Proof logging for some interesting constraint propagation algorithms

Matthew McIlree

WHOOPS, Copenhagen, 23rd May 2024

University of Glasgow

Royal Academy of Engineering

Assuming you are happy with... :-)

- Proof logging being a useful thing

Assuming you are happy with... :-)

- Proof logging being a useful thing
- VeriPB proof rules

Assuming you are happy with... :-)

- Proof logging being a useful thing
- VeriPB proof rules
- Reifying PB constraints

Assuming you are happy with... :-)

- Proof logging being a useful thing
- VeriPB proof rules
- Reifying PB constraints
- Basic Ideas of Constraint Programming (Search, Propagation)

Assuming you are happy with... :-)

- Proof logging being a useful thing
- VeriPB proof rules
- Reifying PB constraints
- Basic Ideas of Constraint Programming (Search, Propagation)
- Encoding CP Variables for proofs (Binary and Direct Encoding)

Assuming you are happy with... :-)

- Proof logging being a useful thing
- VeriPB proof rules
- Reifying PB constraints
- Basic Ideas of Constraint Programming (Search, Propagation)
- Encoding CP Variables for proofs (Binary and Direct Encoding)
- The general idea for CP proof logging (RUP on backtrack, propagators log justifications)

vec_eq_tuple

visible

weighted_partial_alldiff

xor

zero_or_not_zero

zero_or_not_zero_vectors

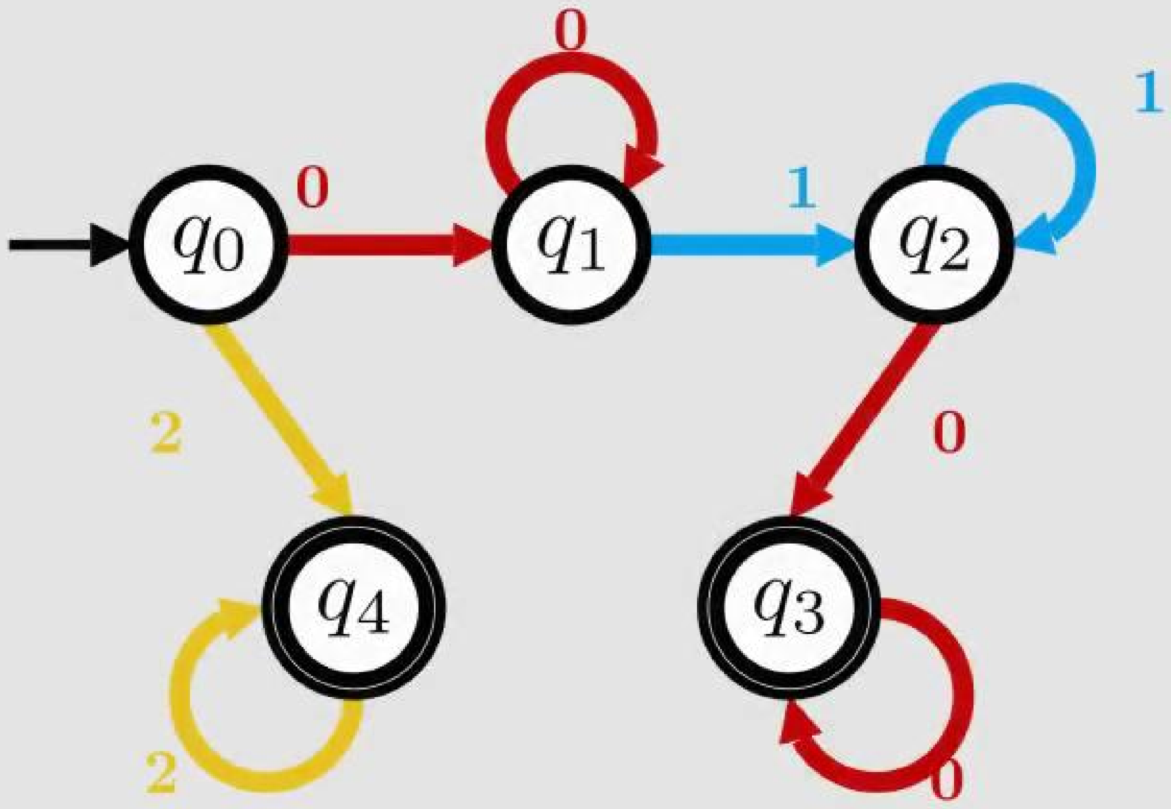
Smart Table

X	Y	Z
$< Y$	$\in \{1, 2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

Smart Table

X	Y	Z
$< Y$	$\in \{1, 2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

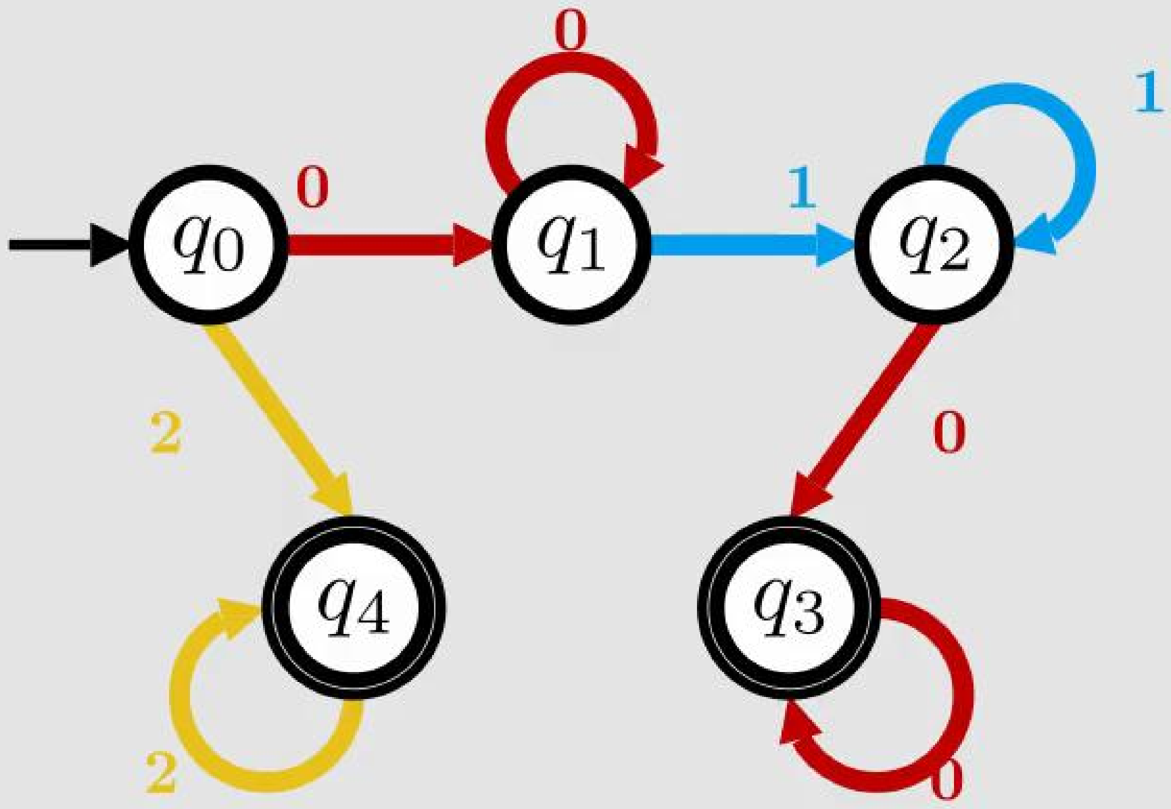
Regular



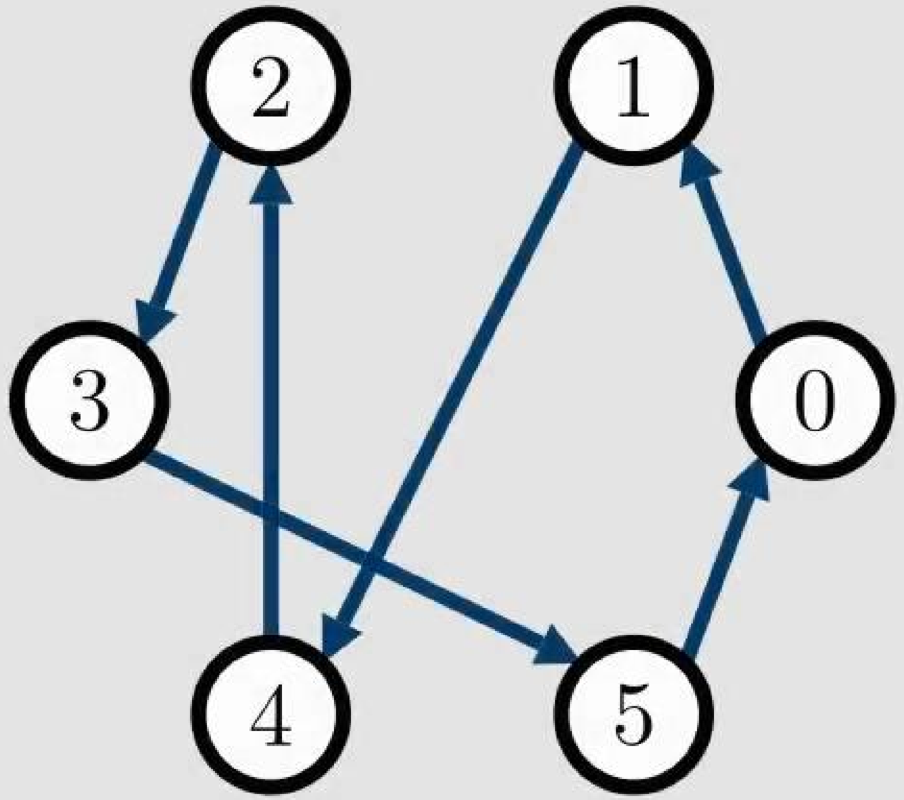
Smart Table

X	Y	Z
$< Y$	$\in \{1, 2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

Regular



Circuit



Smart Table PB Encoding

X	Y	Z
$< Y$	$\in \{1, 2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

Smart Table PB Encoding

X	Y	Z
$< Y$	$\in \{1, 2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

Smart Table PB Encoding

X	Y	Z
$< Y$	$\in \{1, 2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

$$X, Y, Z \in \{1, 2, 3\}$$

Smart Table PB Encoding

X	Y	Z
$< Y$	$\in \{1,2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

$$X, Y, Z \in \{1,2,3\}$$

$$e_{X<Y} \iff -x_{b0} - 2x_{b1} - 4x_{b2} + y_{b0} + 2y_{b1} + 4y_{b2} \geq 1$$

Smart Table PB Encoding

X	Y	Z
$< Y$	$\in \{1, 2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

$$X, Y, Z \in \{1, 2, 3\}$$

$$8 \cdot \overline{e_{X<Y}} + -x_{b0} - 2x_{b1} - 4x_{b2} + y_{b0} + 2y_{b1} + 4y_{b2} \geq 1$$

Smart Table PB Encoding

X	Y	Z
$< Y$	$\in \{1, 2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

$$X, Y, Z \in \{1, 2, 3\}$$

$$8 \cdot \overline{e_{X<Y}} + -x_{b0} - 2x_{b1} - 4x_{b2} + y_{b0} + 2y_{b1} + 4y_{b2} \geq 1$$

$$8 \cdot e_{X<Y} + x_{b0} + 2x_{b1} + 4x_{b2} - y_{b0} - 2y_{b1} - 4y_{b2} \geq 0$$

Smart Table PB Encoding

X	Y	Z
$< Y$	$\in \{1, 2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

$$X, Y, Z \in \{1, 2, 3\}$$

$$e_{X<Y} \iff -x_{b0} - 2x_{b1} - 4x_{b2} + y_{b0} + 2y_{b1} + 4y_{b2} \geq 1$$

Smart Table PB Encoding

X	Y	Z
$< Y$	$\in \{1,2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

$$X, Y, Z \in \{1,2,3\}$$

$$e_{X<Y} \iff -x_{b0} - 2x_{b1} - 4x_{b2} + y_{b0} + 2y_{b1} + 4y_{b2} \geq 1$$

$$e_{Y\in\{1,2\}} \iff \dots$$

Smart Table PB Encoding

X	Y	Z
$< Y$	$\in \{1,2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

$$X, Y, Z \in \{1, 2, 3\}$$

$$e_{X<Y} \iff -x_{b0} - 2x_{b1} - 4x_{b2} + y_{b0} + 2y_{b1} + 4y_{b2} \geq 1$$

$$e_{Y\in\{1,2\}} \iff \dots$$

$$e_{Z=3} \iff \dots$$

Smart Table PB Encoding

X	Y	Z
$< Y$	$\in \{1,2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

$$X, Y, Z \in \{1, 2, 3\}$$

$e_{X<Y} \iff -x_{b0} - 2x_{b1} - 4x_{b2} + y_{b0} + 2y_{b1} + 4y_{b2} \geq 1$

$e_{Y\in\{1,2\}} \iff \dots$

$e_{Z=3} \iff \dots$

$e_{X\neq 1} \iff \dots$

$e_{Y=X} \iff \dots$

$e_{Z\leq Y} \iff \dots$

Smart Table PB Encoding

X	Y	Z
$< Y$	$\in \{1,2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

$$X, Y, Z \in \{1, 2, 3\}$$

$$e_{X<Y} \iff -x_{b0} - 2x_{b1} - 4x_{b2} + y_{b0} + 2y_{b1} + 4y_{b2} \geq 1$$
$$e_{Y\in\{1,2\}} \iff \dots$$
$$e_{Z=3} \iff \dots$$
$$e_{X\neq 1} \iff \dots$$
$$e_{Y=X} \iff \dots$$
$$e_{Z\leq Y} \iff \dots$$
$$s_1 \iff e_{X<Y} + e_{Y\in\{1,2\}} + e_{Z=3} \geq 3$$

Smart Table PB Encoding

X	Y	Z
$< Y$	$\in \{1, 2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

$$X, Y, Z \in \{1, 2, 3\}$$

$$e_{X<Y} \iff -x_{b0} - 2x_{b1} - 4x_{b2} + y_{b0} + 2y_{b1} + 4y_{b2} \geq 1$$

$$e_{Y\in\{1,2\}} \iff \dots$$

$$e_{Z=3} \iff \dots$$

$$e_{X\neq 1} \iff \dots$$

$$e_{Y=X} \iff \dots$$

$$e_{Z\leq Y} \iff \dots$$

$$s_1 \iff e_{X<Y} + e_{Y\in\{1,2\}} + e_{Z=3} \geq 3$$

$$s_2 \iff e_{X\neq 1} + e_{Y=X} + e_{Z\leq Y} \geq 3$$

Smart Table PB Encoding

X	Y	Z
$< Y$	$\in \{1, 2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

$$X, Y, Z \in \{1, 2, 3\}$$

$e_{X<Y} \iff -x_{b0} - 2x_{b1} - 4x_{b2} + y_{b0} + 2y_{b1} + 4y_{b2} \geq 1$

$e_{Y\in\{1,2\}} \iff \dots$

$e_{Z=3} \iff \dots$

$e_{X\neq 1} \iff \dots$

$e_{Y=X} \iff \dots$

$e_{Z\leq Y} \iff \dots$

$s_1 \iff e_{X<Y} + e_{Y\in\{1,2\}} + e_{Z=3} \geq 3$

$s_2 \iff e_{X\neq 1} + e_{Y=X} + e_{Z\leq Y} \geq 3$

$s_1 + s_2 \geq 1$

Justifying Smart Table

X	Y	Z
$< Y$	$\in \{1,2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

$$X, Y, Z \in \{1, 2, 3\}$$

$\cdots$

$s_1 \iff e_{X<Y} + e_{Y\in\{1,2\}} + e_{Z=3} \geq 3$

$s_2 \iff e_{X\neq 1} + e_{Y=X} + e_{Z\leq Y} \geq 3$

$s_1 + s_2 \geq 1$

Justifying Smart Table

X	Y	Z
$< Y$	$\in \{1, 2\}$	$= 3$
$\neq 1$	$= X$	$\leq Y$

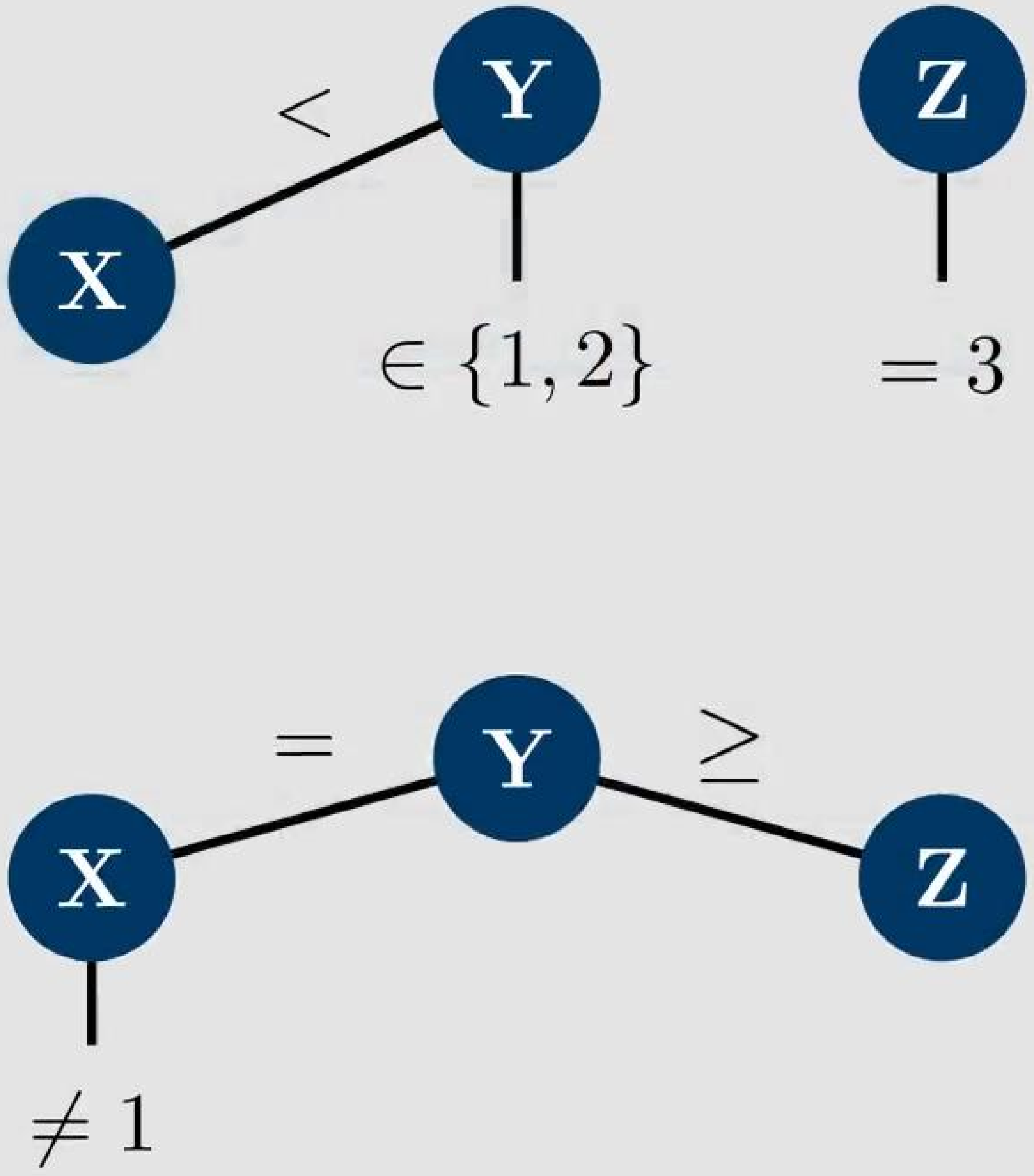
$\dots$

$s_1 \iff e_{X<Y} + e_{Y\in\{1,2\}} + e_{Z=3} \geq 3$

$s_2 \iff e_{X\neq 1} + e_{Y=X} + e_{Z\leq Y} \geq 3$

$s_1 + s_2 \geq 1$

Justifying Smart Table



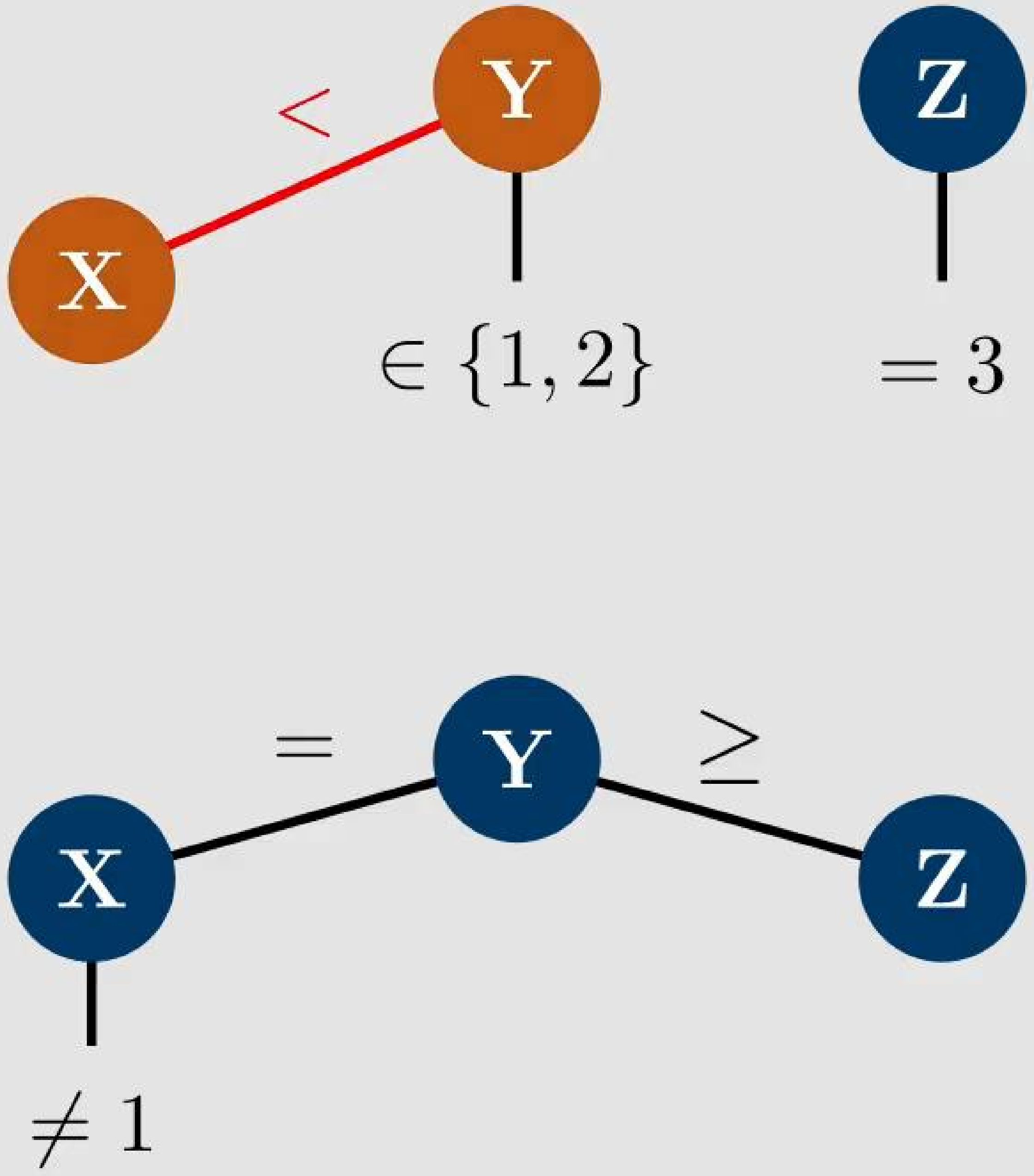
. . .

$s_1 \iff e_{X<Y} + e_{Y\in\{1,2\}} + e_{Z=3} \geq 3$

$s_2 \iff e_{X\neq 1} + e_{Y=X} + e_{Z\leq Y} \geq 3$

$s_1 + s_2 \geq 1$

Justifying Smart Table



...

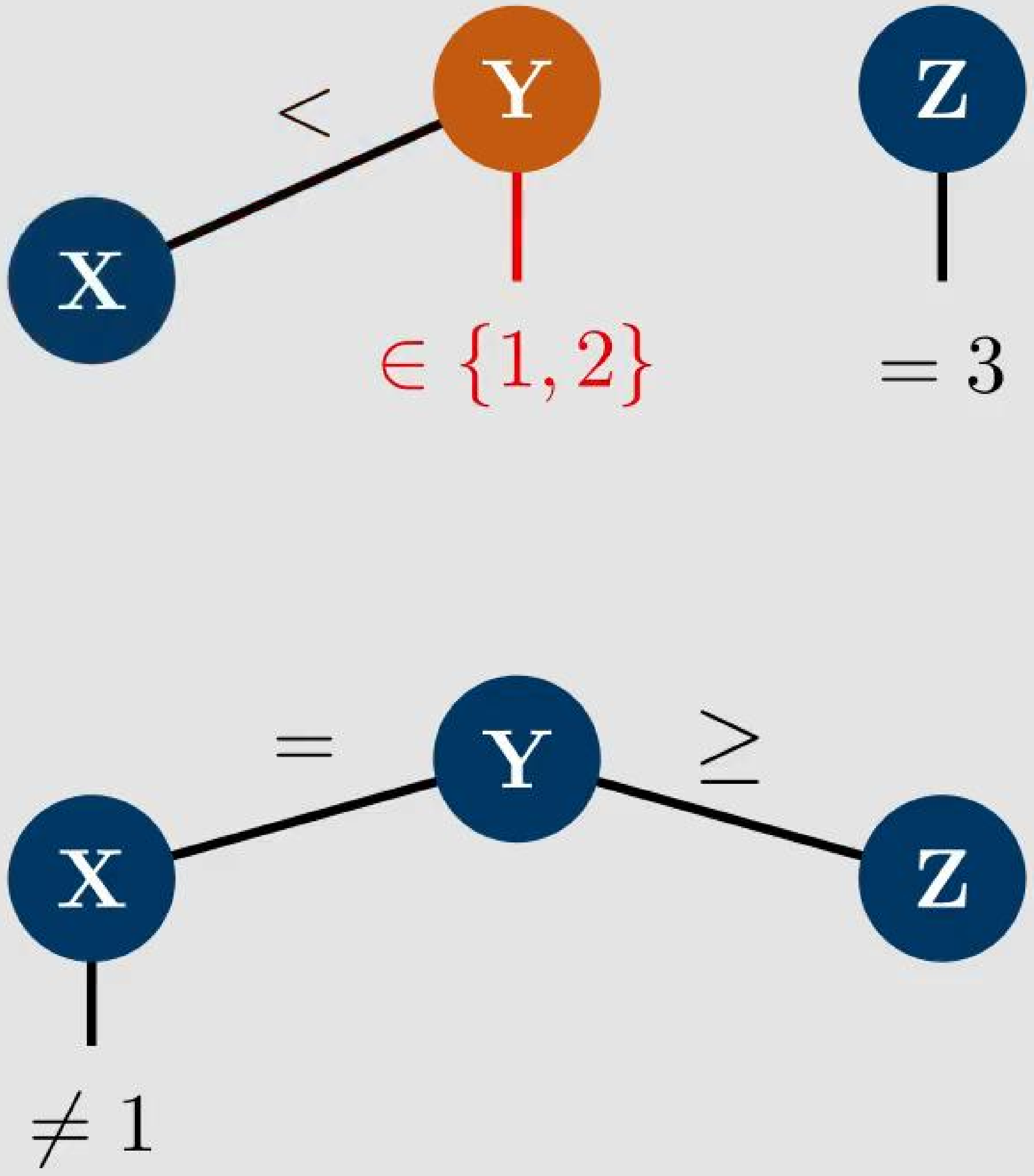
$s_1 \iff e_{X<Y} + e_{Y\in\{1,2\}} + e_{Z=3} \geq 3$

$s_2 \iff e_{X\neq 1} + e_{Y=X} + e_{Z\leq Y} \geq 3$

$s_1 + s_2 \geq 1$

RUP $s_1 \implies \bar{y}_{=1};$

Justifying Smart Table



$\cdots$

$s_1 \iff e_{X<Y} + e_{Y \in \{1,2\}} + e_{Z=3} \geq 3$

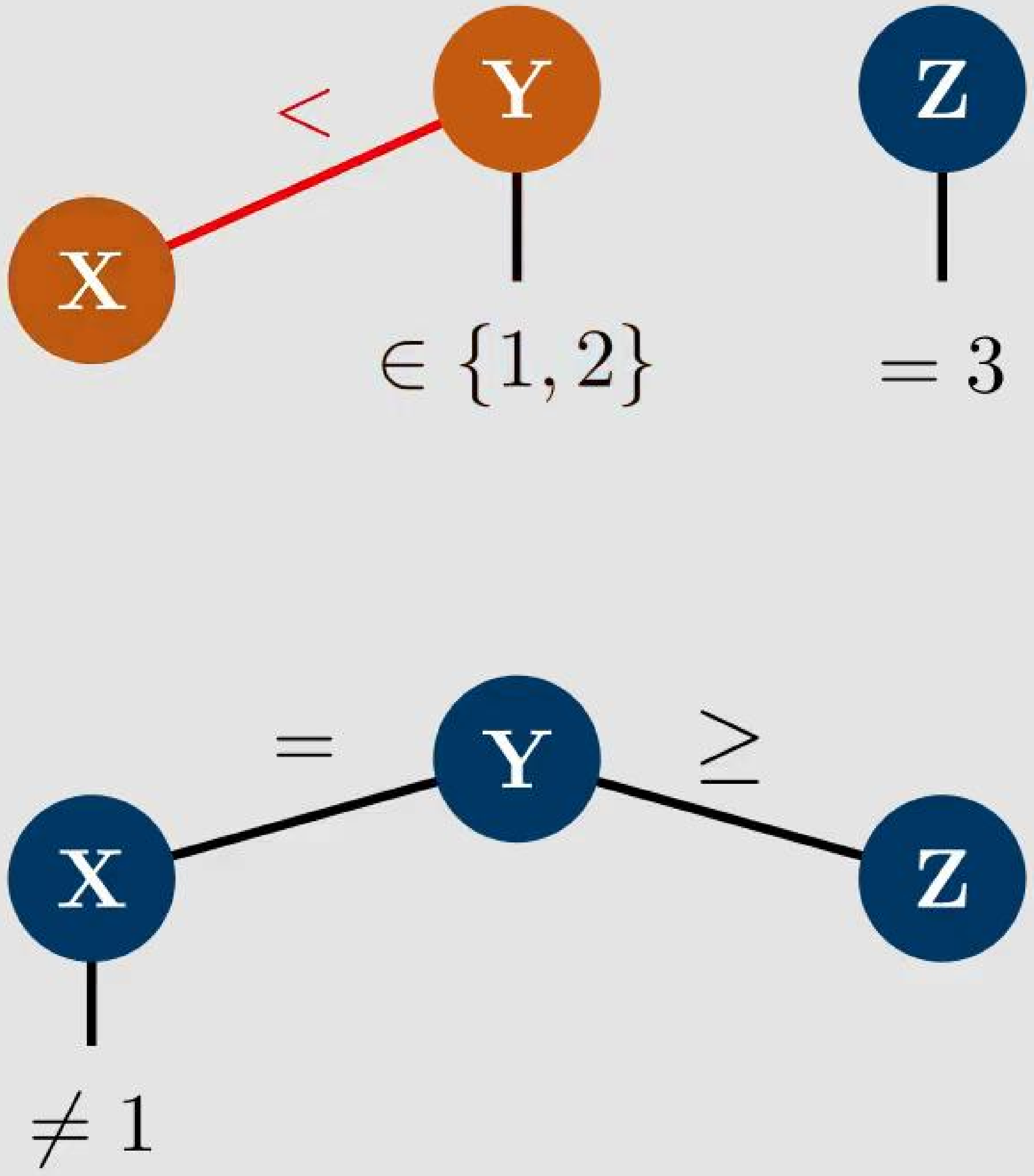
$s_2 \iff e_{X \neq 1} + e_{Y=X} + e_{Z \leq Y} \geq 3$

$s_1 + s_2 \geq 1$

RUP $s_1 \implies \bar{y}_{=1}$;

RUP $s_1 \implies \bar{y}_{=3}$;

Justifying Smart Table



...

$s_1 \iff e_{X<Y} + e_{Y\in\{1,2\}} + e_{Z=3} \geq 3$

$s_2 \iff e_{X\neq 1} + e_{Y=X} + e_{Z\leq Y} \geq 3$

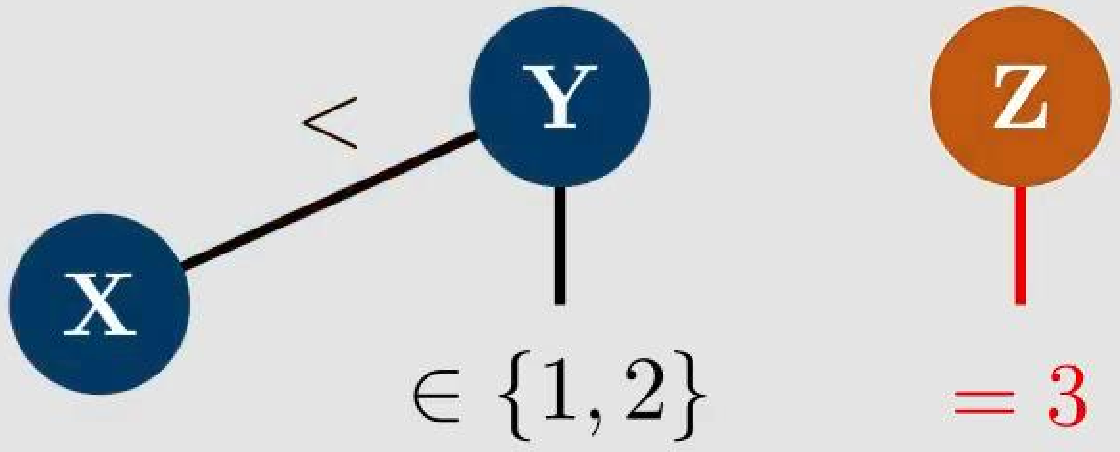
$s_1 + s_2 \geq 1$

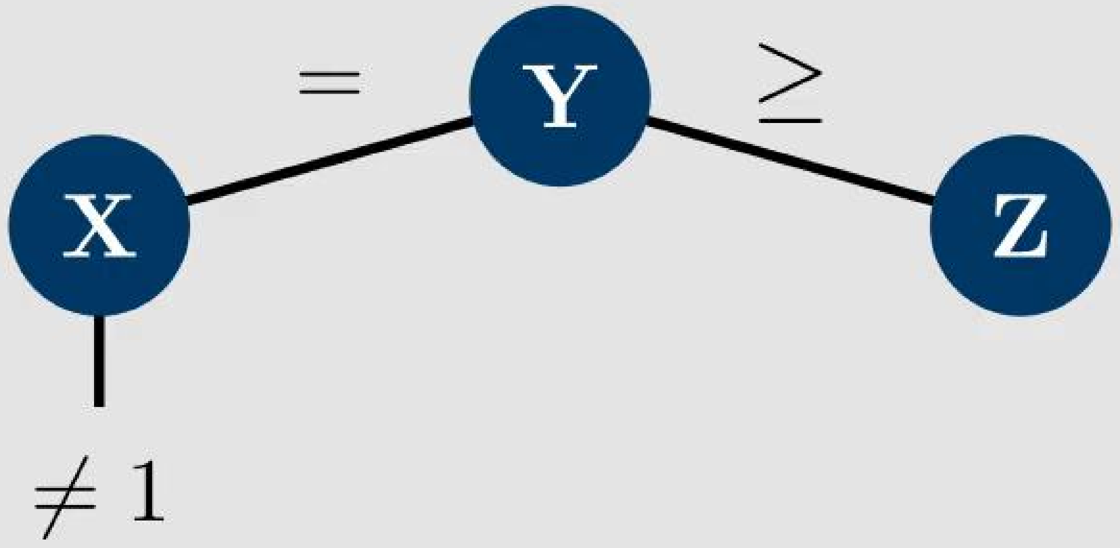
RUP $s_1 \implies \bar{y}_{=1}$;

RUP $s_1 \implies \bar{y}_{=3}$;

RUP $s_1 \implies x_{=1}$;

Justifying Smart Table





...

$s_1 \iff e_{X<Y} + e_{Y \in \{1,2\}} + e_{Z=3} \geq 3$

$s_2 \iff e_{X \neq 1} + e_{Y=X} + e_{Z \leq Y} \geq 3$

$s_1 + s_2 \geq 1$

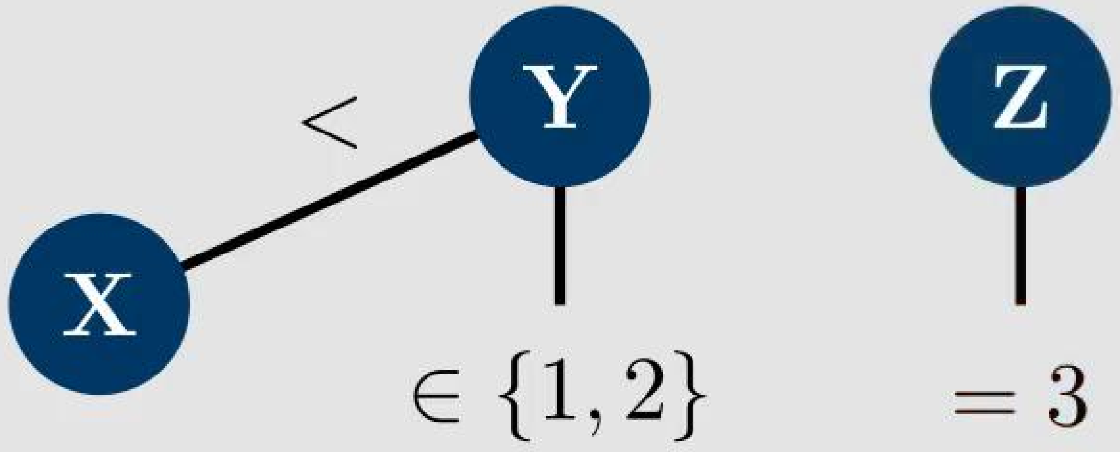
RUP $s_1 \implies \bar{y}_{=1}$;

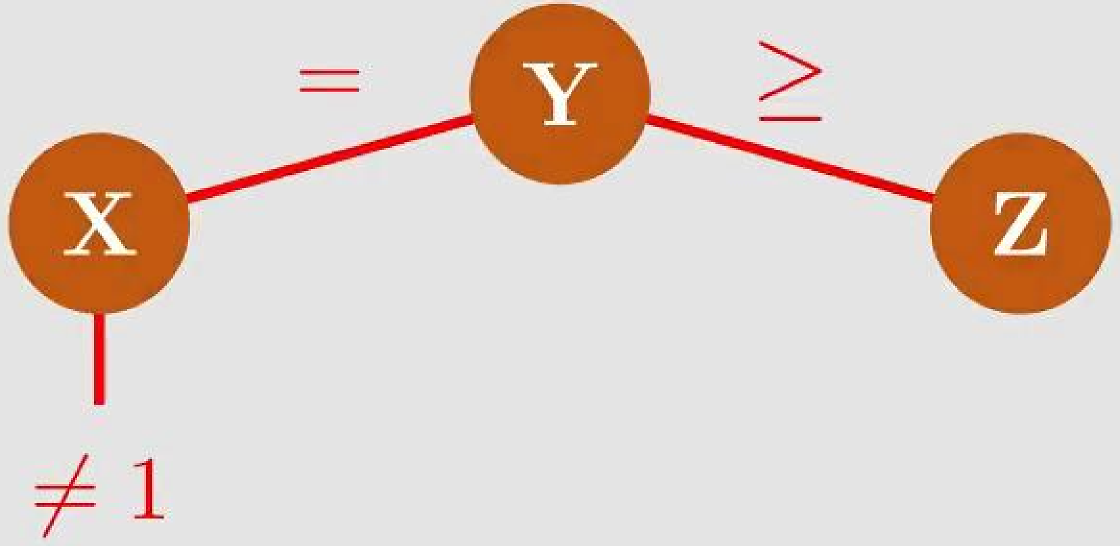
RUP $s_1 \implies \bar{y}_{=3}$;

RUP $s_1 \implies x_{=1}$;

RUP $s_1 \implies z_{=3}$;

Justifying Smart Table





...

$s_1 \iff e_{X<Y} + e_{Y\in\{1,2\}} + e_{Z=3} \geq 3$

$s_2 \iff e_{X\neq 1} + e_{Y=X} + e_{Z\leq Y} \geq 3$

$s_1 + s_2 \geq 1$

RUP $s_1 \implies \bar{y}_{=1}$;

RUP $s_1 \implies \bar{y}_{=3}$;

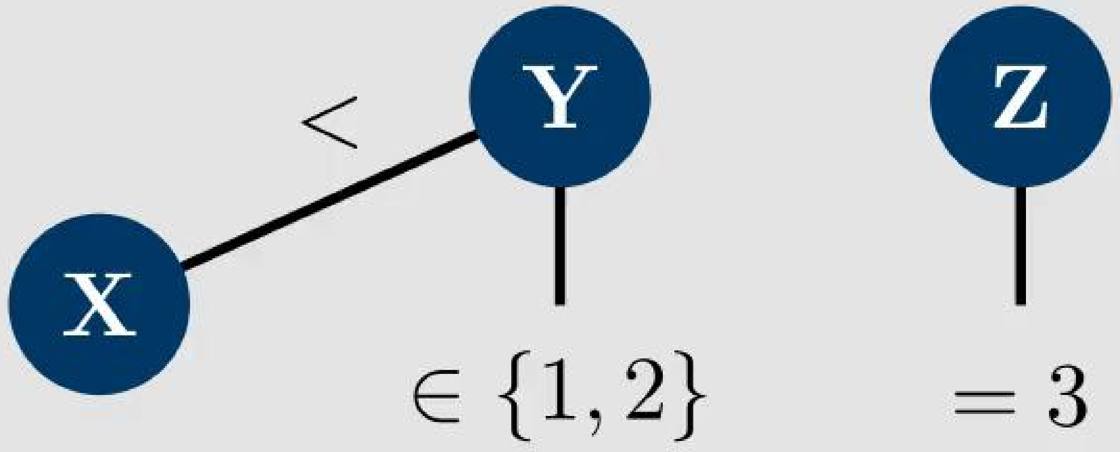
RUP $s_1 \implies x_{=1}$;

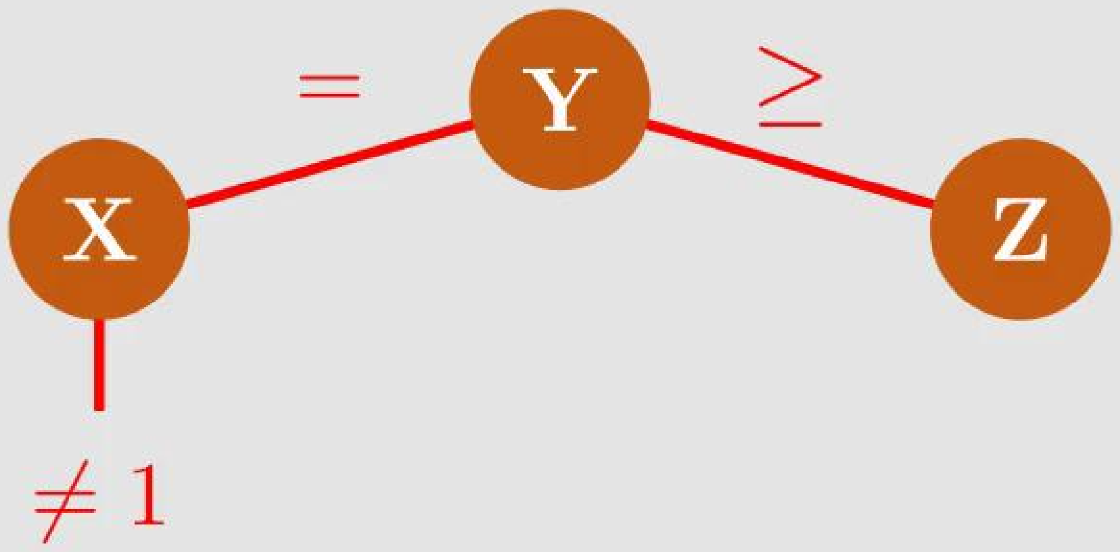
RUP $s_1 \implies z_{=3}$;

RUP $s_2 \implies \bar{x}_{=1}$;

RUP $s_2 \implies \bar{y}_{=1}$;

Justifying Smart Table





$\ldots$

$s_1 \iff e_{X<Y} + e_{Y\in\{1,2\}} + e_{Z=3} \geq 3$

$s_2 \iff e_{X\neq 1} + e_{Y=X} + e_{Z\leq Y} \geq 3$

$s_1 + s_2 \geq 1$

RUP $s_1 \implies \bar{y}_{=1}$;

RUP $s_1 \implies \bar{y}_{=3}$;

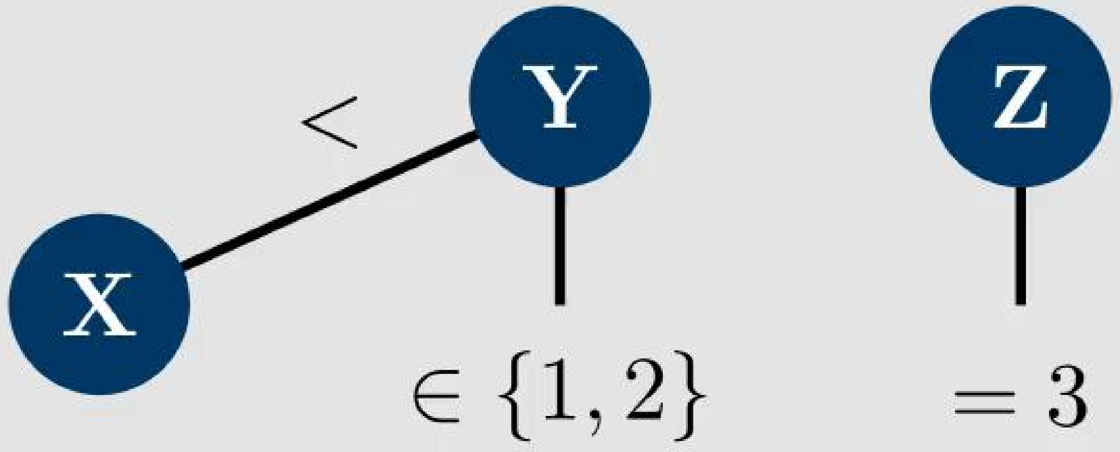
RUP $s_1 \implies x_{=1}$;

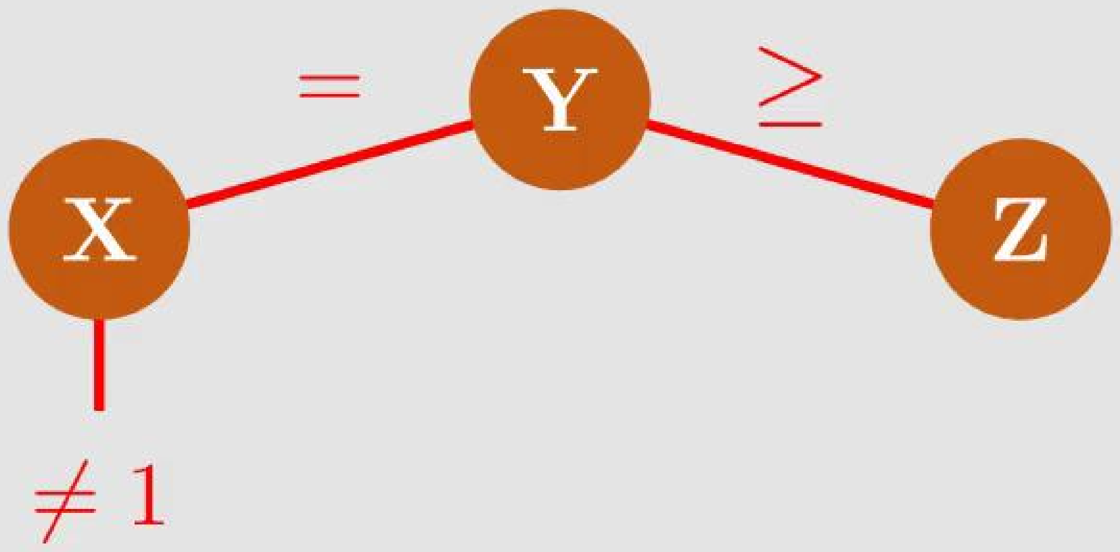
RUP $s_1 \implies z_{=3}$;

RUP $s_2 \implies \bar{x}_{=1}$;

RUP $s_2 \implies \bar{y}_{=1}$;

Justifying Smart Table





...

$s_1 \iff e_{X<Y} + e_{Y\in\{1,2\}} + e_{Z=3} \geq 3$

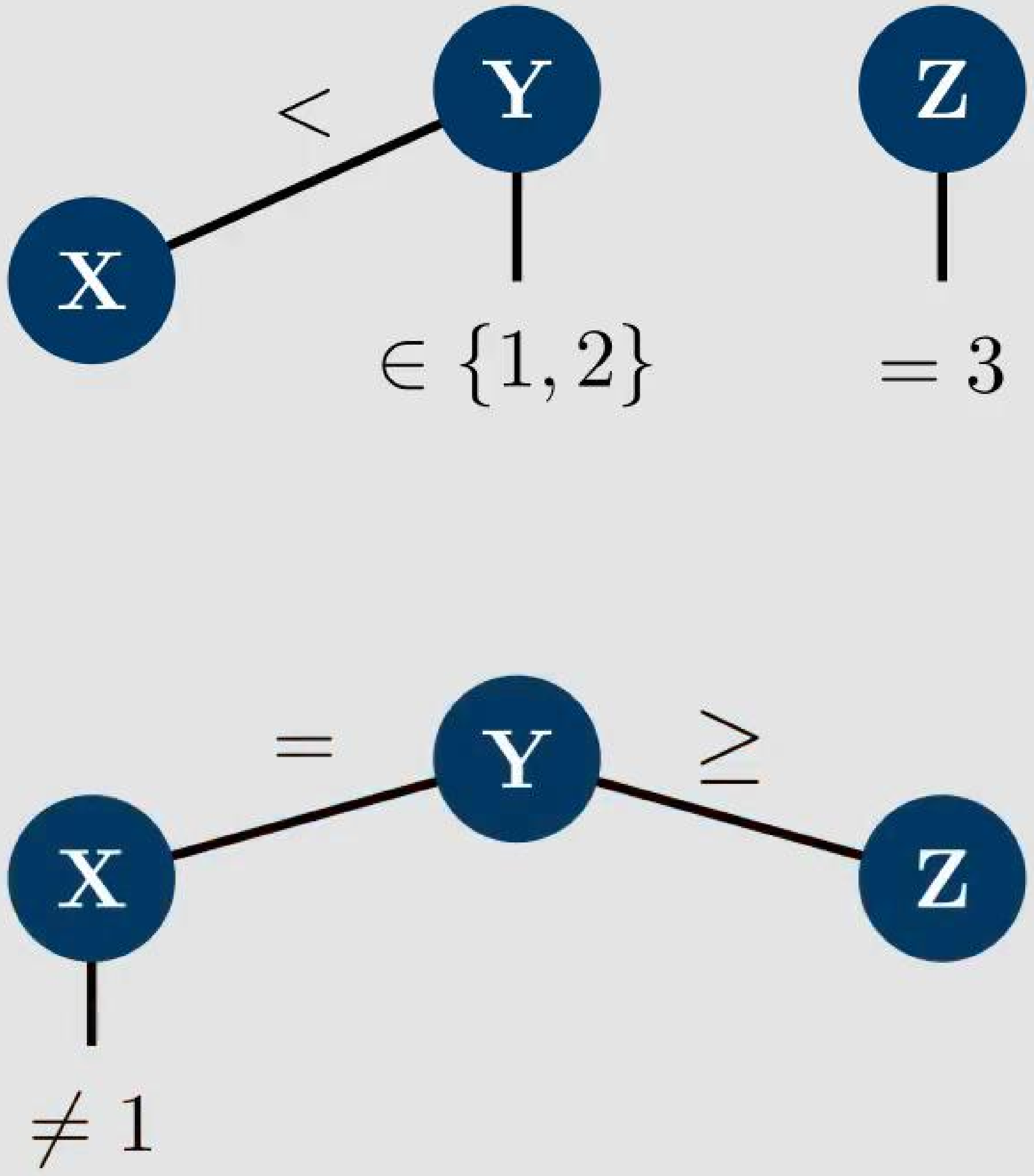
$s_2 \iff e_{X\neq 1} + e_{Y=X} + e_{Z\leq Y} \geq 3$

$s_1 + s_2 \geq 1$

RUP $s_1 \implies \bar{y}_{=1}$;

RUP $s_2 \implies \bar{y}_{=1}$;

Justifying Smart Table



...

$s_1 \iff e_{X<Y} + e_{Y\in\{1,2\}} + e_{Z=3} \geq 3$

$s_2 \iff e_{X\neq 1} + e_{Y=X} + e_{Z\leq Y} \geq 3$

$s_1 + s_2 \geq 1$

RUP $s_1 \implies \bar{y}_{=1}$;

RUP $s_2 \implies \bar{y}_{=1}$;

RUP $\bar{y}_{=1}$;

Proofs Under Implications

Theorem:

Proofs Under Implications

Theorem:

Let F be a formula, ρ be a partial assignment and suppose that from $F\restriction_\rho$ we can derive a constraint D using a cutting planes and RUP derivation of length L. Then we can construct a derivation of length $O(n \cdot L)$ from F of the constraint

$$\bigwedge_{\ell \in \rho} \ell \implies D$$

Proofs Under Implications

Theorem:

Proofs Under Implications

Theorem:

$$3x + 2y + z \geq 1$$
$$2y \geq 3$$
$$3x + 4y + z \geq 4$$

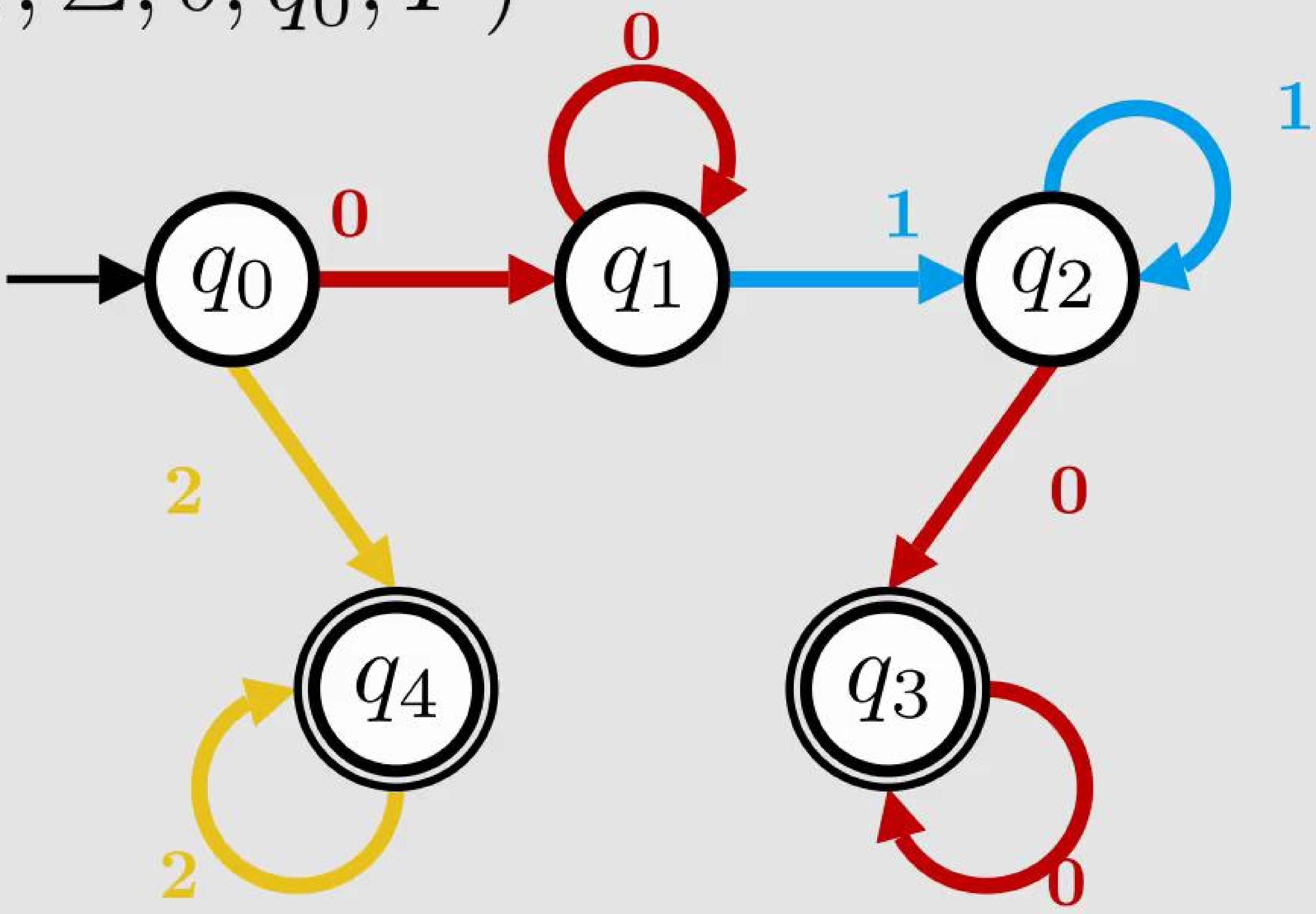
Proofs Under Implications

Theorem:

$$\ell \implies 3x + 2y + z \geq 1$$
$$\ell \implies 2y \geq 3$$
$$\ell \implies 3x + 4y + z \geq 4$$

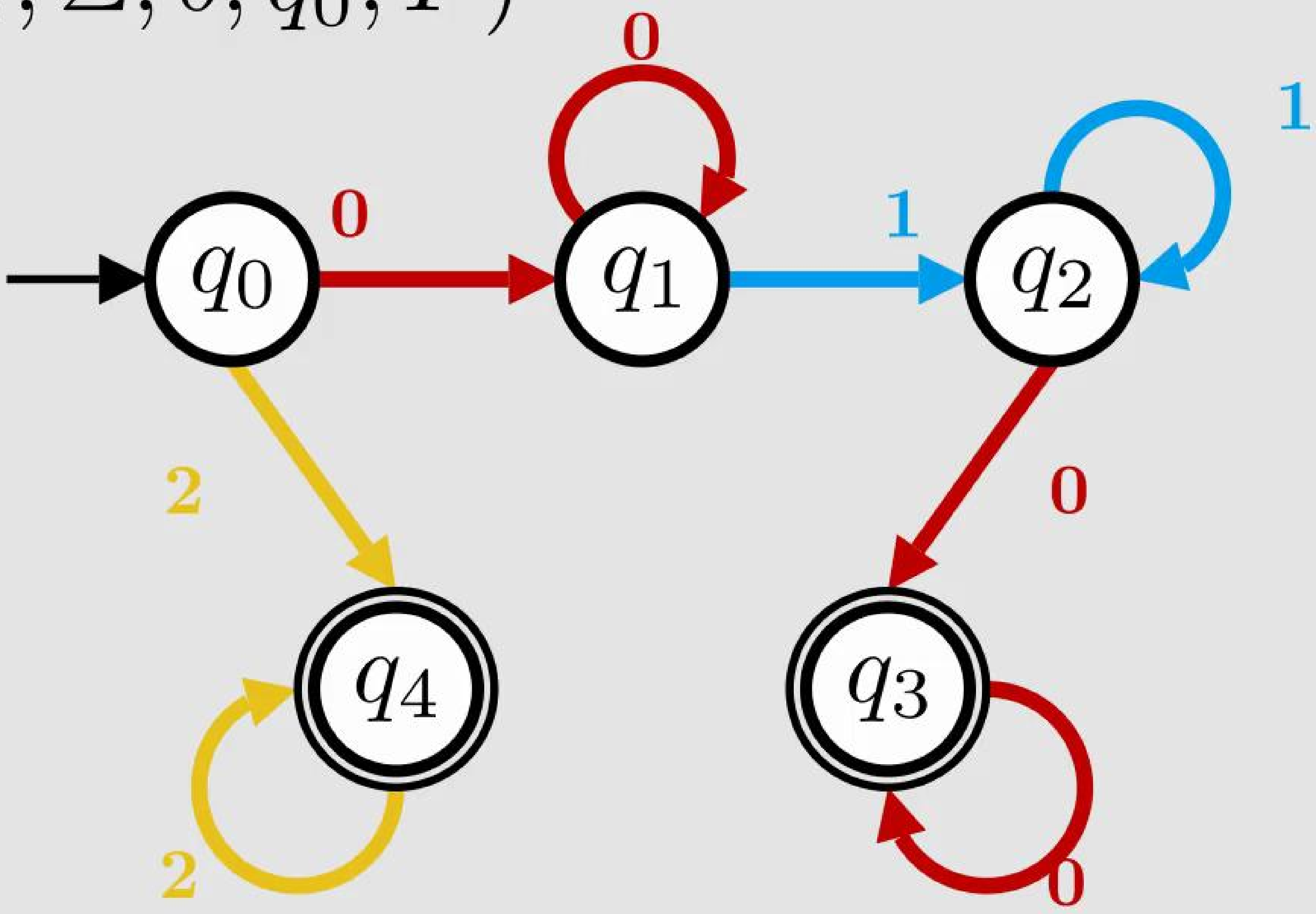
Regular PB Encoding

$(Q, \Sigma, \delta, q_0, F)$



Regular PB Encoding

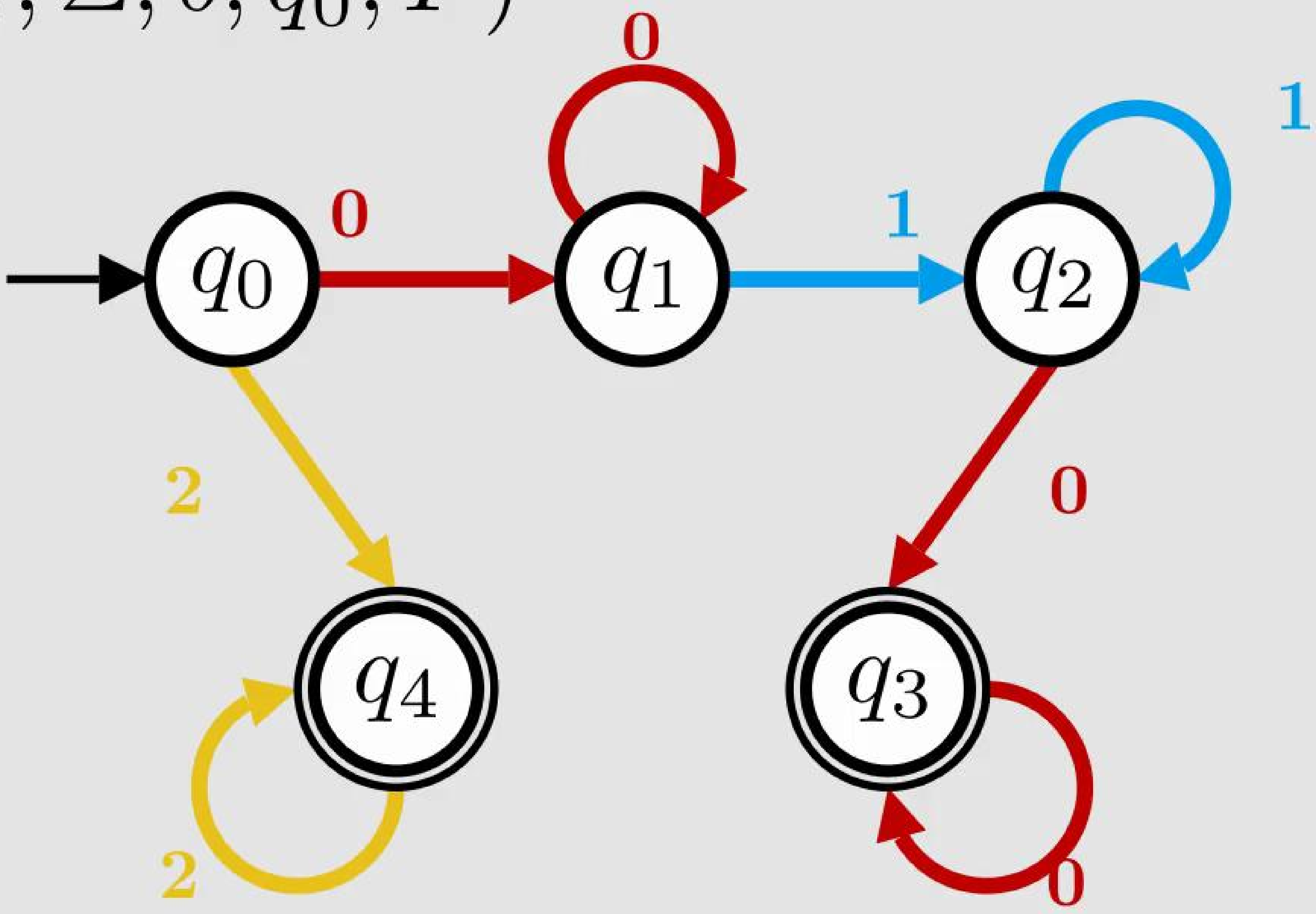
$(Q, \Sigma, \delta, q_0, F)$



$s_i :=$ The state after processing i variables

Regular PB Encoding

$(Q, \Sigma, \delta, q_0, F)$

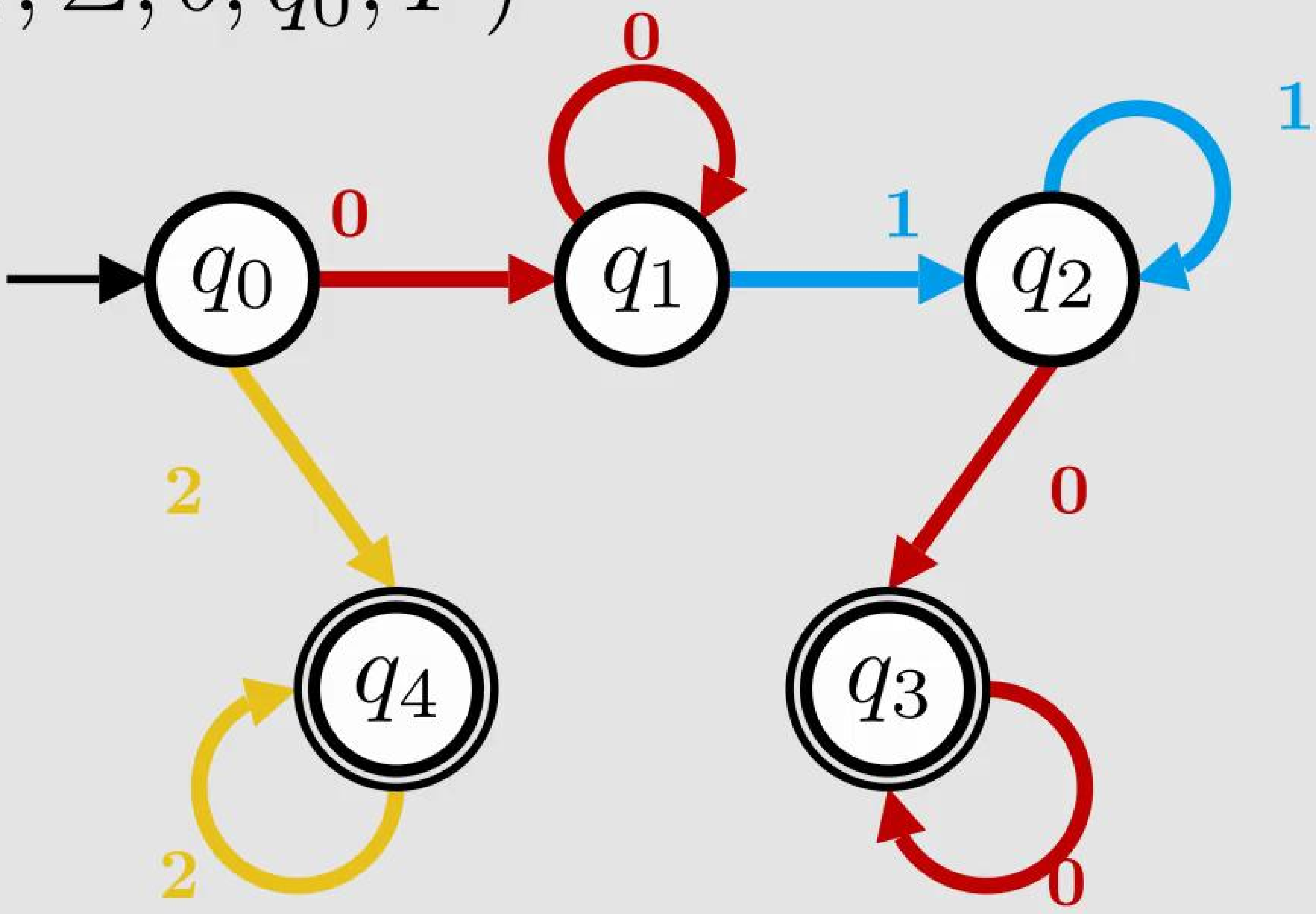


$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

Regular PB Encoding

$(Q, \Sigma, \delta, q_0, F)$



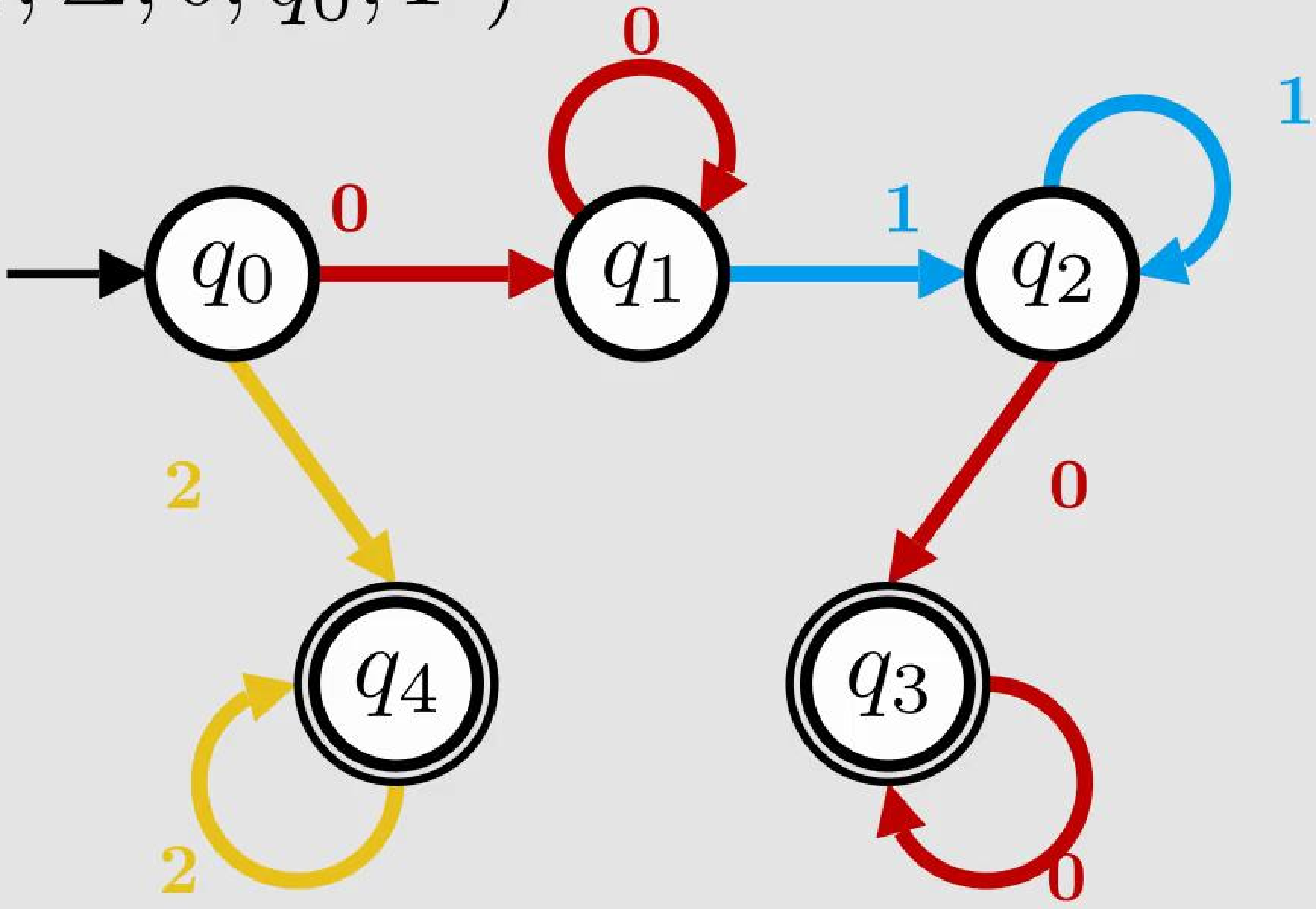
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

Regular PB Encoding

$(Q, \Sigma, \delta, q_0, F)$



$s_i :=$ The state after processing i variables

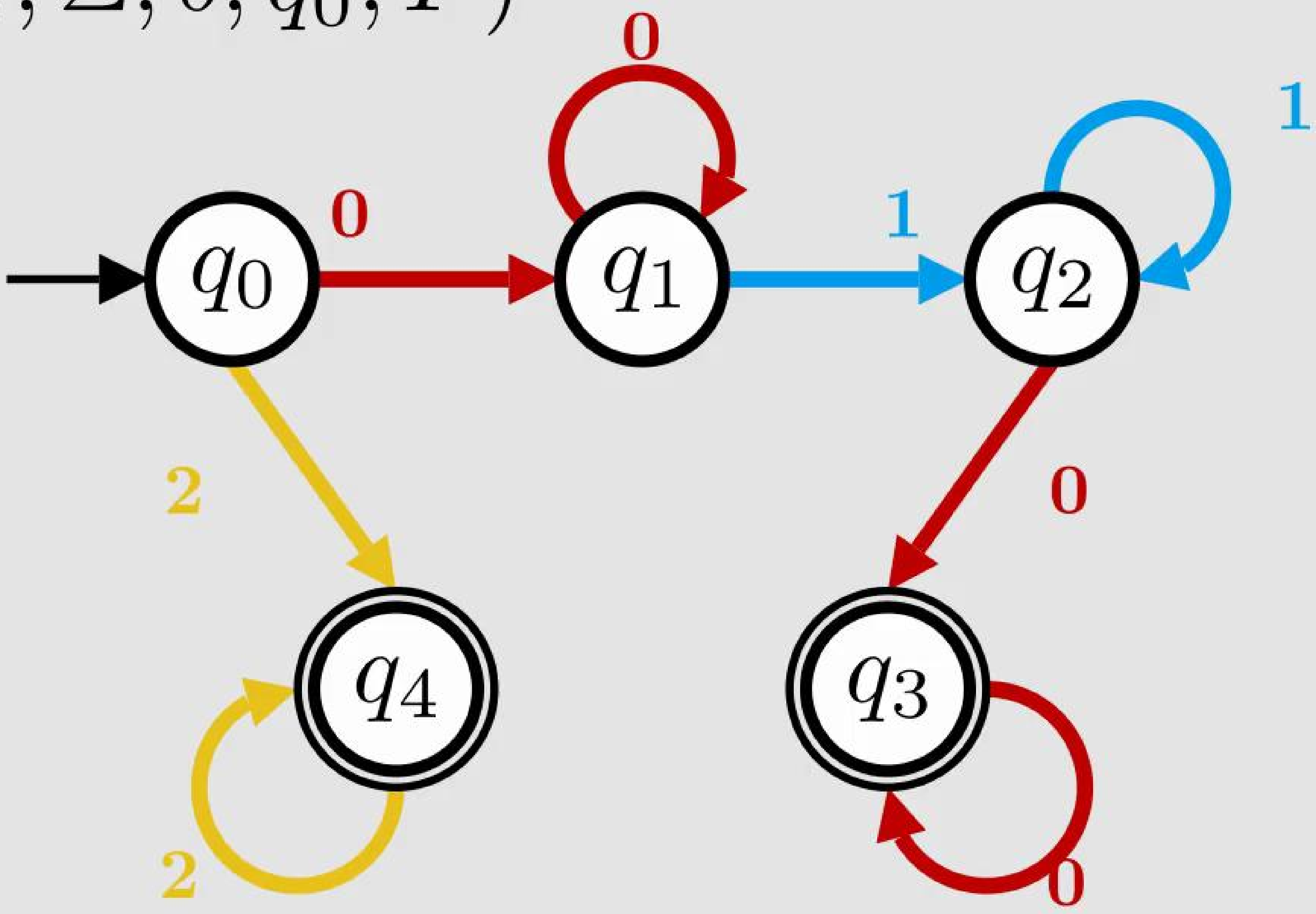
$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

Regular PB Encoding

$(Q, \Sigma, \delta, q_0, F)$



$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

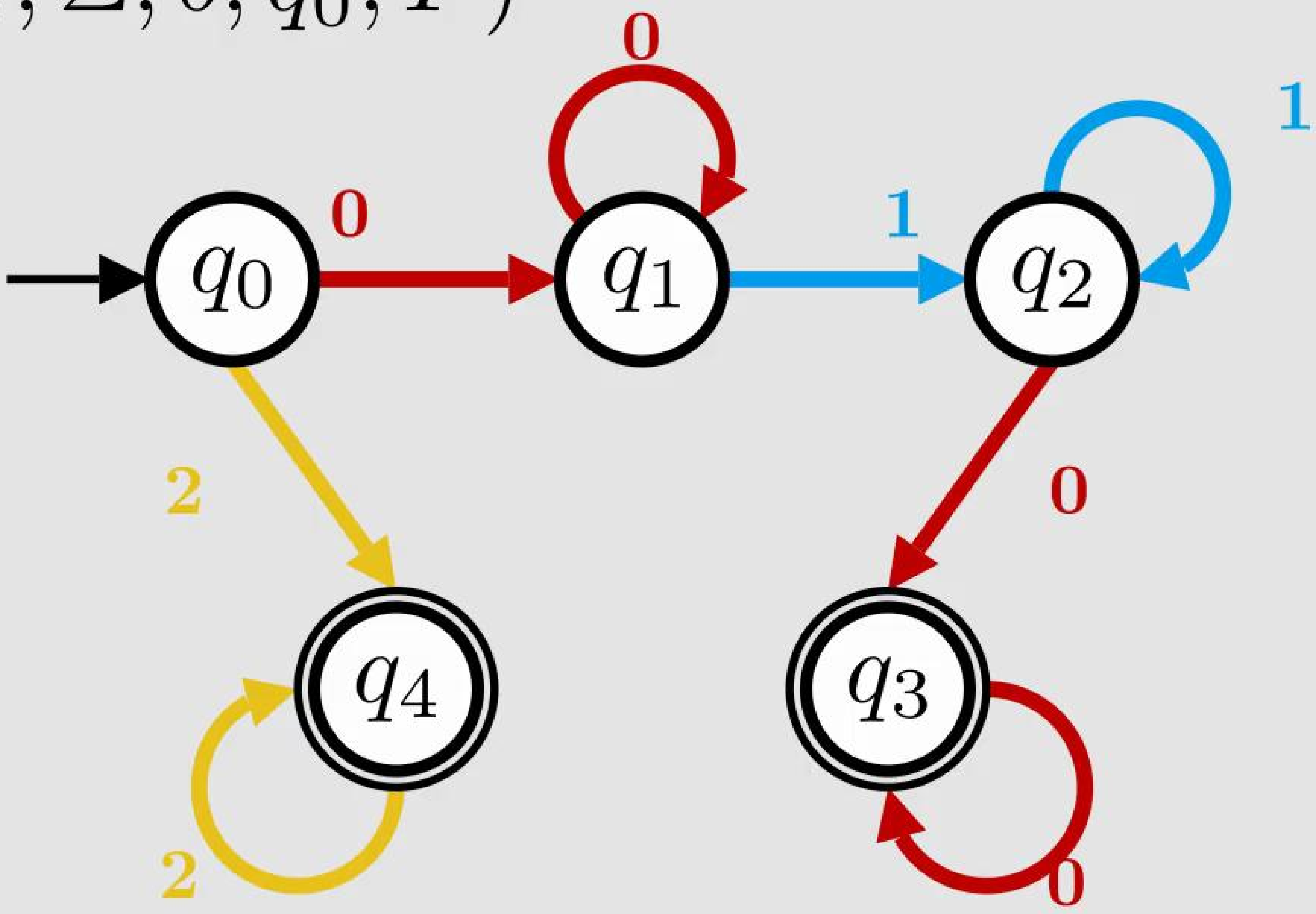
$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$

Regular PB Encoding

$(Q, \Sigma, \delta, q_0, F)$



$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \land s_{i=q} \implies s_{i+1=\delta(q,j)}$

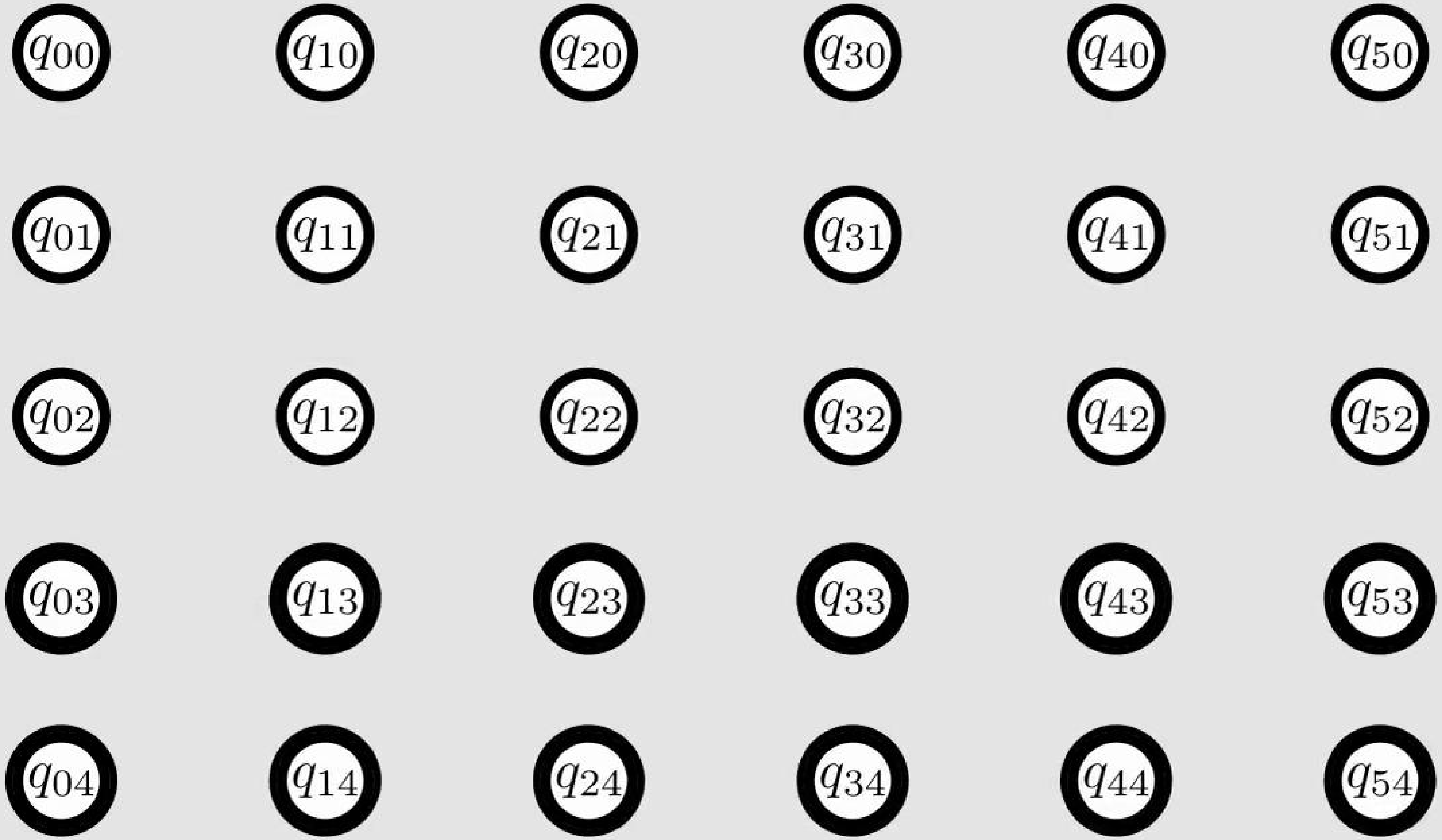
s_i := The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



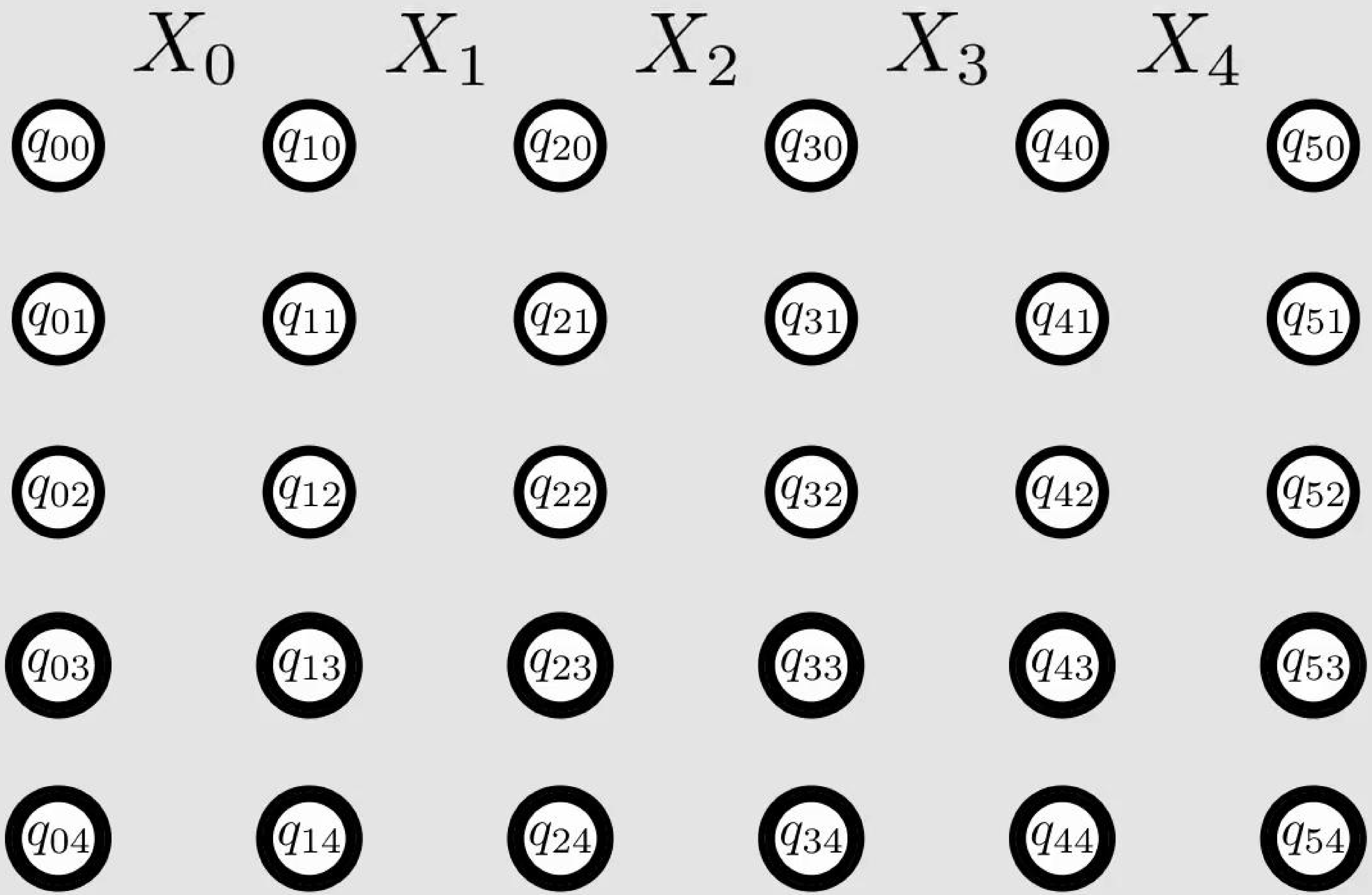
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



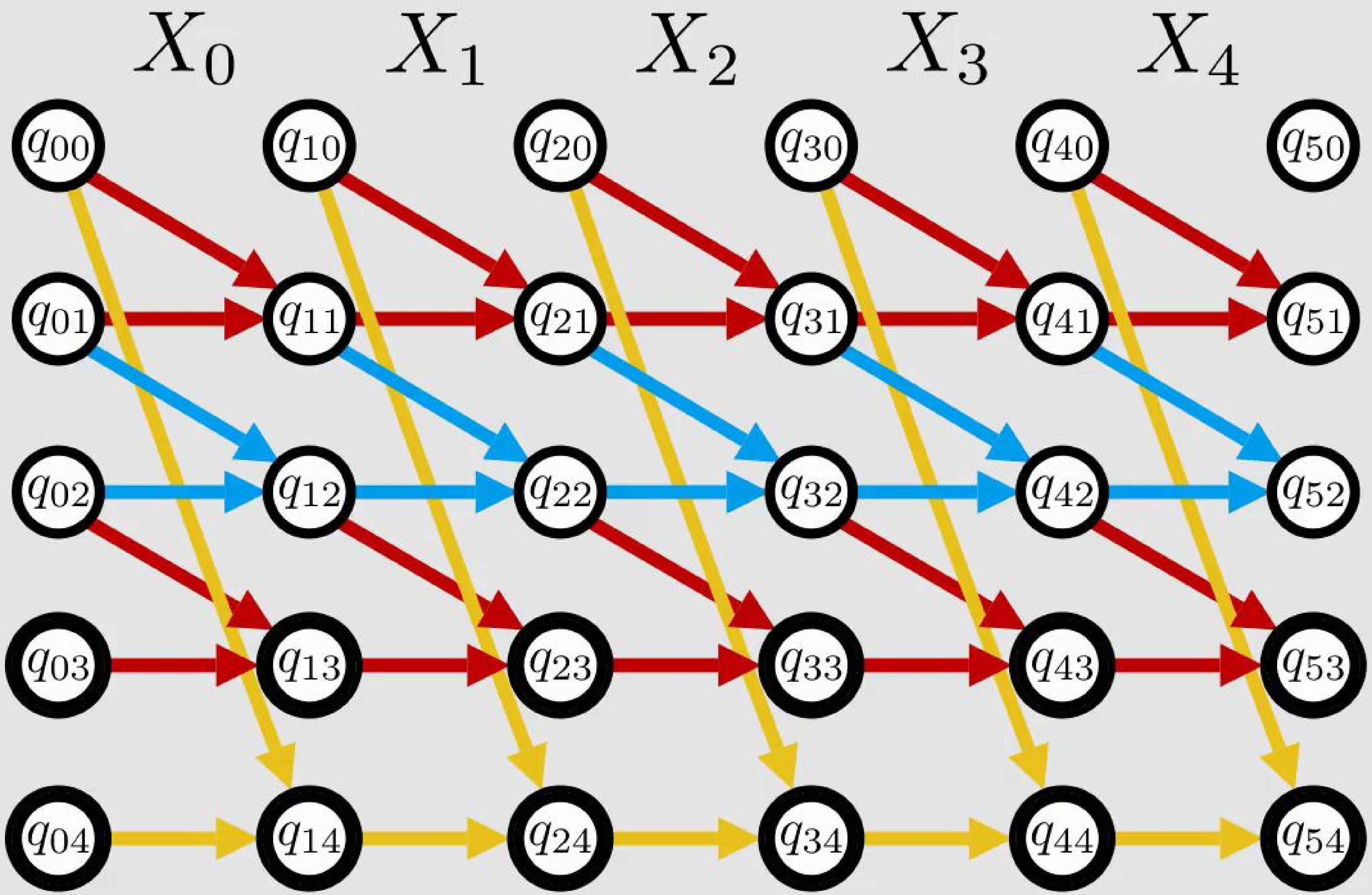
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



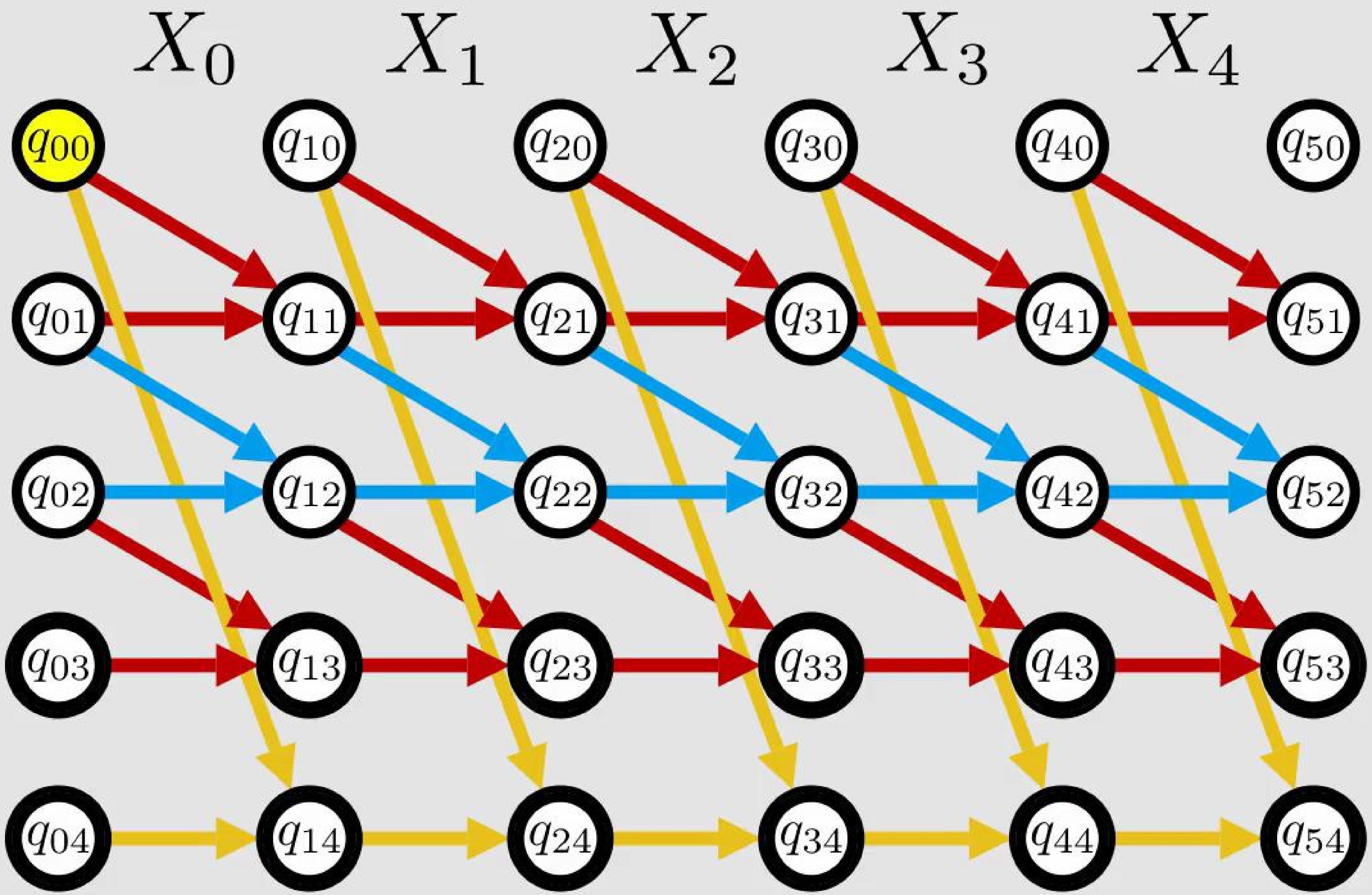
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



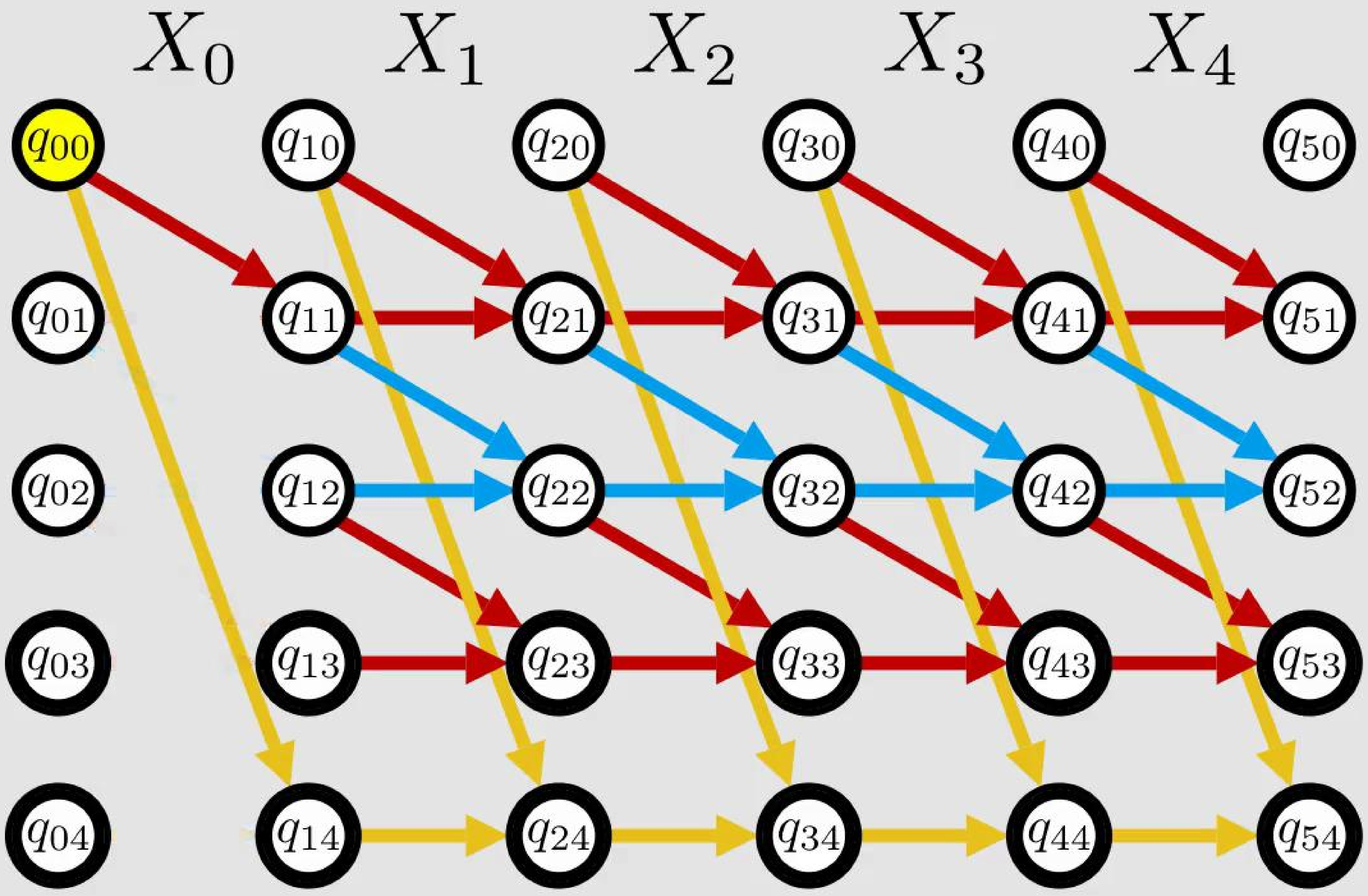
s_i := The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \land s_{i=q} \implies s_{i+1=\delta(q,j)}$



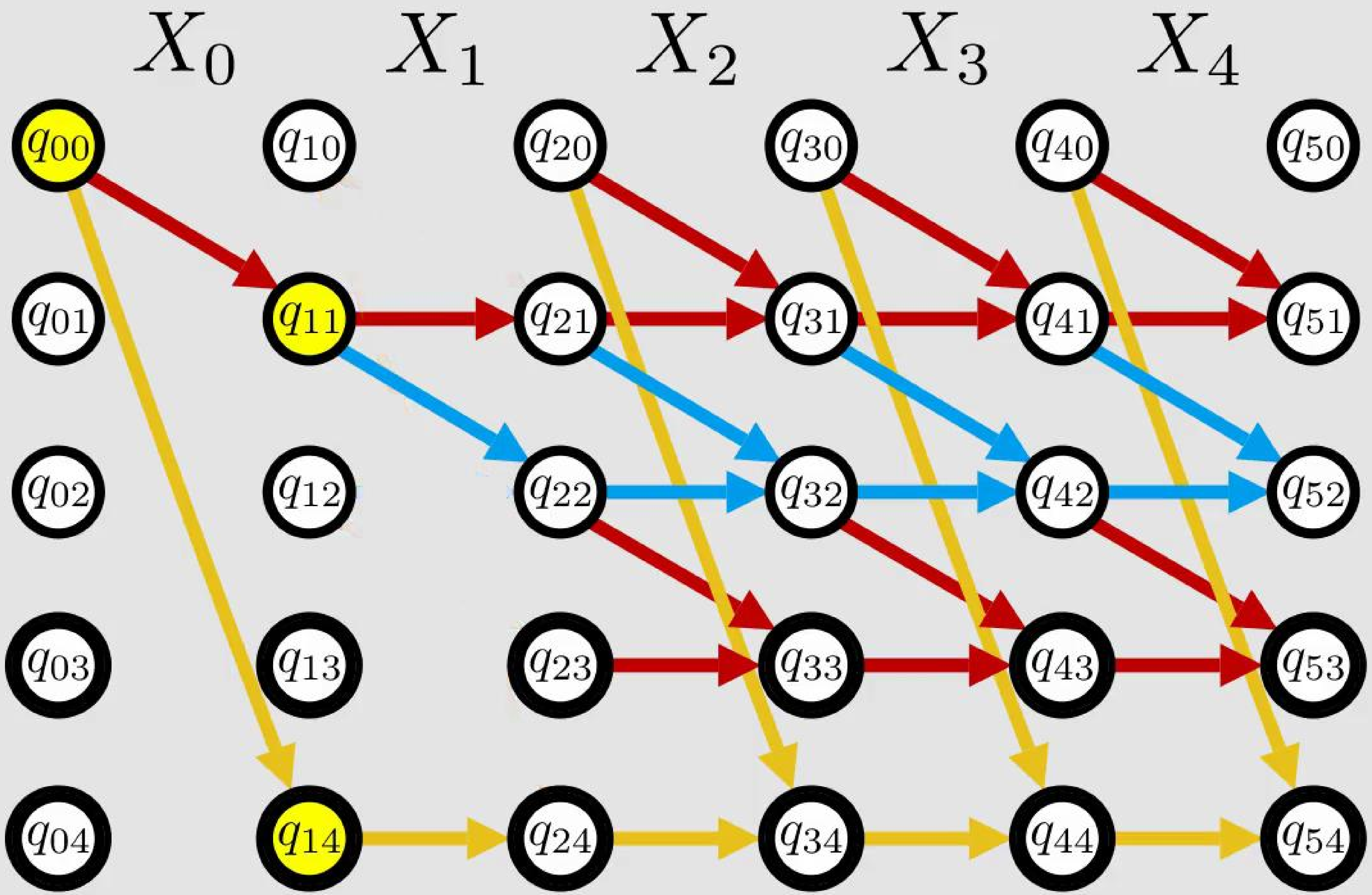
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



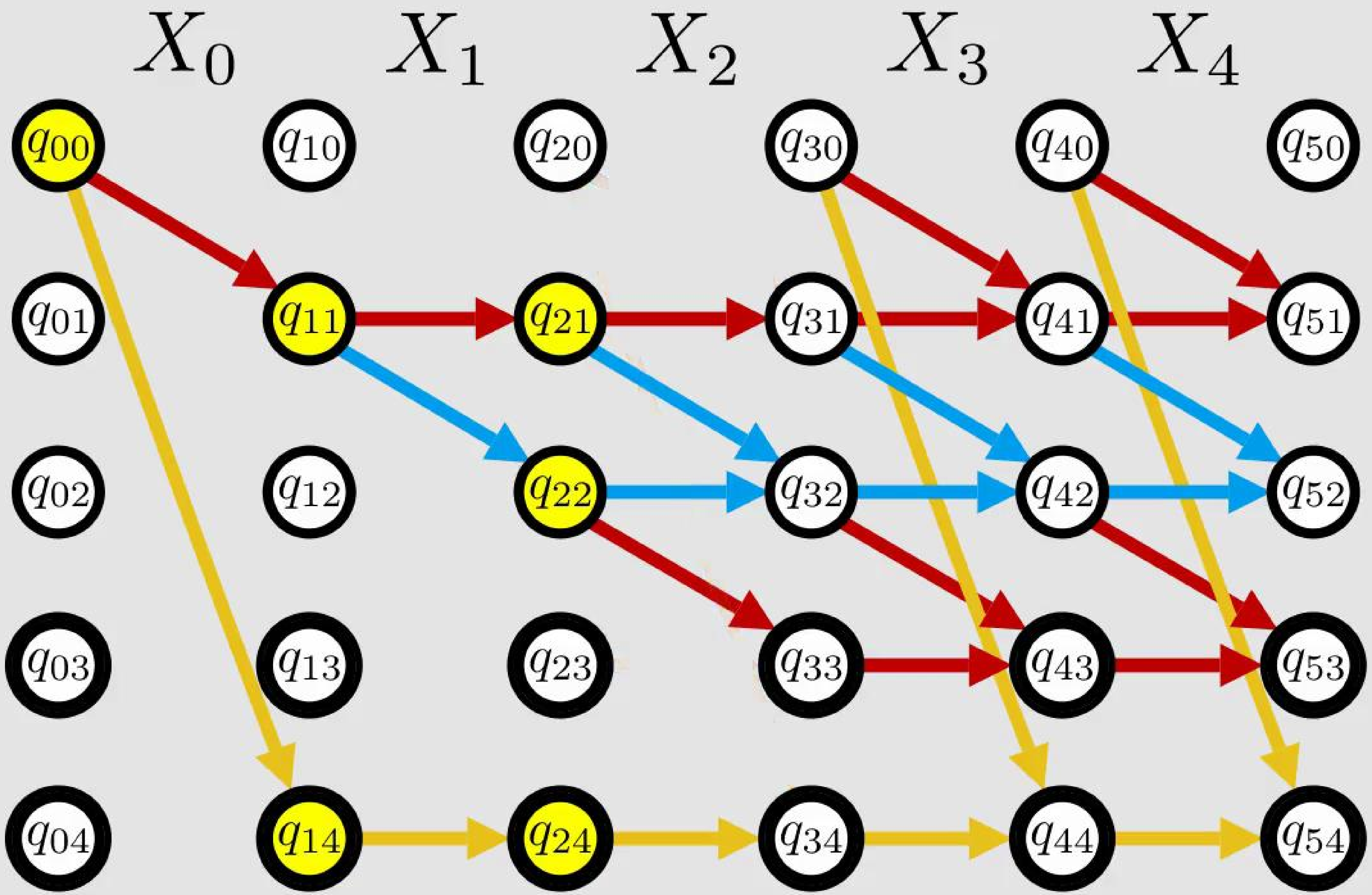
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



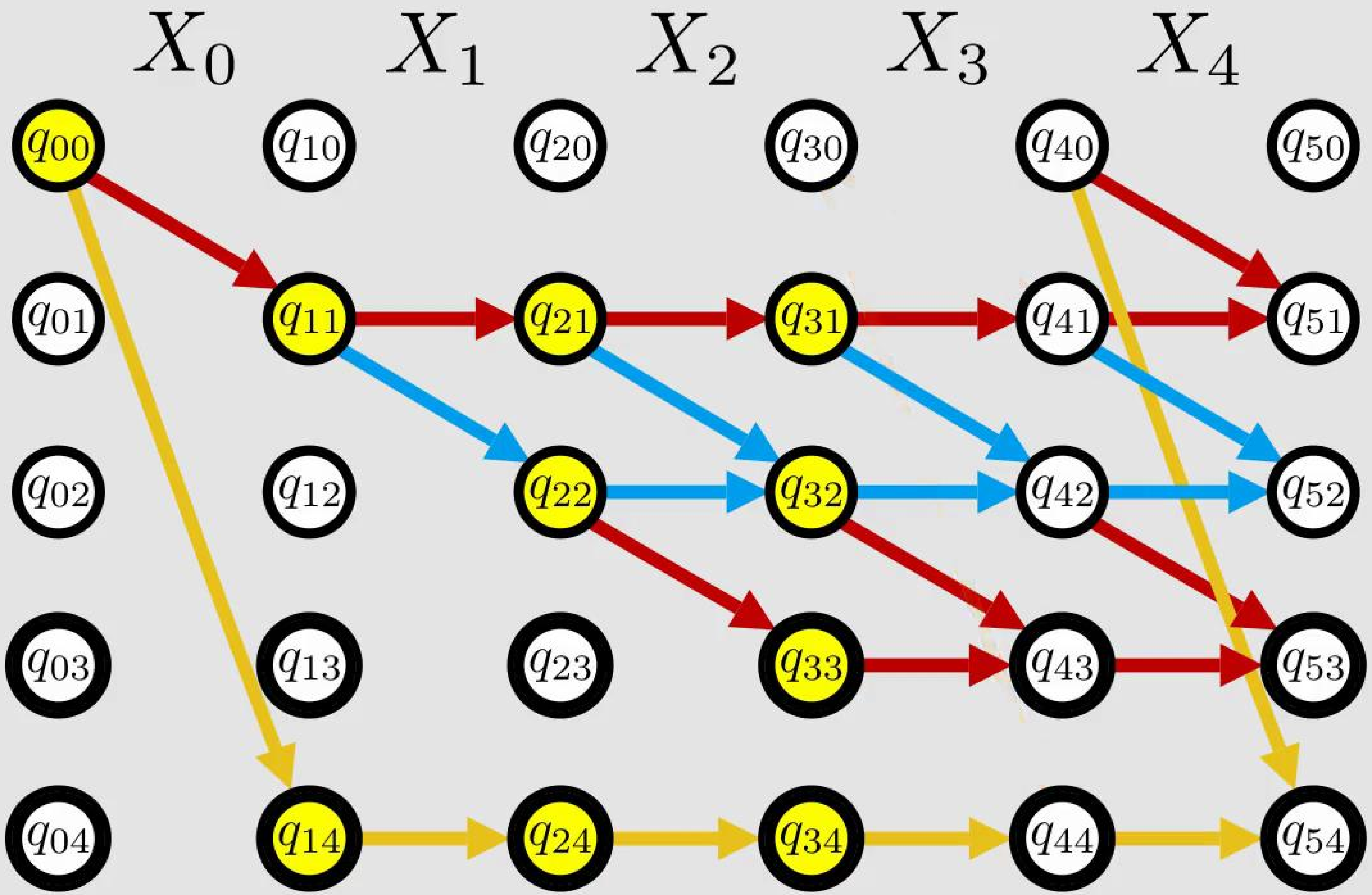
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \mathrm{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



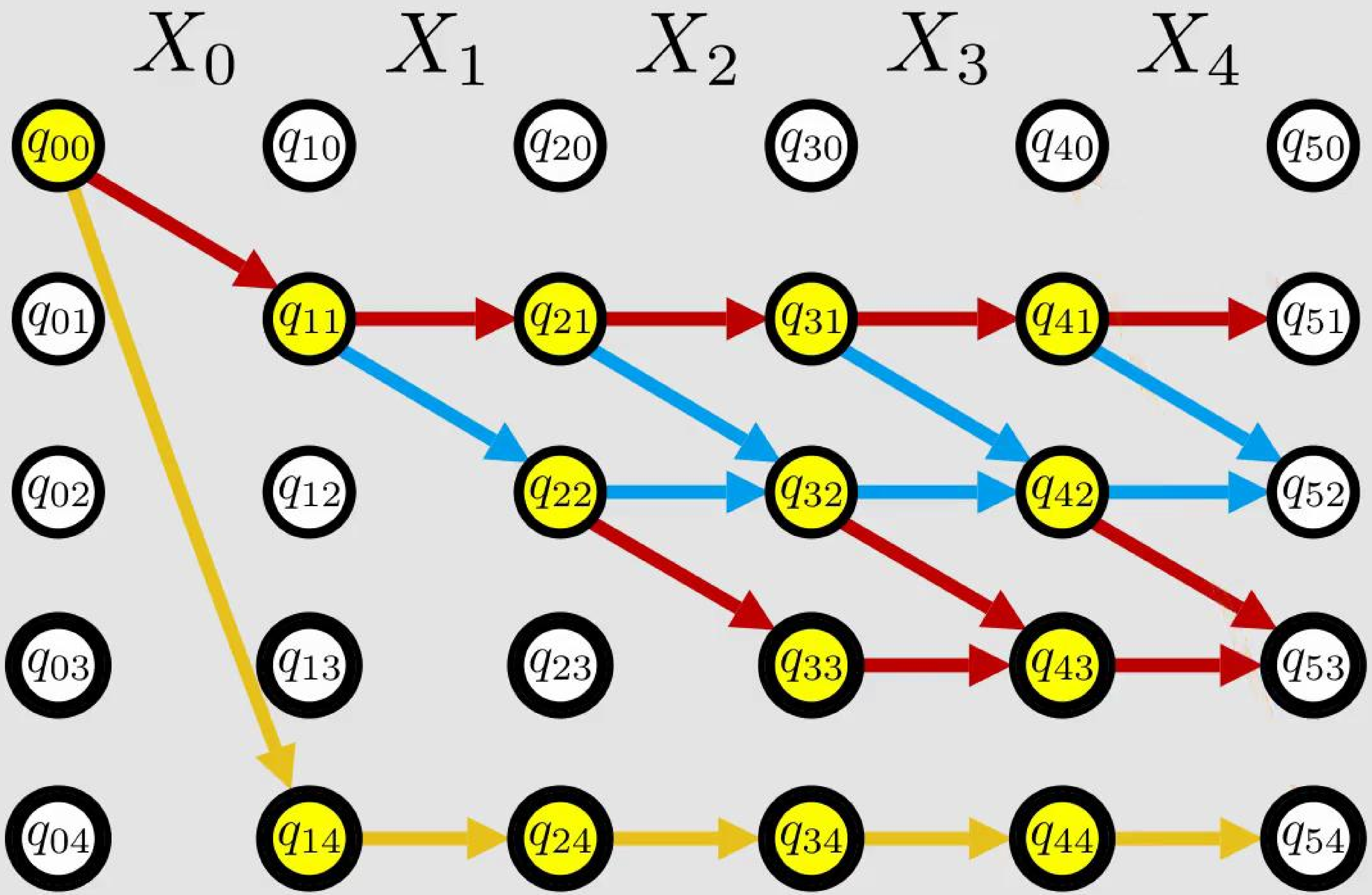
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



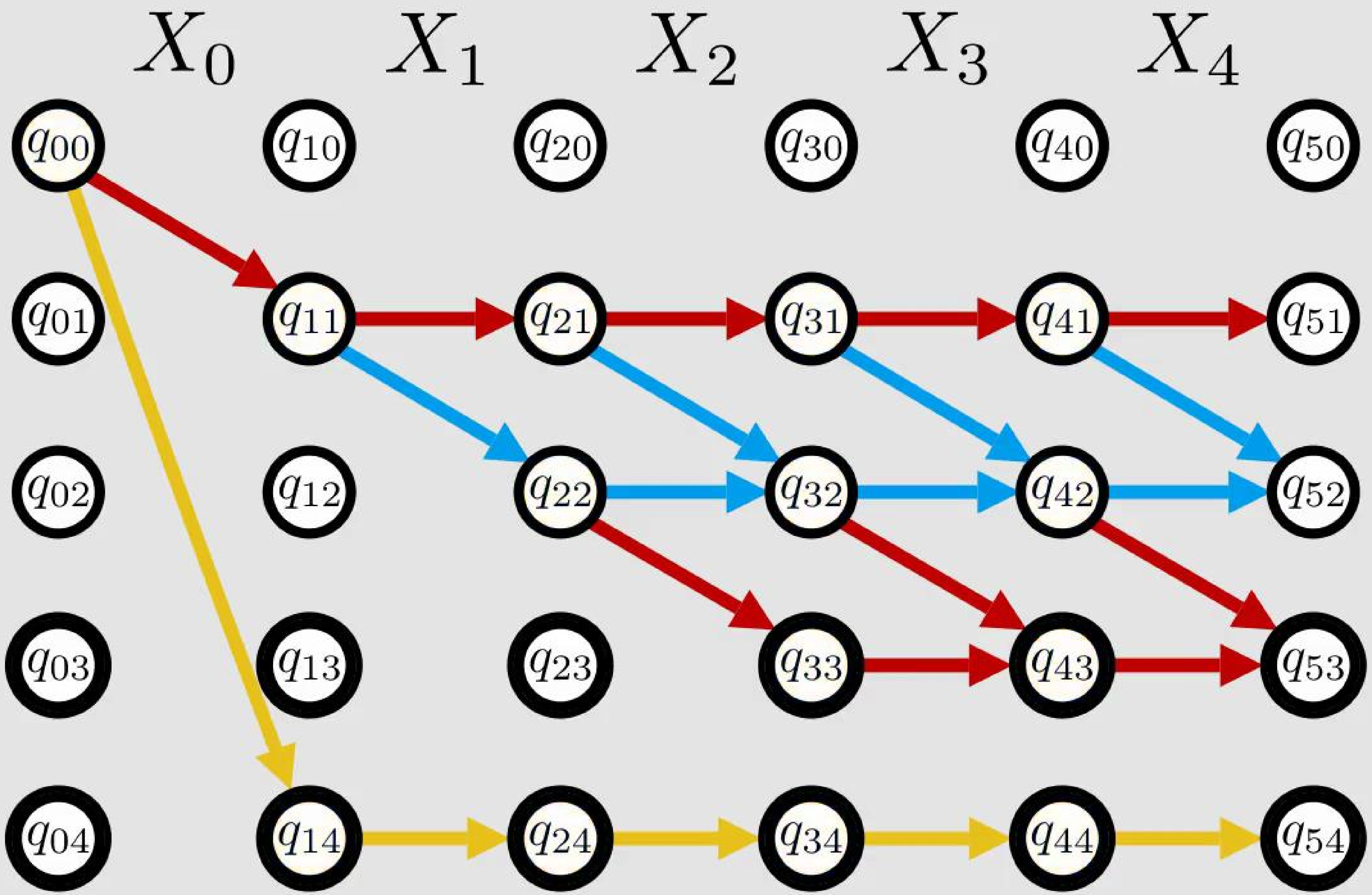
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \mathrm{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \land s_{i=q} \implies s_{i+1=\delta(q,j)}$



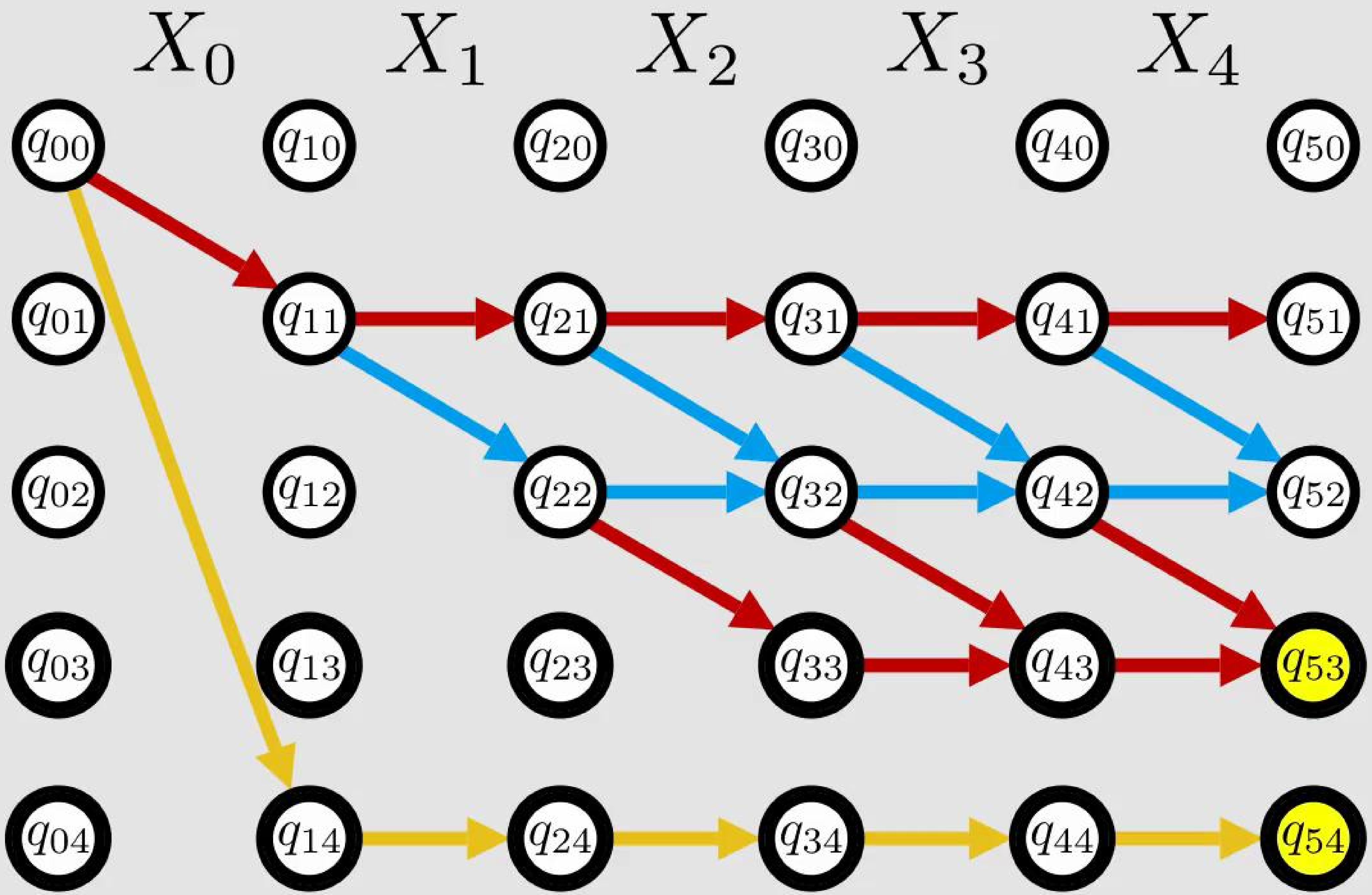
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



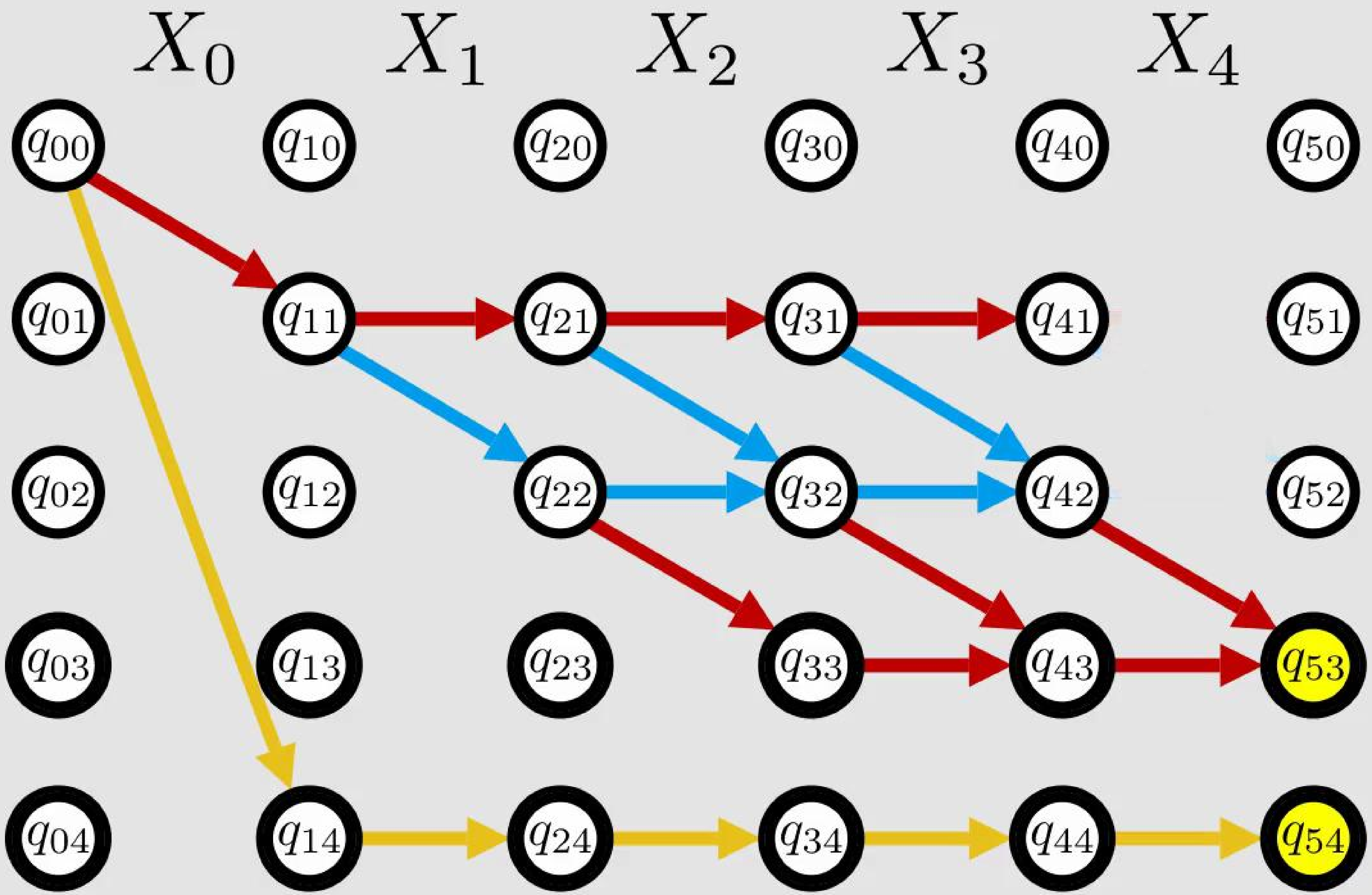
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



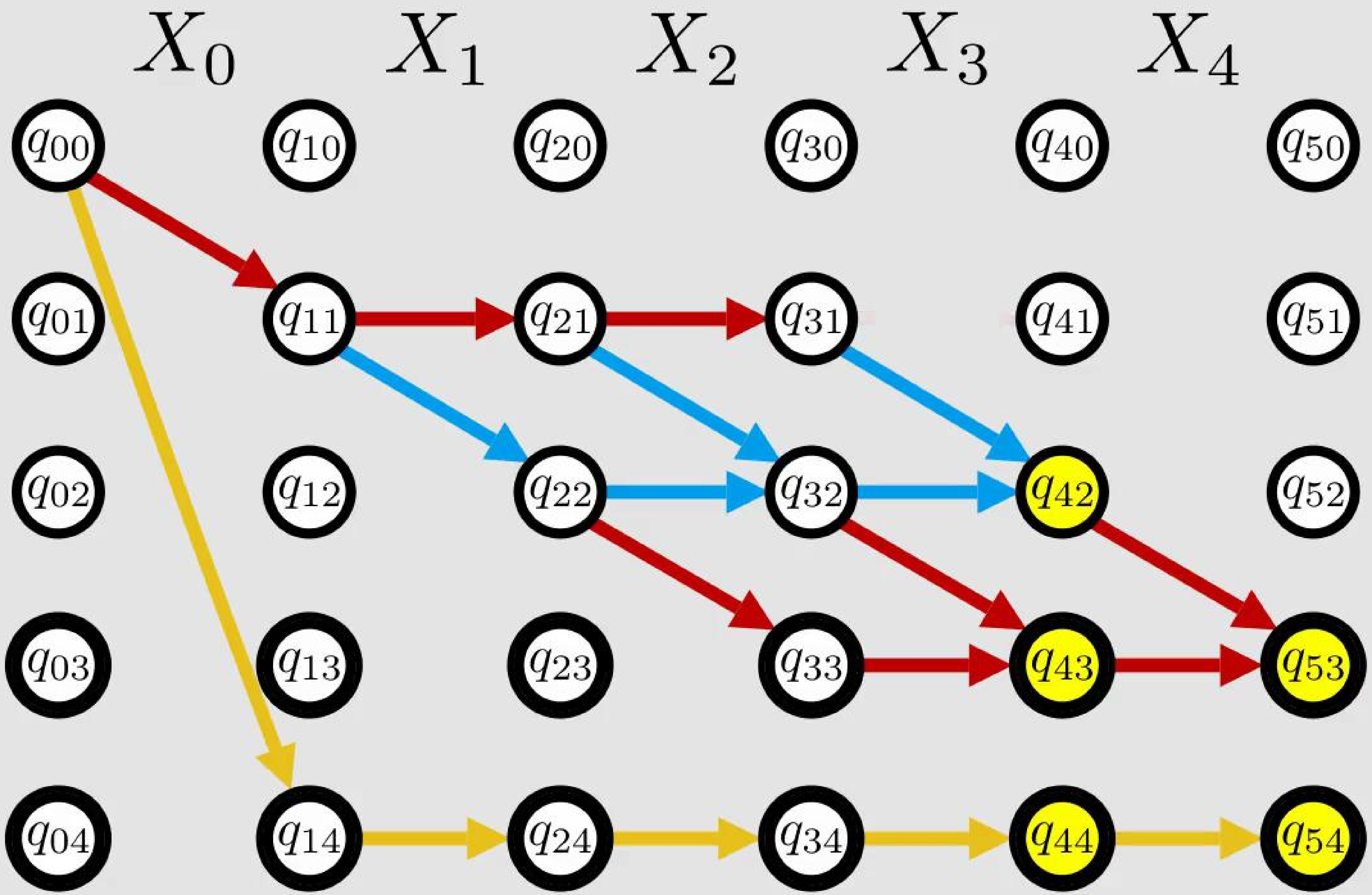
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i),$ and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



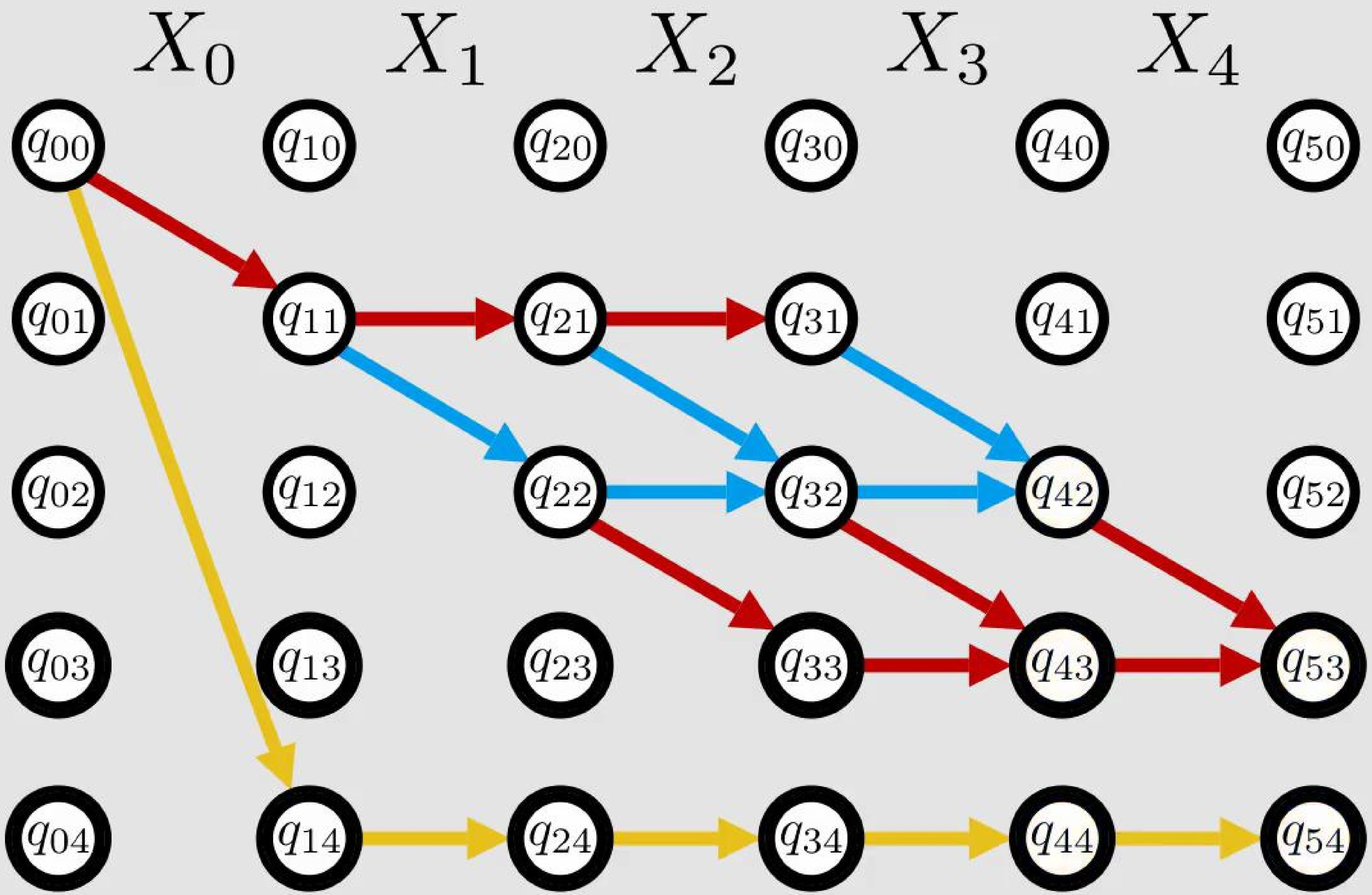
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



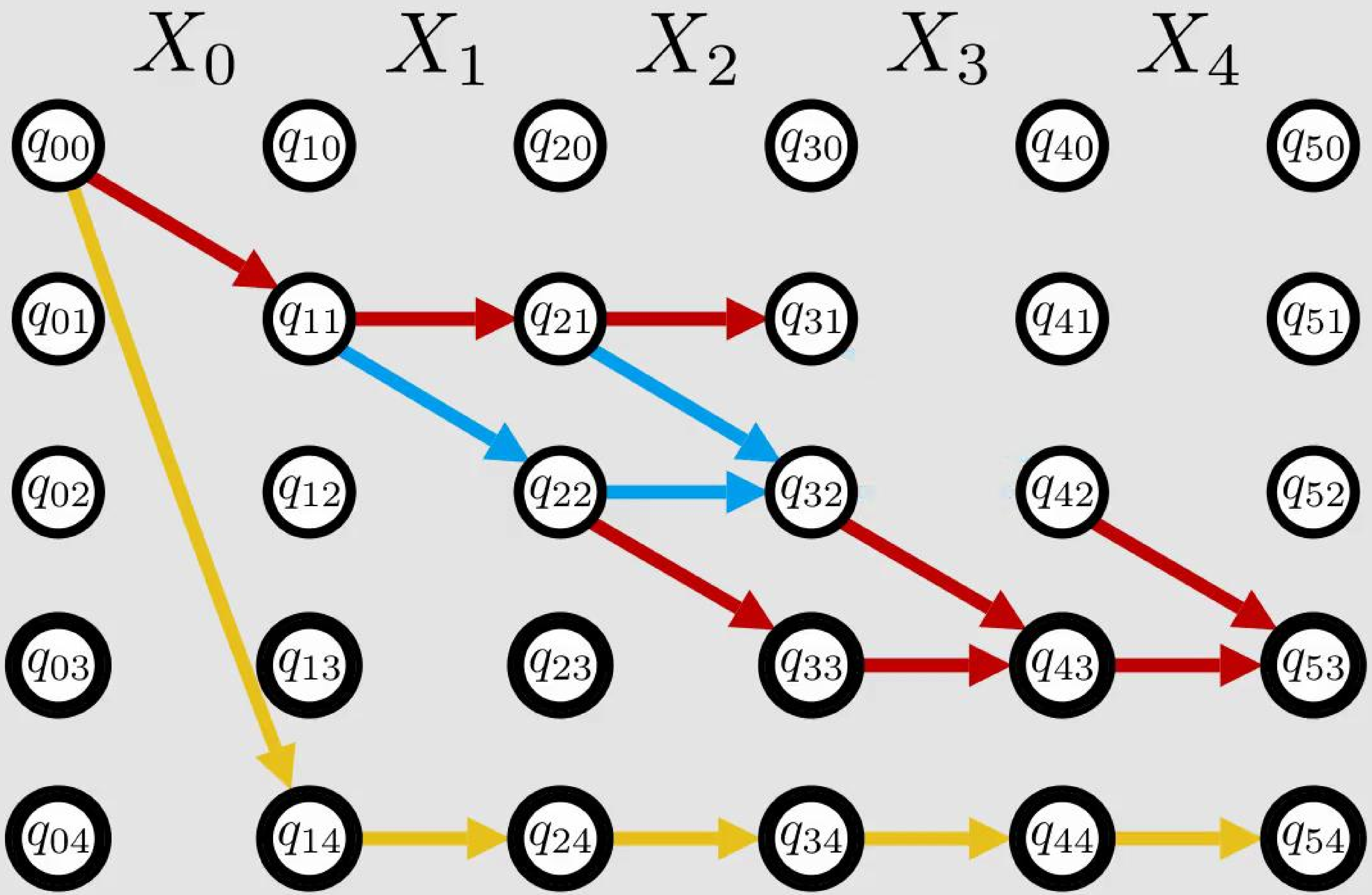
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \mathrm{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



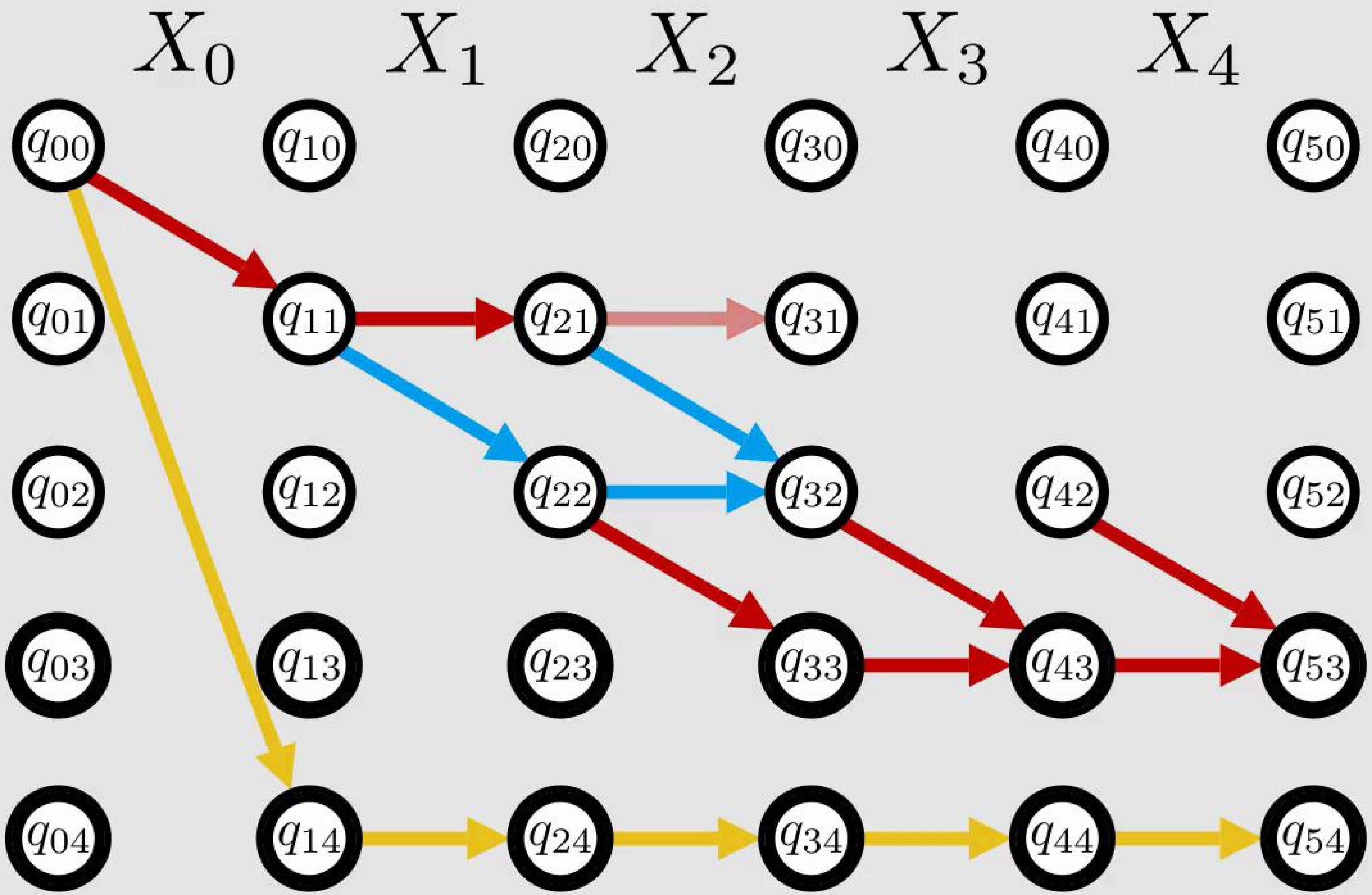
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



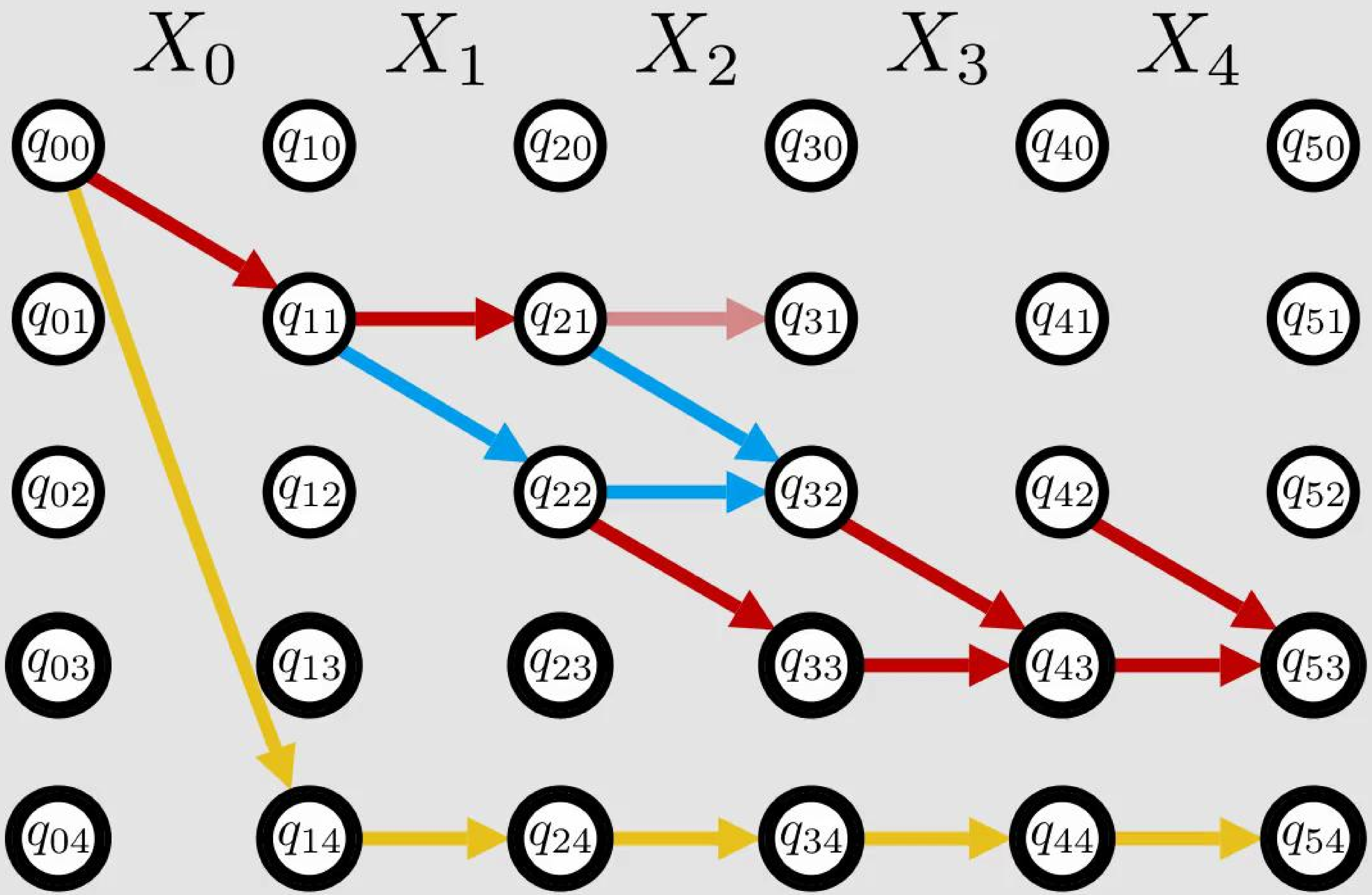
$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$



$s_i :=$ The state after processing i variables

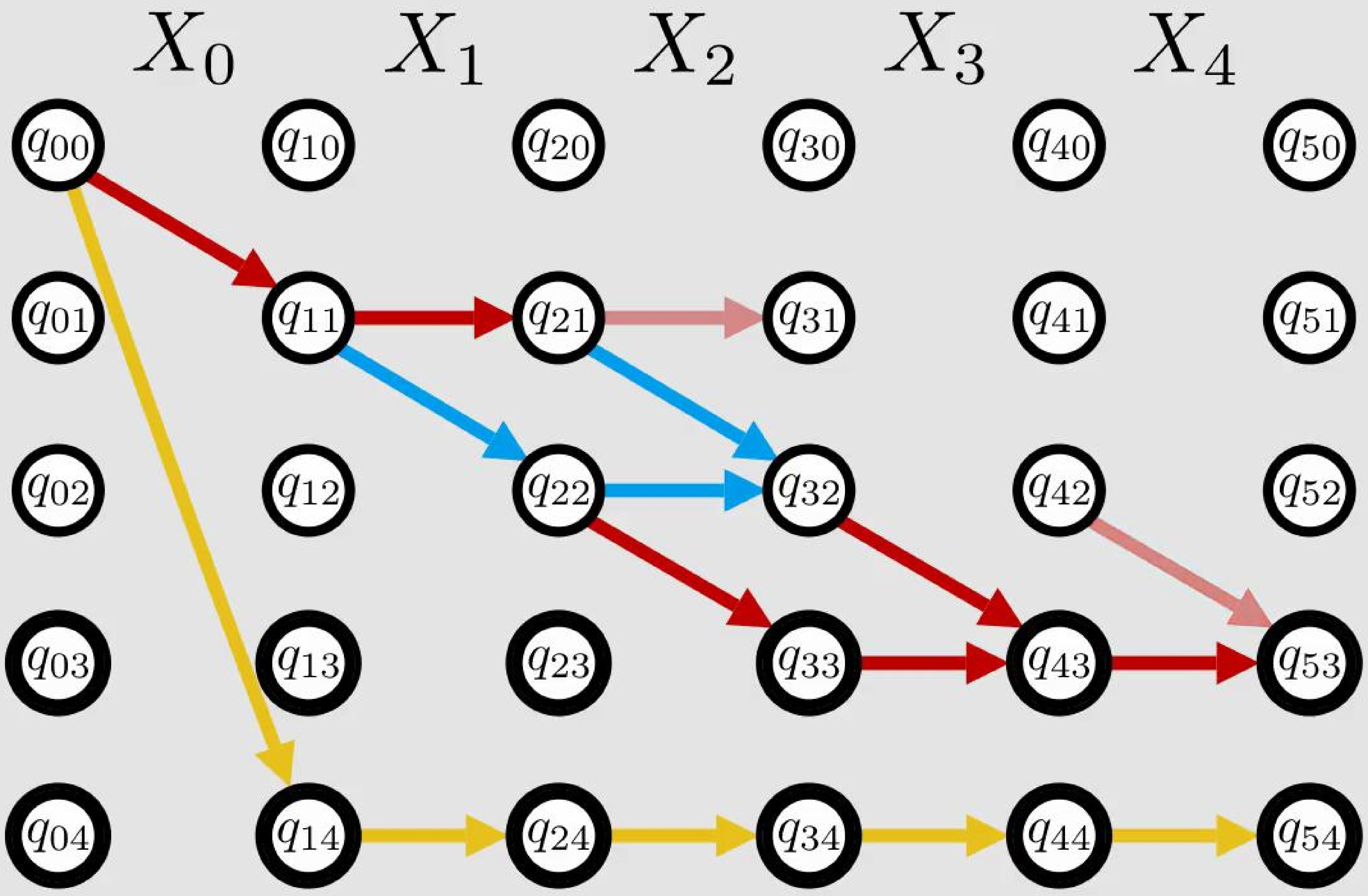
$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \mathrm{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$

RUP $\bar{s}_{2=1} \vee \bar{x}_{2=1}$



$s_i :=$ The state after processing i variables

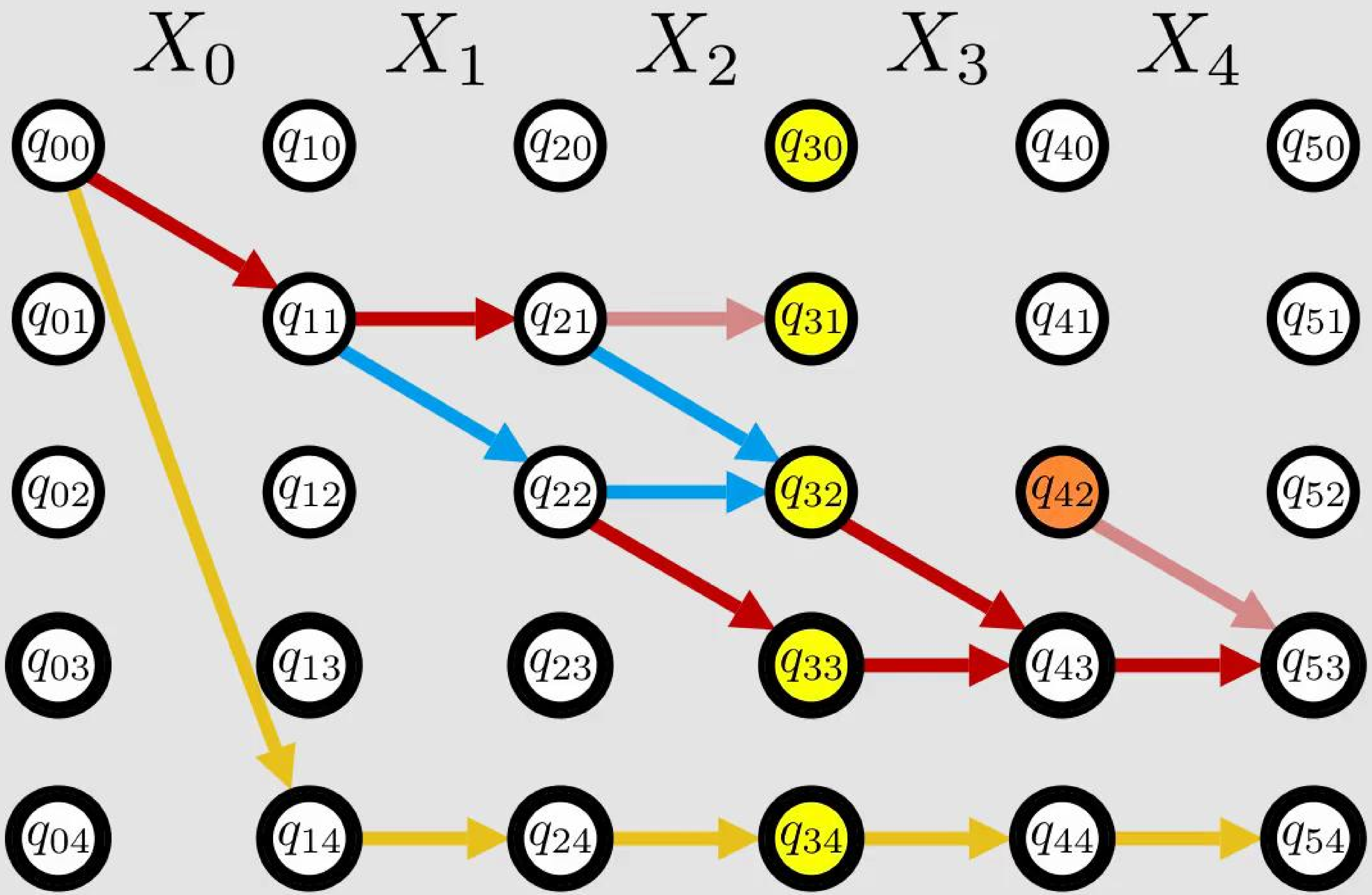
$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \land s_{i=q} \implies s_{i+1=\delta(q,j)}$

RUP $\bar{s}_{2=1} \lor \bar{x}_{2=1}$



$s_i :=$ The state after processing i variables

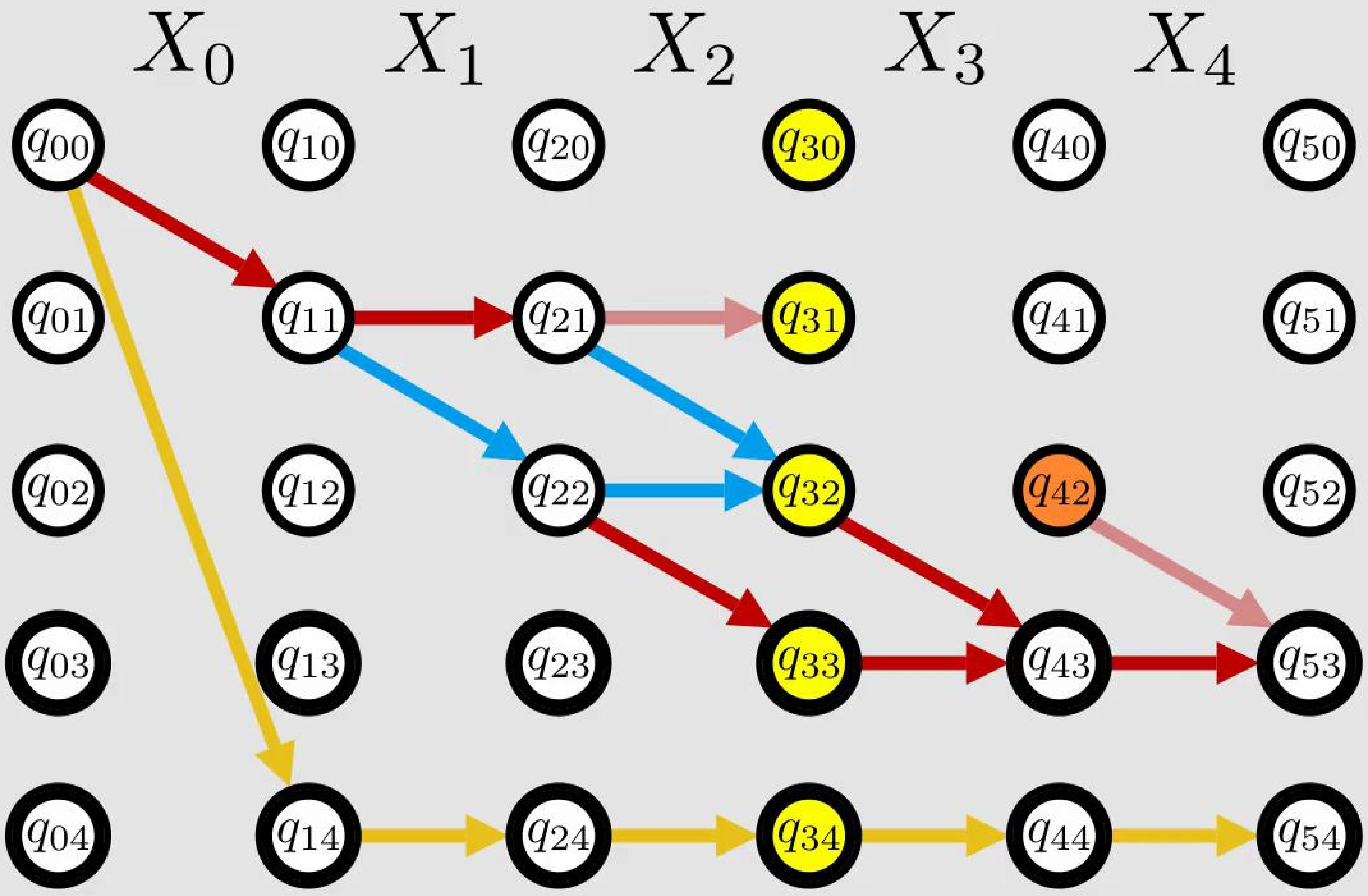
$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$

RUP $\bar{s}_{2=1} \vee \bar{x}_{2=1}$



$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$

RUP $\bar{s}_{2=1} \vee \bar{x}_{2=1}$

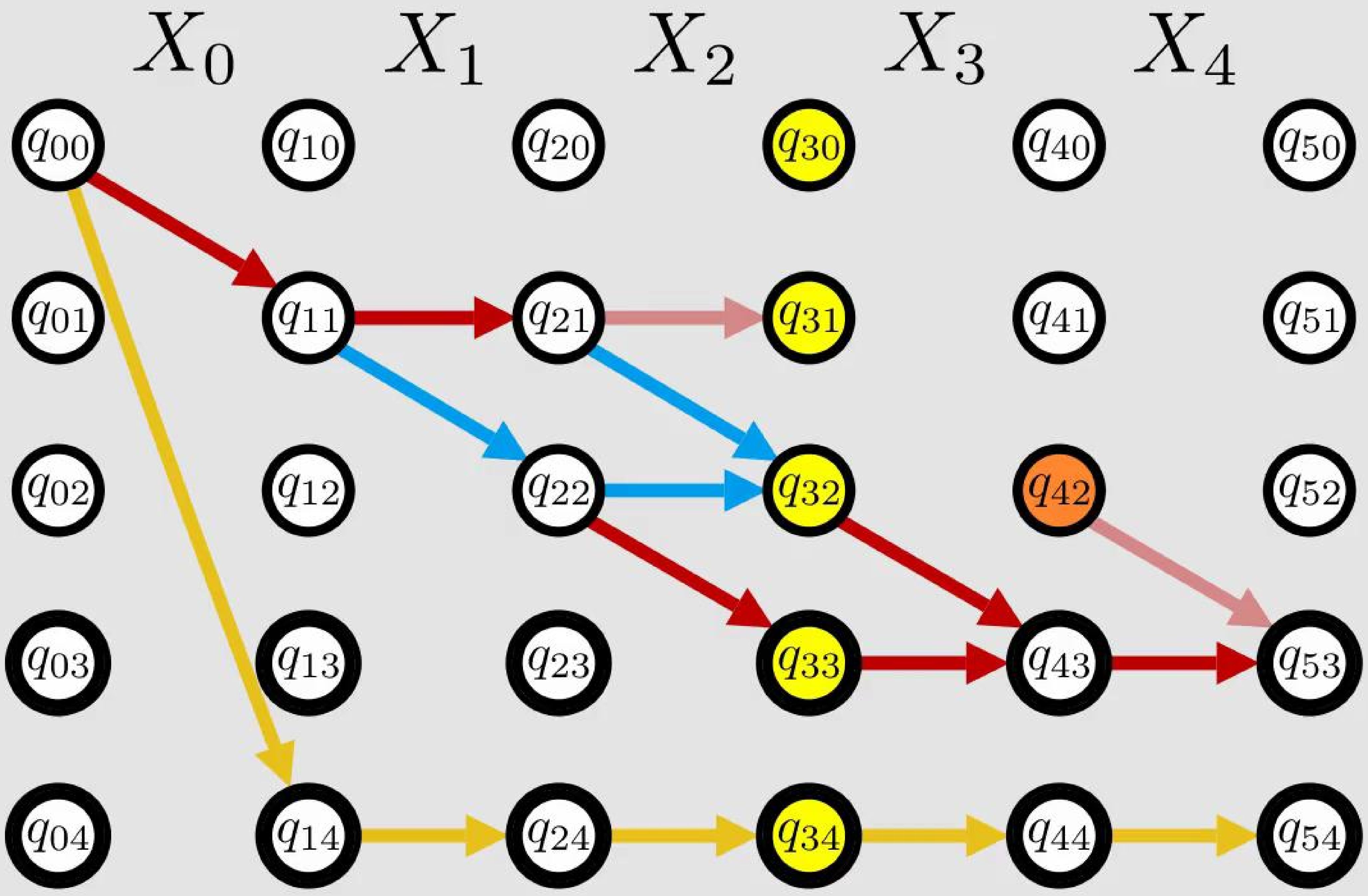
RUP $s_{3=0} \implies \bar{s}_{4=2}$

RUP $s_{3=1} \implies \bar{s}_{4=2}$

RUP $s_{3=2} \implies \bar{s}_{4=2}$

RUP $s_{3=3} \implies \bar{s}_{4=2}$

RUP $s_{3=4} \implies \bar{s}_{4=2}$



$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \wedge s_{i=q} \implies s_{i+1=\delta(q,j)}$

RUP $\bar{s}_{2=1} \vee \bar{x}_{2=1}$

RUP $s_{3=0} \implies \bar{s}_{4=2}$

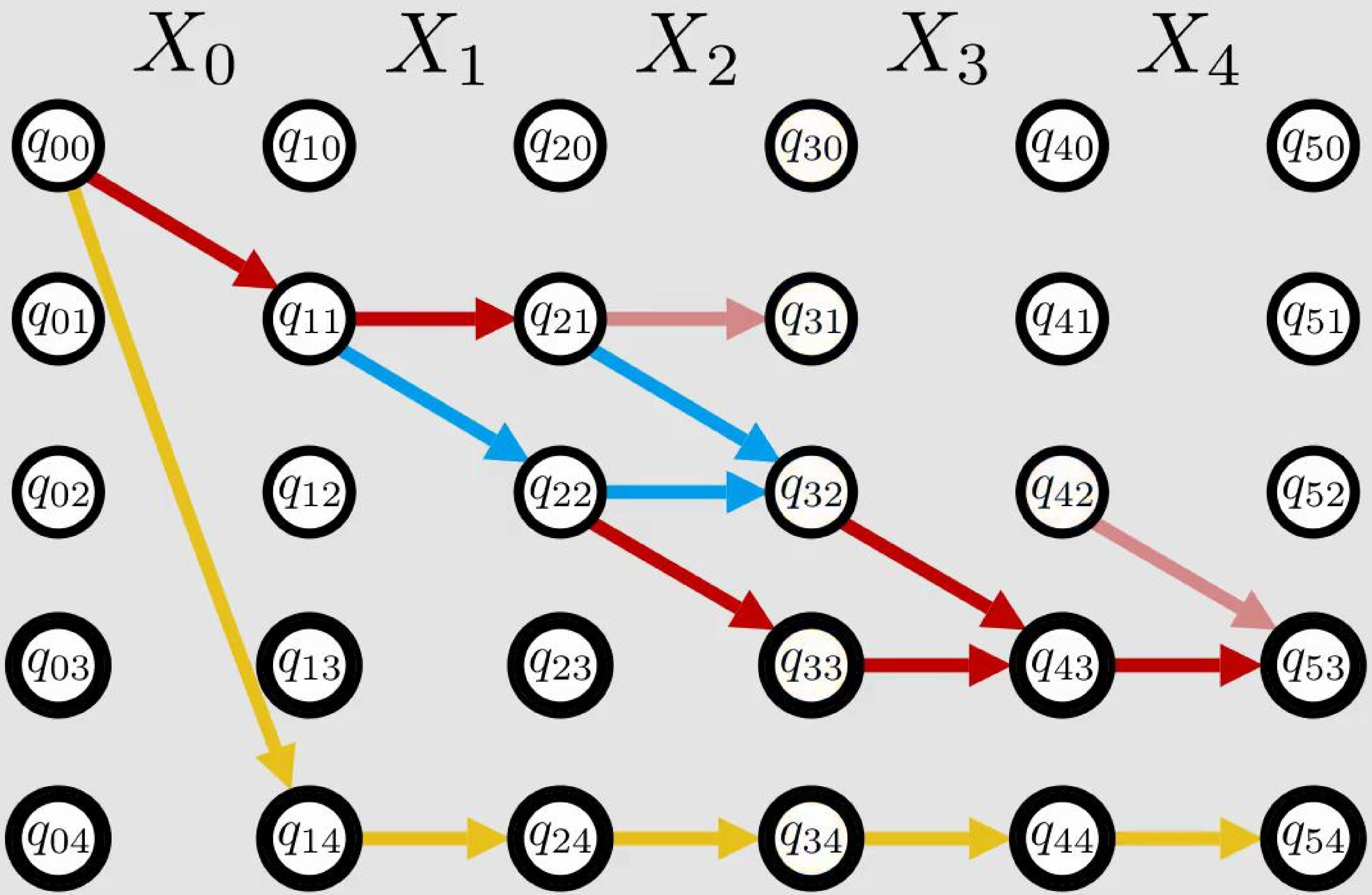
RUP $s_{3=1} \implies \bar{s}_{4=2}$

RUP $s_{3=2} \implies \bar{s}_{4=2}$

RUP $s_{3=3} \implies \bar{s}_{4=2}$

RUP $s_{3=4} \implies \bar{s}_{4=2}$

RUP $\bar{s}_{4=2} \vee s_{5=3}$



$s_i :=$ The state after processing i variables

$s_{0=0} \geq 1$

$s_{5=3} + s_{5=4} \geq 1$

For each $X_i, j \in \text{dom}(X_i)$, and $q \in Q$:

$x_{i=j} \land s_{i=q} \implies s_{i+1=\delta(q,j)}$

RUP $\bar{s}_{2=1} \lor \bar{x}_{2=1}$

RUP $s_{3=0} \implies \bar{s}_{4=2}$

RUP $s_{3=1} \implies \bar{s}_{4=2}$

RUP $s_{3=2} \implies \bar{s}_{4=2}$

RUP $s_{3=3} \implies \bar{s}_{4=2}$

RUP $s_{3=4} \implies \bar{s}_{4=2}$

RUP $\bar{s}_{4=2} \lor s_{5=3}$

The Circuit constraint

$$X_0, \ldots, X_{n-1}$$

$$\{0, \ldots, n-1\}$$

The Circuit constraint

$$\text{Circuit}(X_0, \ldots, X_{n-1})$$

$$\{0, \ldots, n-1\}$$

The Circuit constraint

$$\text{Circuit}(X_0, X_1, X_2, X_3, X_4, X_5)$$

$$\{0, \ldots, n-1\}$$

The Circuit constraint

$$\text{Circuit}(X_0, X_1, X_2, X_3, X_4, X_5)$$

$$\{0, 1, 2, 3, 4, 5\}$$

The Circuit constraint

X_0

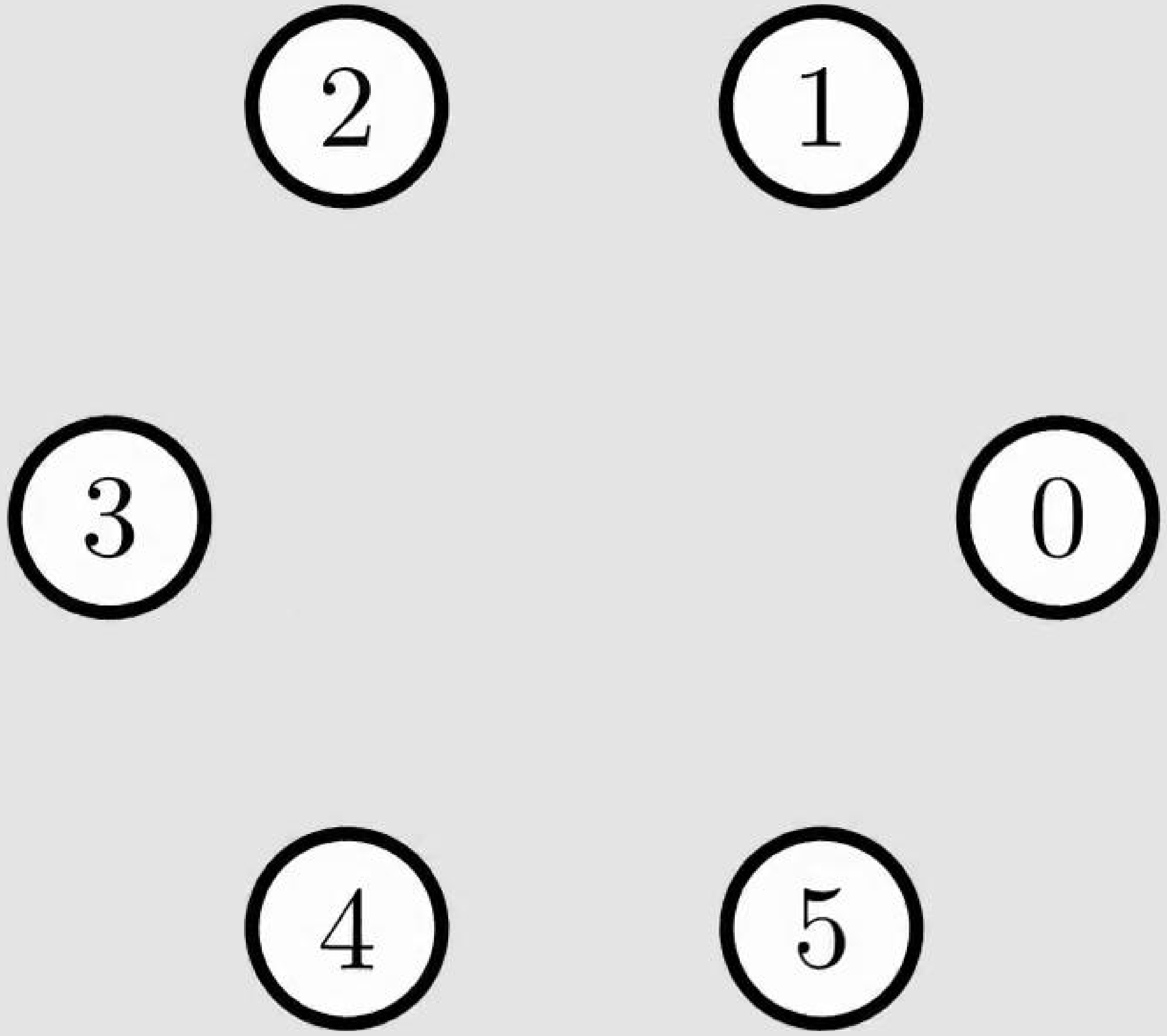
X_1

X_2

X_3

X_4

X_5



The Circuit constraint

X_0

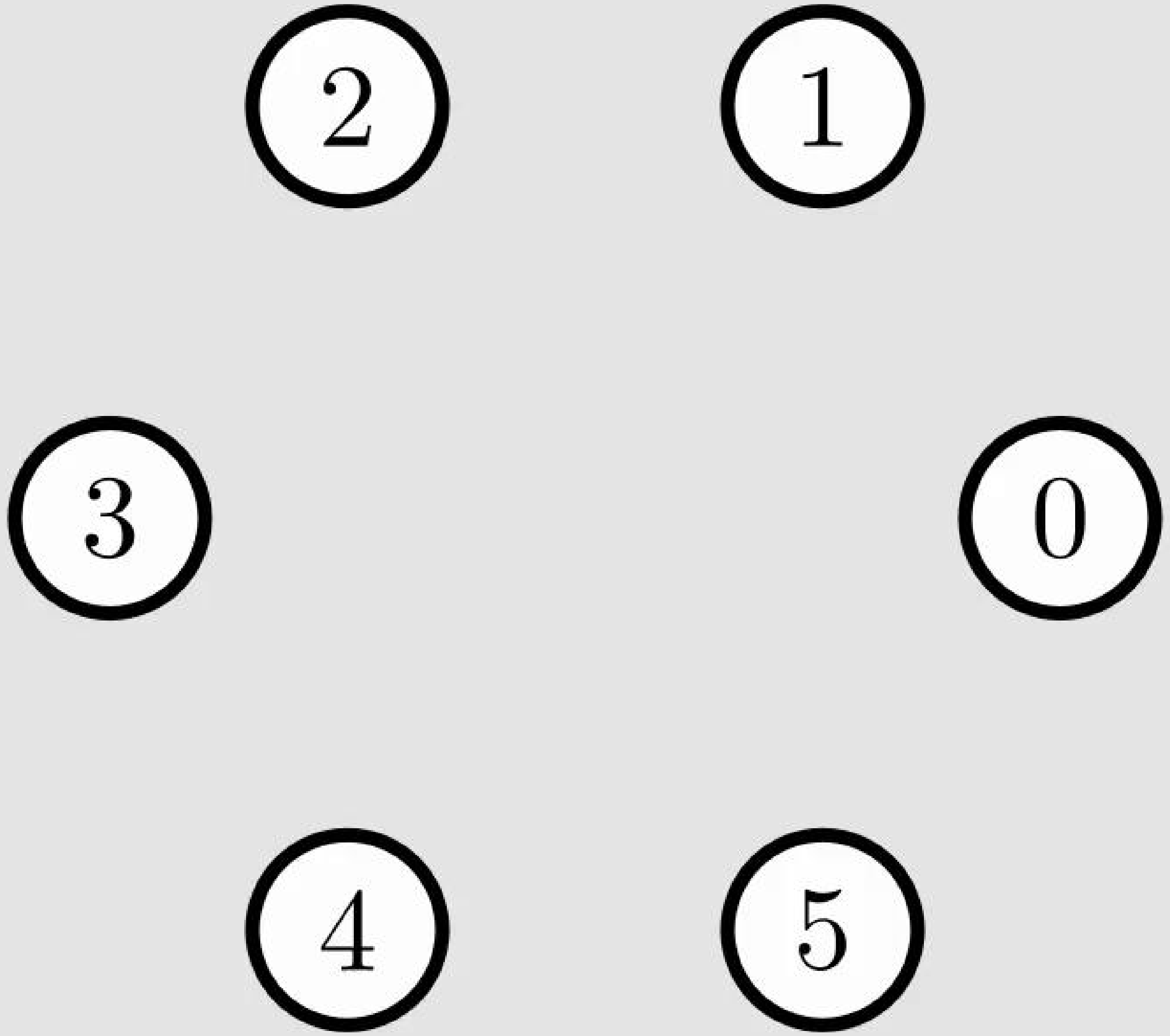
X_1

$X_2 = 5$

X_3

X_4

X_5



The Circuit constraint

X_0

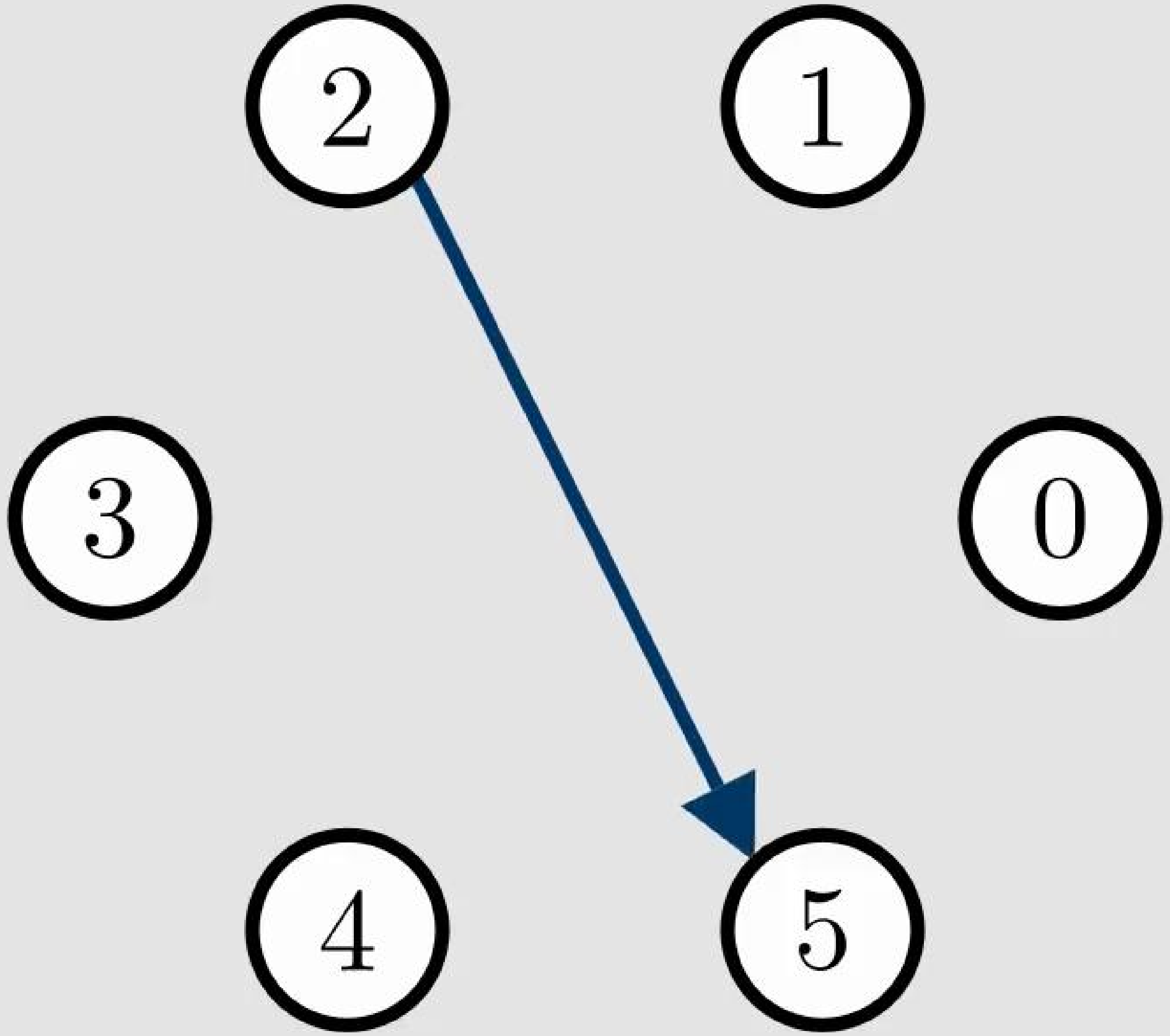
X_1

$X_2 = 5$

X_3

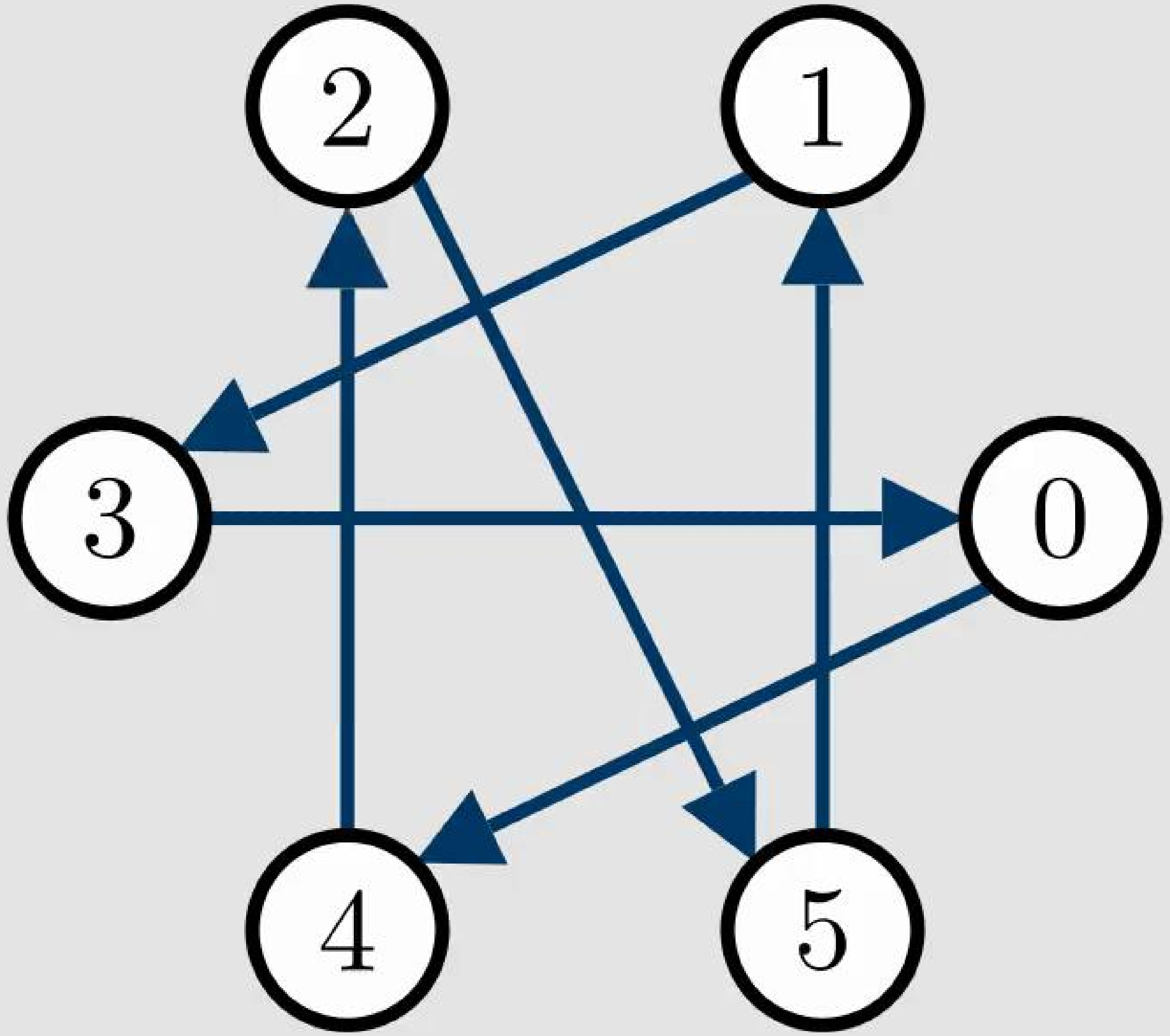
X_4

X_5



The Circuit constraint

$$X_0 = 4$$
$$X_1 = 3$$
$$X_2 = 5$$
$$X_3 = 0$$
$$X_4 = 2$$
$$X_5 = 1$$



Enforcing Circuit:

X_0

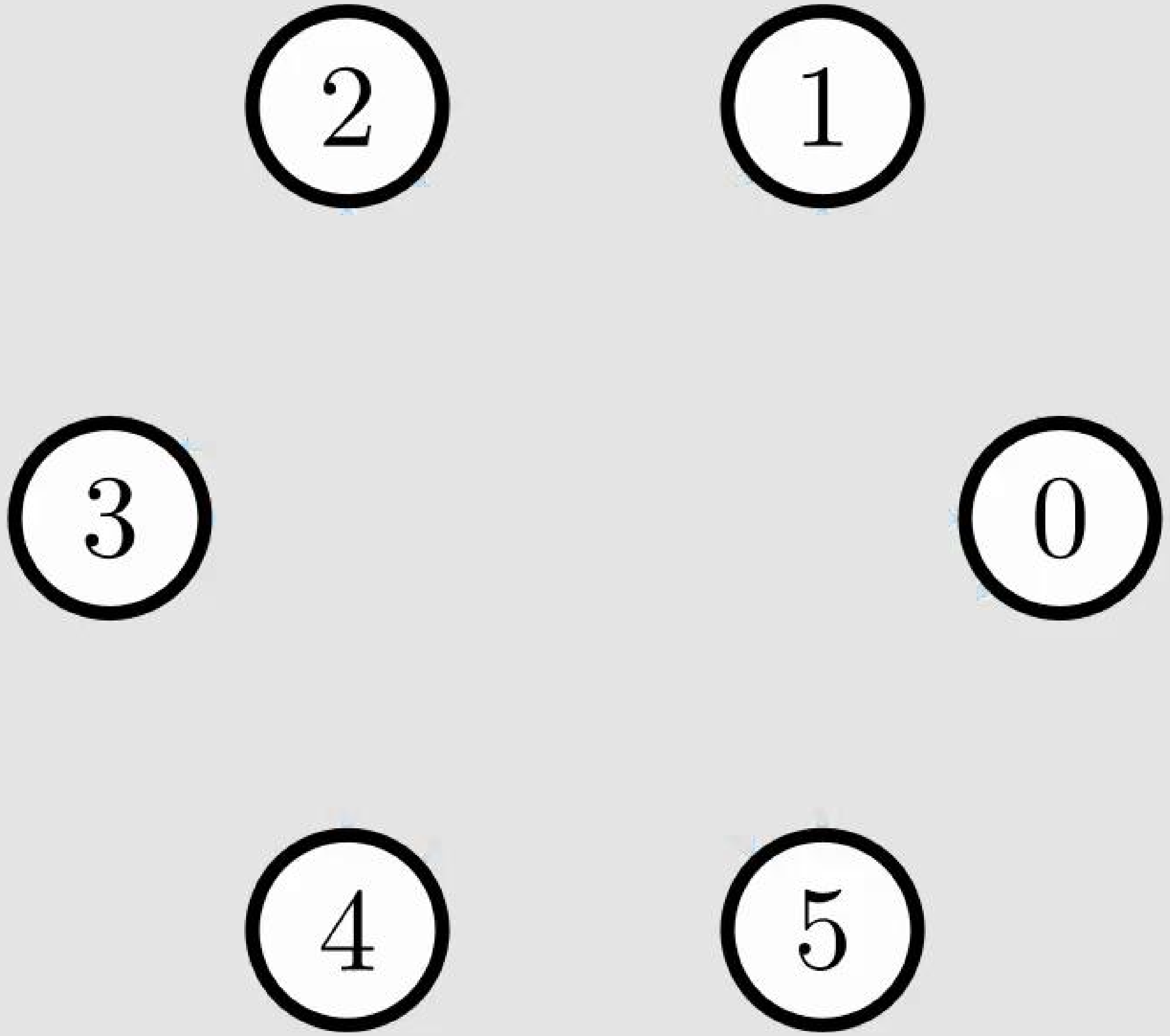
X_1

X_2

X_3

X_4

X_5



Enforcing Circuit:

X_0

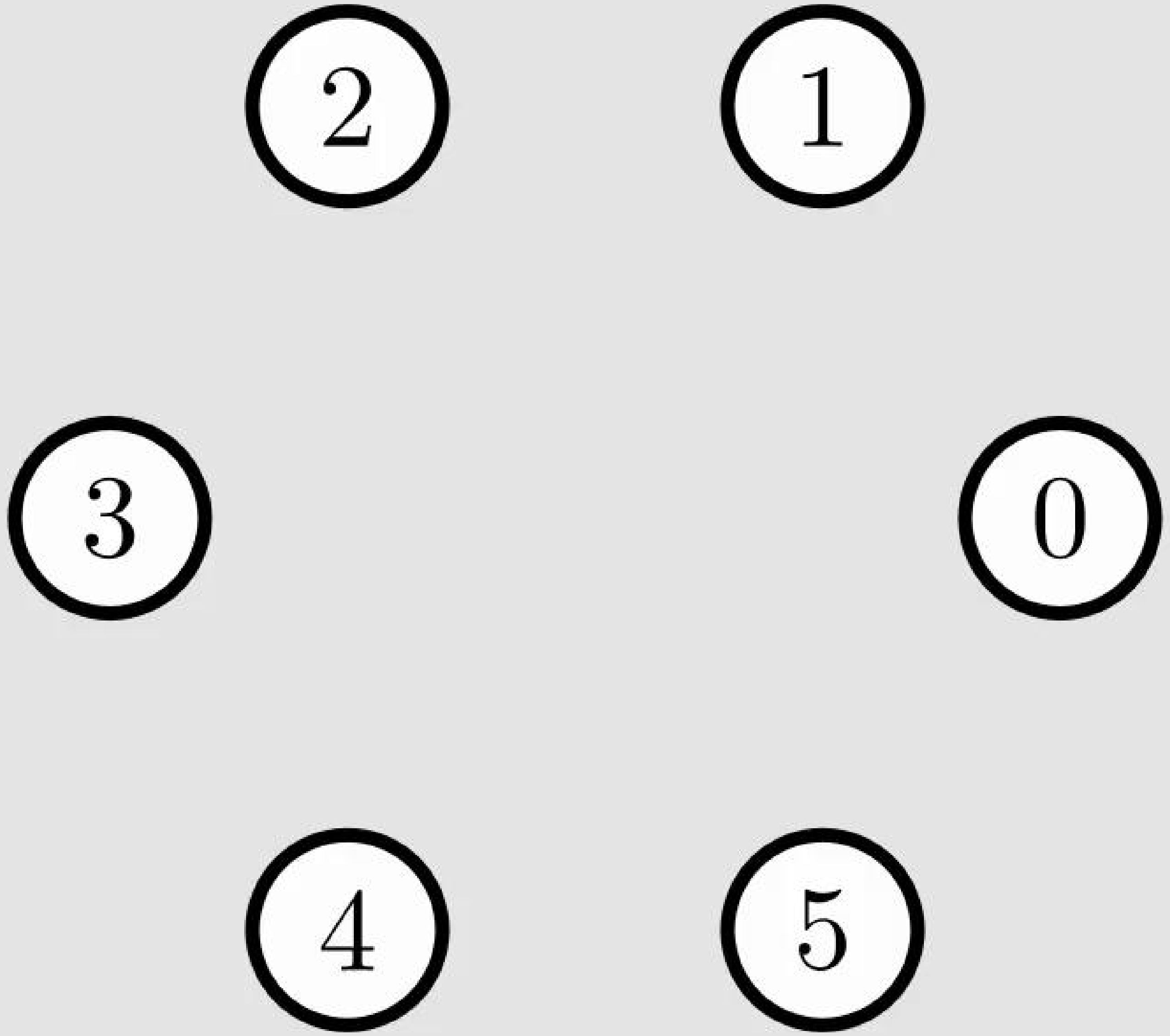
X_1

X_2

X_3

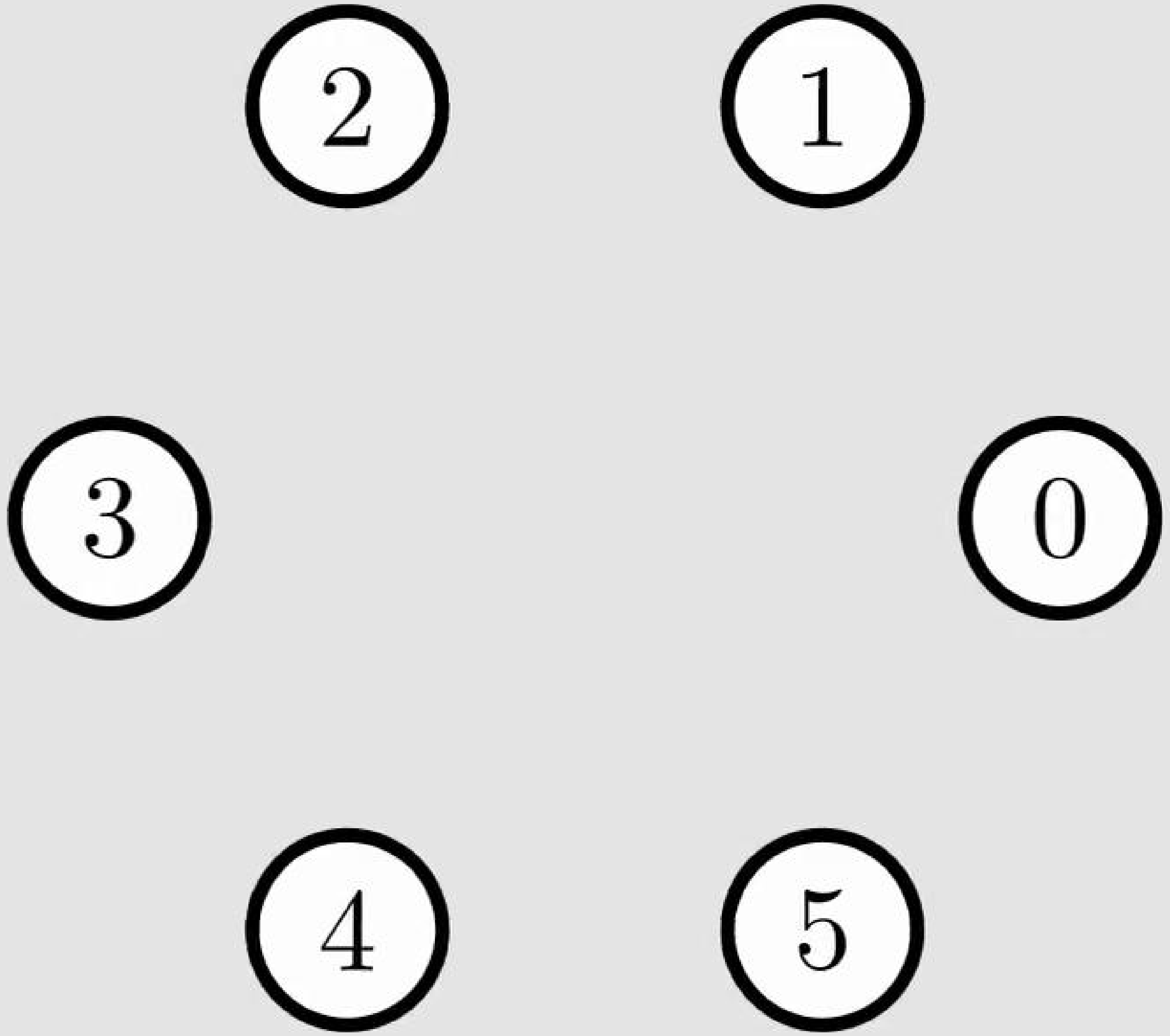
X_4

X_5



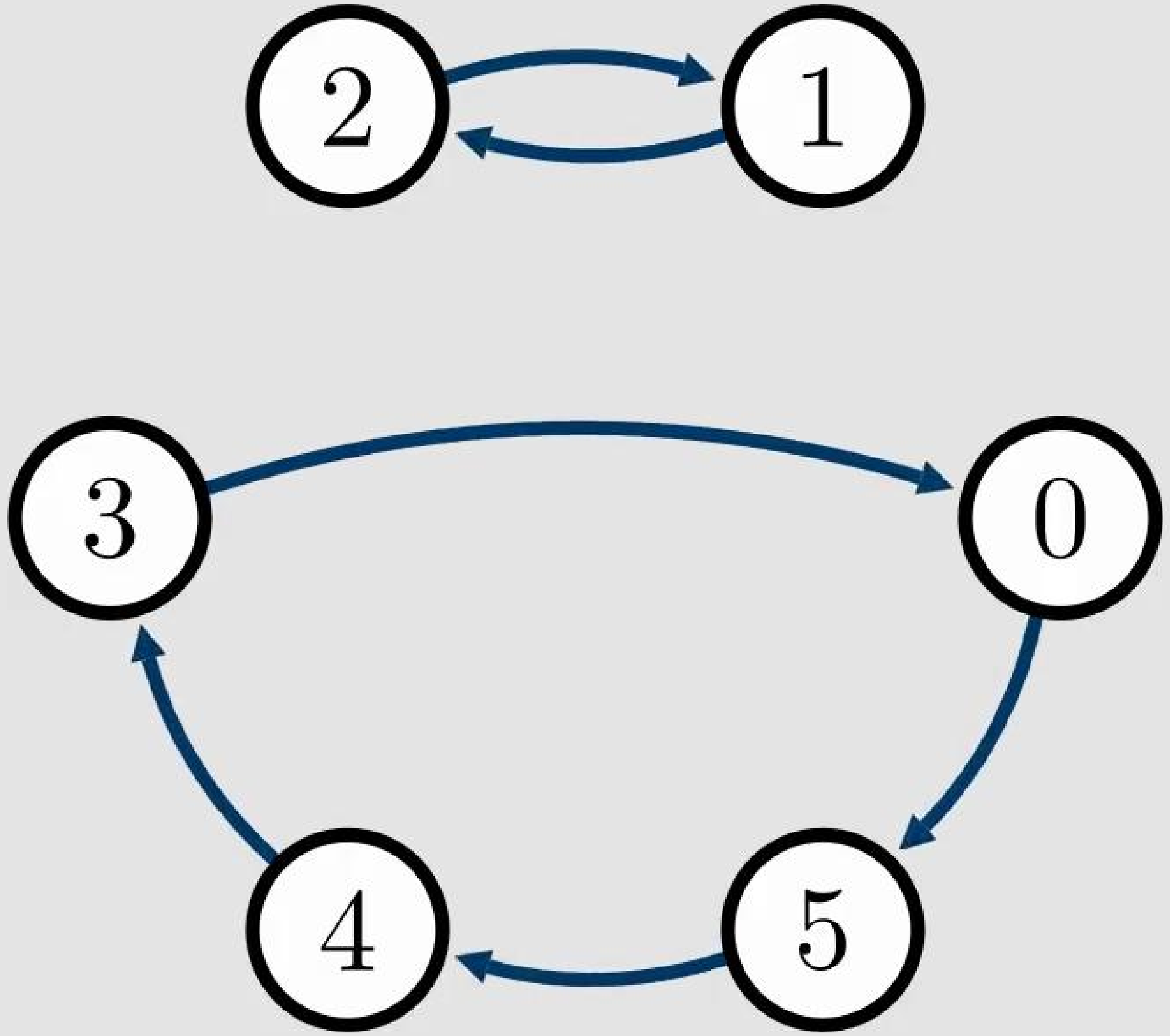
Enforcing Circuit:

$\text{AllDiff}(X_0, X_1, X_2, X_3, X_4, X_5)$



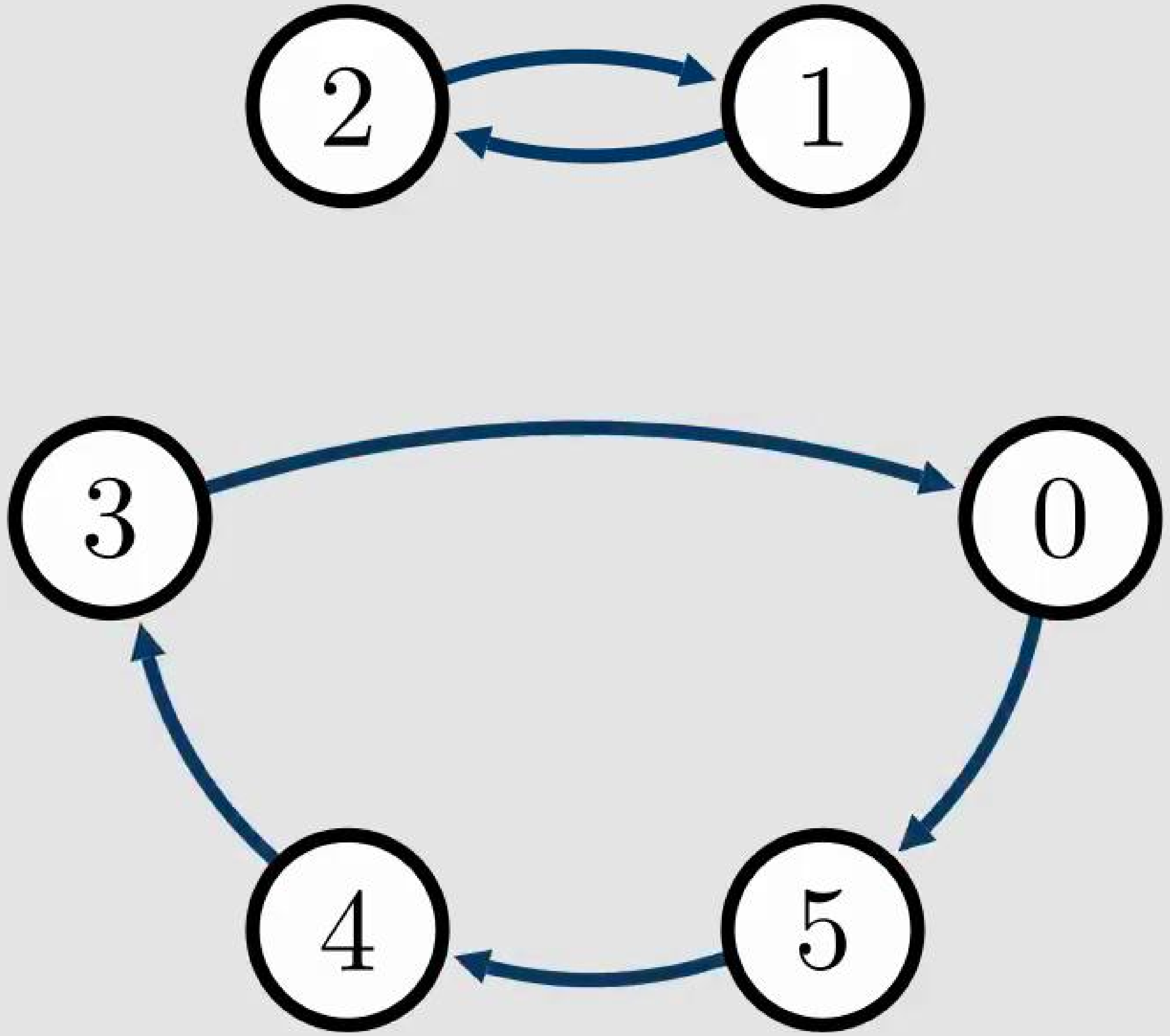
Enforcing Circuit:

AllDiff$(X_0, X_1, X_2, X_3, X_4, X_5)$



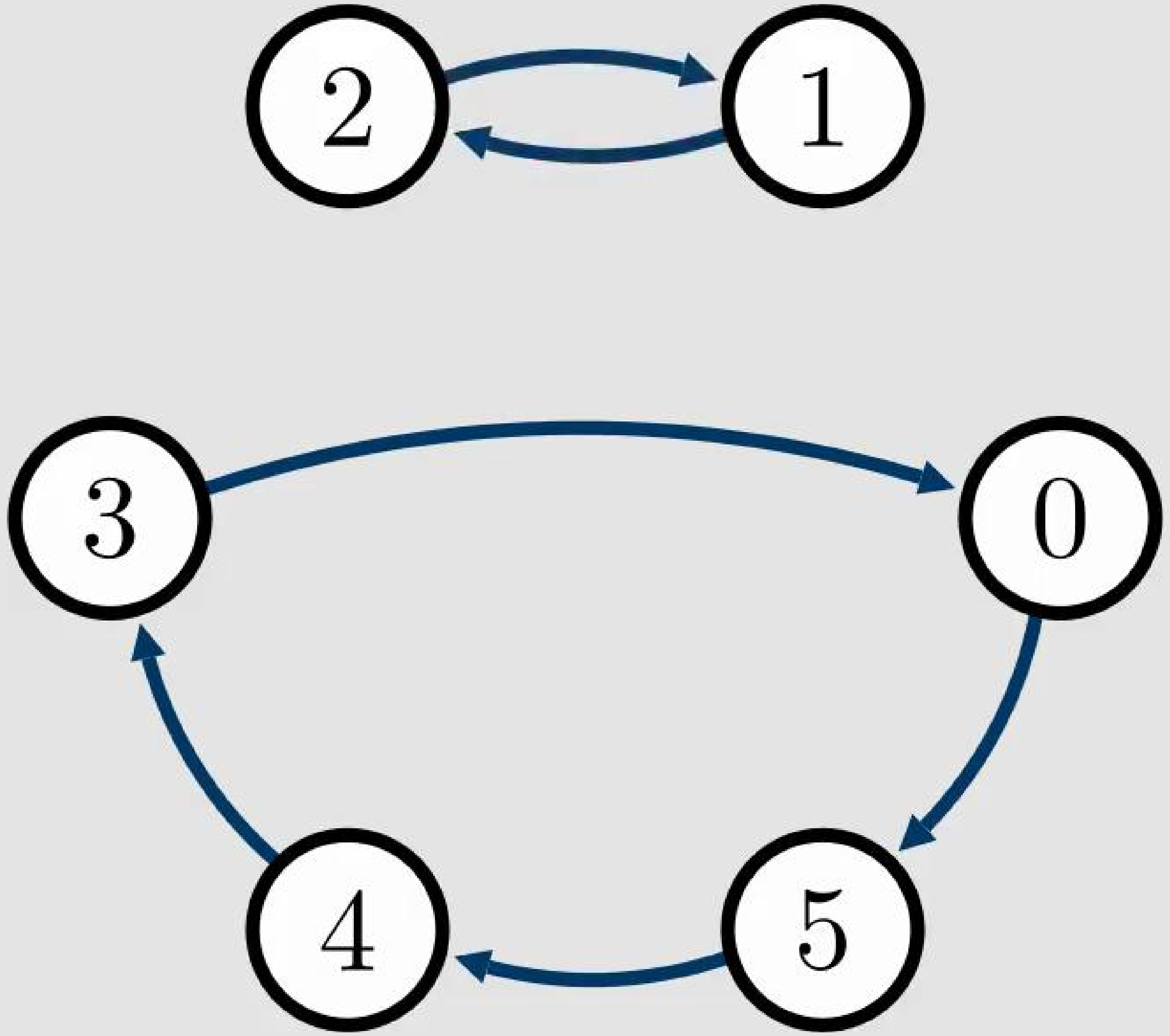
Enforcing Circuit:

AllDiff$(X_0, X_1, X_2, X_3, X_4, X_5)$



Enforcing Circuit:

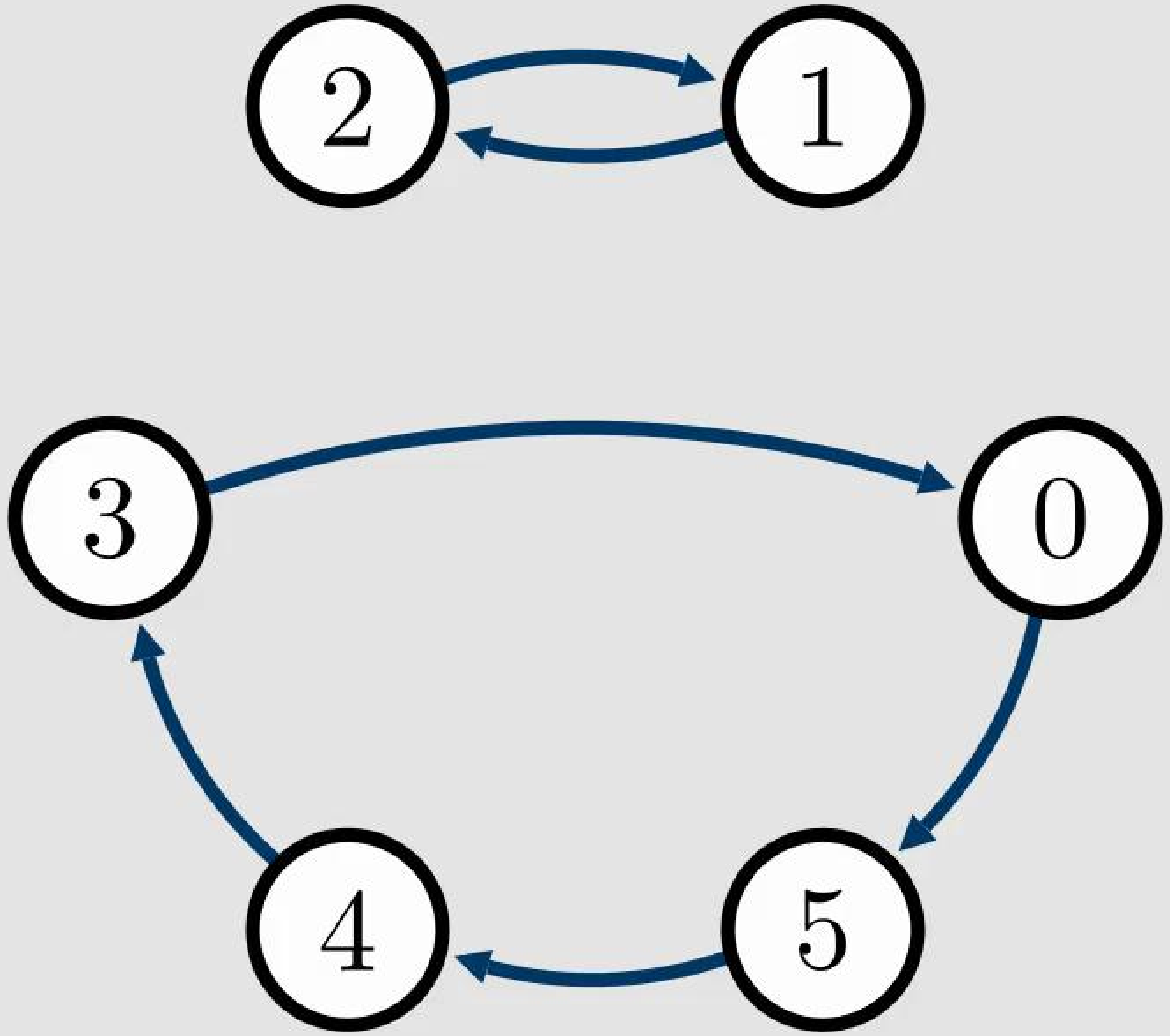
AllDiff$(X_0, X_1, X_2, X_3, X_4, X_5)$



Enforcing Circuit:

$\text{AllDiff}(X_0, X_1, X_2, X_3, X_4, X_5)$

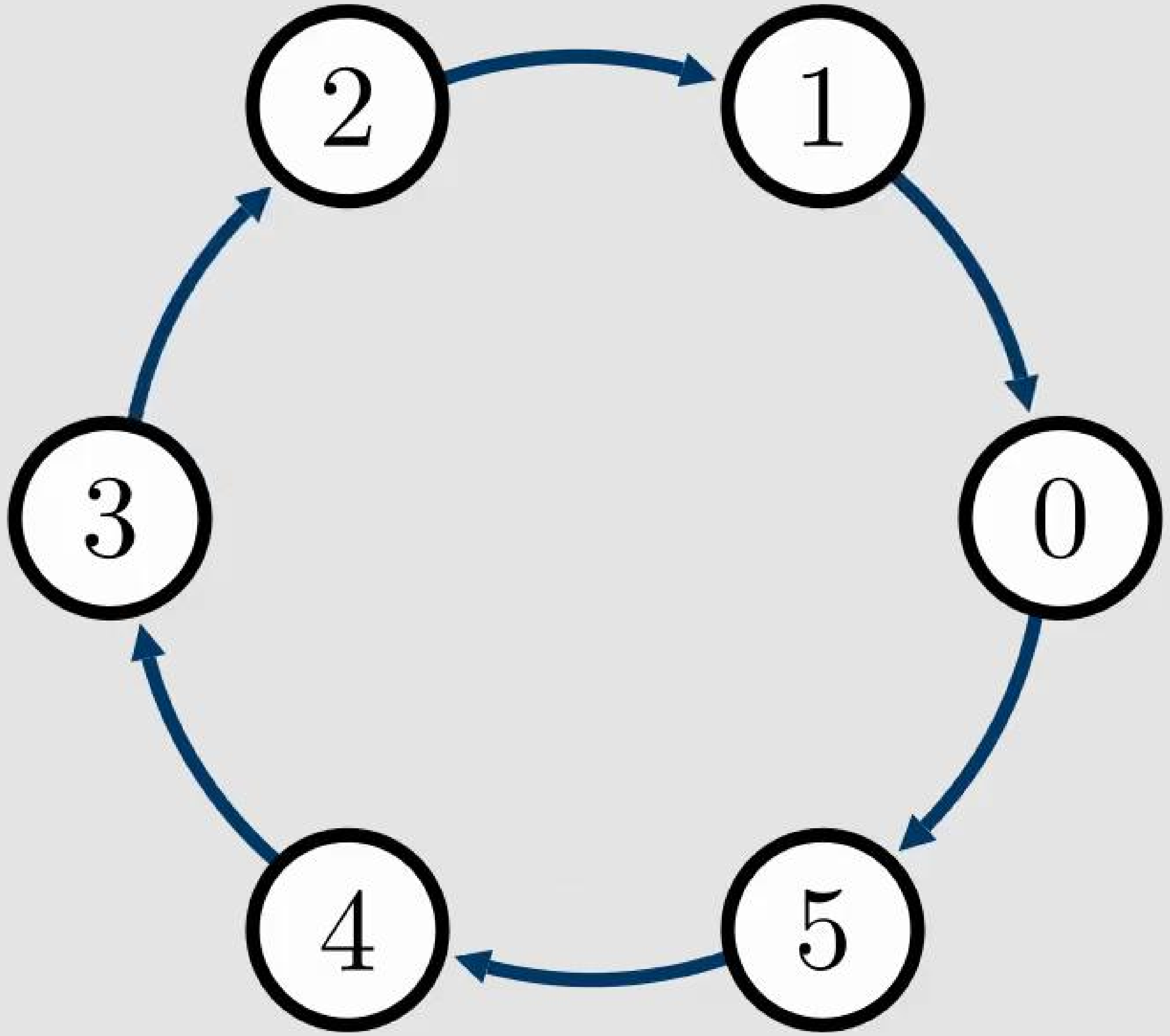
$\text{NoCycle}(X_0, X_1, X_2, X_3, X_4, X_5)$



Enforcing Circuit:

$\text{AllDiff}(X_0, X_1, X_2, X_3, X_4, X_5)$

$\text{NoCycle}(X_0, X_1, X_2, X_3, X_4, X_5)$



Consistency for Circuit:

X_0

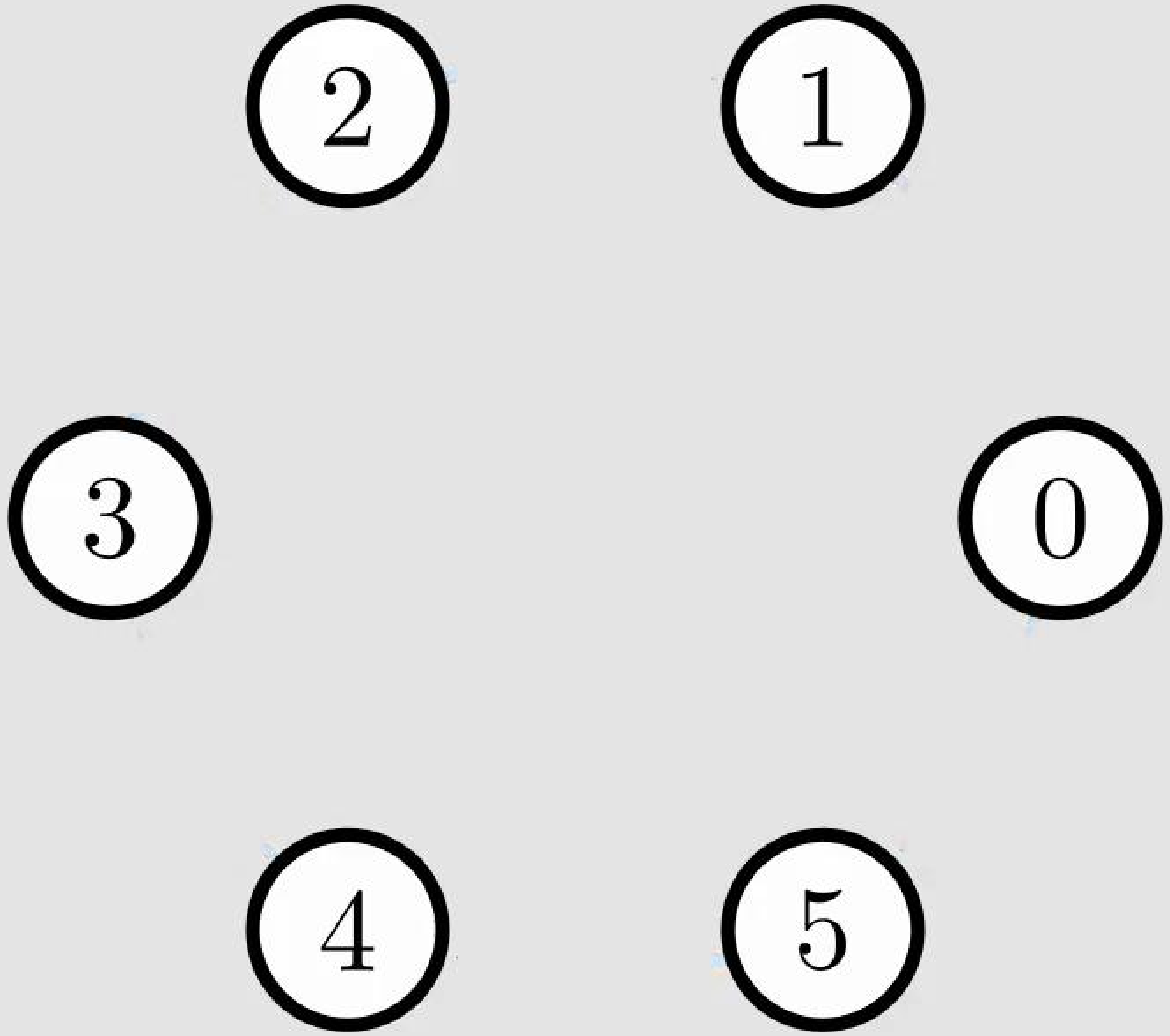
X_1

X_2

X_3

X_4

X_5



Consistency for Circuit:

X_0

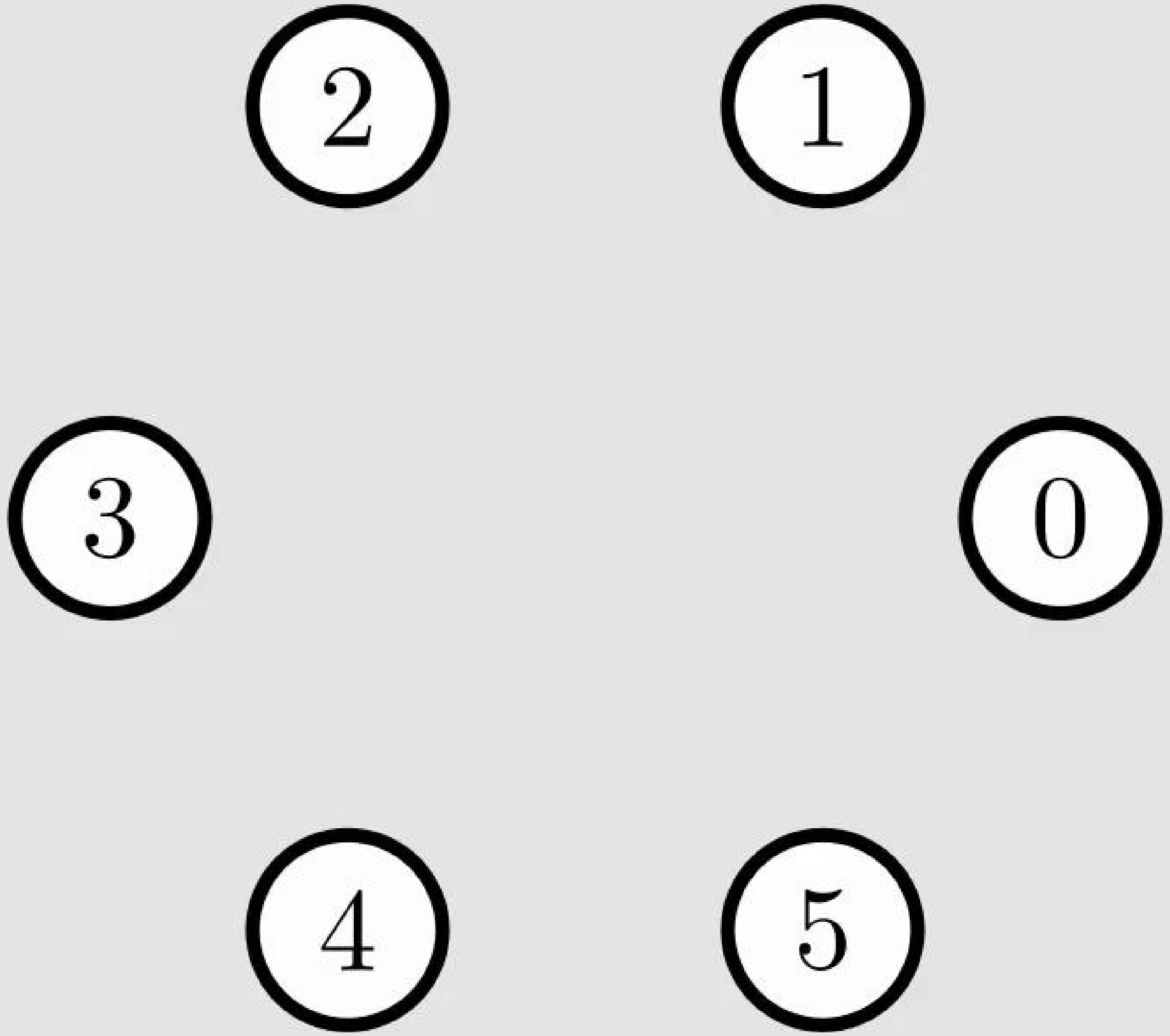
X_1

X_2

X_3

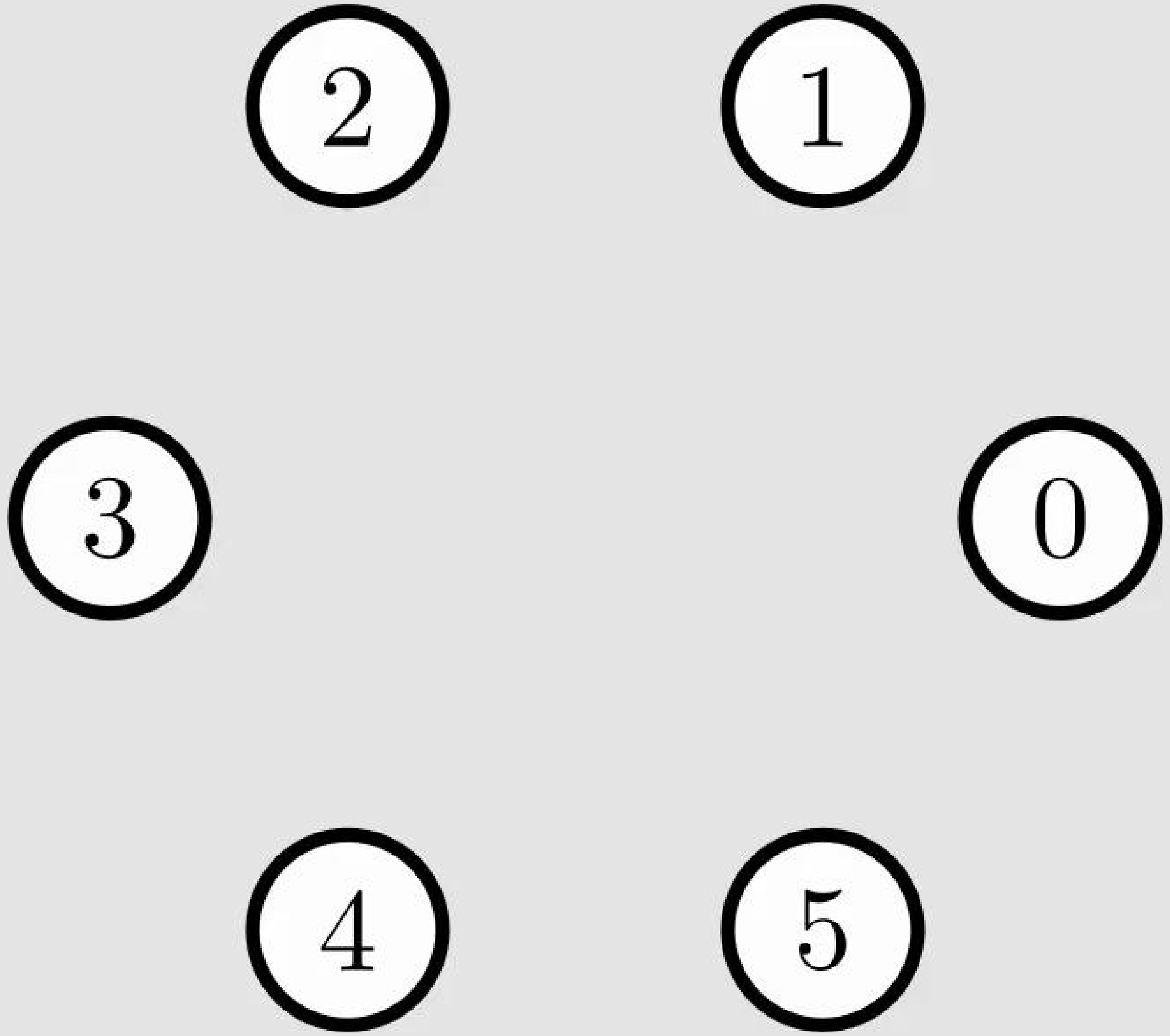
X_4

X_5



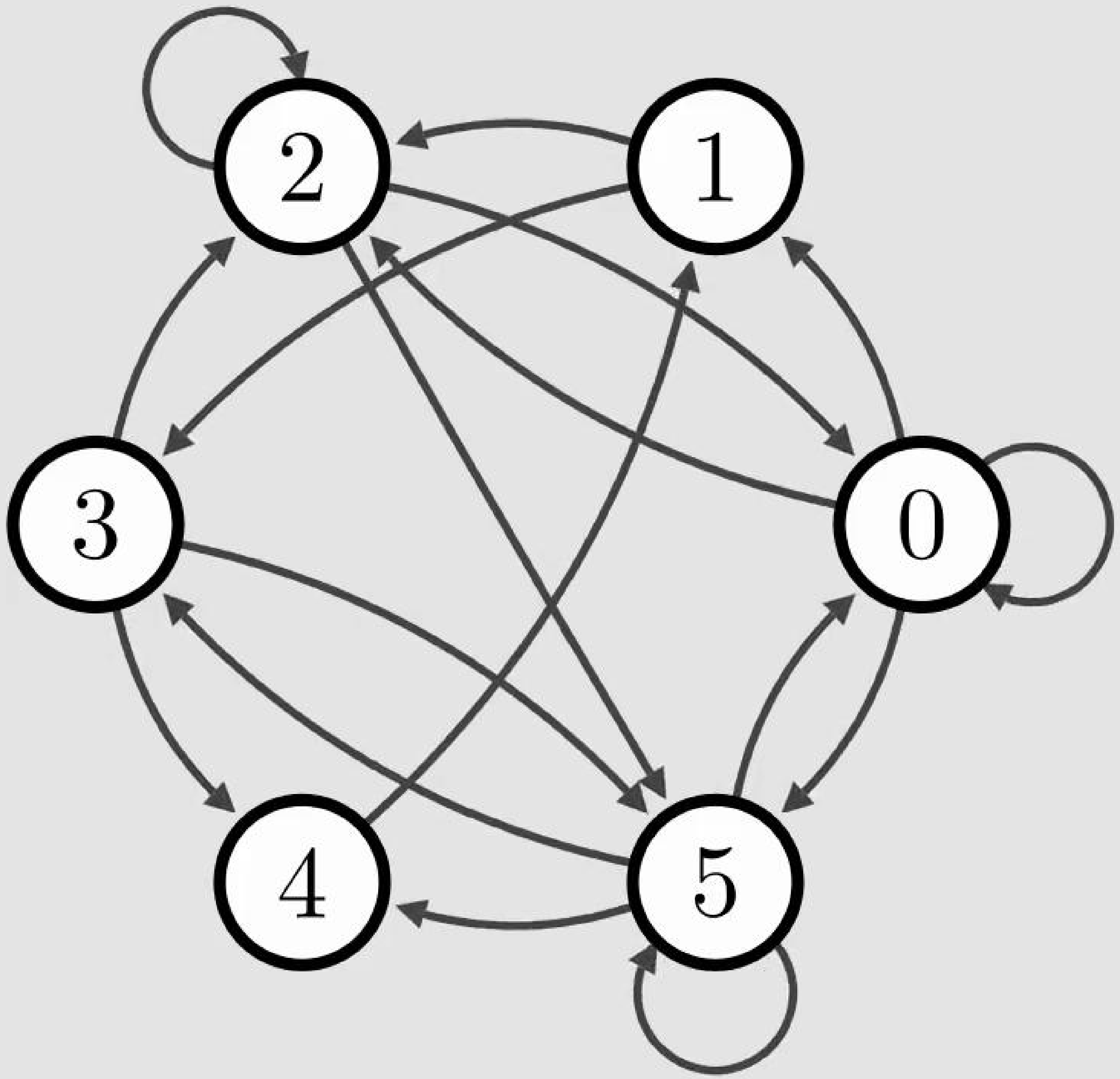
Consistency for Circuit:

$$X_0 \in \{0, 1, 2, 5\}$$
$$X_1 \in \{2, 3\}$$
$$X_2 \in \{0, 2, 5\}$$
$$X_3 \in \{2, 4, 5\}$$
$$X_4 \in \{1\}$$
$$X_5 \in \{0, 3, 4, 5\}$$



Consistency for Circuit:

$$X_0 \in \{0, 1, 2, 5\}$$
$$X_1 \in \{2, 3\}$$
$$X_2 \in \{0, 2, 5\}$$
$$X_3 \in \{2, 4, 5\}$$
$$X_4 \in \{1\}$$
$$X_5 \in \{0, 3, 4, 5\}$$



Consistency for Circuit:

$$X_0 \in \{0, 1, 2, 5\}$$

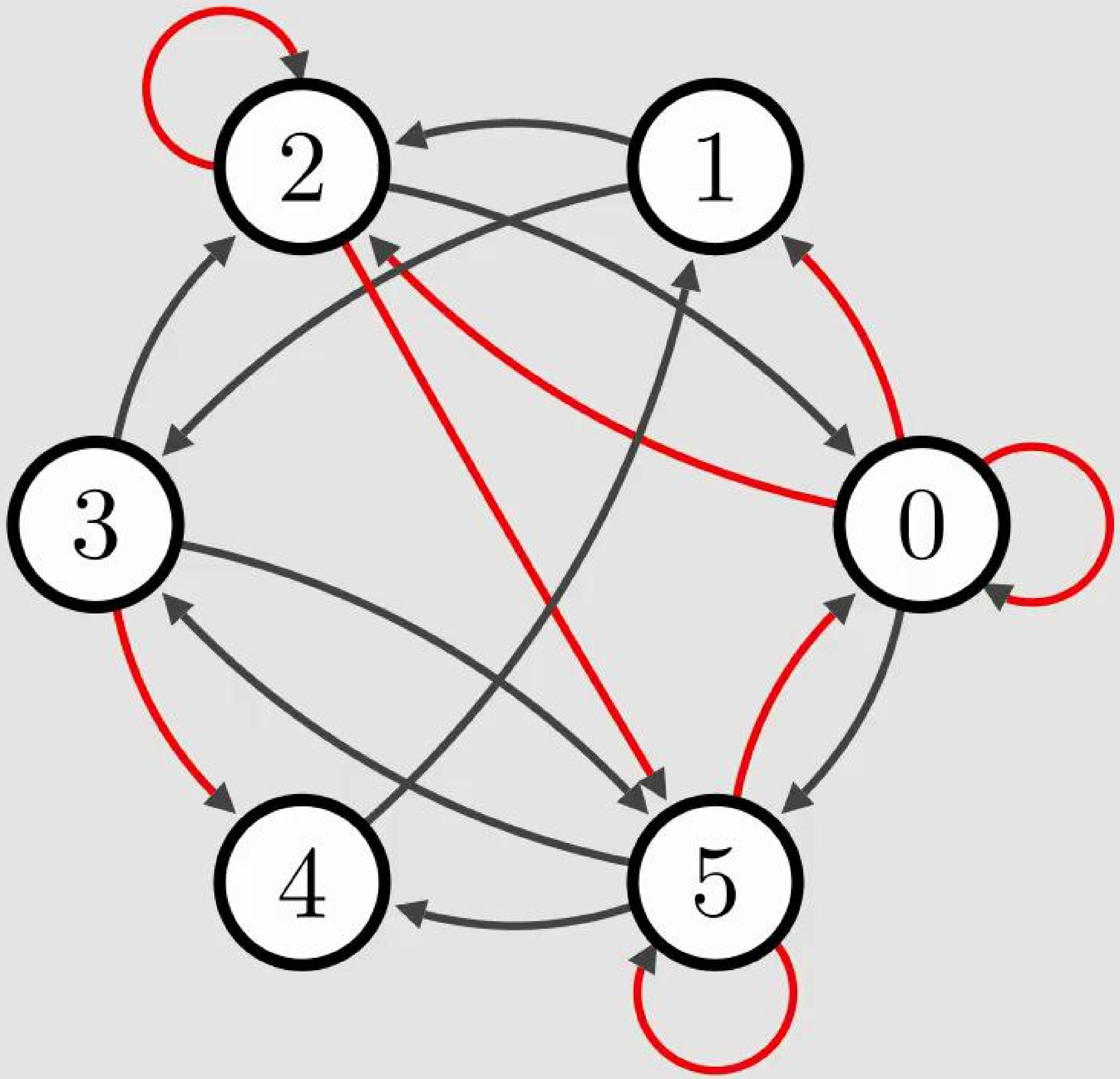
$$X_1 \in \{2, 3\}$$

$$X_2 \in \{0, 2, 5\}$$

$$X_3 \in \{2, 4, 5\}$$

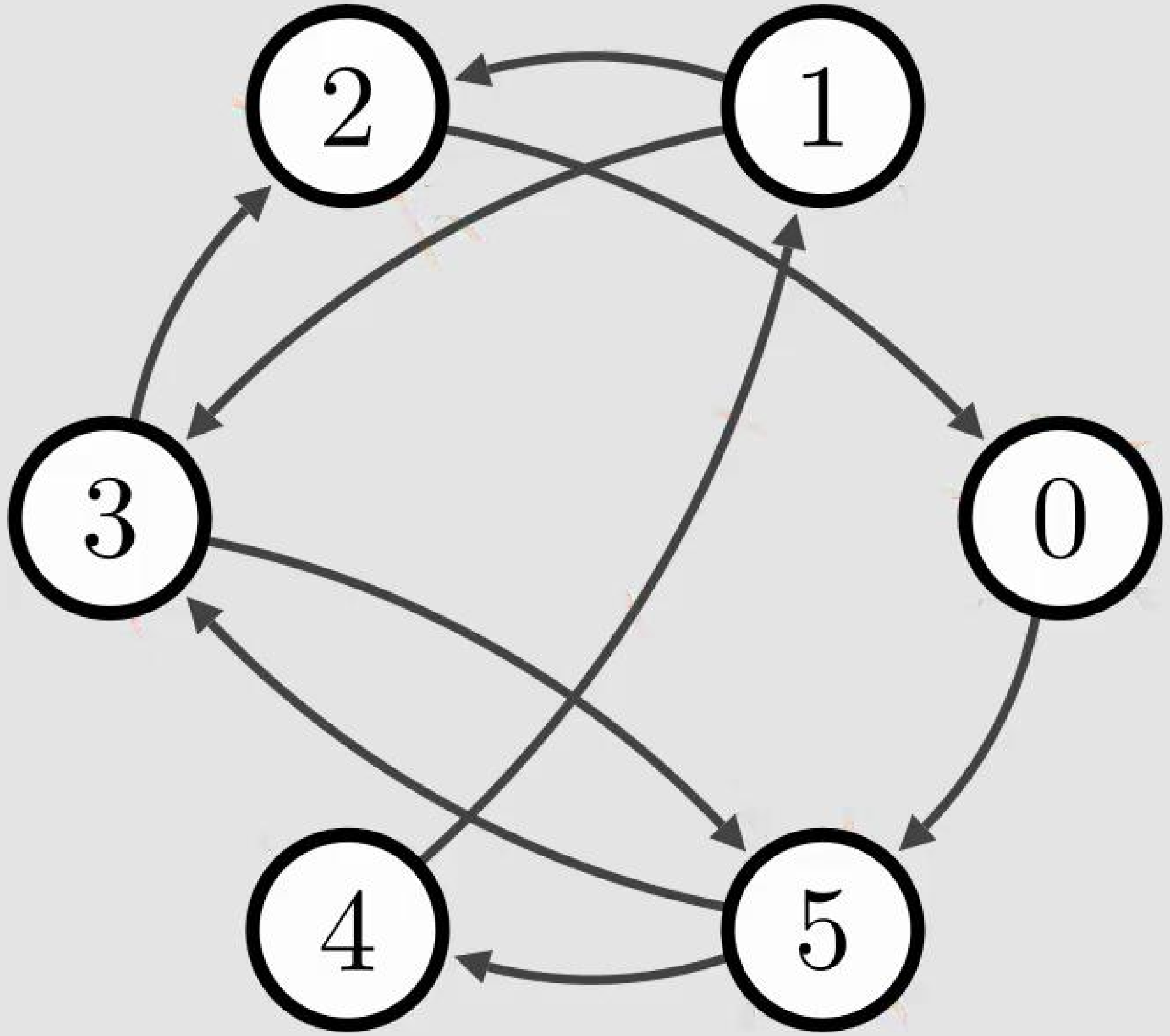
$$X_4 \in \{1\}$$

$$X_5 \in \{0, 3, 4, 5\}$$



Consistency for Circuit:

$$X_0 \in \{5\}$$
$$X_1 \in \{2, 3\}$$
$$X_2 \in \{0\}$$
$$X_3 \in \{2, 5\}$$
$$X_4 \in \{1\}$$
$$X_5 \in \{3, 4\}$$



Consistency for Circuit:

$$X_0 \in \{5\}$$

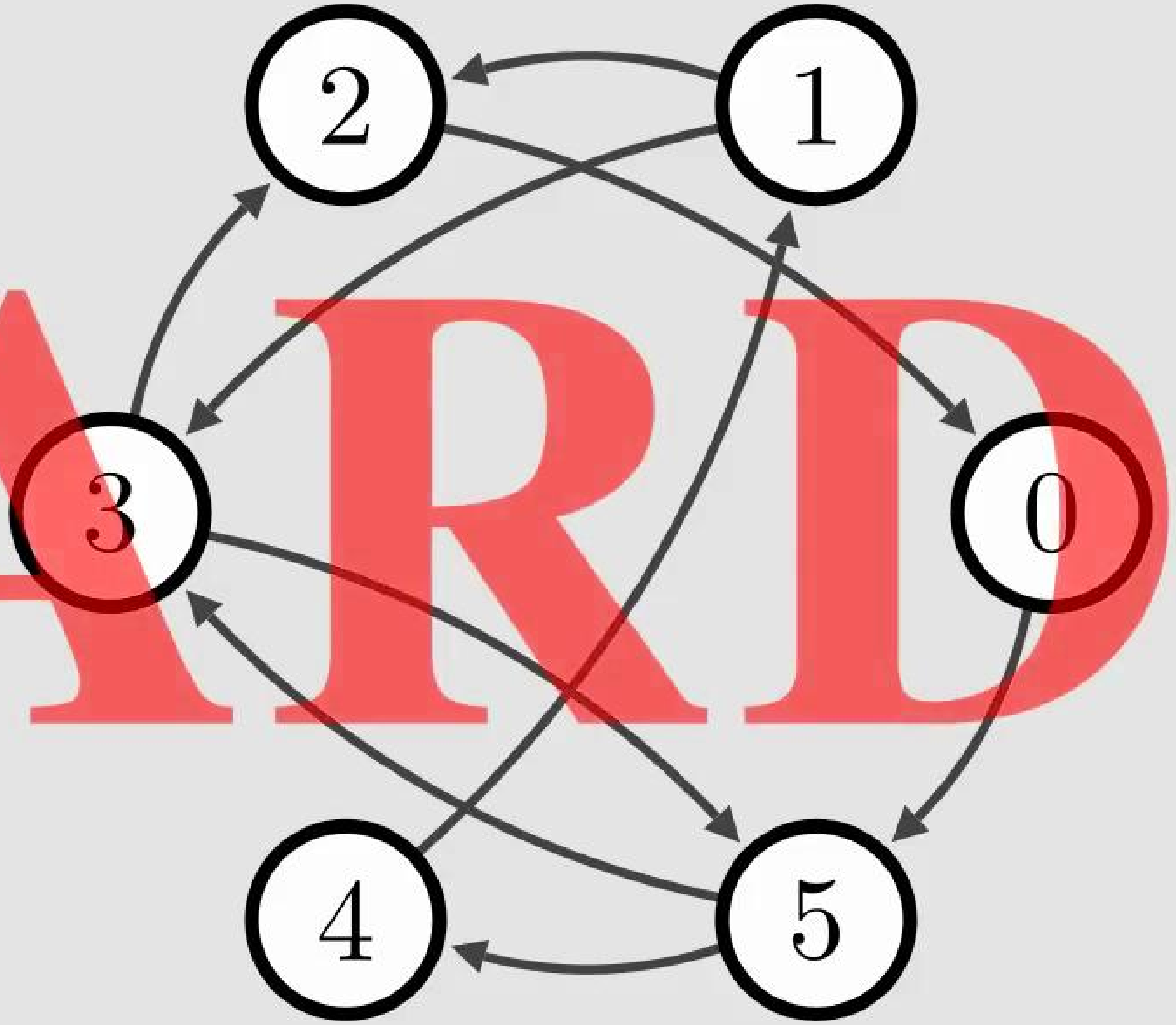
$$X_1 \in \{2, 3\}$$

$$X_2 \in \{0\}$$

$$X_3 \in \{2, 5\}$$

$$X_4 \in \{1\}$$

$$X_5 \in \{3, 4\}$$



NP HARD

(Partial) Consistency for Circuit

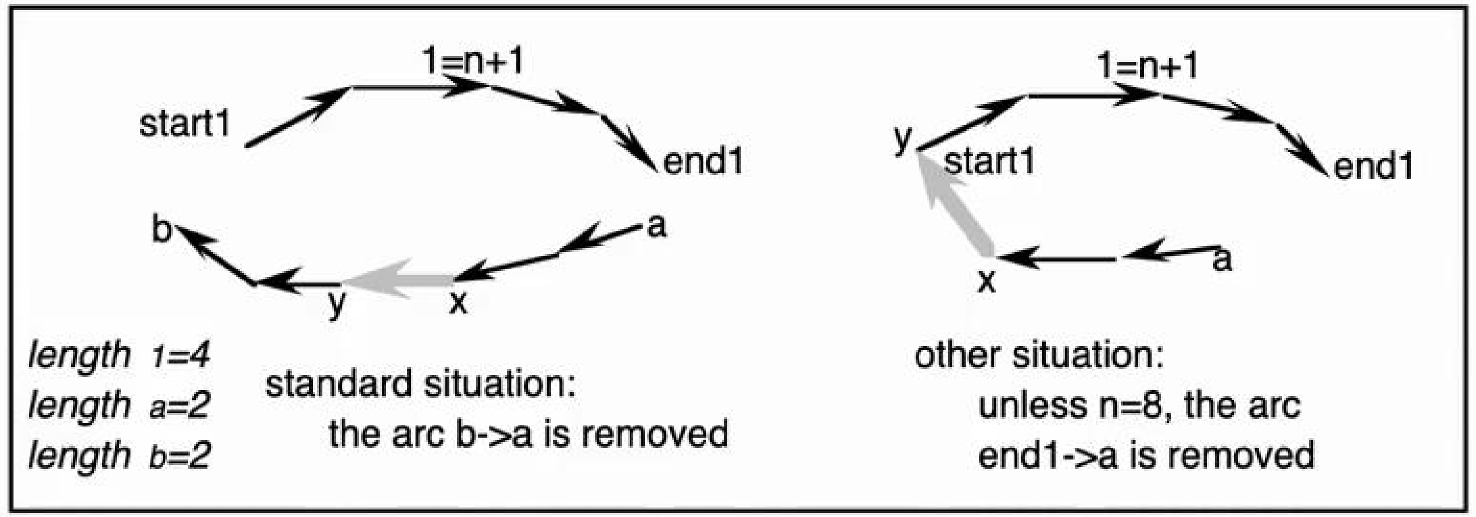


Figure 1: Propagation of the nocycle constraint

- If $x=end_1$ and $length_1+length_b<n\text{-}2$ we infer $Next(b) \neq start_1$.
- If $y=start_1$ and $length_1+length_a<n\text{-}2$ we infer $Next(end_1) \neq a$
- Otherwise, we infer $Next(b) \neq a$.

Caseau, Y. and Laburthe, F., 1997, July.
Solving Small TSPs with Constraints. In ICLP (Vol. 97, p. 104).

16 Constraints (2014) 19:1–29

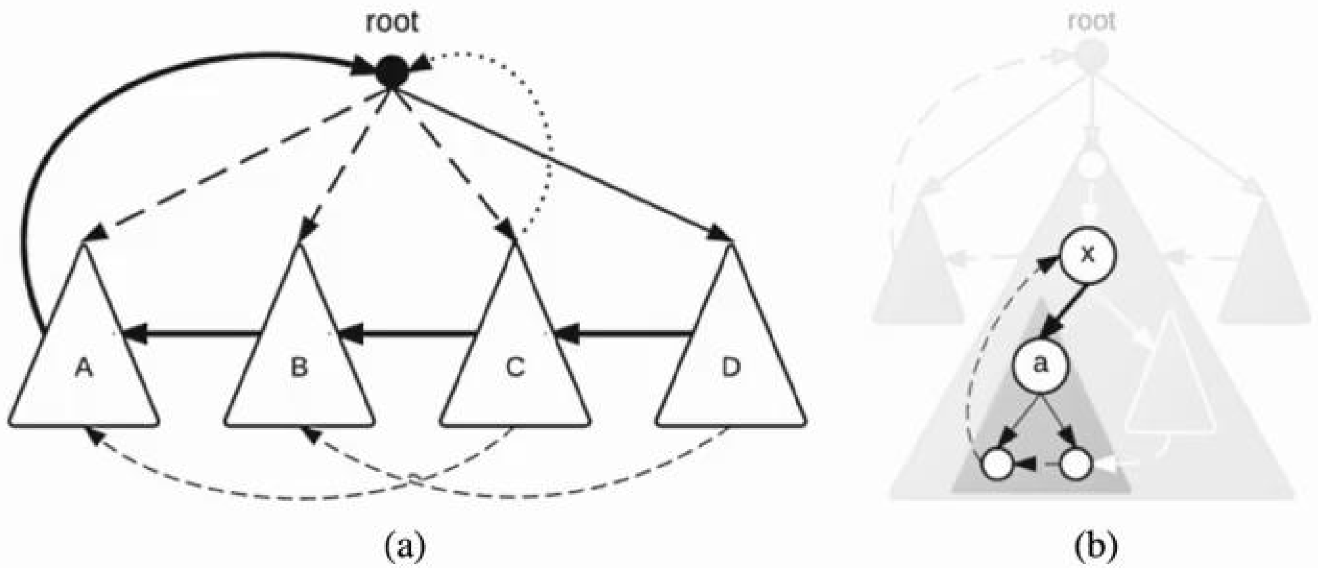
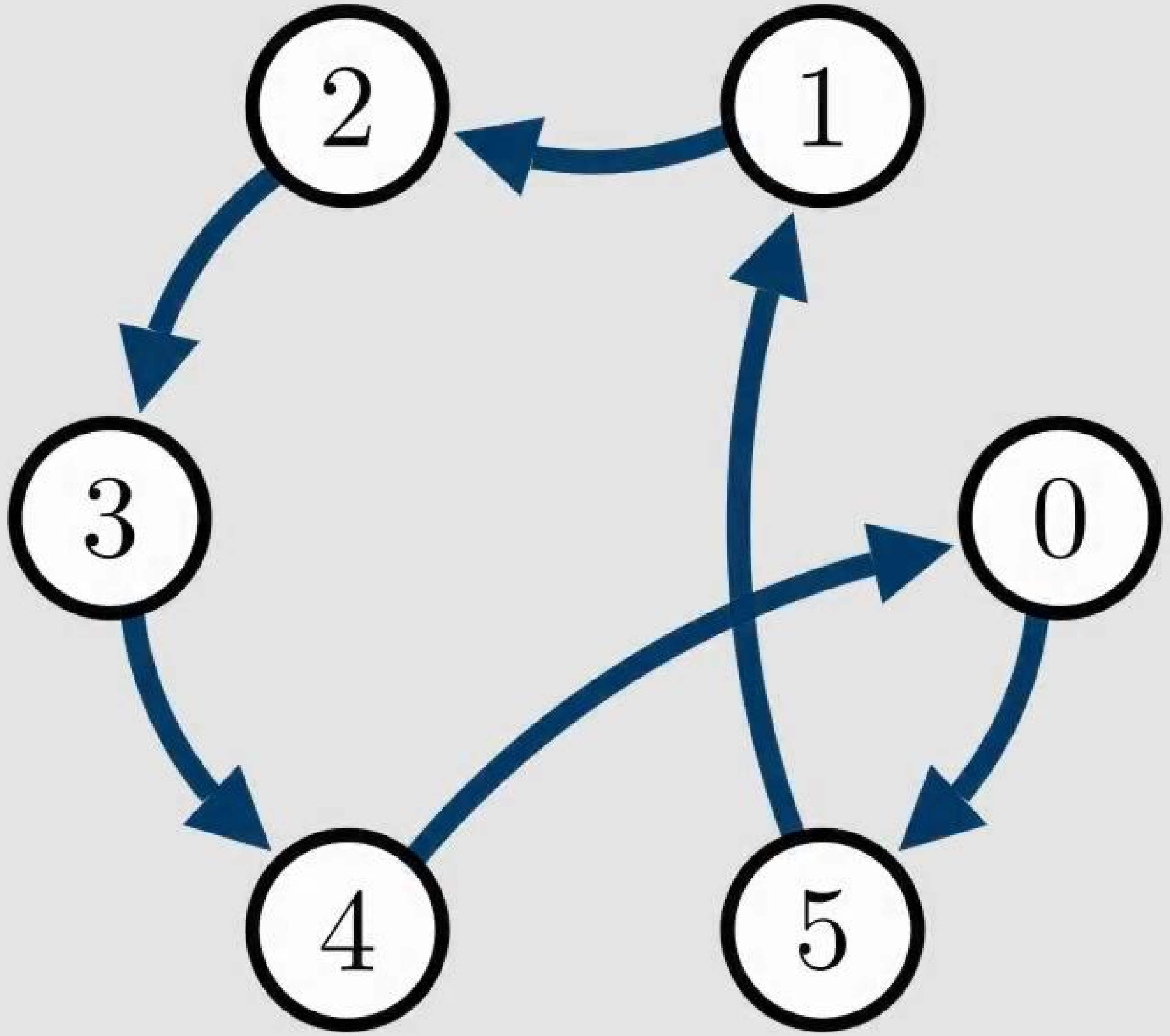


Fig. 5 **a** The SCC exploration graph for *circuit* starting from root. At least one (*thick*) edge from A to the root, from D to C, C to B, and B to A must exist (rule 1). Backwards (*dotted*) edges to the root from B, C or D cannot be used (rule 1). The (*thin-dashed*) edges from C to A and D to B cannot be used (rule 2). The (*thick-dashed*) edges leading from root to A, B and C cannot be used (rule 3). **b** Illustration of *prune-within* (rule 4). The edge from x to a cannot be used otherwise we cannot escape the subtree rooted at a (*dark grey*). We need to enter the subtree from elsewhere

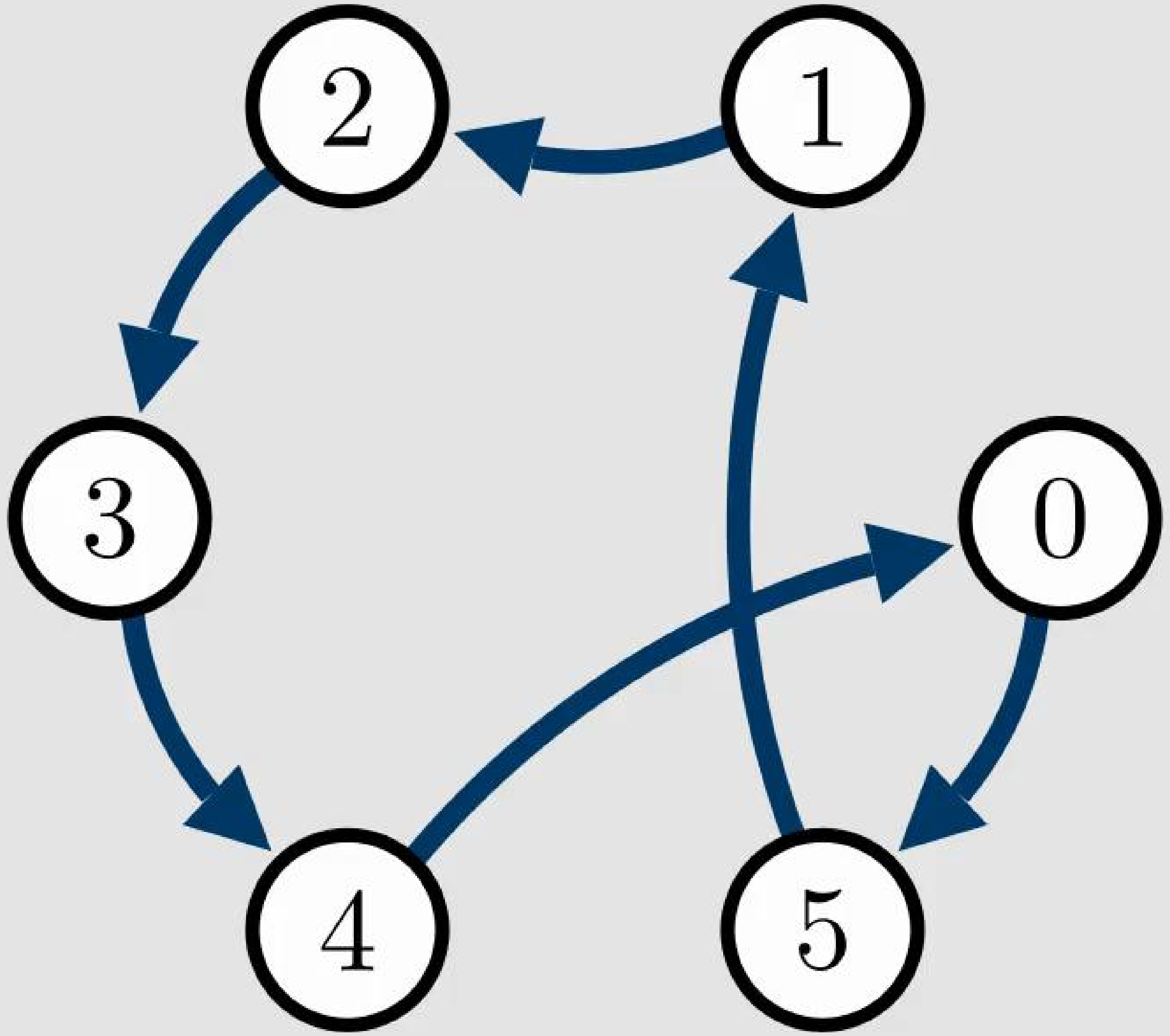
Francis, K.G. and Stuckey, P.J., 2014.
Explaining circuit propagation. Constraints, 19, pp.1-29.

Circuit PB Encoding



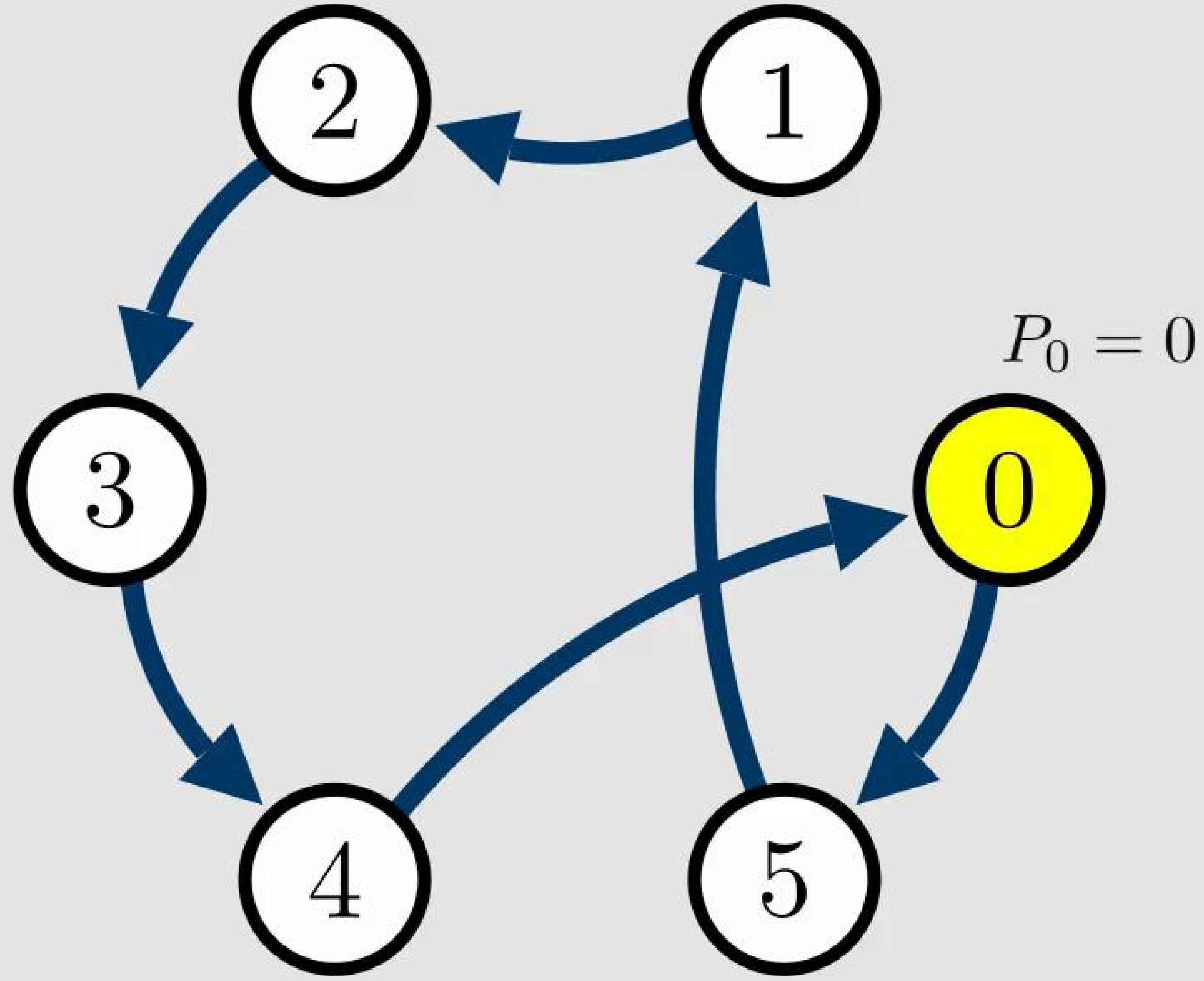
Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0



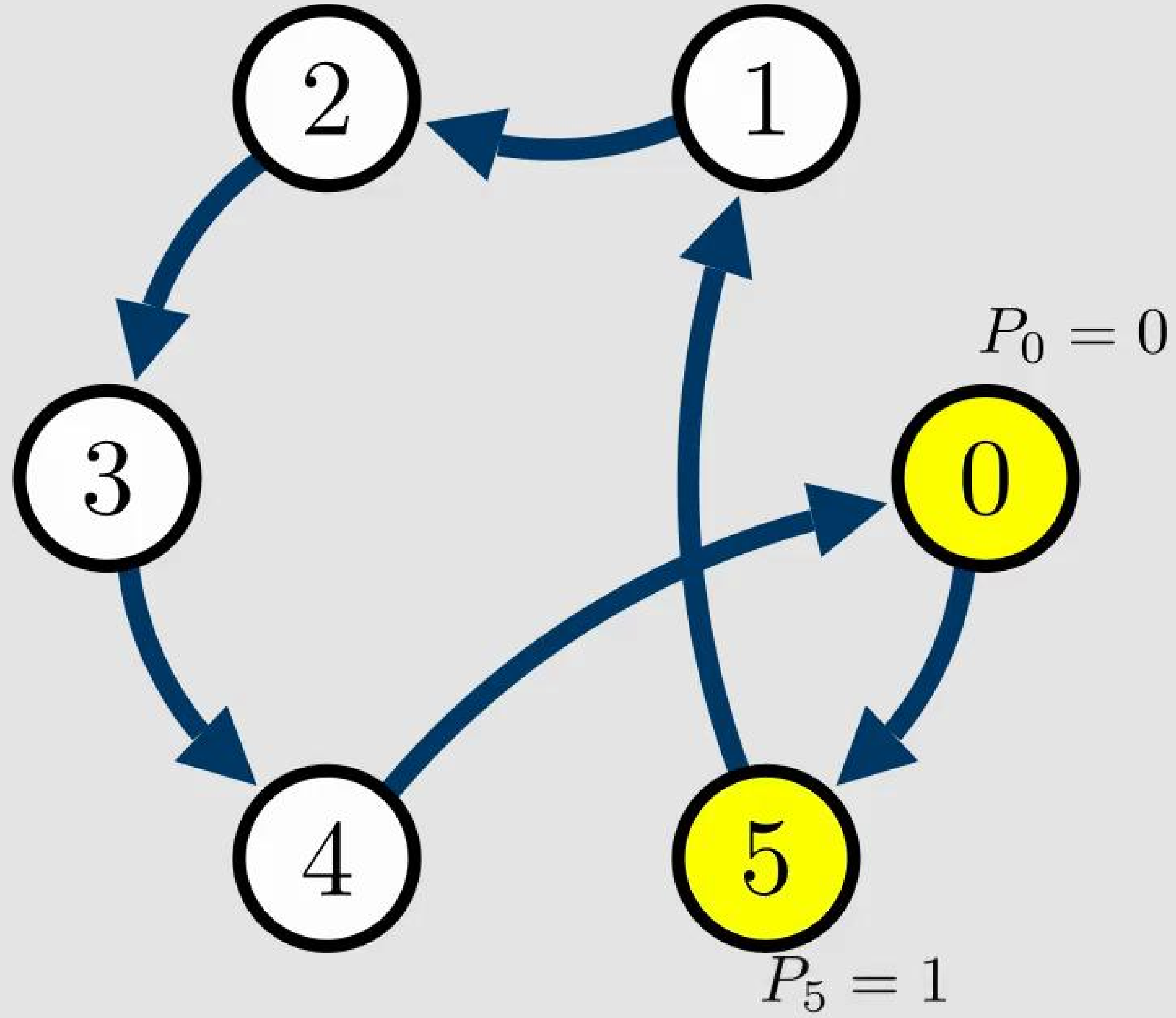
Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0



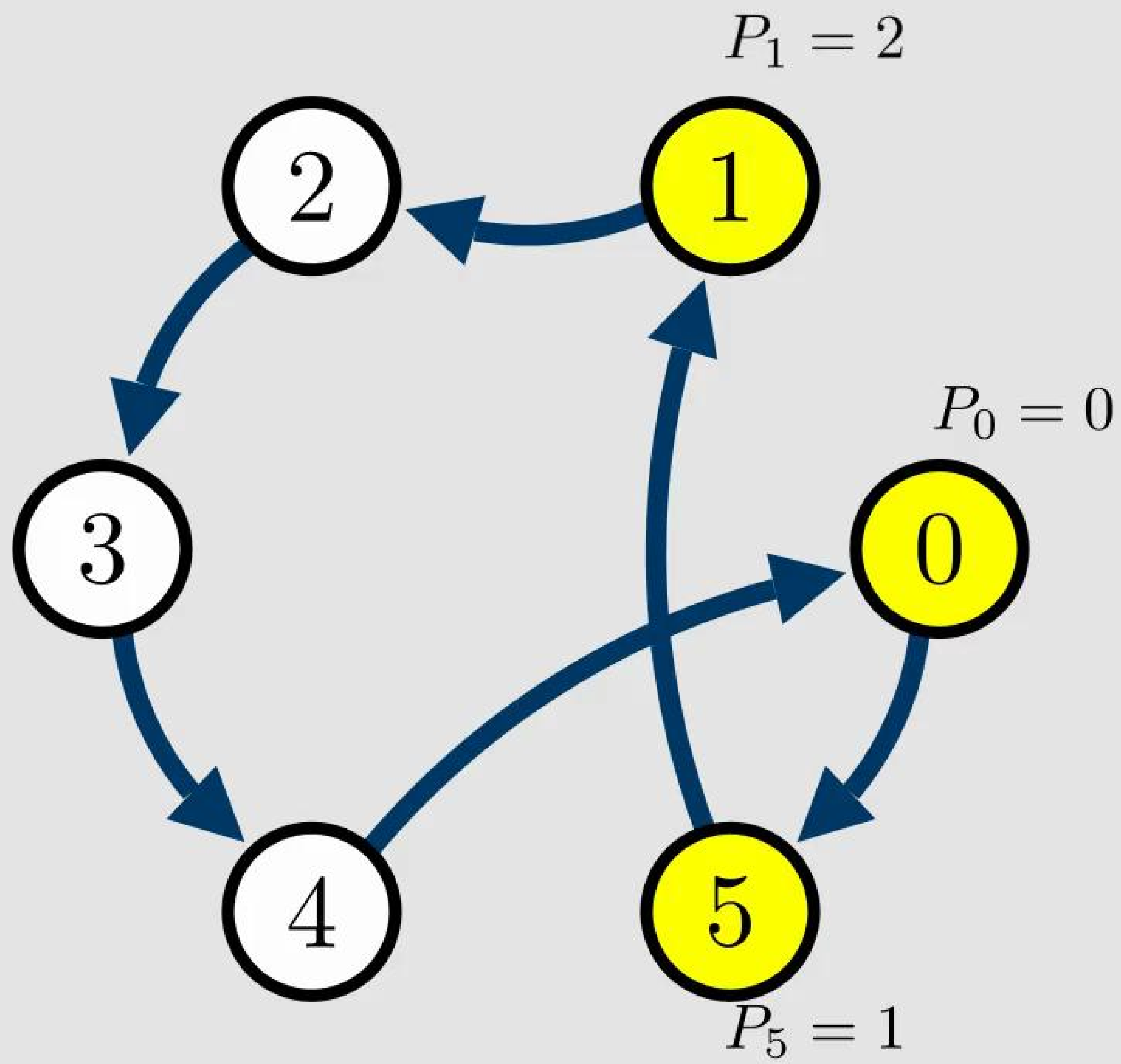
Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0



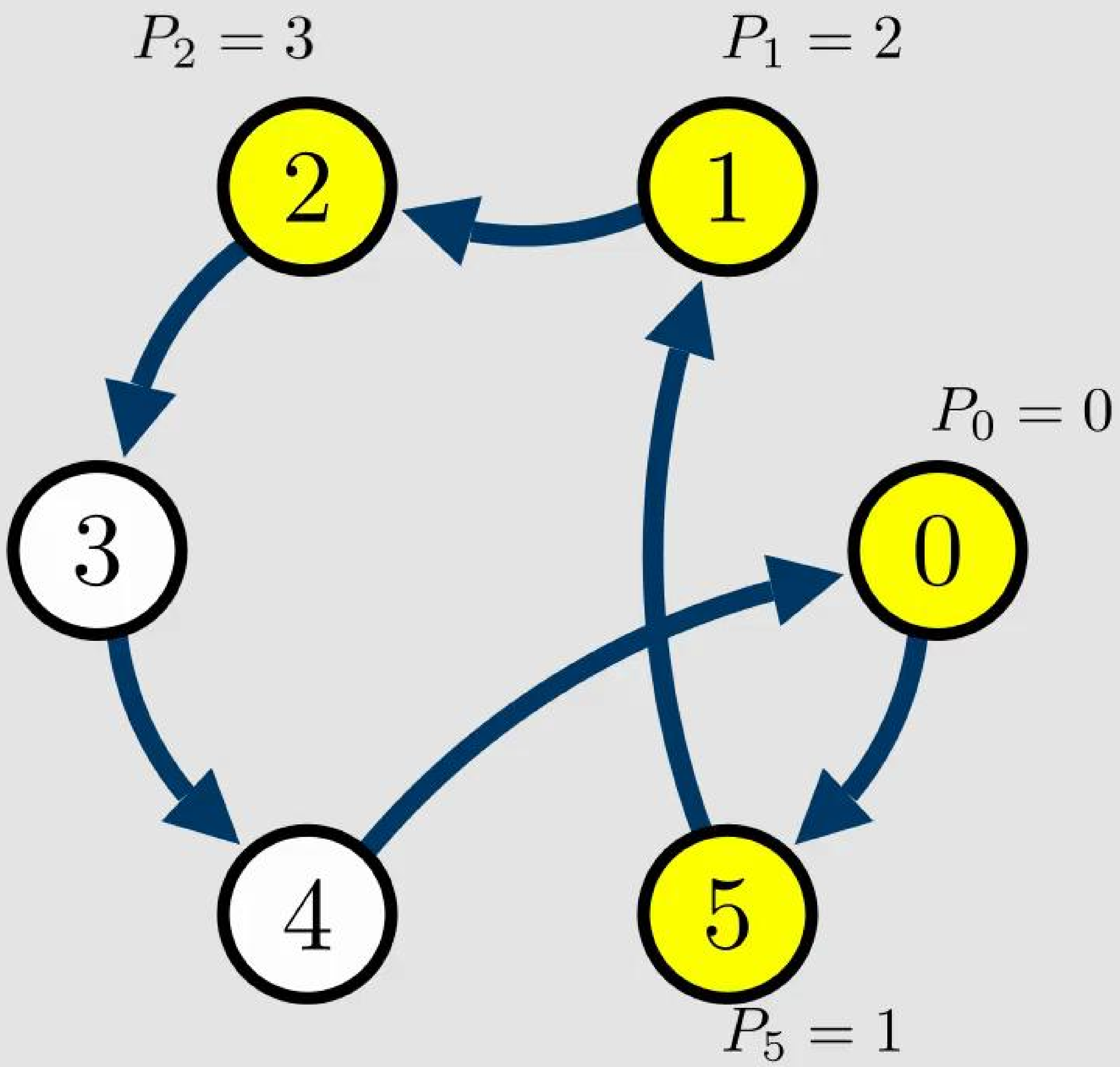
Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0



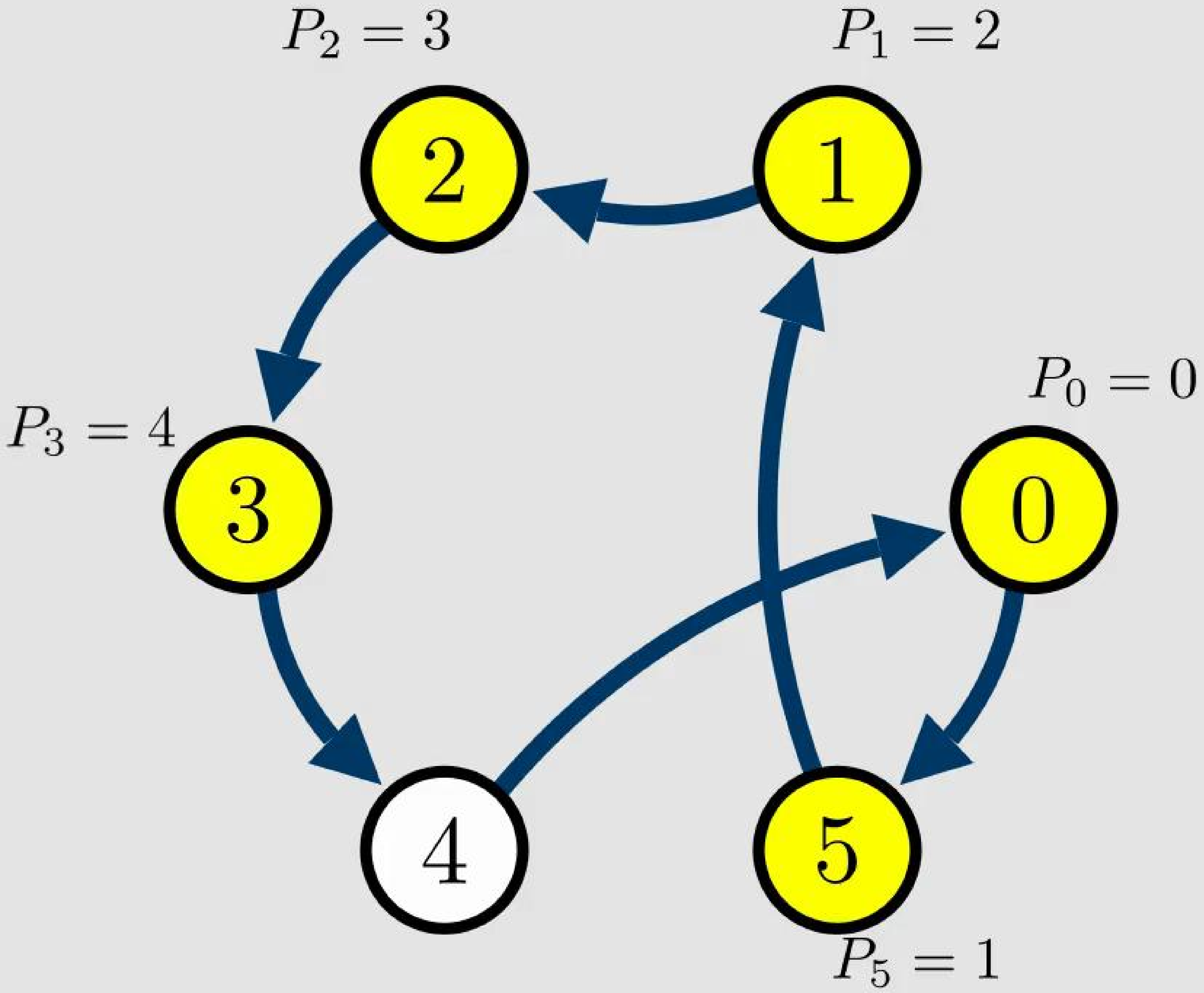
Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0



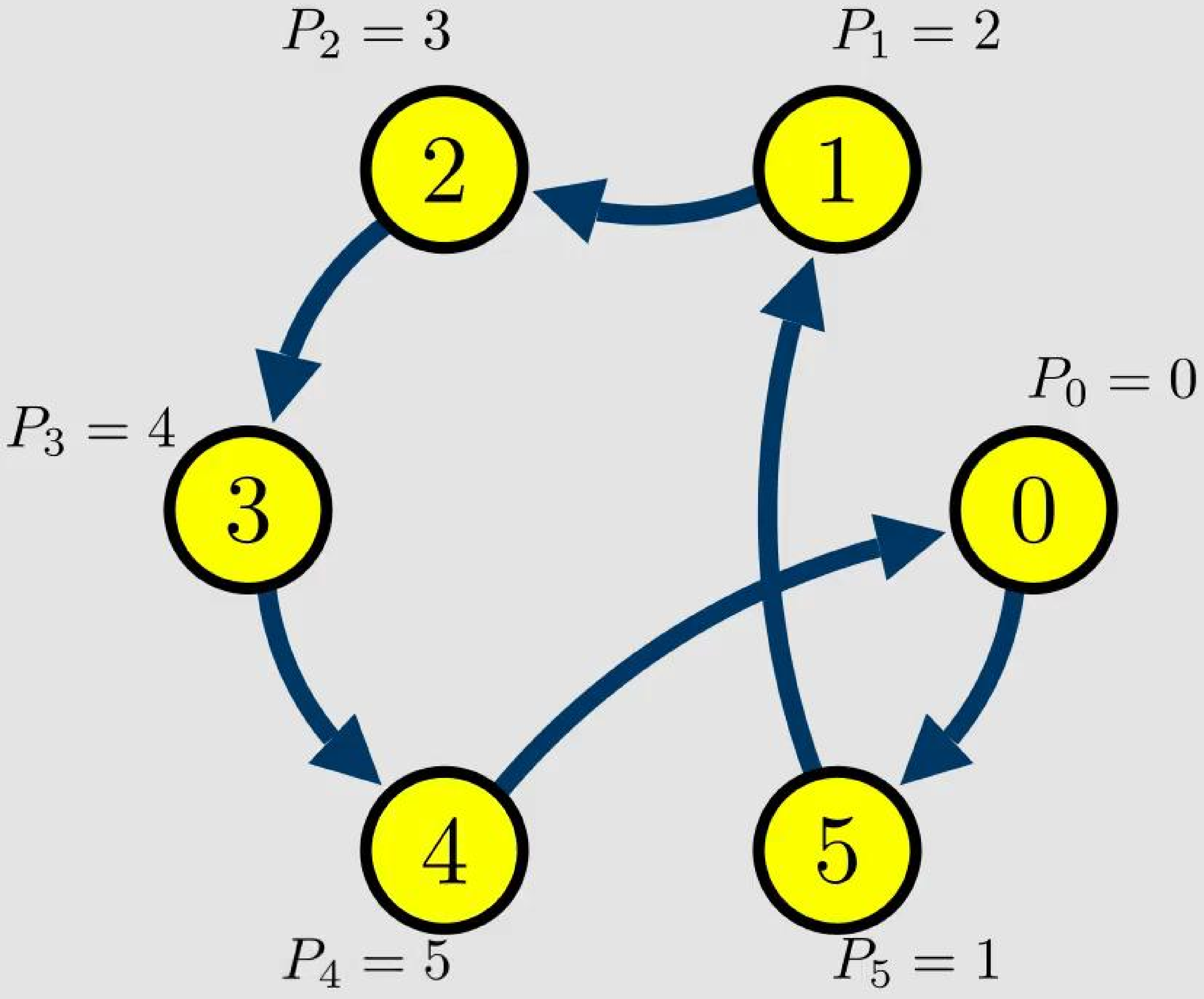
Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0



Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

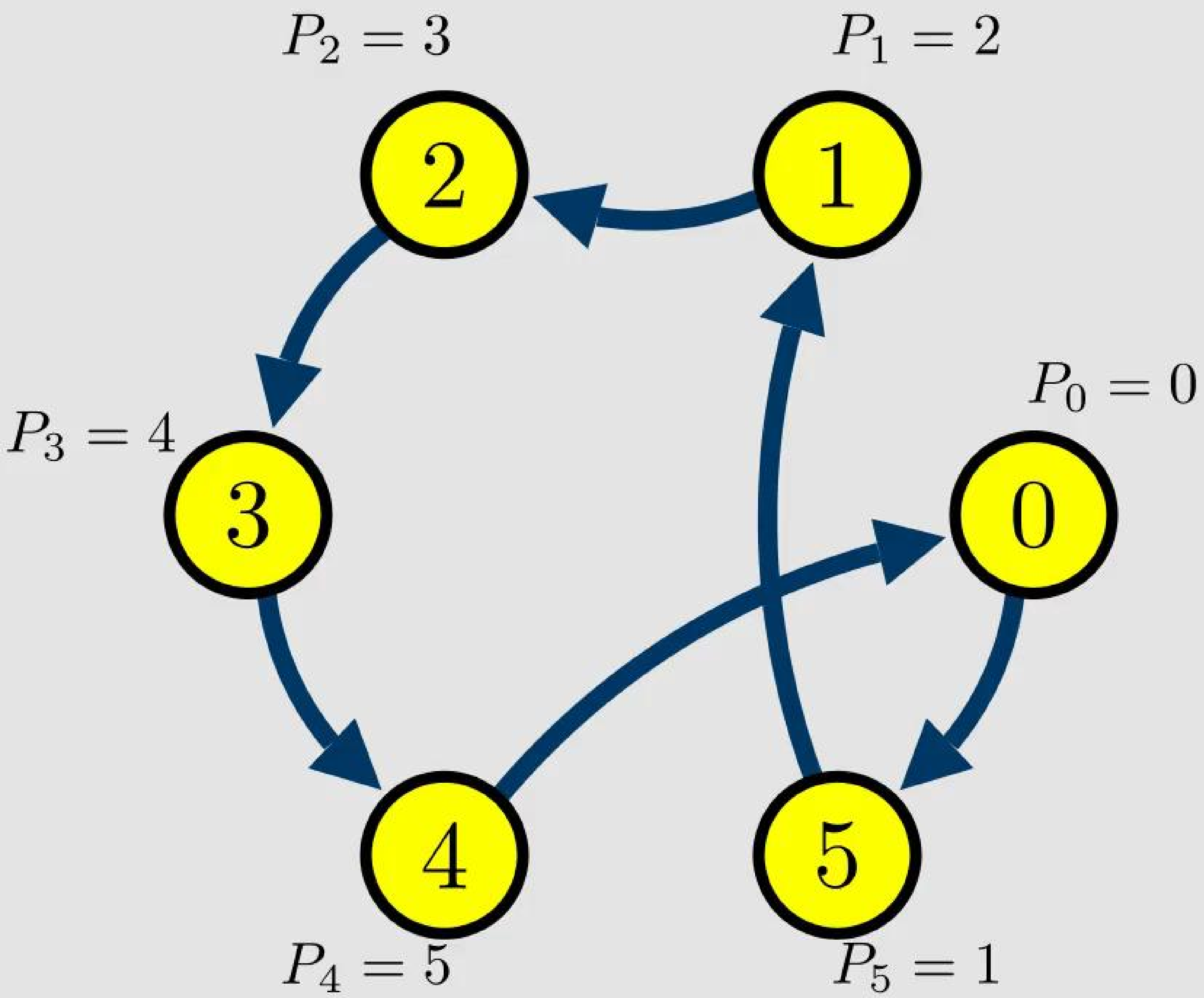


Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$

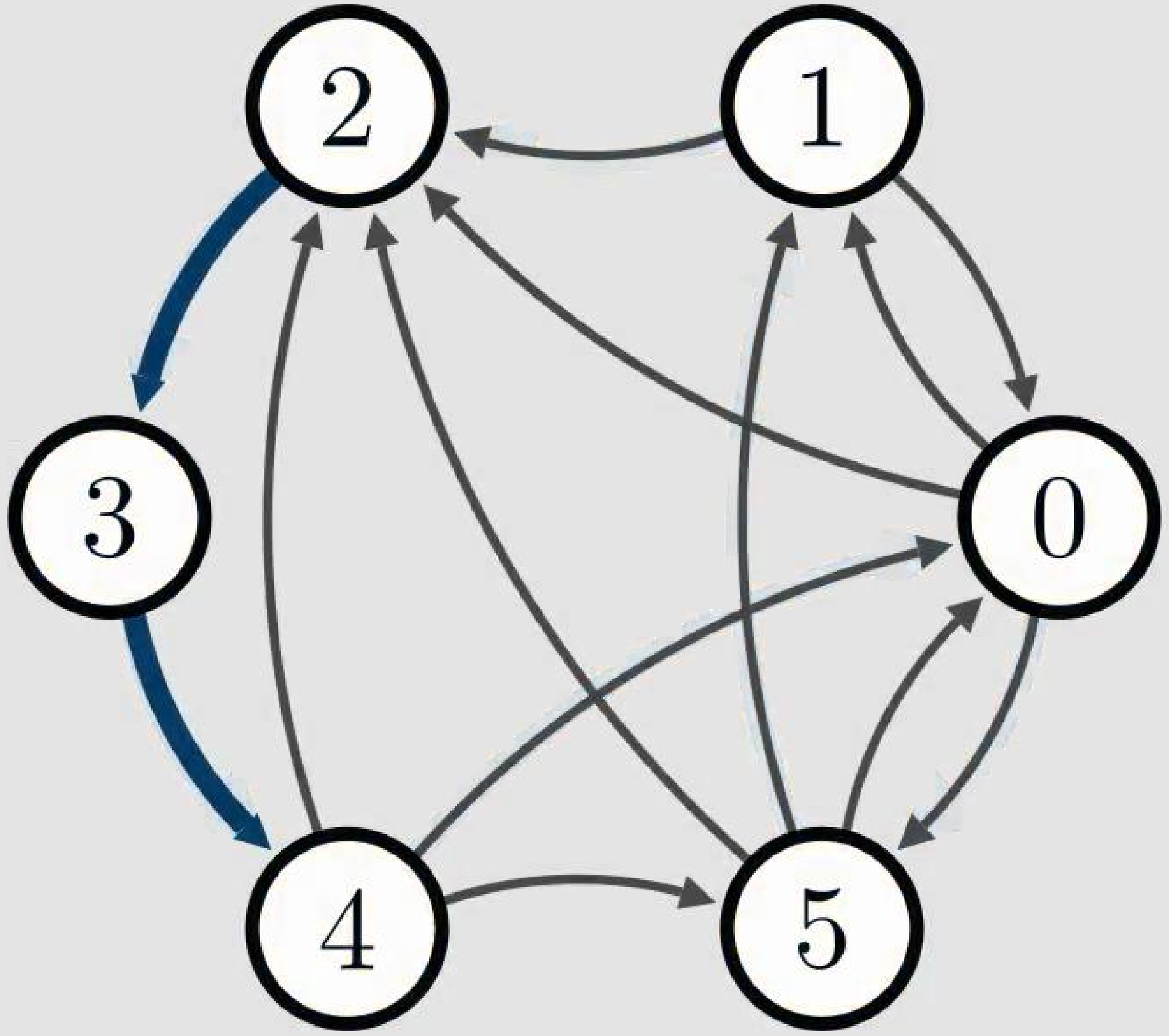


Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$

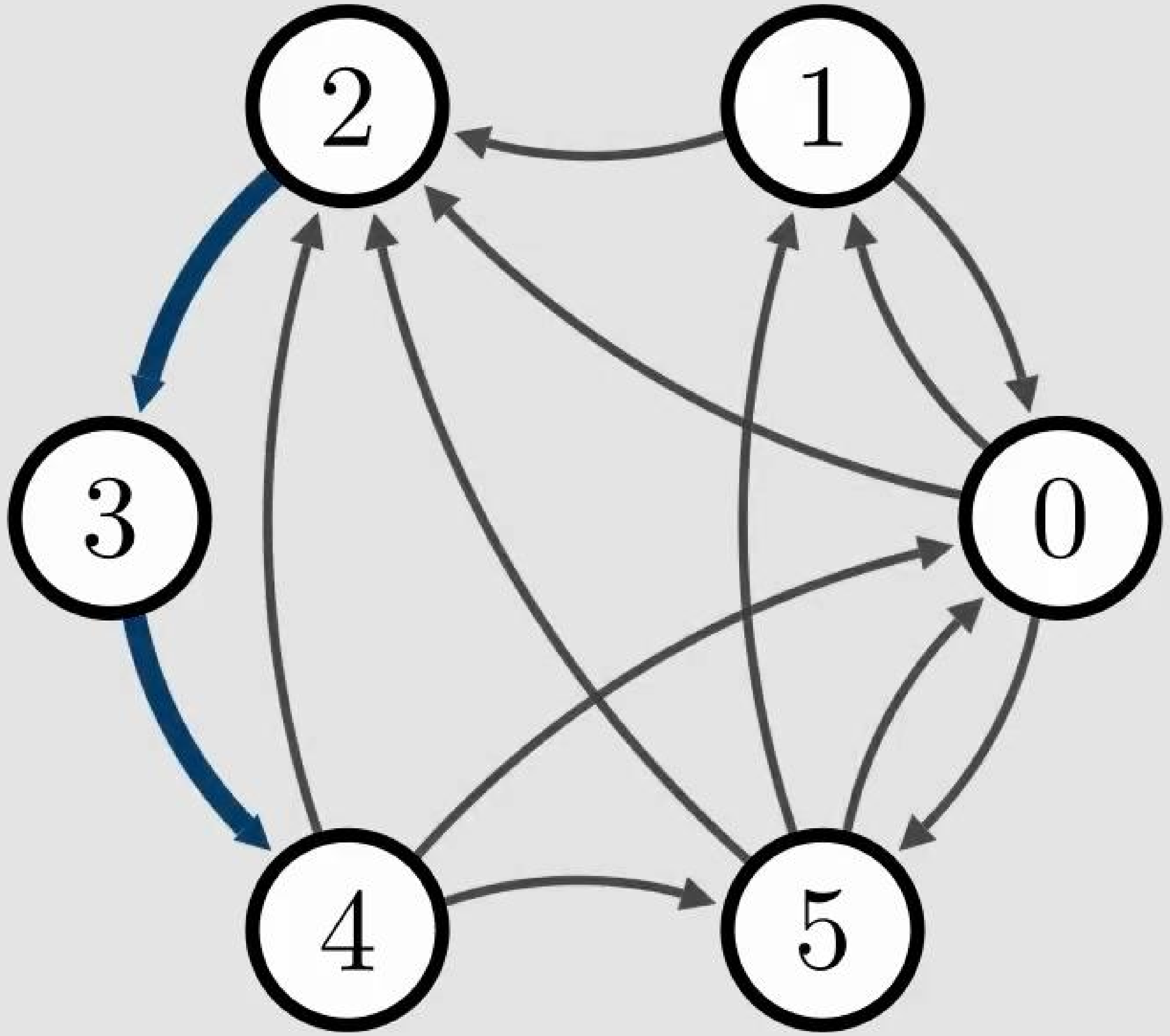


Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$

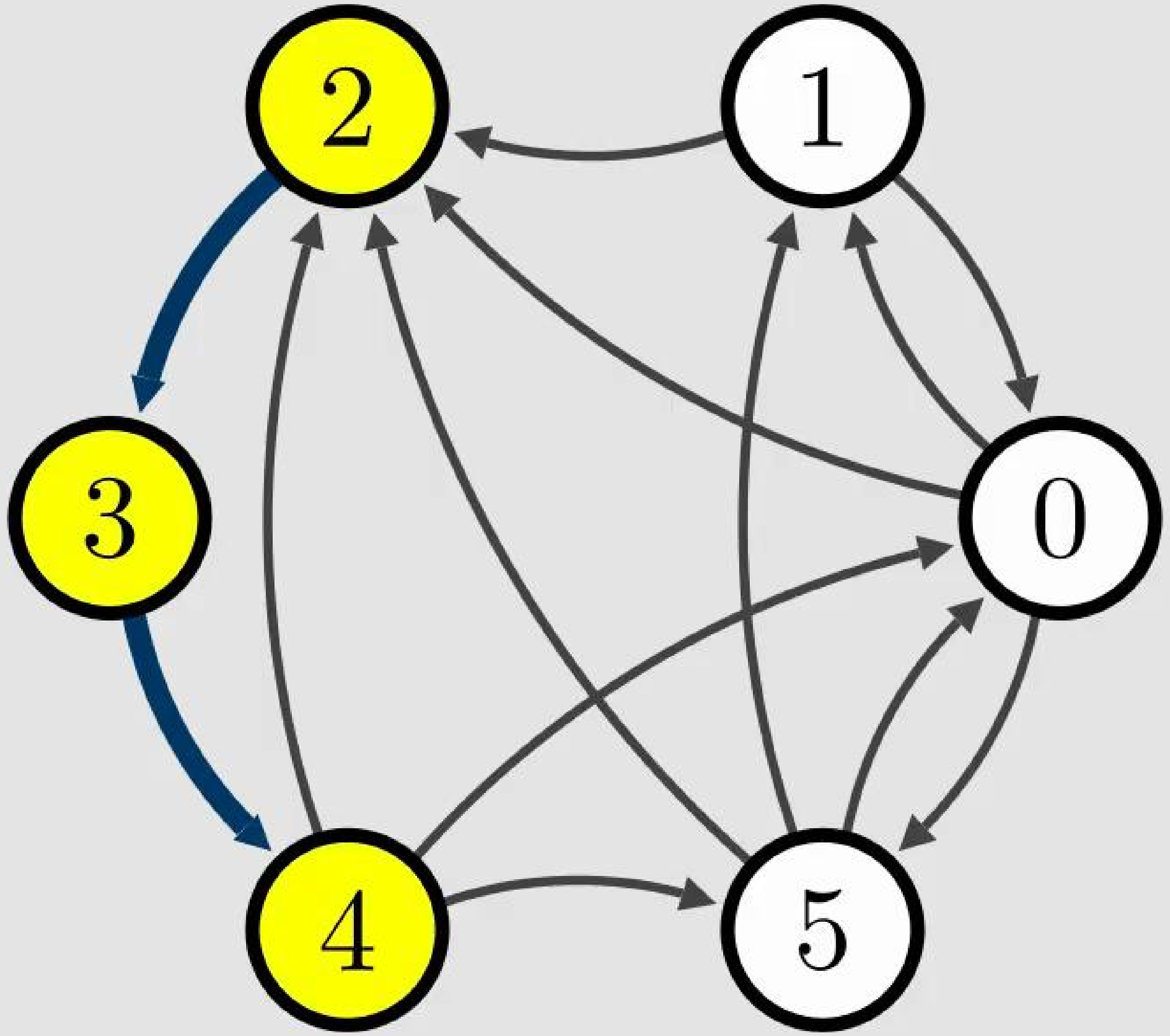


Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$

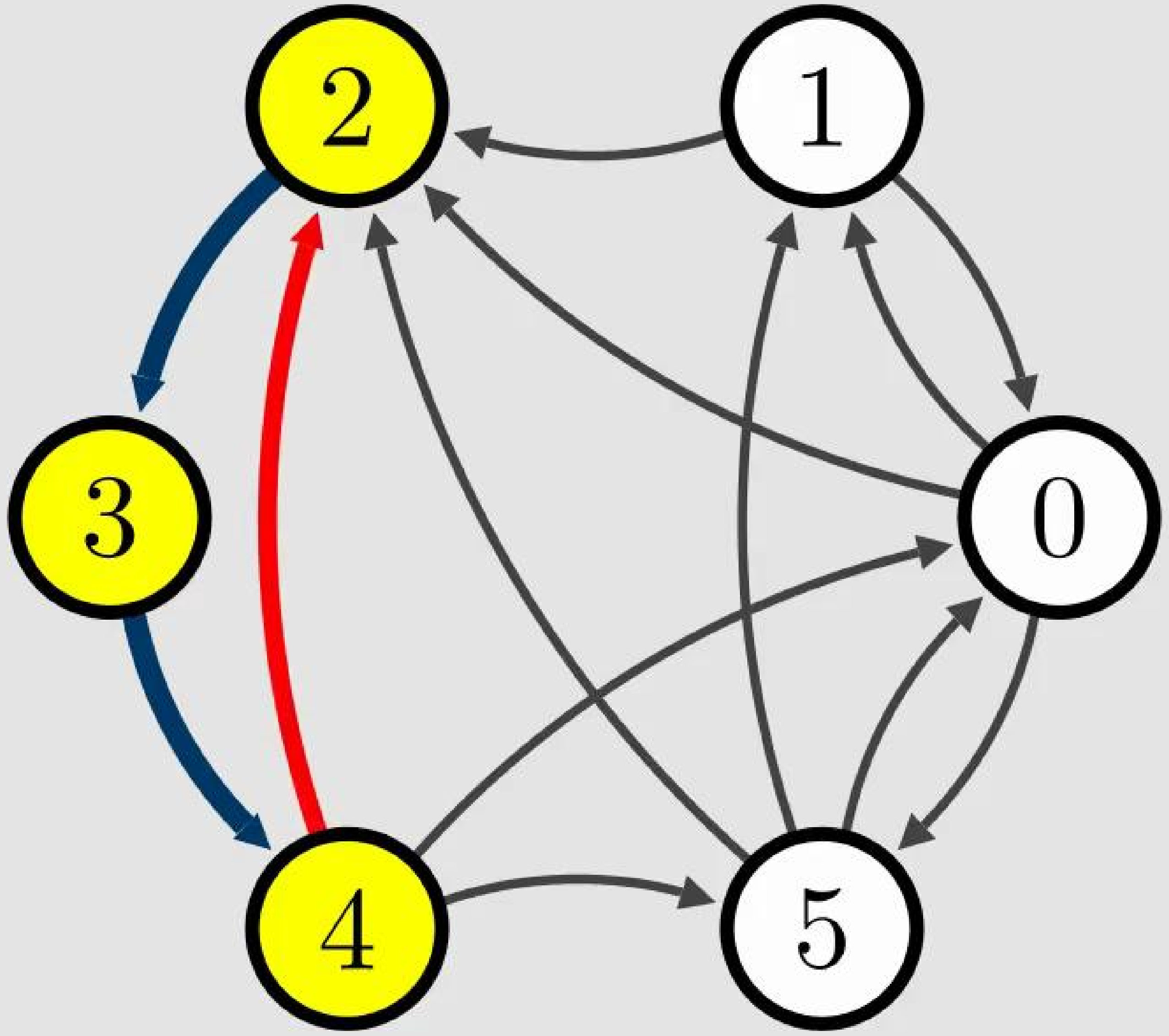


Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$

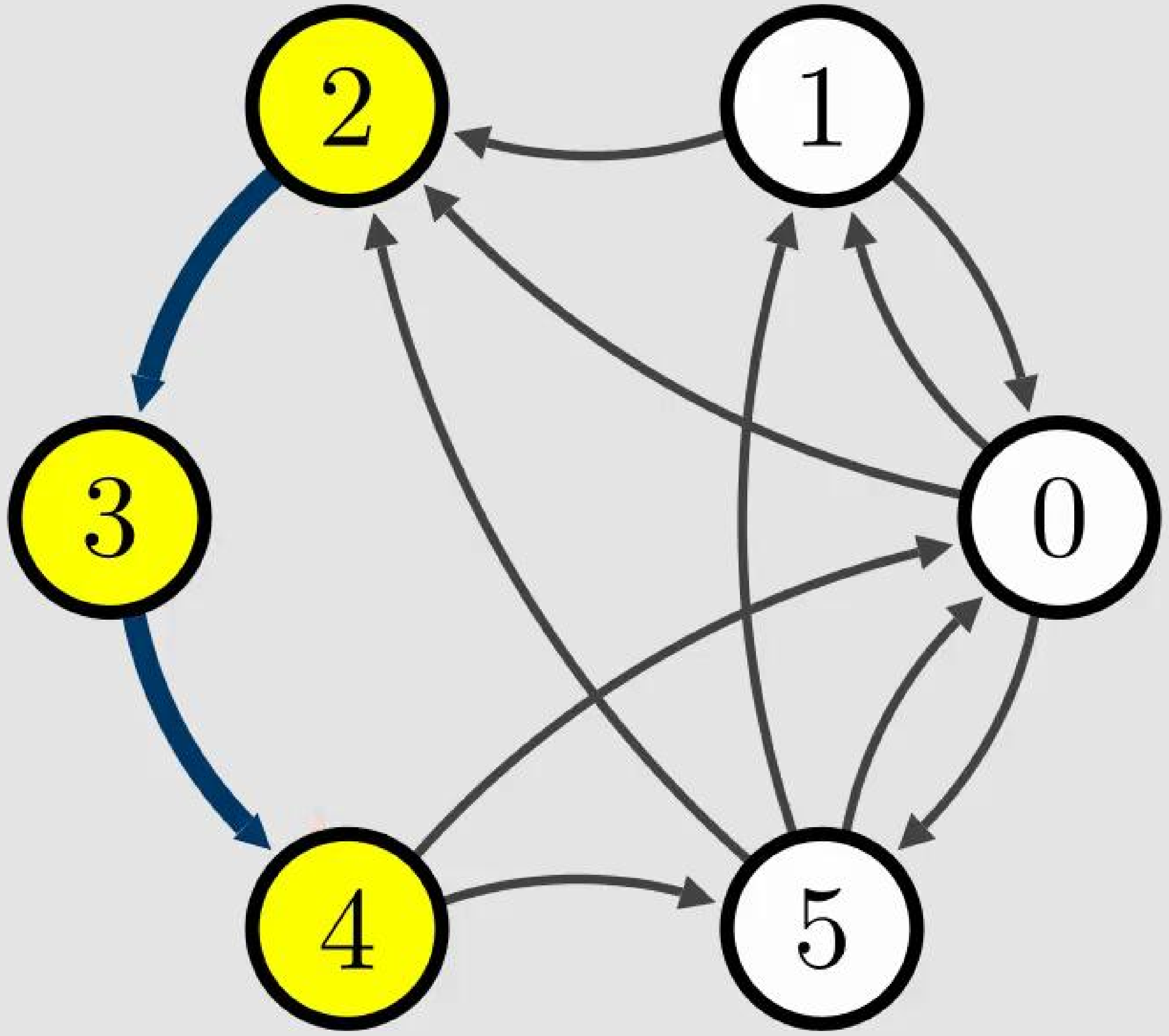


Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$



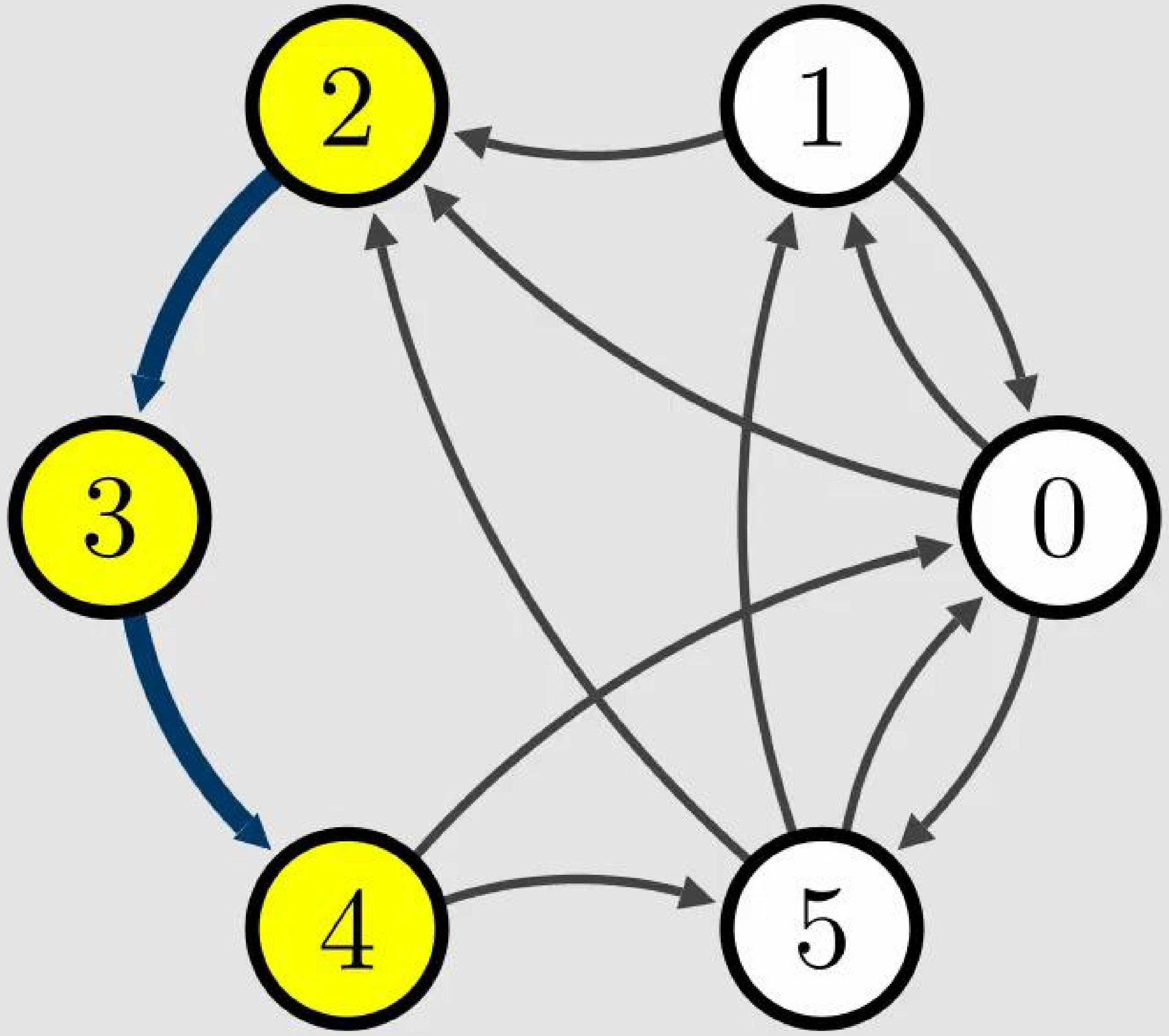
Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$

From encoding:



Circuit PB Encoding

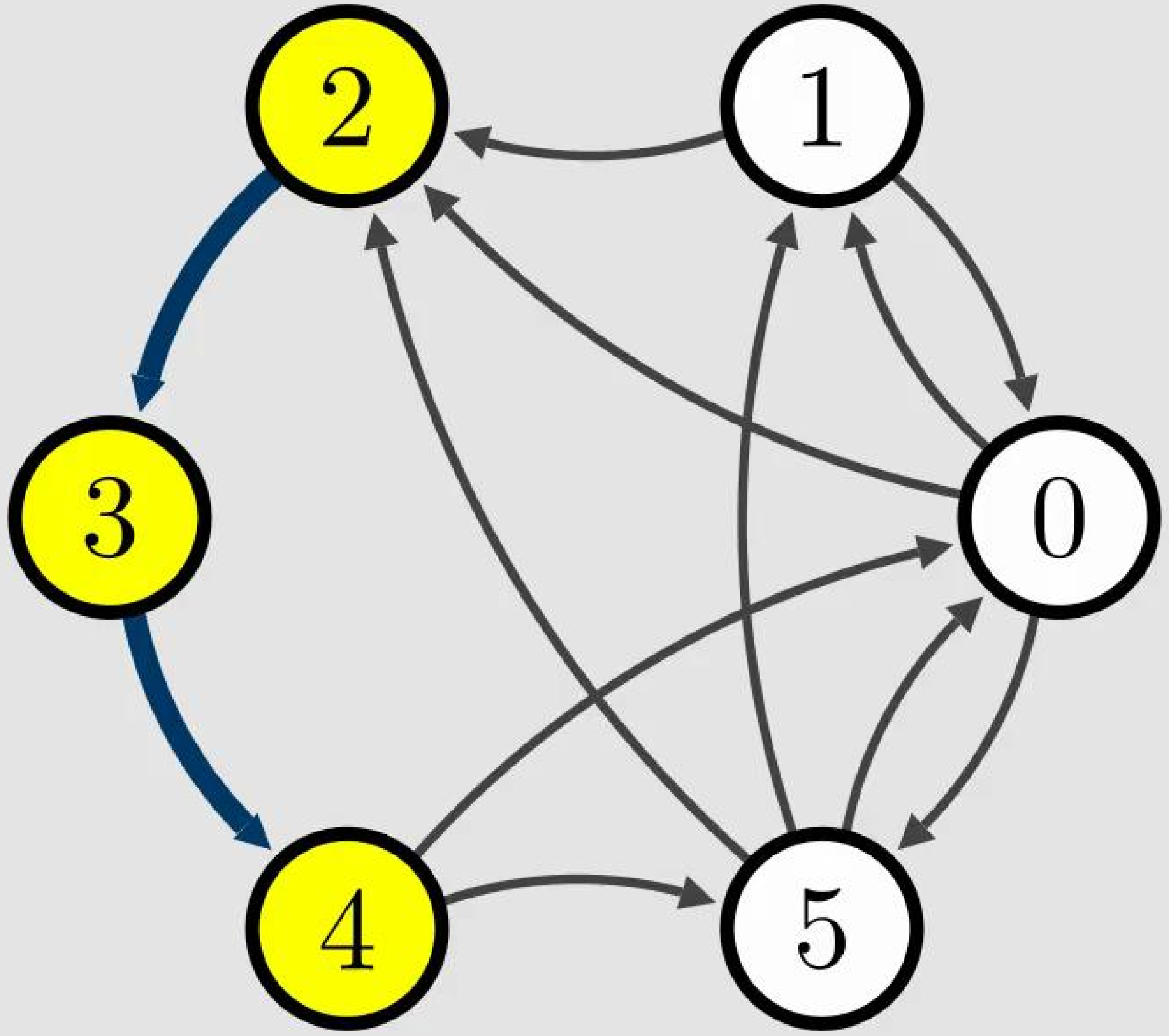
$P_i :=$ Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$

From encoding:

$x_{2=3} \implies P_3 = P_2 + 1$



Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

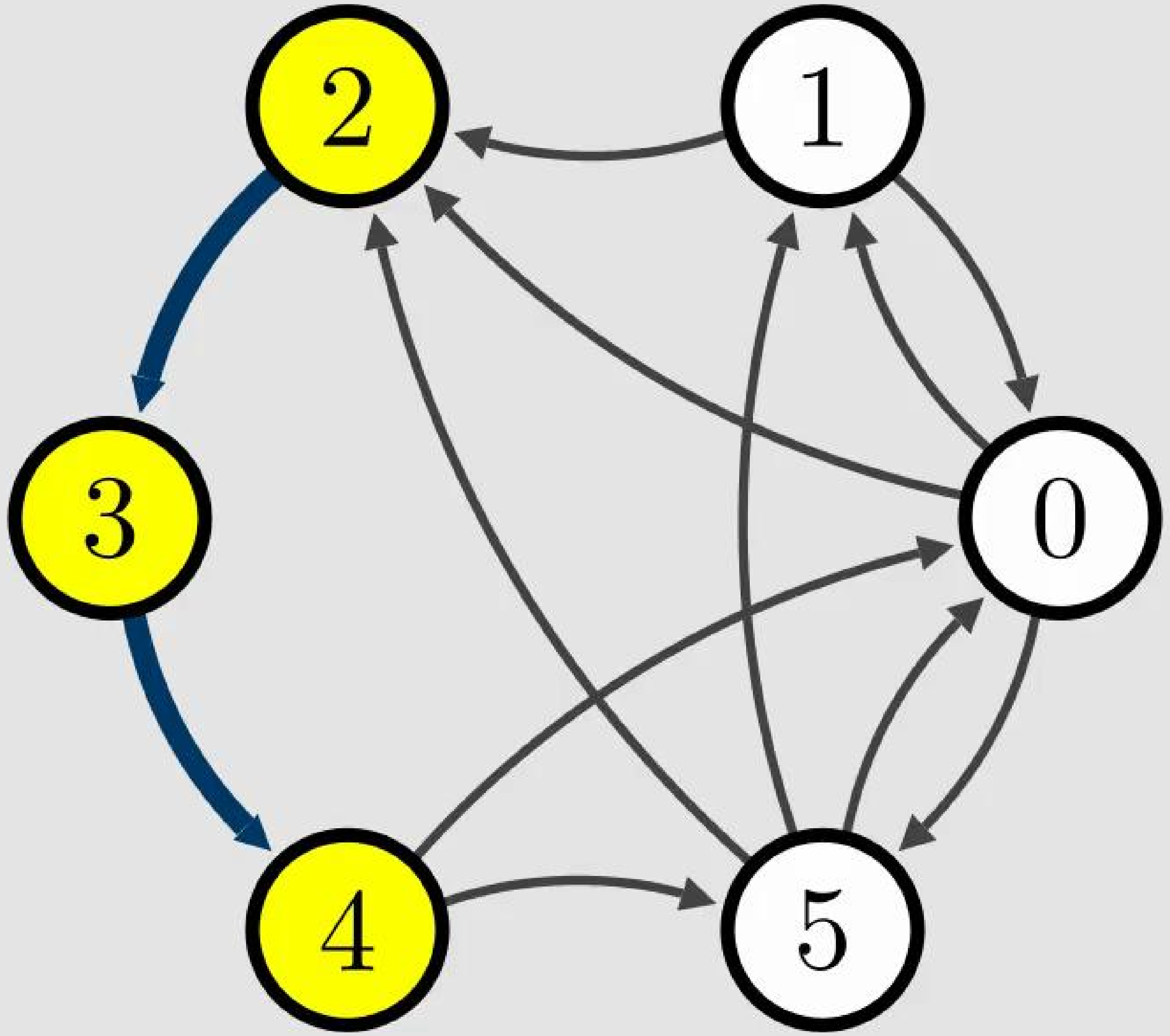
For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$

From encoding:

$x_{2=3} \implies P_3 = P_2 + 1$

$x_{3=4} \implies P_4 = P_3 + 1$



Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

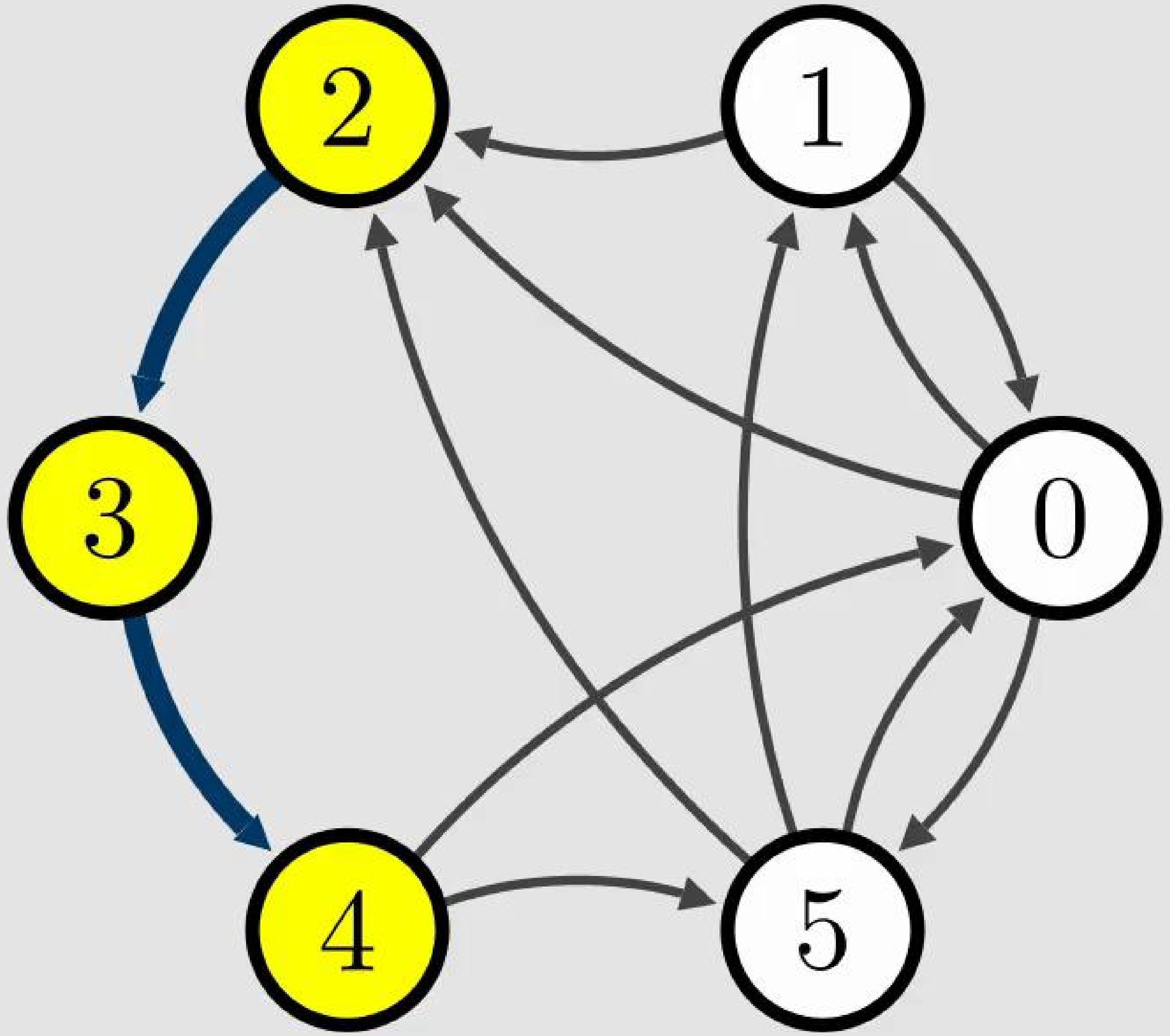
$x_{i=j} \implies P_j = P_i + 1$

From encoding:

$x_{2=3} \implies P_3 = P_2 + 1$

$x_{3=4} \implies P_4 = P_3 + 1$

$x_{4=2} \implies P_2 = P_4 + 1$



Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$

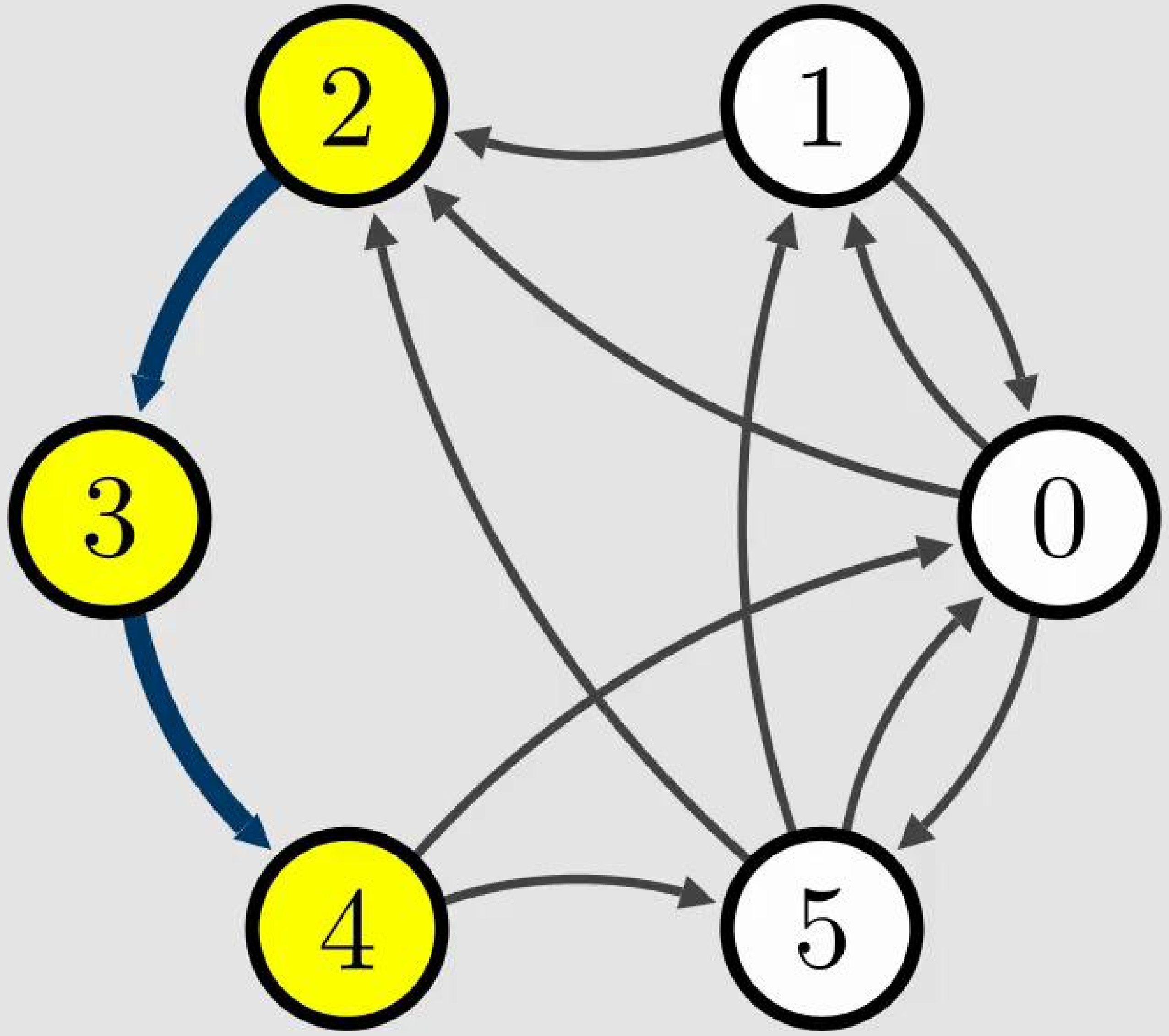
From encoding:

$x_{2=3} \implies P_3 = P_2 + 1$

$x_{3=4} \implies P_4 = P_3 + 1$

$x_{4=2} \implies P_2 = P_4 + 1$

Cutting planes addition:



Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$

From encoding:

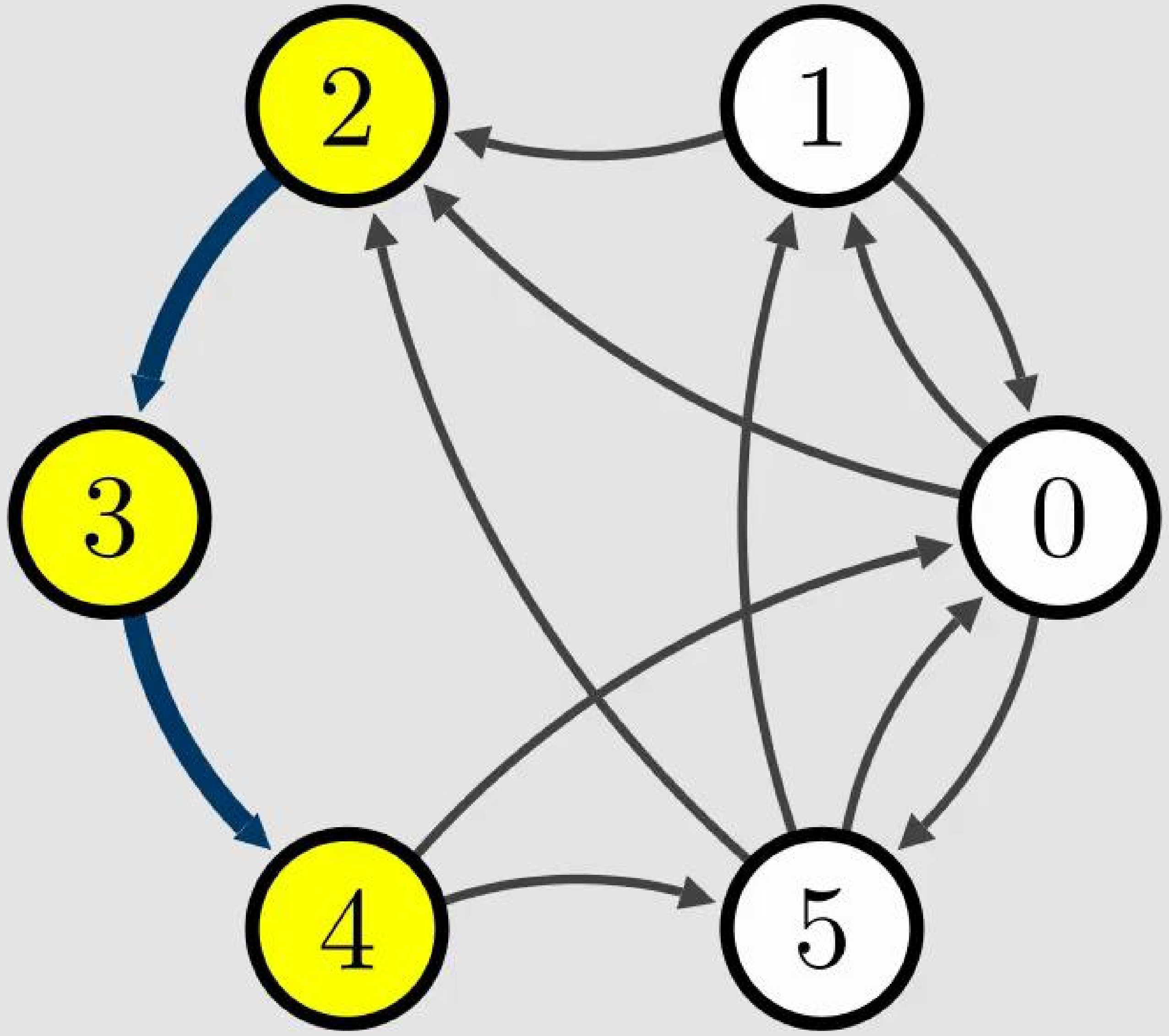
$x_{2=3} \implies P_3 = P_2 + 1$

$x_{3=4} \implies P_4 = P_3 + 1$

$x_{4=2} \implies P_2 = P_4 + 1$

Cutting planes addition:

$$x_{2=3} \wedge x_{3=4} \wedge x_{4=2} \implies P_3 - P_2 + P_4 - P_3 + P_2 - P_4 = 1 + 1 + 1$$



Circuit PB Encoding

P_i := Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$

From encoding:

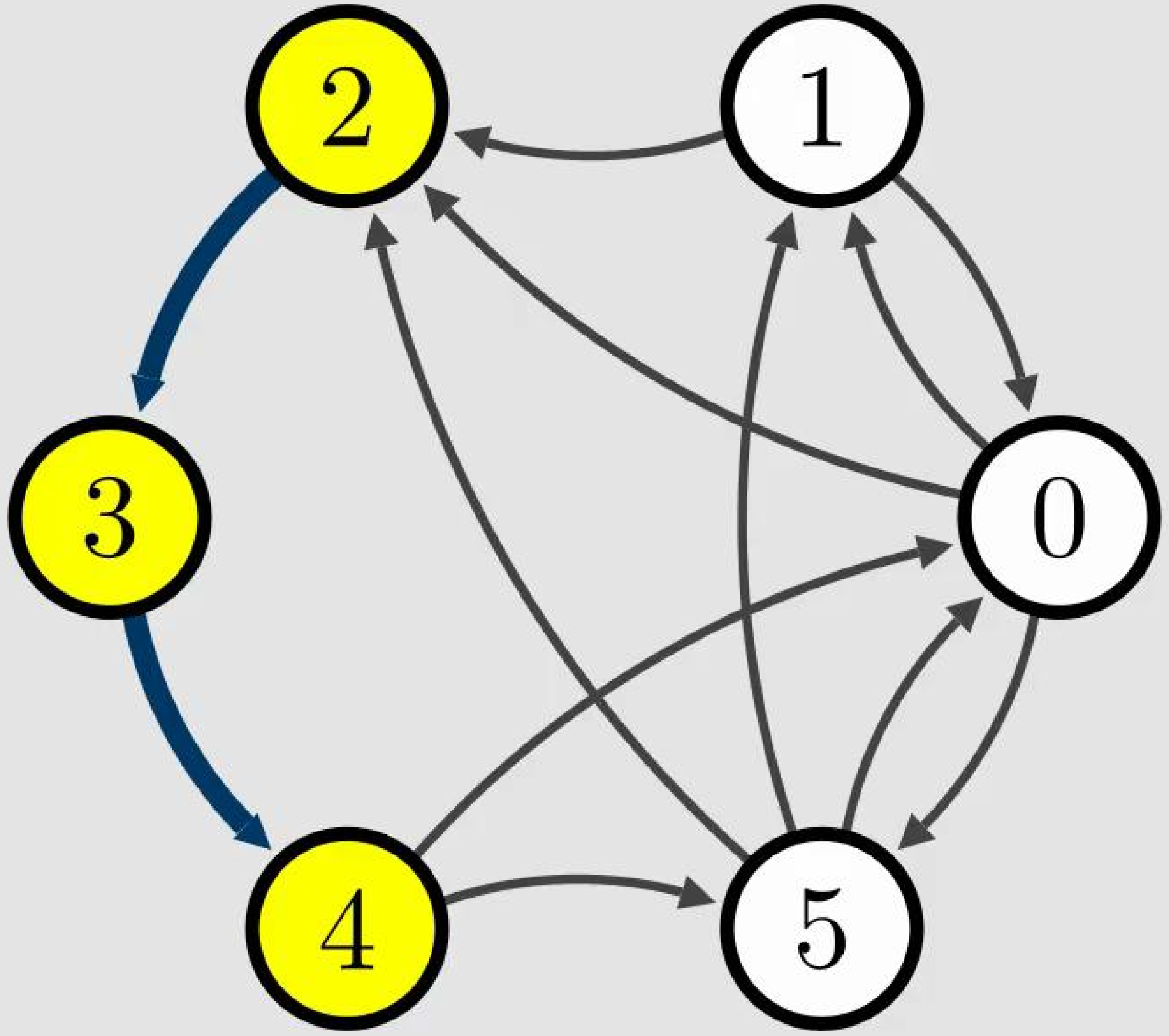
$x_{2=3} \implies P_3 = P_2 + 1$

$x_{3=4} \implies P_4 = P_3 + 1$

$x_{4=2} \implies P_2 = P_4 + 1$

Cutting planes addition:

$x_{2=3} \wedge x_{3=4} \wedge x_{4=2} \implies 0 = 3$



Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$

From encoding:

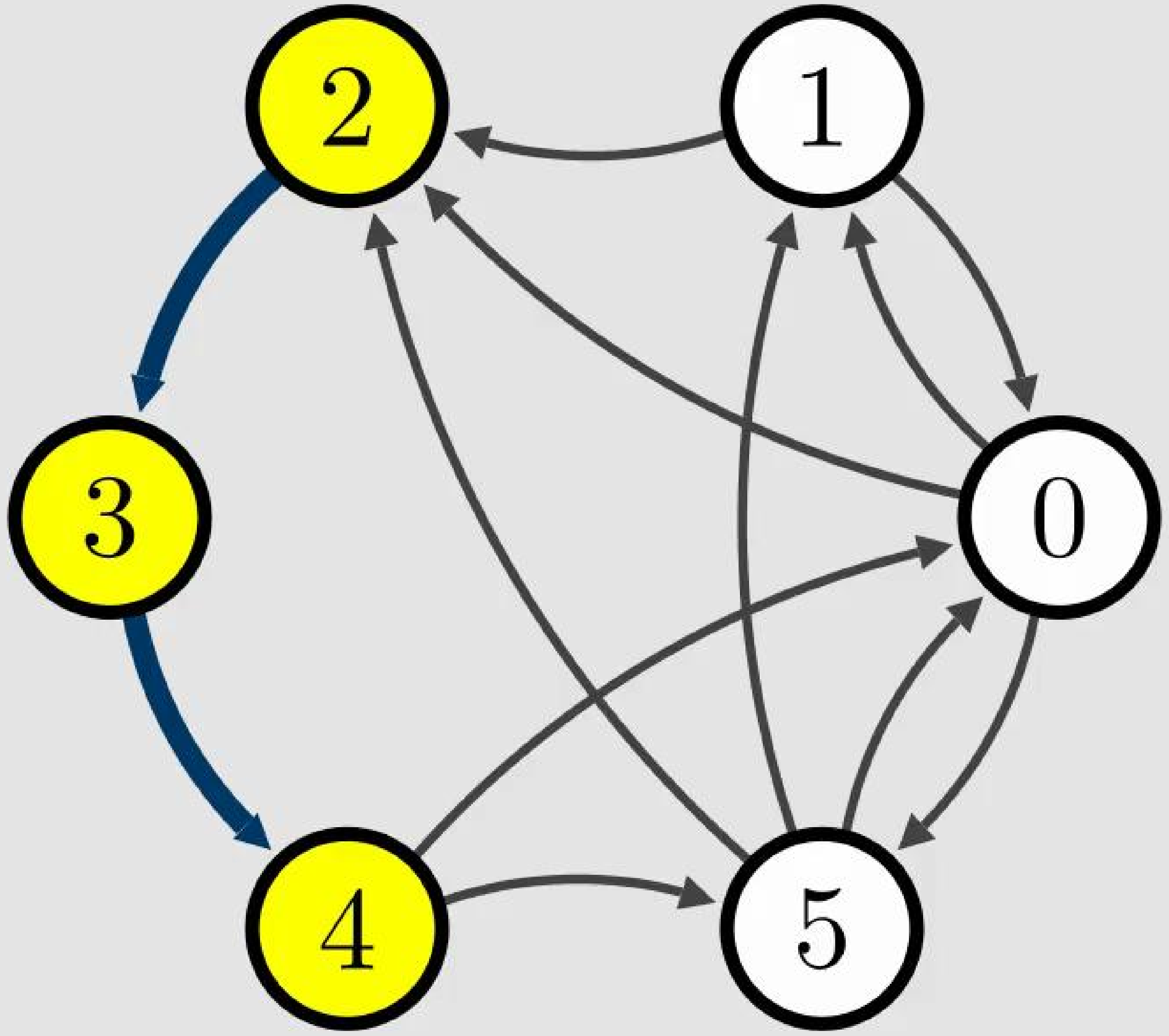
$x_{2=3} \implies P_3 = P_2 + 1$

$x_{3=4} \implies P_4 = P_3 + 1$

$x_{4=2} \implies P_2 = P_4 + 1$

Cutting planes addition:

$\overline{x_{2=3}} \vee \overline{x_{3=4}} \vee \overline{x_{4=2}}$



Circuit PB Encoding

$P_i :=$ Position of vertex i relative to 0

For each $X_i, j \in dom(X_i)\ j \neq 0$:

$x_{i=j} \implies P_j = P_i + 1$

From encoding:

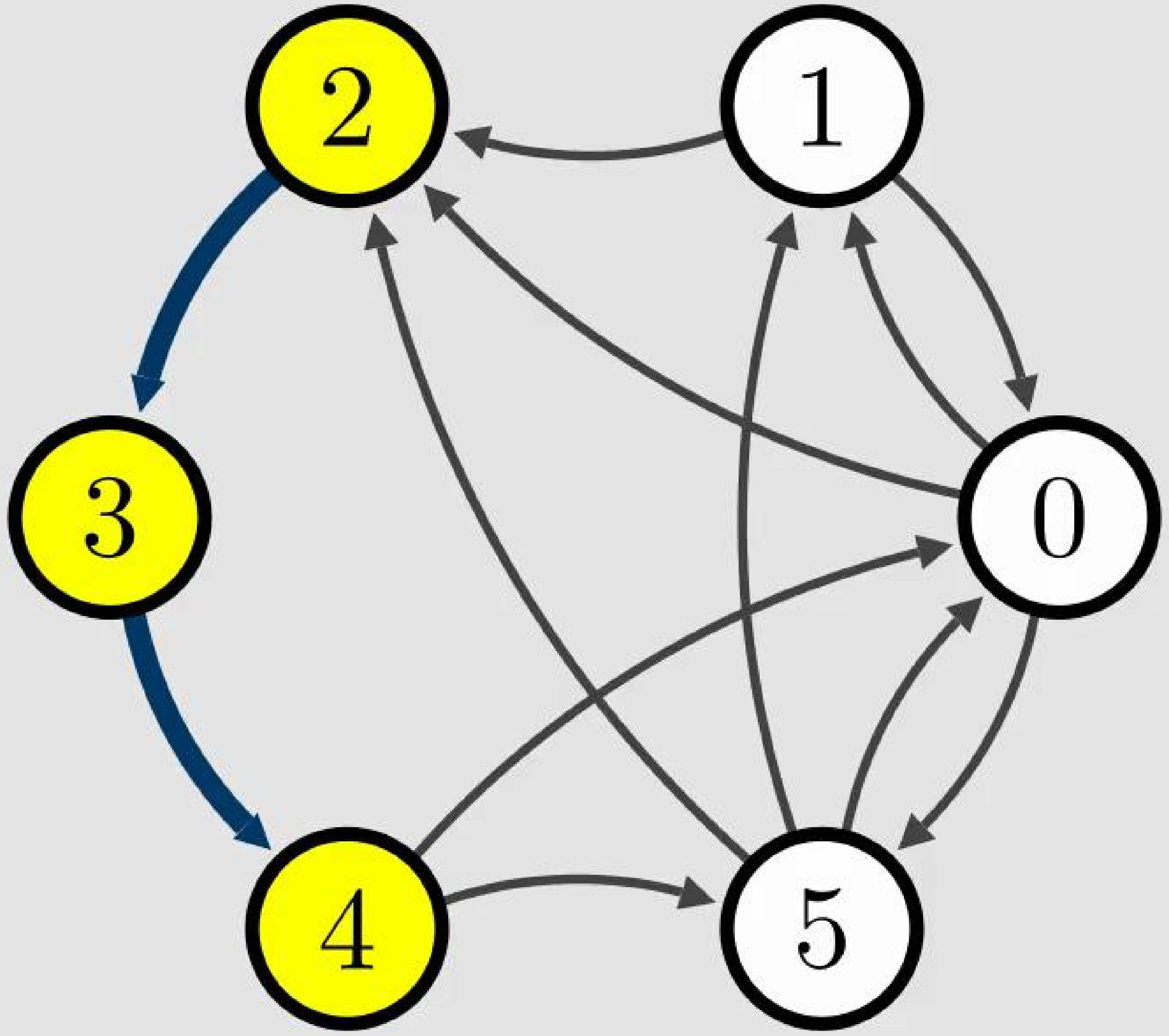
$x_{2=3} \implies P_3 = P_2 + 1$

$x_{3=4} \implies P_4 = P_3 + 1$

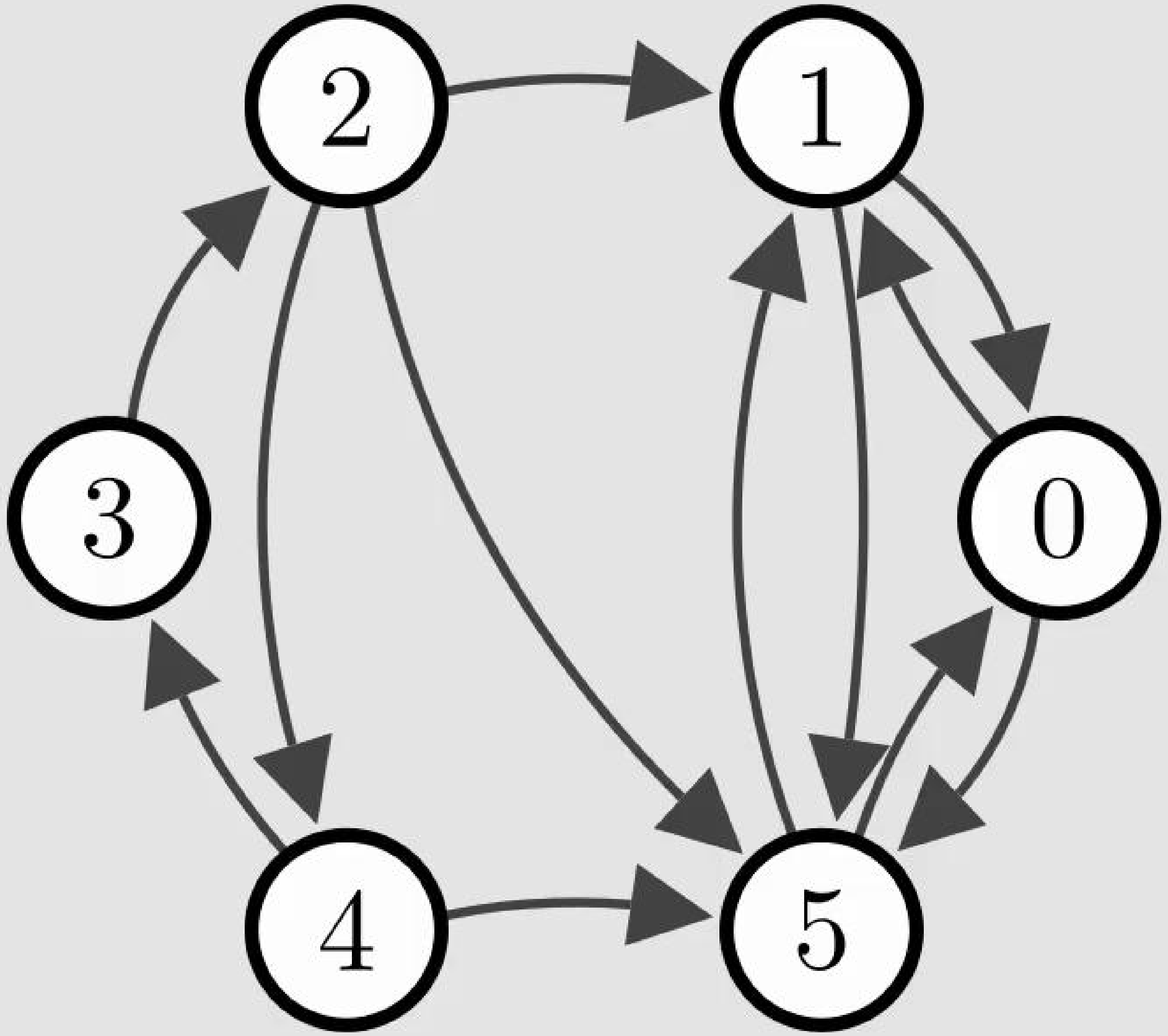
$x_{4=2} \implies P_2 = P_4 + 1$

Cutting planes addition:

$x_{2=3} \wedge x_{3=4} \implies \overline{x_{4=2}}$

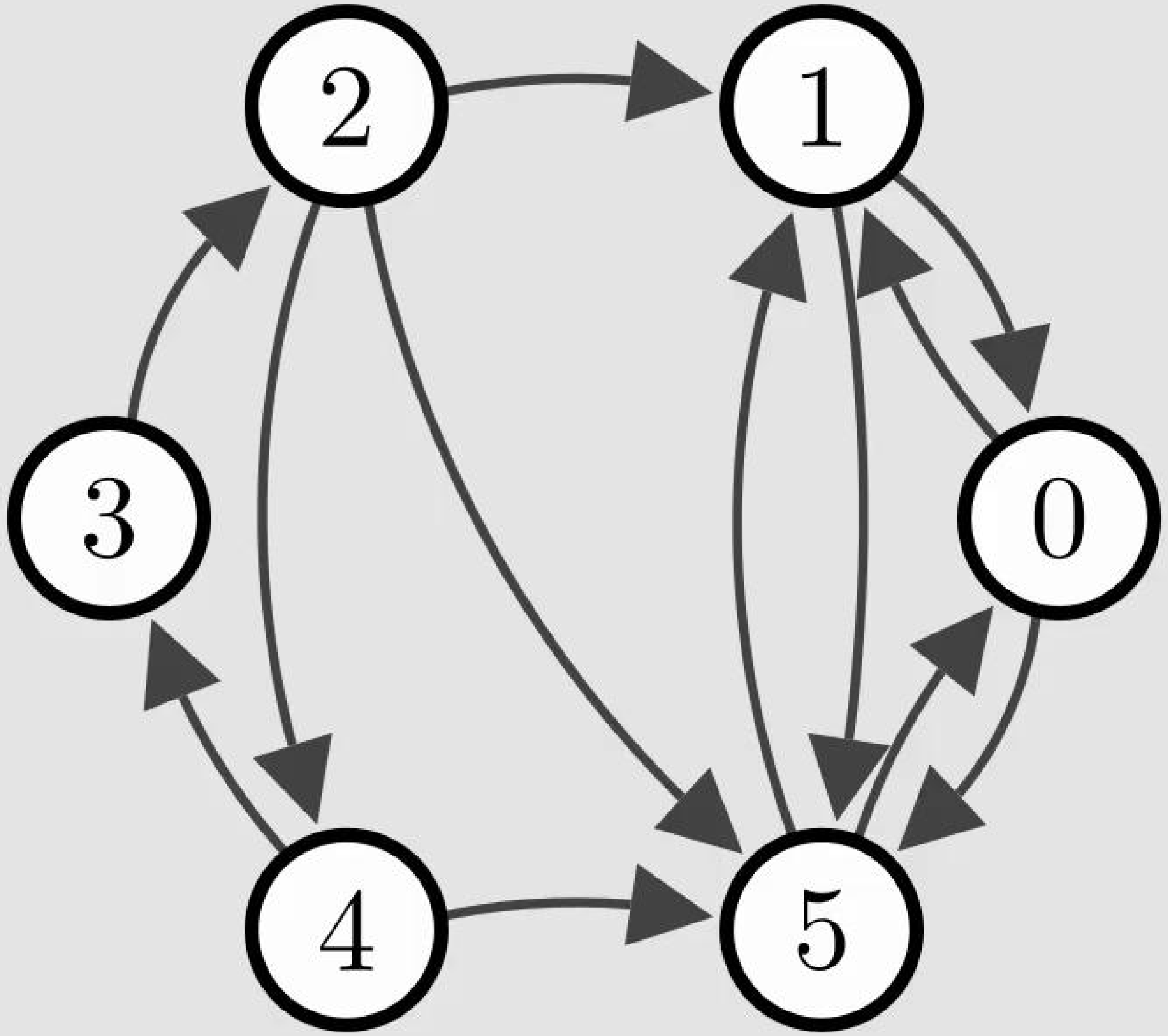


SCC Propagation



SCC Propagation

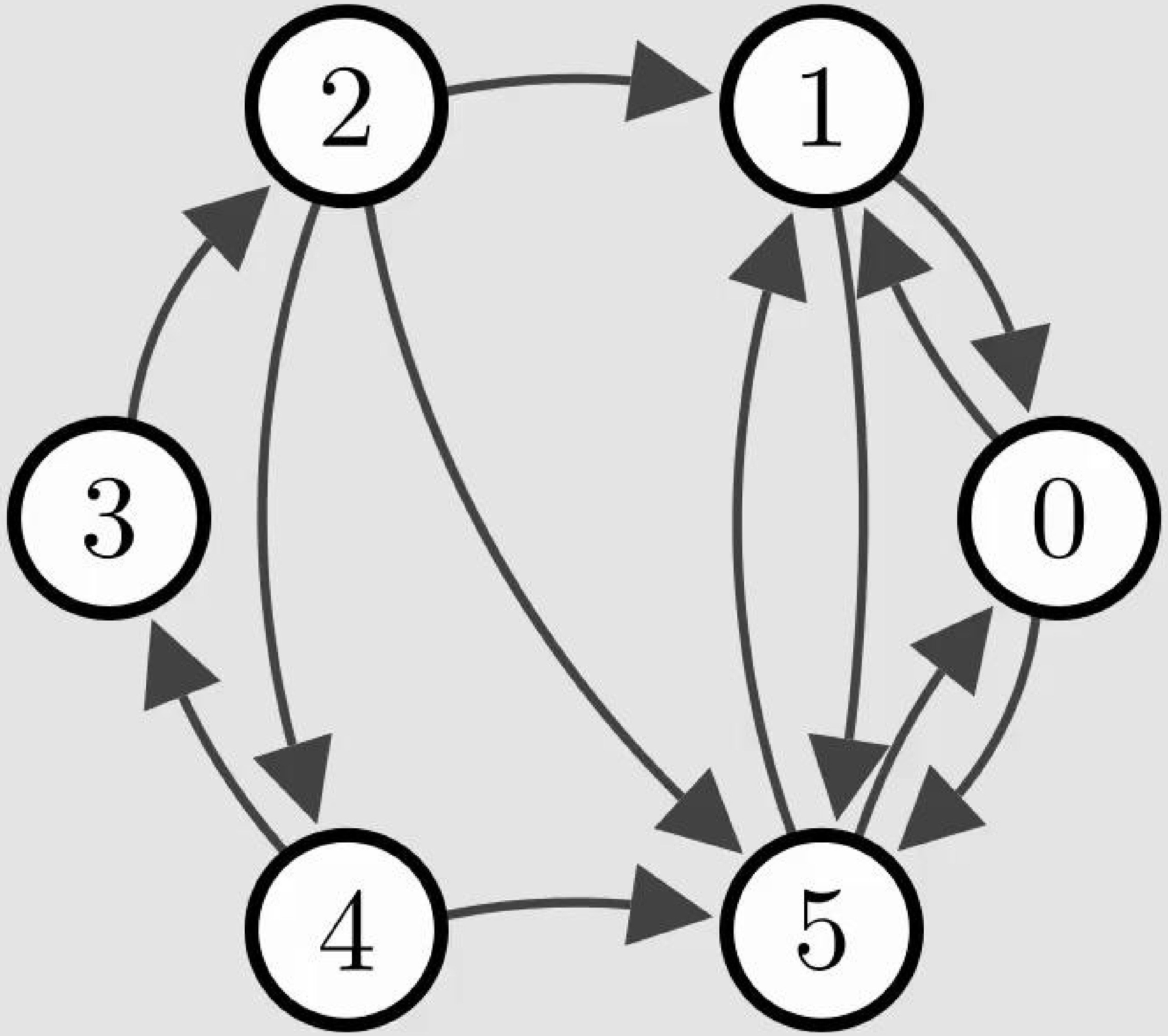
If AllDiff is enforced:



SCC Propagation

If AllDiff is enforced:

No subcycles

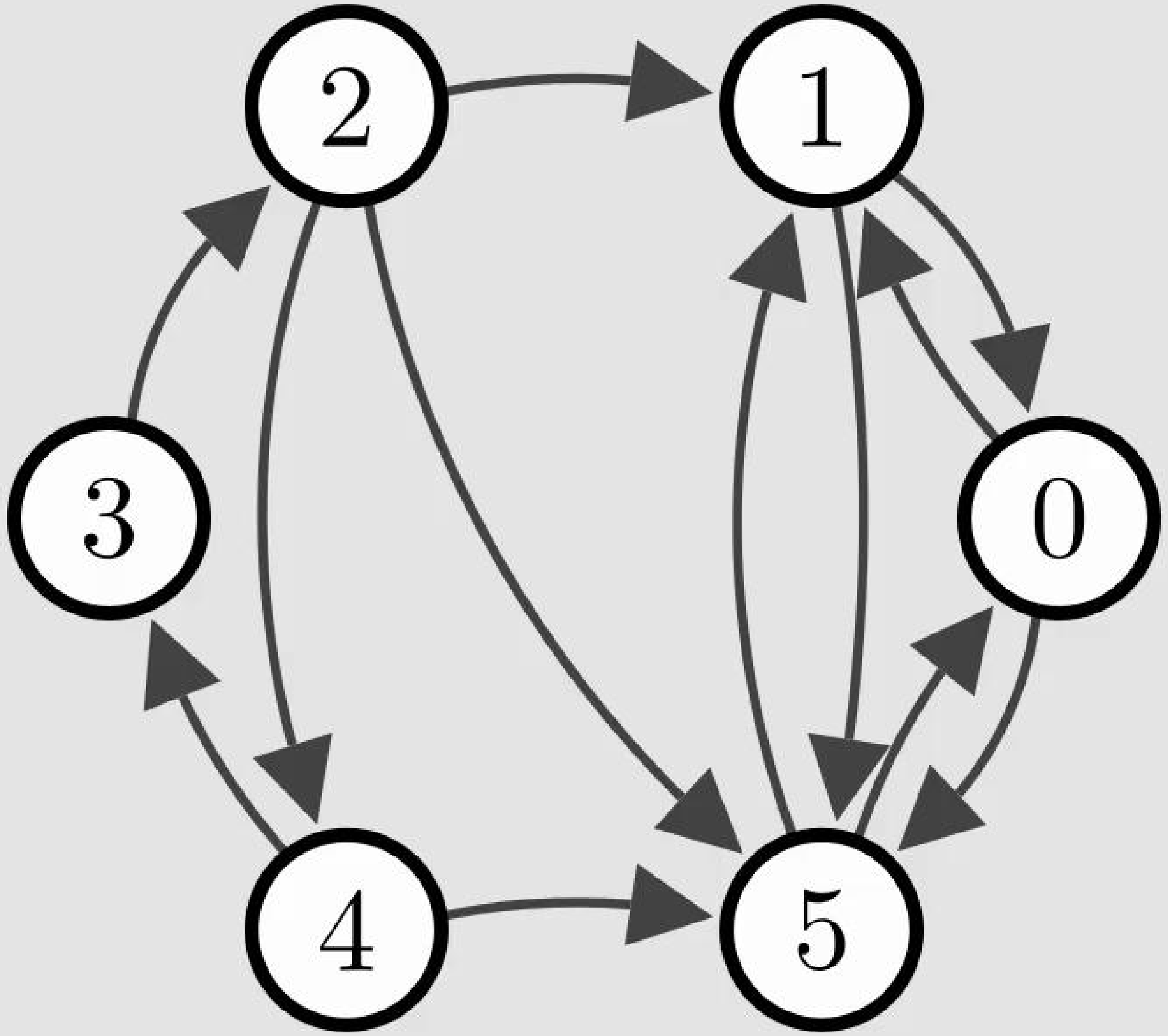


SCC Propagation

If AllDiff is enforced:

No subcycles

$\iff$



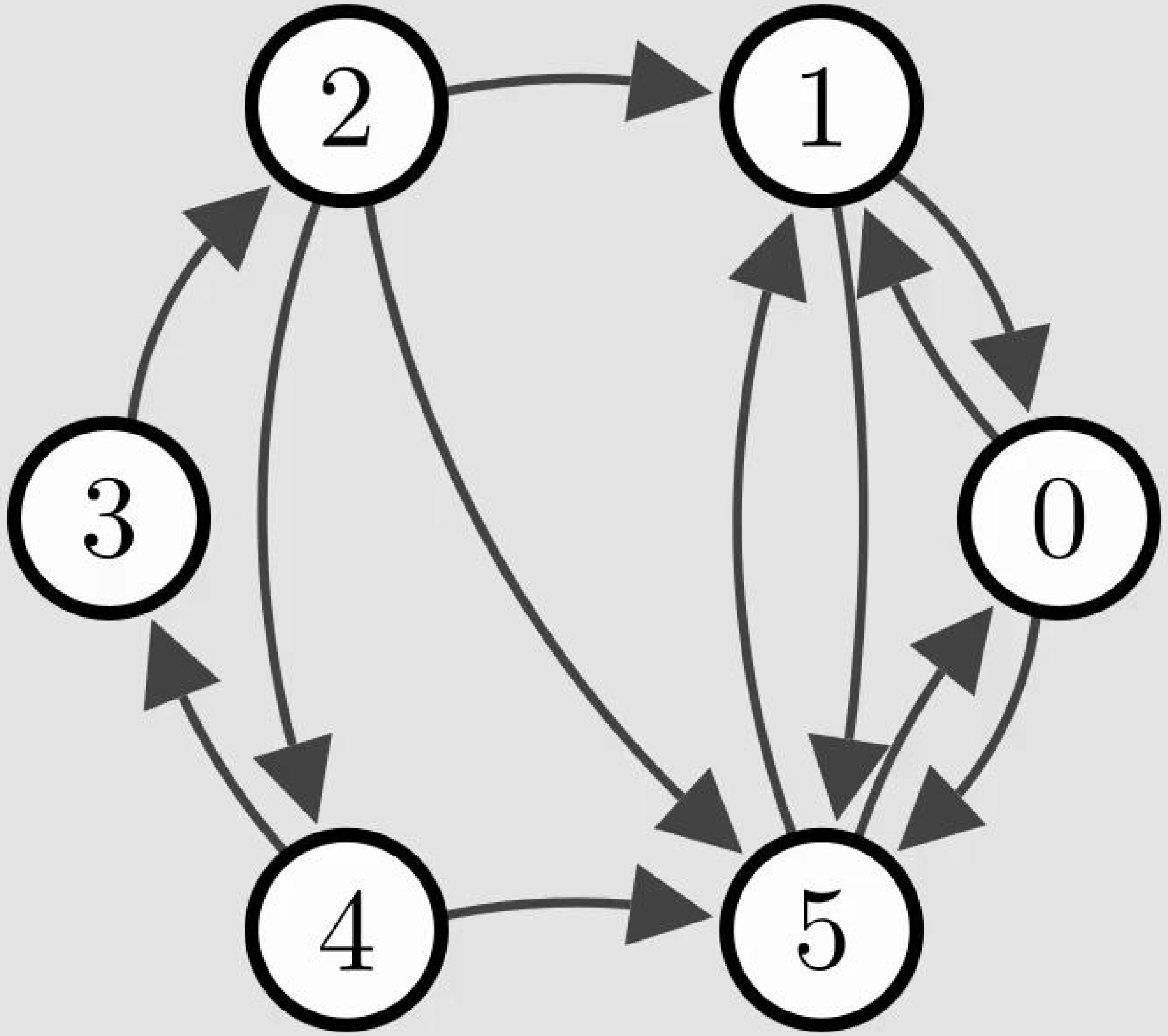
SCC Propagation

If AllDiff is enforced:

No subcycles

$\iff$

All vertices part of one cycle



SCC Propagation

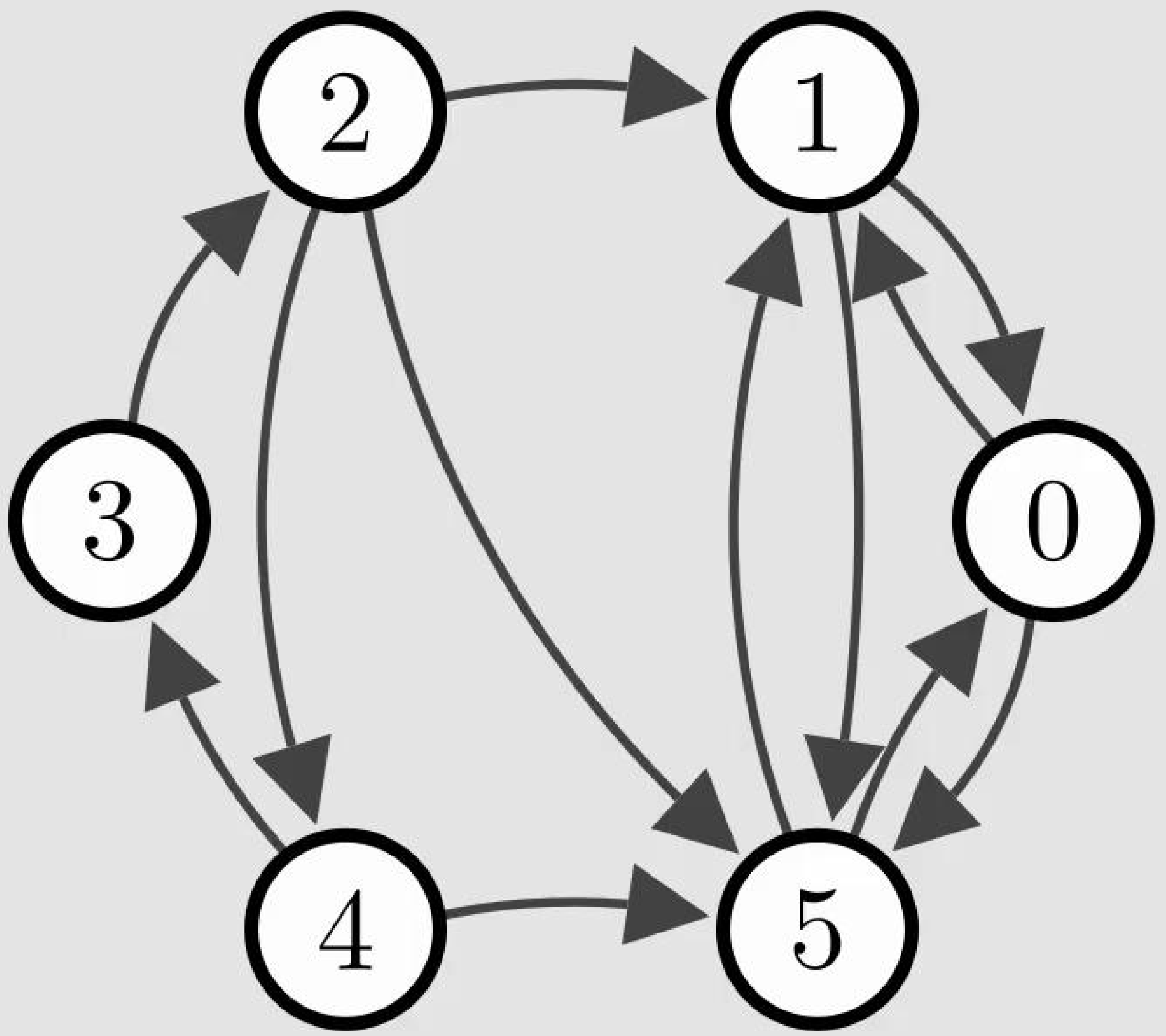
If AllDiff is enforced:

No subcycles

$\iff$

All vertices part of one cycle

$\iff$



SCC Propagation

If AllDiff is enforced:

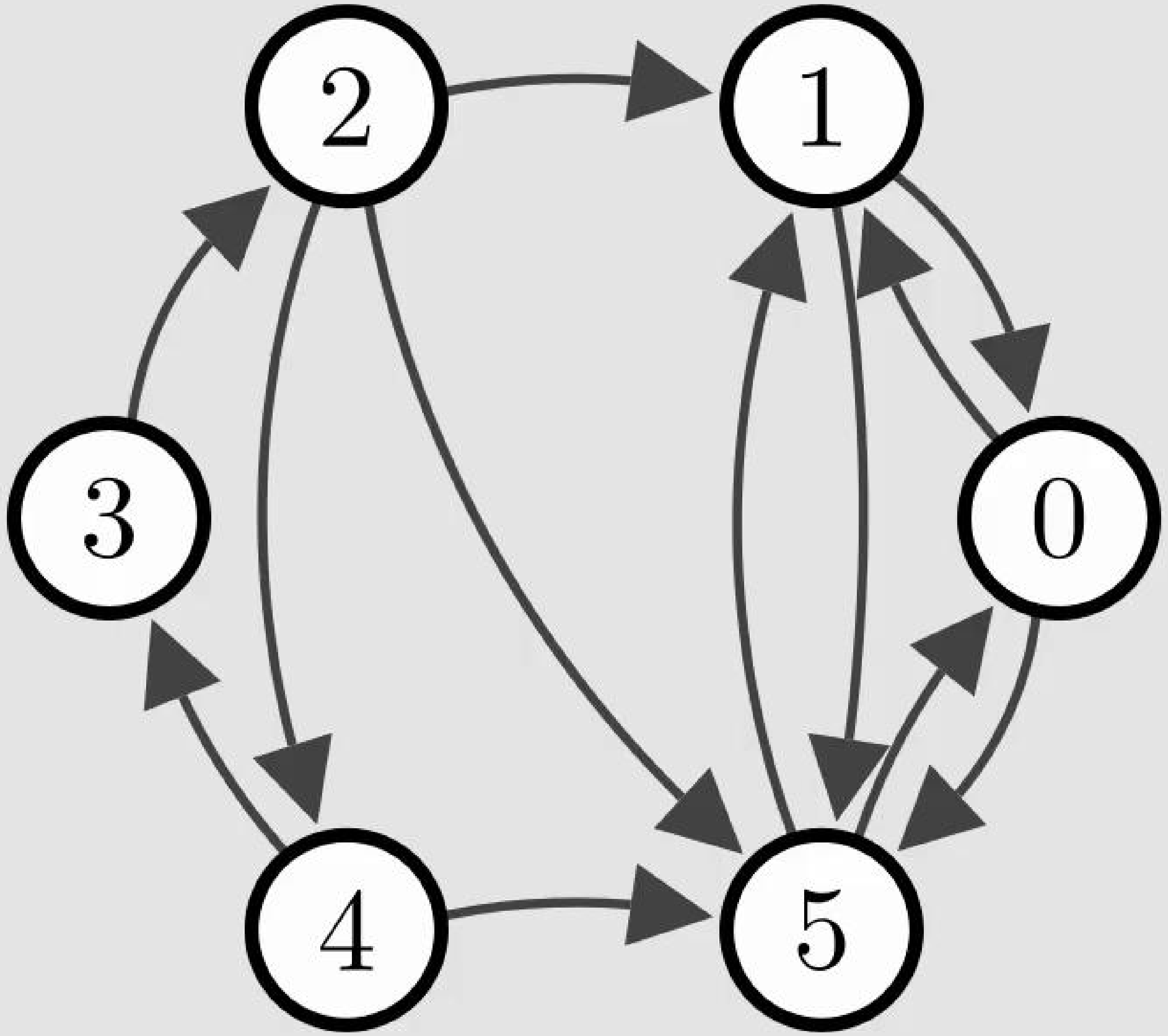
No subcycles

$\iff$

All vertices part of one cycle

$\iff$

Every vertex reachable from every vertex



SCC Propagation

If AllDiff is enforced:

No subcycles

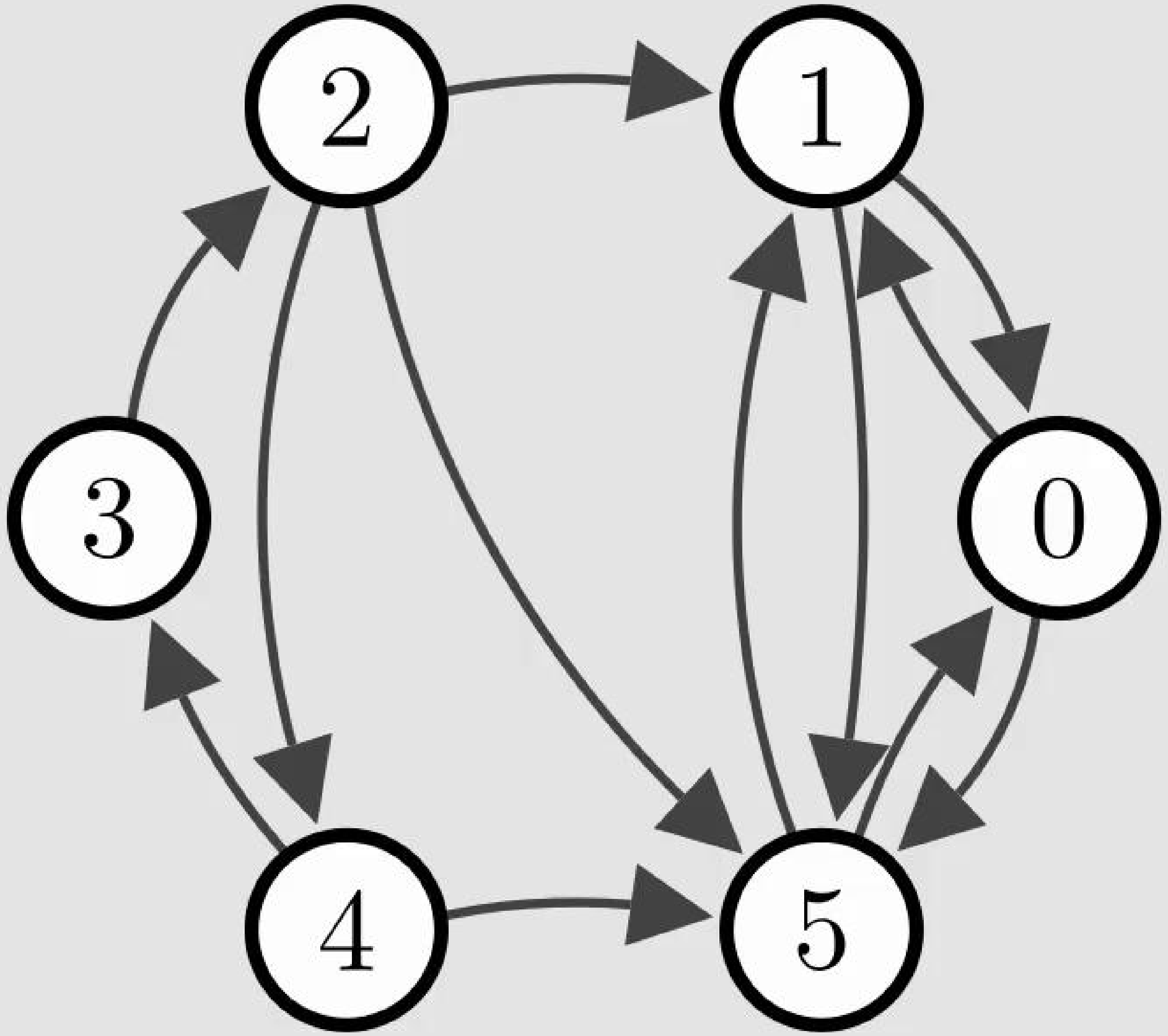
$\iff$

All vertices part of one cycle

$\iff$

Every vertex reachable from every vertex

$\iff$



SCC Propagation

If AllDiff is enforced:

No subcycles

$\iff$

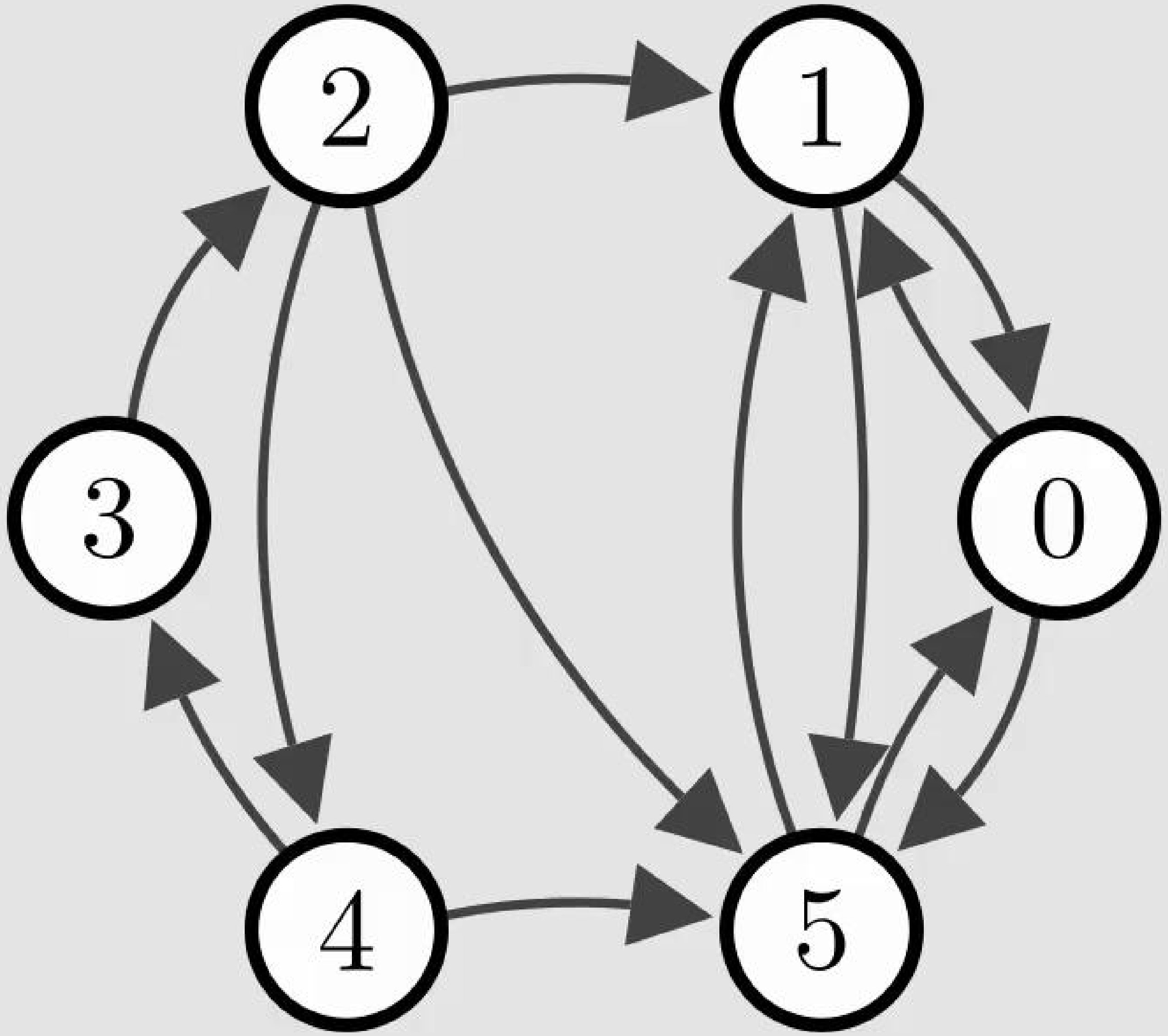
All vertices part of one cycle

$\iff$

Every vertex reachable from every vertex

$\iff$

One one strongly connected component



SCC Propagation

If AllDiff is enforced:

No subcycles

$\iff$

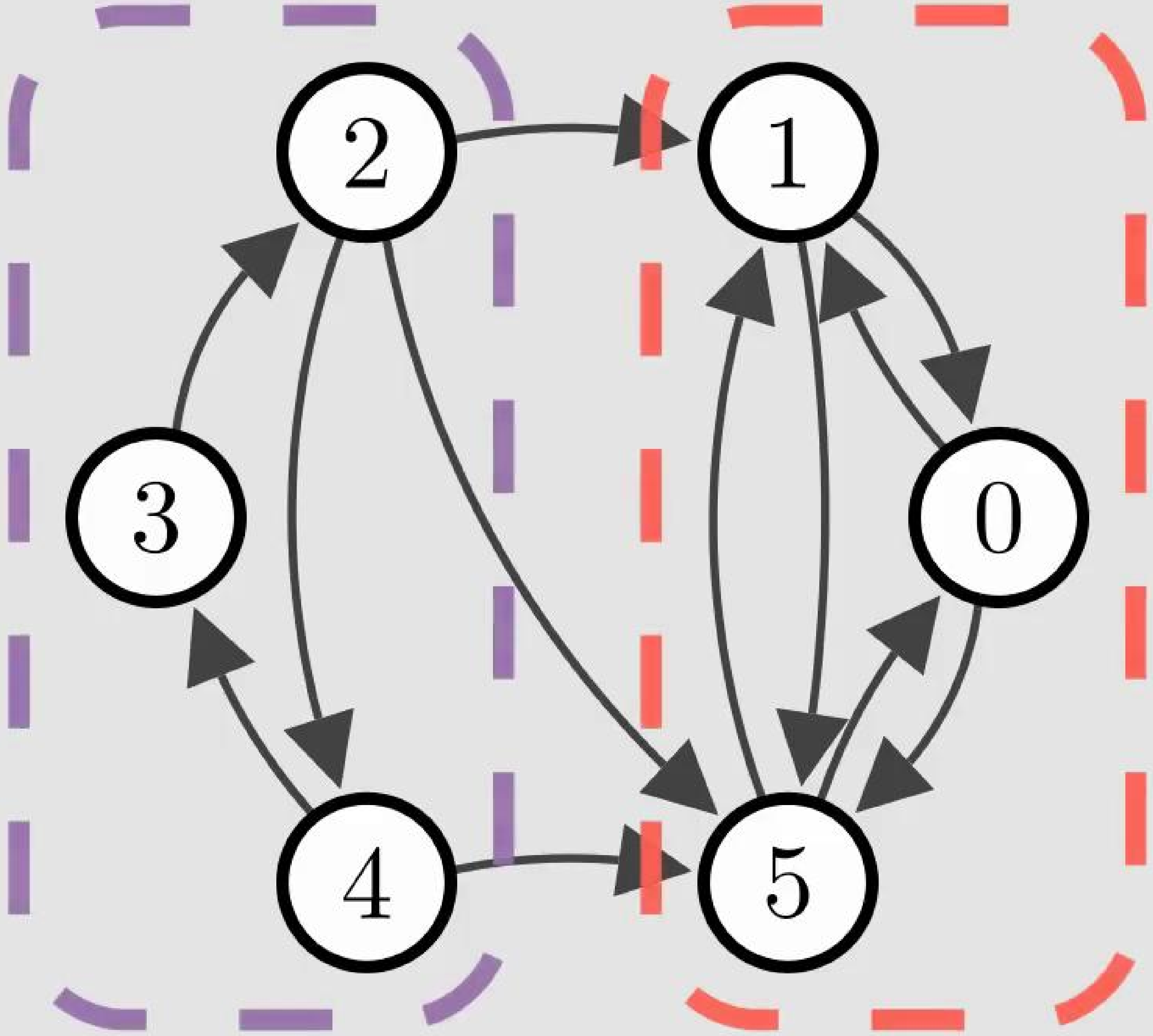
All vertices part of one cycle

$\iff$

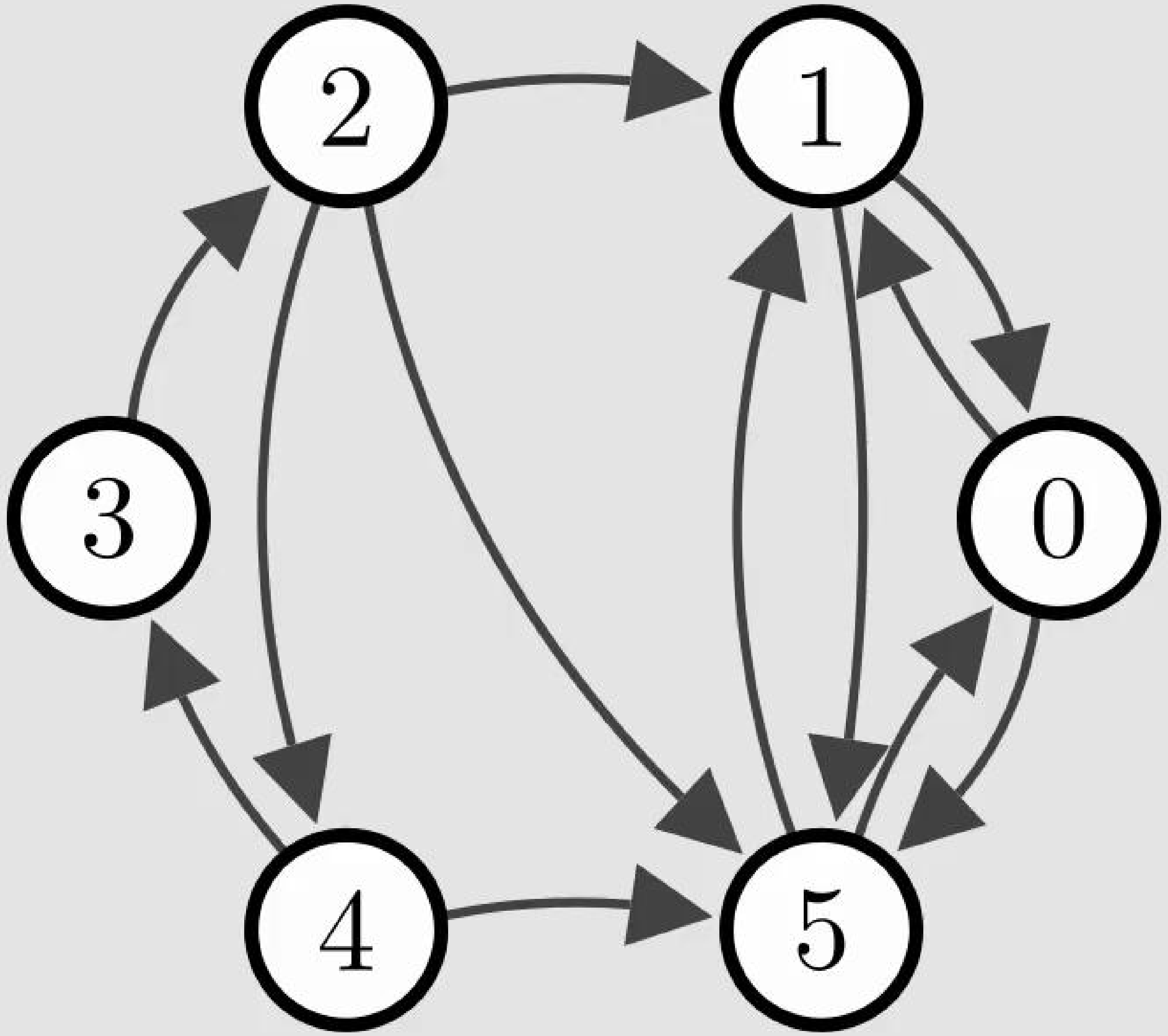
Every vertex reachable from every vertex

$\iff$

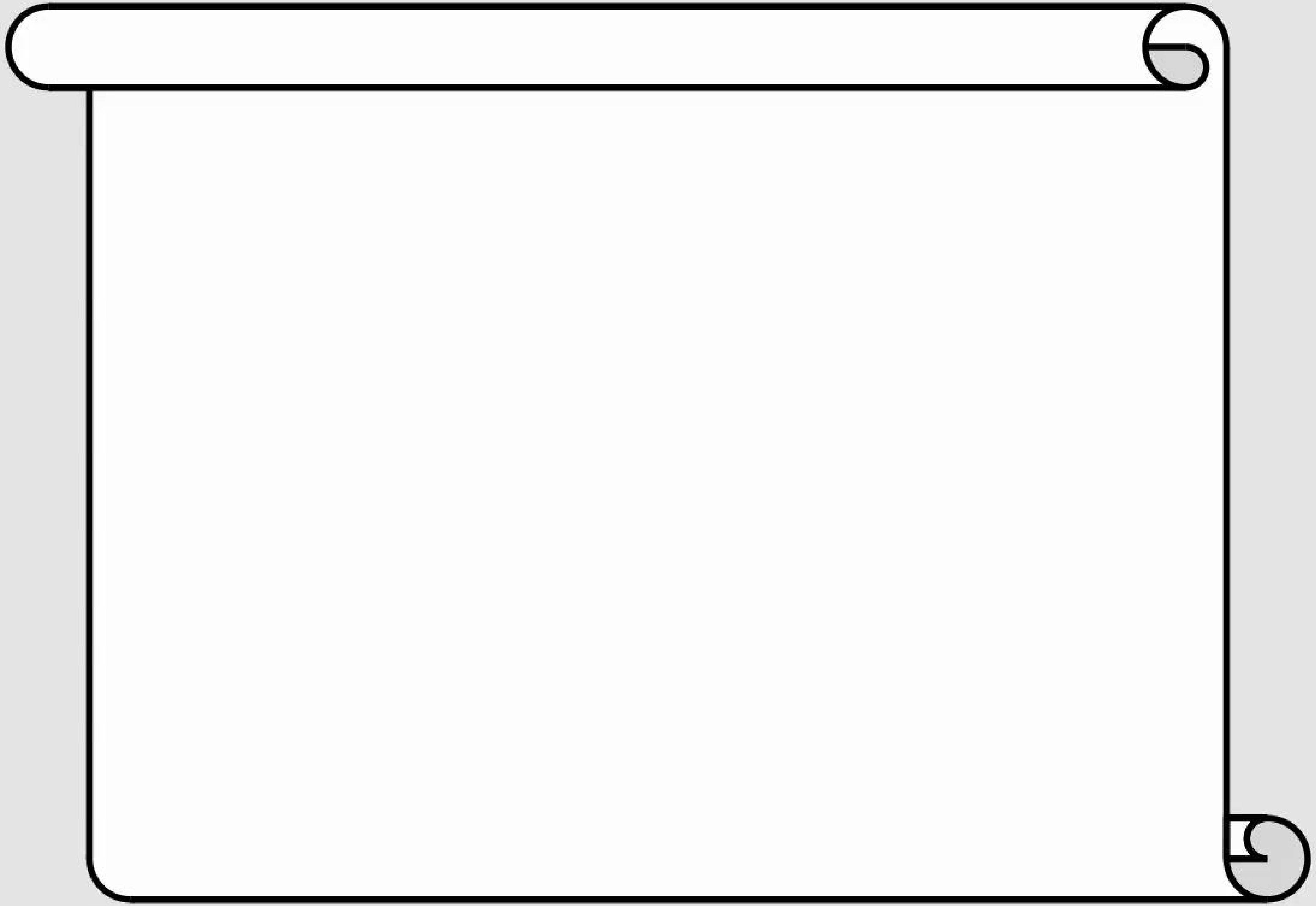
One one strongly connected component

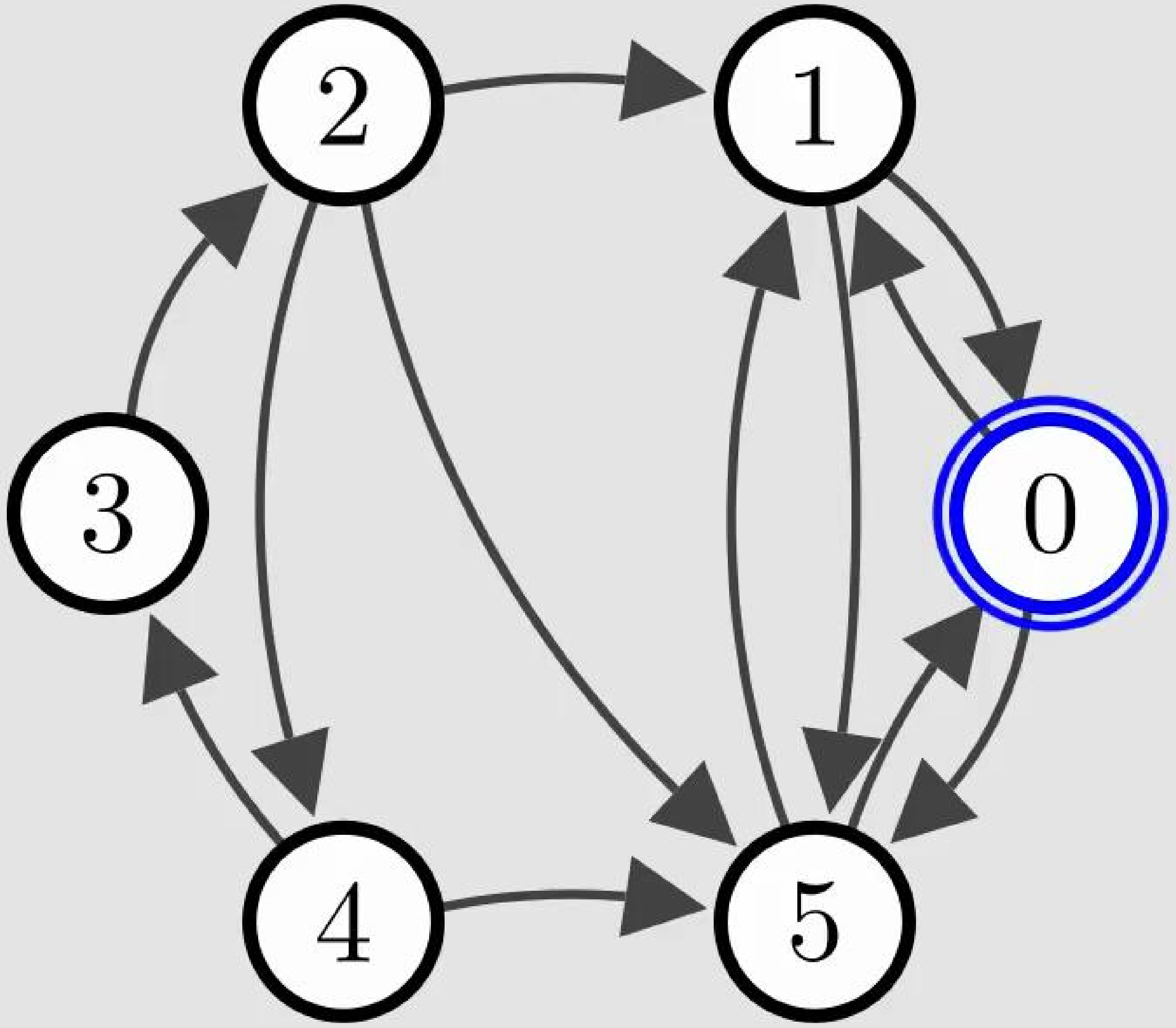


SCC Propagation



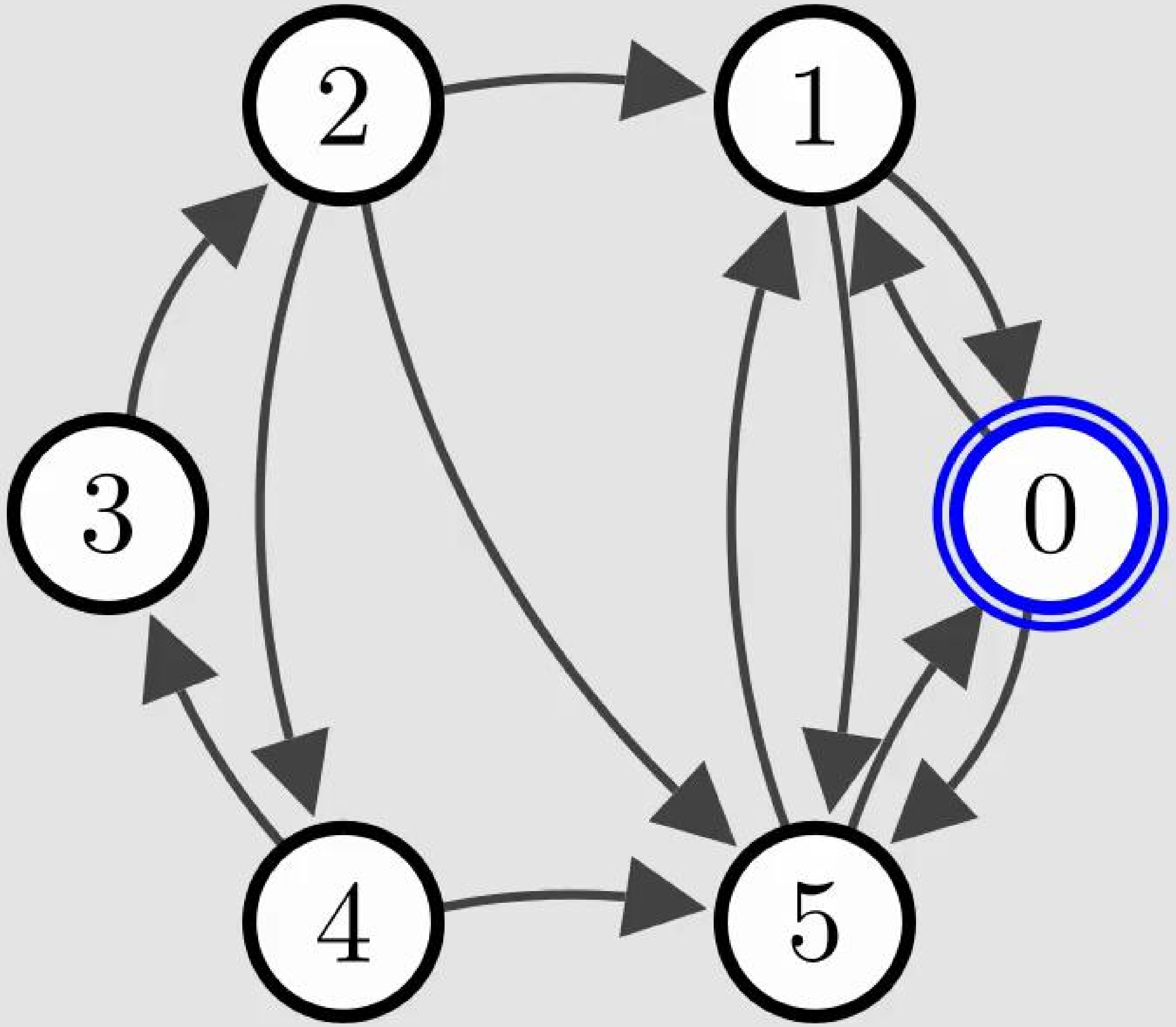
SCC Propagation





SCC Propagation

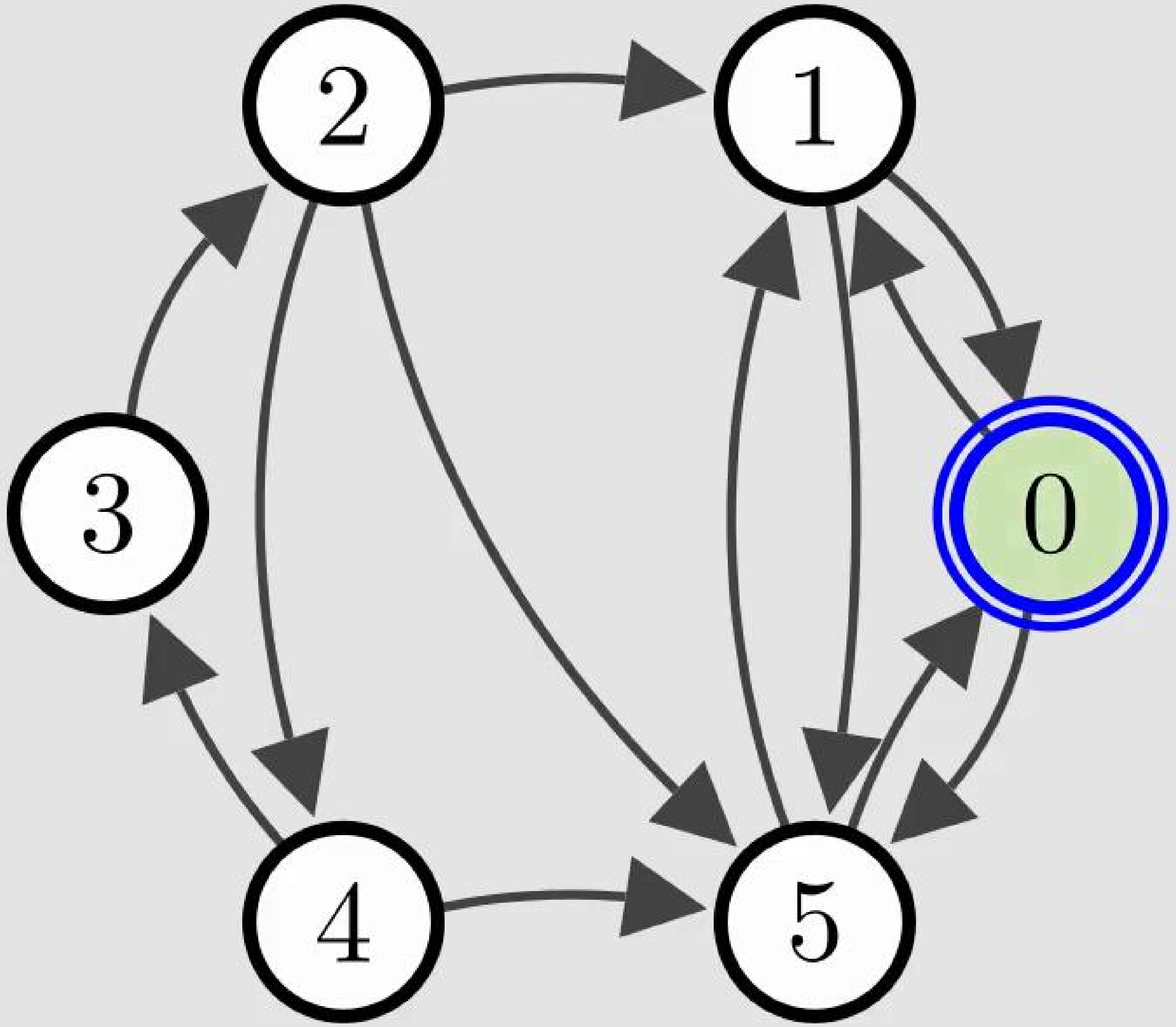
```
ReachTooSmall( 0 )
```



SCC Propagation

ReachTooSmall(0)

$$\{P_0\} = 0$$

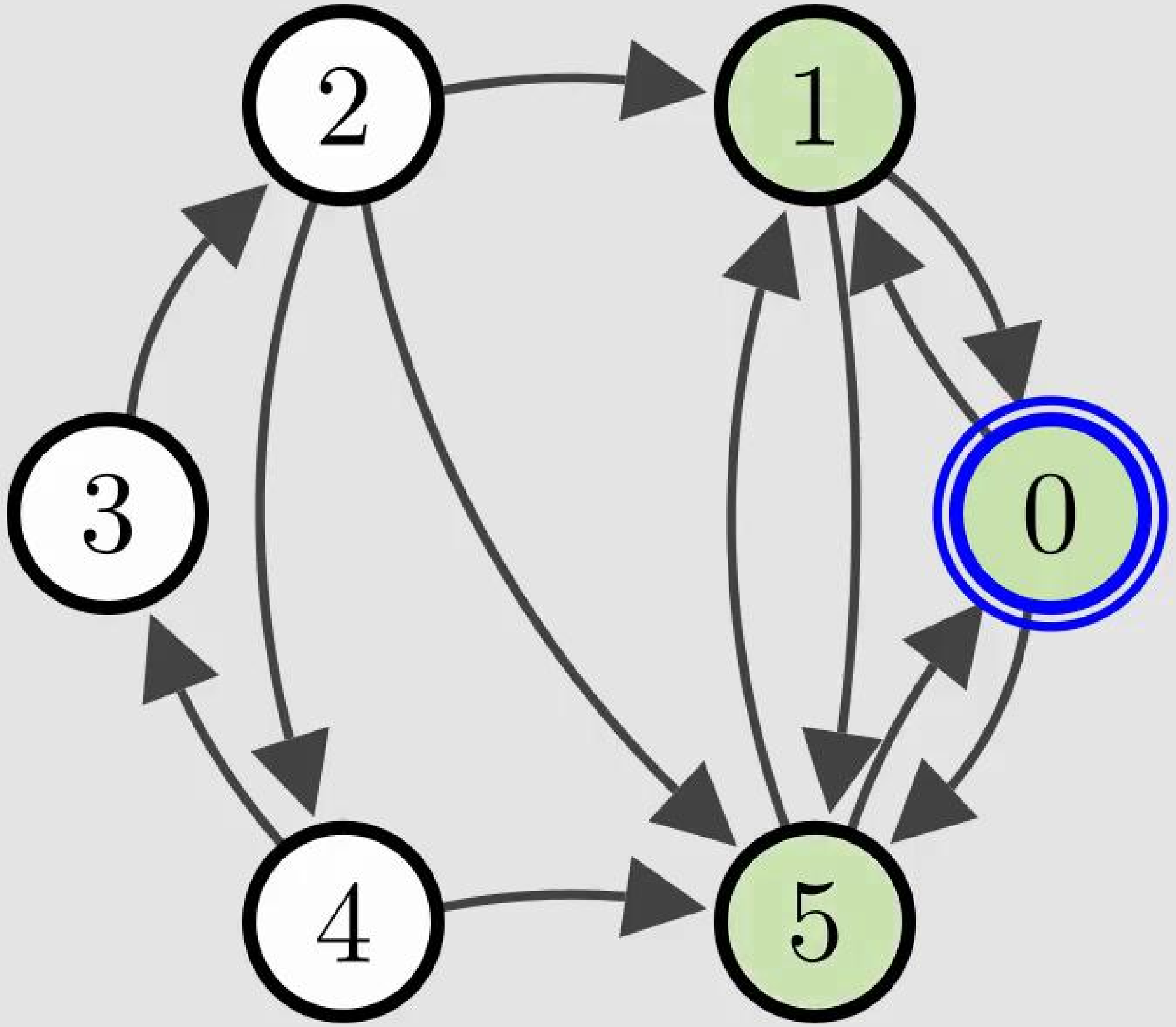


SCC Propagation

```
ReachTooSmall( 0 )
```

$$\{P_0\} = 0$$

$$\{P_1, P_5\} = 1$$



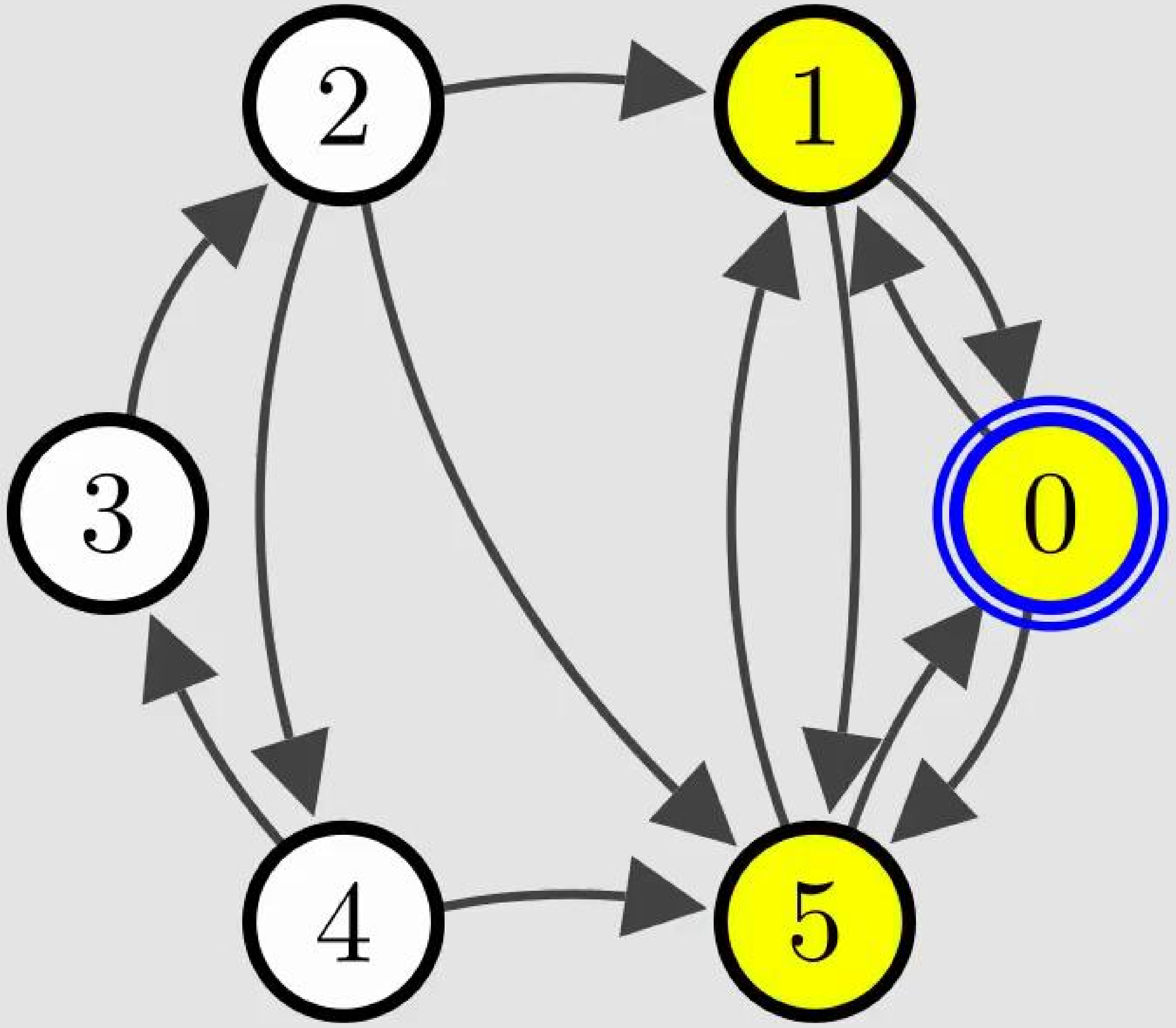
SCC Propagation

```
ReachTooSmall( 0 )
```

$$\{P_0\} = 0$$

$$\{P_1, P_5\} = 1$$

$$\{P_0, P_1, P_5\} = 2$$



SCC Propagation

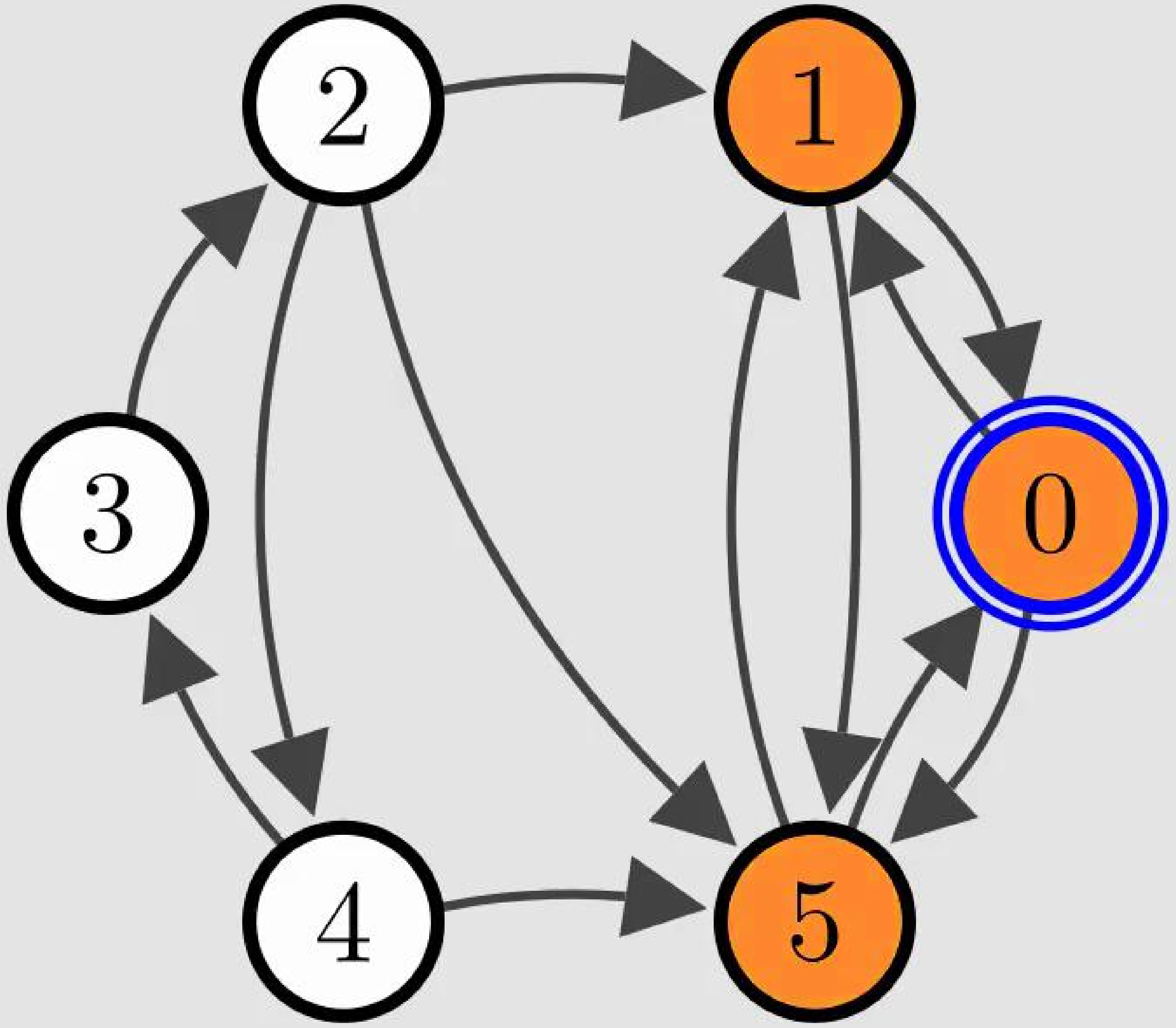
```
ReachTooSmall( 0 )
```

$$\{P_0\} = 0$$

$$\{P_1, P_5\} = 1$$

$$\{P_0, P_1, P_5\} = 2$$

$$\{P_0, P_1, P_5\} = 3$$



SCC Propagation

```
ReachTooSmall( 0 )
```

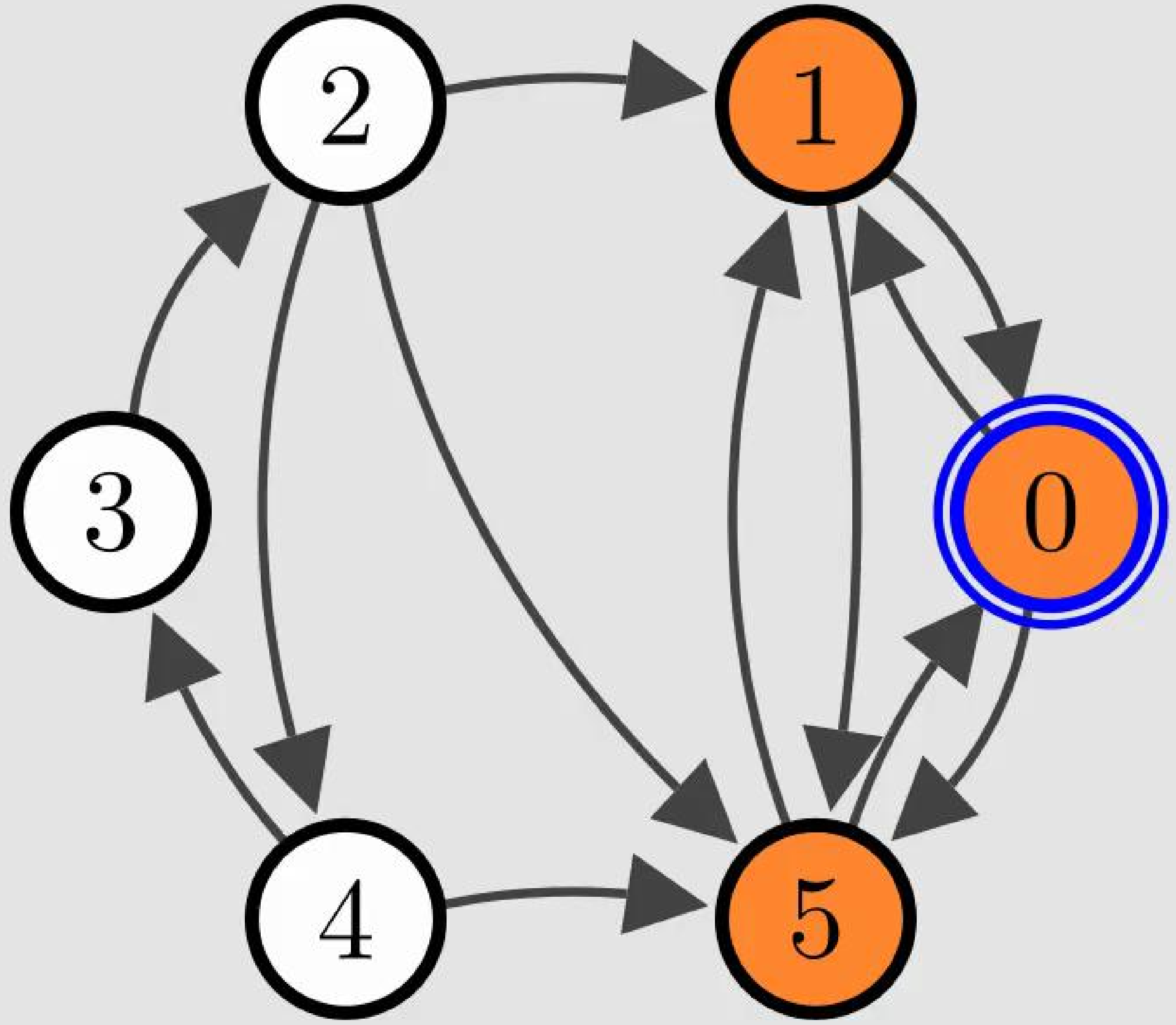
$$\{P_0\} = 0$$

$$\{P_1, P_5\} = 1$$

$$\{P_0, P_1, P_5\} = 2$$

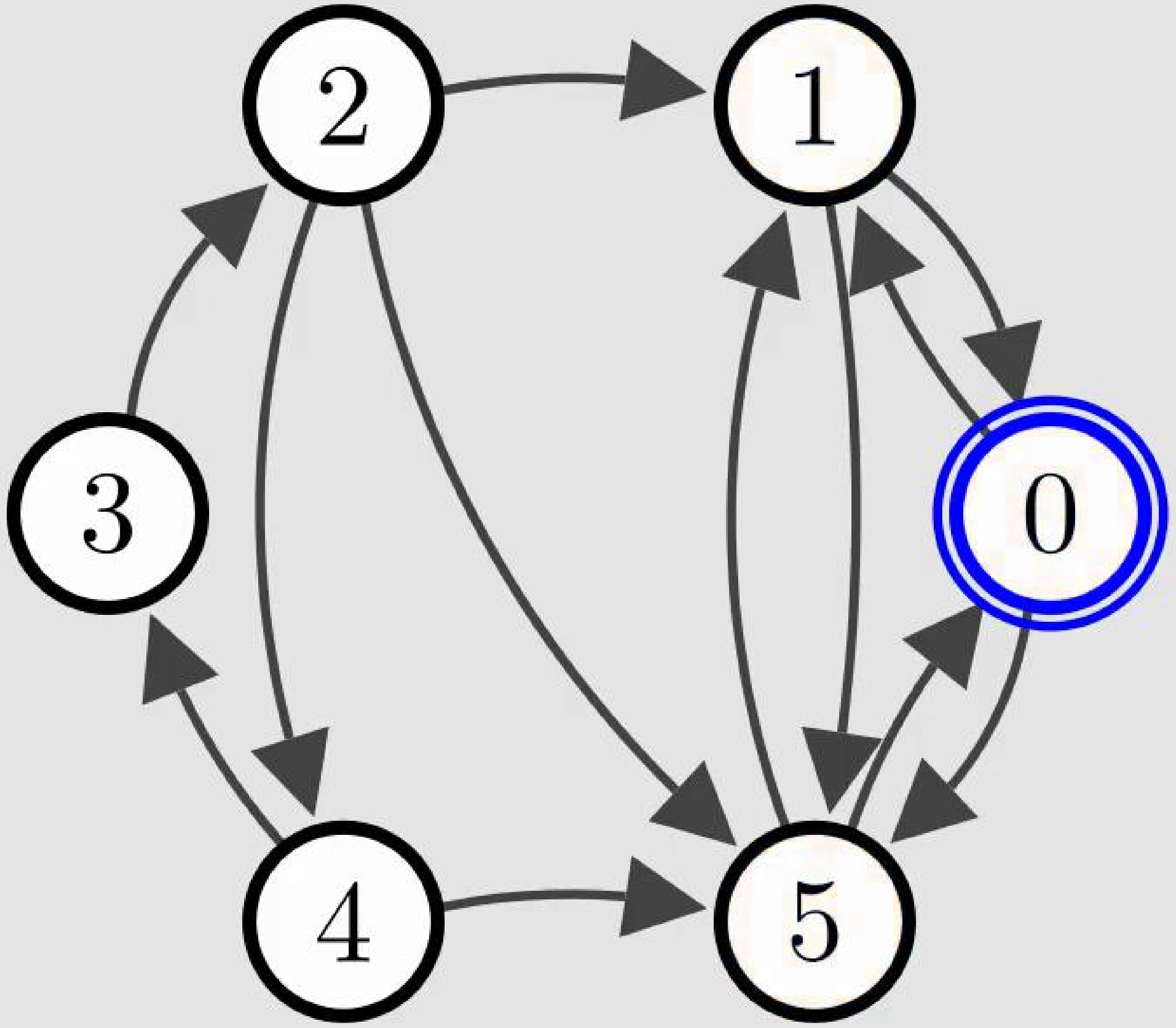
$$\{P_0, P_1, P_5\} = 3$$

$$\mathcal{G} \implies 0 \geq 1$$



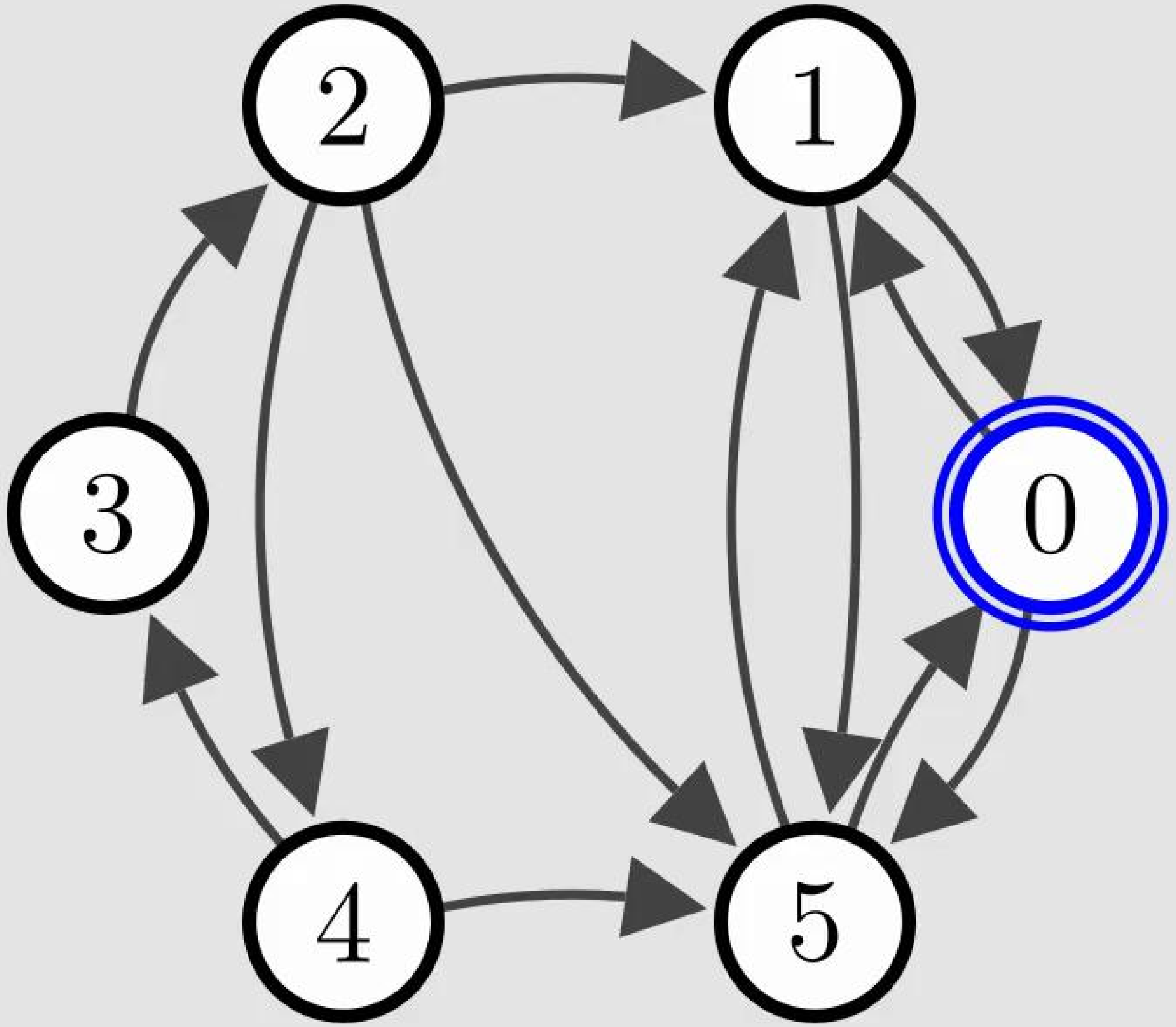
SCC Propagation

ReachTooSmall(0)



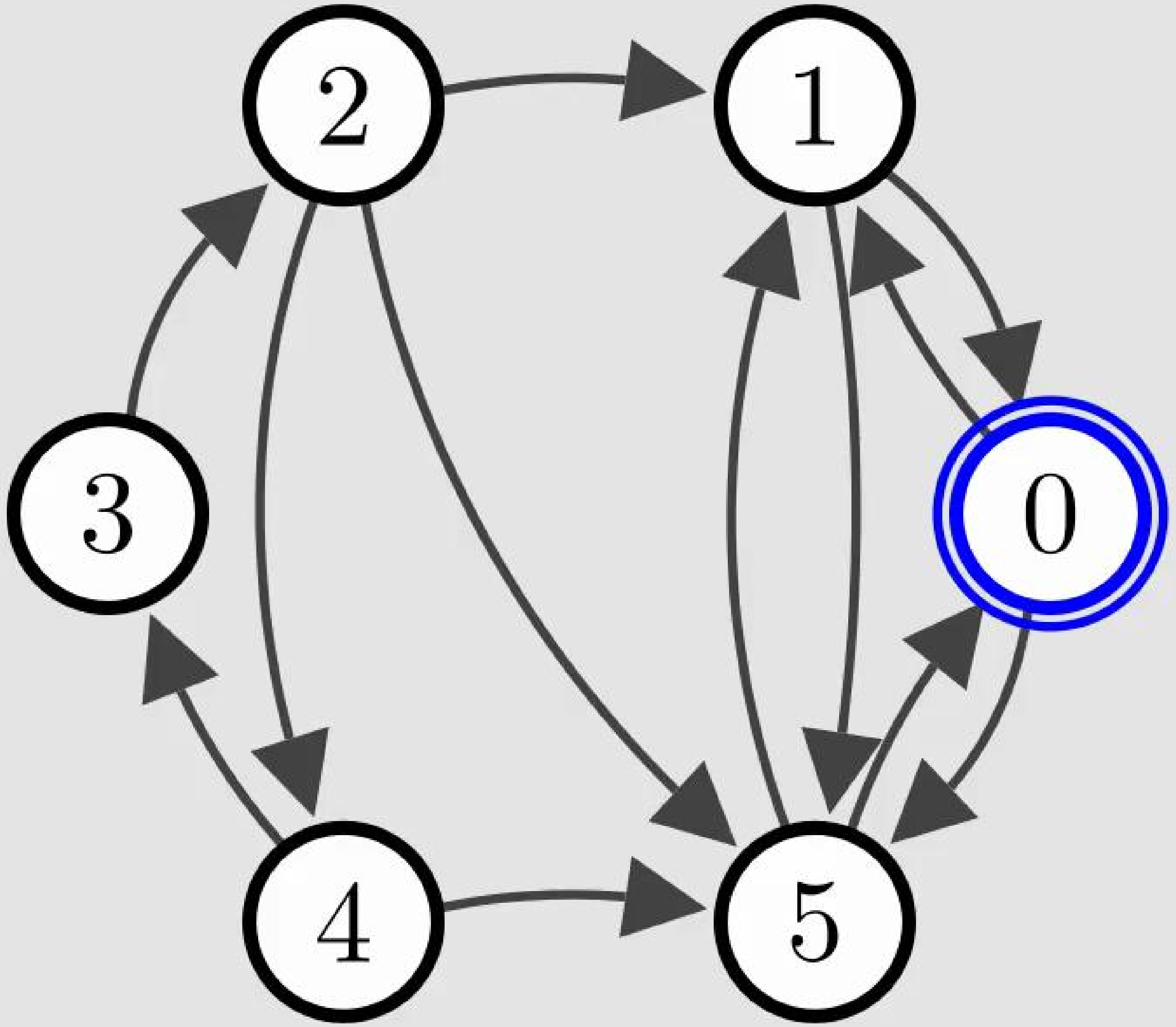
SCC Propagation

`ReachTooSmall( v )`



SCC Propagation

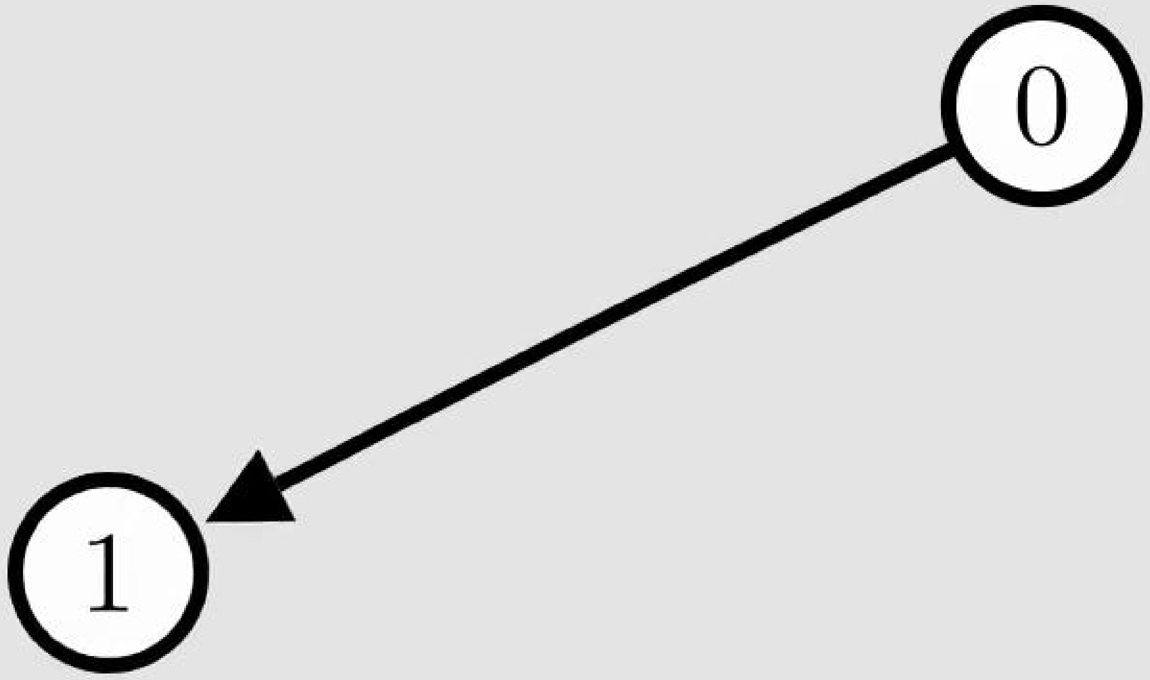
$c_1 \implies$ `ReachTooSmall( v )`



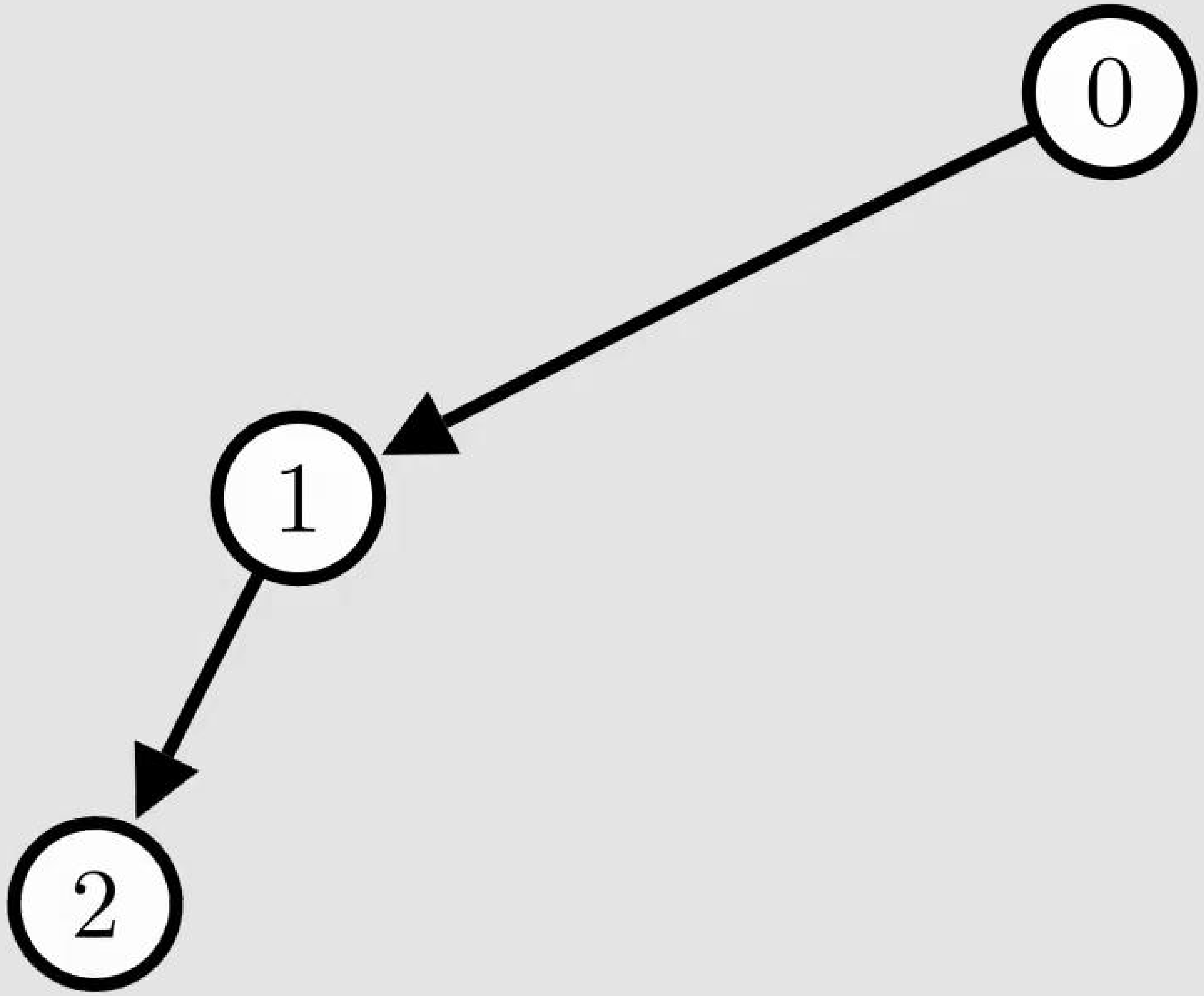
Further Propagation Rules

Further Propagation Rules

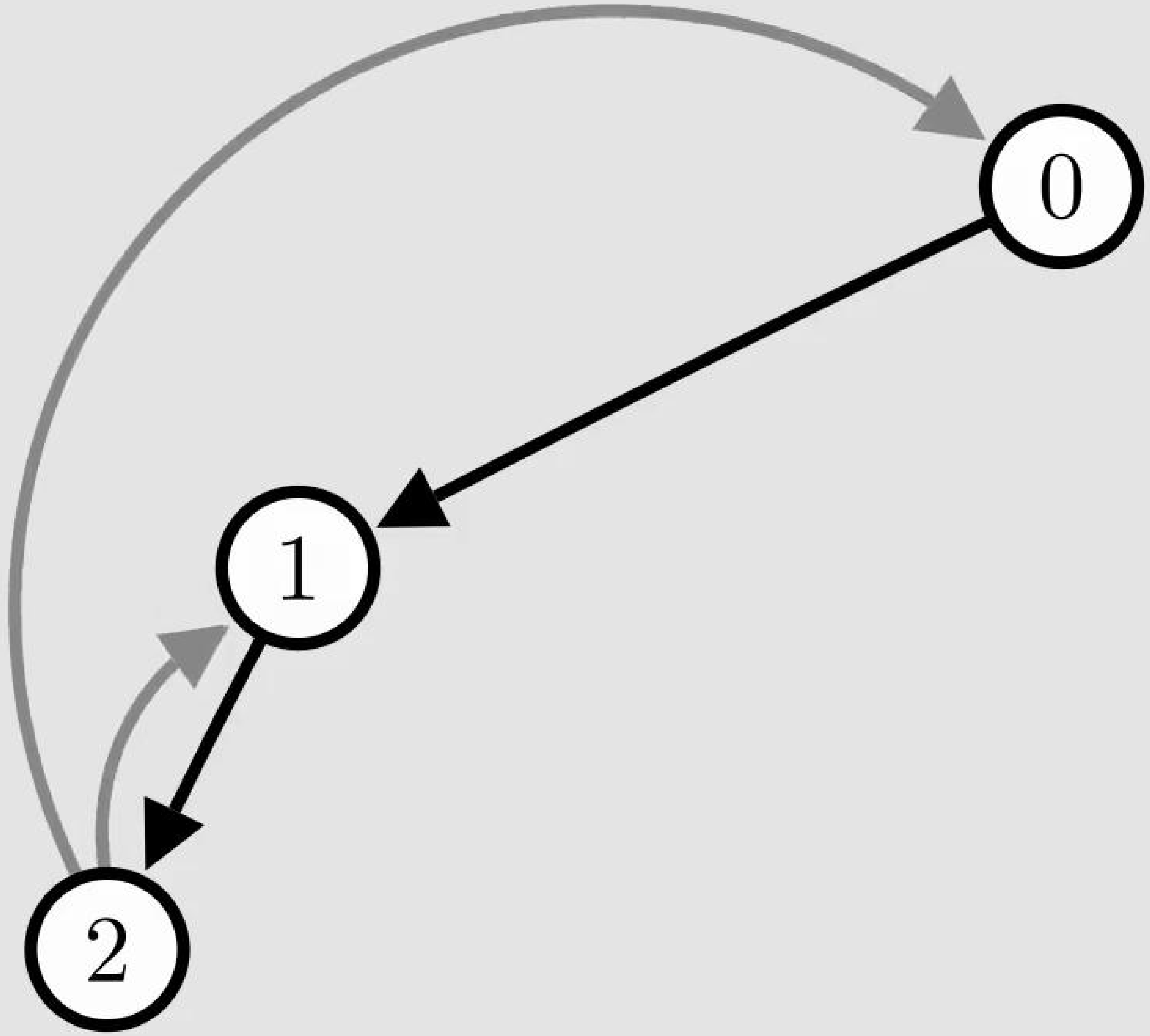
Further Propagation Rules



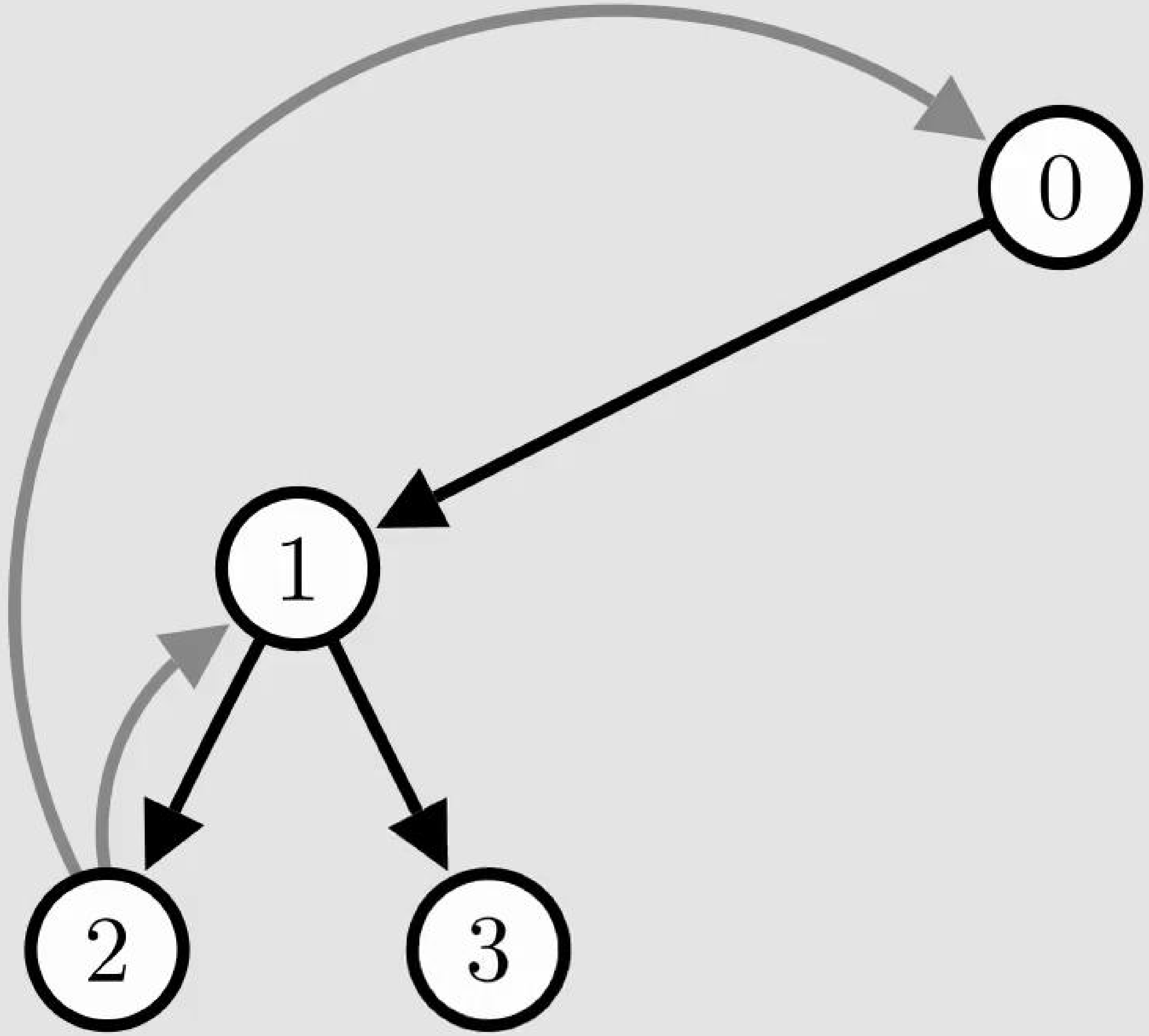
Further Propagation Rules



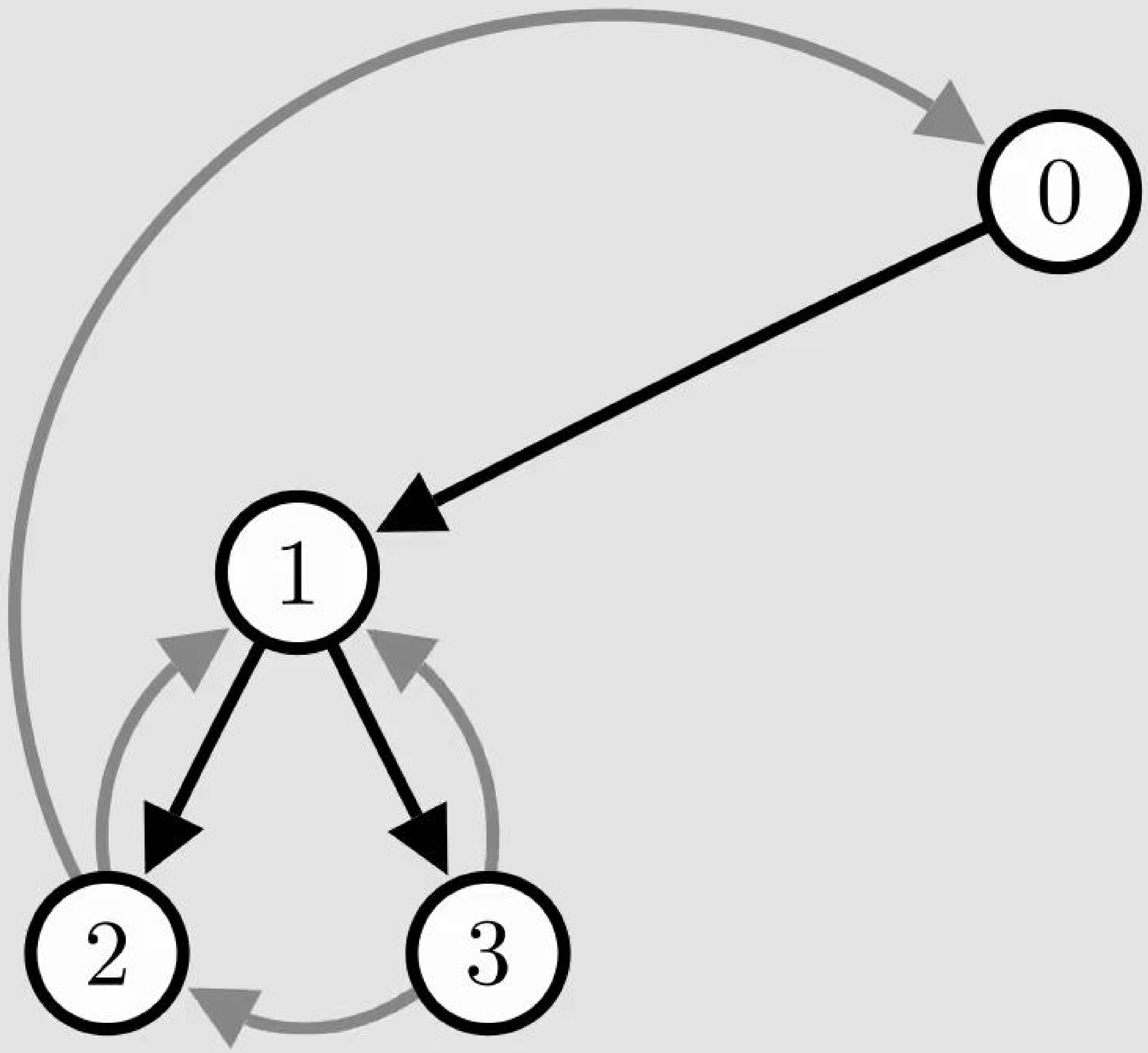
Further Propagation Rules



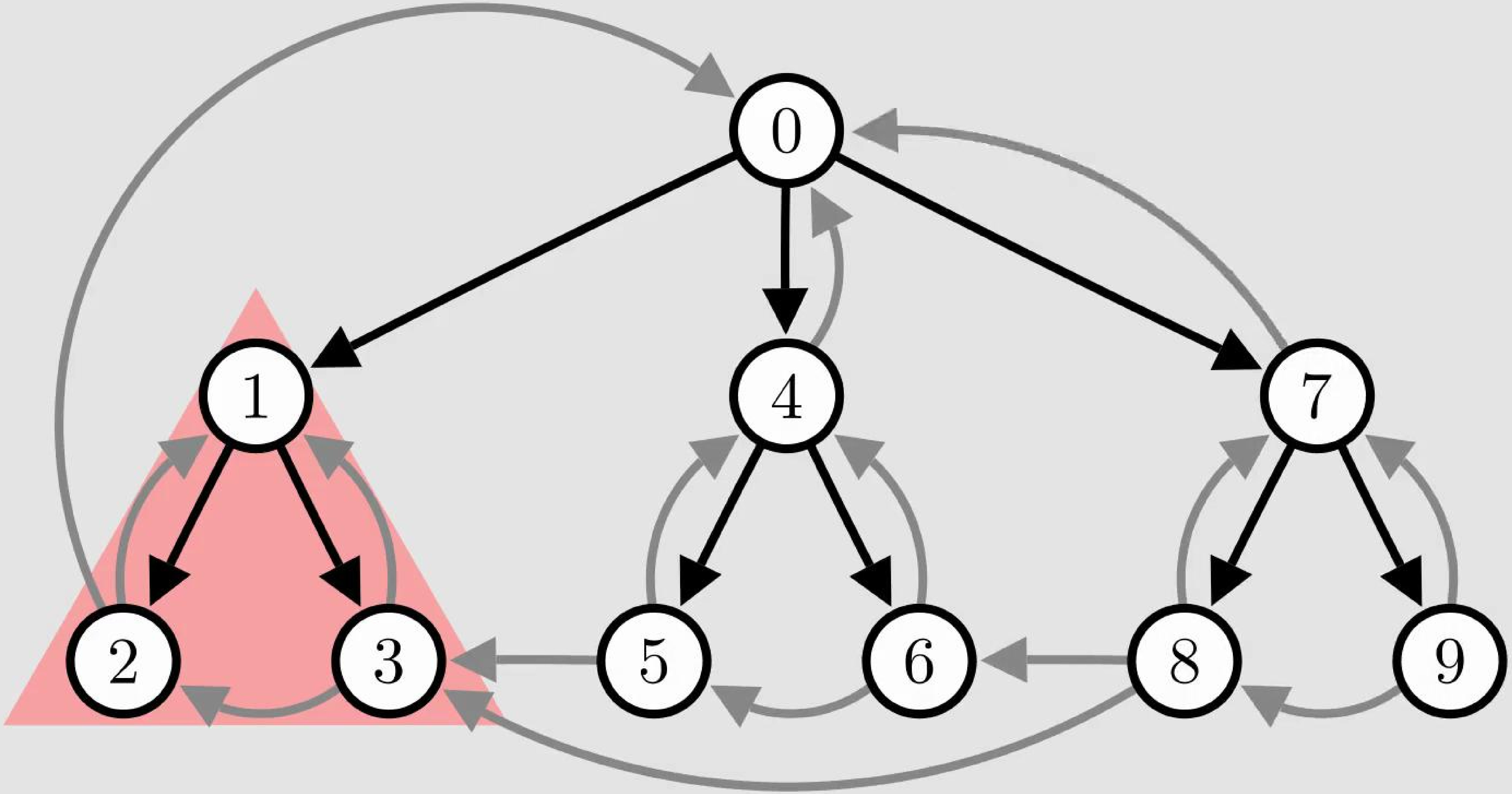
Further Propagation Rules



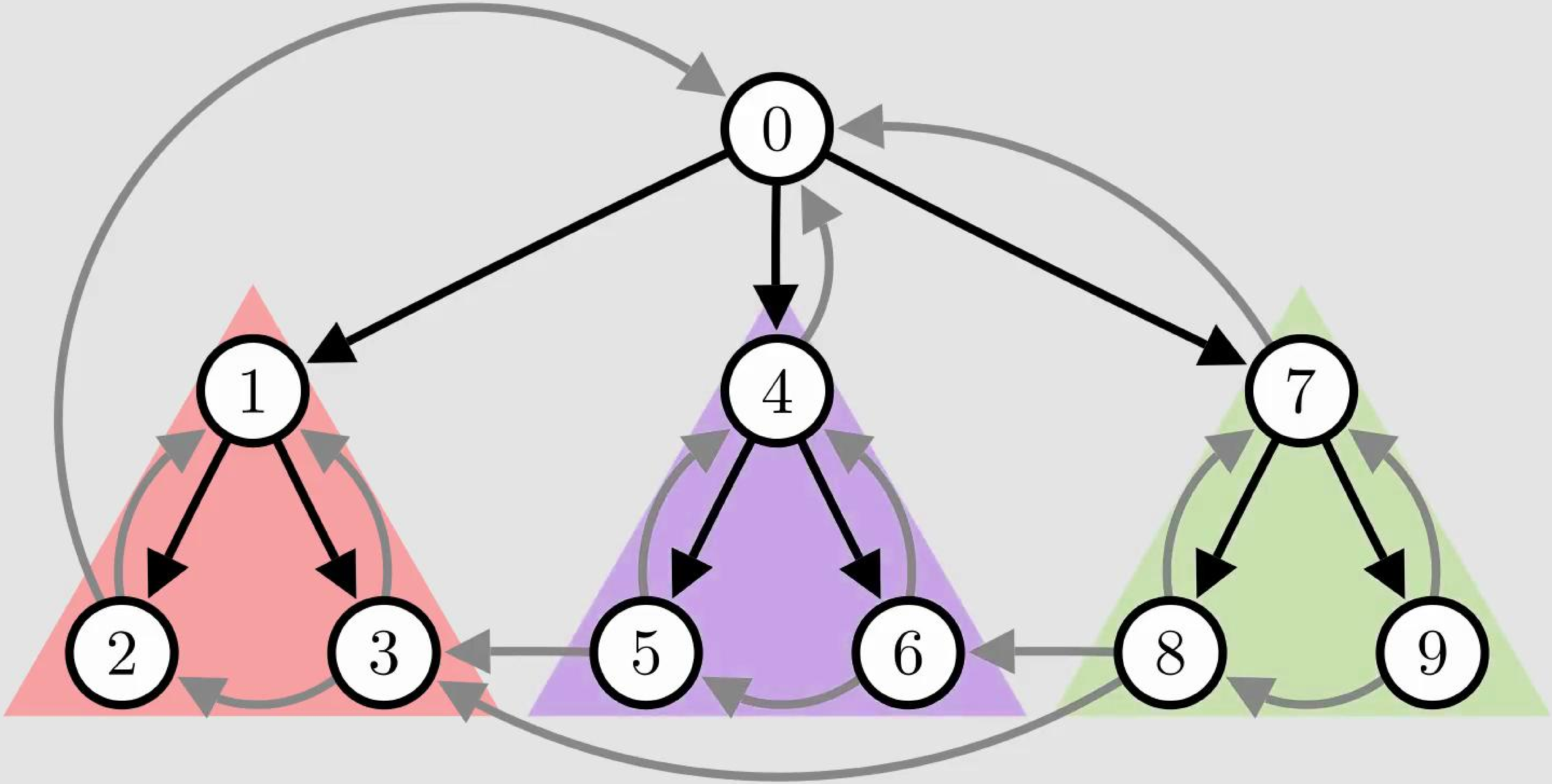
Further Propagation Rules



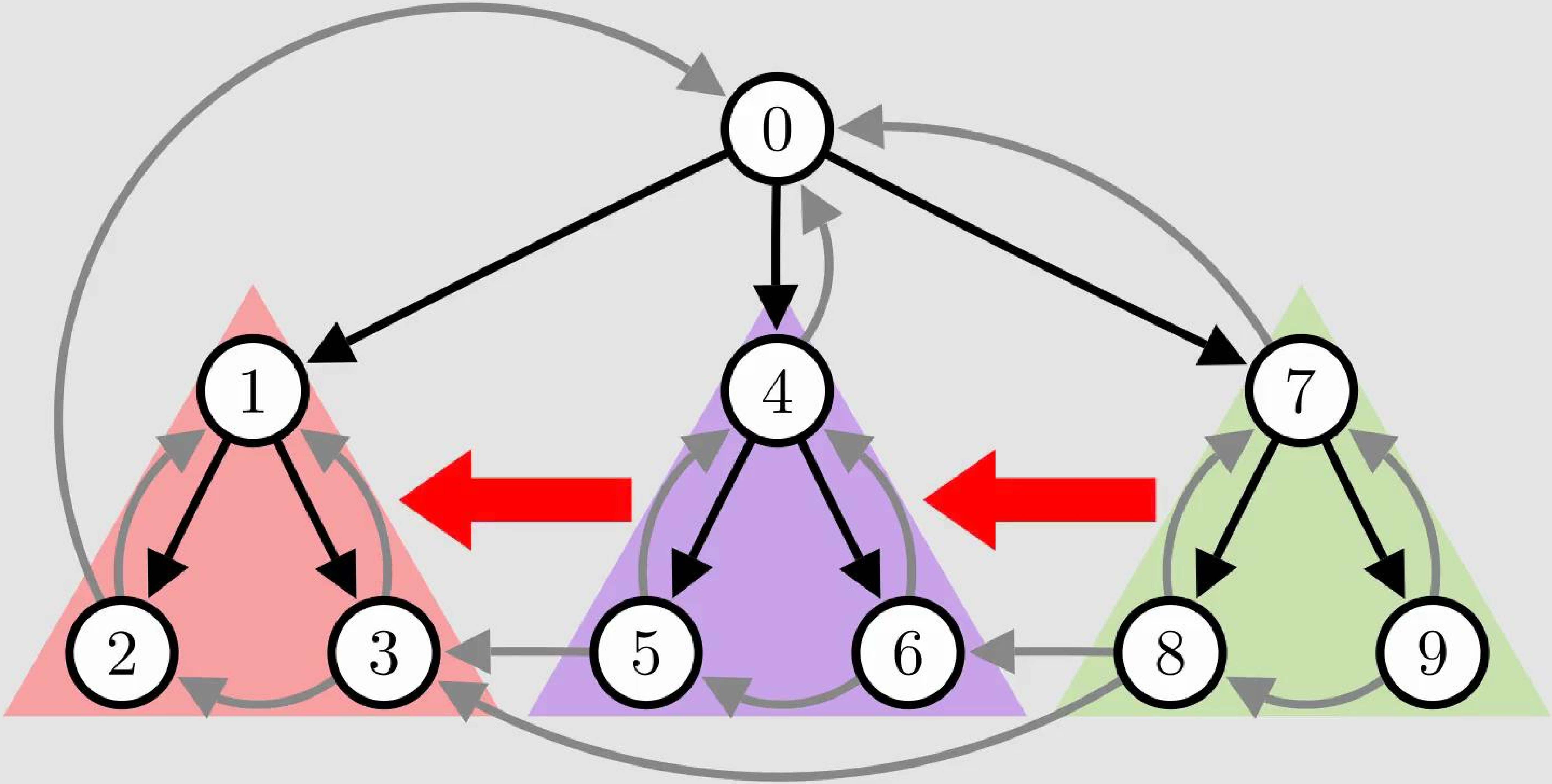
Further Propagation Rules



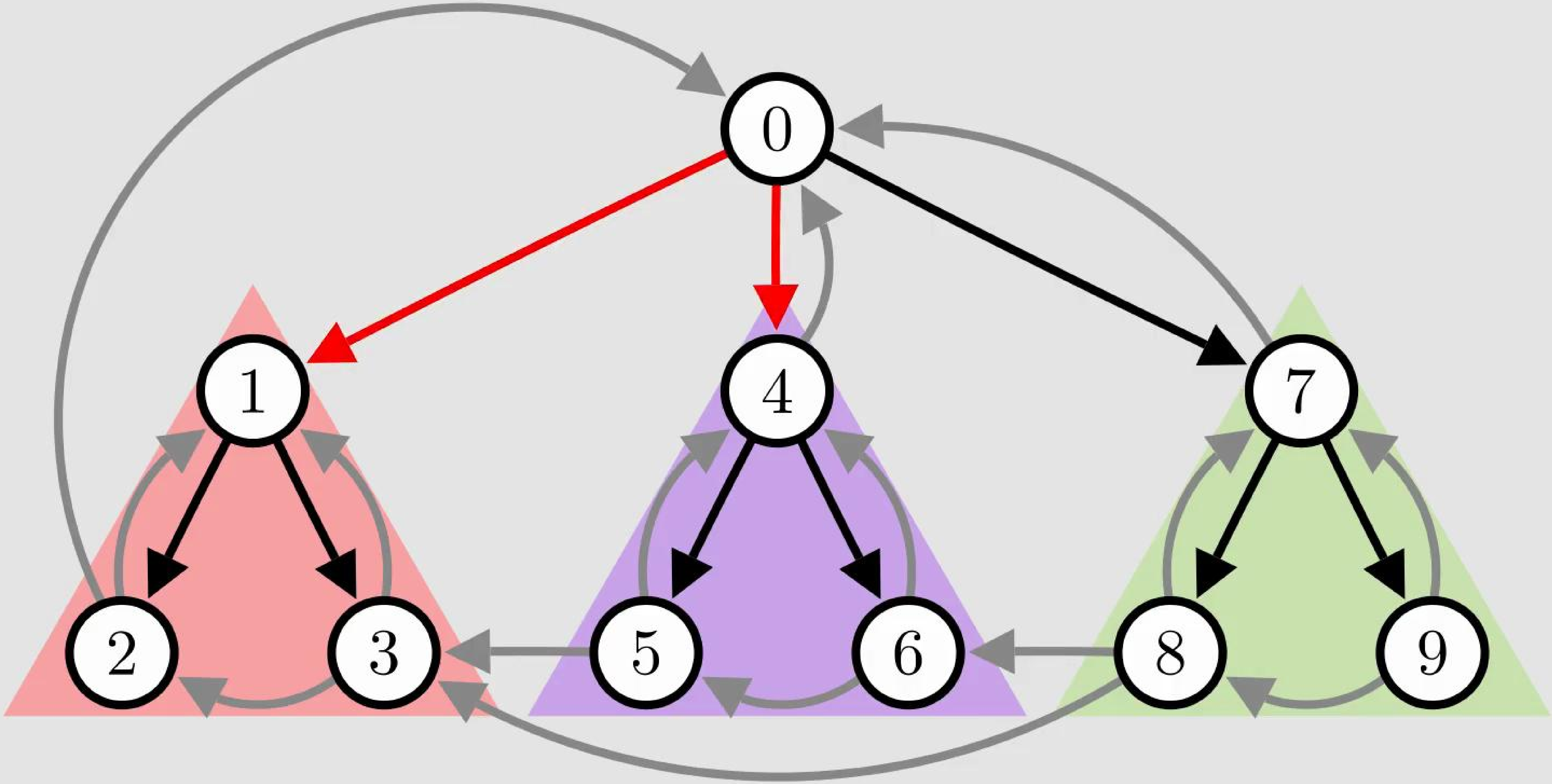
Further Propagation Rules



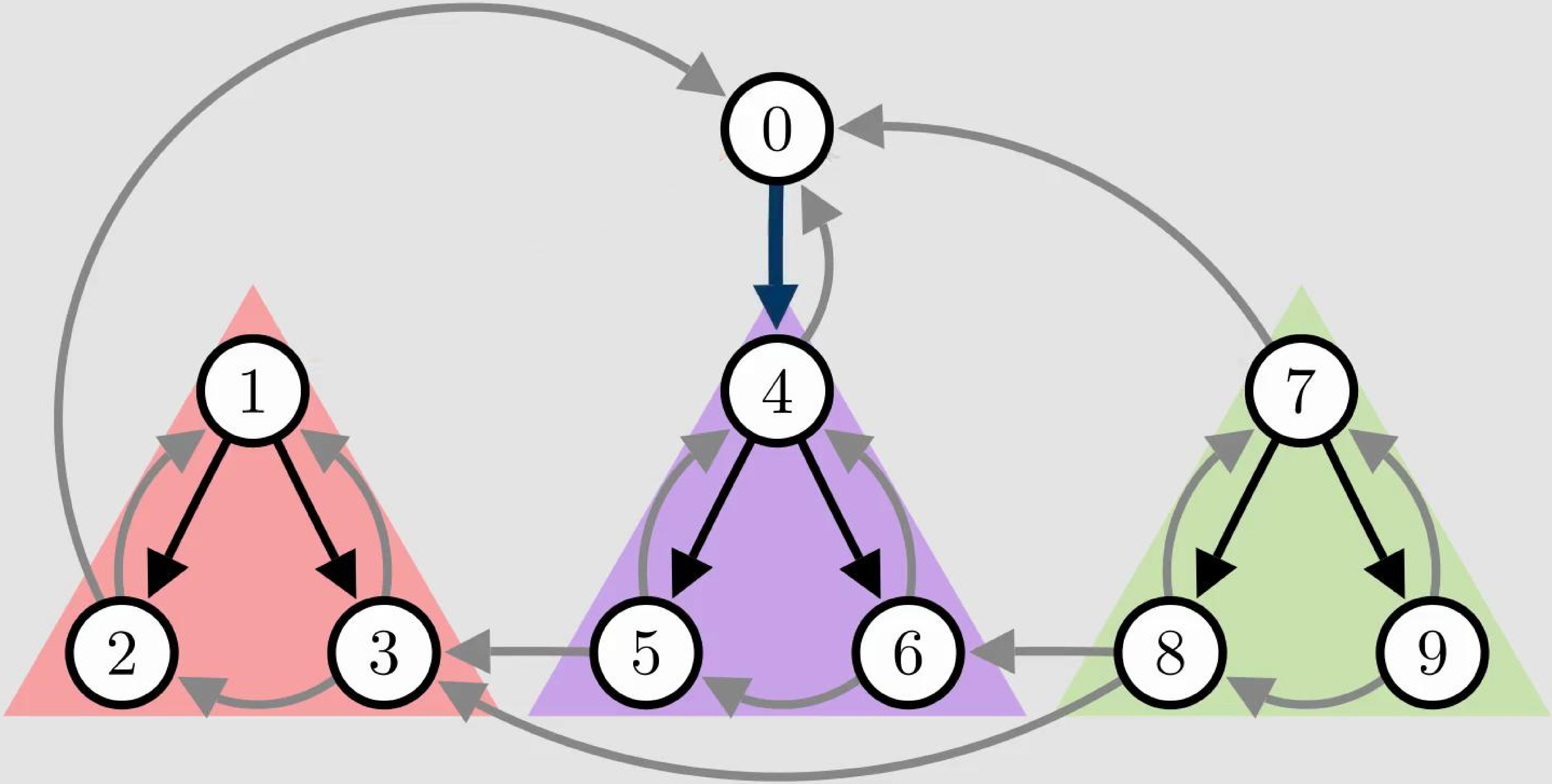
Further Propagation Rules



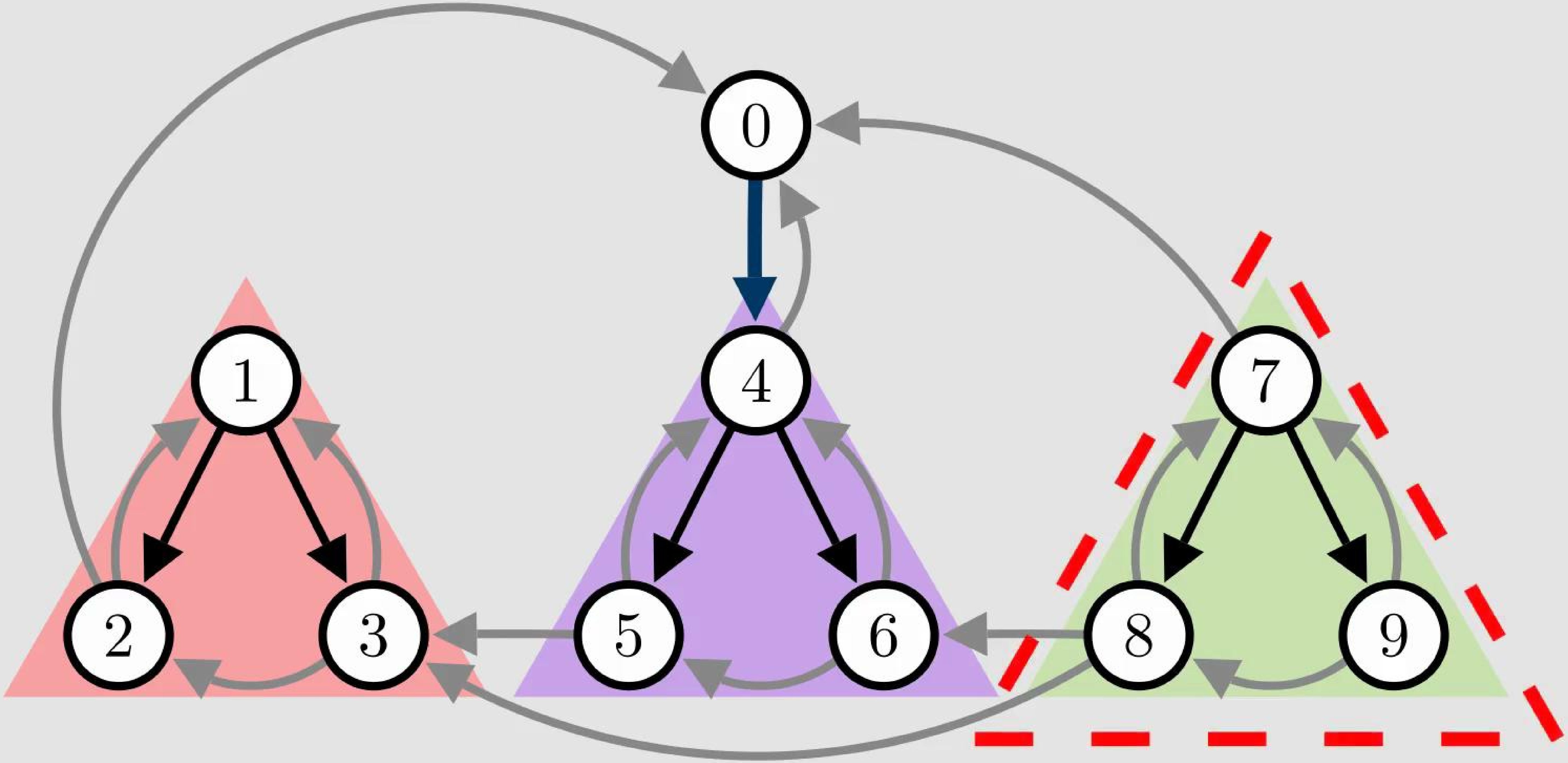
Further Propagation Rules: 'Prune Root'



Further Propagation Rules: 'Prune Root'

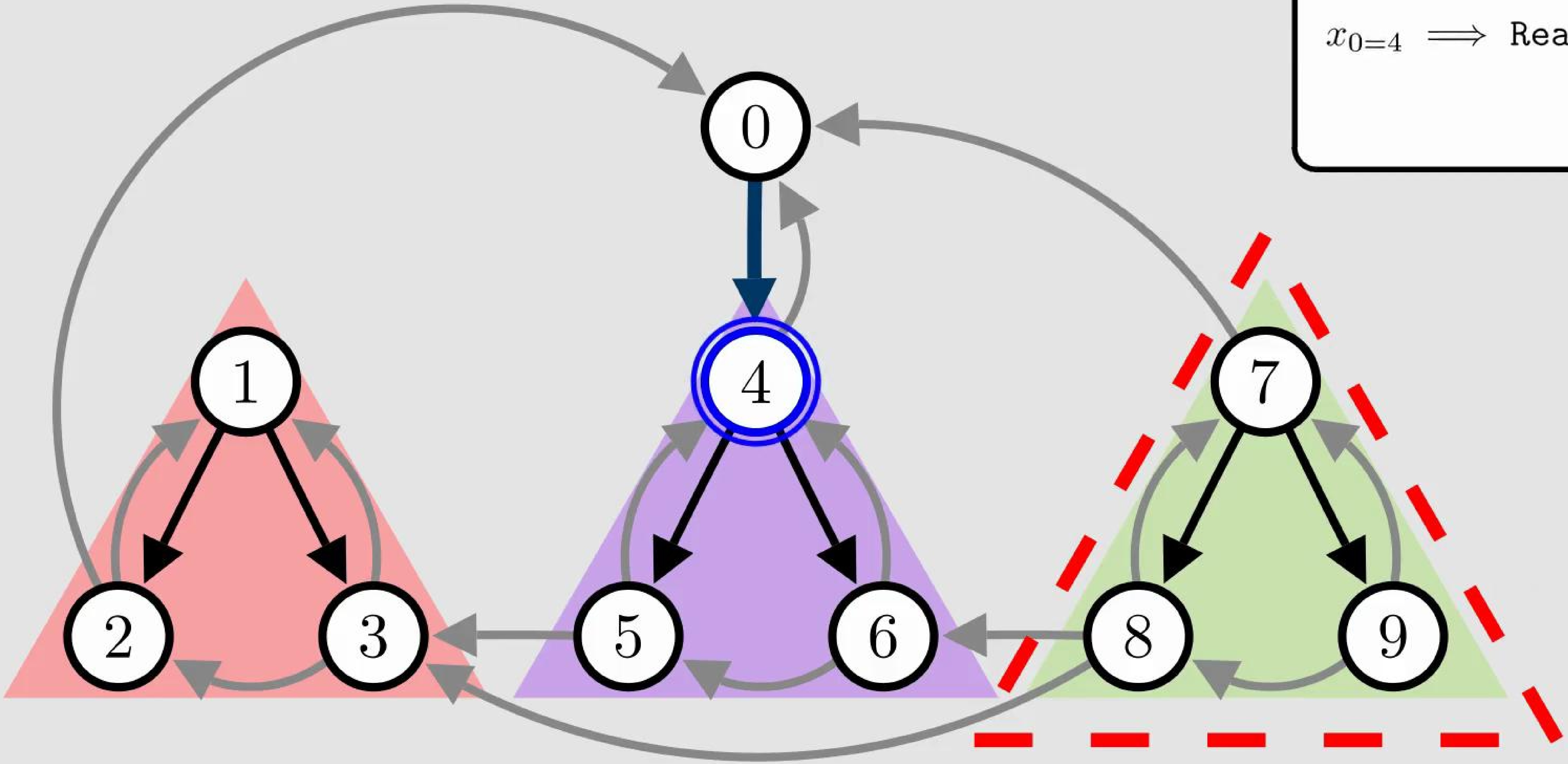


Further Propagation Rules: 'Prune Root'

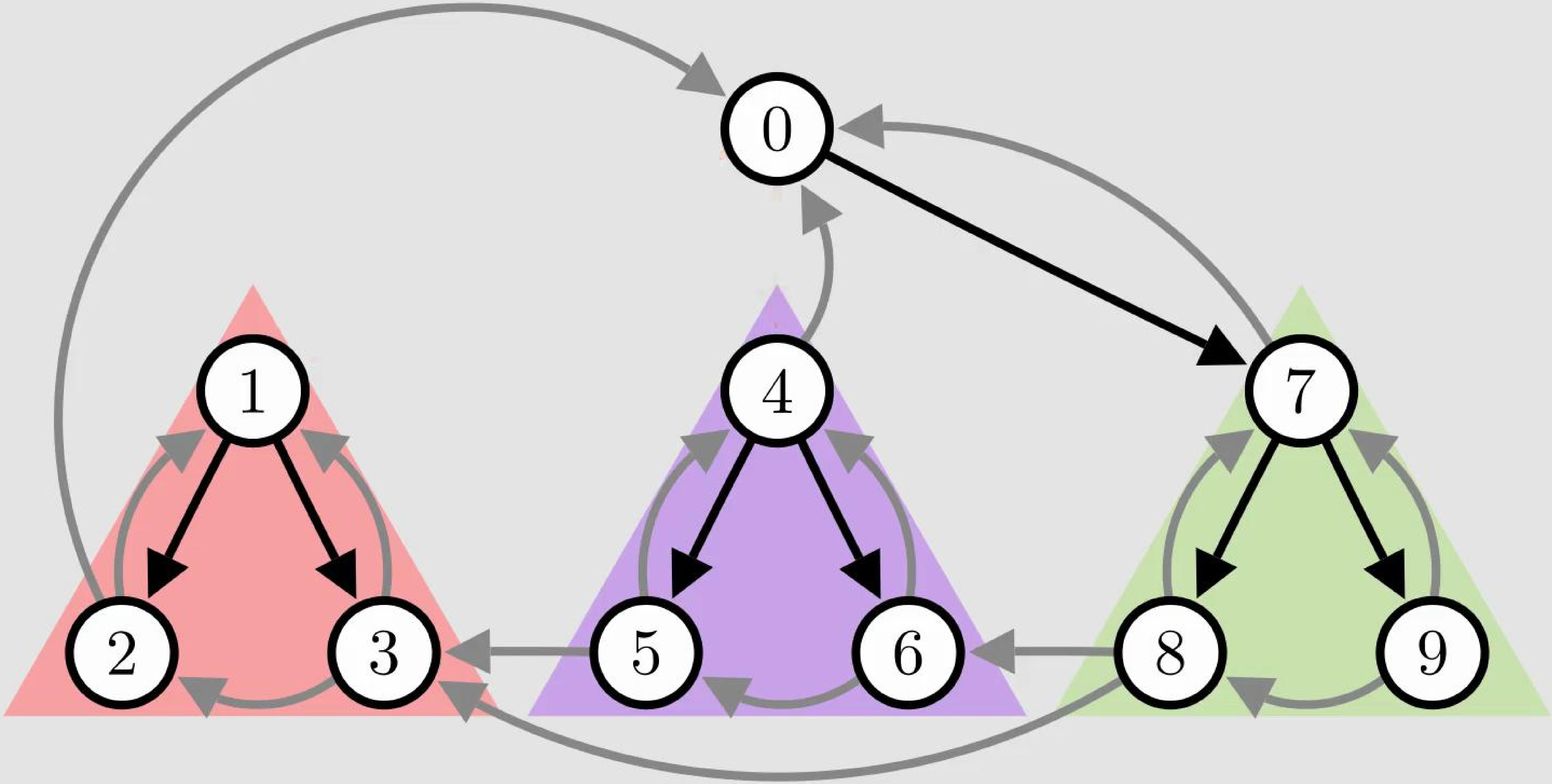


Further Propagation Rules: 'Prune Root'

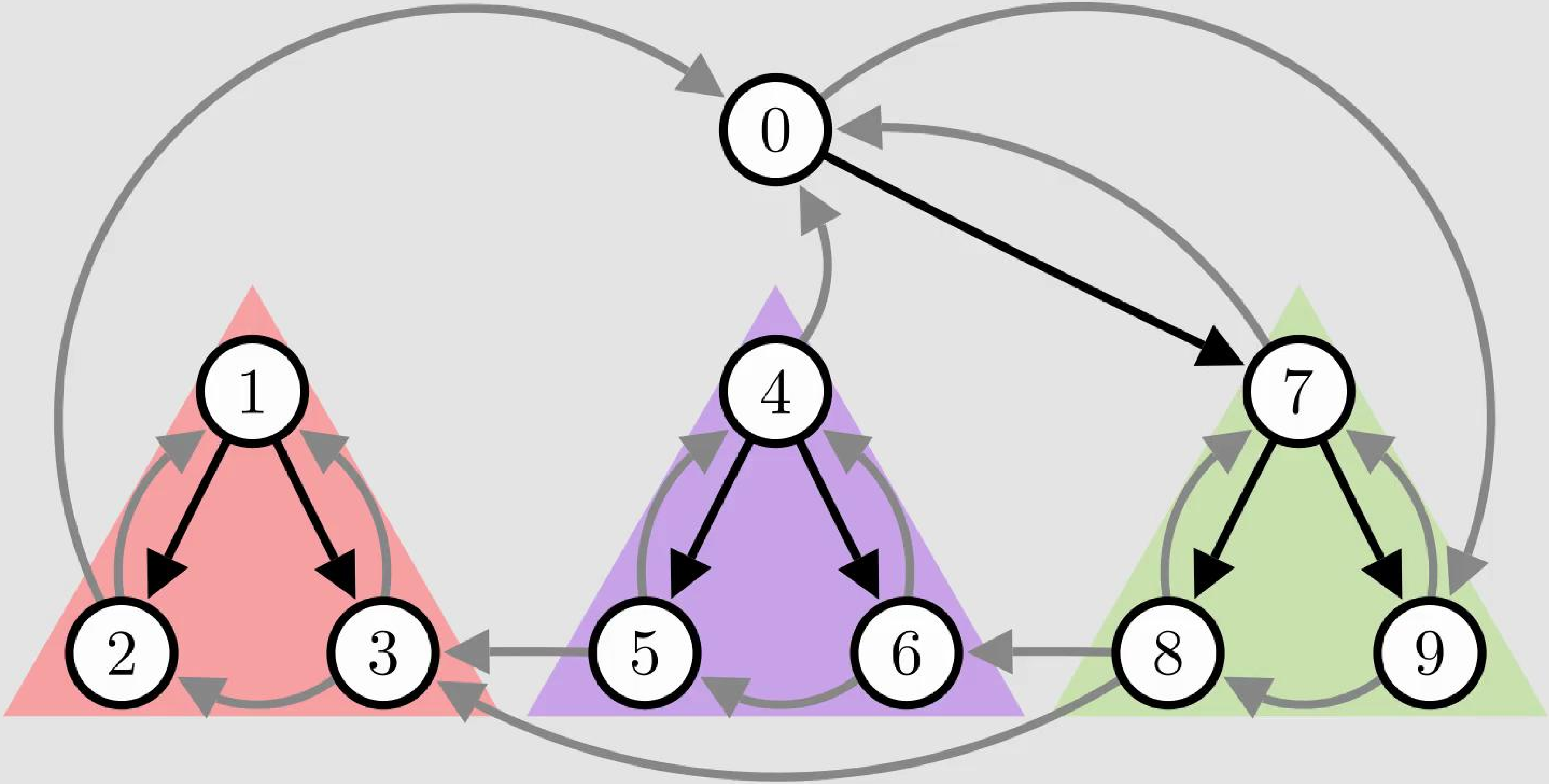
$x_{0=4} \implies$ `ReachTooSmall(4)`



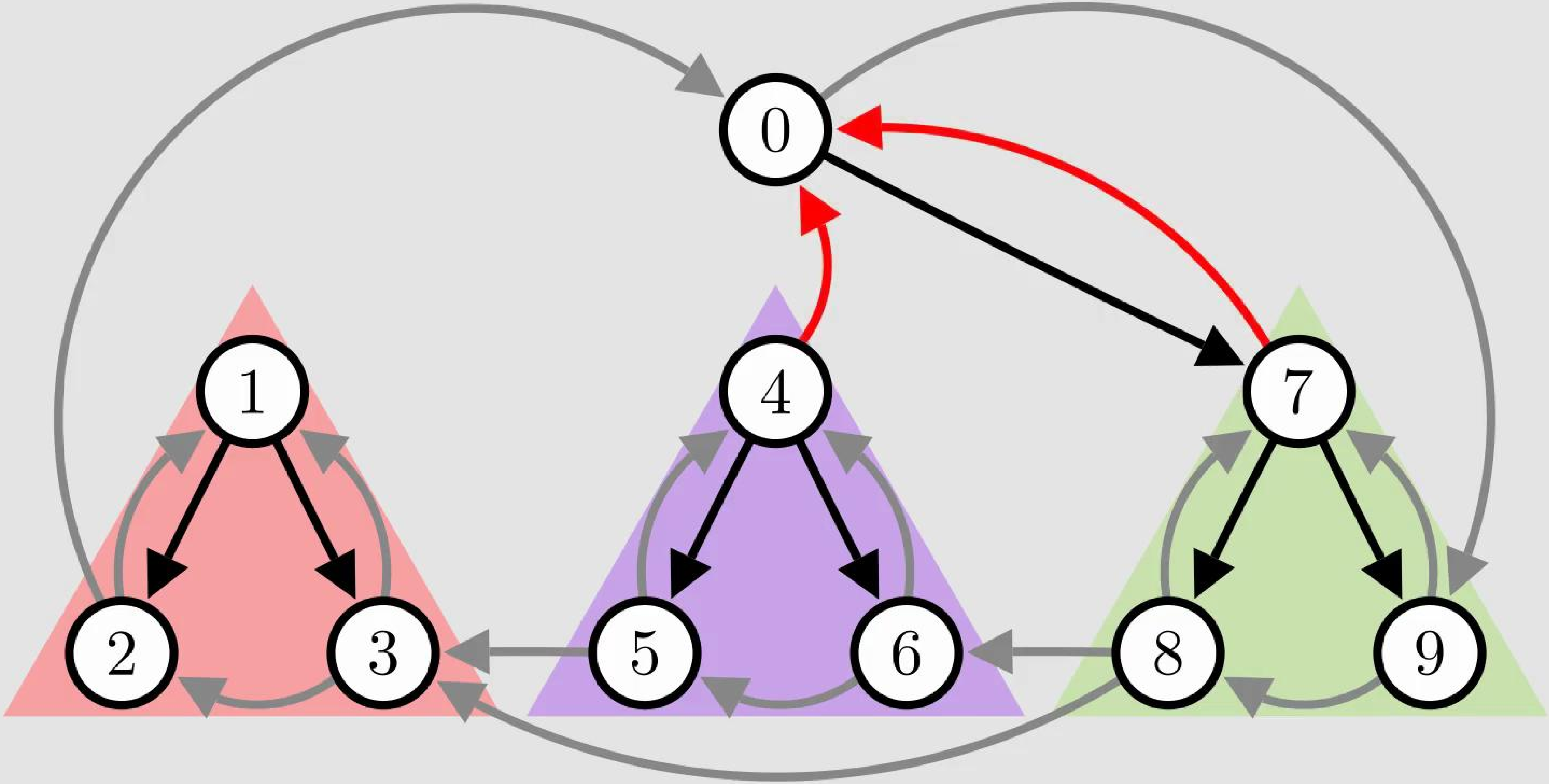
Further Propagation Rules: 'Prune Root'



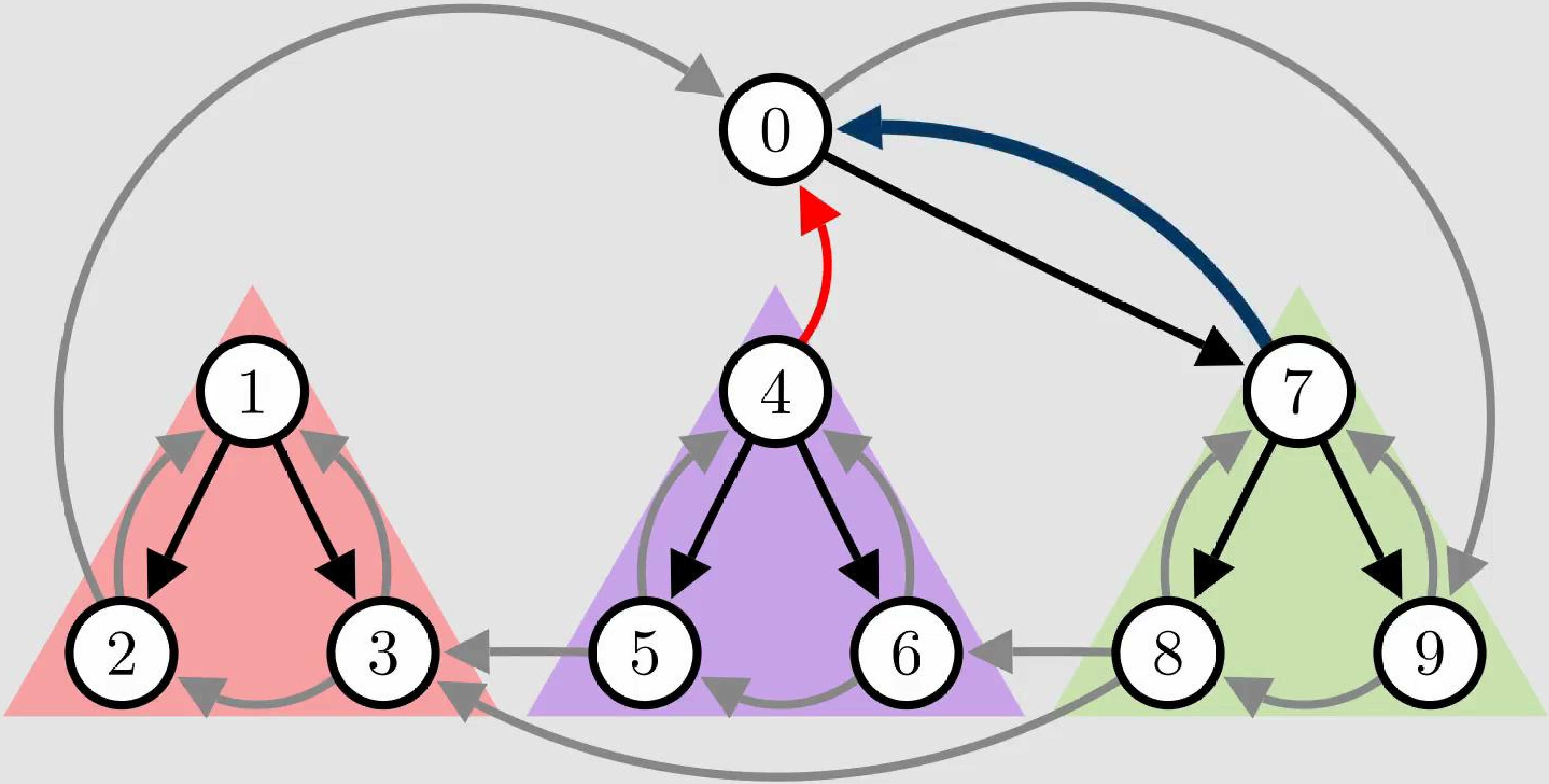
Further Propagation Rules: 'Prune Root'



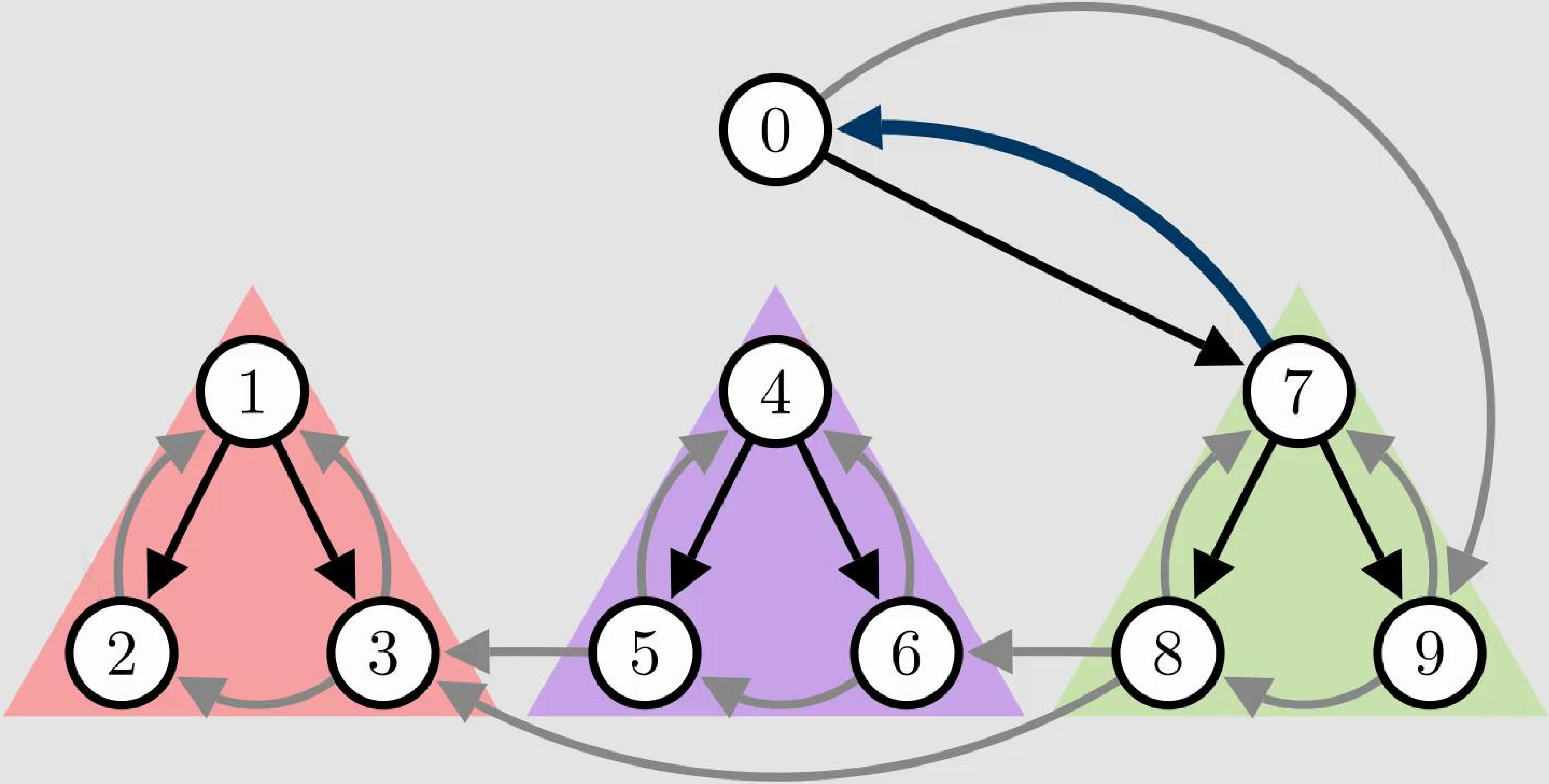
Further Propagation Rules: 'Prune Root'



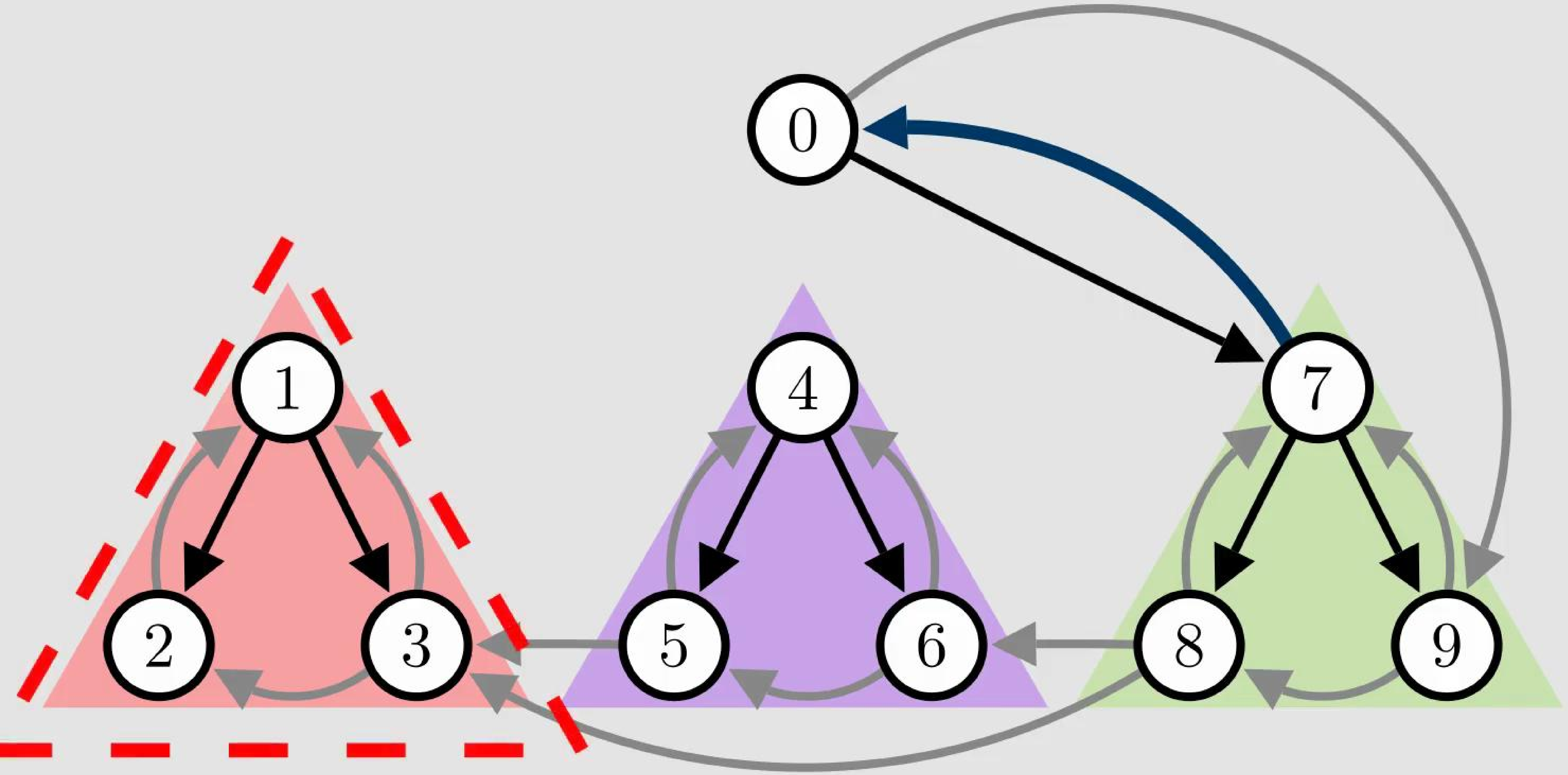
Further Propagation Rules: 'Prune Root'



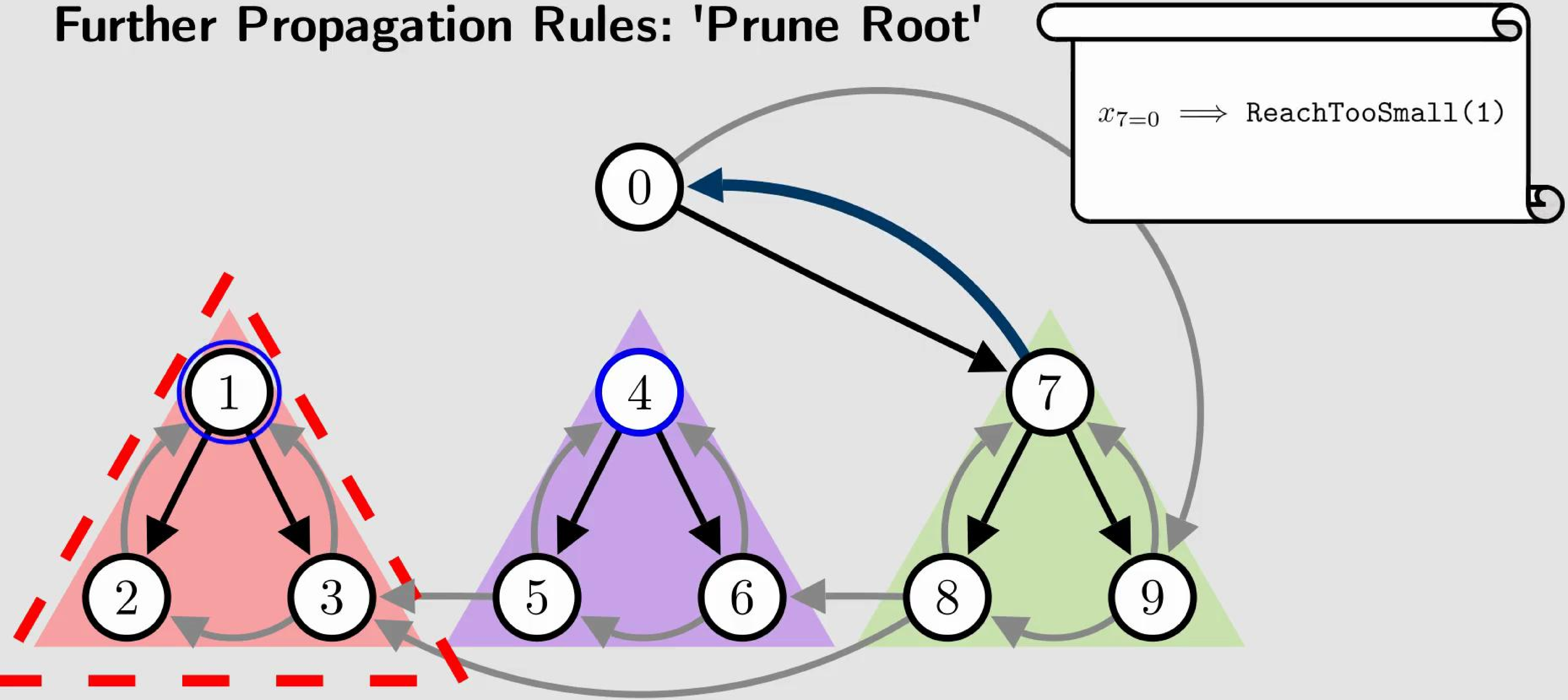
Further Propagation Rules: 'Prune Root'



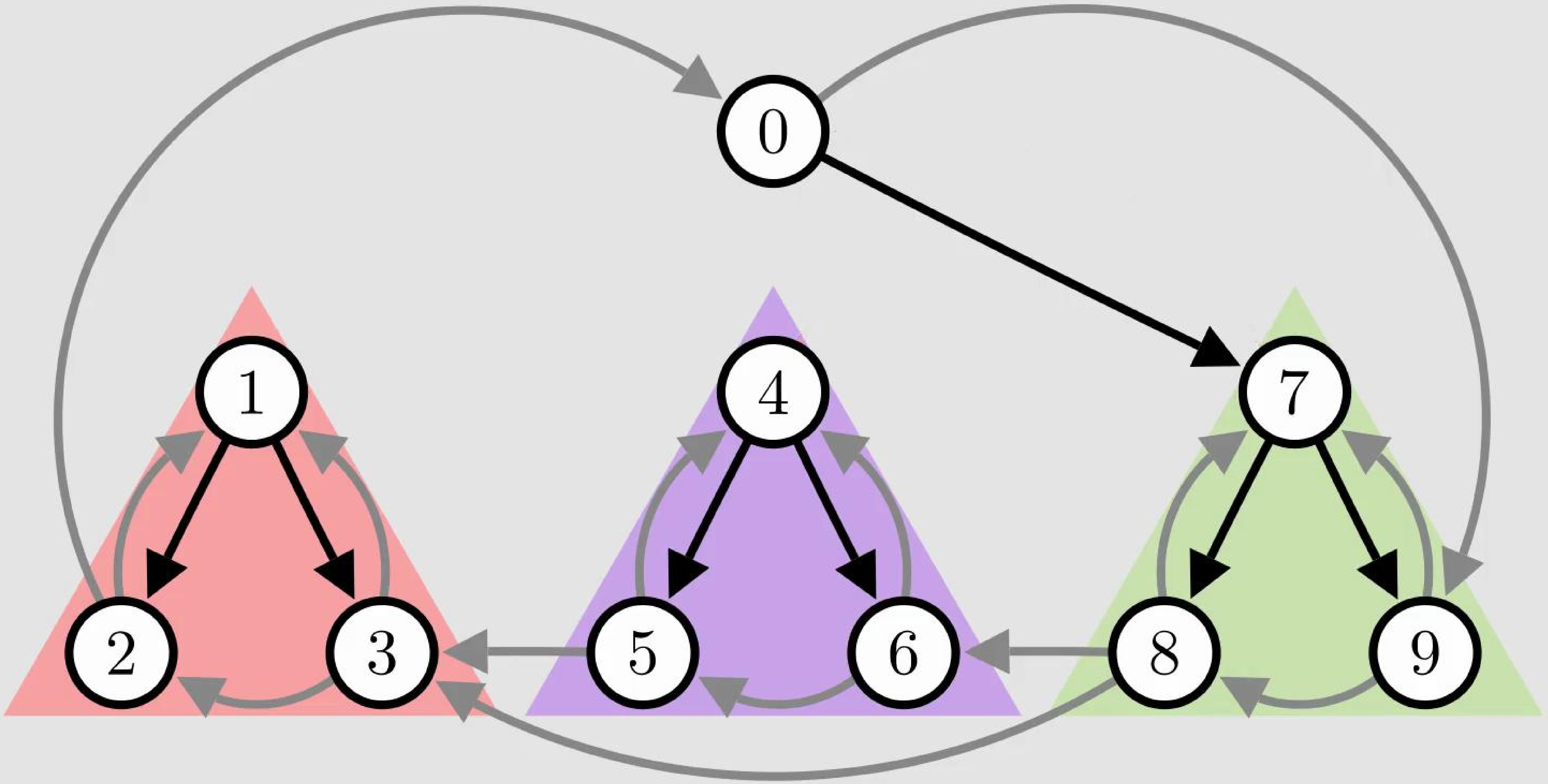
Further Propagation Rules: 'Prune Root'



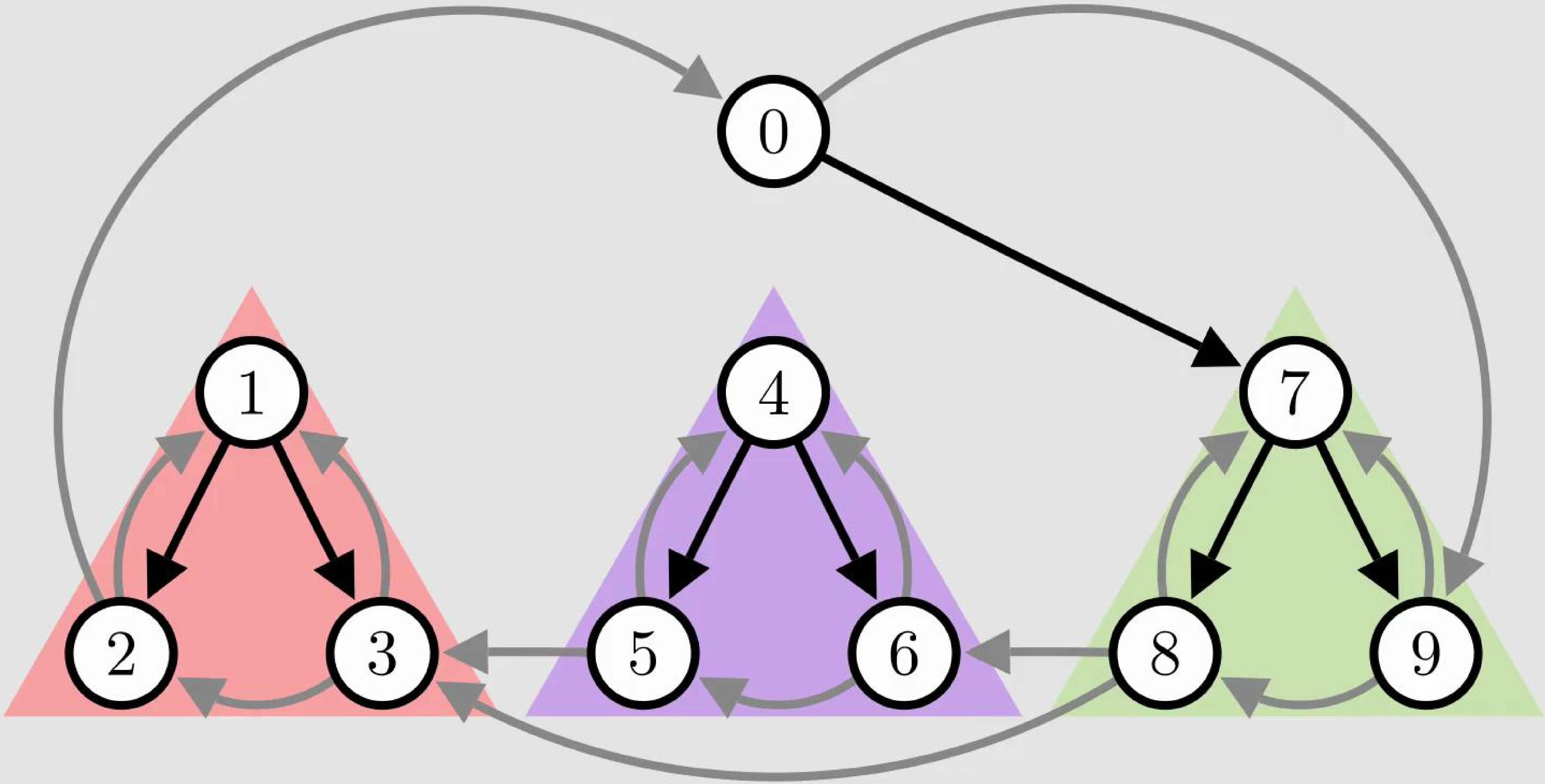
Further Propagation Rules: 'Prune Root'



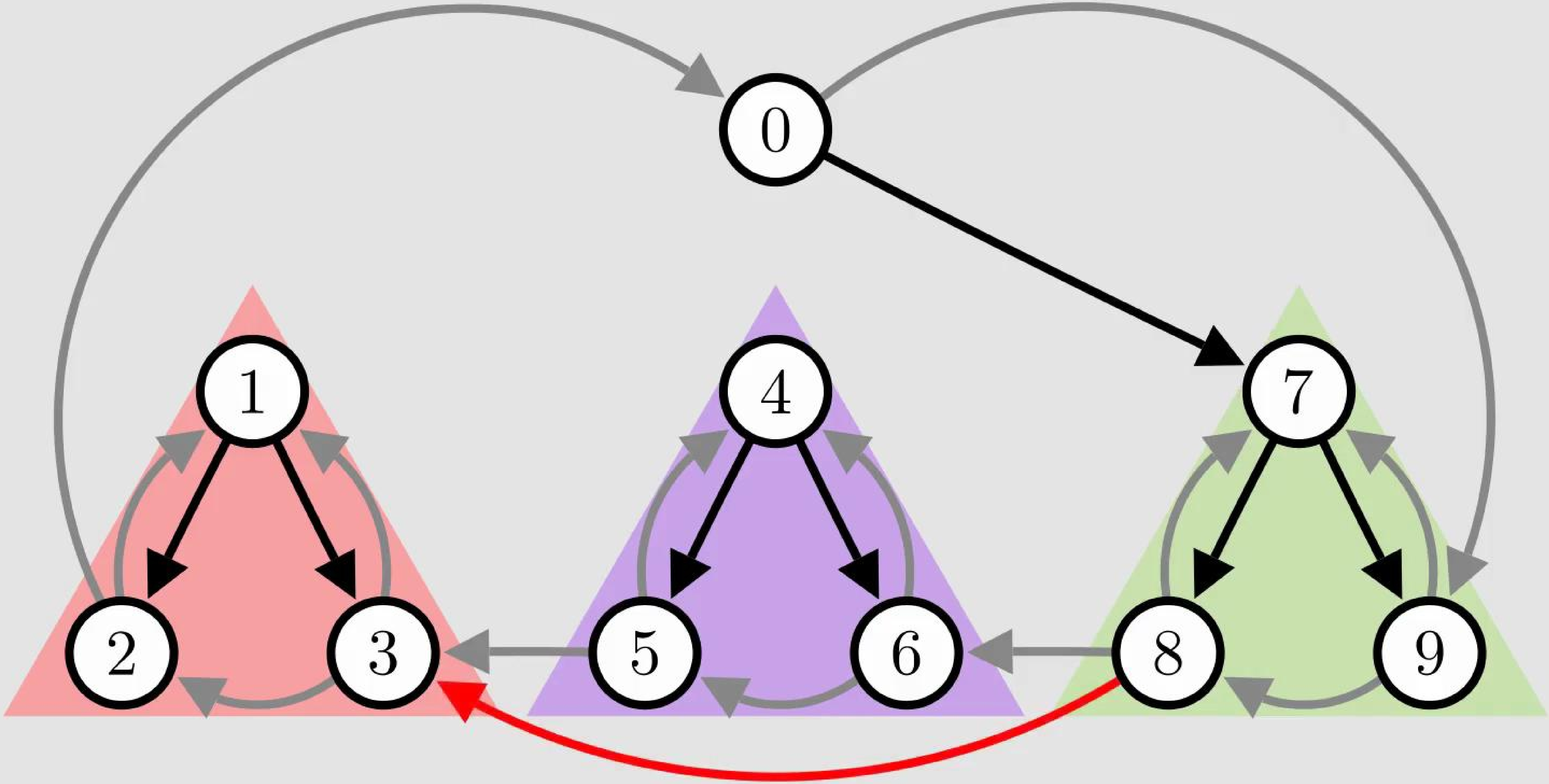
Further Propagation Rules: 'Prune Skip'



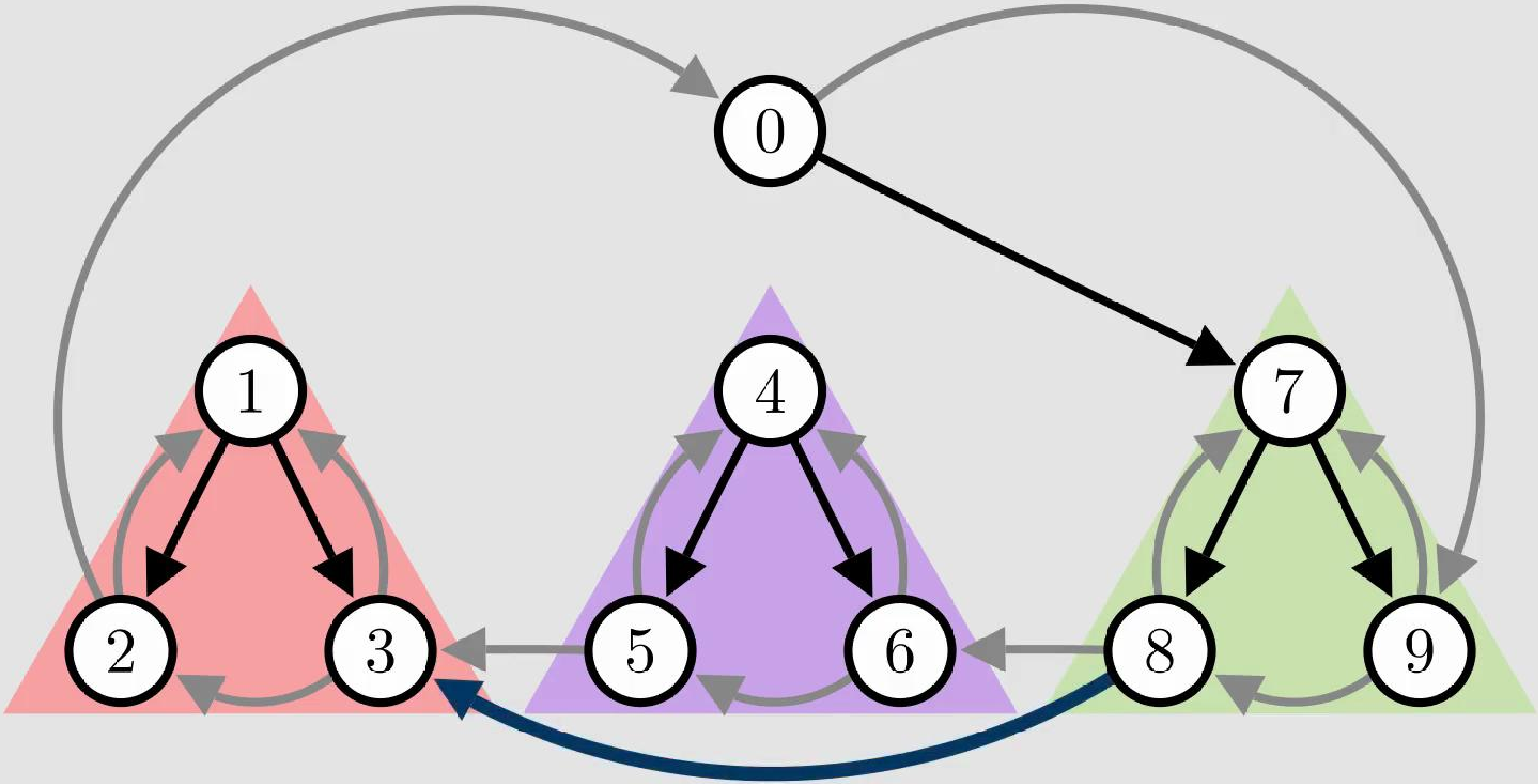
Further Propagation Rules: 'Prune Skip'



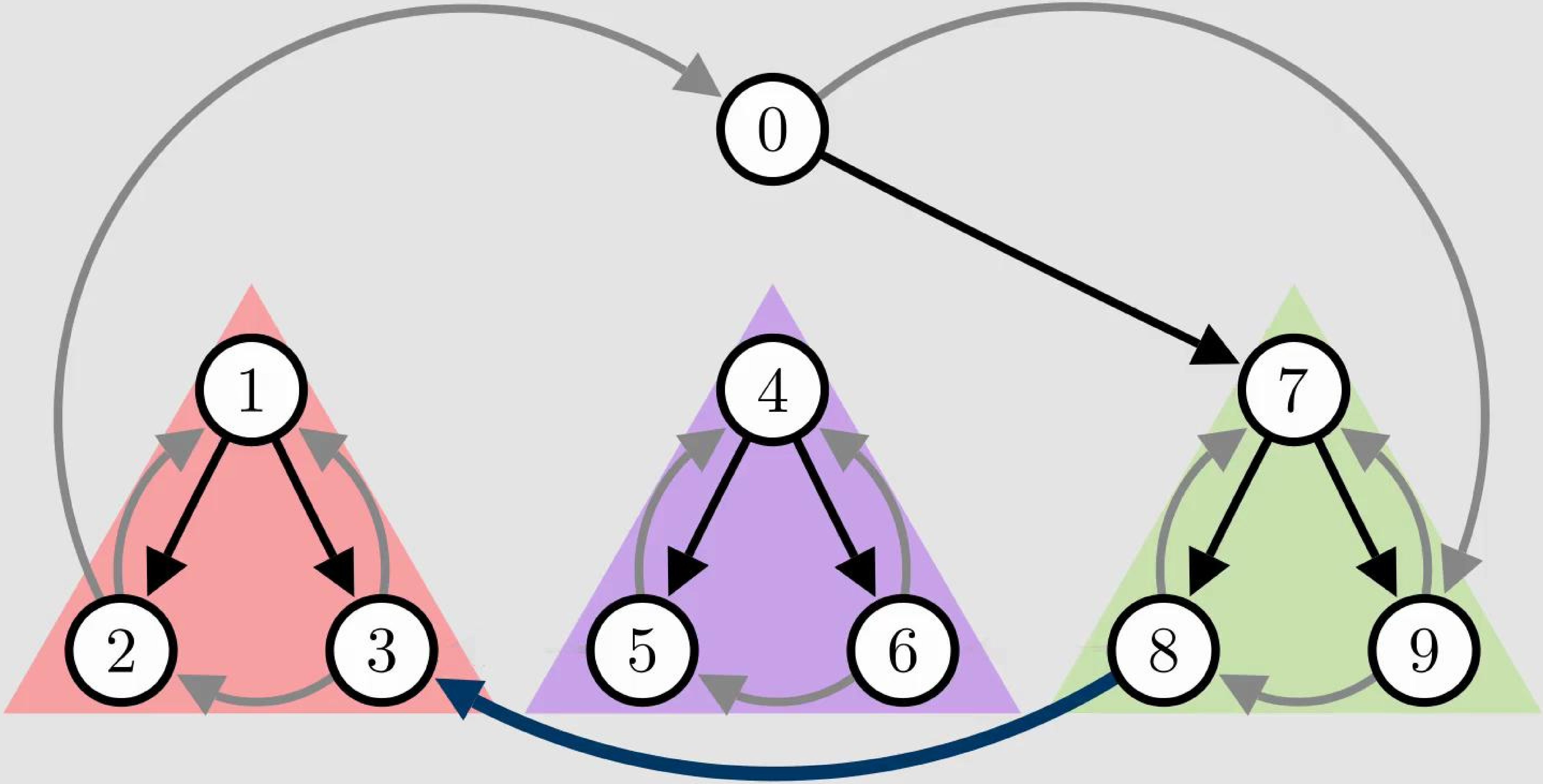
Further Propagation Rules: 'Prune Skip'



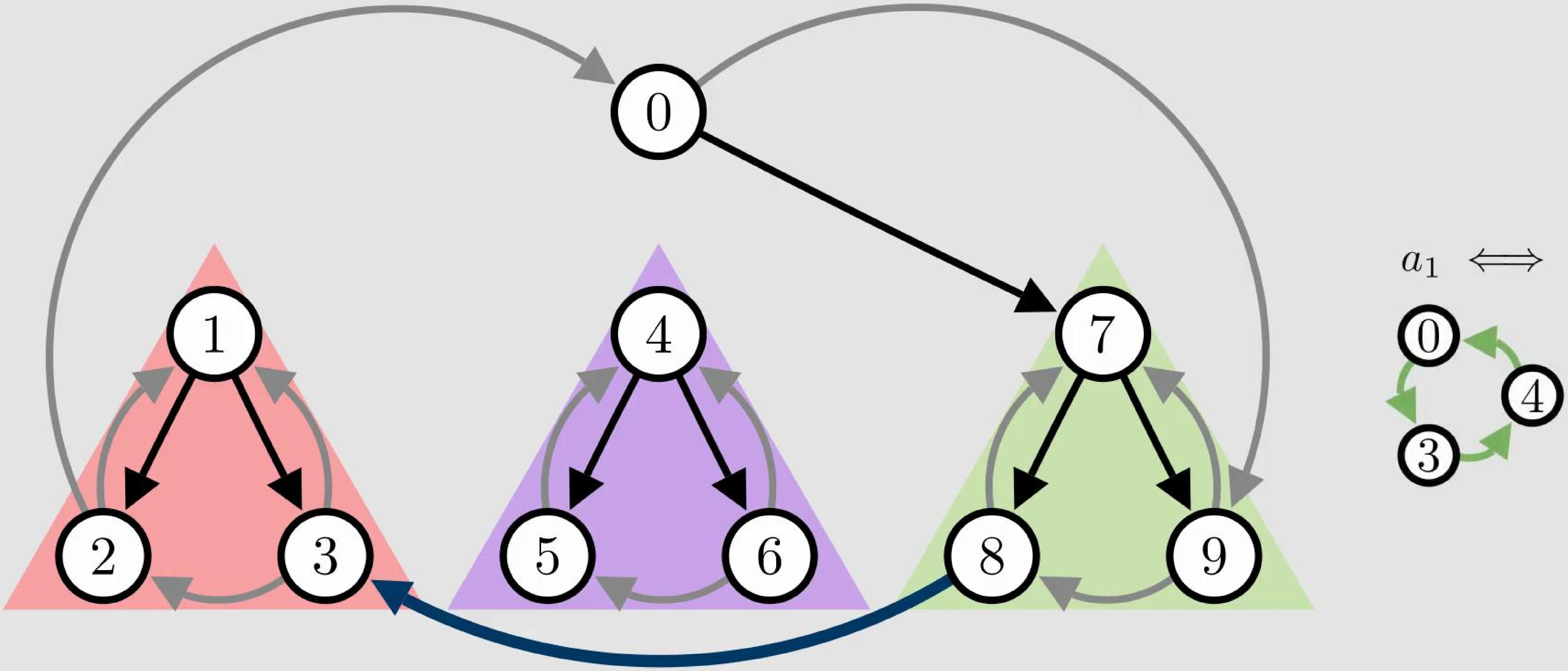
Further Propagation Rules: 'Prune Skip'



Further Propagation Rules: 'Prune Skip'

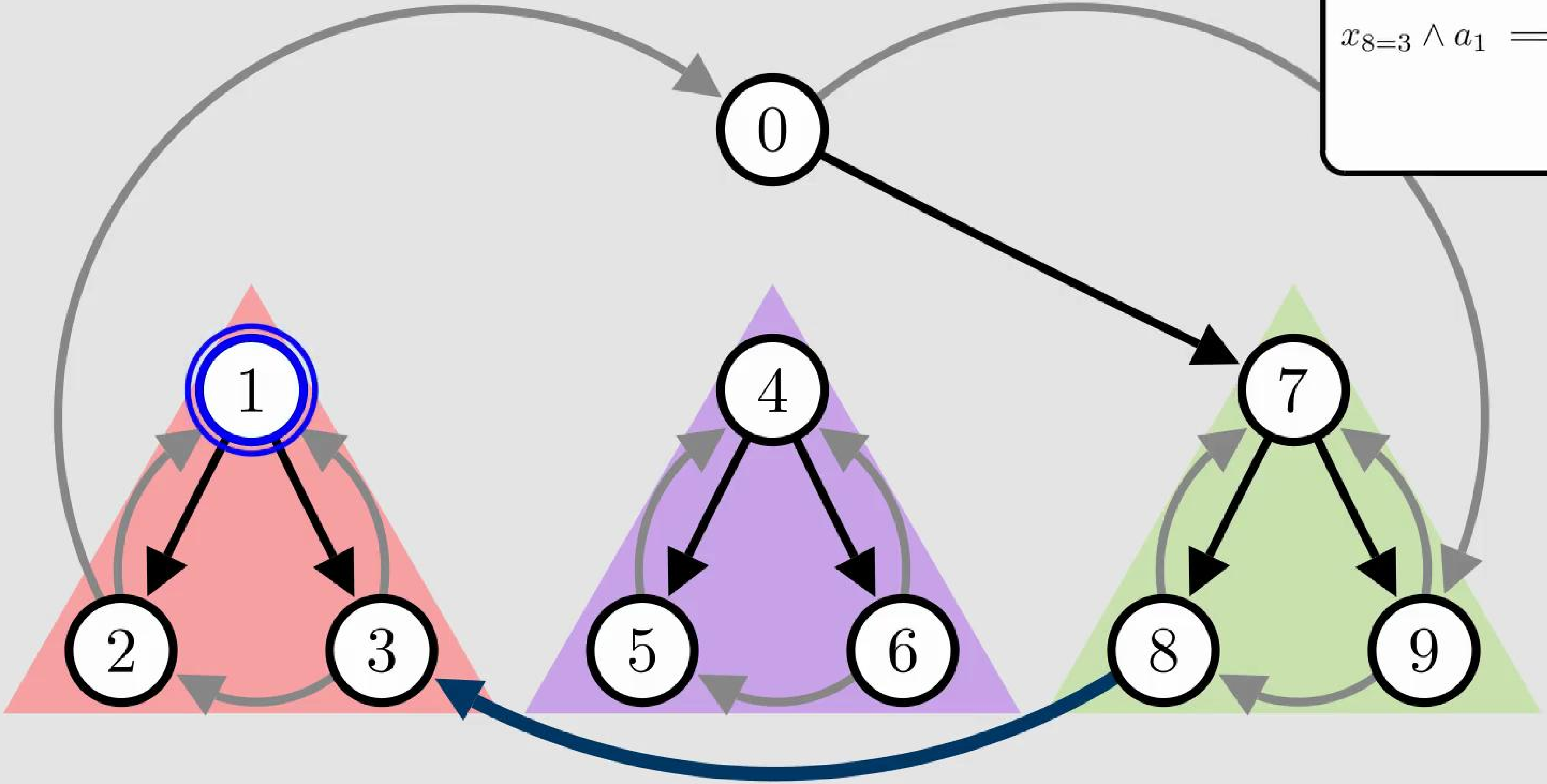


Further Propagation Rules: 'Prune Skip'

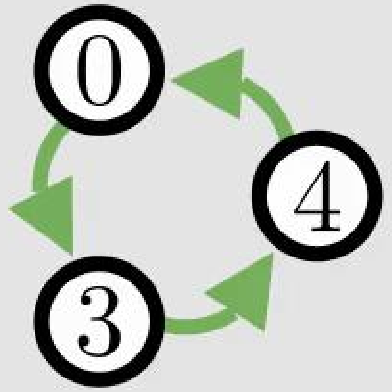


Further Propagation Rules: 'Prune Skip'

$x_{8=3} \wedge a_1 \implies$ `ReachTooSmall(1)`

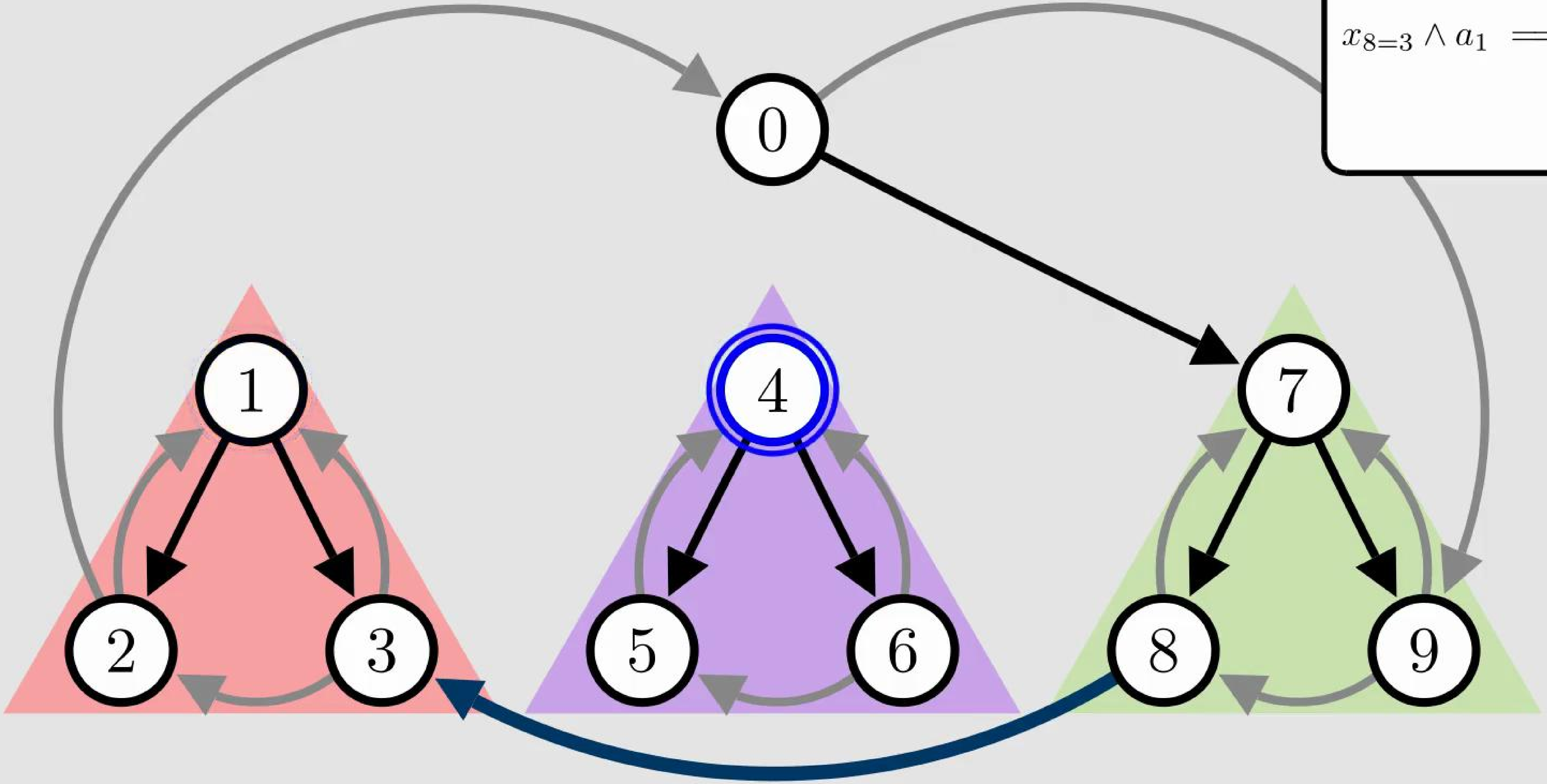


$a_1 \iff$

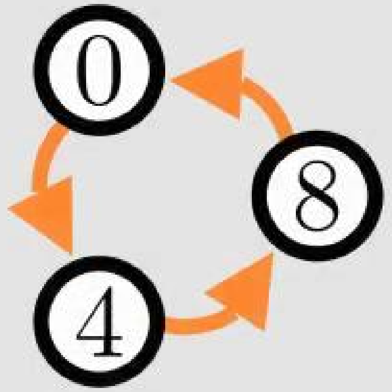


Further Propagation Rules: 'Prune Skip'

$x_{8=3} \wedge a_1 \implies$ ReachTooSmall(1)



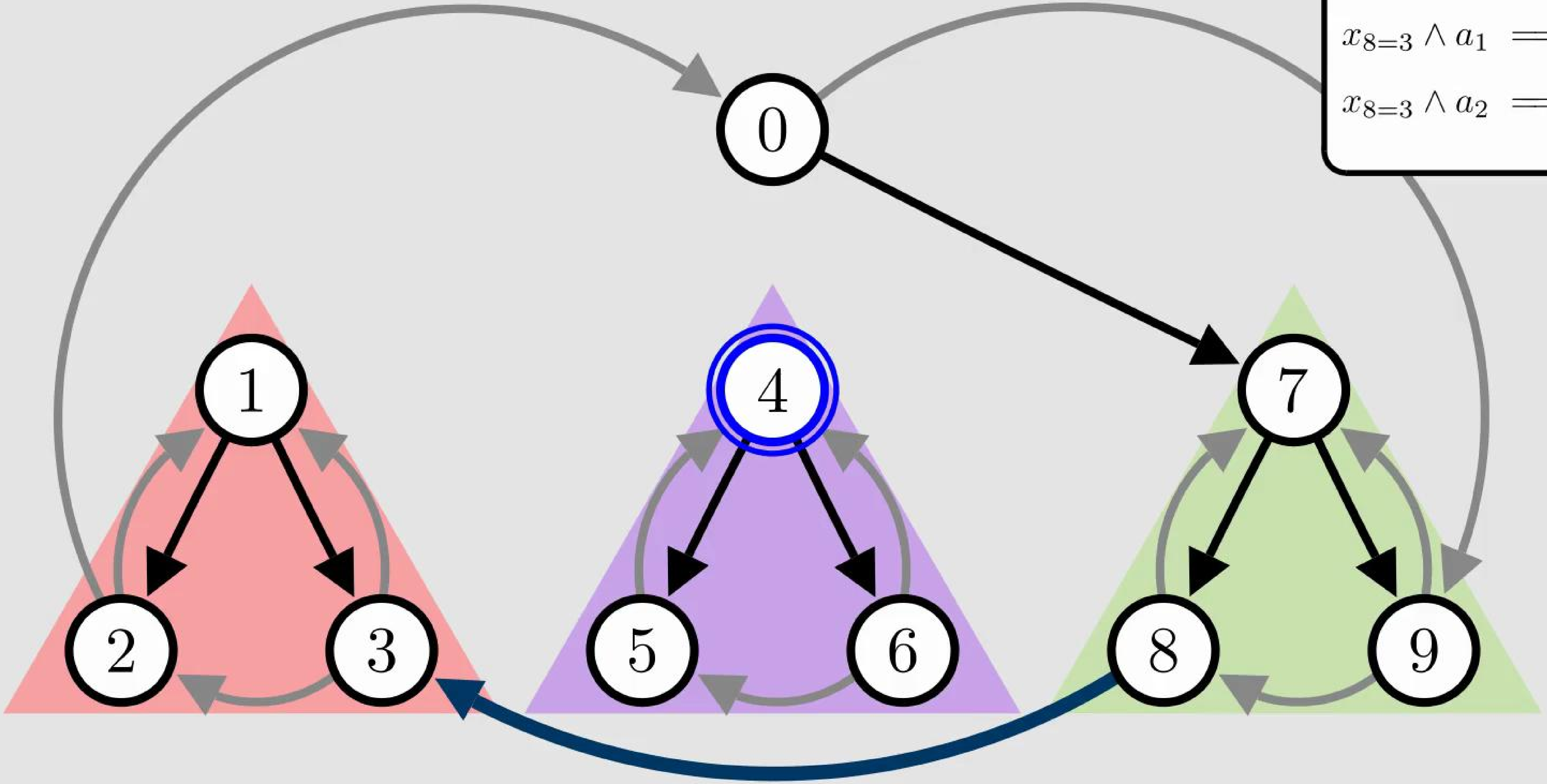
$a_2 \iff$



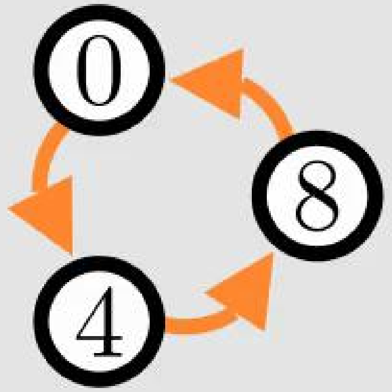
Further Propagation Rules: 'Prune Skip'

$x_{8=3} \wedge a_1 \implies$ ReachTooSmall(1)

$x_{8=3} \wedge a_2 \implies$ ReachTooSmall(4)



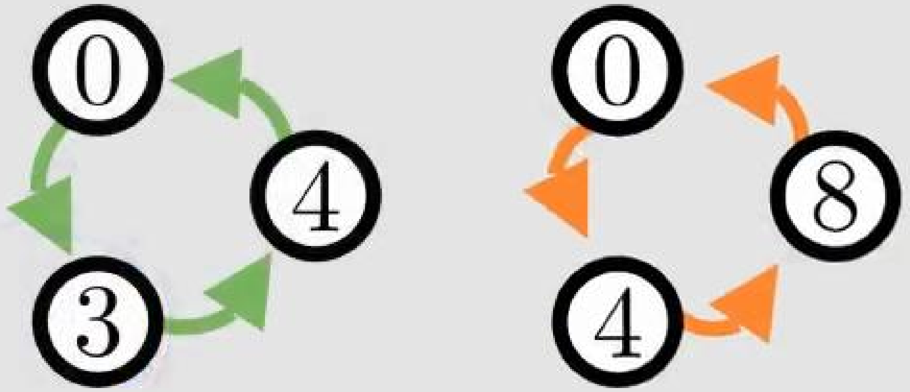
$a_2 \iff$



Further Propagation Rules: 'Prune Skip'

$x_{8=3} \wedge a_1 \implies$ `ReachTooSmall(1)`

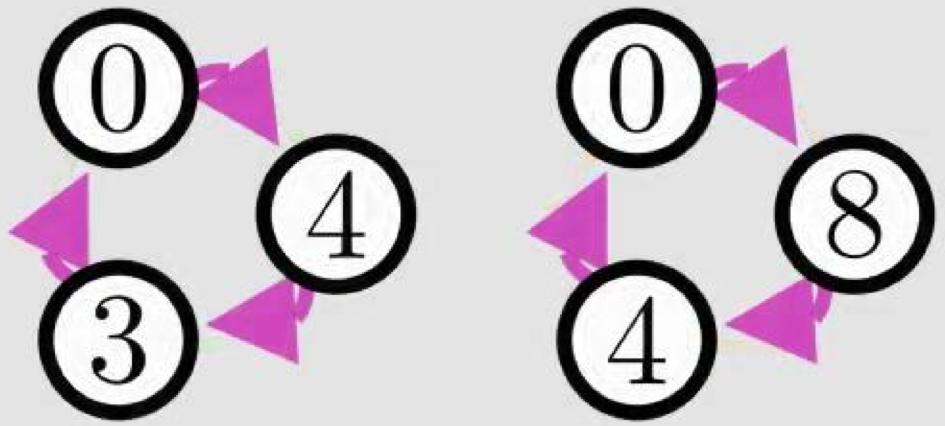
$x_{8=3} \wedge a_2 \implies$ `ReachTooSmall(4)`



Further Propagation Rules: 'Prune Skip'

$x_{8=3} \wedge a_1 \implies$ ReachTooSmall(1)

$x_{8=3} \wedge a_2 \implies$ ReachTooSmall(4)



Conclusions:

- Lots of complex reasoning is easy to capture with VeriPB/

Conclusions:

- Lots of complex reasoning is easy to capture with VeriPB/
- Proofs under implications / assumptions are quite powerful.

Conclusions:

- Lots of complex reasoning is easy to capture with VeriPB/
- Proofs under implications / assumptions are quite powerful.
- Not restricted by the kind of consistency enforced by CP propagators.

Conclusions:

- Lots of complex reasoning is easy to capture with VeriPB/
- Proofs under implications / assumptions are quite powerful.
- Not restricted by the kind of consistency enforced by CP propagators.
- Can confirm the power of proof logging as a debugging tool.

Future work:

- Many more propagators to do :-D

Future work:

- Many more propagators to do :-D
- Regular -> Cost Regular, MDD

Future work:

- Many more propagators to do :-D
- Regular -> Cost Regular, MDD
- Circuit -> Subcircuit, Path

Future work:

- Many more propagators to do :-D
- Regular -> Cost Regular, MDD
- Circuit -> Subcircuit, Path
- Also, other kinds of consistency: can chat about bounds-consistent multiplication.