Safety Problems are NP-complete for Flat Integer Programs with Octagonal Loops

Marius Bozga, Radu Iosif (Verimag/CNRS, France)
Filip Konecny (EPFL, Switzerland)

Motivation

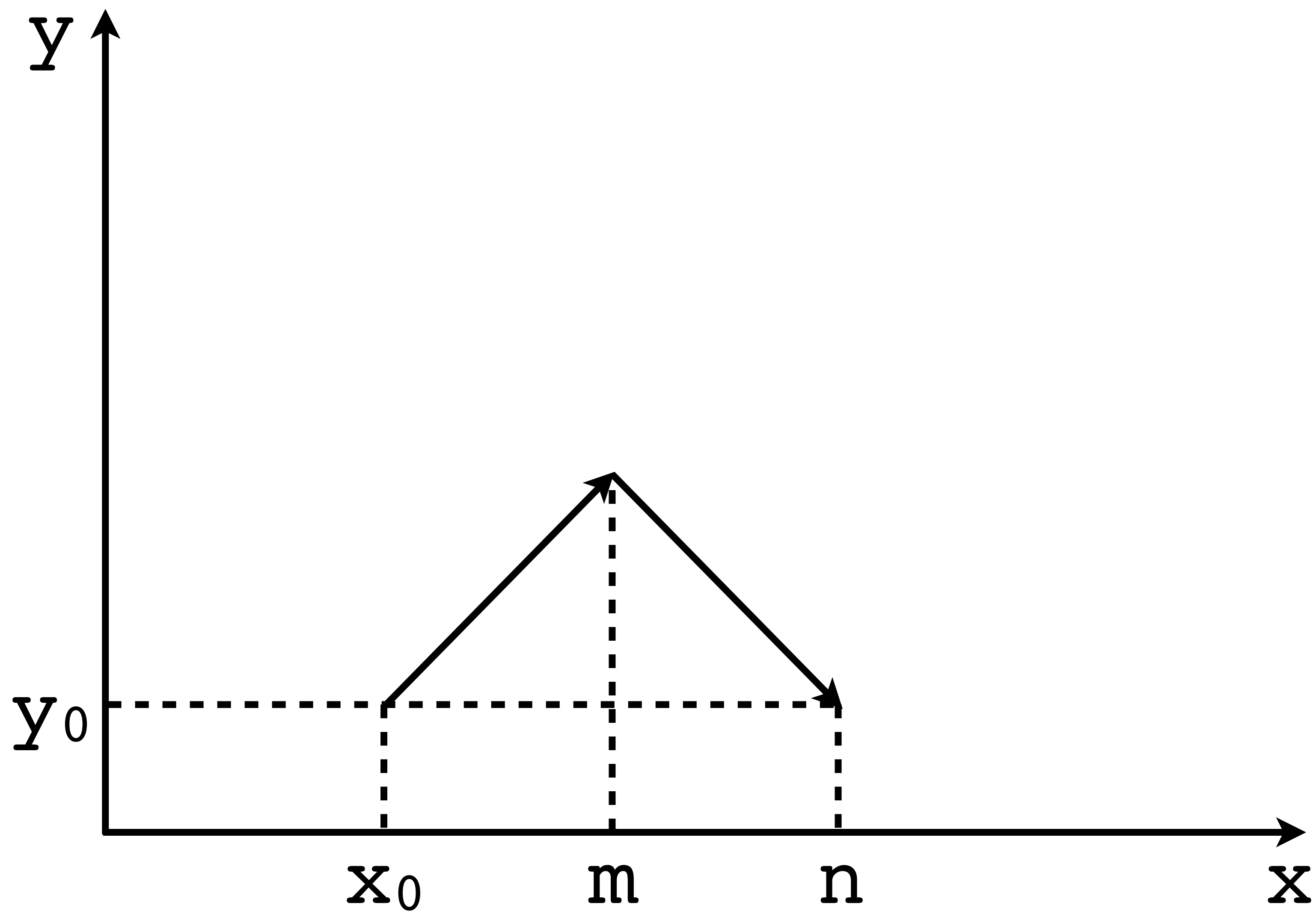
- Infinite state systems are, in general, undecidable
- Few complexity results for the decidable cases:
  - ➡ VAS coverage (EXPSPACE-complete) [Rackoff 1978]
  - ➡ inequivalence of reversal-bounded CM (NP-complete) [Ibarra, Gurari 1981]
  - ➡ gap-order constraints (PSPACE-complete) [Bozzelli, Pinchinat 2012]
- Efficient algorithm for flat integer programs with difference bounds and octagonal loops [BIK'10]
  - ➡ worst case EXPTIME, yet good average performance
- NP-completeness explains the behavior of our algorithm
  - ➡ educated guessing may solve NP-complete problems efficiently

Flat Integer Programs

```
int y = y0;
int n = 2*m - x;

while (x < n) {
  if (x < m) {
    x ++;
    y ++;
  } else {
    x ++;
    y --;
}}

assert(y == y0);
```

Flat Integer Programs

```
int y = y0;
int n = 2*m - x;

while (x < n) {
  if (x < m) {
    x ++;
    y ++;
  } else {
    x ++;
    y --;
}}

assert(y == y0);
```

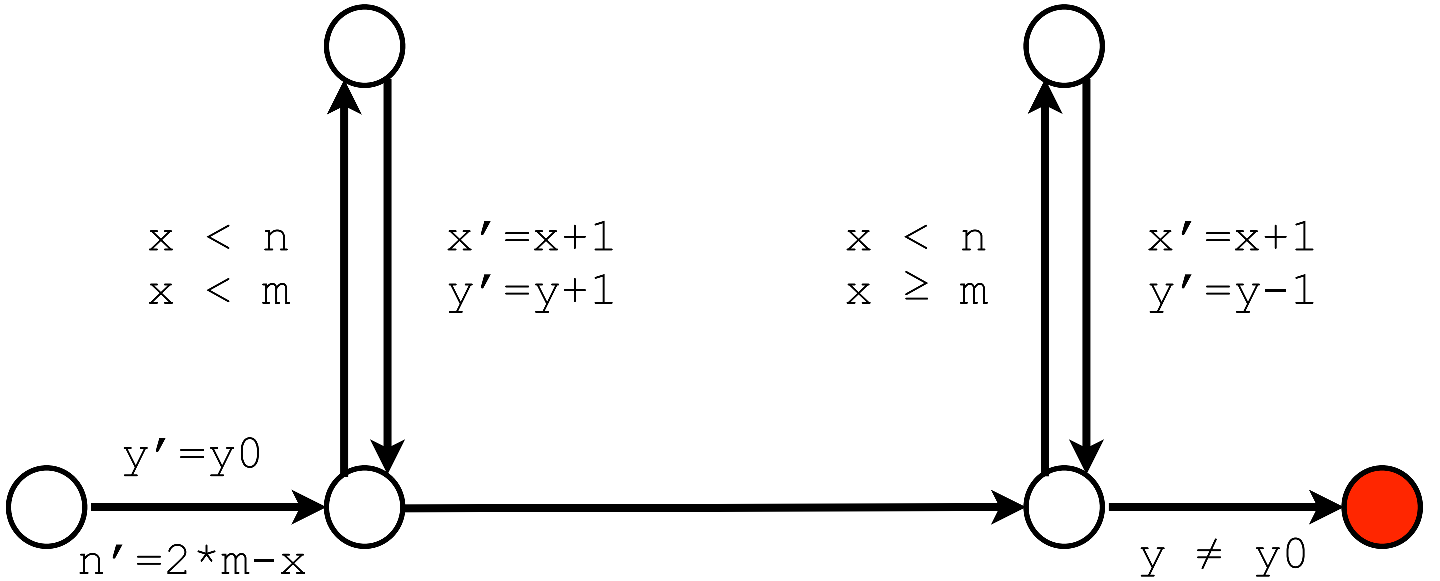


Flat Integer Programs

```
int y = y0;
int n = 2*m - x;

while (x < n) {
  if (x < m) {
    x ++;
    y ++;
  } else {
    x ++;
    y --;
}}

assert(y == y0);
```

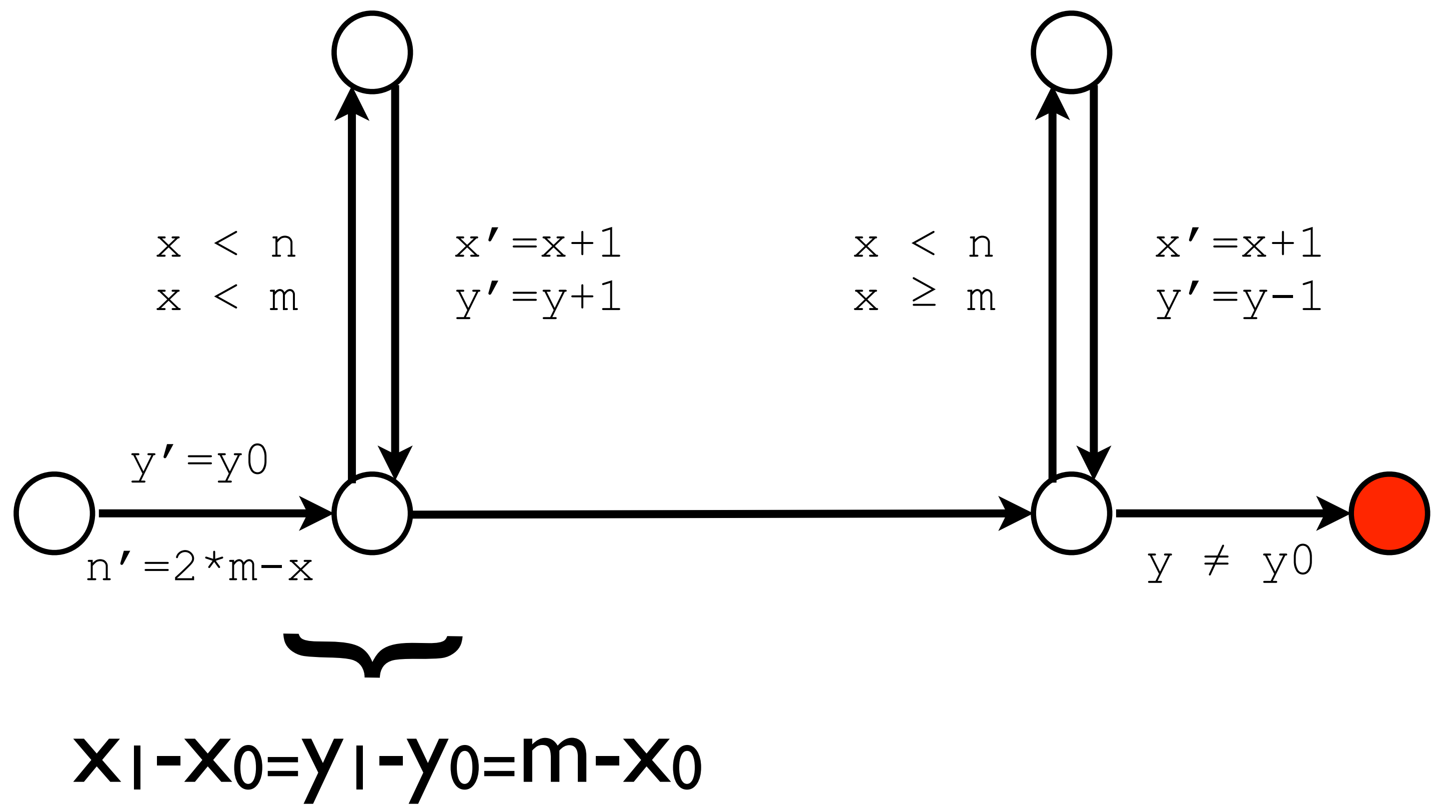
Flat Integer Programs

```
int y = y0;
int n = 2*m - x;

while (x < n) {
  if (x < m) {
    x ++;
    y ++;
  } else {
    x ++;
    y --;
}}

assert(y == y0);
```



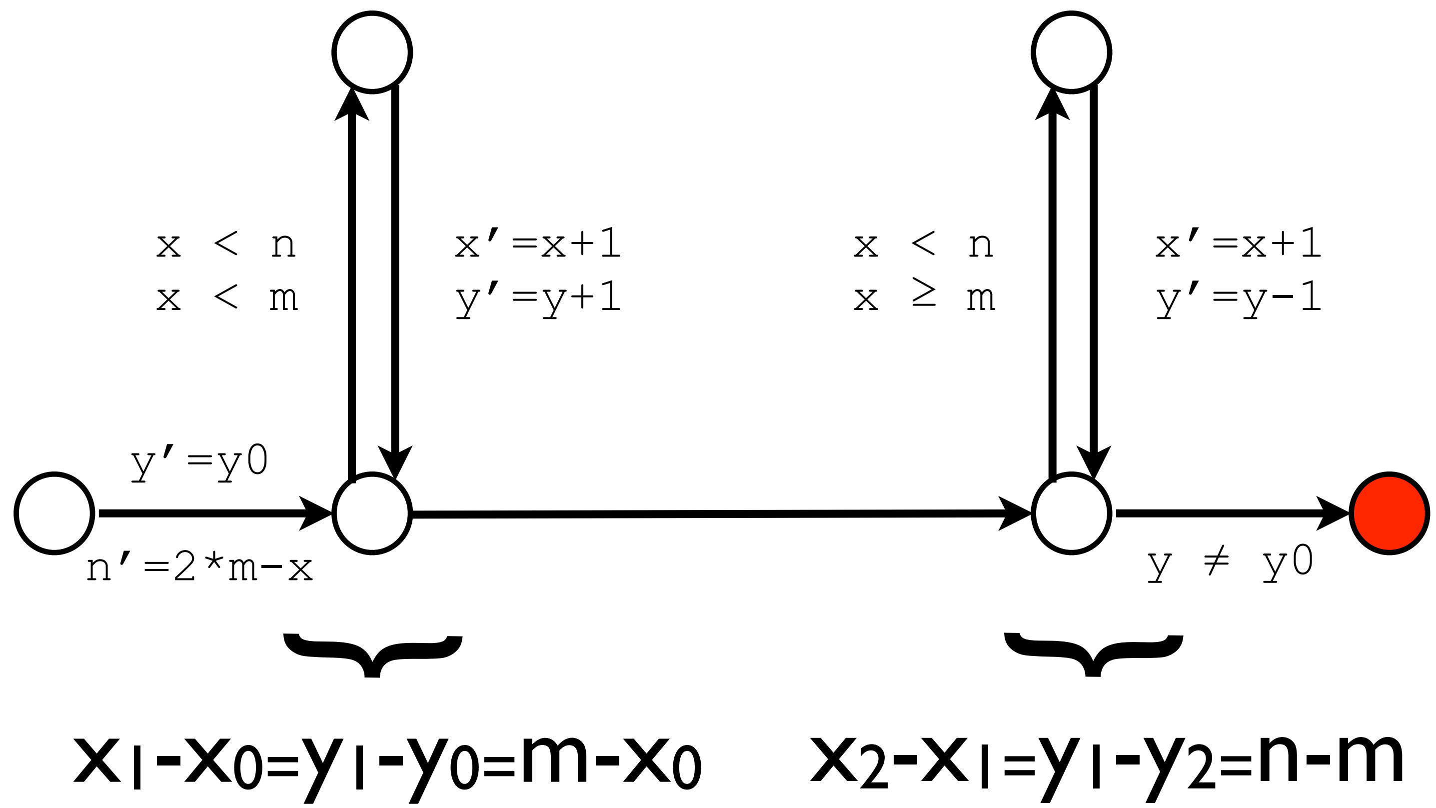
Flat Integer Programs

```
int y = y0;
int n = 2*m - x;

while (x < n) {
  if (x < m) {
    x ++;
    y ++;
  } else {
    x ++;
    y --;
}}

assert(y == y0);
```



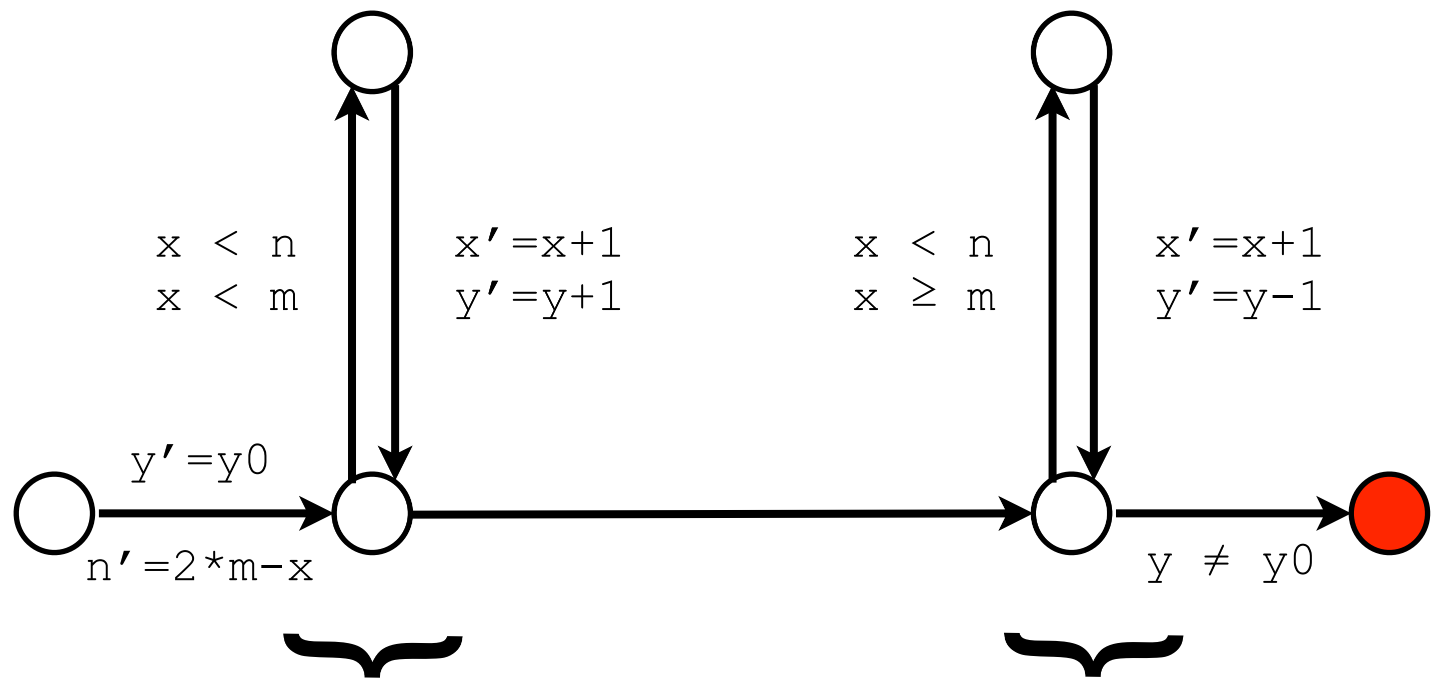
Flat Integer Programs

```
int y = y0;
int n = 2*m - x;

while (x < n) {
  if (x < m) {
    x ++;
    y ++;
  } else {
    x ++;
    y --;
}}

assert(y == y0);
```



Flat Integer Programs

```
int y = y0;
int n = 2*m - x;

while (x < n) {
  if (x < m) {
    x ++;
    y ++;
  } else {
    x ++;
    y --;
}}

assert(y == y0);
```



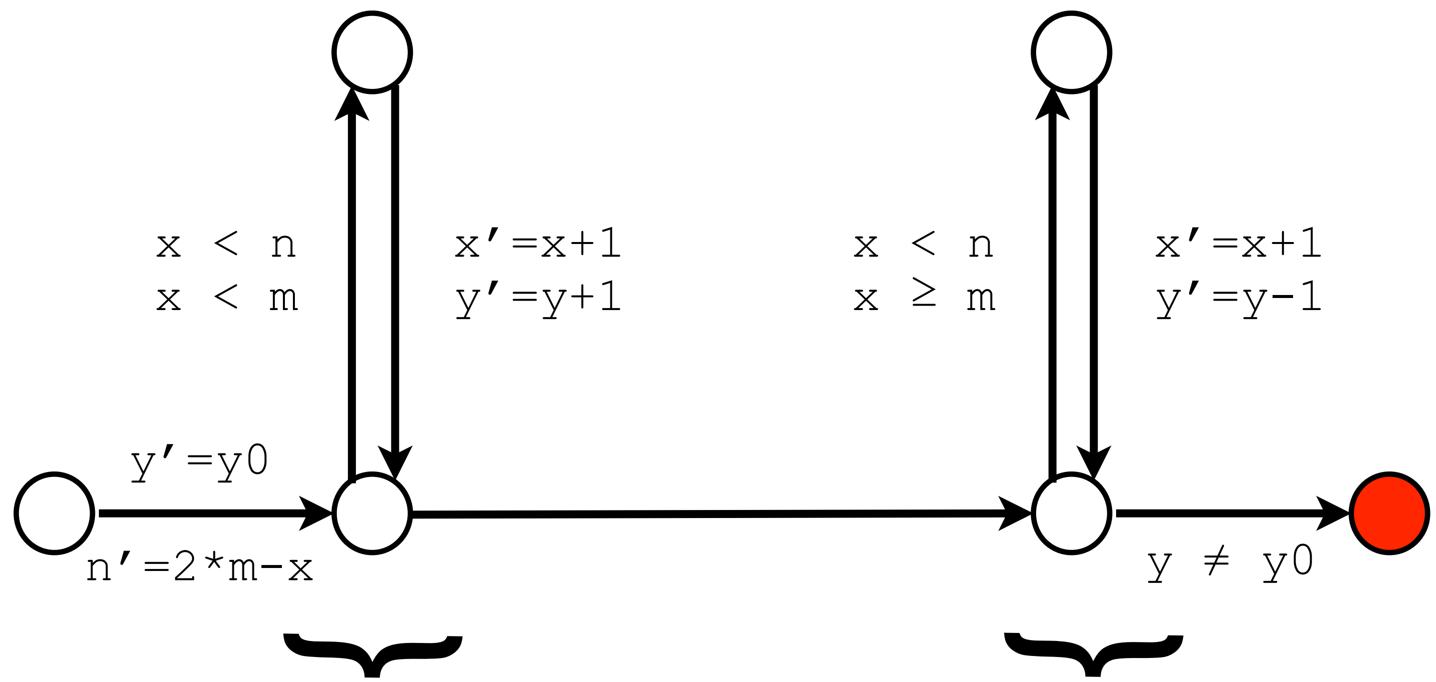
$x_1 - x_0 = y_1 - y_0 = m - x_0$

$x_2 - x_1 = y_1 - y_2 = n - m$

$y_1 - y_0 = y_1 - y_2$

Flat Integer Programs

```
int y = y0;
int n = 2*m - x;

while (x < n) {
  if (x < m) {
    x ++;
    y ++;
  } else {
    x ++;
    y --;
}}

assert(y == y0);
```



$x_1 - x_0 = y_1 - y_0 = m - x_0$ $x_2 - x_1 = y_1 - y_2 = n - m$

$y_1 - y_0 = y_1 - y_2$

$\boxed{y_0 = y_2}$

Flat Integer Programs

- Reachability is decidable if the relations labeling the loops belong to certain classes of linear inequalities

- Difference bounds constraints:

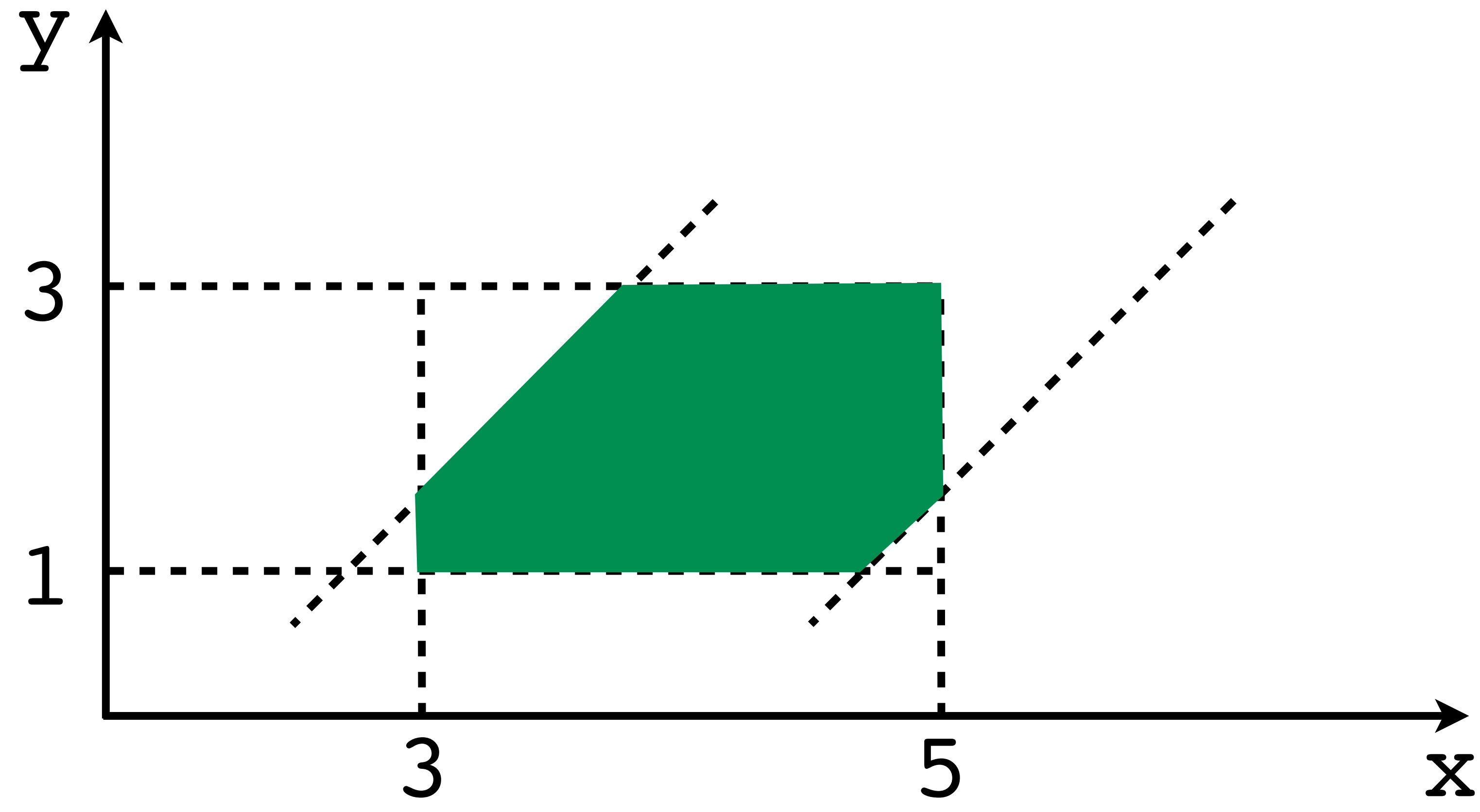
$$3 \leq x \leq 5 \wedge 1 \leq y \leq 3 \wedge 2 \leq x - y \leq 4$$

Flat Integer Programs

- Reachability is decidable if the relations labeling the loops belong to certain classes of linear inequalities

- Difference bounds constraints:

$$3 \leq x \leq 5 \wedge 1 \leq y \leq 3 \wedge 2 \leq x - y \leq 4$$



Flat Integer Programs

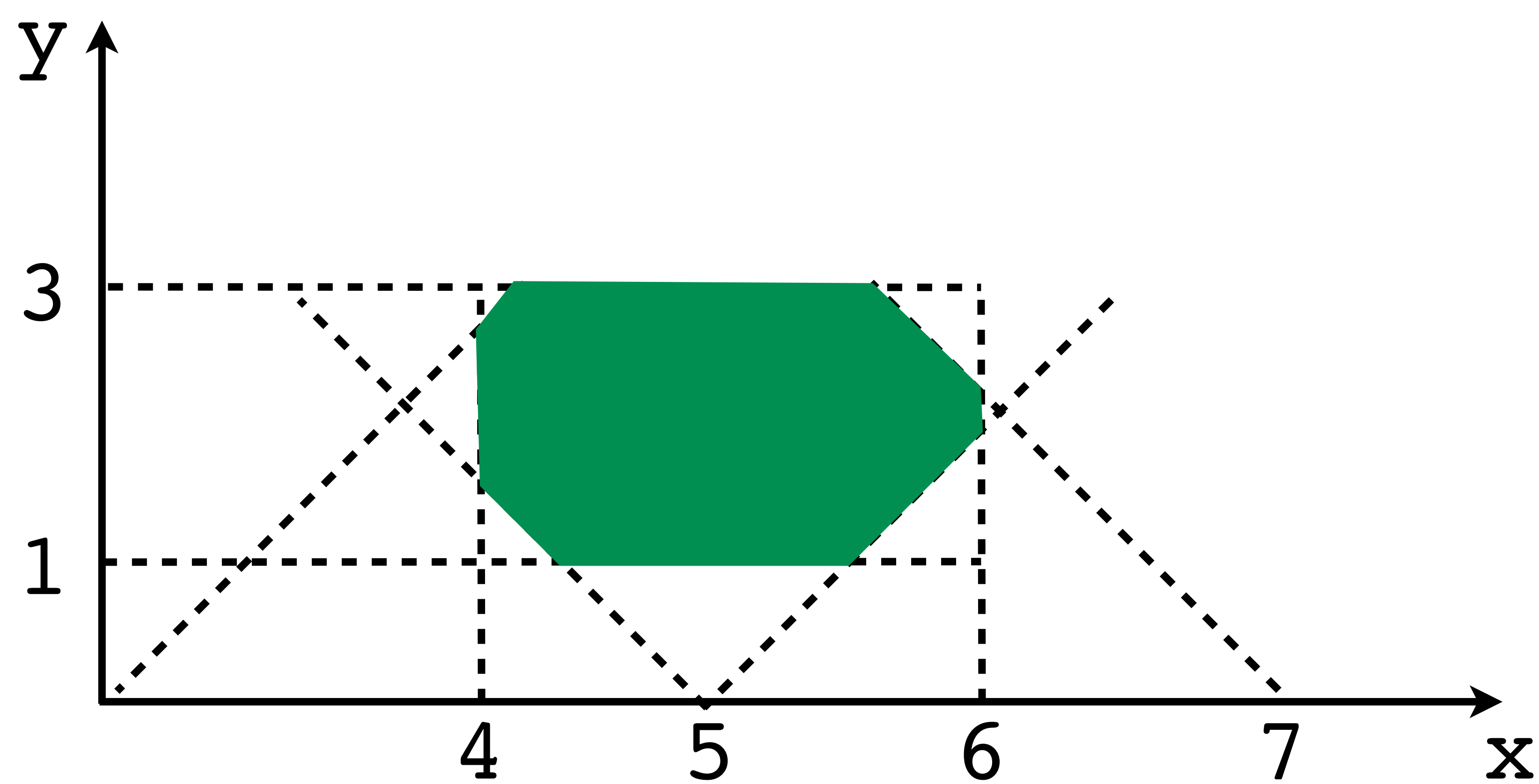
- Reachability is decidable if the relations labeling the loops belong to certain classes of linear inequalities
- Octagonal constraints:

$$4 \leq x \leq 6 \wedge 1 \leq y \leq 3 \wedge 0 \leq x - y \leq 5 \wedge 5 \leq x + y \leq 7$$

Flat Integer Programs

- Reachability is decidable if the relations labeling the loops belong to certain classes of linear inequalities
- Octagonal constraints:

$$4 \le x \le 6 \wedge 1 \le y \le 3 \wedge 0 \le x - y \le 5 \wedge 5 \le x + y \le 7$$

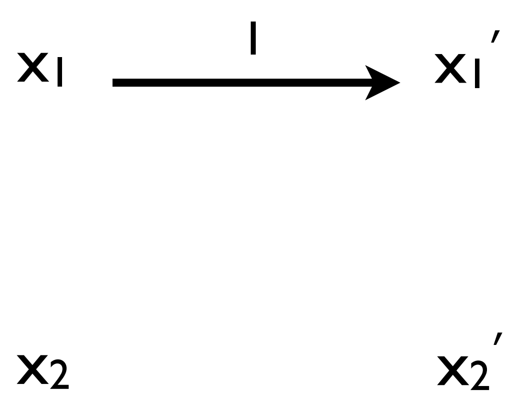


Difference Bounds Relations

x_1 x_1'

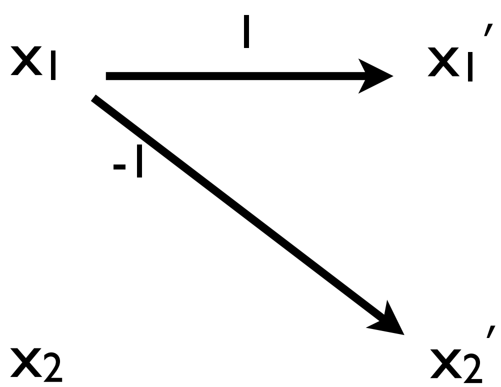
x_2 x_2'

Difference Bounds Relations



$x_1 - x_1' \leq 1$

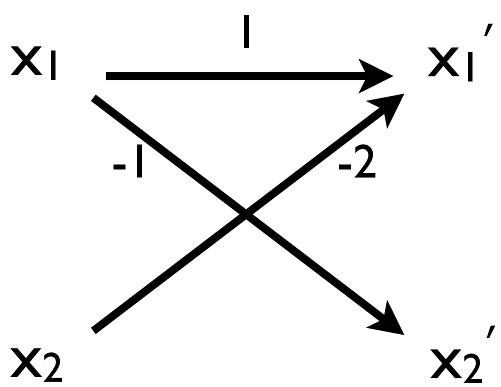
Difference Bounds Relations



$x_1 - x_1' \leq 1$

$x_1 - x_2' \leq -1$

Difference Bounds Relations

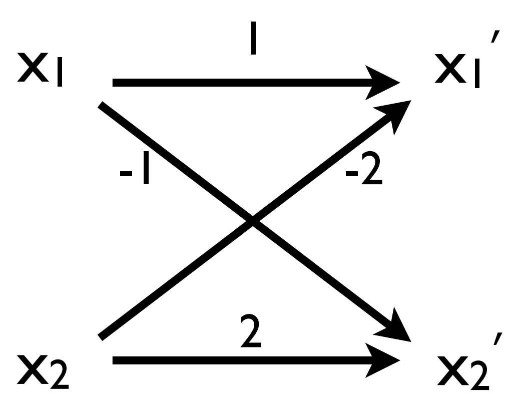


$x_1 - x_1' \leq 1$

$x_1 - x_2' \leq -1$

$x_2 - x_1' \leq -2$

Difference Bounds Relations



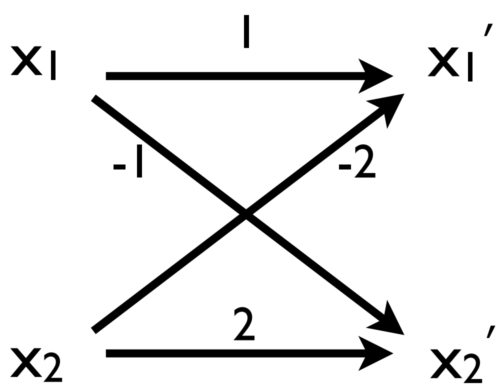
$x_1 - x_1' \leq 1$

$x_1 - x_2' \leq -1$

$x_2 - x_1' \leq -2$

$x_2 - x_2' \leq 2$

Difference Bounds Relations



$x_1 - x_1' \leq 1$

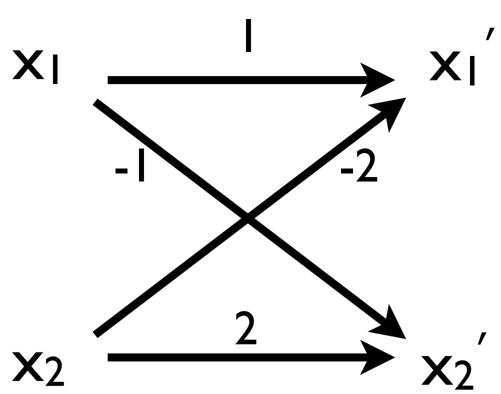
$x_1 - x_2' \leq -1$

$x_2 - x_1' \leq -2$

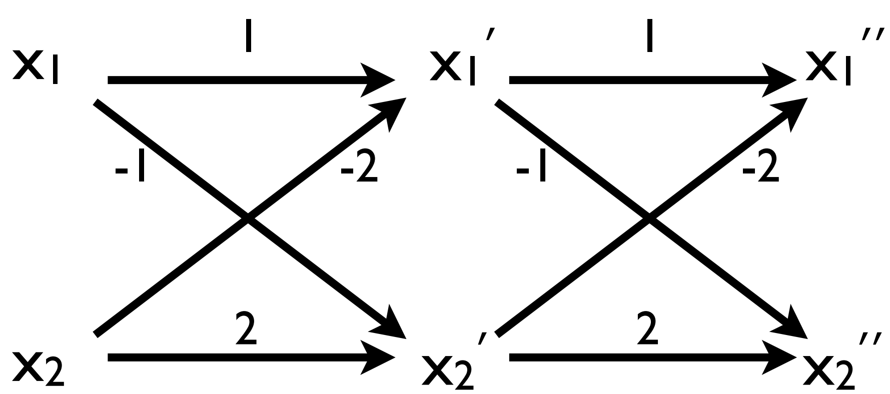
$x_2 - x_2' \leq 2$

	x_1	x_2	x_1'	x_2'
x_1	0	∞	1	-1
x_2	∞	0	-2	2
x_1'	∞	∞	0	∞
x_2'	∞	∞	∞	0

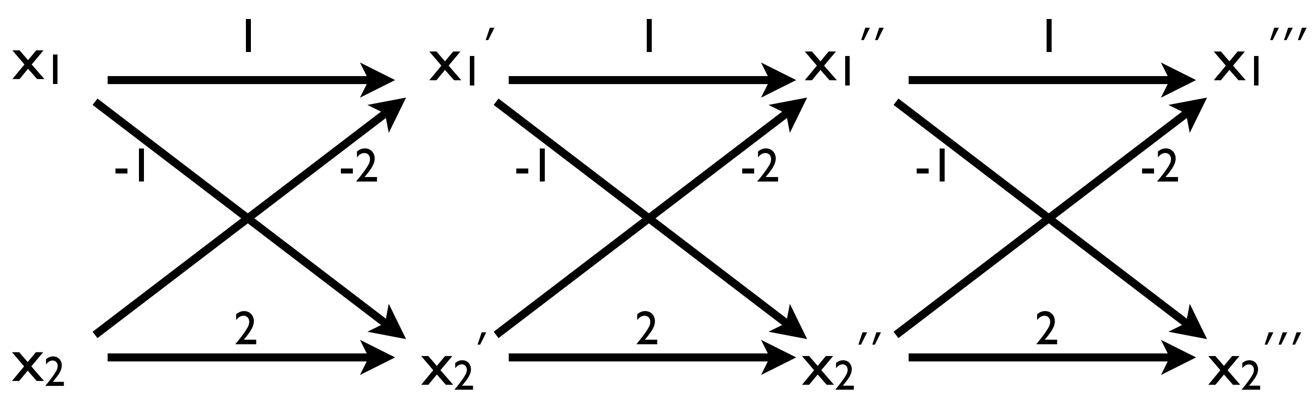
Difference Bounds Relations



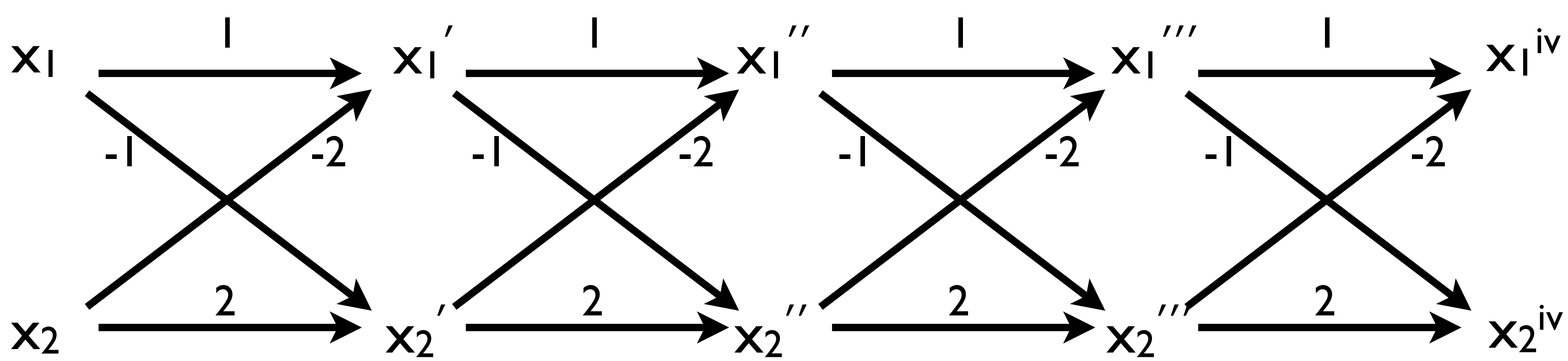
Difference Bounds Relations



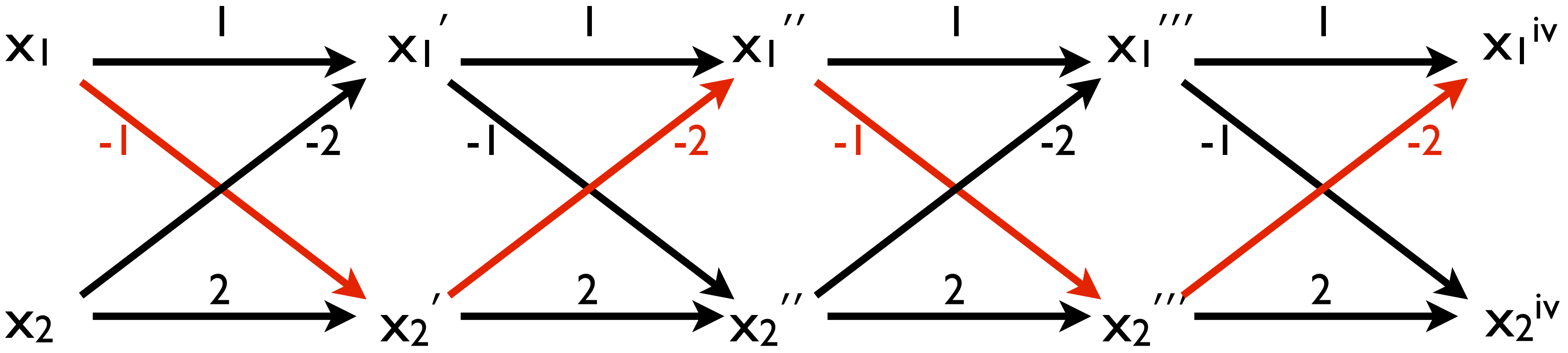
Difference Bounds Relations



Difference Bounds Relations

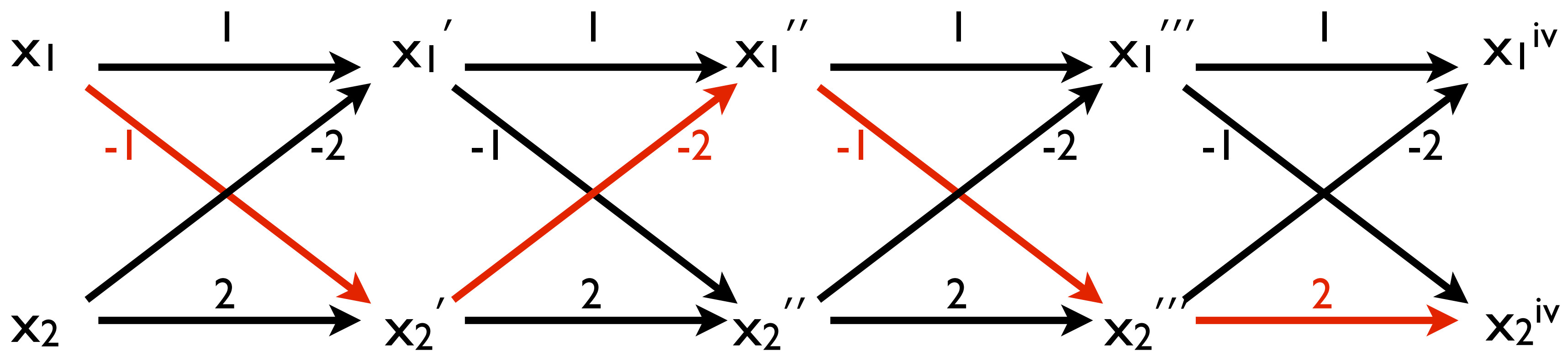


Difference Bounds Relations



$x_1 - x_1^{iv} \leq -6$

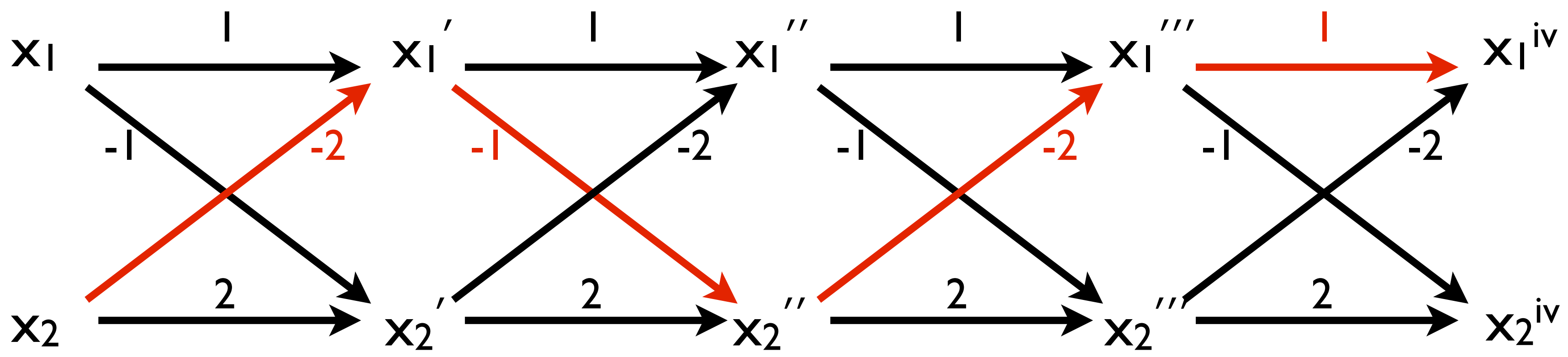
Difference Bounds Relations



$x_1 - x_1^{iv} \leq -6$

$x_1 - x_2^{iv} \leq -2$

Difference Bounds Relations

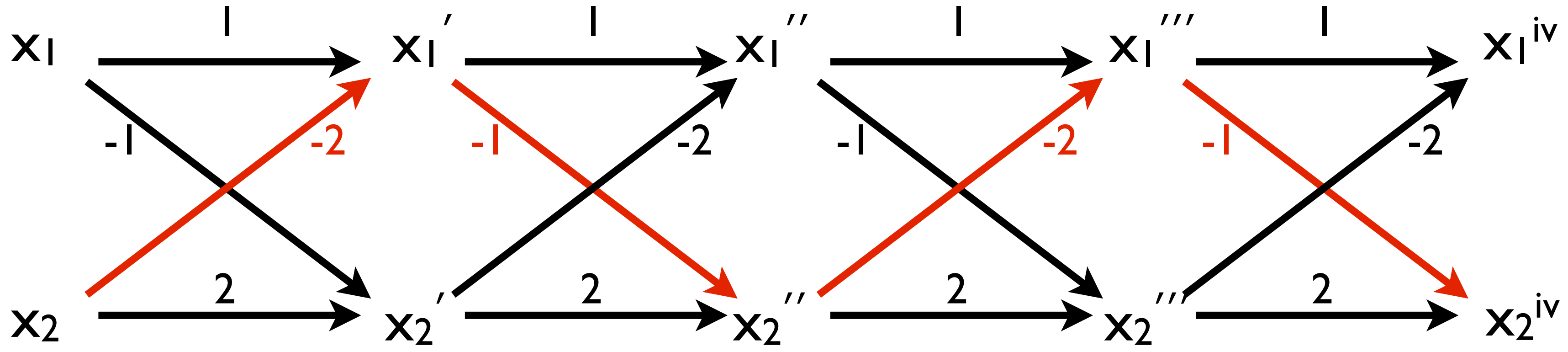


$x_1 - x_1^{iv} \leq -6$

$x_1 - x_2^{iv} \leq -2$

$x_2 - x_1^{iv} \leq -4$

Difference Bounds Relations



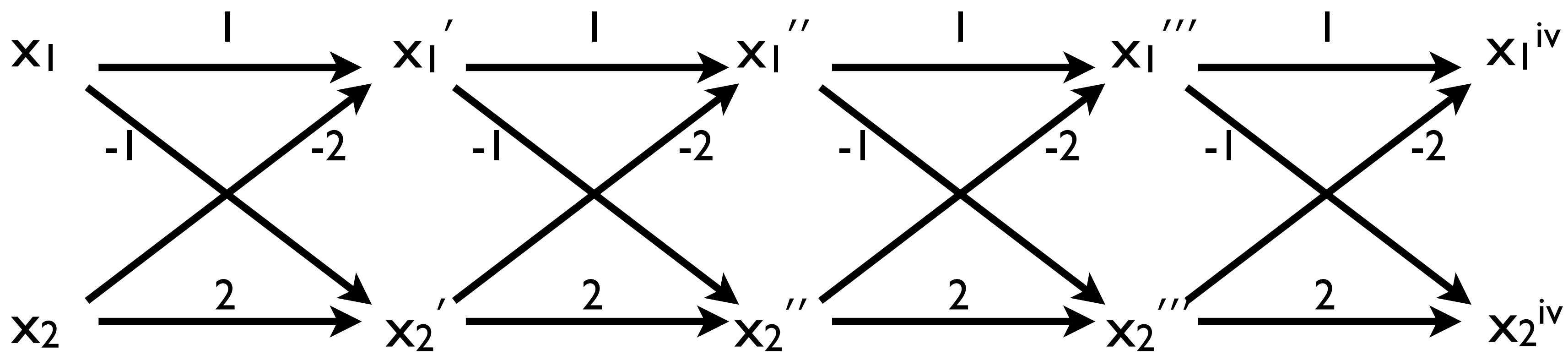
$x_1 - x_1^{iv} \leq -6$

$x_1 - x_2^{iv} \leq -2$

$x_2 - x_1^{iv} \leq -4$

$x_2 - x_2^{iv} \leq -6$

Difference Bounds Relations



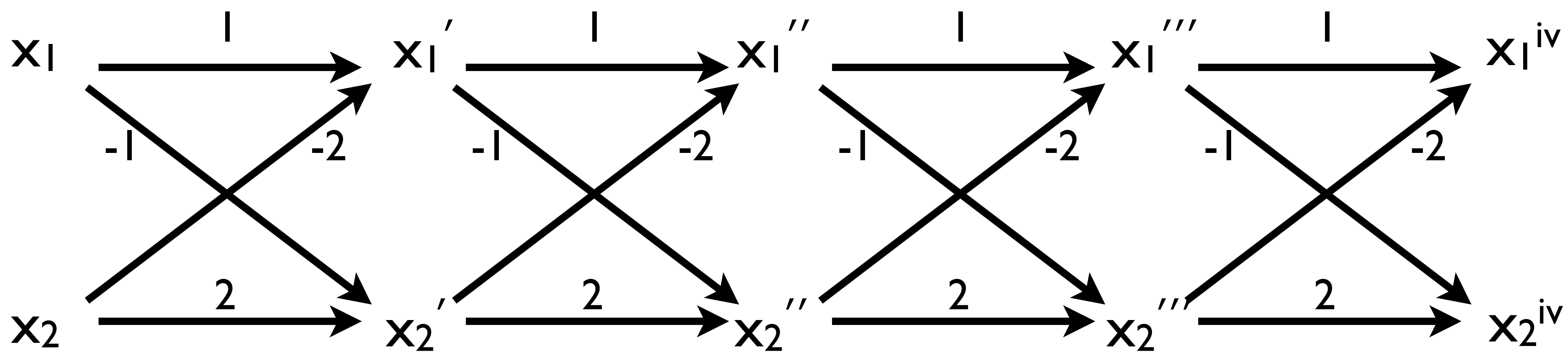
$x_1 - x_1^{iv} \leq -6$

$x_1 - x_2^{iv} \leq -2$

$x_2 - x_1^{iv} \leq -4$

$x_2 - x_2^{iv} \leq -6$

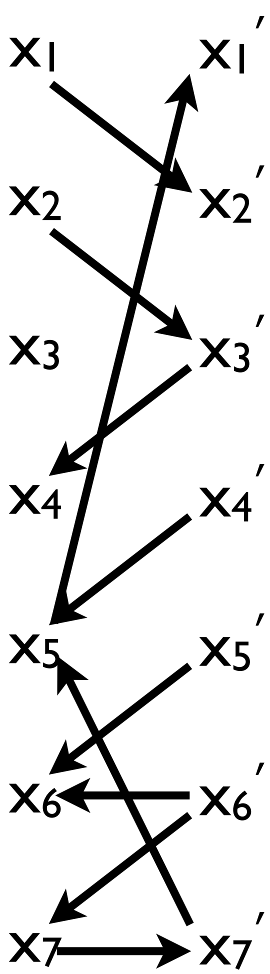
Difference Bounds Relations



$x_1 - x_1^{iv} \leq -6$
$x_1 - x_2^{iv} \leq -2$
$x_2 - x_1^{iv} \leq -4$
$x_2 - x_2^{iv} \leq -6$

- The n-th power of a DB relation is again a DB relation:
  ➡ the class of DB has quantifier elimination
- We are interested in computing minimal weight paths
- The graph for the n-th power has (n+1)×(#vars) nodes
- The paths in the graph are regular

Difference Bounds Relations



$x_1 - x_2' \leq 0$

$x_2 - x_3' \leq 0$

$x_3' - x_2 \leq 0$

$x_4' - x_5 \leq 0$

$x_5' - x_6 \leq 0$

$x_6' - x_6 \leq 1$

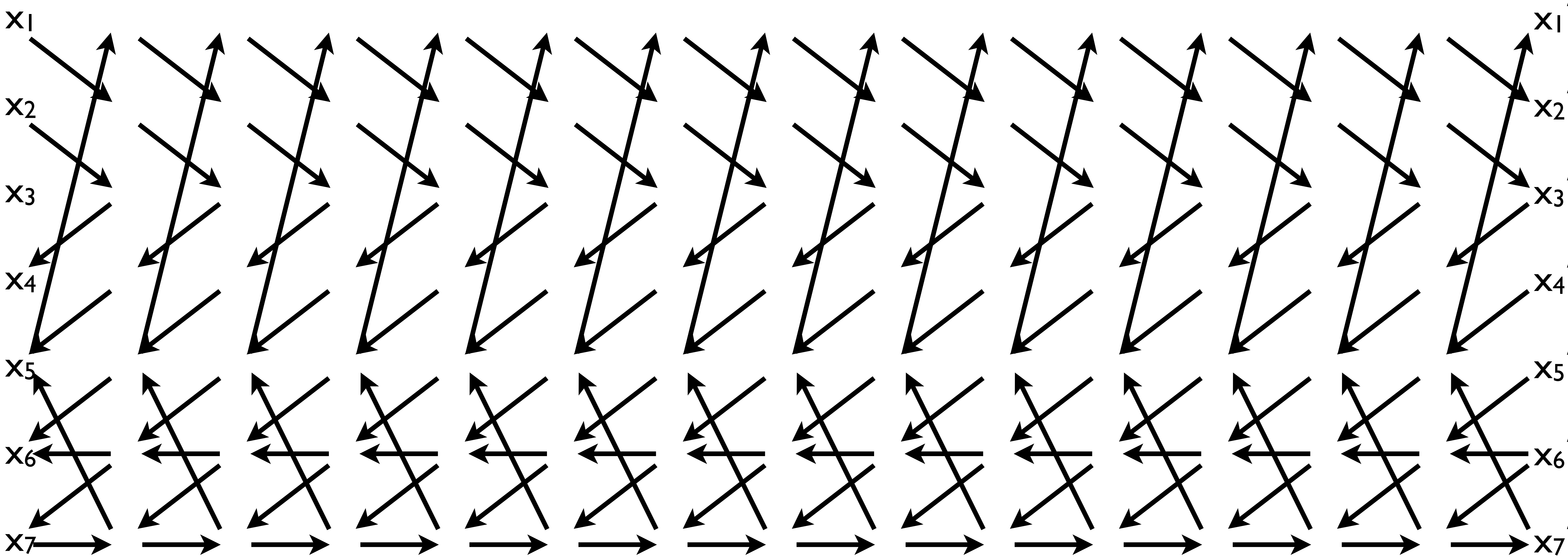
$x_6' - x_7 \leq 0$

$x_7 - x_7' \leq 1$

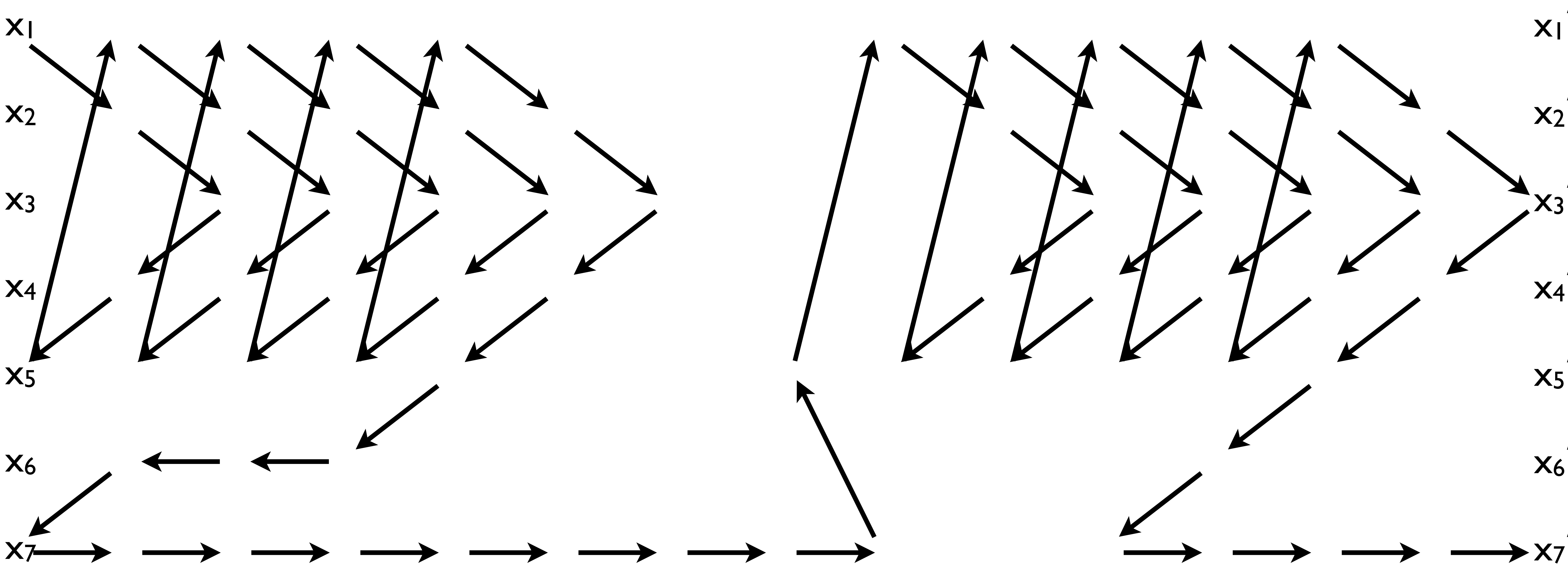
$x_7' - x_5 \leq 0$

$x_5 - x_1' \leq -1$

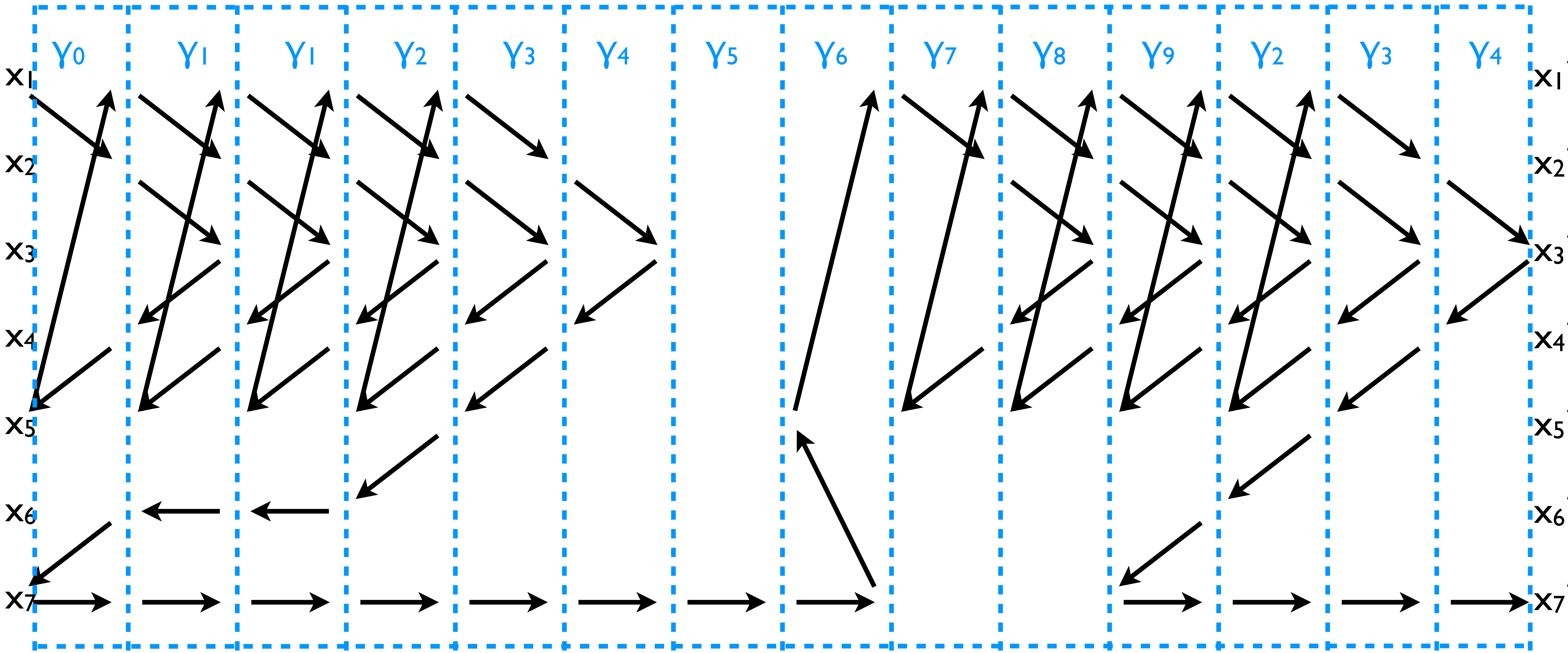
Difference Bounds Relations



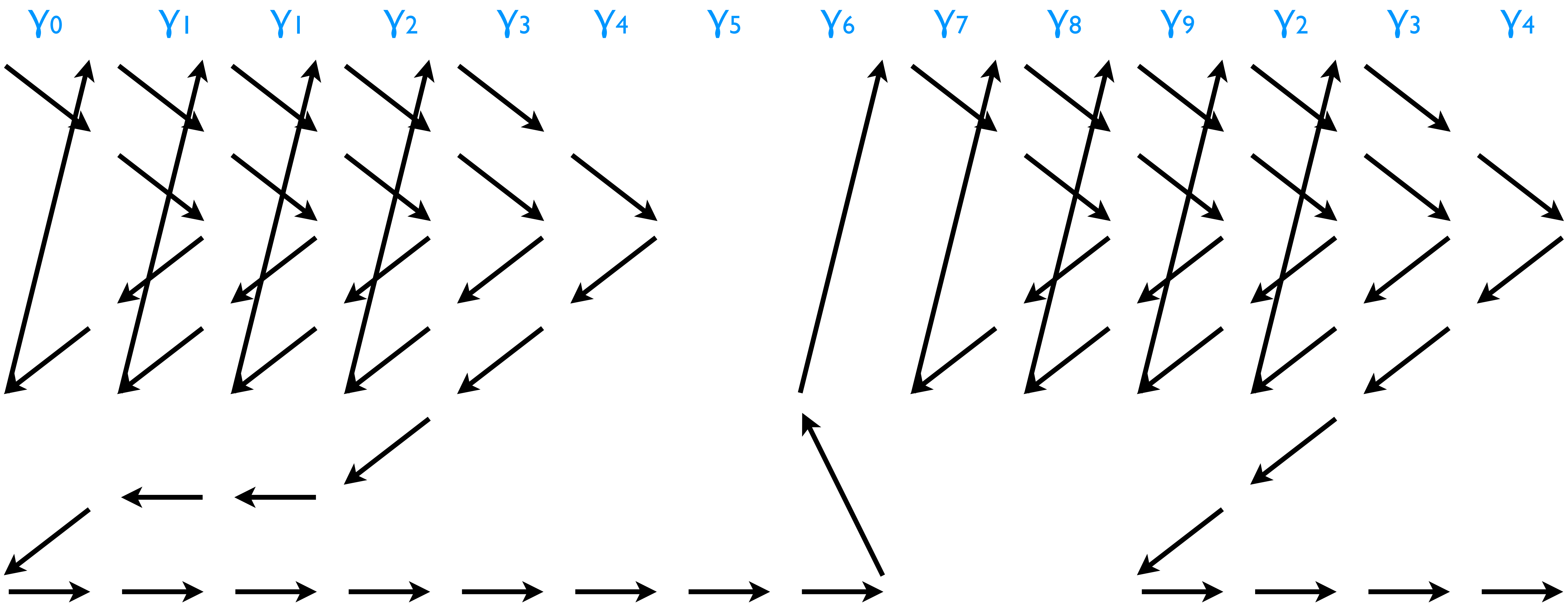
Difference Bounds Relations



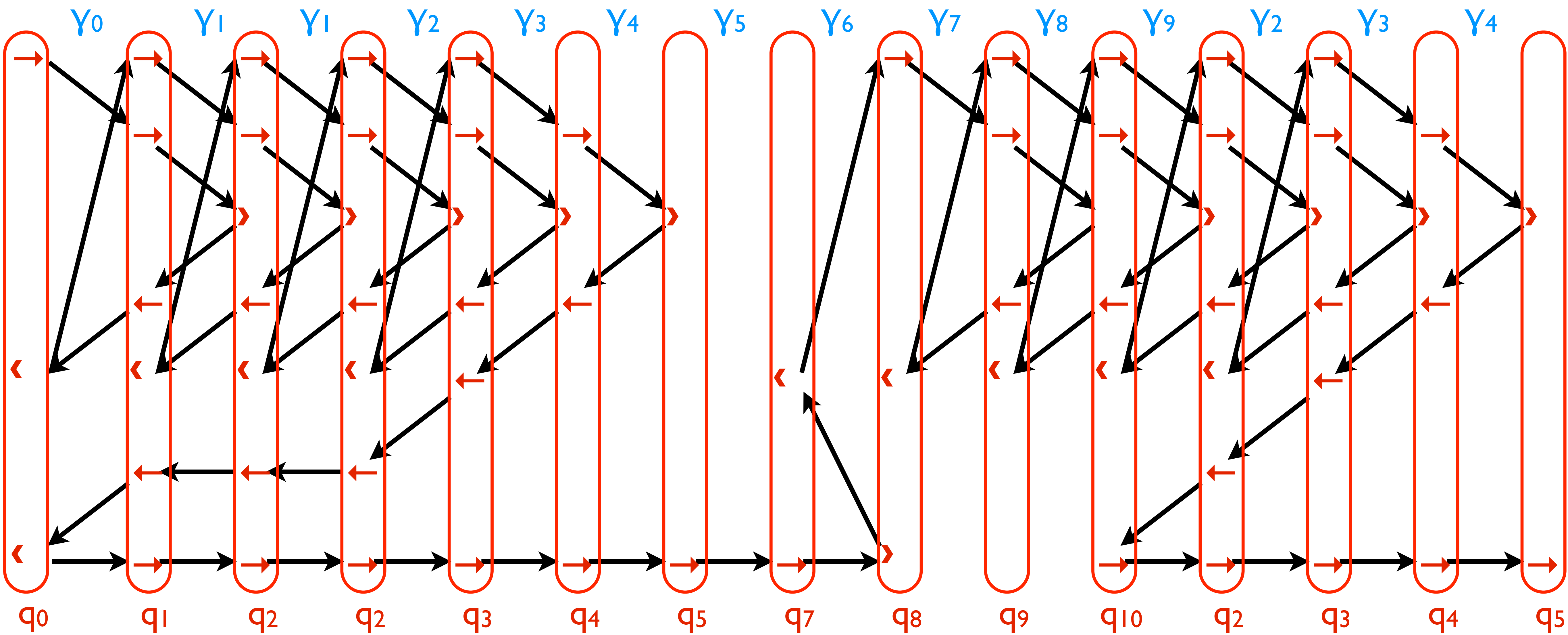
Difference Bounds Relations



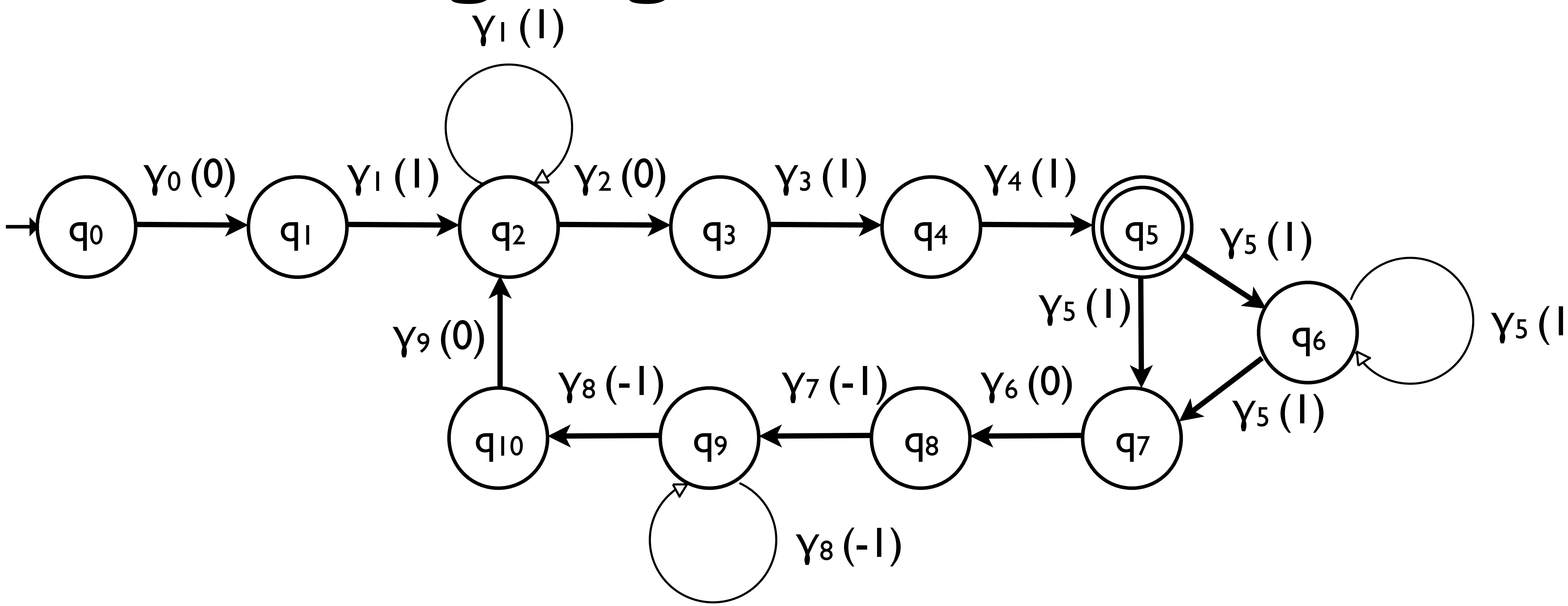
Difference Bounds Relations



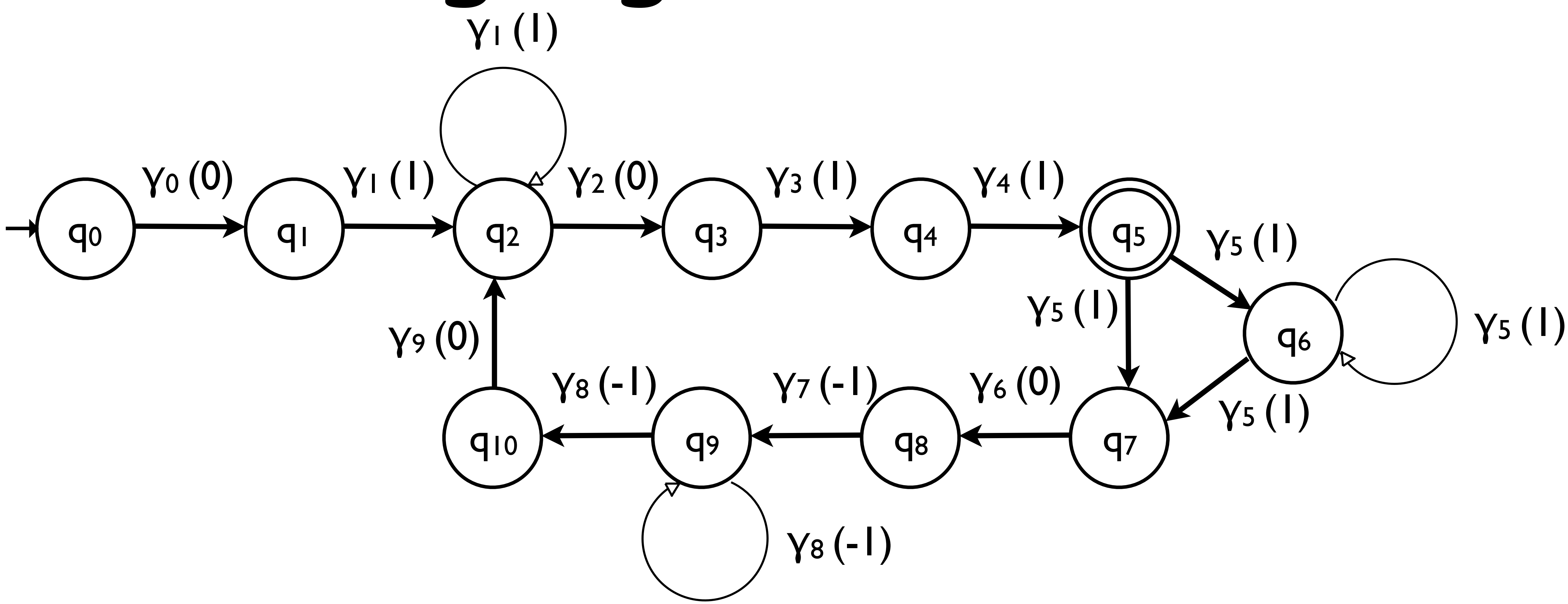
Difference Bounds Relations



Zigzag Automata



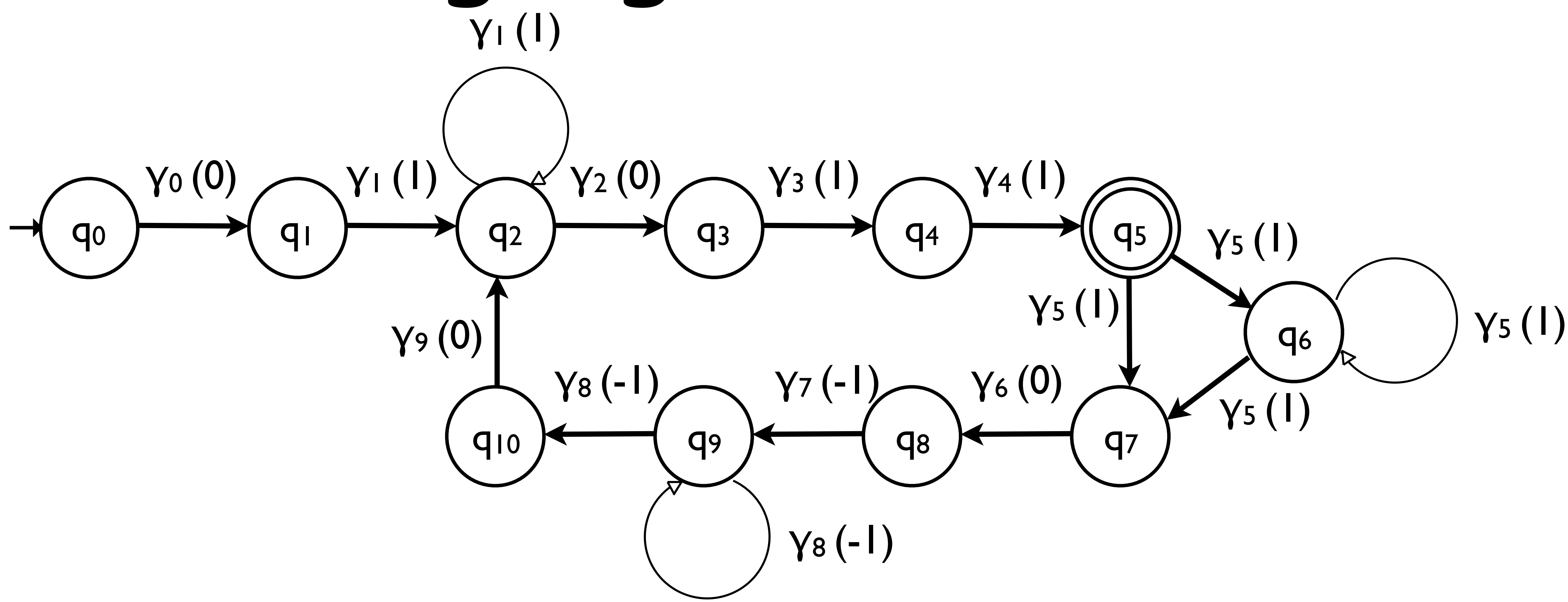
Zigzag Automata



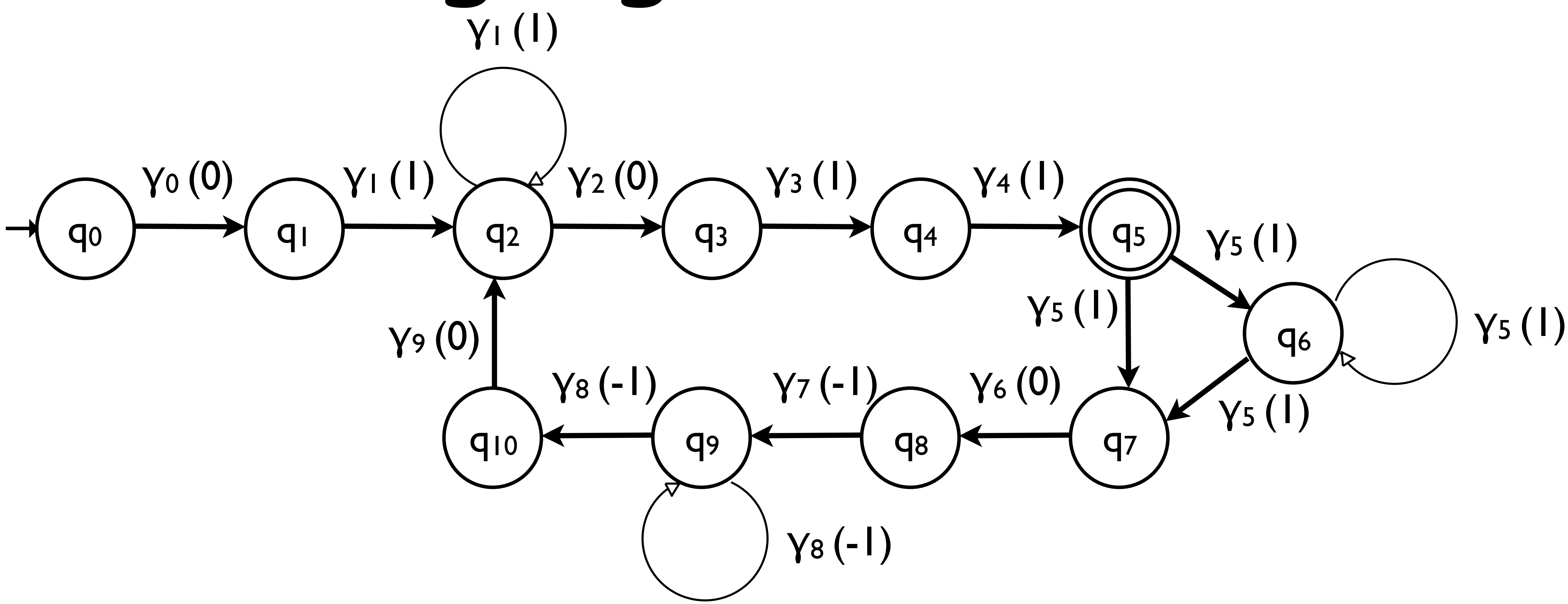
- All paths in the n-th unfolding of the constraint graph are encoded as runs of weighted automata [BIL'06]

- Minimal weight paths become minimal weight runs

Zigzag Automata



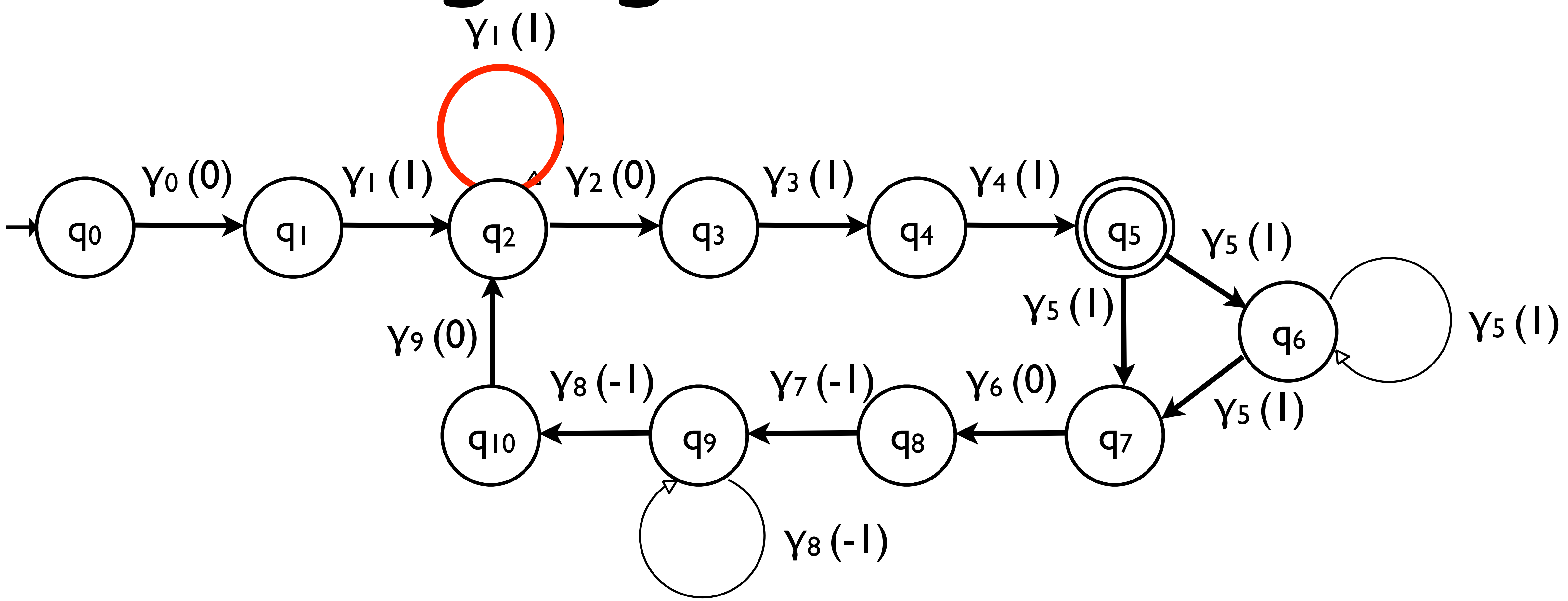
Zigzag Automata



- We compute a function on the automaton:
  $\text{min\_weight}_A(n) = \min\{\omega(\rho) \mid \rho \text{ is a run of } A, |\rho|=n\}$

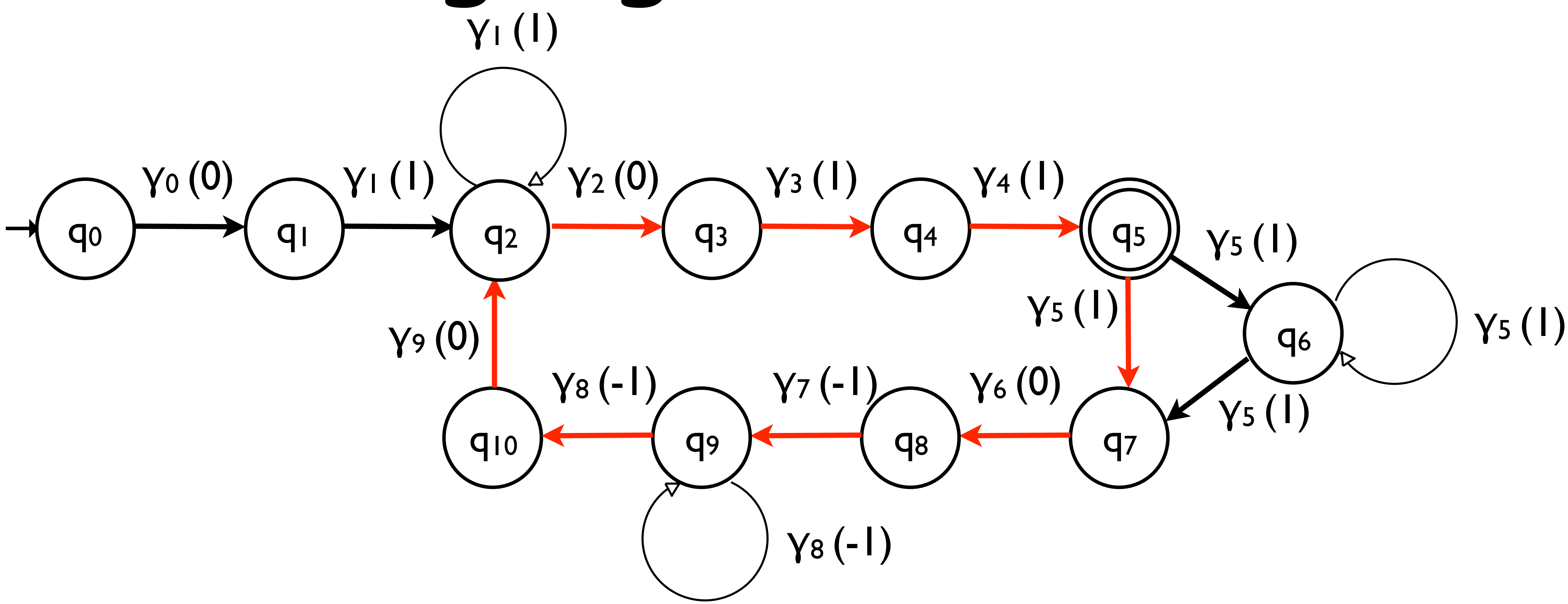
- Minimal weight functions are periodic [deSchutter'00]
  ➡ mininal weight runs iterate through critical cycles

Zigzag Automata



$\varpi(\gamma_1^*) = \omega(\gamma_1) \,/\, |\,\gamma_1\,| = 1$

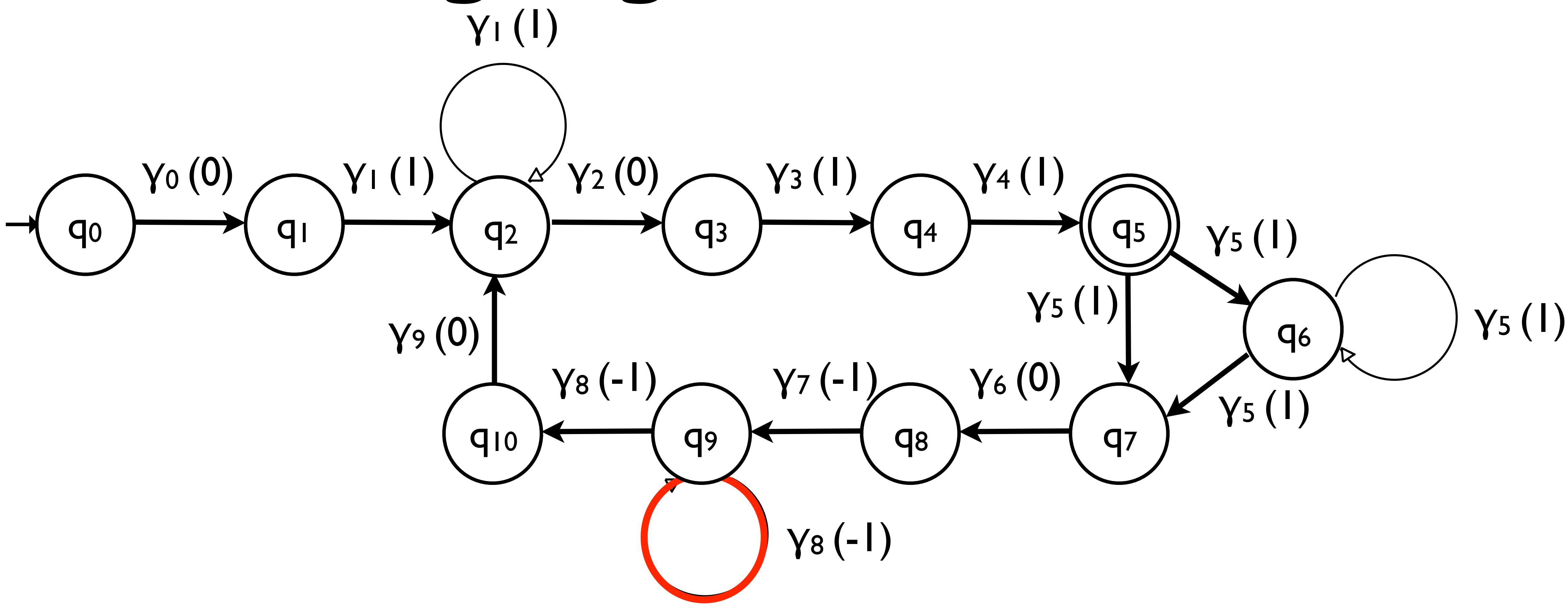
Zigzag Automata



$\varpi(\gamma_1^*) = \omega(\gamma_1) \,/\, |\, \gamma_1 \,| = 1$

$\varpi((\gamma_2\, \gamma_3\, \gamma_4\, \gamma_5\, \gamma_6\, \gamma_7\, \gamma_8\, \gamma_9)^*) = 1/8$

Zigzag Automata

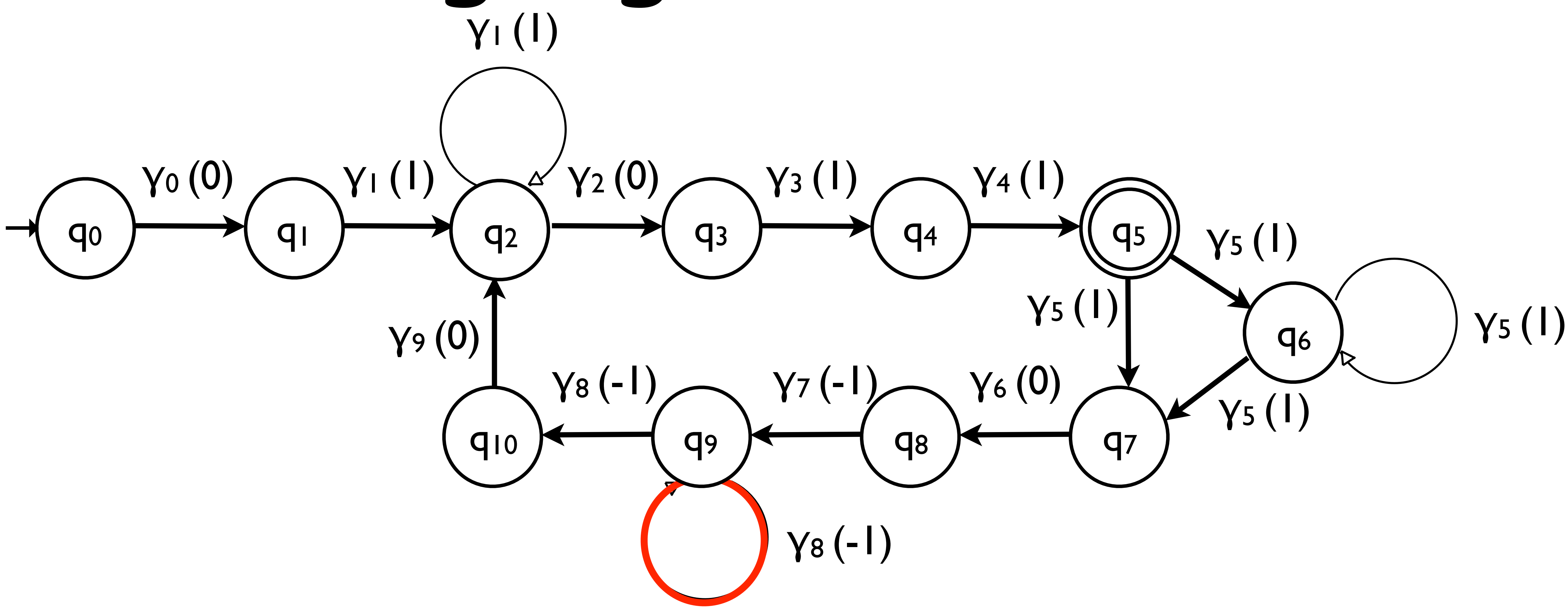


$\varpi(\gamma_1^*) = \omega(\gamma_1) / |\gamma_1| = 1$

$\varpi((\gamma_2\ \gamma_3\ \gamma_4\ \gamma_5\ \gamma_6\ \gamma_7\ \gamma_8\ \gamma_9)^*) = 1/8$

$\varpi(\gamma_8^*) = -1$

Zigzag Automata



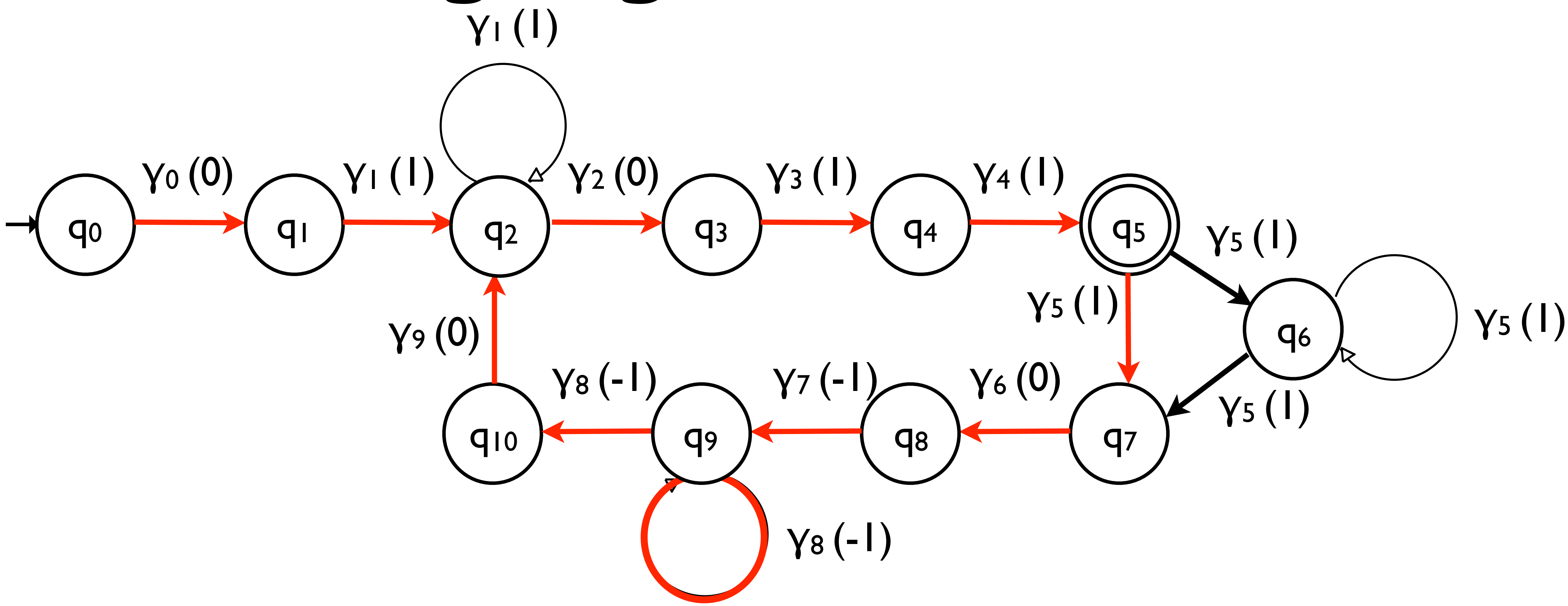
$\varpi(\gamma_1^*) = \omega(\gamma_1) / | \gamma_1 | = 1$

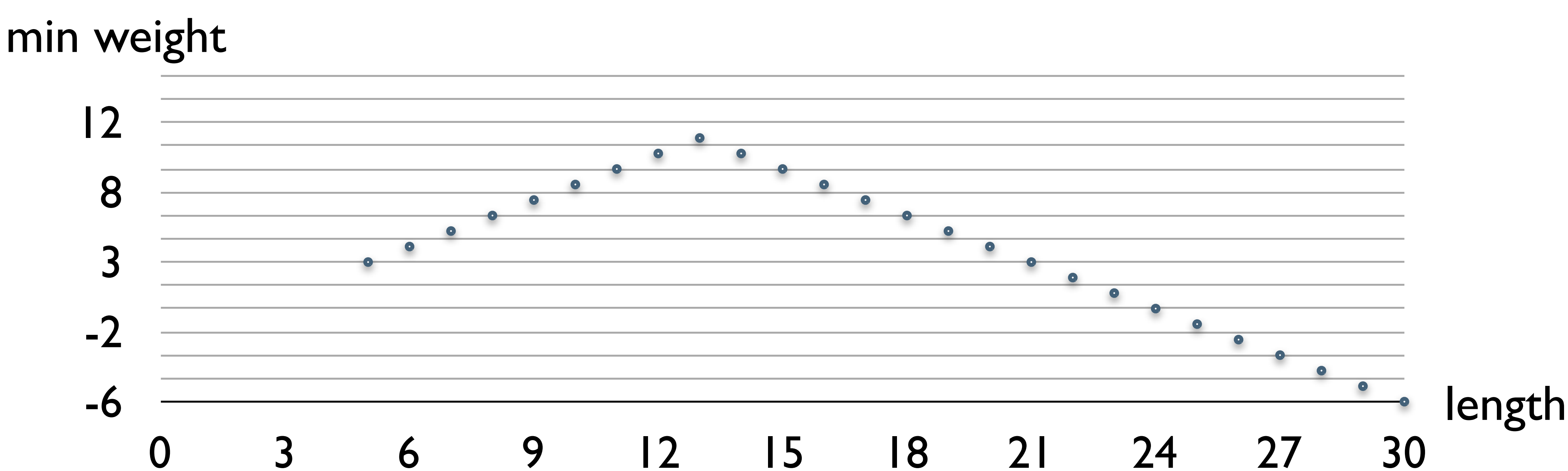
$\varpi((\gamma_2 \gamma_3 \gamma_4 \gamma_5 \gamma_6 \gamma_7 \gamma_8 \gamma_9)^*) = 1/8$

$\varpi(\gamma_8^*) = -1$

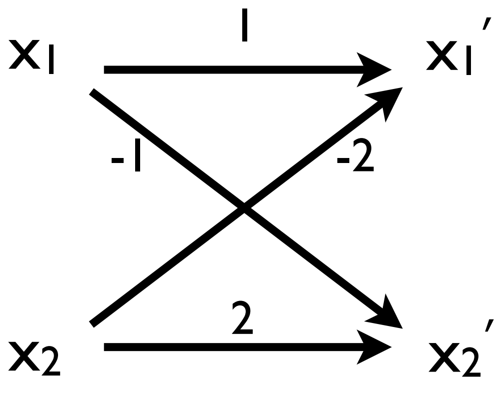
γ_8^* is a critical cycle in its SCC

Zigzag Automata



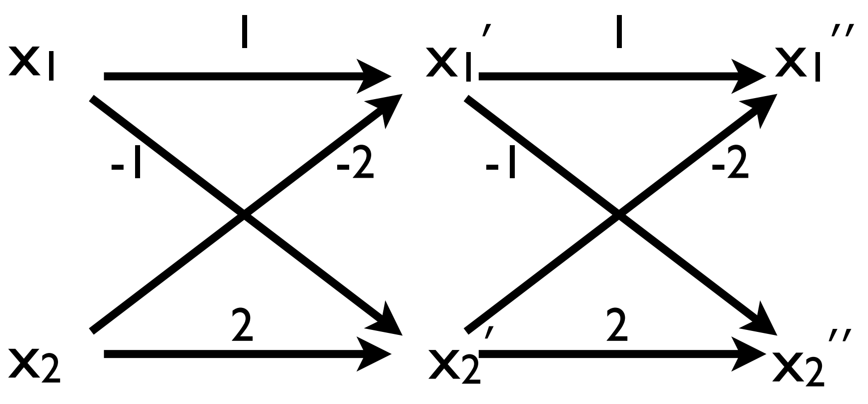


Periodic Relations



0	∞	1	-1
∞	0	-2	2
∞	∞	0	∞
∞	∞	∞	0

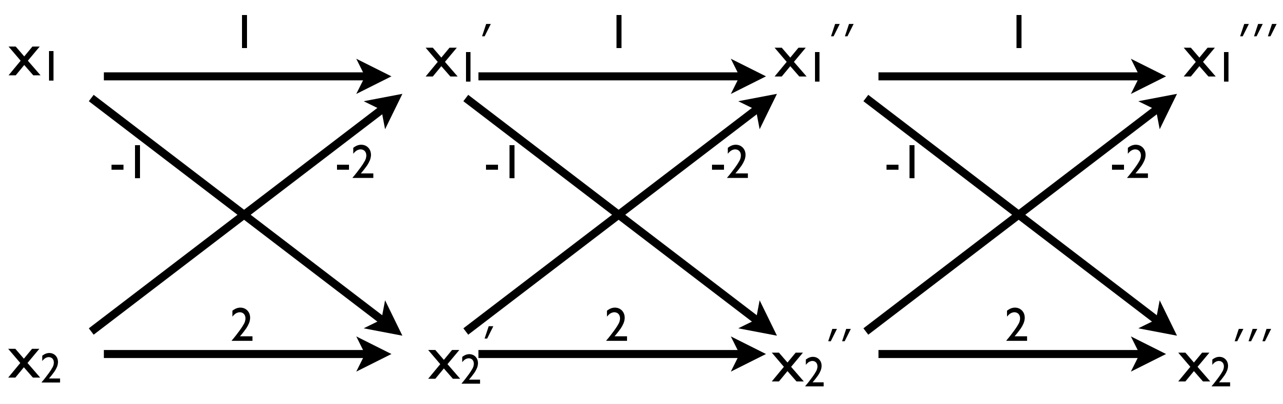
Periodic Relations



0	∞	1	-1
∞	0	-2	2
∞	∞	0	∞
∞	∞	∞	0

0	∞	-3	0
∞	0	-1	-3
∞	∞	0	∞
∞	∞	∞	0

Periodic Relations

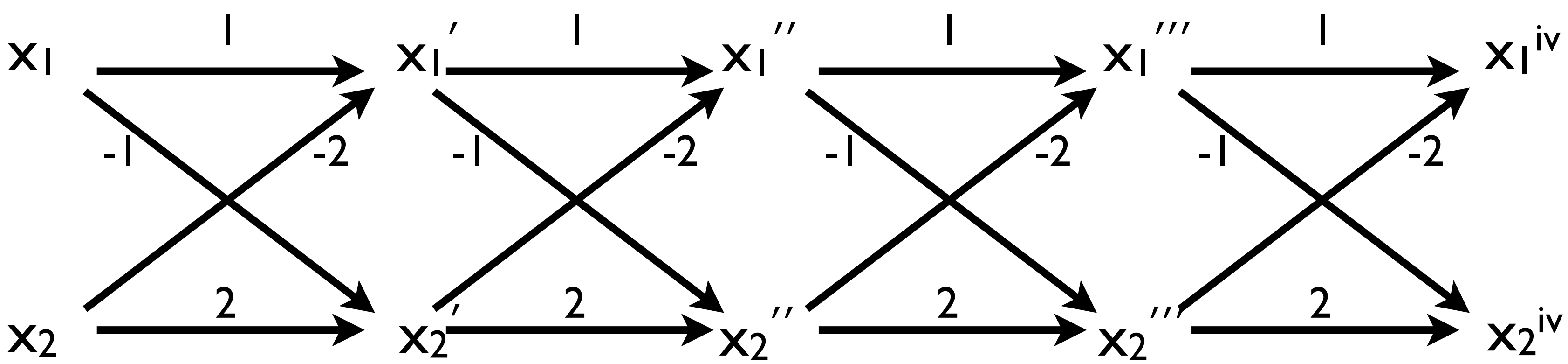


0	∞	1	-1
∞	0	-2	2
∞	∞	0	∞
∞	∞	∞	0

0	∞	-3	0
∞	0	-1	-3
∞	∞	0	∞
∞	∞	∞	0

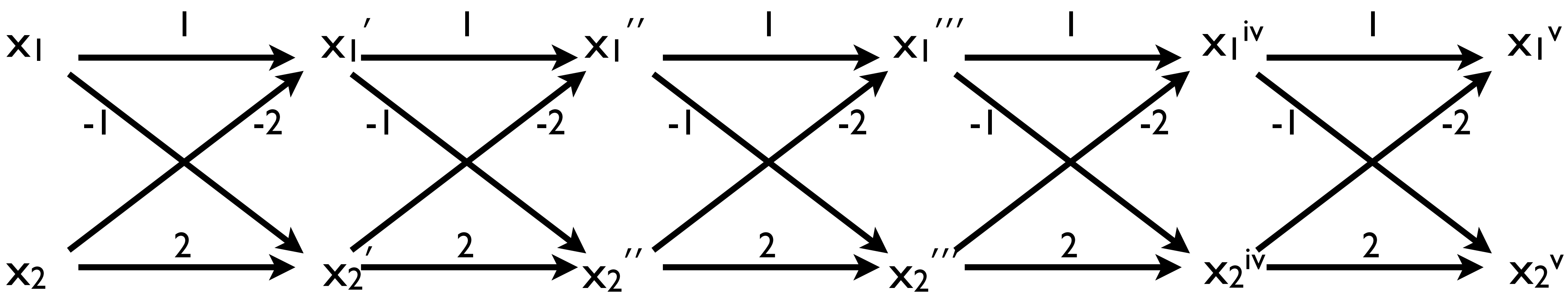
0	∞	-2	-4
∞	0	-5	-1
∞	∞	0	∞
∞	∞	∞	0

Periodic Relations



$$
\begin{matrix} 0 & \infty & 1 & -1 \\ \infty & 0 & -2 & 2 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix}
\qquad
\begin{matrix} 0 & \infty & -3 & 0 \\ \infty & 0 & -1 & -3 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix}
\qquad
\begin{matrix} 0 & \infty & -2 & -4 \\ \infty & 0 & -5 & -1 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix}
\qquad
\begin{matrix} 0 & \infty & -6 & -2 \\ \infty & 0 & -4 & -6 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix}
$$

Periodic Relations



0	∞	1	-1
∞	0	-2	2
∞	∞	0	∞
∞	∞	∞	0

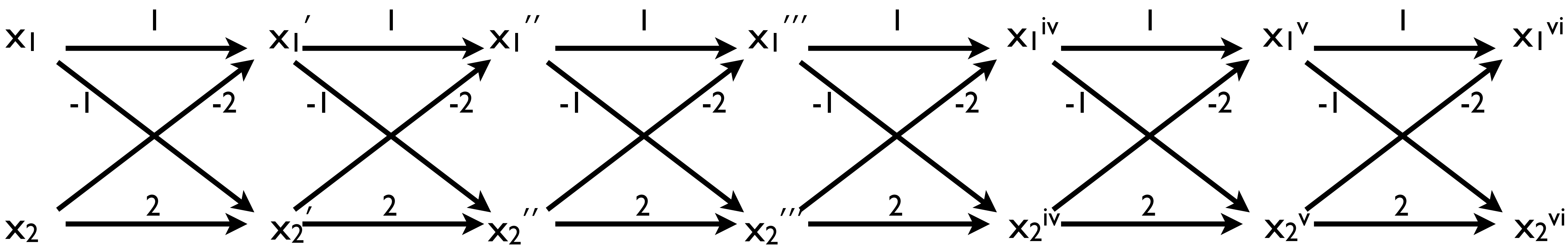
0	∞	-3	0
∞	0	-1	-3
∞	∞	0	∞
∞	∞	∞	0

0	∞	-2	-4
∞	0	-5	-1
∞	∞	0	∞
∞	∞	∞	0

0	∞	-6	-2
∞	0	-4	-6
∞	∞	0	∞
∞	∞	∞	0

0	∞	-5	-7
∞	0	-8	-4
∞	∞	0	∞
∞	∞	∞	0

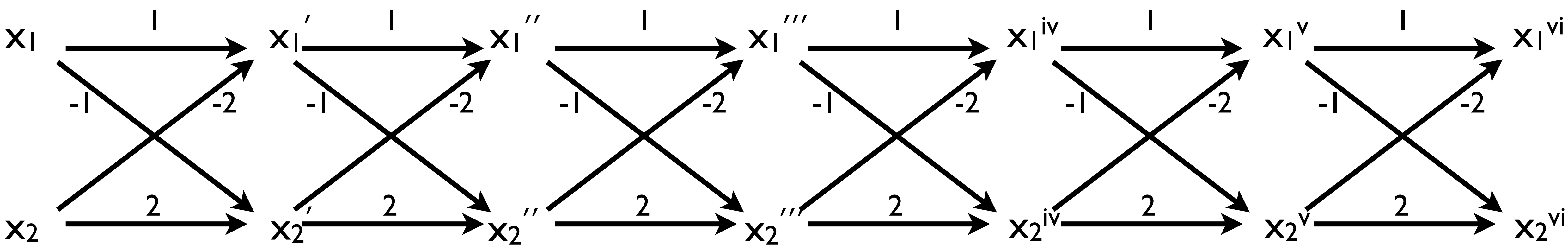
Periodic Relations



$$\begin{matrix} 0 & \infty & 1 & -1 \\ \infty & 0 & -2 & 2 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix} \qquad \begin{matrix} 0 & \infty & -3 & 0 \\ \infty & 0 & -1 & -3 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix} \qquad \begin{matrix} 0 & \infty & -2 & -4 \\ \infty & 0 & -5 & -1 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix}$$

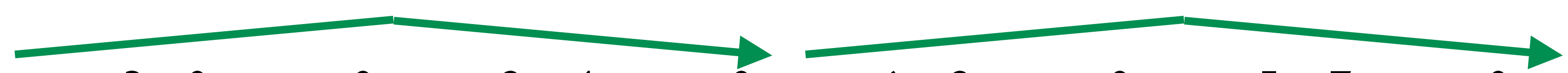
$$\begin{matrix} 0 & \infty & -6 & -2 \\ \infty & 0 & -4 & -6 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix} \qquad \begin{matrix} 0 & \infty & -5 & -7 \\ \infty & 0 & -8 & -4 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix} \qquad \begin{matrix} 0 & \infty & -9 & -4 \\ \infty & 0 & -7 & -9 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix}$$

Periodic Relations



0	∞	-3	-2
∞	0	-3	-3
∞	∞	0	∞
∞	∞	∞	0

0	∞	-3	-2
∞	0	-3	-3
∞	∞	0	∞
∞	∞	∞	0



0	∞	1	-1
∞	0	-2	2
∞	∞	0	∞
∞	∞	∞	0

0	∞	-3	0
∞	0	-1	-3
∞	∞	0	∞
∞	∞	∞	0

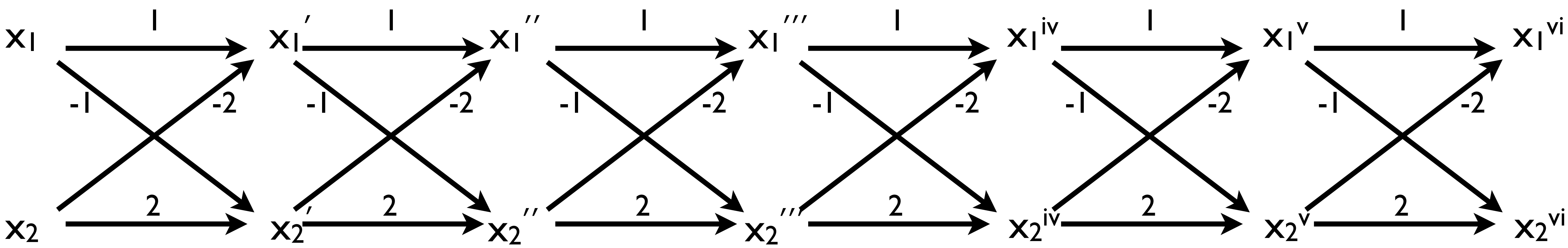
0	∞	-2	-4
∞	0	-5	-1
∞	∞	0	∞
∞	∞	∞	0

0	∞	-6	-2
∞	0	-4	-6
∞	∞	0	∞
∞	∞	∞	0

0	∞	-5	-7
∞	0	-8	-4
∞	∞	0	∞
∞	∞	∞	0

0	∞	-9	-4
∞	0	-7	-9
∞	∞	0	∞
∞	∞	∞	0

Periodic Relations



0	∞	-3	-2
∞	0	-3	-3
∞	∞	0	∞
∞	∞	∞	0

0	∞	-3	-2
∞	0	-3	-3
∞	∞	0	∞
∞	∞	∞	0

0	∞	1	-1
∞	0	-2	2
∞	∞	0	∞
∞	∞	∞	0

0	∞	-3	0
∞	0	-1	-3
∞	∞	0	∞
∞	∞	∞	0

0	∞	-2	-4
∞	0	-5	-1
∞	∞	0	∞
∞	∞	∞	0

0	∞	-6	-2
∞	0	-4	-6
∞	∞	0	∞
∞	∞	∞	0

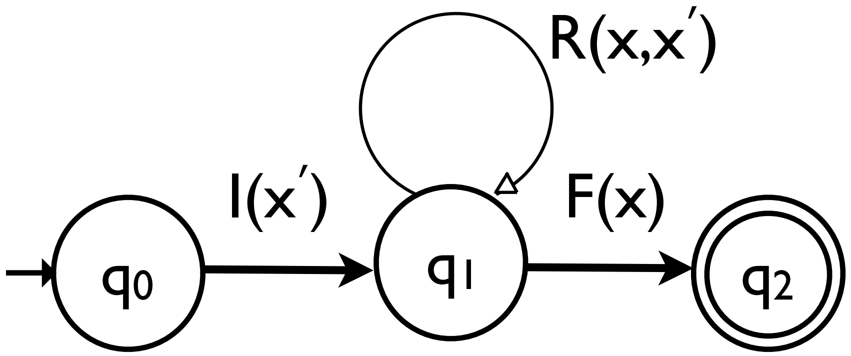
0	∞	-5	-7
∞	0	-8	-4
∞	∞	0	∞
∞	∞	∞	0

0	∞	-9	-4
∞	0	-7	-9
∞	∞	0	∞
∞	∞	∞	0

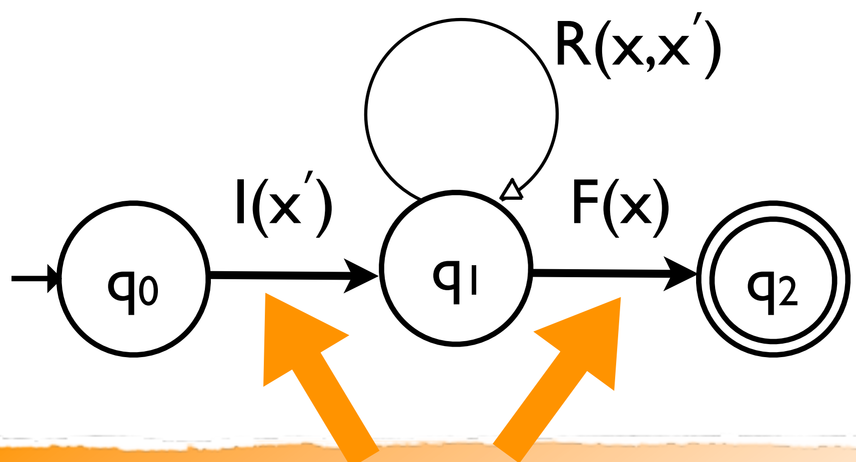
0	∞	-3	-3
∞	0	-3	-3
∞	∞	0	∞
∞	∞	∞	0

0	∞	-3	-3
∞	0	-3	-3
∞	∞	0	∞
∞	∞	∞	0

Periodicity and NTIME Safety

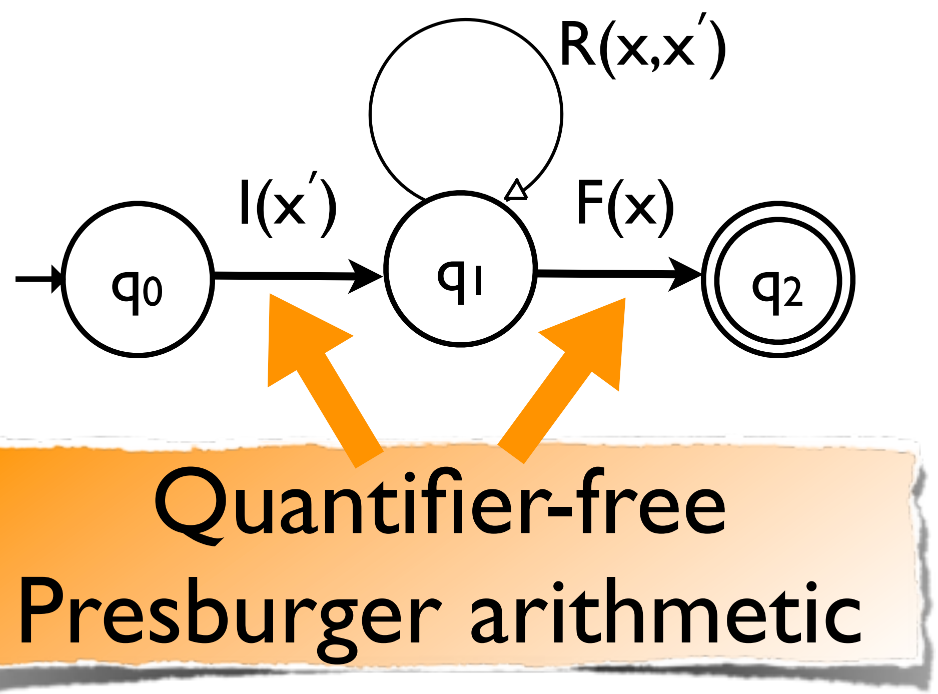


Periodicity and NTIME Safety



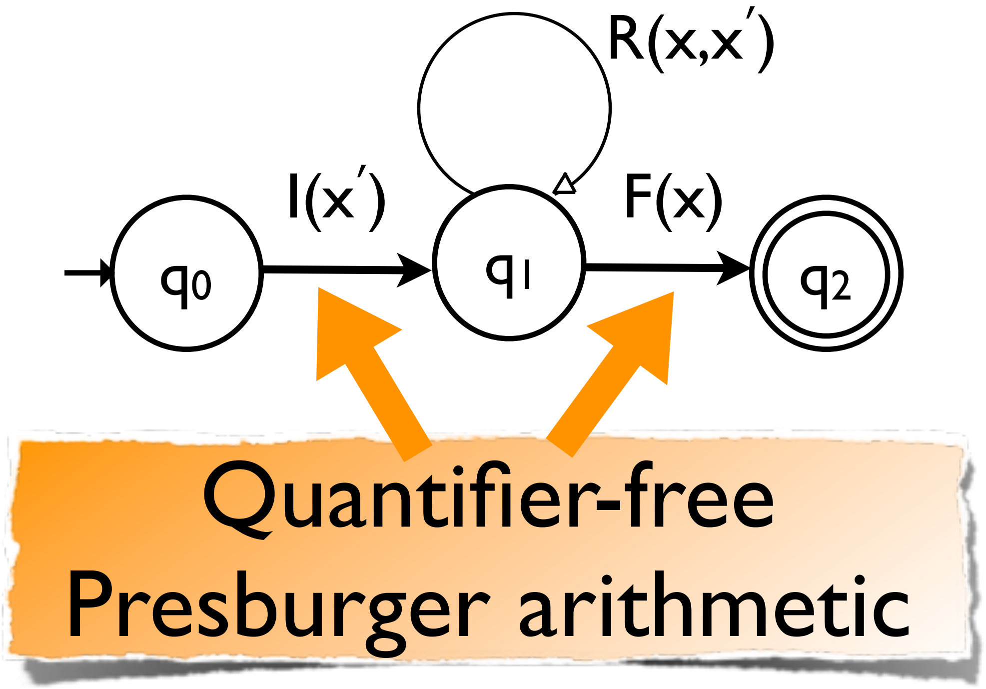
Quantifier-free
Presburger arithmetic

Periodicity and NTIME Safety



The program is safe
iff q_2 is unreachable

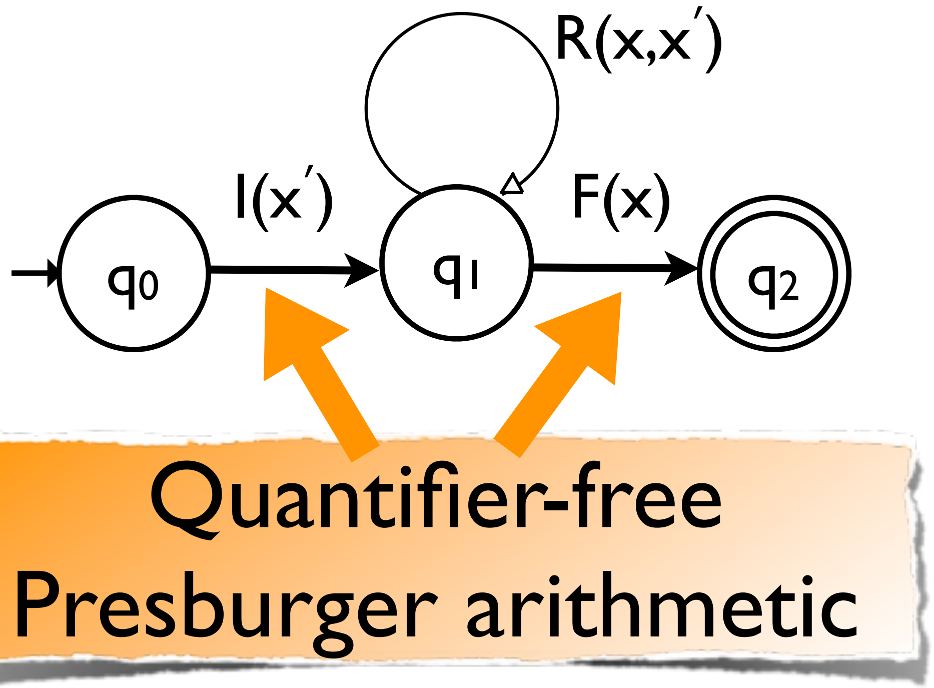
Periodicity and NTIME Safety



`guess` if $\exists k>0 \,.\, R^k = \varnothing$ holds

The program is safe
iff q_2 is unreachable

Periodicity and NTIME Safety



The program is safe iff q_2 is unreachable

`guess` if $\exists k>0 \,.\, R^k=\emptyset$ holds

yes ↓

`guess` $b>0$

`check` $R^{b-1} \neq \emptyset$ and $R^b = \emptyset$

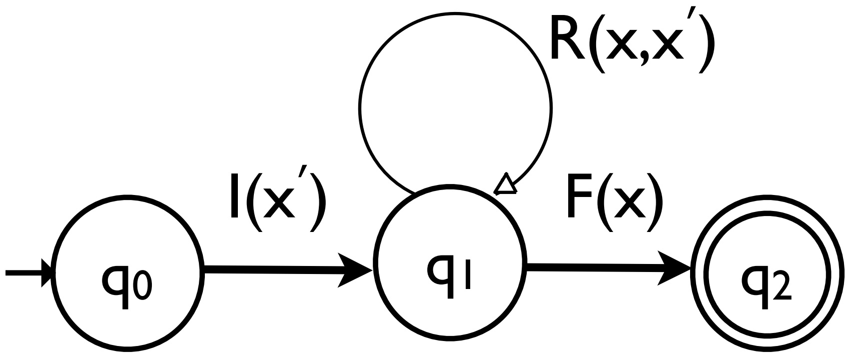
`compute` R^i `for some` $0 \leq i < b$

$I(x) \wedge R^i(x,x') \wedge F(x')$ `sat?`

yes ↓

unsafe

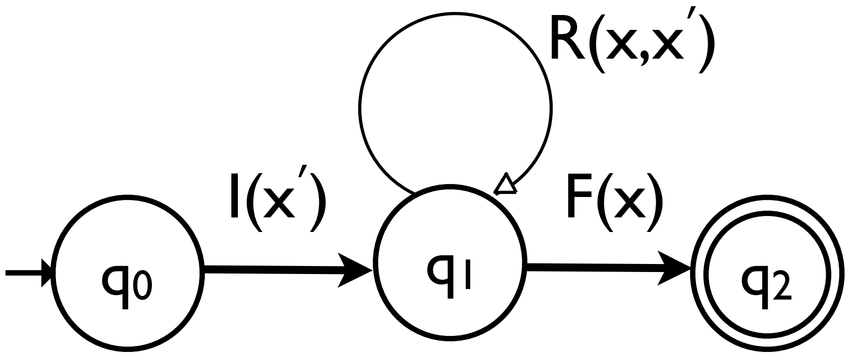
Periodicity and NTIME Safety



`guess` if $\exists k>0 \,.\, R^k = \emptyset$ holds

The program is safe
iff q_2 is unreachable

Periodicity and NTIME Safety



guess if $\exists k>0 \,.\, R^k=\emptyset$ holds

no

guess $b>0, c>0$

compute R^{b+j}, R^{b+c+j} for some $0\leq j<c$

compute Λ_j such that $R^{b+c+j} = R^{b+j} \oplus \Lambda_j$

check $\forall k\geq 0 \,.\, k\cdot\Lambda_j \oplus R^{b+j} \neq \emptyset$ and

$\forall k\geq 0 \,.\, (k\cdot\Lambda_j \oplus R^{b+j})\bullet R^c = (k+1)\cdot\Lambda_j \oplus R^{b+j}$

compute R^i for some $0\leq i<b$

$I \wedge [R^i \vee (k\geq 0 \wedge k\cdot\Lambda_j \oplus R^{b+j})] \wedge F$ sat?

yes

unsafe

The program is safe iff q_2 is unreachable

Computing EXP Powers in PTIME

Def. A class of relations is poly-logarithmic iff:

1. $||R^n||_2 = O((||R||_2 \cdot \log_2 n)^k)$, for some $k>0$
2. $P \bullet Q$ can be computed in $PTIME(||P||_2 + ||Q||_2)$

Computing EXP Powers in PTIME

Def. A class of relations is poly-logarithmic iff:

1. $||R^n||_2 = O((||R||_2 \cdot \log_2 n)^k)$, for some $k > 0$
2. $P \bullet Q$ can be computed in $PTIME(||P||_2 + ||Q||_2)$

```
FastPower(R,n)
  Q←R
  P←Id
  for i=1,...,⌈log₂ n⌉
    if the i-th bit of n is 1 then
      P←P•Q
    Q←Q•Q
  return P
```

Computing EXP Powers in PTIME

Def. A class of relations is poly-logarithmic iff:

1. $||R^n||_2 = O((||R||_2 \cdot \log_2 n)^k)$, for some $k>0$
2. $P \bullet Q$ can be computed in $PTIME(||P||_2 + ||Q||_2)$

```
FastPower(R,n)
  Q←R
  P←Id
  for i=1,...,⌈log₂ n⌉
    if the i-th bit of n is 1 then
      P←P•Q
    Q←Q•Q
  return P
```

If R is poly-logarithmic, R^k can be computed in PTIME(n), for $k=O(2^n)$

Deciding safety in NPTIME

Def. A periodic class of relations is exponential iff:

1. the prefix b and period c of any relation R are both $EXP(||R||_2)$
2. for all $0 \leq i < c$, if $R^{b+c+i} = R^{b+c} \oplus \Lambda_i$, the following conditions:
   - $\forall k \geq 0 \,.\, k \cdot \Lambda_i \oplus R^{b+i} \neq \emptyset$
   - $\forall k \geq 0 \,.\, (k \cdot \Lambda_j \oplus R^{b+j}) \bullet R^c = (k+1) \cdot \Lambda_j \oplus R^{b+j}$

can be checked in $NPTIME(||R||_2)$.

Deciding safety in NPTIME

Def. A periodic class of relations is exponential iff:

1. the prefix b and period c of any relation R are both $EXP(||R||_2)$
2. for all $0 \leq i < c$, if $R^{b+c+i} = R^{b+c} \oplus \Lambda_i$, the following conditions:
   - $\forall k \geq 0 \,.\, k \cdot \Lambda_i \oplus R^{b+i} \neq \emptyset$ (*-consistency)
   - $\forall k \geq 0 \,.\, (k \cdot \Lambda_j \oplus R^{b+j}) \bullet R^c = (k+1) \cdot \Lambda_j \oplus R^{b+j}$ (periodicity)

can be checked in $NPTIME(||R||_2)$.

Deciding safety in NPTIME

Def. A periodic class of relations is exponential iff:

1. the prefix b and period c of any relation R are both $EXP(||R||_2)$
2. for all $0 \leq i < c$, if $R^{b+c+i} = R^{b+c} \oplus \Lambda_i$, the following conditions:
   - $\forall k \geq 0 \,.\, k \cdot \Lambda_i \oplus R^{b+i} \neq \emptyset$ (*-consistency)
   - $\forall k \geq 0 \,.\, (k \cdot \Lambda_j \oplus R^{b+j}) \bullet R^c = (k+1) \cdot \Lambda_j \oplus R^{b+j}$ (periodicity)

can be checked in $NPTIME(||R||_2)$.

If R is exponential, all branches of the non-deterministic decision procedure for safety take $PTIME(||R||_2)$. Then:

- $||I(x) \wedge R^i(x,x') \wedge F(x')||_2 = O((||I||_2 + ||R||_2 + ||F||_2)^k)$
- $||I(x) \wedge [R^i \vee (k \geq 0 \wedge k \cdot \Lambda_j \oplus R^{b+j})] \wedge F(x')||_2 = O((||I||_2 + ||R||_2 + ||F||_2)^k)$

for some k>0.

Deciding safety in NPTIME

Def. A periodic class of relations is exponential iff:

1. the prefix b and period c of any relation R are both $EXP(||R||_2)$
2. for all $0 \leq i < c$, if $R^{b+c+i} = R^{b+c} \oplus \Lambda_i$, the following conditions:
   - $\forall k \geq 0 \, . \, k \cdot \Lambda_i \oplus R^{b+i} \neq \emptyset$ (*-consistency)
   - $\forall k \geq 0 \, . \, (k \cdot \Lambda_j \oplus R^{b+j}) \bullet R^c = (k+1) \cdot \Lambda_j \oplus R^{b+j}$ (periodicity)

can be checked in $NPTIME(||R||_2)$.

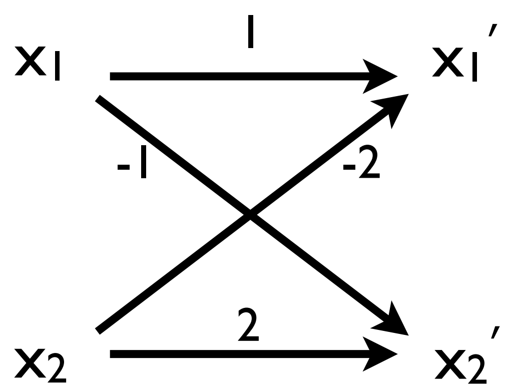
If R is exponential, all branches of the non-deterministic decision procedure for safety take $PTIME(||R||_2)$. Then:

- $||I(x) \wedge R^i(x,x') \wedge F(x')||_2 = O((||I||_2 + ||R||_2 + ||F||_2)^k)$
- $||I(x) \wedge [R^i \vee (k \geq 0 \wedge k \cdot \Lambda_j \oplus R^{b+j})] \wedge F(x')||_2 = O((||I||_2 + ||R||_2 + ||F||_2)^k)$

for some k>0.

Since these are quantifier-free Presburger formulae, then SAT (and also safety) is in $NPTIME(||I||_2 + ||R||_2 + ||F||_2)$!

NP *-Consistency and Periodicity

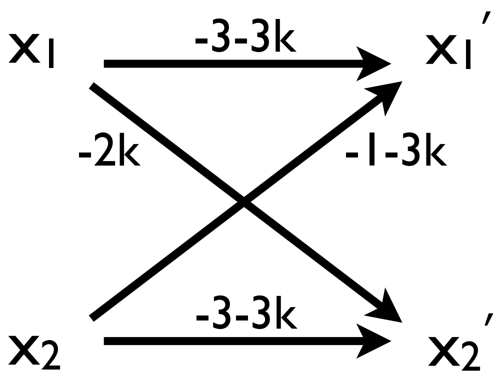


prefix b = 2

period c = 2

$$\text{rate } \Lambda = \begin{matrix} 0 & \infty & -3 & -2 \\ \infty & 0 & -3 & -3 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix}$$

NP *-Consistency and Periodicity



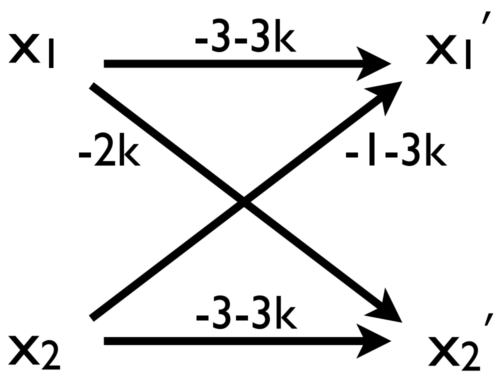
$\{R^{2+2k}\}_{k\geq 0}$

prefix b = 2

period c = 2

rate $\Lambda = \begin{matrix} 0 & \infty & -3 & -2 \\ \infty & 0 & -3 & -3 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix}$

NP *-Consistency and Periodicity



$$\{R^{2+2k}\}_{k\geq 0}$$

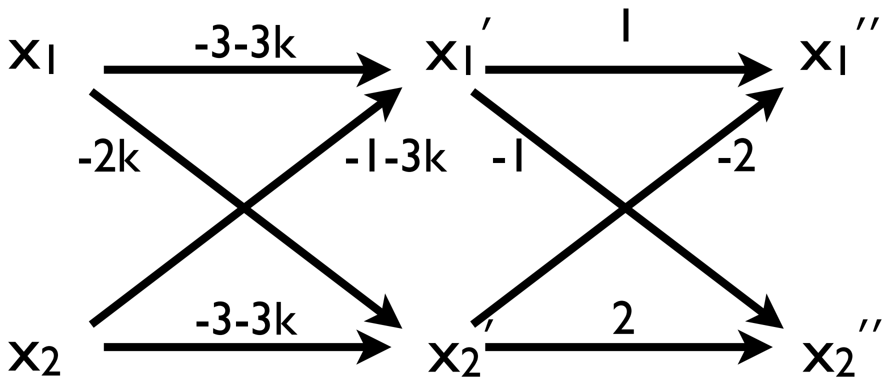
prefix b = 2

period c = 2

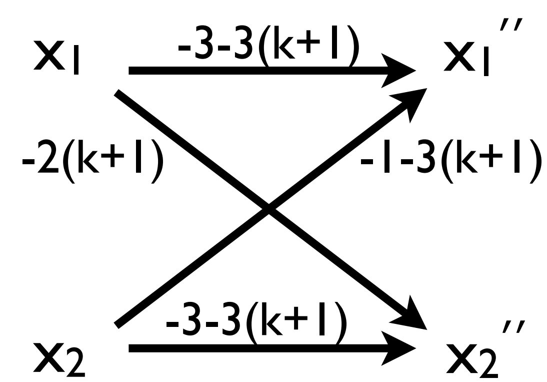
$$\text{rate } \Lambda = \begin{matrix} 0 & \infty & -3 & -2 \\ \infty & 0 & -3 & -3 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix}$$

$$\forall k\geq 0 \,.\, k\cdot\Lambda \oplus R^b \neq \emptyset \quad ?$$

NP *-Consistency and Periodicity



? ≡



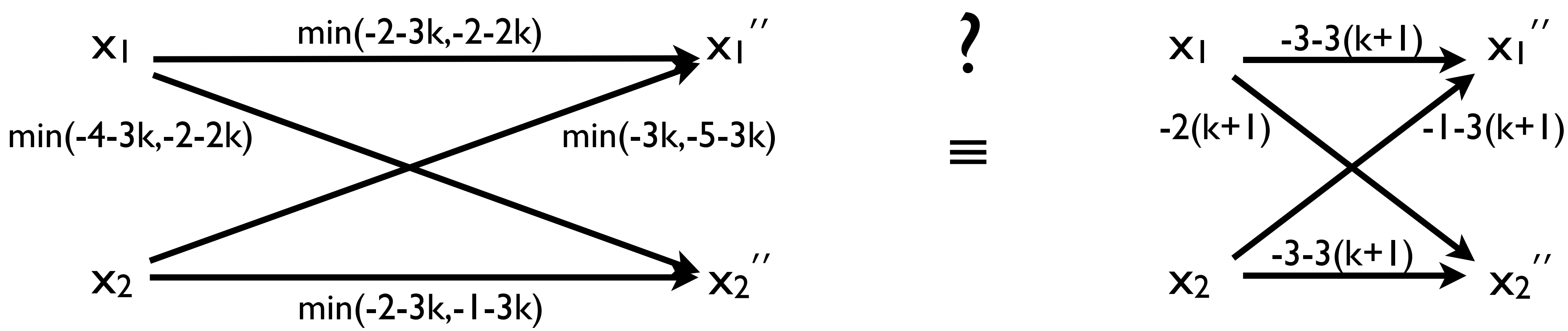
prefix b = 2

period c = 2

rate Λ =
$$\begin{matrix} 0 & \infty & -3 & -2 \\ \infty & 0 & -3 & -3 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix}$$

$$\forall k \geq 0 \;.\; (k \cdot \Lambda \oplus R^b) \bullet R^c = (k+1) \cdot \Lambda \oplus R^b \;?$$

NP *-Consistency and Periodicity



prefix b = 2

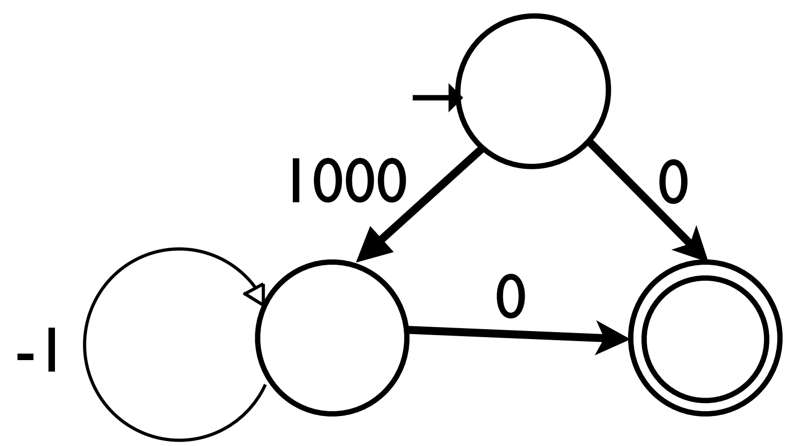
period c = 2

rate $\Lambda = \begin{matrix} 0 & \infty & -3 & -2 \\ \infty & 0 & -3 & -3 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{matrix}$

$$\forall k \geq 0 \,.\, (k \cdot \Lambda \oplus R^b) \bullet R^c = (k+1) \cdot \Lambda \oplus R^b \ ?$$

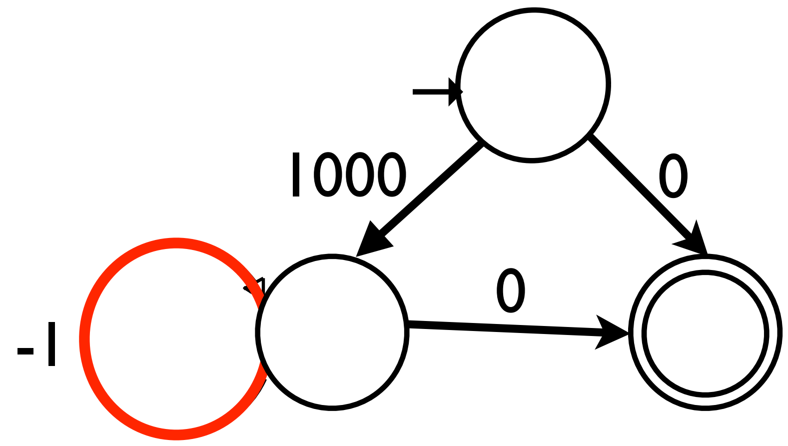
Bounding the Prefix

Thm. Given a weighted graph G with n nodes, the weights of the minimal paths between two vertices form a periodic sequence with prefix at most $\max(n^4, n^6 \cdot M)$, where M is the maximum absolute value among the labels of G.



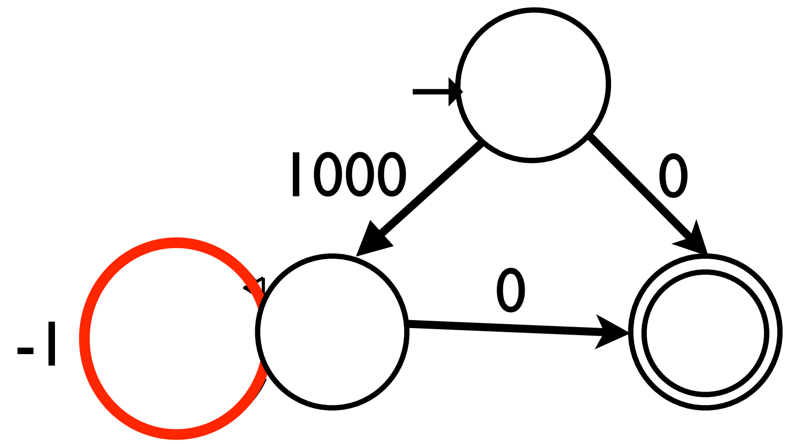
Bounding the Prefix

Thm. Given a weighted graph G with n nodes, the weights of the minimal paths between two vertices form a periodic sequence with prefix at most max(n^4, $n^6 \cdot M$), where M is the maximum absolute value among the labels of G.



Bounding the Prefix

Thm. Given a weighted graph G with n nodes, the weights of the minimal paths between two vertices form a periodic sequence with prefix at most $\max(n^4, n^6 \cdot M)$, where M is the maximum absolute value among the labels of G.



A zigzag automaton has at most $5^N = 2^{O(N)}$ states, where N is the number of dimensions of the DB relation $R \subseteq Z^N \times Z^N$

➡ states are N-tuples from the set $\{\rightarrow, \leftarrow, \langle, \rangle, \perp\}$, of cardinality 5

➡ the absolute values of the labels are of the order of $2^{O(||R||_2)}$

Bounding the Period

<u>Thm.[deSchutter00]</u> Given a weighted graph G, and a partition of G in SCCs W_1, ..., W_k, the weights of the minimal paths between two vertices form a periodic sequence of period lcm(c_1, ..., c_k):

- $c_i = \gcd \{ |\rho| \mid \rho$ is a critical cycle in $W_i\}$, for all i=1,...,k.

Bounding the Period

Thm.[deSchutter00] Given a weighted graph G, and a partition of G in SCCs $W_1, \ldots, W_k$, the weights of the minimal paths between two vertices form a periodic sequence of period $\mathrm{lcm}(c_1, \ldots, c_k)$:

- $c_i = \gcd \{ |\rho| \mid \rho$ is a critical cycle in $W_i\}$, for all $i=1,\ldots,k$.

Every SCC of a zigzag automaton A has a critical cycle ρ of length:

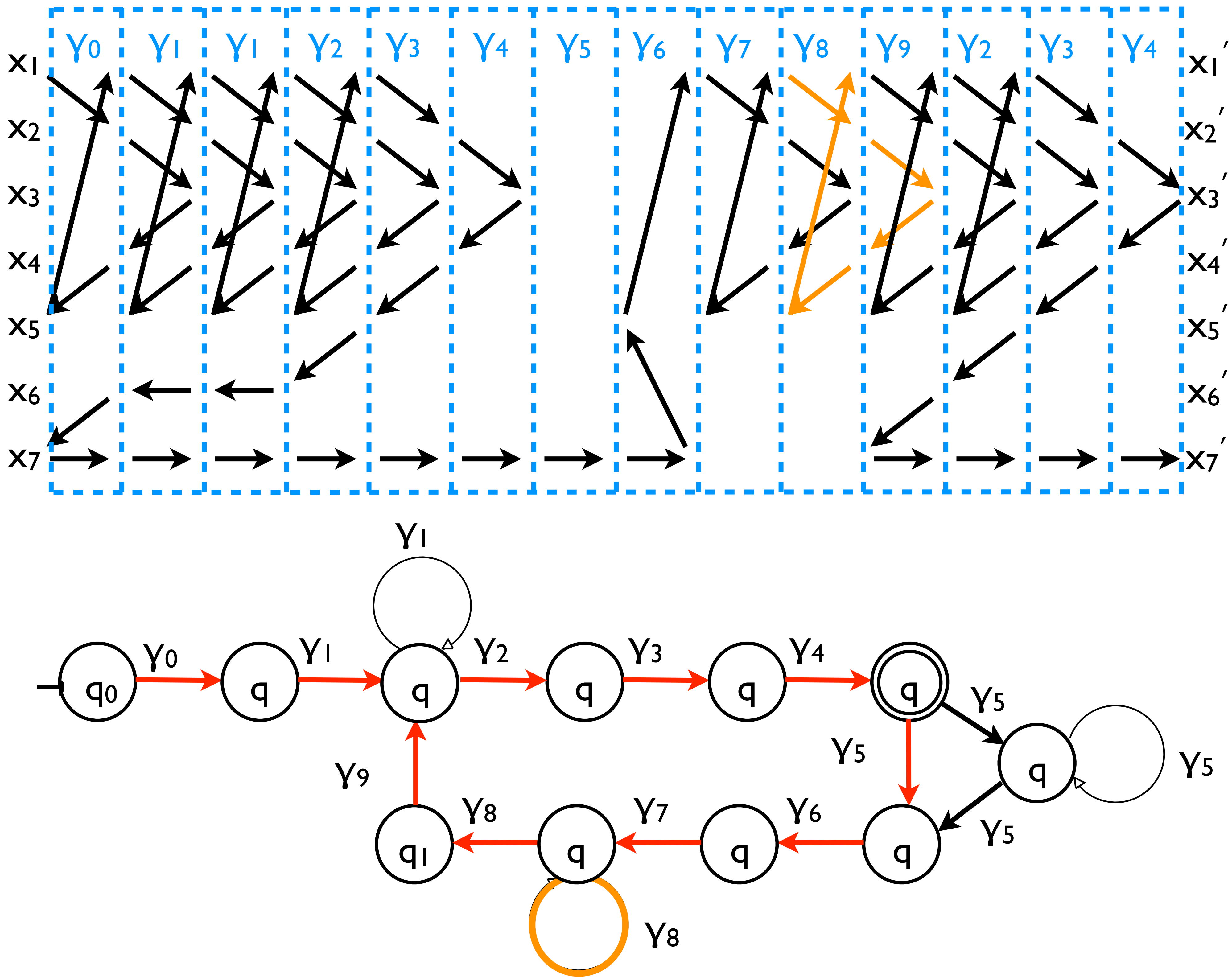
$$|\rho| \mid \mathrm{lcm}(1,\ldots,N)$$

where $R \subseteq Z^N \times Z^N$ is the DB relation for A

➡ c_i divides $\mathrm{lcm}(1,\ldots,N)$, for all $i = 1,\ldots,k$

➡ the period is at most $\mathrm{lcm}(1,\ldots,N) = 2^{O(N)} = 2^{O(||R||_2)}$

Bounding the Period

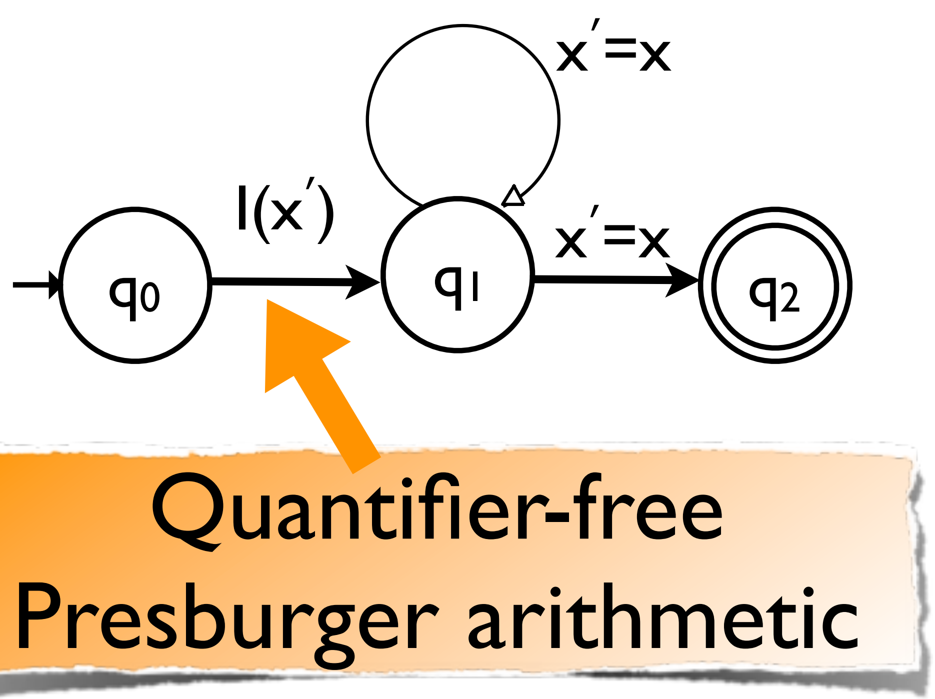


NP-complete Safety for DB Loops

- Difference bounds relations are exponential
  - ➡ the prefix and period of R are of the order of $2^{O(||R||_2)}$
- Safety of flat integer programs with DB loops is in NP
- NP-hardness is by reduction from satisfiability of Quantifier-free Presburger Arithmetic

NP-complete Safety for DB Loops

- Difference bounds relations are exponential
  - ➡ the prefix and period of R are of the order of $2^{O(||R||_2)}$
- Safety of flat integer programs with DB loops is in NP
- NP-hardness is by reduction from satisfiability of Quantifier-free Presburger Arithmetic



Octagonal Relations

- Octagonal relations are encoded as DB relations on twice the number of dimensions

$$x + y' \leq 1 \quad \equiv \quad \begin{array}{l} x_+ - y_-' \leq 1 \\ y_+' - x_- \leq 1 \end{array}$$

Octagonal Relations

- Octagonal relations are encoded as DB relations on twice the number of dimensions

$$x + y' \leq 1 \quad \equiv \quad \begin{array}{l} x_{+} - y_{-}' \leq 1 \\ y_{+}' - x_{-} \leq 1 \end{array}$$

$$x_{+} + x_{-} = 0$$
$$y_{+}' + y_{-}' = 0$$

Octagonal Relations

- Octagonal relations are encoded as DB relations on twice the number of dimensions

$$x + y' \leq 1 \quad \equiv \quad \begin{array}{l} x_+ - y_-' \leq 1 \\ y_+' - x_- \leq 1 \end{array}$$

$x_+ + x_- = 0$
$y_+' + y_-' = 0$

- Closed under relational composition:
  - ➡ composition of octagonal relations requires an additional tightening step
- Oct. relations are periodic, poly-logarithmic and exponential
  - ➡ the prefix and period of R are also of the order of $2^{O(||R||_2)}$
- Safety problems are NP-complete for integer flat programs with octagonal loops

Conclusions

- Safety can be decided for integer programs whenever:
  - ➡ there are no nested loops in the control structure
  - ➡ all loops are labeled with relations definable by octagonal constraints
- The safety problems are NP-complete in these cases
- We have implemented an efficient algorithm [BIK'10]:
  - ➡ function summarization in inter-procedural analysis
  - ➡ abstraction refinement for interpolation-based model checking
  - ➡ termination analysis
  - ➡ analysis of programs with integer arrays

Conclusions

- Safety can be decided for integer programs whenever:
  - ➡ there are no nested loops in the control structure
  - ➡ all loops are labeled with relations definable by octagonal constraints
- The safety problems are NP-complete in these cases
- We have implemented an efficient algorithm [BIK'10]:
  - ➡ function summarization in inter-procedural analysis
  - ➡ abstraction refinement for interpolation-based model checking
  - ➡ termination analysis
  - ➡ analysis of programs with integer arrays

http://nts.imag.fr/index.php/Flata