Global Optimizations

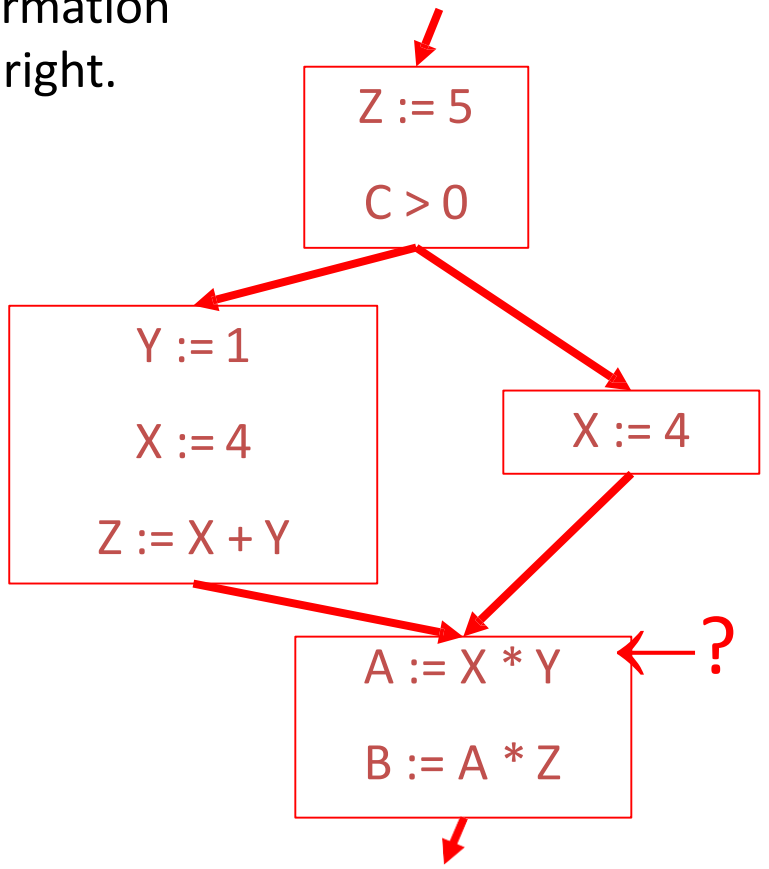
Lecture 12

Exercise

Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

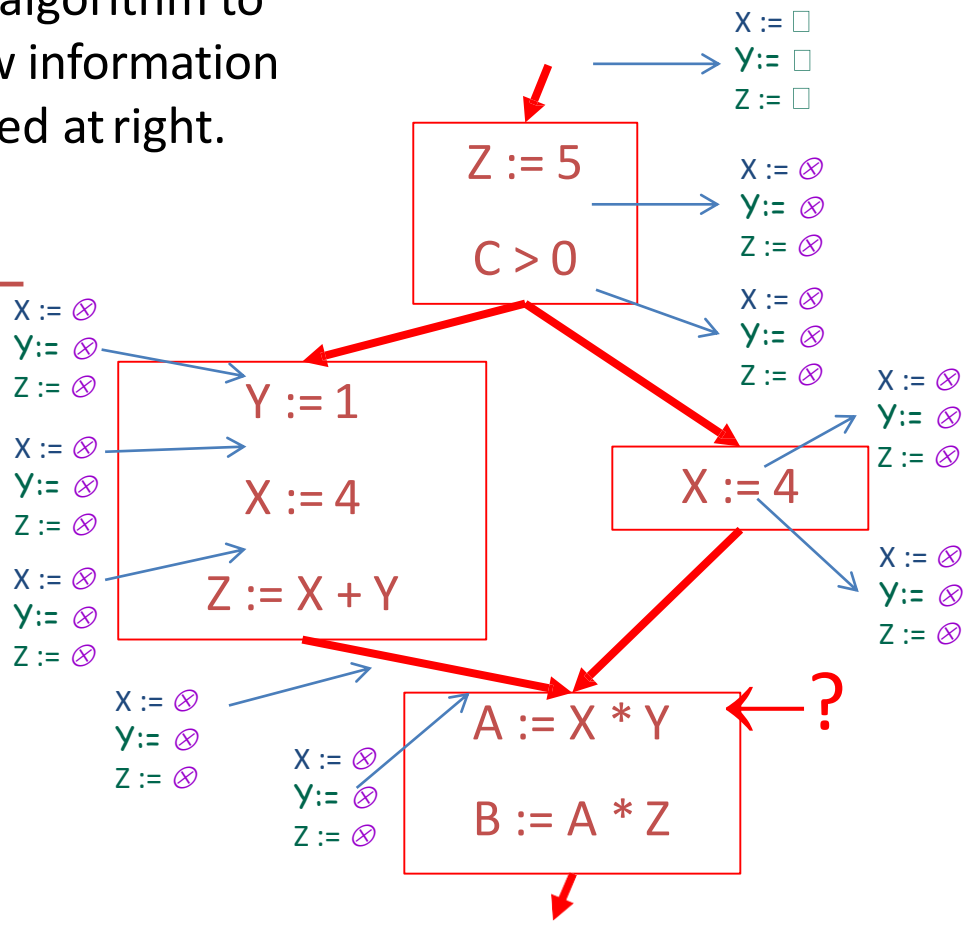
	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

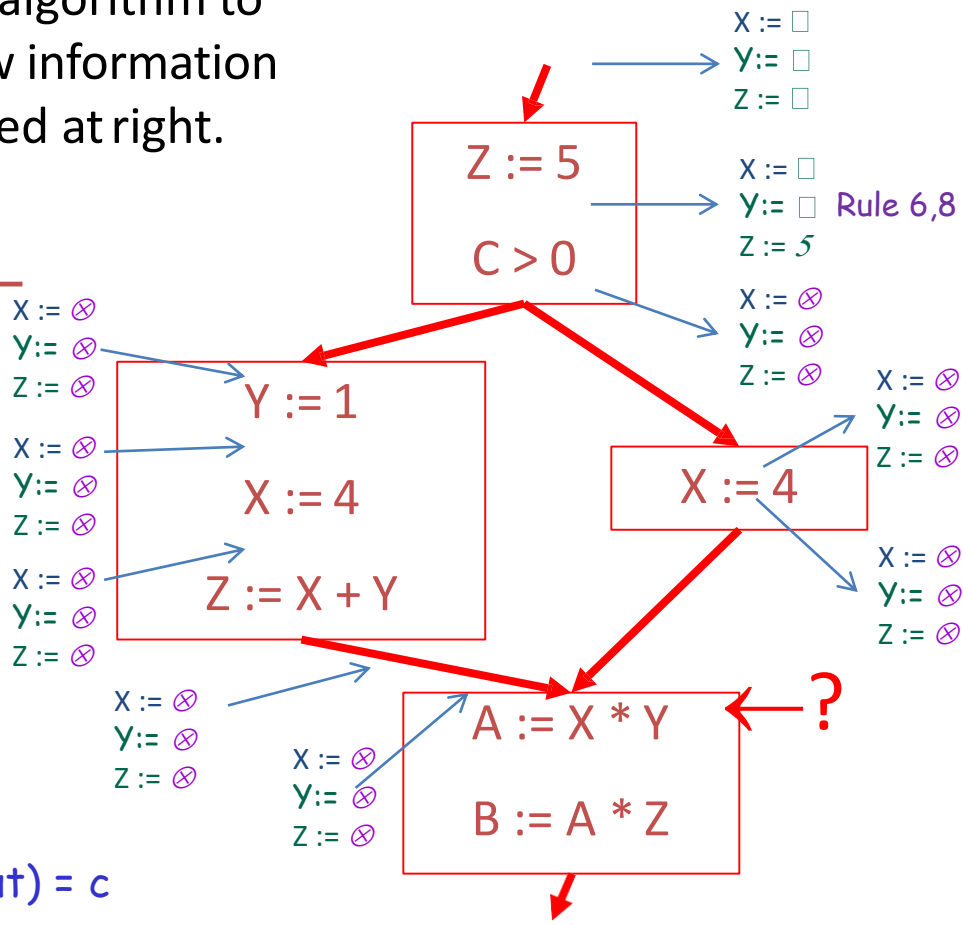
	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



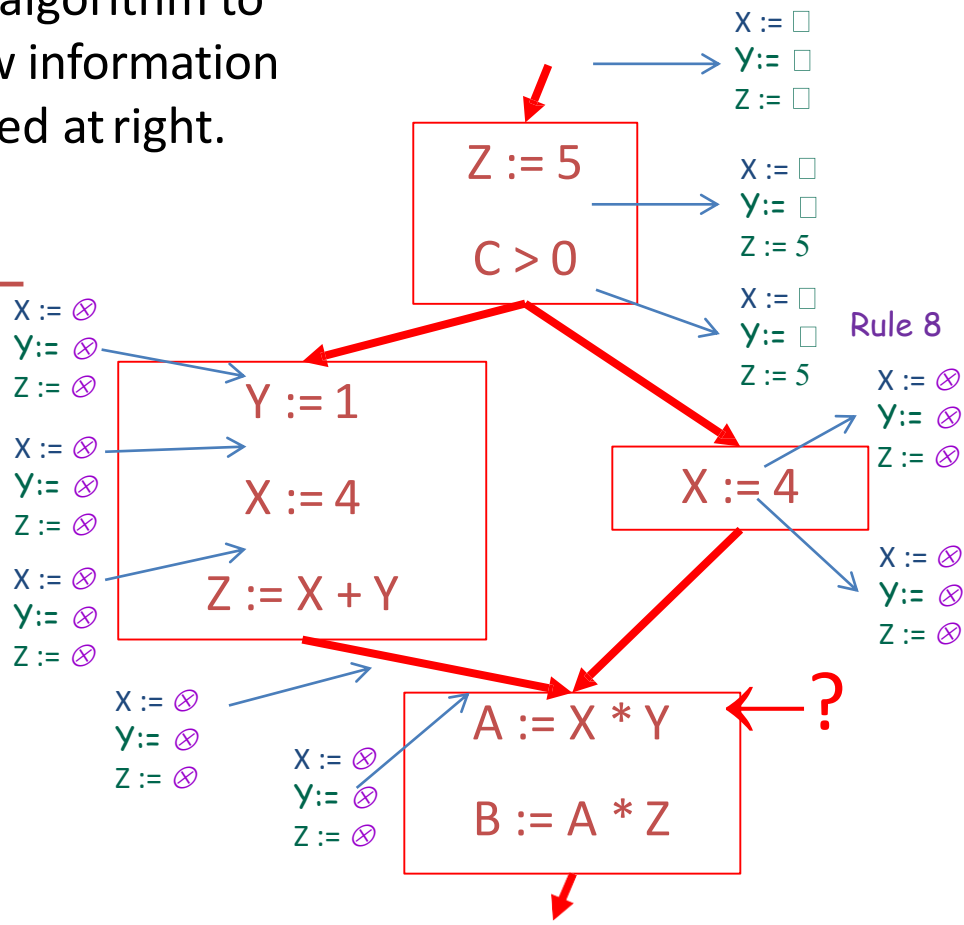
Rule 6: if c is a constant then C(x := c, x, out) = c

Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□

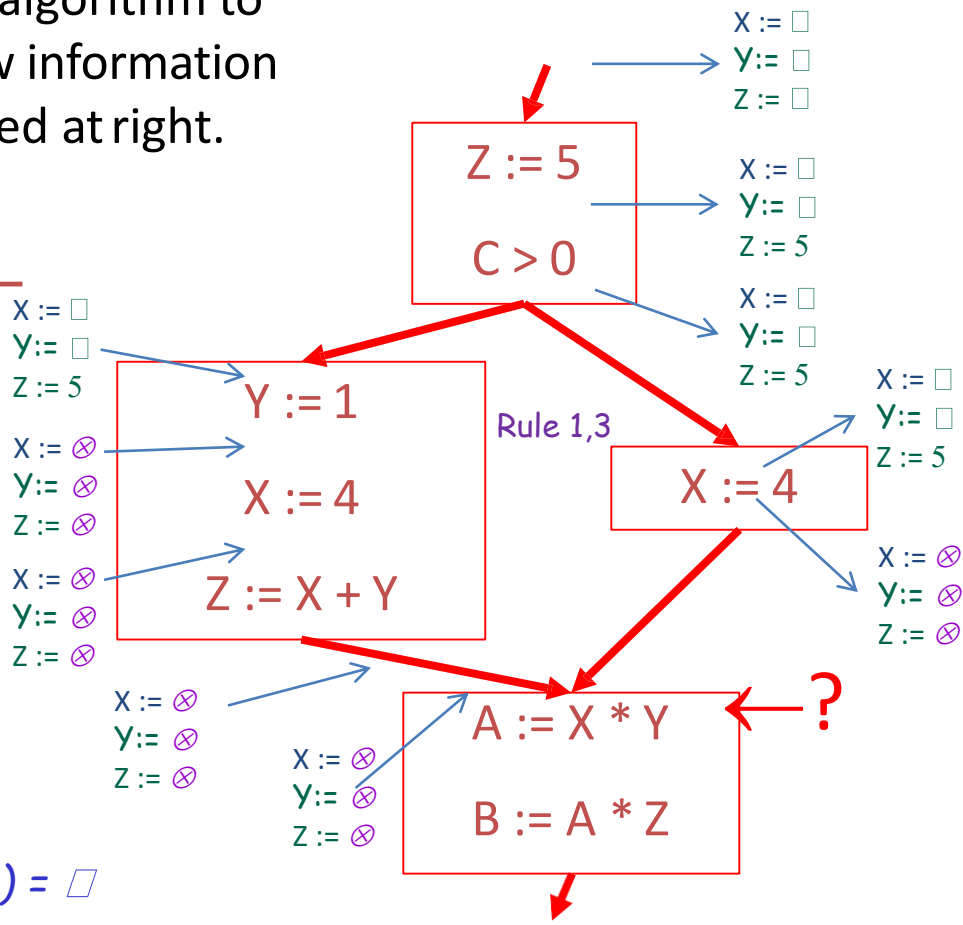


Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



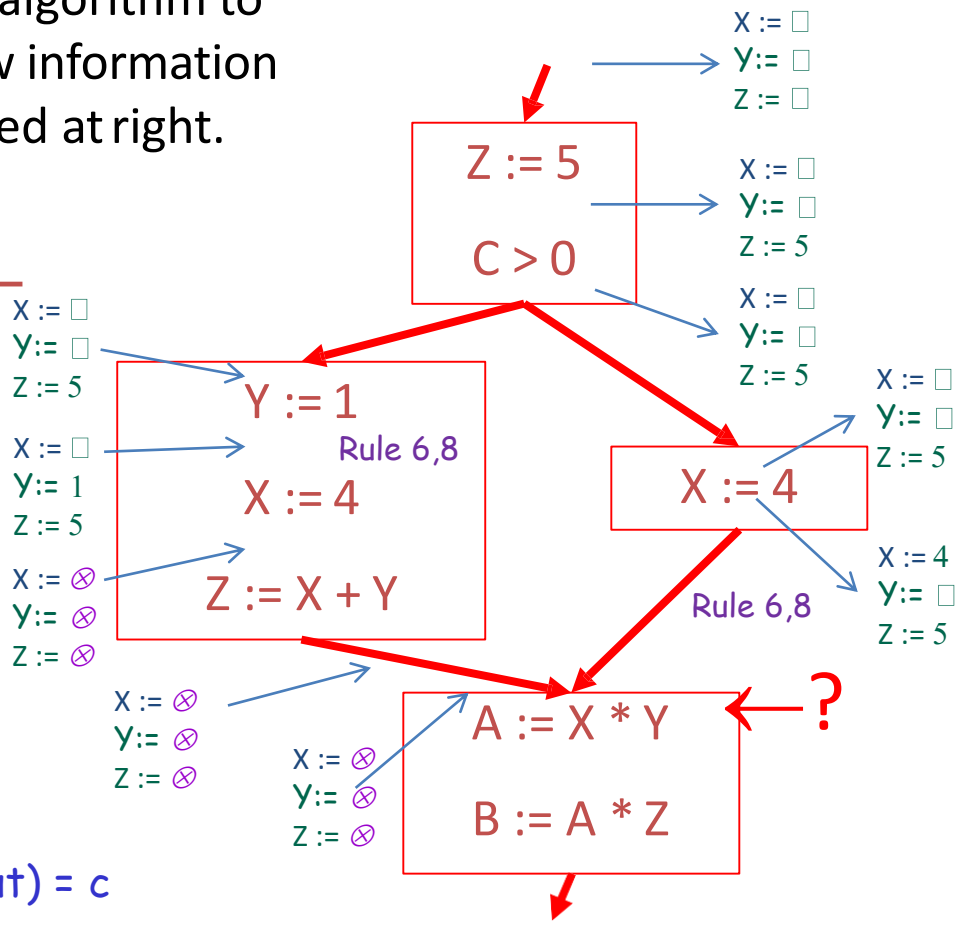
Rule 1: *if* $\exists_i$ $(C(p_i, x, out) = \square)$ *then* $C(s, x, in) = \square$

Rule 3: *if* $\forall_i$ $(C(p_i, x, out) = c$ *or* $\otimes)$ *then* $C(s, x, in) = c$

Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



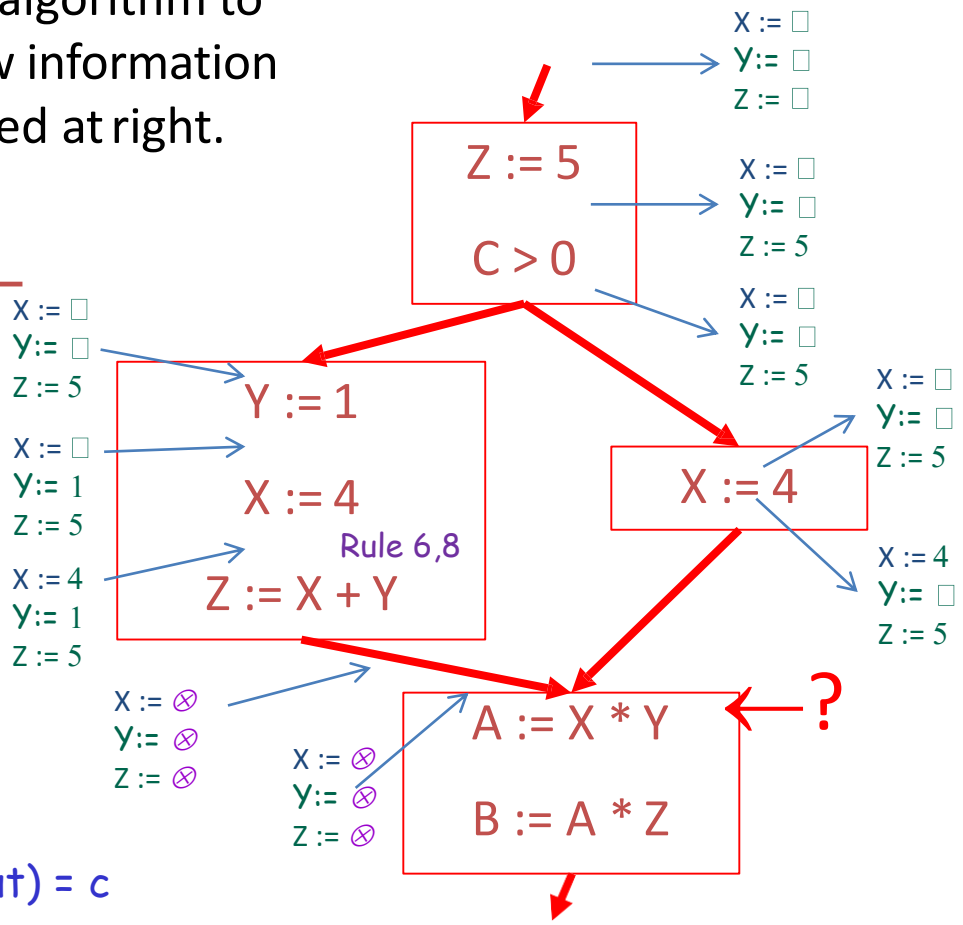
Rule 6: if c is a constant then C(x := c, x, out) = c

Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



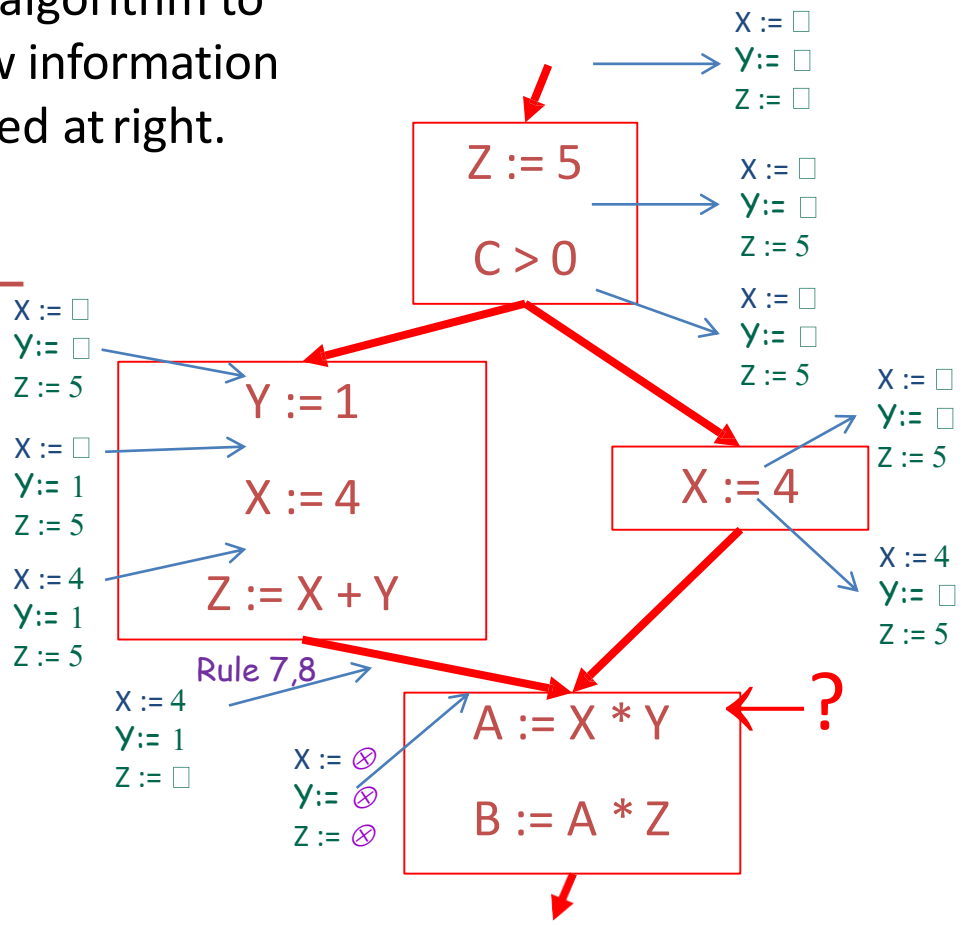
Rule 6: if c is a constant then C(x := c, x, out) = c

Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



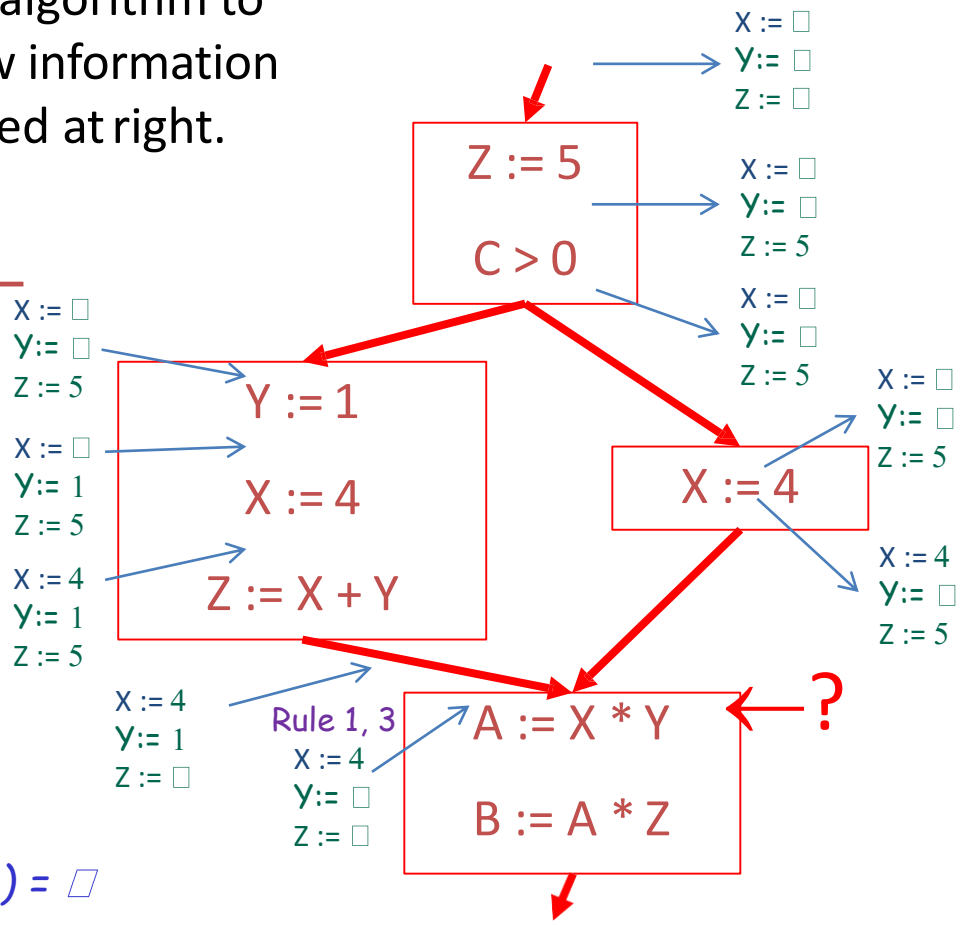
Rule 7: $C(x := f(\ldots), x, out) = \square$

Rule 8: if $x \leftrightarrow y$ then $C(y := \ldots, x, out) = C(y := \ldots, x, in)$

Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



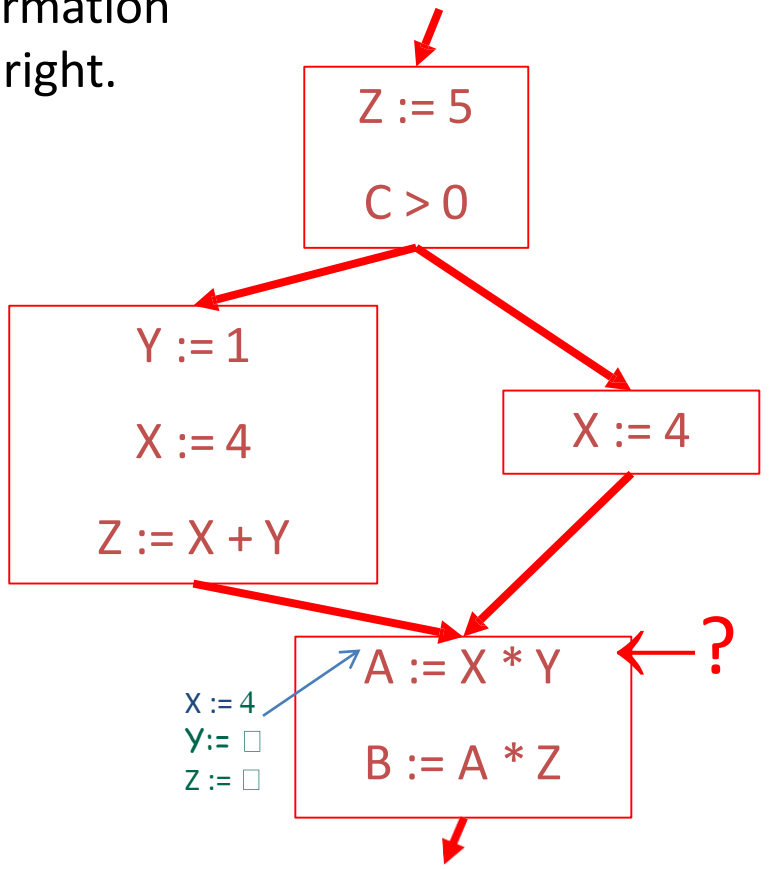
Rule 1: *if* $\exists_i$ ($C(p_i, x, out) = \square$) *then* $C(s, x, in) = \square$

Rule 3: *if* $\forall_i$ ($C(p_i, x, out) = c$ *or* $\otimes$) *then* $C(s, x, in) = c$

Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

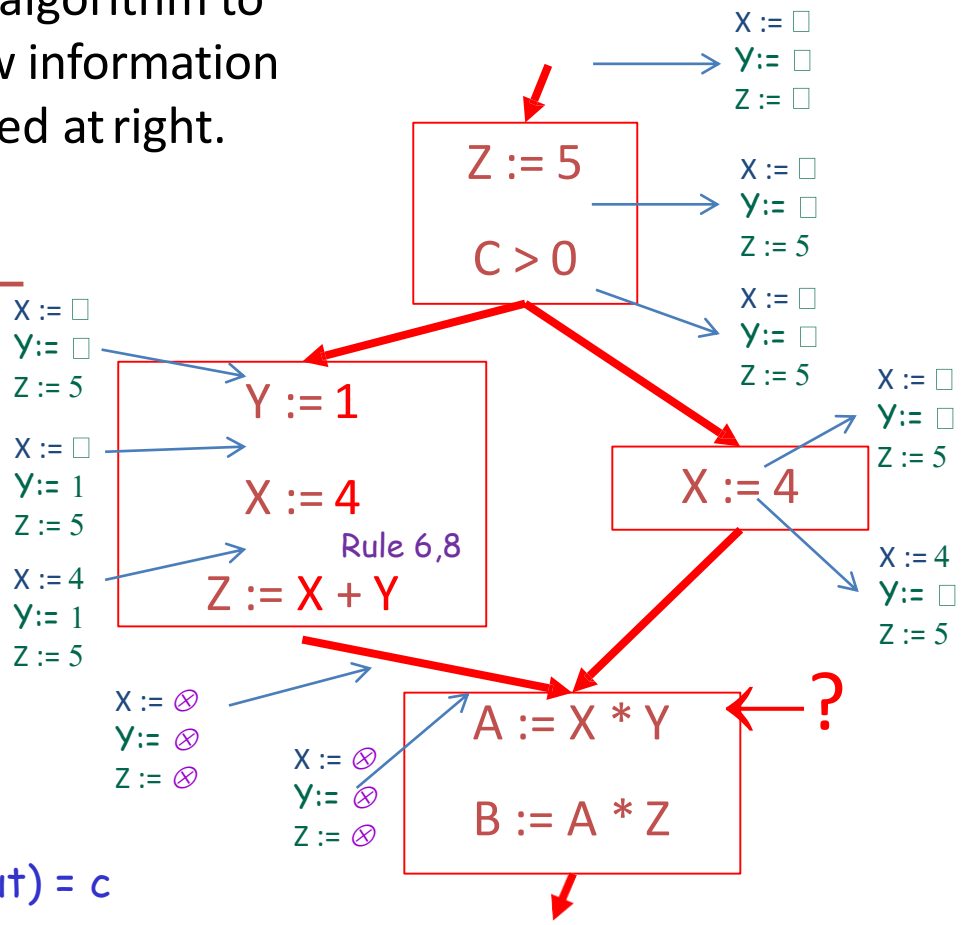
	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



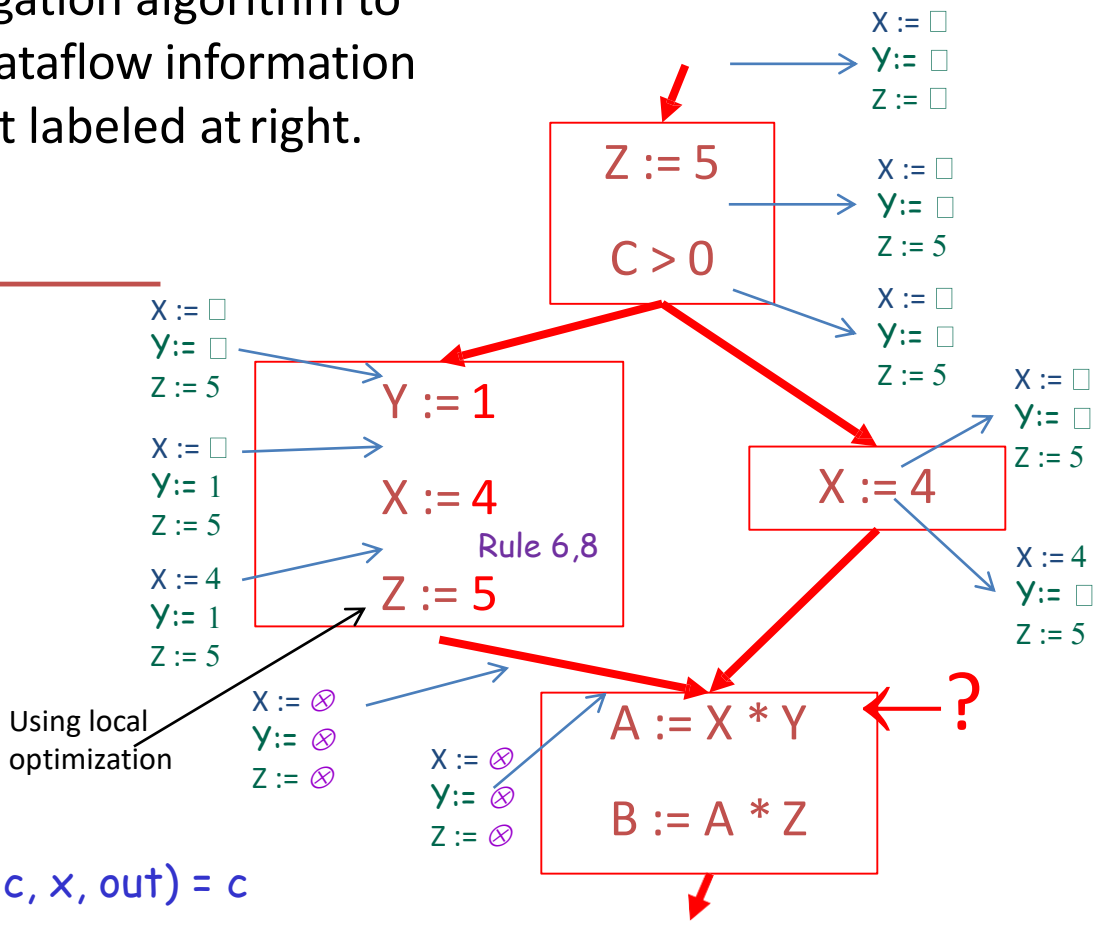
Rule 6: if c is a constant then C(x := c, x, out) = c

Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



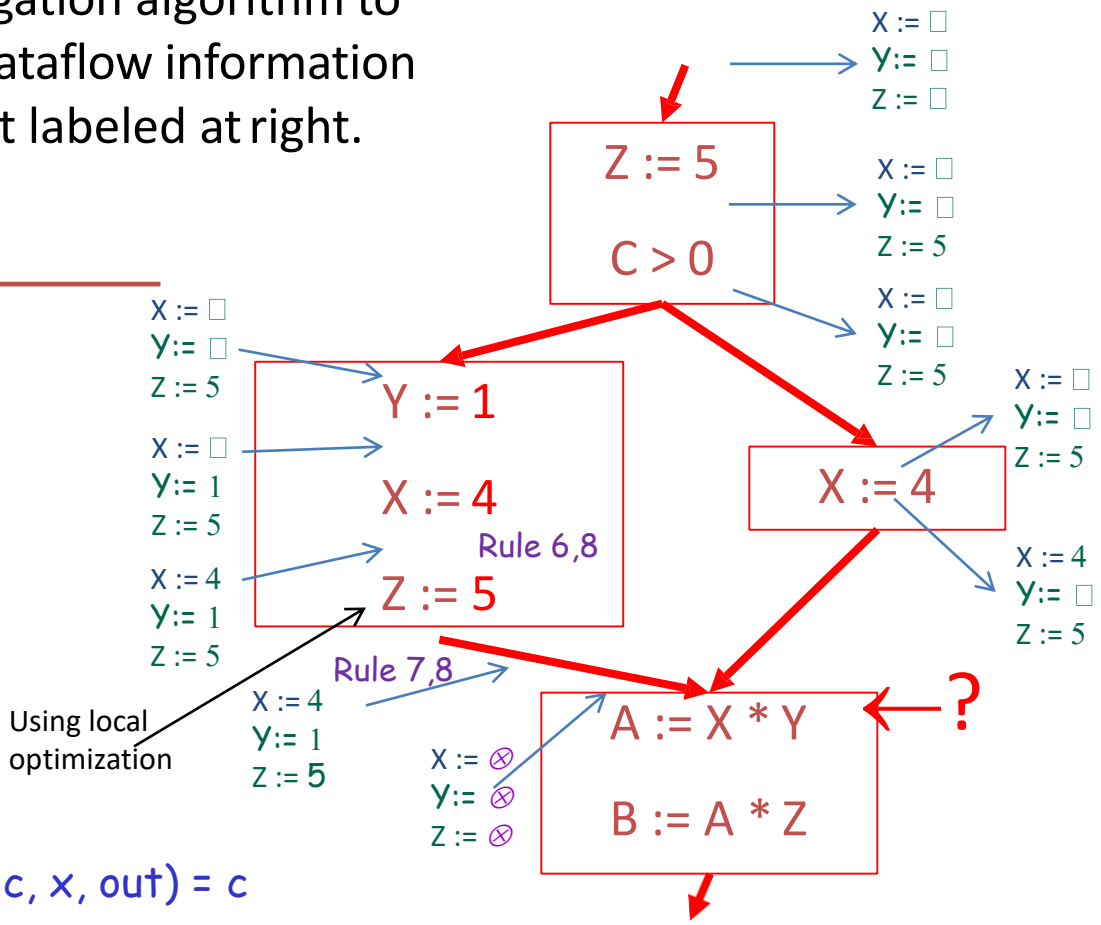
Rule 6: if c is a constant then C(x := c, x, out) = c

Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



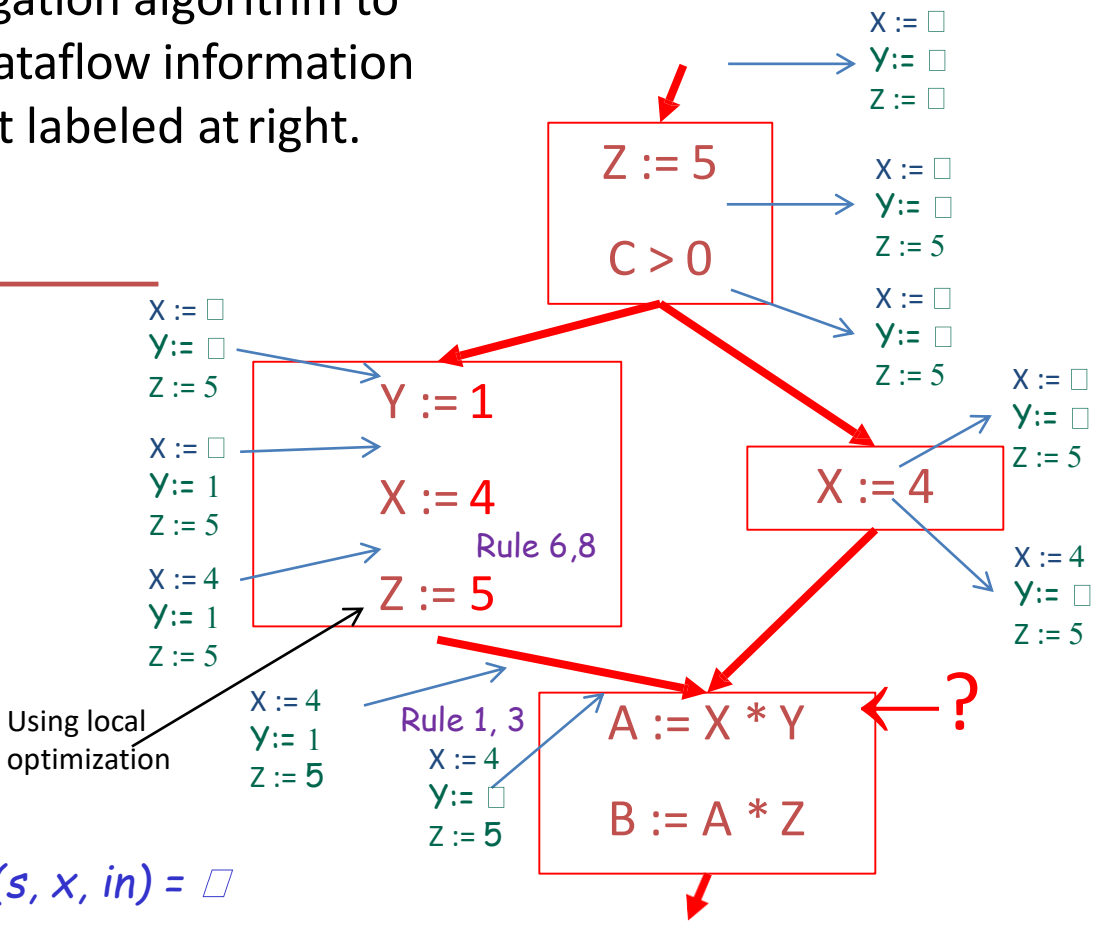
Rule 6: if c is a constant then C(x := c, x, out) = c

Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



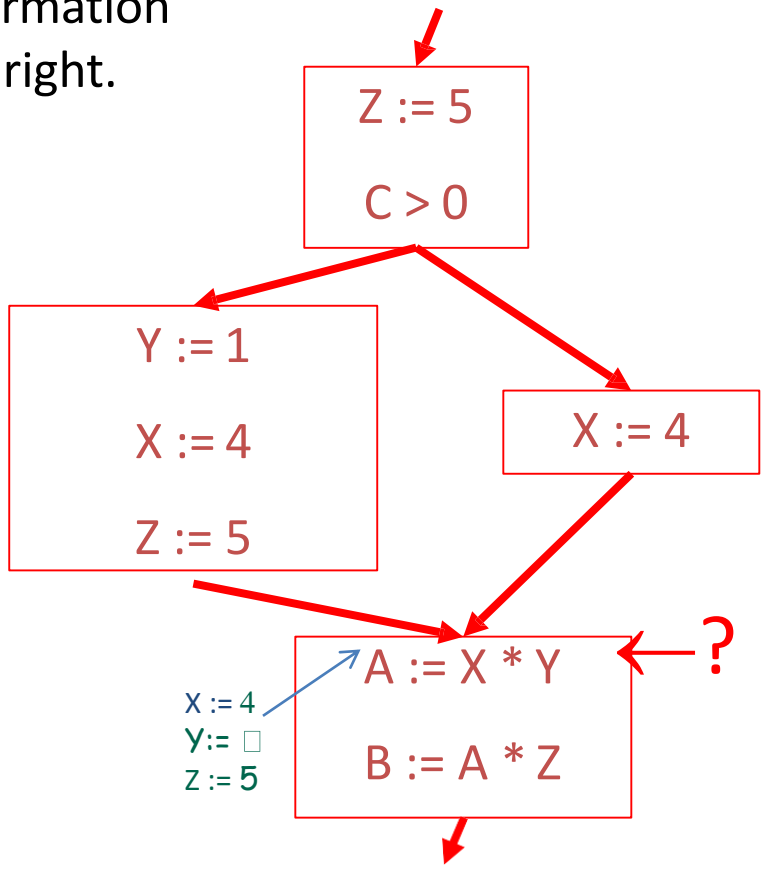
Rule 1: *if* $\exists_i$ ($C(p_i, x, out) = \square$) *then* $C(s, x, in) = \square$

Rule 3: *if* $\forall_i$ ($C(p_i, x, out) = c$ *or* $\otimes$) *then* $C(s, x, in) = c$

Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

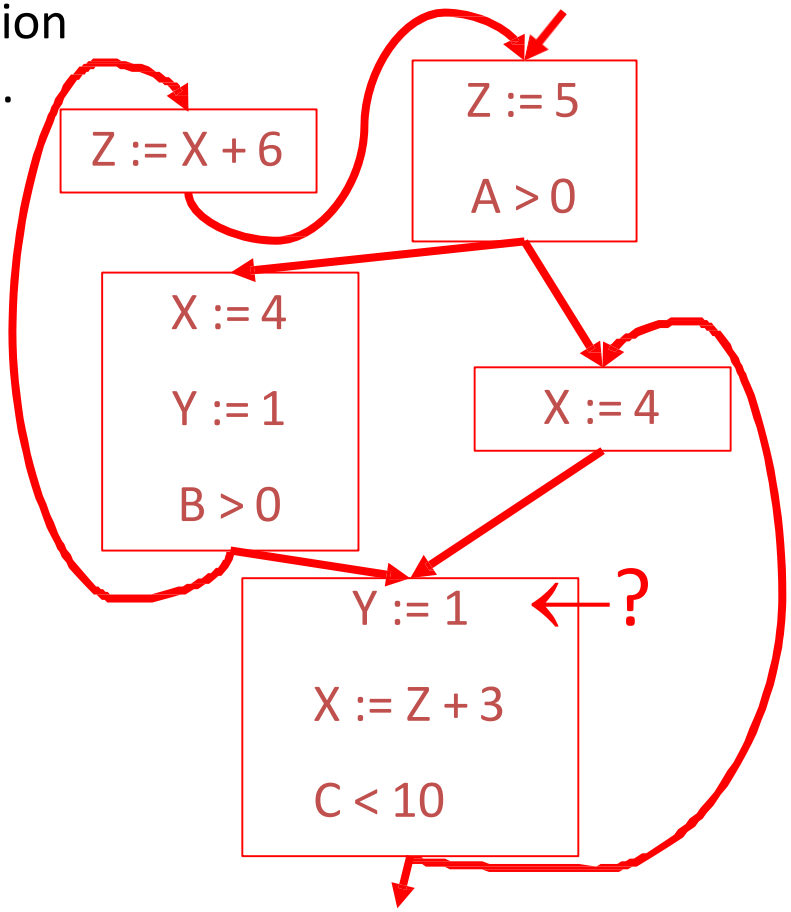
	X	Y	Z
○	4	□	□
○	4	□	5
○	4	1	5
○	□	□	□



Question?

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

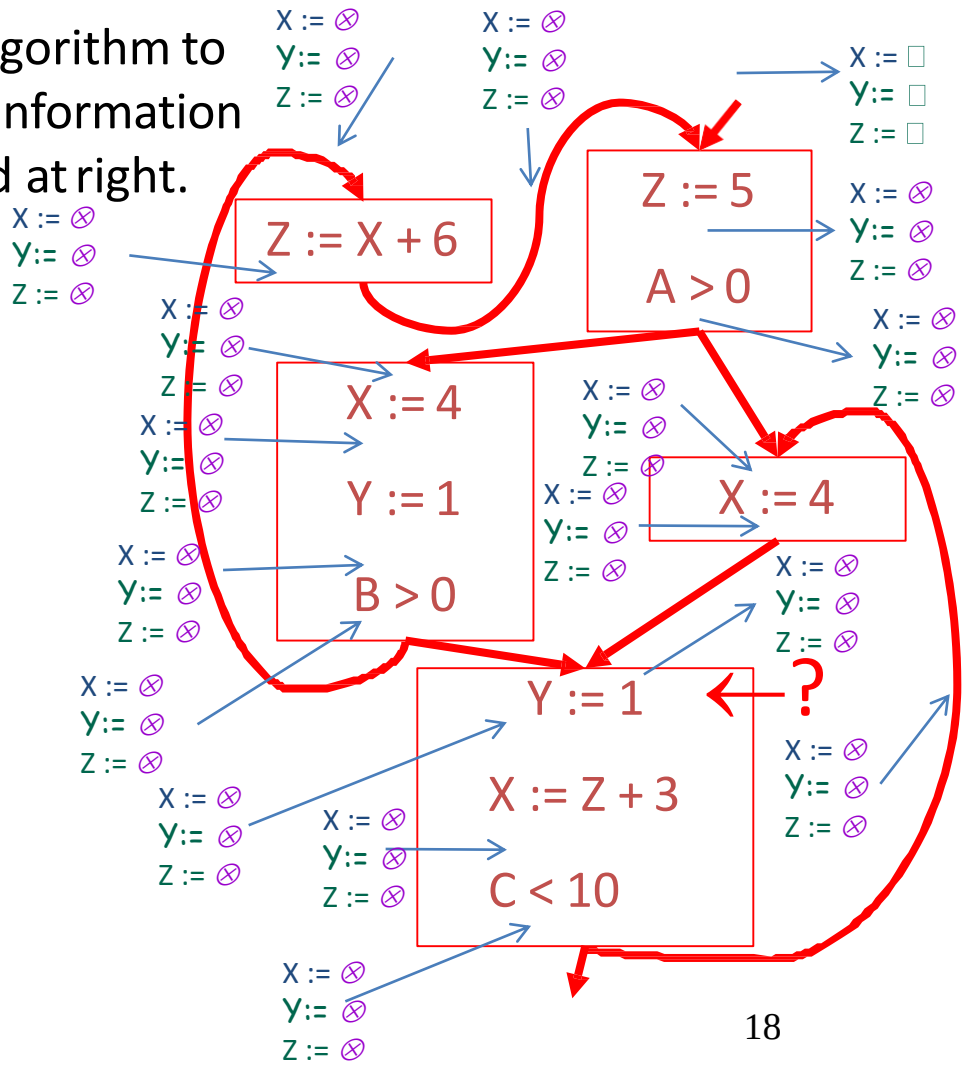
	X	Y	Z
○	□	1	□
○	4	□	5
○	4	1	5
○	4	□	□



Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

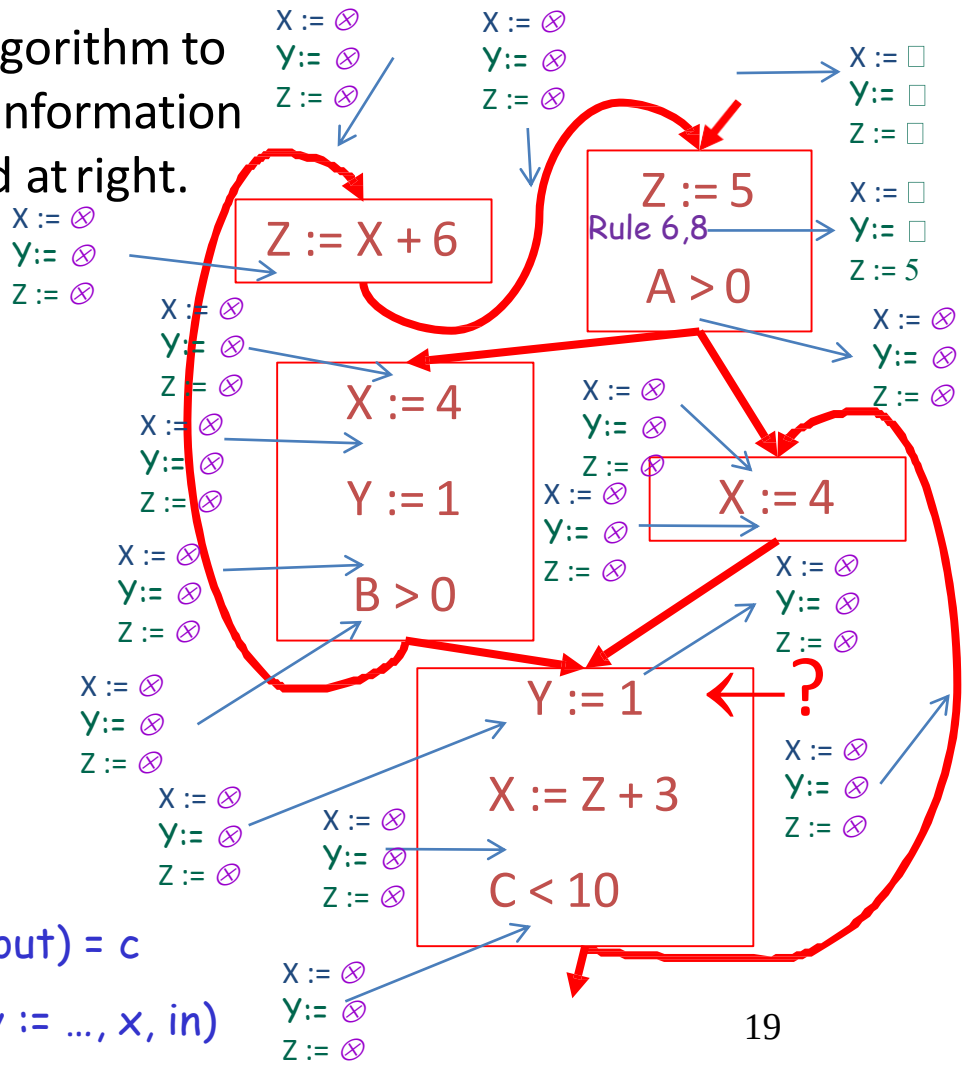
	X	Y	Z
○	□	1	□
○	4	□	5
○	4	1	5
○	4	□	□



Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	□	1	□
○	4	□	5
○	4	1	5
○	4	□	□



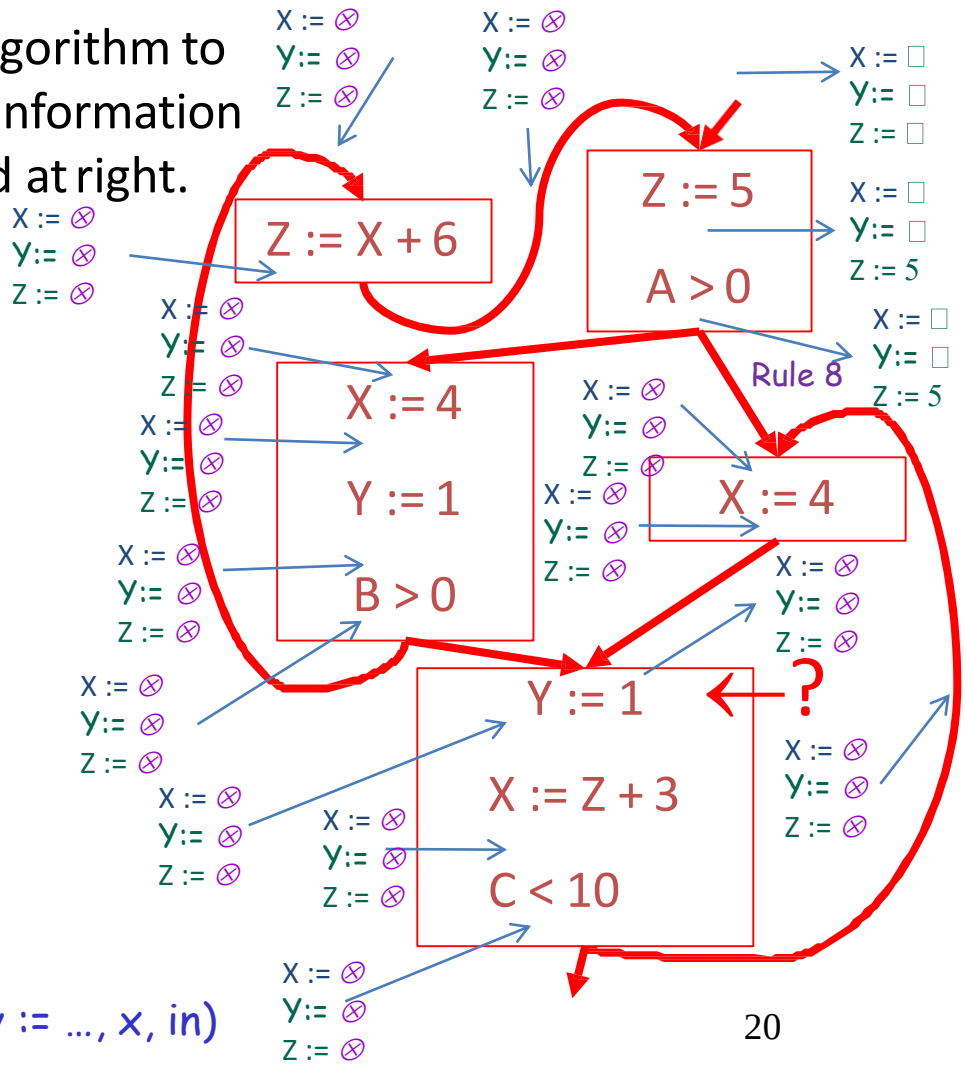
Rule 6: if c is a constant then C(x := c, x, out) = c

Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	□	1	□
○	4	□	5
○	4	1	5
○	4	□	□

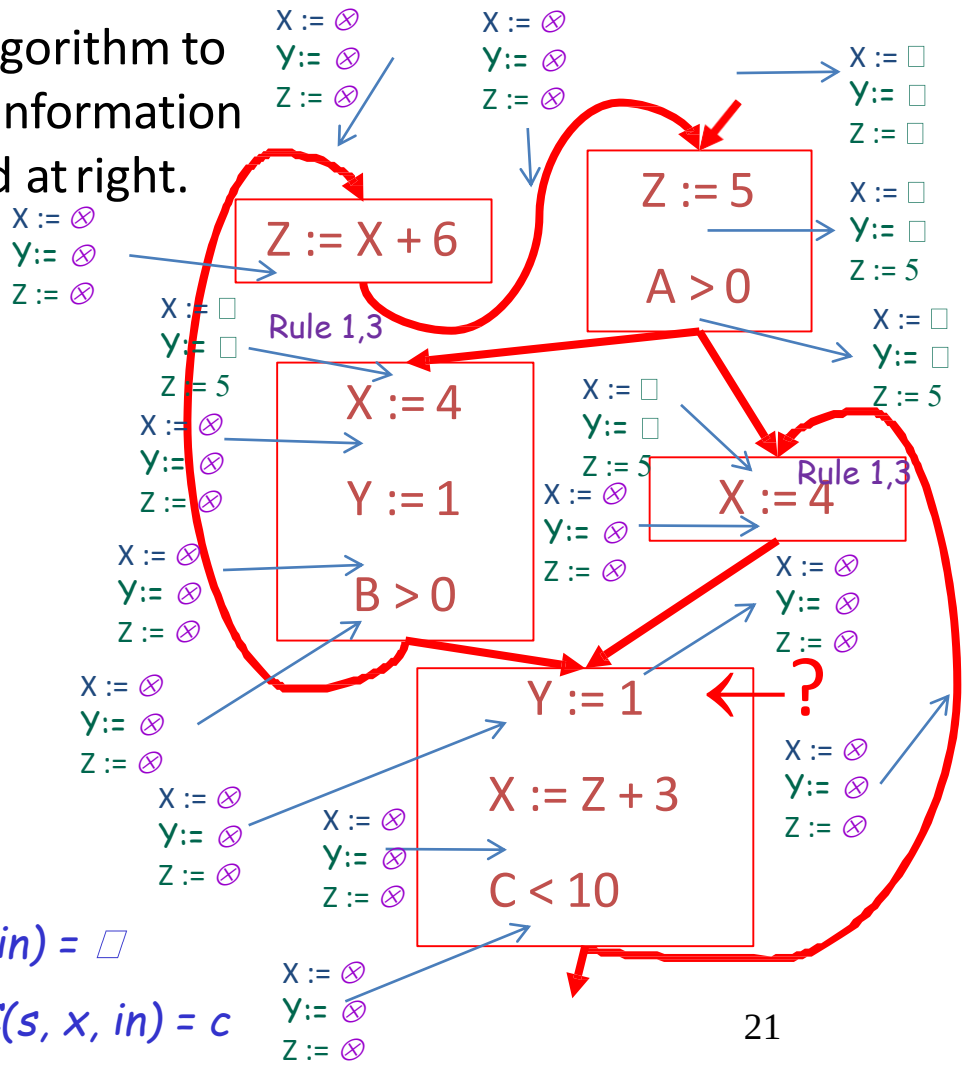


Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	□	1	□
○	4	□	5
○	4	1	5
○	4	□	□



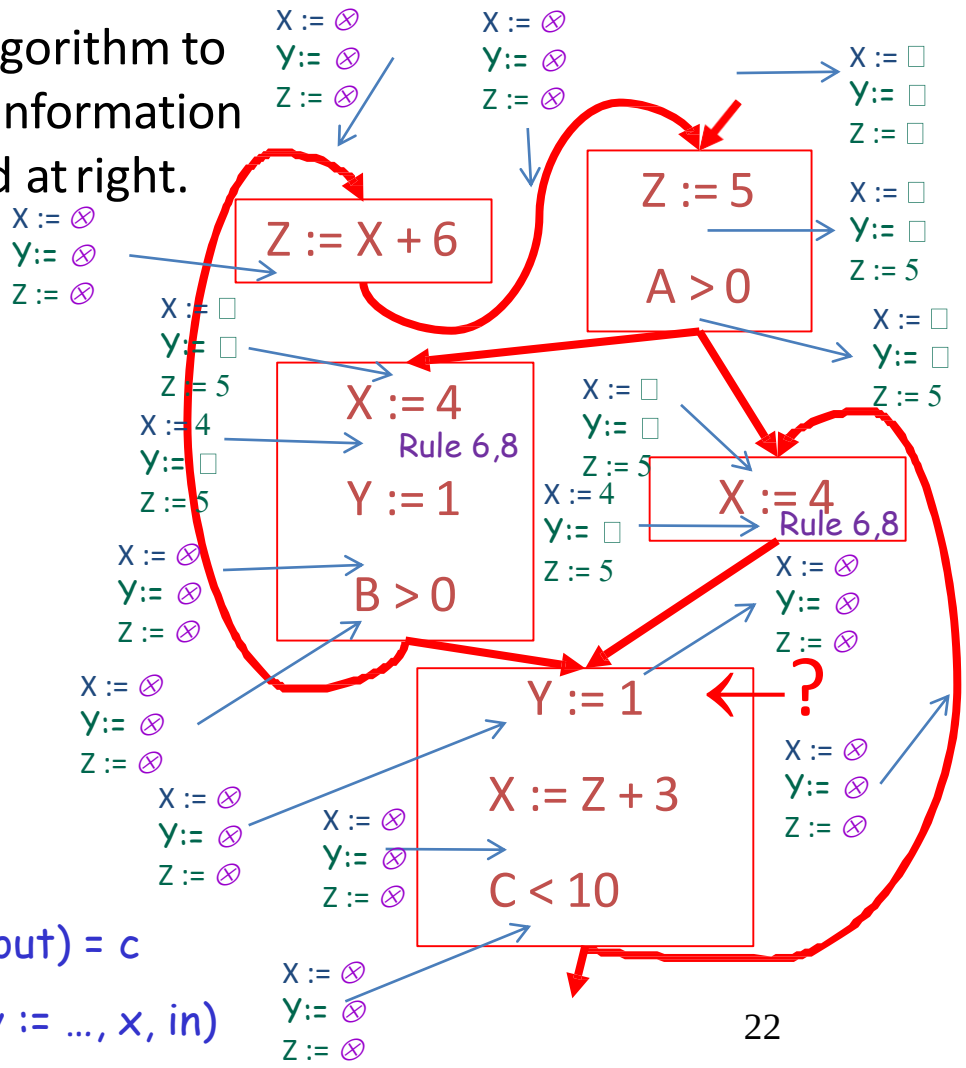
Rule 1: *if* $\exists_i$ ($C(p_i, x, out) = \square$) *then* $C(s, x, in) = \square$

Rule 3: *if* $\forall_i$ ($C(p_i, x, out) = c$ *or* $\otimes$) *then* $C(s, x, in) = c$

Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	□	1	□
○	4	□	5
○	4	1	5
○	4	□	□



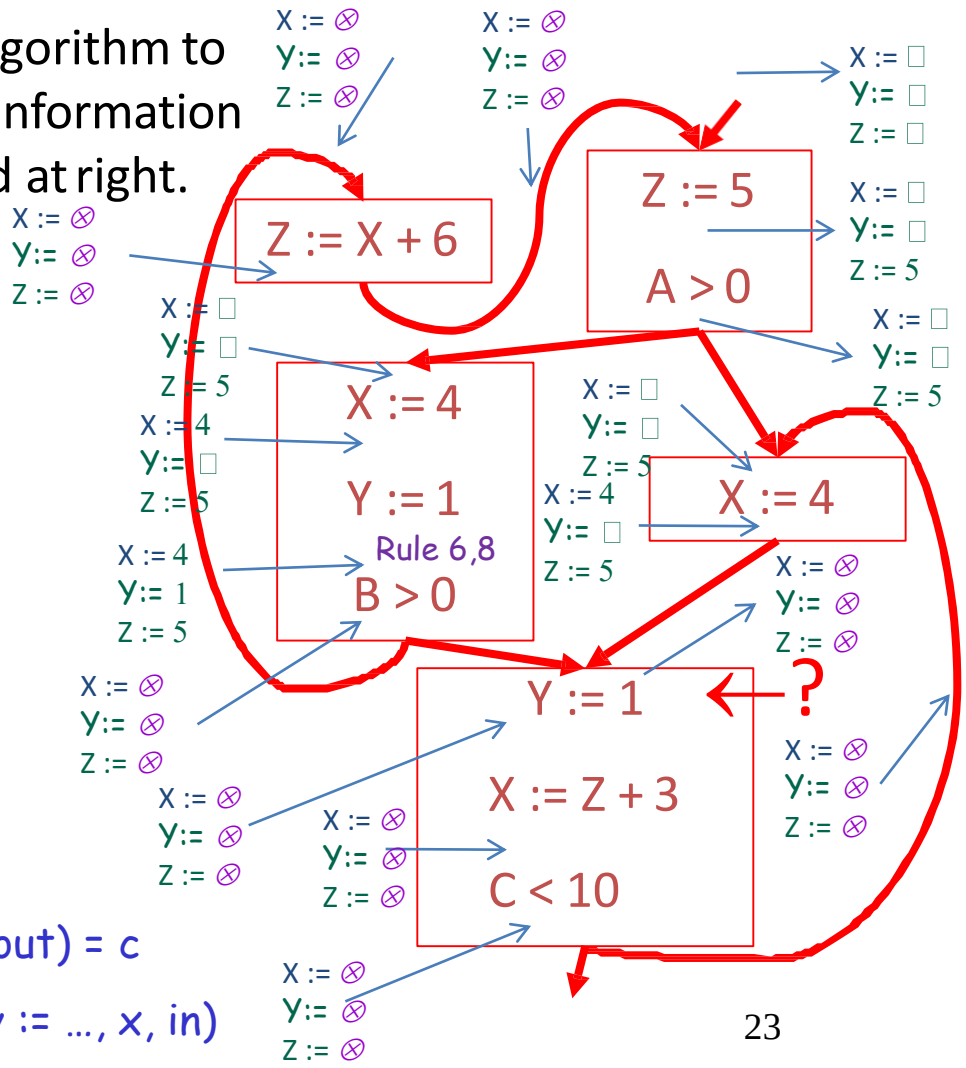
Rule 6: if c is a constant then C(x := c, x, out) = c

Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	□	1	□
○	4	□	5
○	4	1	5
○	4	□	□



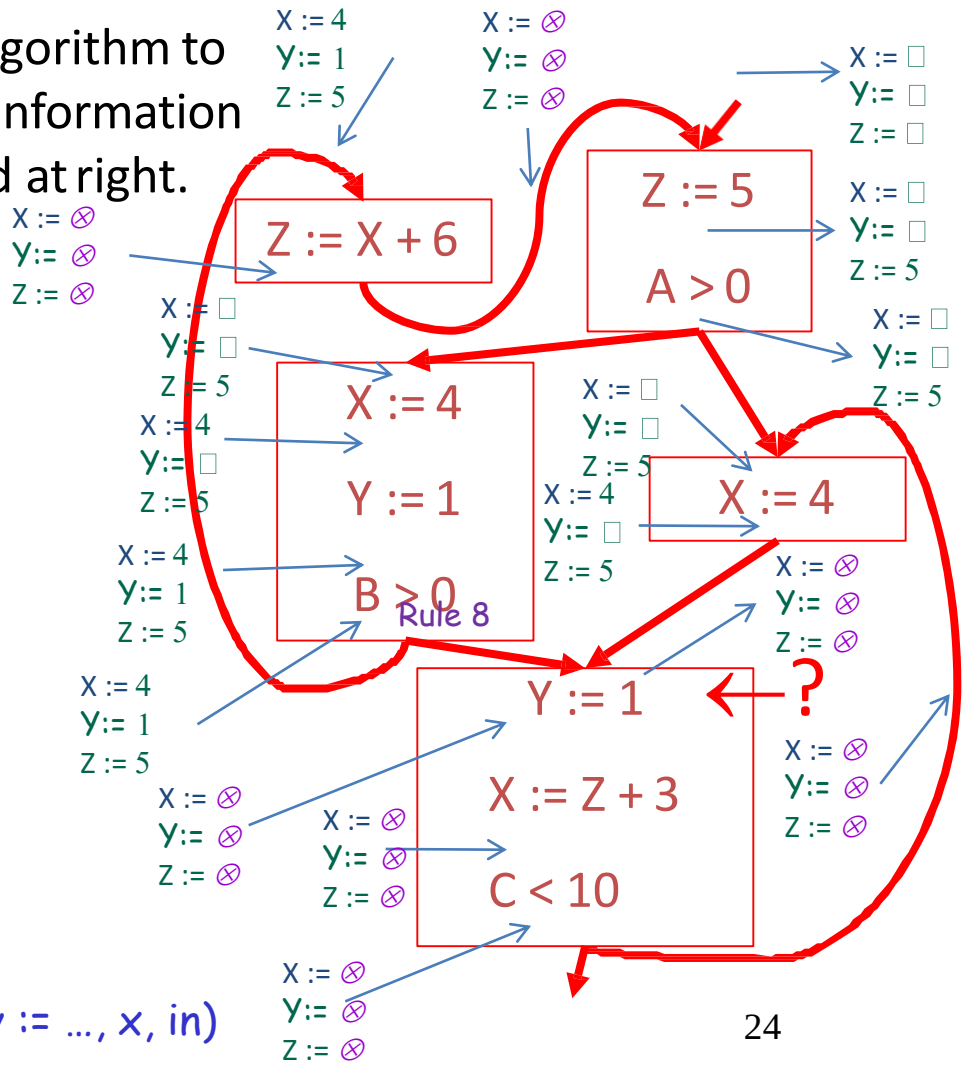
Rule 6: if c is a constant then C(x := c, x, out) = c

Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	□	1	□
○	4	□	5
○	4	1	5
○	4	□	□

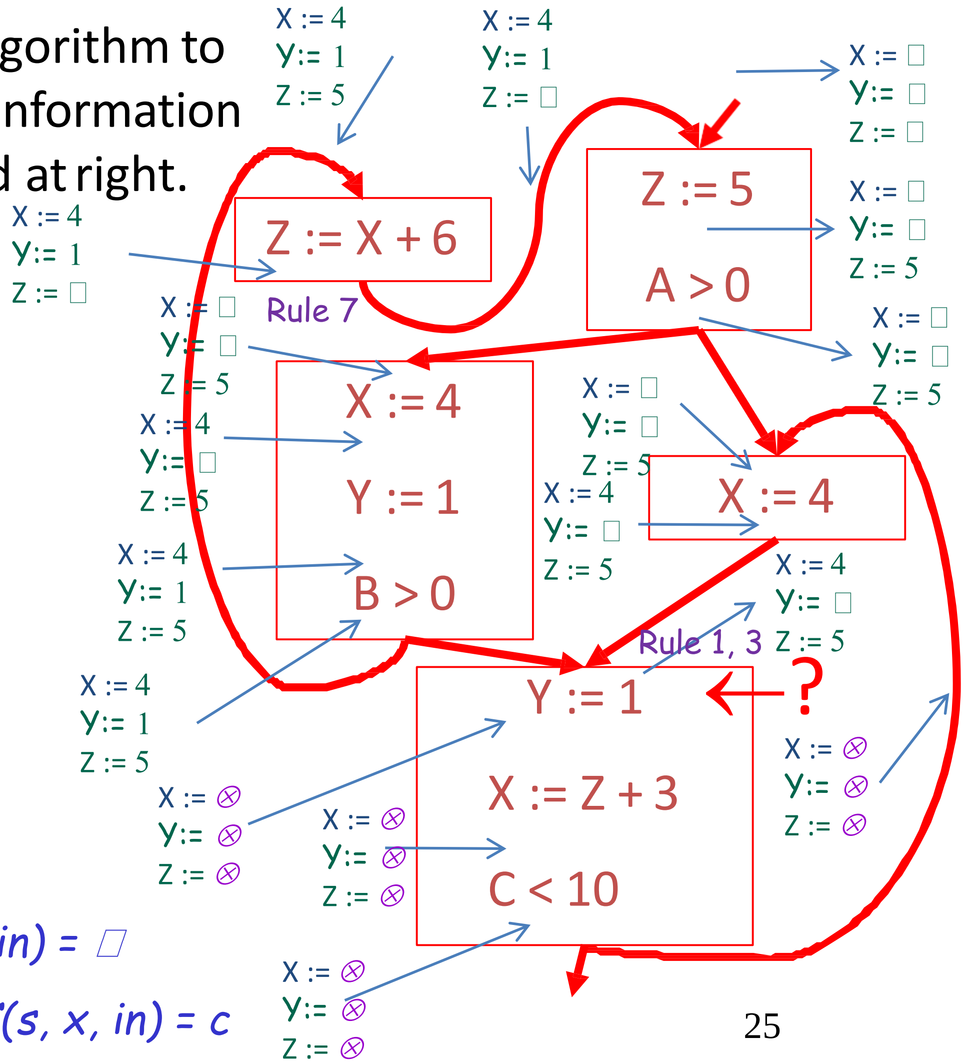


Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	□	1	□
○	4	□	5
○	4	1	5
○	4	□	□



Rule 1: *if* $\exists_i$ ($C(p_i, x, out)$ = □) *then* $C(s, x, in)$ = □

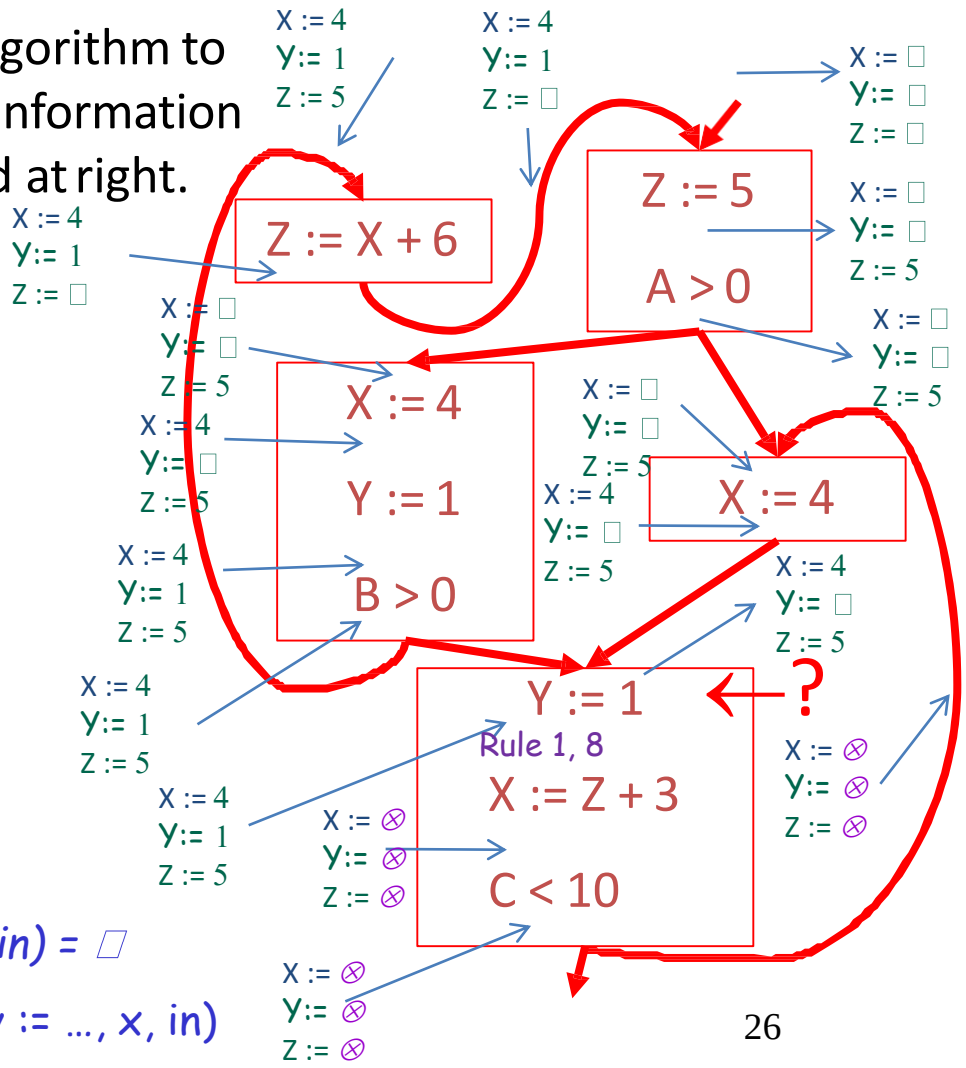
Rule 3: *if* $\forall_i$ ($C(p_i, x, out) = c$ *or* ⊗) *then* $C(s, x, in) = c$

Rule 7: $C(x := f(\ldots), x, out)$ = □

Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	□	1	□
○	4	□	5
○	4	1	5
○	4	□	□



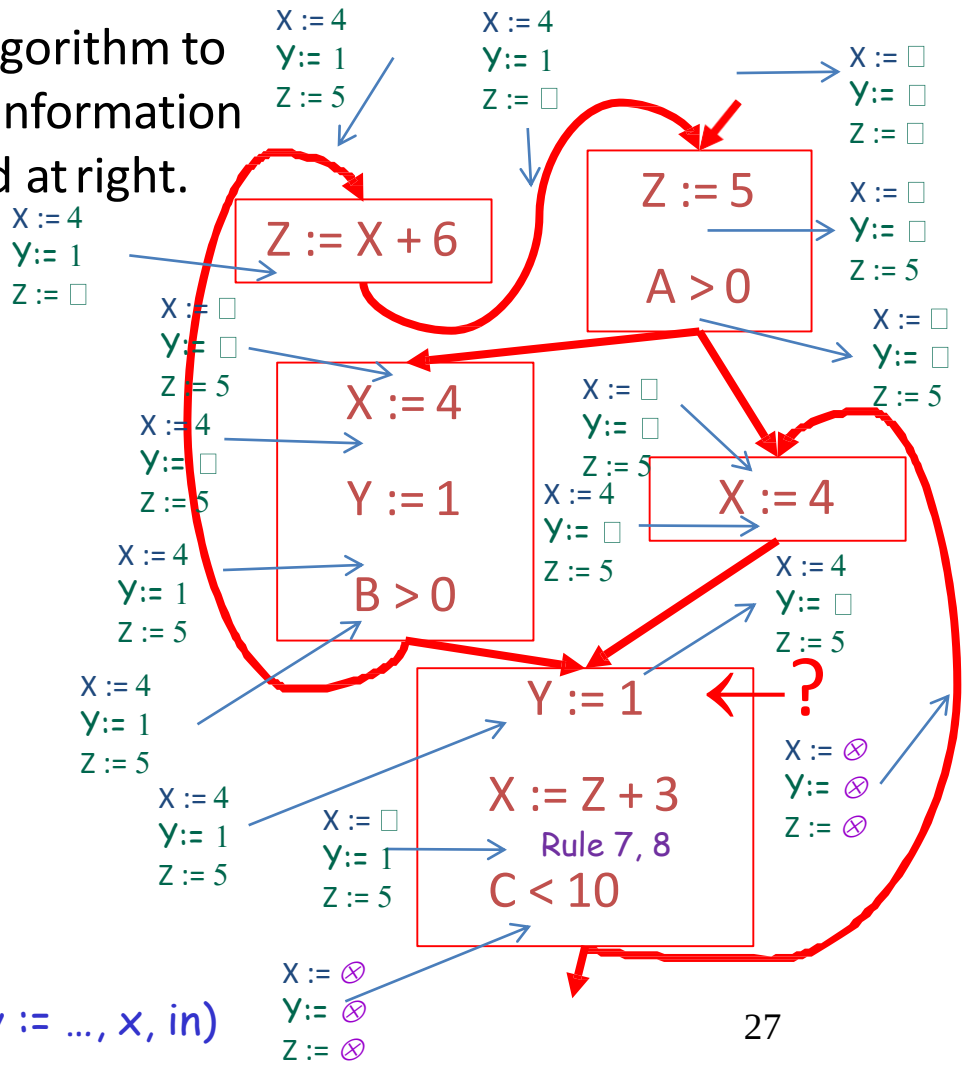
Rule 1: *if* $\exists_i$ $(C(p_i, x, out) = \square)$ *then* $C(s, x, in) = \square$

Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	□	1	□
○	4	□	5
○	4	1	5
○	4	□	□



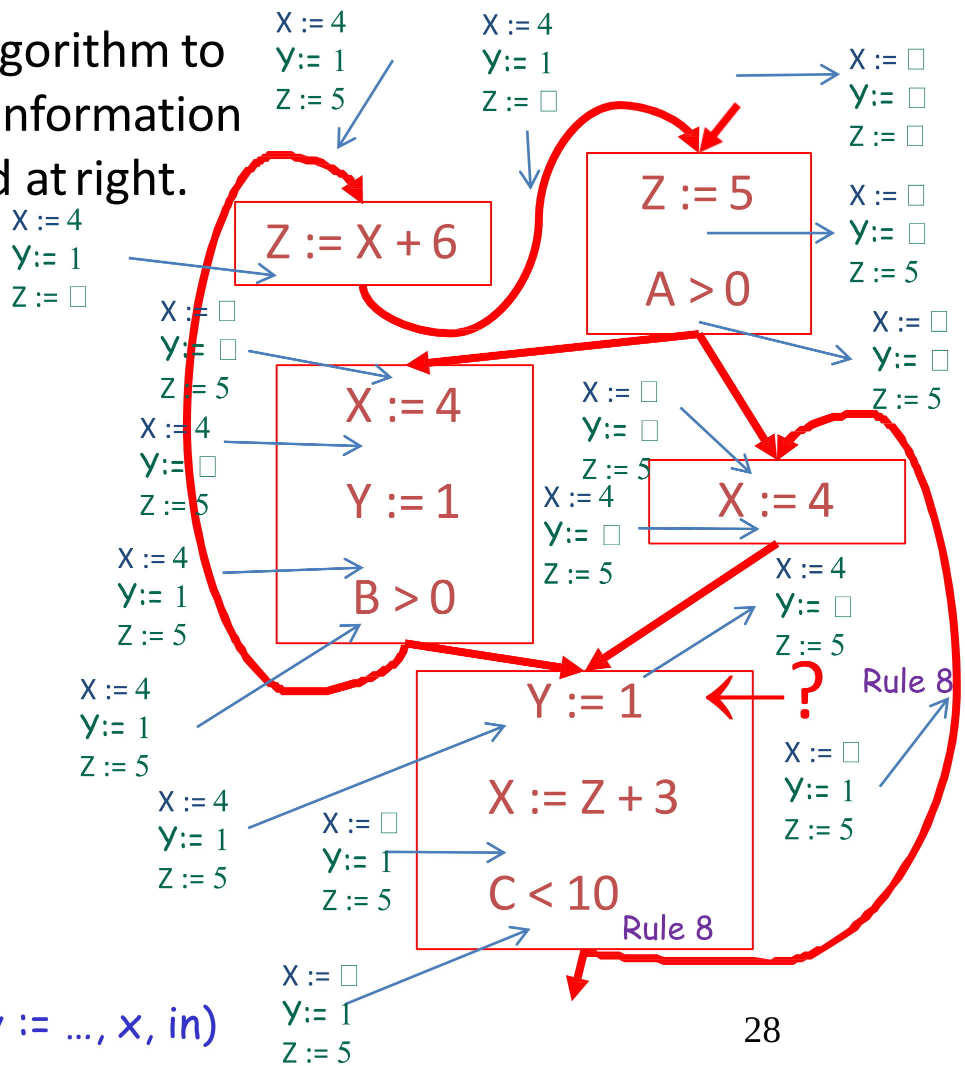
Rule 7: *C(x := f(…), x, out) = □*

Rule 8: if x <> y then *C(y := …, x, out) = C(y := …, x, in)*

Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

	X	Y	Z
○	□	1	□
○	4	□	5
○	4	1	5
○	4	□	□

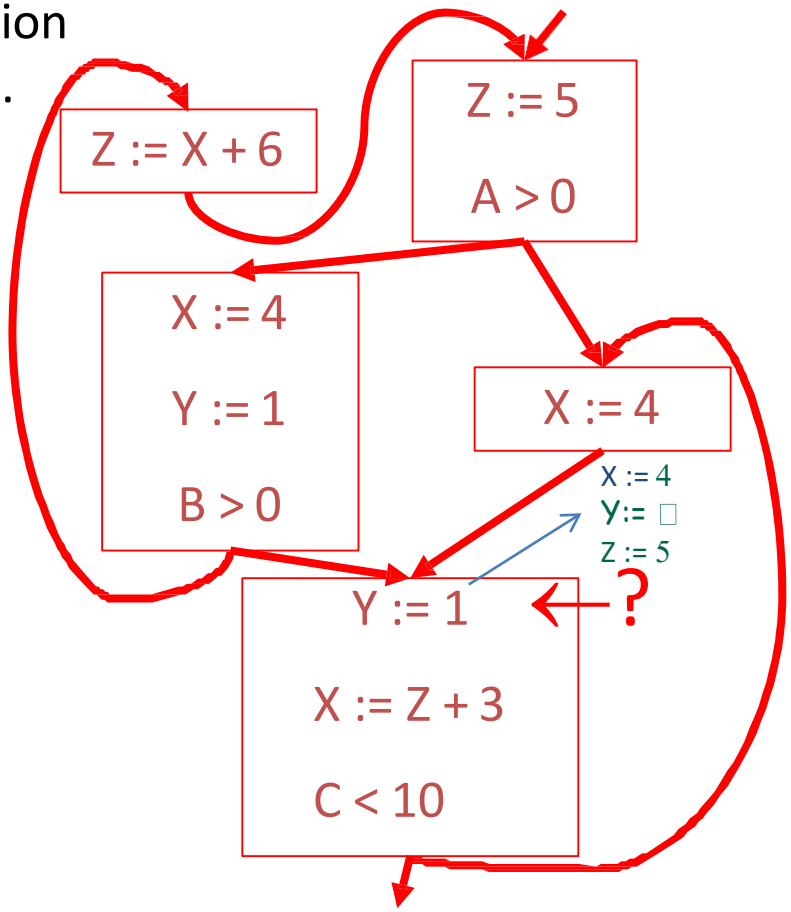


Rule 8: if x <> y then C(y := …, x, out) = C(y := …, x, in)

Answer!

After running the constant propagation algorithm to completion, choose the correct dataflow information for X, Y, and Z at the program point labeled at right.

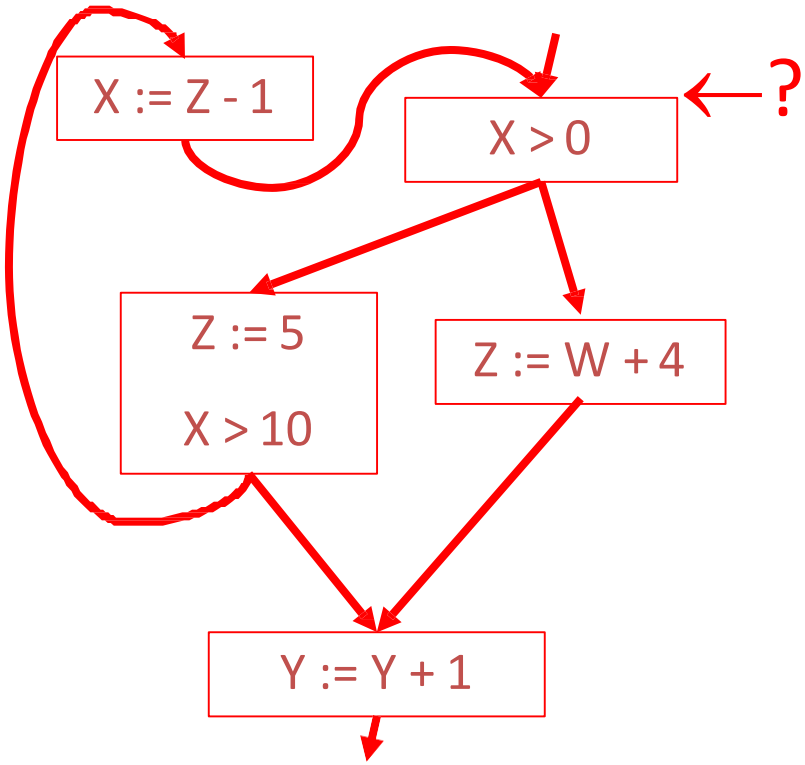
	X	Y	Z
○	□	1	□
○	4	□	5
○	4	1	5
○	4	□	□



Question?

After running the liveness analysis algorithm to completion, which of W, X, Y, and Z are live at the program point labeled at right? Assume all variables are dead on exit.

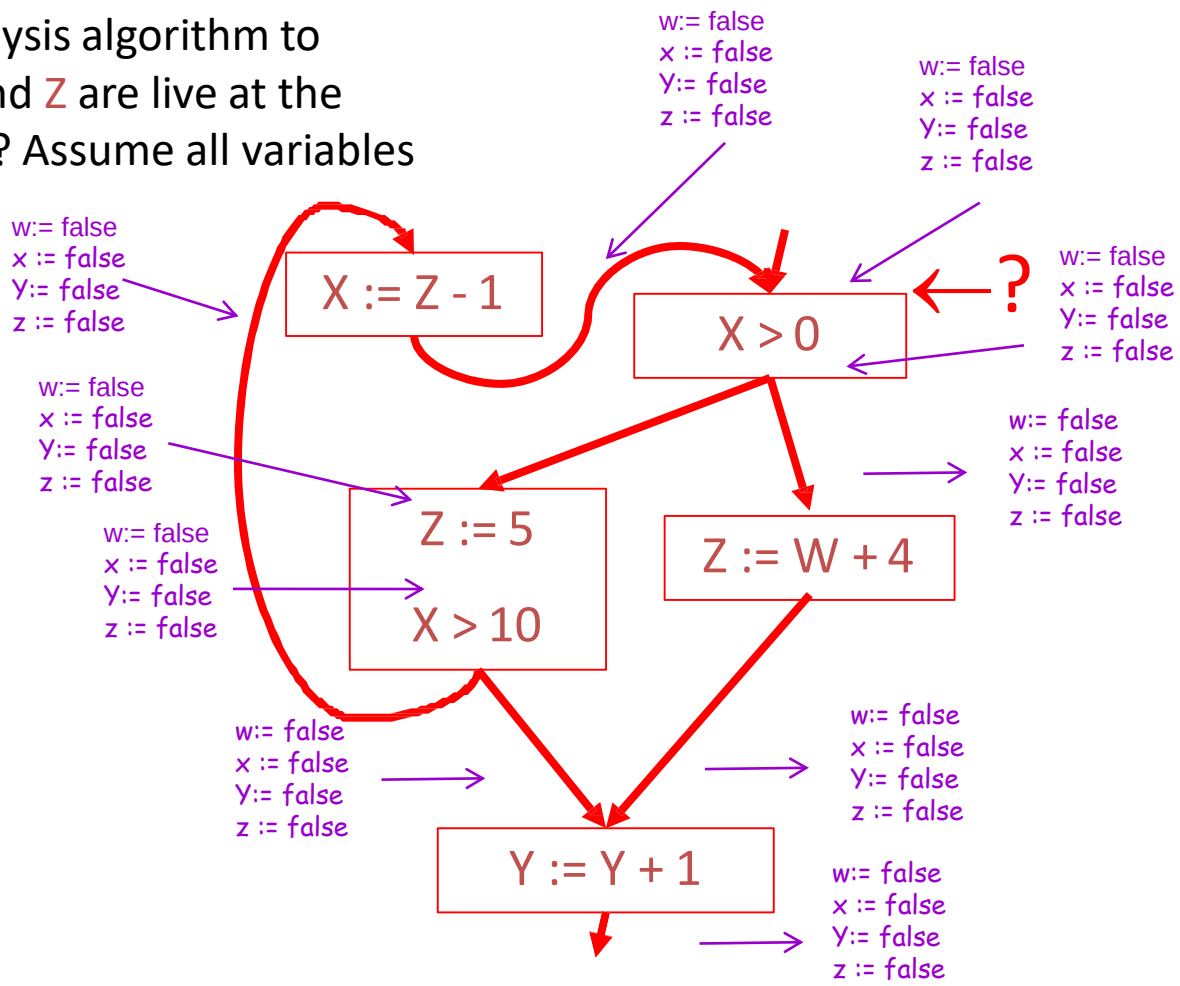
- ☐ W
- ☐ X
- ☐ Y
- ☐ Z



Question?

After running the liveness analysis algorithm to completion, which of W, X, Y, and Z are live at the program point labeled at right? Assume all variables are dead on exit.

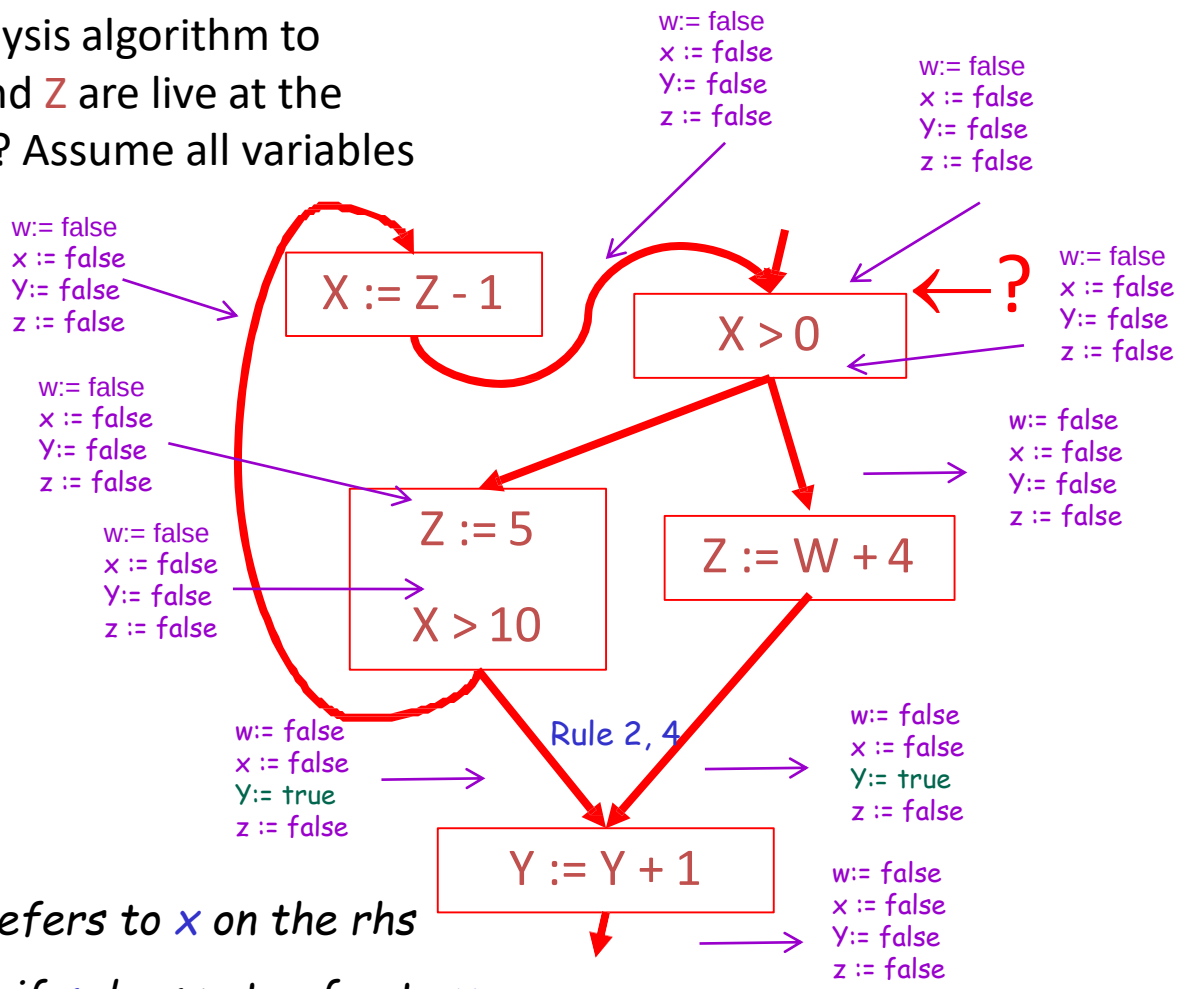
- ☐ W
- ☐ X
- ☐ Y
- ☐ Z



Question?

After running the liveness analysis algorithm to completion, which of W, X, Y, and Z are live at the program point labeled at right? Assume all variables are dead on exit.

- ☐ W
- ☐ X
- ☐ Y
- ☐ Z



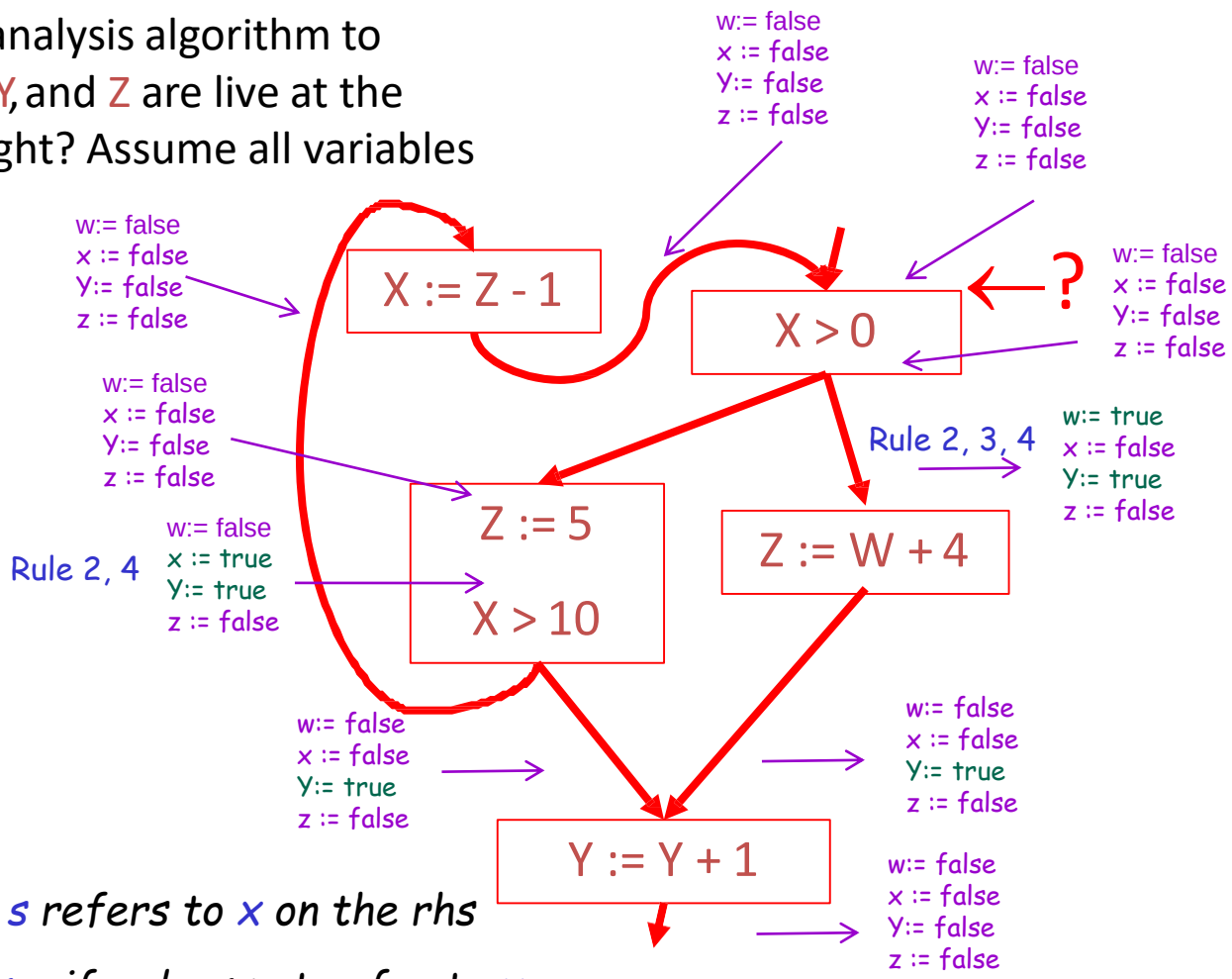
Rule 2: *L(s, x, in) = true if s refers to x on the rhs*

Rule 4: *L(s, x, in) = L(s, x, out) if s does not refer to x*

Question?

After running the liveness analysis algorithm to completion, which of W, X, Y, and Z are live at the program point labeled at right? Assume all variables are dead on exit.

- ☐ W
- ☐ X
- ☐ Y
- ☐ Z



Rule 2: *L(s, x, in) = true if s refers to x on the rhs*

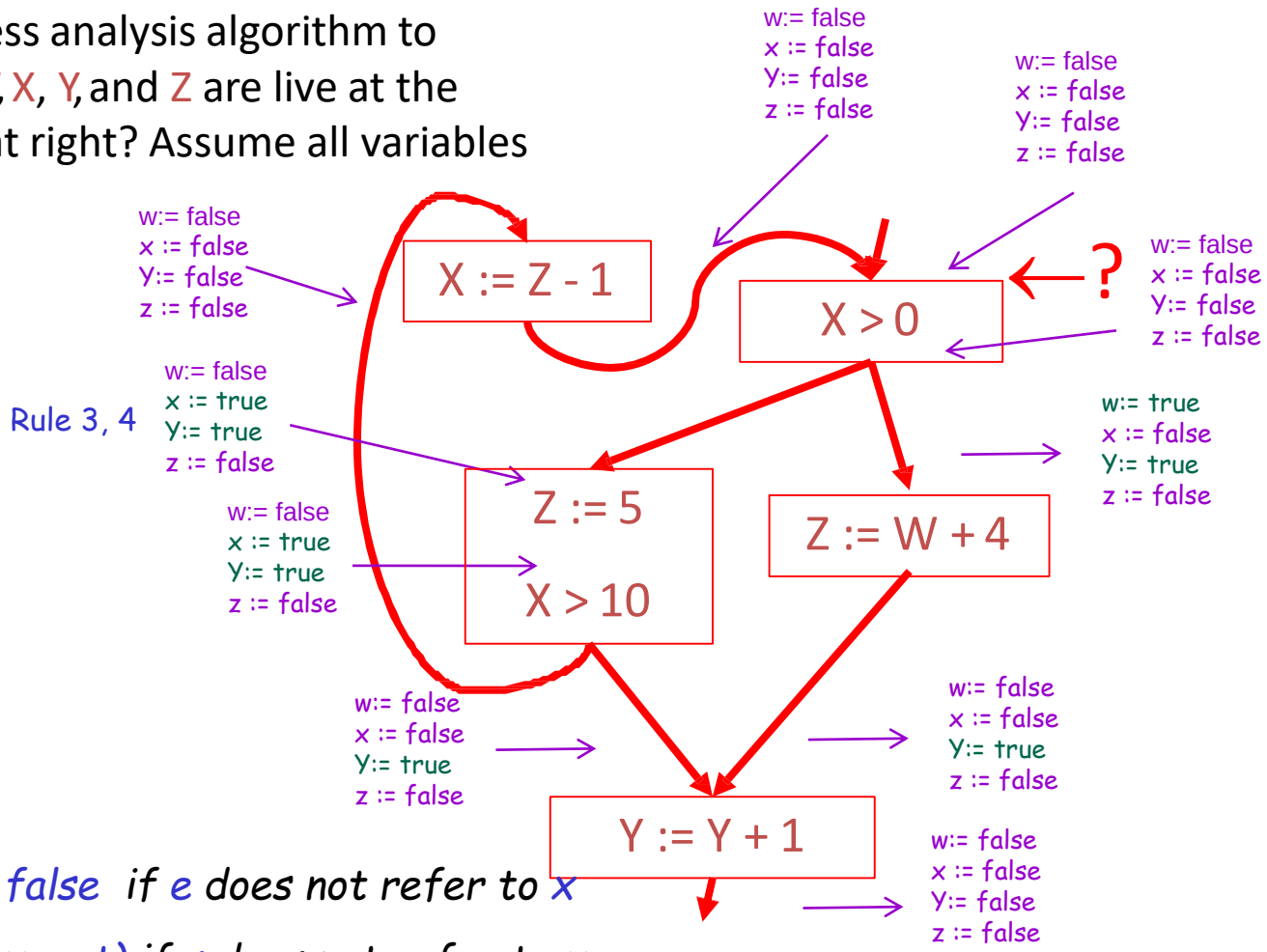
Rule 3: *L(x := e, x, in) = false if e does not refer to x*

Rule 4: *L(s, x, in) = L(s, x, out) if s does not refer to x*

Question?

After running the liveness analysis algorithm to completion, which of W, X, Y, and Z are live at the program point labeled at right? Assume all variables are dead on exit.

- ☐ W
- ☐ X
- ☐ Y
- ☐ Z



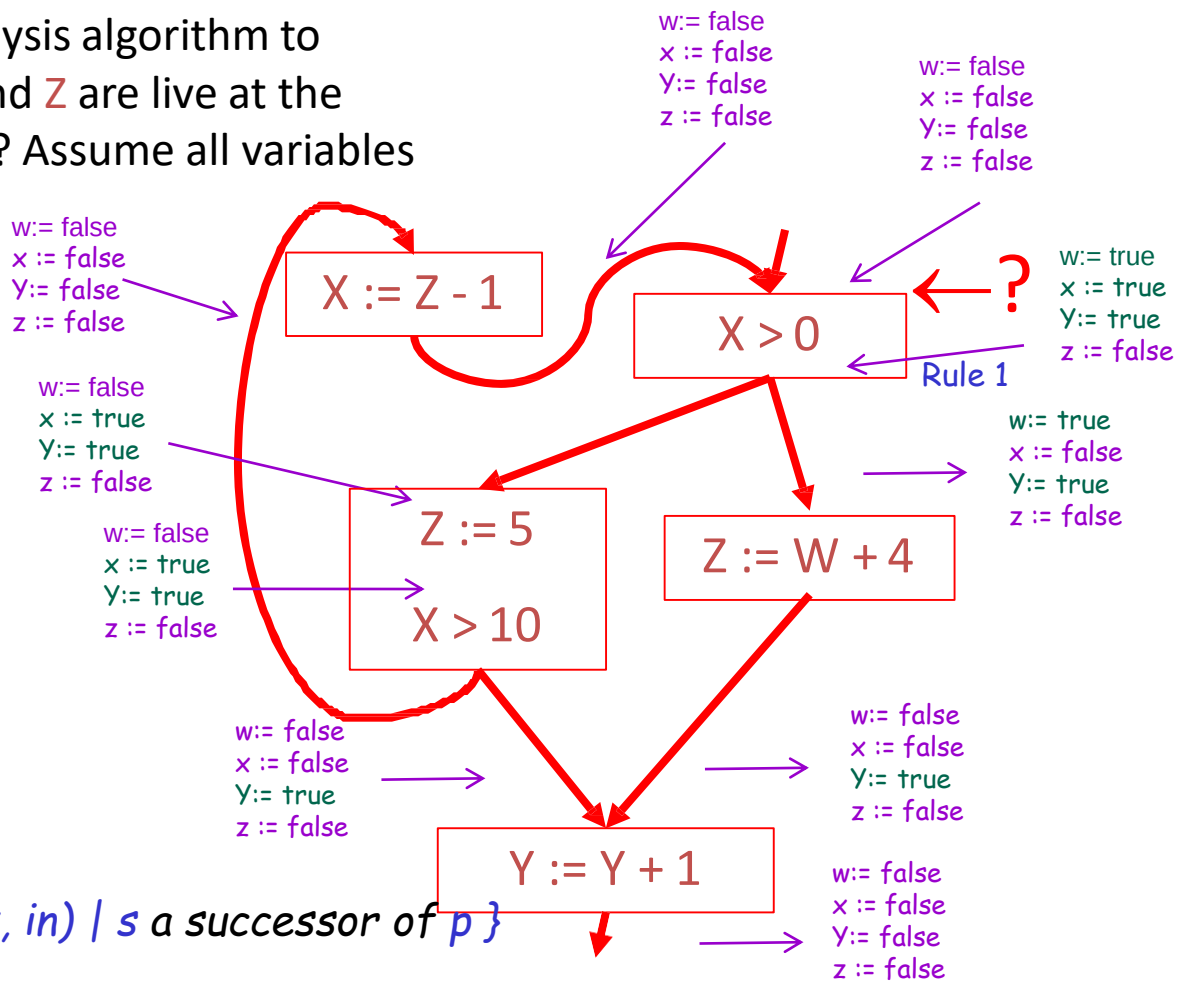
Rule 3: *L(x := e, x, in) = false if e does not refer to x*

Rule 4: *L(s, x, in) = L(s, x, out) if s does not refer to x*

Question?

After running the liveness analysis algorithm to completion, which of W, X, Y, and Z are live at the program point labeled at right? Assume all variables are dead on exit.

- ☐ W
- ☐ X
- ☐ Y
- ☐ Z

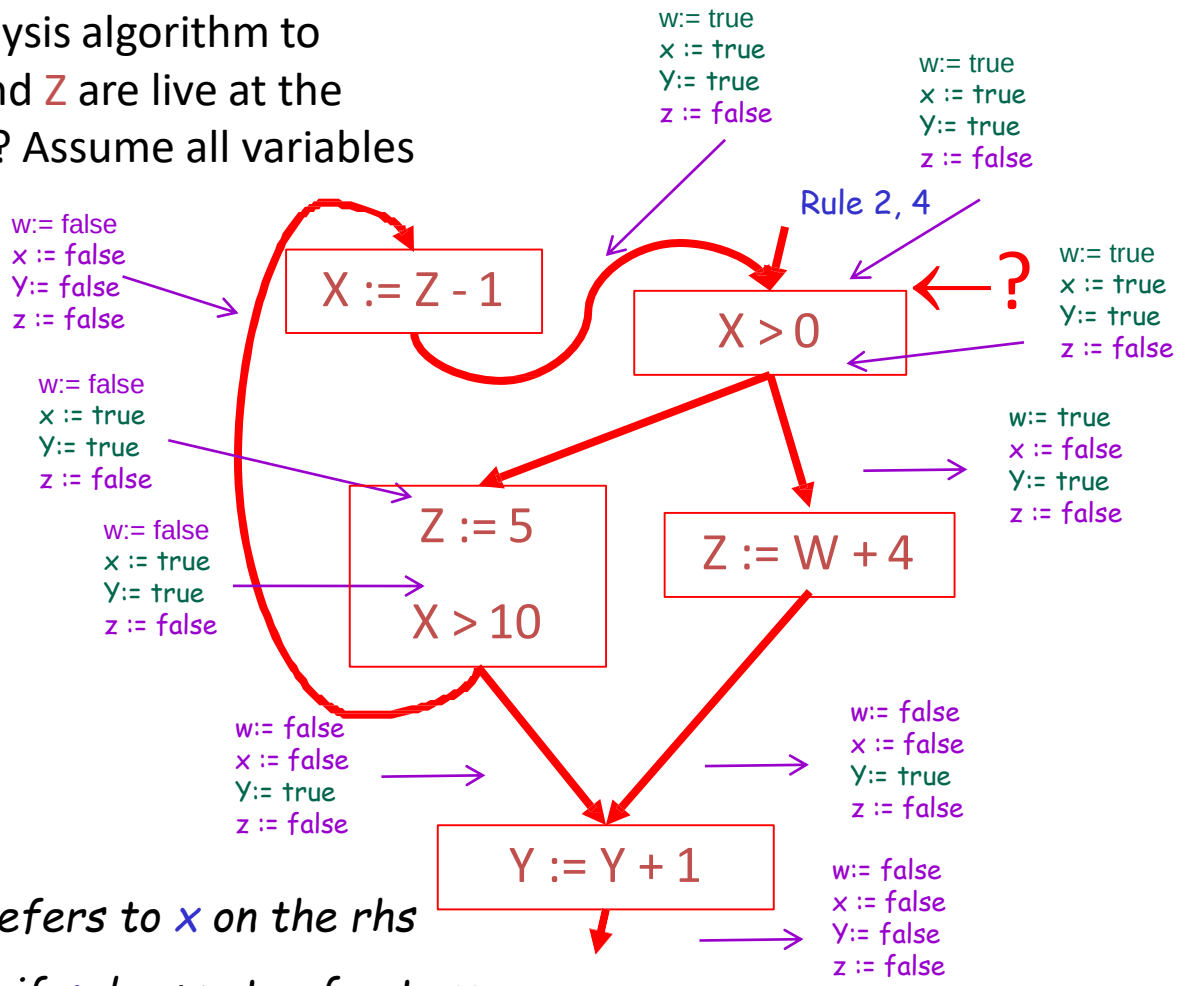


Rule 1: *L(p, x, out) = ∨{ L(s, x, in) | s a successor of p }*

Question?

After running the liveness analysis algorithm to completion, which of W, X, Y, and Z are live at the program point labeled at right? Assume all variables are dead on exit.

- ☐ W
- ☐ X
- ☐ Y
- ☐ Z



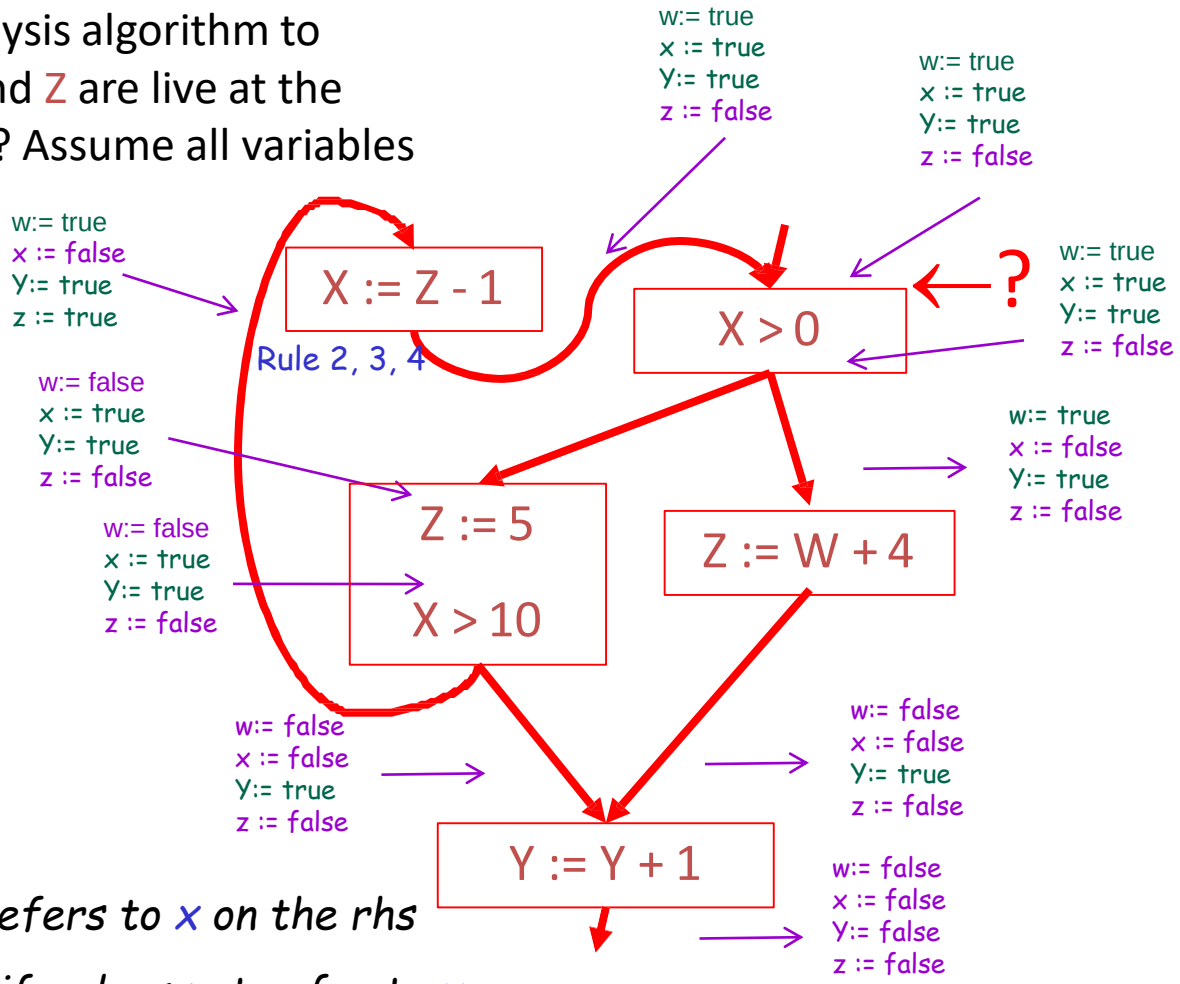
Rule 2: *L(s, x, in) = true if s refers to x on the rhs*

Rule 4: *L(s, x, in) = L(s, x, out) if s does not refer to x*

Question?

After running the liveness analysis algorithm to completion, which of W, X, Y, and Z are live at the program point labeled at right? Assume all variables are dead on exit.

- ☐ W
- ☐ X
- ☐ Y
- ☐ Z



Rule 2: *L(s, x, in) = true if s refers to x on the rhs*

Rule 3: *L(x := e, x, in) = false if e does not refer to x*

Rule 4: *L(s, x, in) = L(s, x, out) if s does not refer to x*

Answer!

After running the liveness analysis algorithm to completion, which of W, X, Y, and Z are live at the program point labeled at right? Assume all variables are dead on exit.

- ☐ W
- ☐ X
- ☐ Y
- ☐ Z