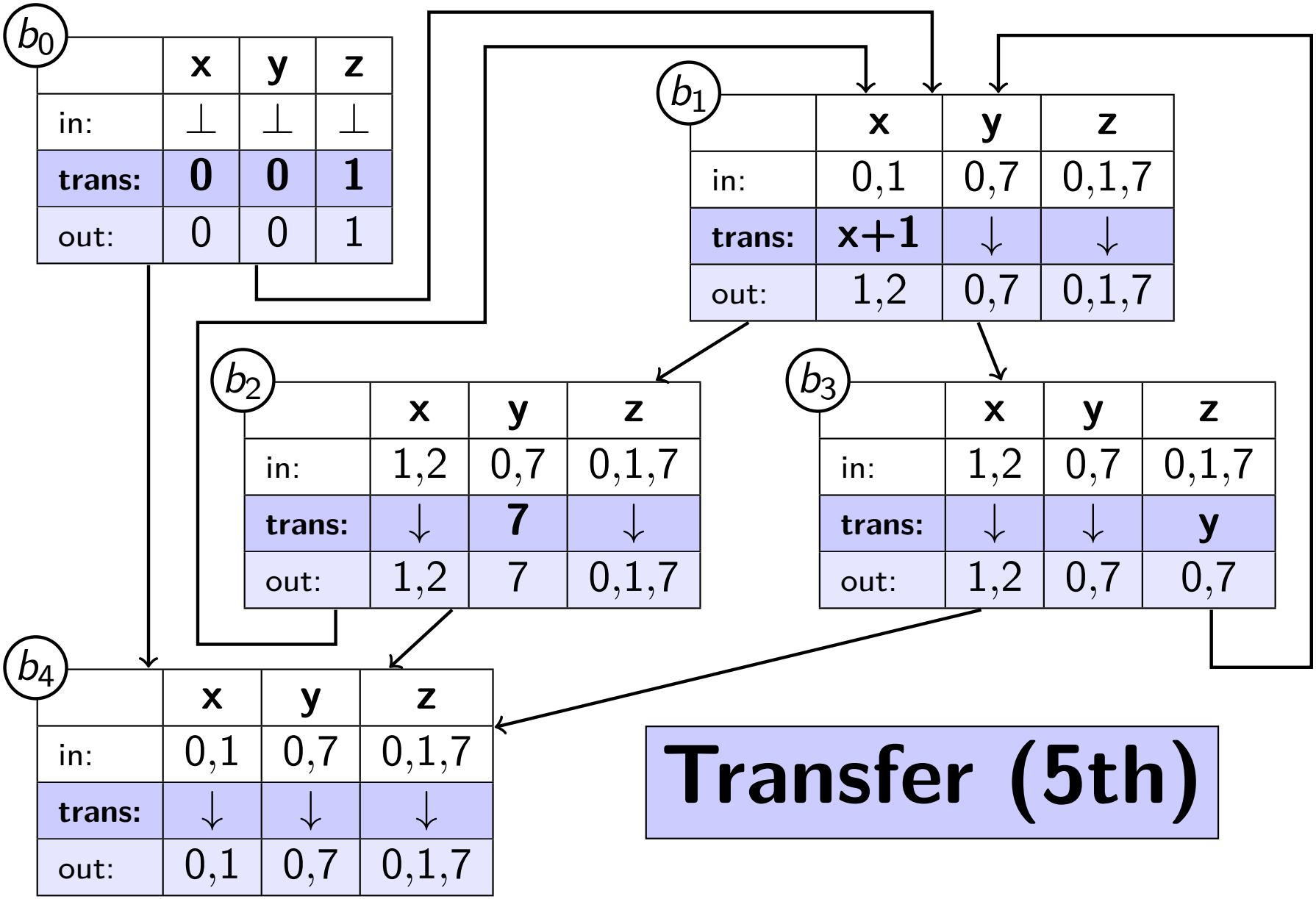
Example: Computing the Fixpoint



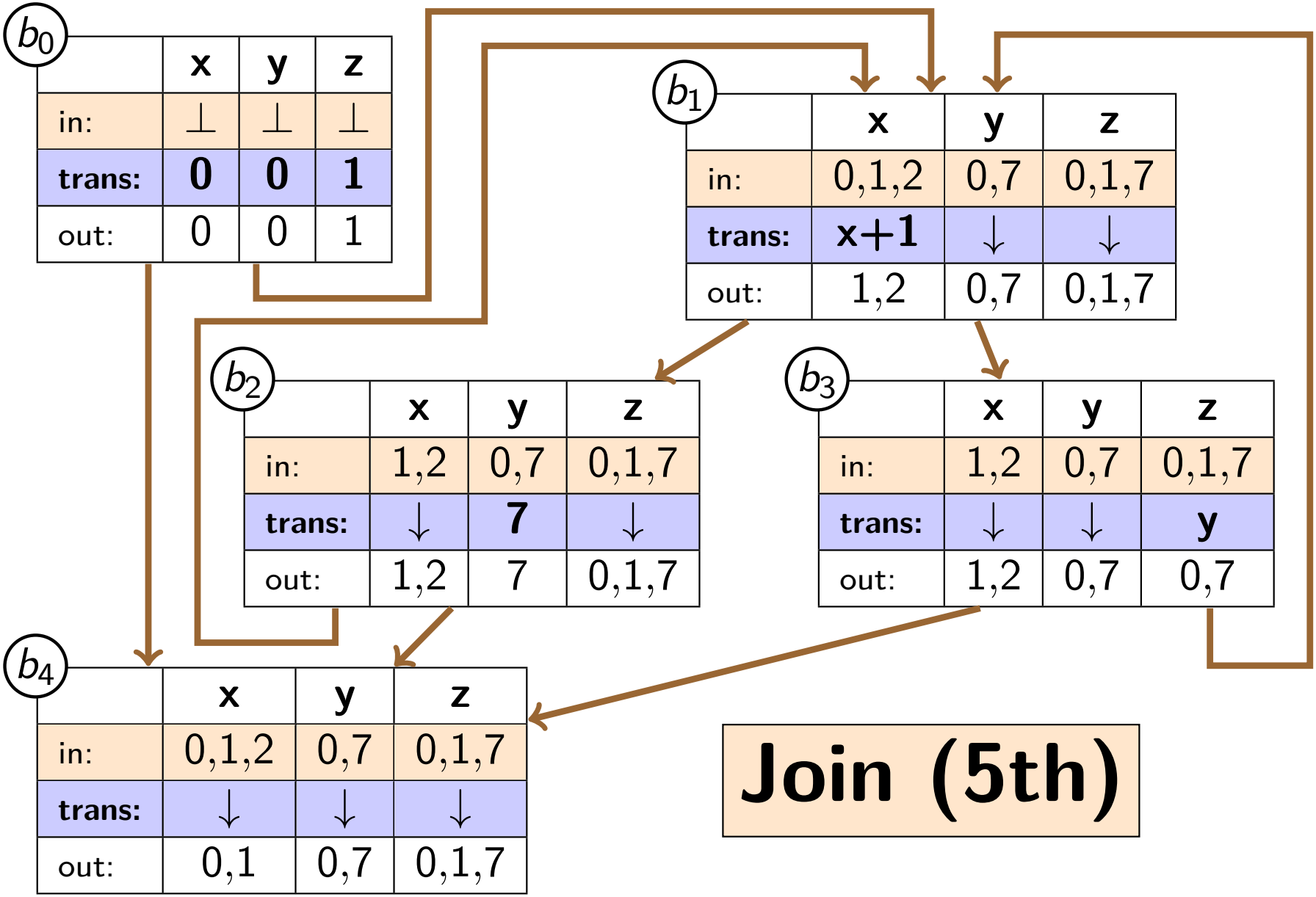
Example: Computing the Fixpoint



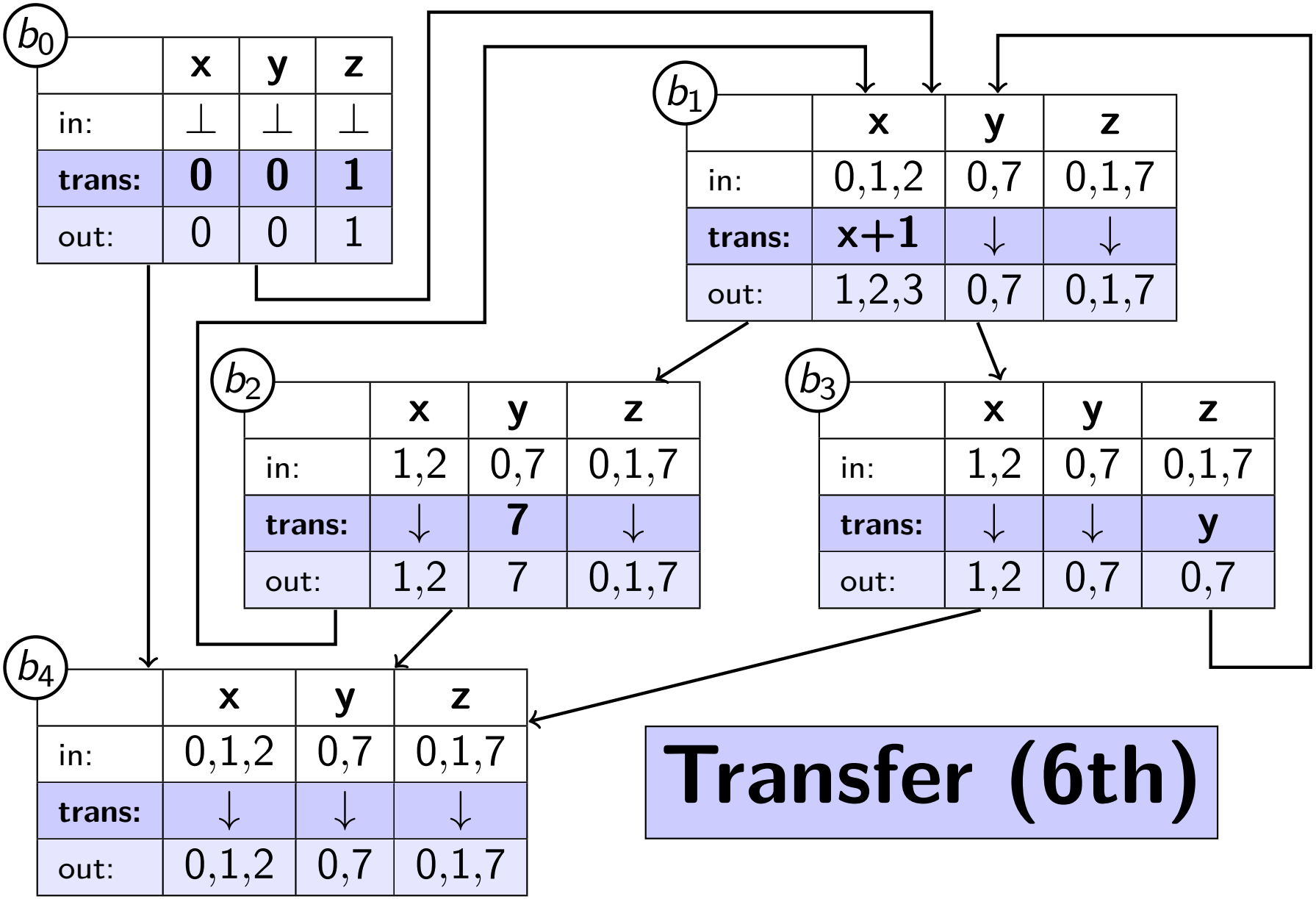
Example: Computing the Fixpoint



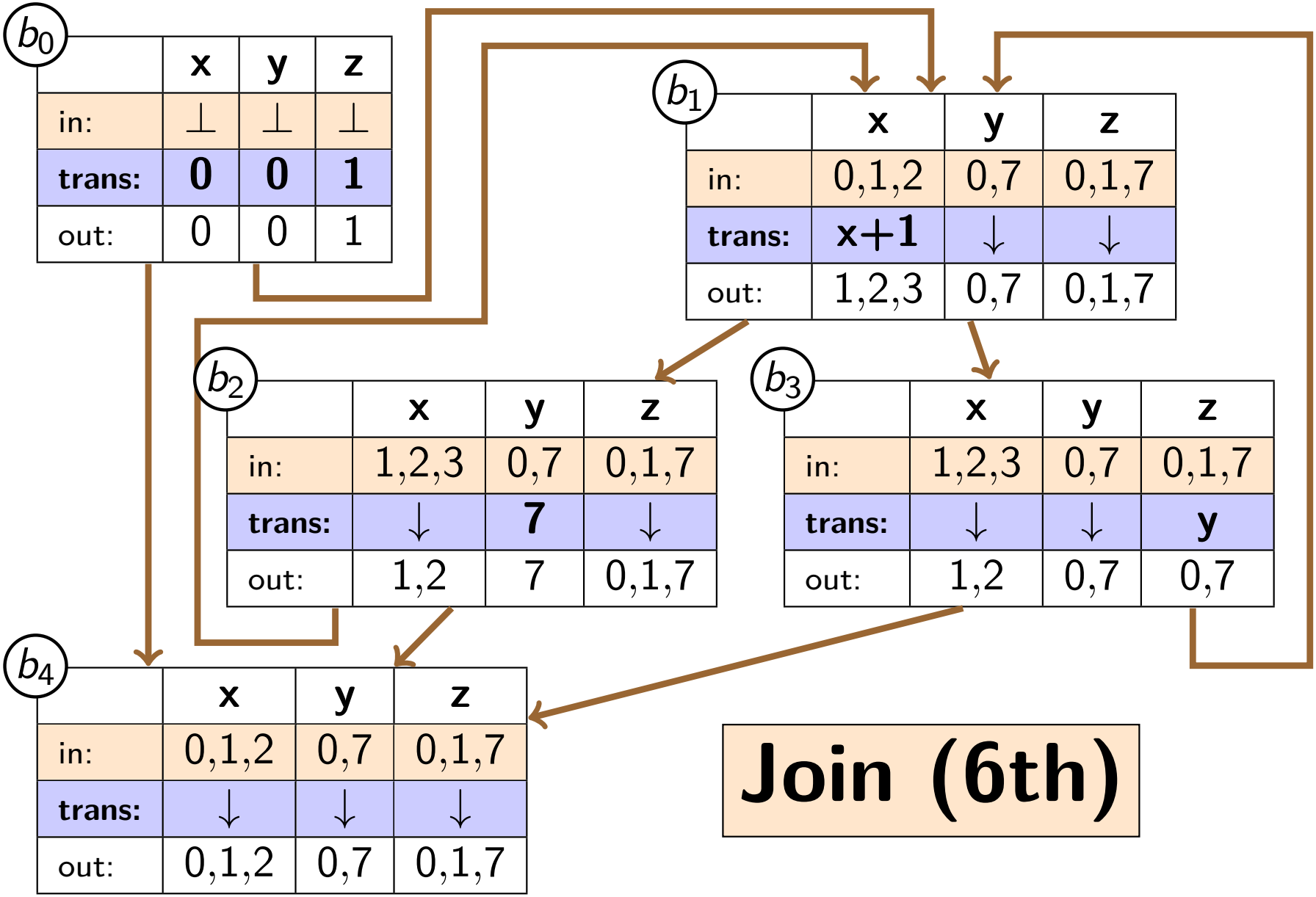
Example: Computing the Fixpoint



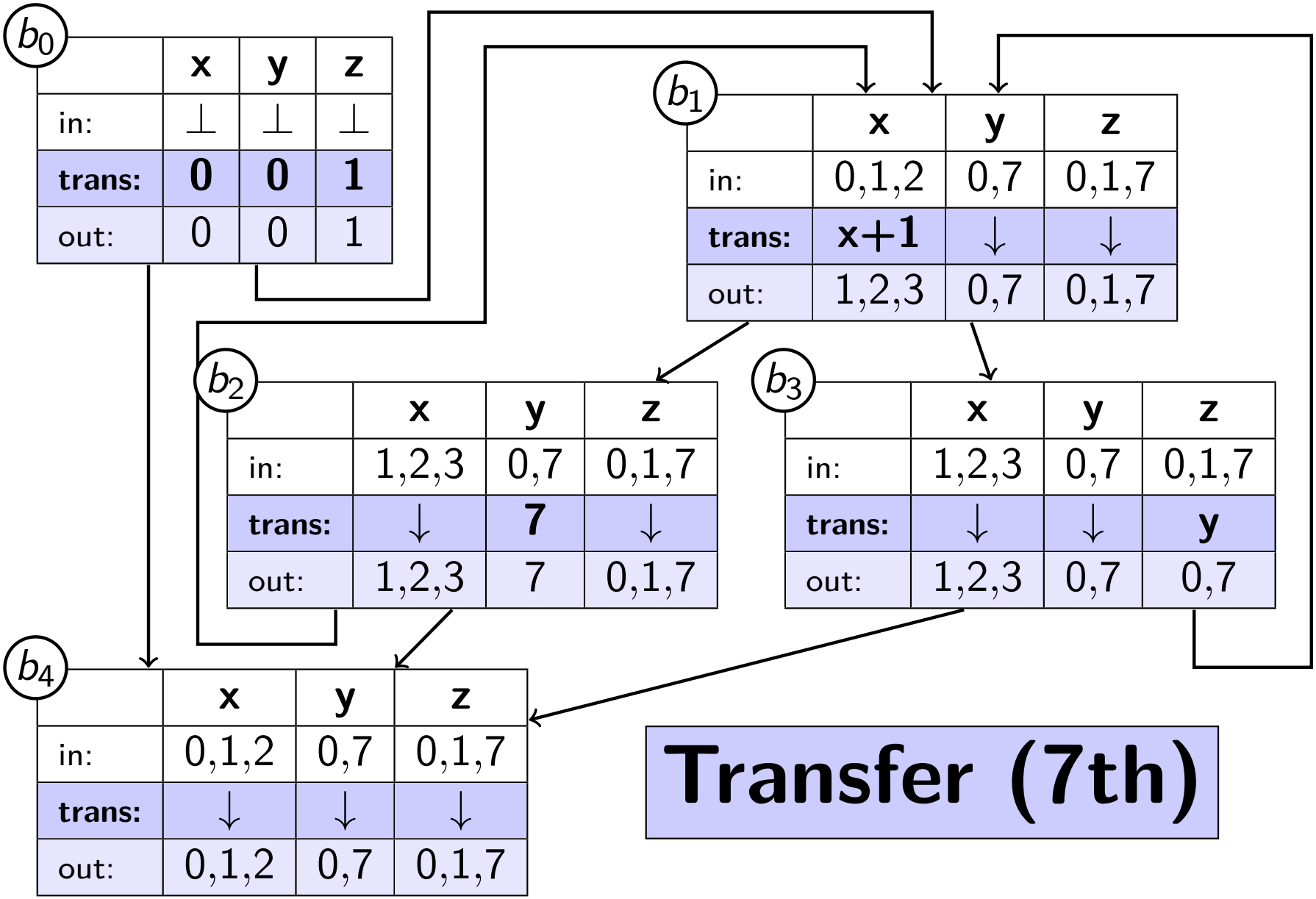
Example: Computing the Fixpoint



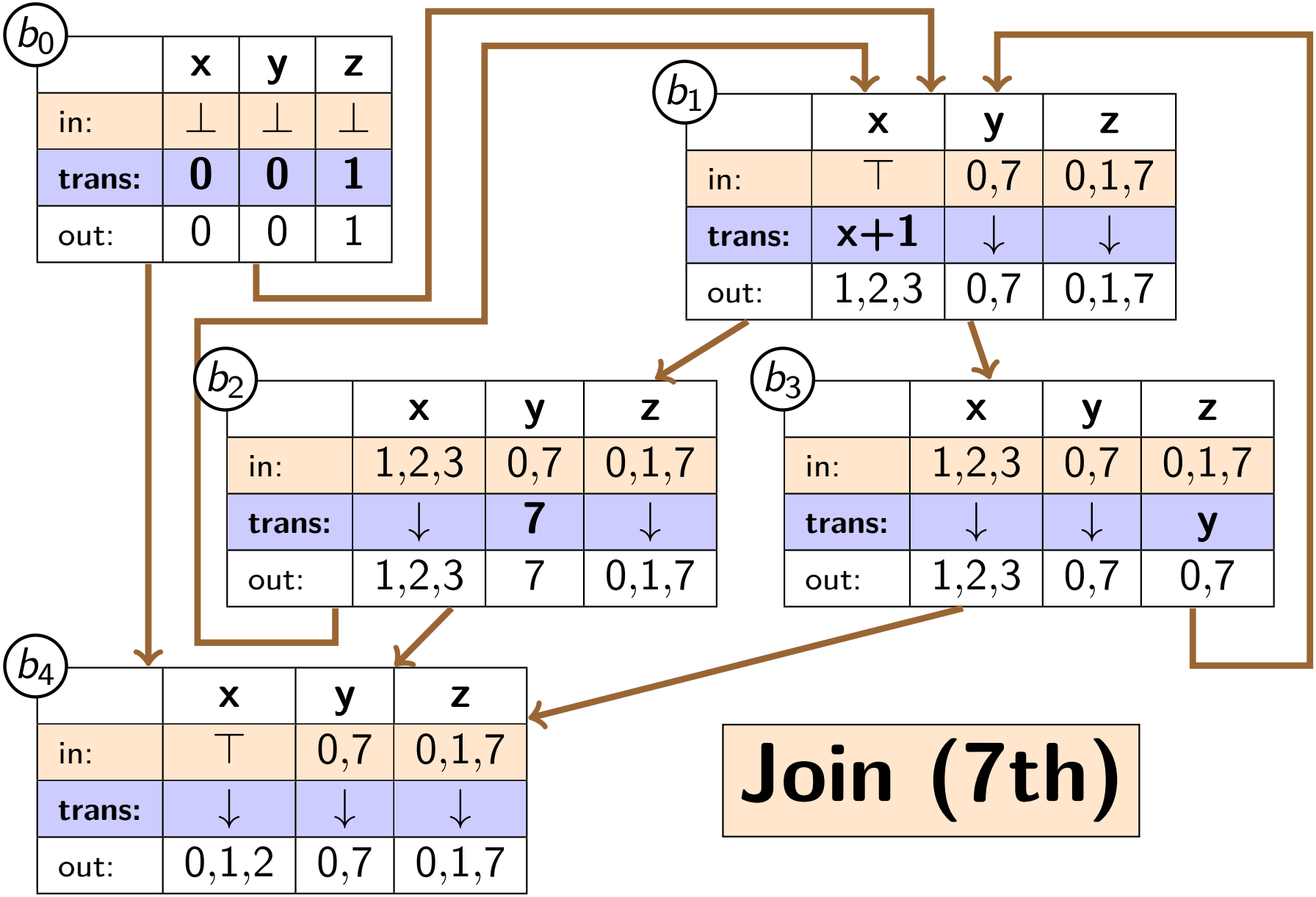
Example: Computing the Fixpoint



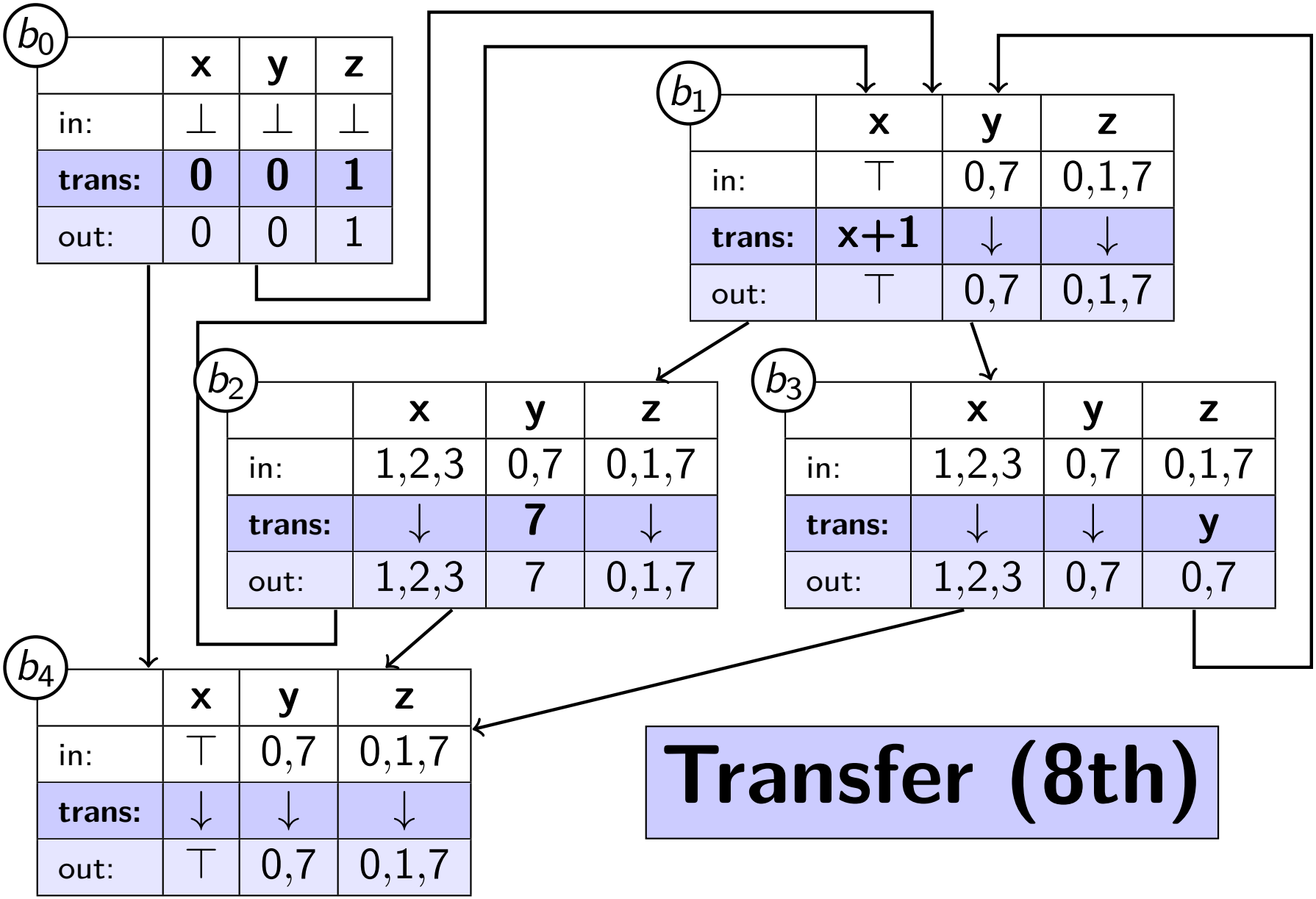
Example: Computing the Fixpoint



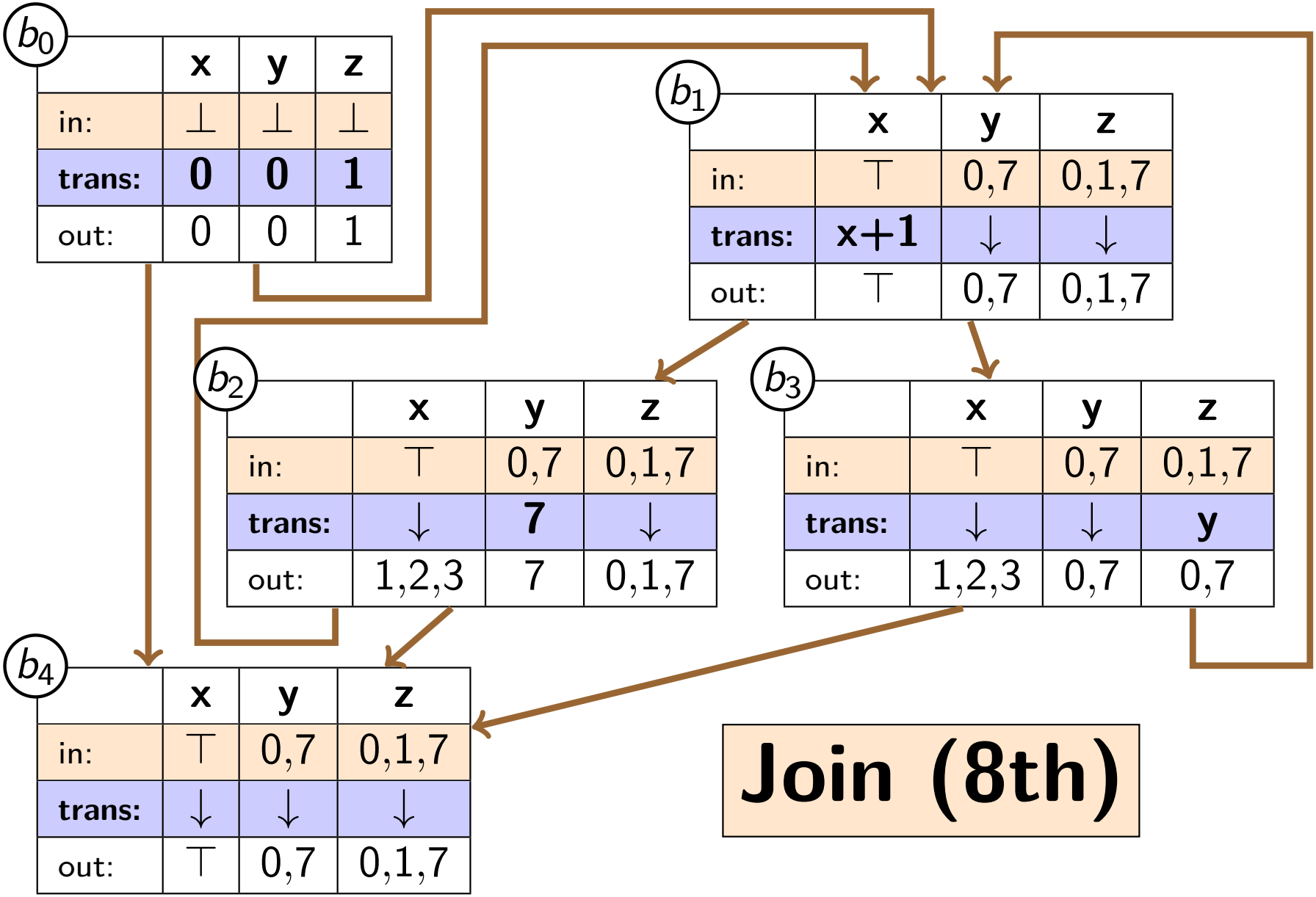
Example: Computing the Fixpoint



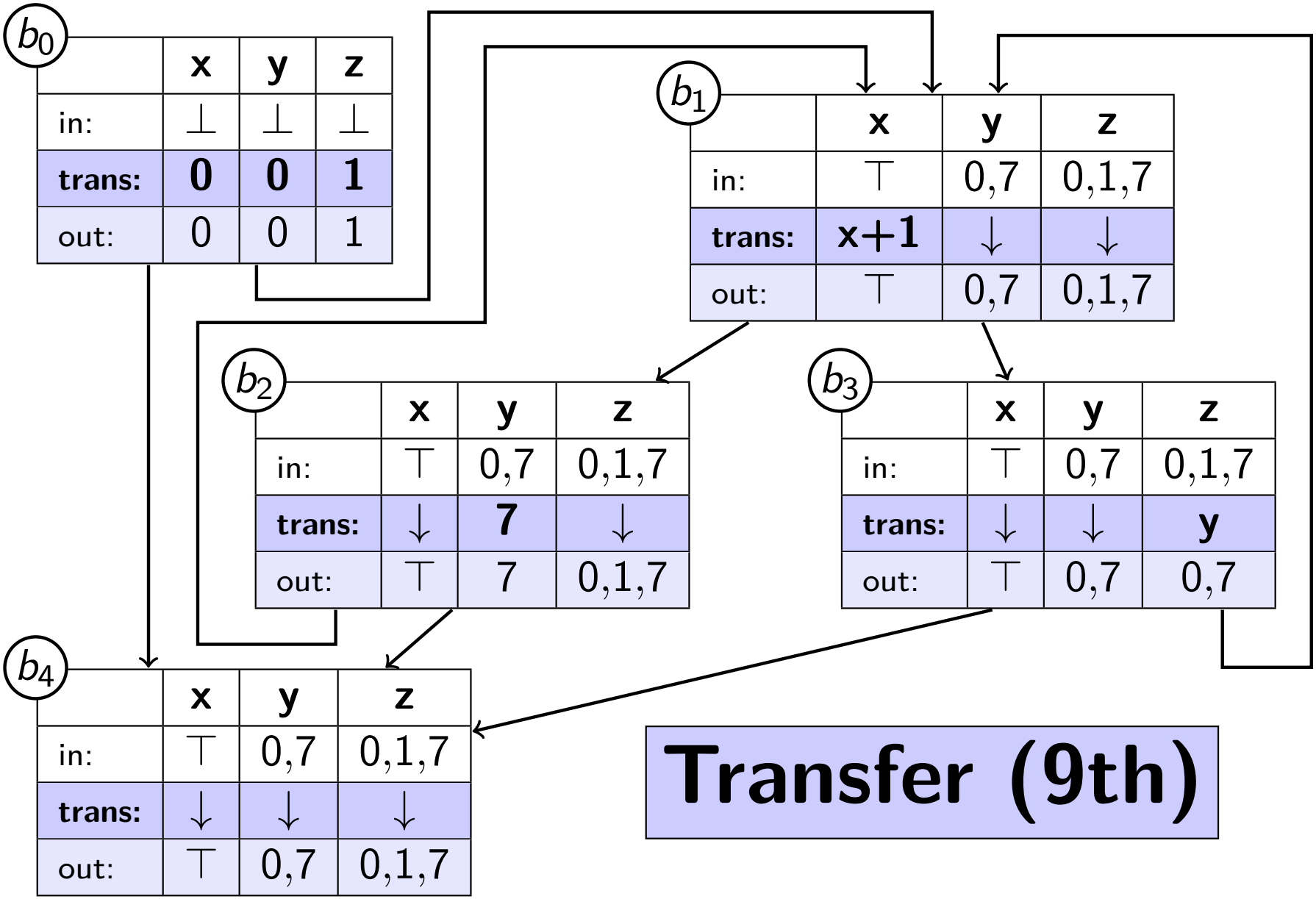
Example: Computing the Fixpoint



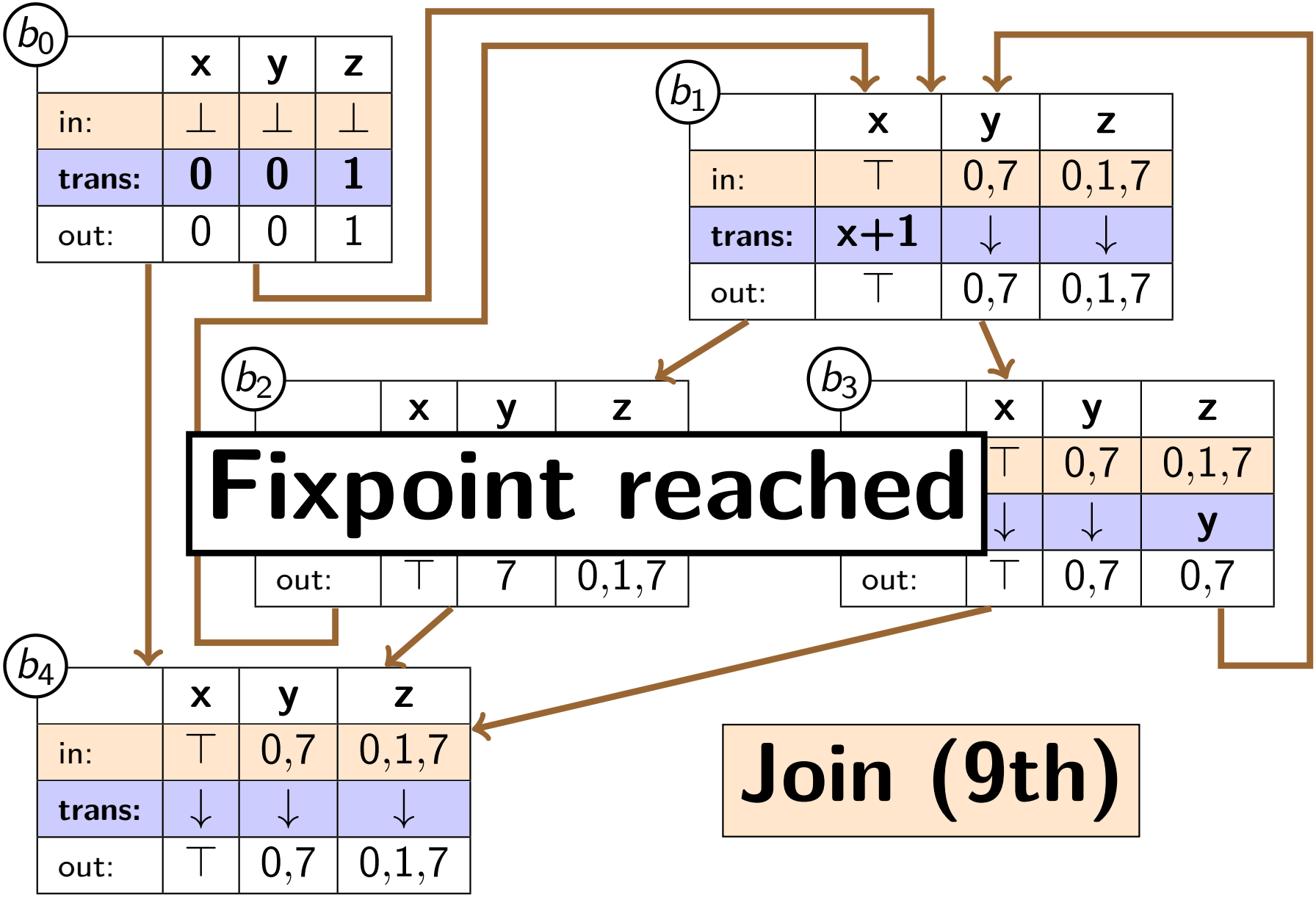
Example: Computing the Fixpoint



Example: Computing the Fixpoint



Example: Computing the Fixpoint



Example: Conclusion

```
var x := 0;
var y := 0;
var z := 1;

while x < 5 {
  x := x + 1;
  if x >= 2 {
    y := 7;
  } else {
    z := y;
} }

return [x, y, z];
```

- Applied abstract domain to three variables
- Reached fixpoint after 9 iterations
- Return values:
  x : $\top$ (unknown/any)
  y : 0 or 7
  z : 0 or 1 or 7
- Conservative approximation of reality
- Once x reached more than 3 values, algorithm gave up and went to $\top$
- *This is only one possible design for this analysis*