3.5 Komplexität nebenläufiger Systeme

3.5.1 Überdeckungsgraph

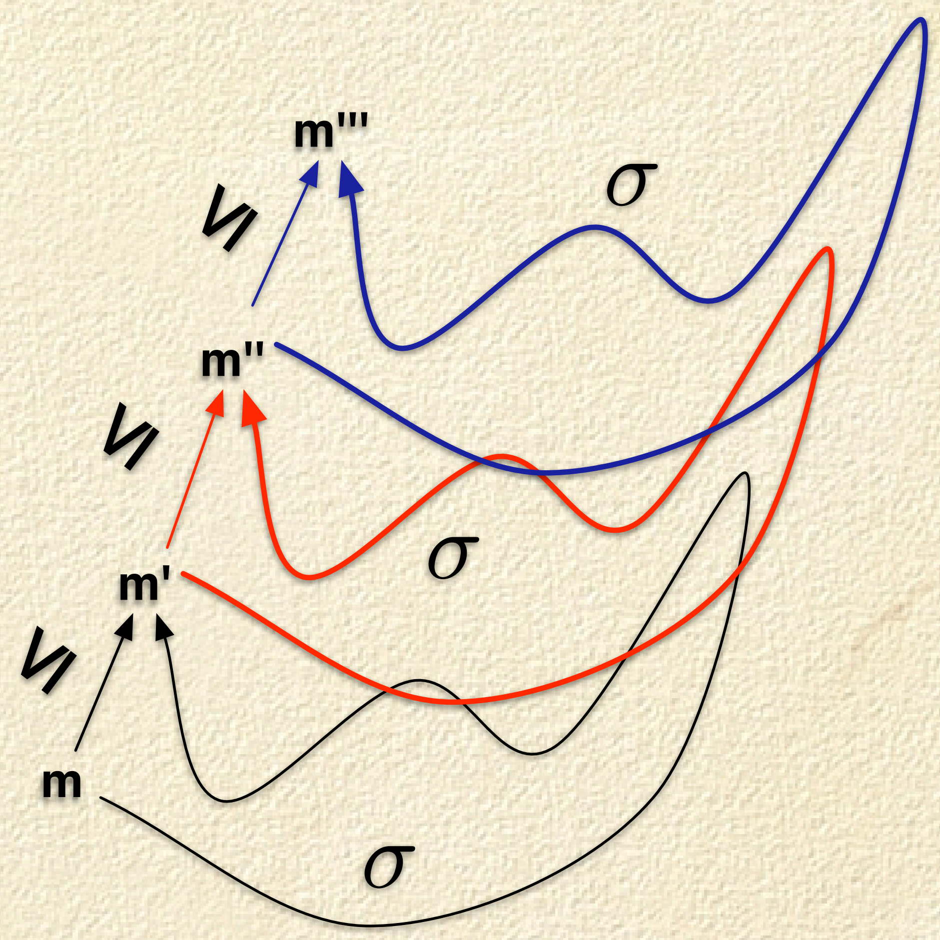
3.5.2 Turing-Mächtigkeit

3.5.3 Komplexität

3.5.1 Überdeckungsgraph

a) $\exists \sigma \in T^* : \mathbf{m} \xrightarrow{\sigma} \mathbf{m}'$

b) $\mathbf{m} \lneq \mathbf{m}'$



Definition 3.22 *Es sei $\mathbb{N}_\omega := \mathbb{N} \cup \{\omega\}$ zusammen mit folgenden Rechenregeln:*

$$\forall n \in \mathbb{N} : \omega > n;$$

$$\forall n \in \mathbb{N}_\omega : \omega + n = \omega - n := \omega;$$

$$\forall n \in \mathbb{N}\backslash\{0\} : n \cdot \omega = \omega \cdot n := \omega; 0 \cdot \omega = \omega \cdot 0 := 0.$$

Ein Vektor $\mathbf{m} \in \mathbb{N}_\omega^P$ *wird* Pseudomarkierung *genannt, wenn in ihm das Symbol* ω *vorkommt, und für diese wird die Schaltregel formal übernommen. Eine Pseudomarkierung* $\mathbf{m}$ *entspricht einer gewöhnlichen (Teil-) Markierung auf den Plätzen* s *mit* $\mathbf{m}(s) \neq \omega$*, wobei die mit* ω *besetzten Komponenten beliebig sind und unberücksichtigt bleiben.*

Für Vektoren $\mathbf{m}_1, \mathbf{m}_2 \in \mathbb{Z}^r$ *seien die Operatoren* $+, -, =$ $\leq$ *jeweils komponentenweise erklärt, d.h.,* $\mathbf{m}_1 \leq \mathbf{m}_2$*, falls* $\forall s \in S : \mathbf{m}_1(s) \leq \mathbf{m}_2(s)$*. Lediglich* $\mathbf{m}_1 < \mathbf{m}_2$ *steht für* $(\mathbf{m}_1 \leq \mathbf{m}_2$ *und* $\mathbf{m}_1 \neq \mathbf{m}_2)$*.*

$$\begin{pmatrix} 0 \\ 20 \\ 1 \end{pmatrix} \leq \begin{pmatrix} 0 \\ \omega \\ 2 \end{pmatrix} \qquad \begin{pmatrix} 0 \\ 20 \\ 1 \end{pmatrix} < \begin{pmatrix} 0 \\ \omega \\ 2 \end{pmatrix}$$

$$X(\mathbf{m}')$$

$$\mathbf{m}_2'' = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 3 \end{pmatrix} \qquad \mathbf{m}_3'' = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 1 \end{pmatrix}$$

$$\mathbf{m}_1'' = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 3 \end{pmatrix}$$

$$X(\mathbf{m}') := \{\mathbf{m}'' \in V \,|\, \mathbf{m}'' \leq \mathbf{m}' \text{ und } \mathbf{m}'' \xrightarrow{*} \mathbf{m}\}$$

$$\mathbf{m}$$

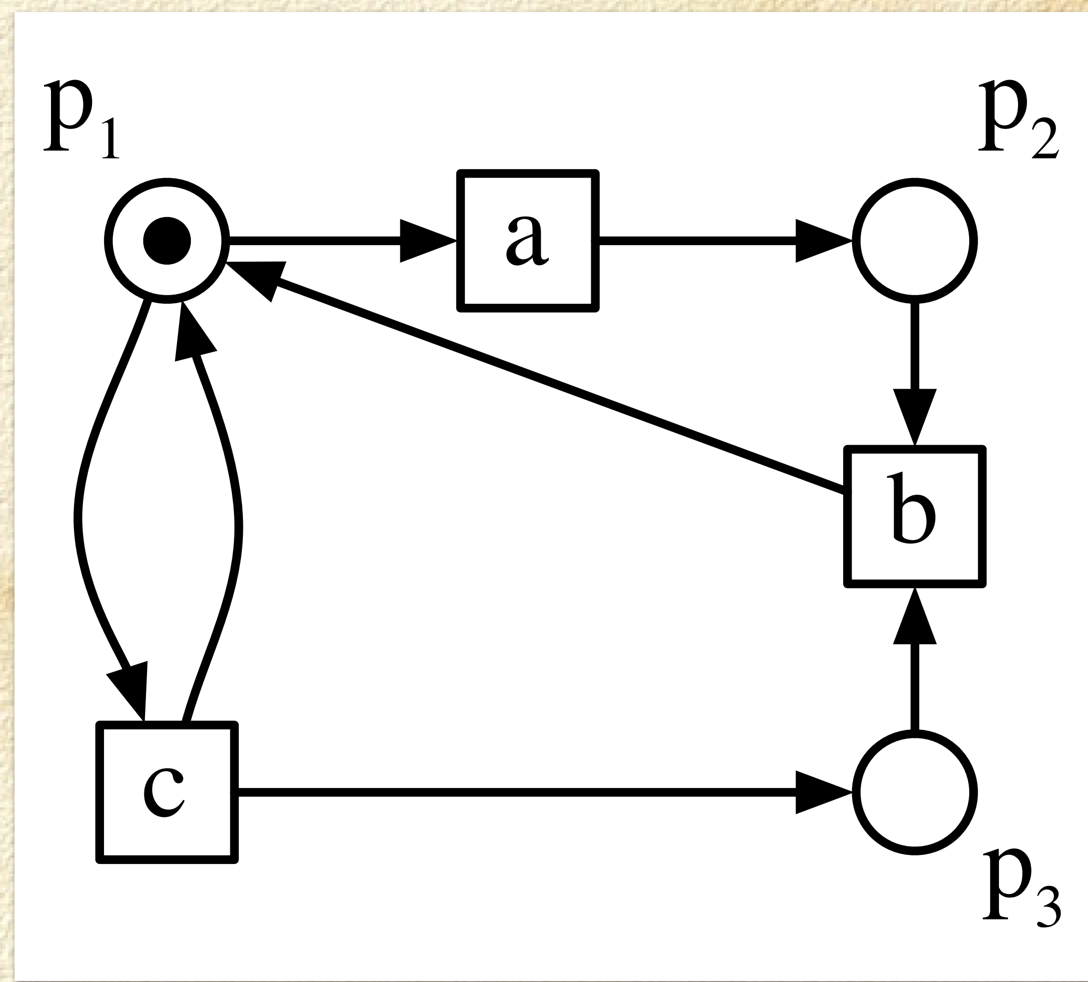
$$\downarrow$$

$$\mathbf{m}' = \cancel{\begin{pmatrix} 1 \\ 2 \\ 1 \\ 3 \end{pmatrix}} \quad \begin{pmatrix} \omega \\ \omega \\ 1 \\ \omega \end{pmatrix}$$

Algorithmus 3.4 (Berechnung des Überdeckungsgraph)

Input - Das P/T-Netz $\mathcal{N} = \langle P, T, F, W, \mathbf{m}_0 \rangle$
Output - Der Überdeckungsgraph Graph $G(\mathcal{N}) = (V, E)$.

1. Initialisiere $G(\mathcal{N}) = (\{\mathbf{m_0}\}, \emptyset)$; $\mathbf{m_0}$ sei ungefärbt;
2. **while** Es gibt ungefärbte Knoten in V **do**
   2.1 Wähle einen ungefärbte Knoten $\mathbf{m} \in V$ und färbe ihn.
   2.2 **for** Für jede in $\mathbf{m}$ aktivierte Transition t **do**
      2.2.1 Berechne $\mathbf{m}'$ mit $\mathbf{m} \xrightarrow{t} \mathbf{m}'$ und $X(\mathbf{m}') := \{\mathbf{m}'' \in V | \mathbf{m}'' \leq \mathbf{m}' \text{ und } \mathbf{m}'' \xrightarrow{*} \mathbf{m}'\}$;
      2.2.2 **if** $X(\mathbf{m}') \neq \emptyset$ **then** $\mathbf{m}_1(p) := \begin{cases} \omega, & \exists \mathbf{m}'' \in X(\mathbf{m}') : \mathbf{m}''(p) < \mathbf{m}'(p) \\ \mathbf{m}'(p), & \text{sonst.} \end{cases}$
      else $\mathbf{m}_1 := \mathbf{m}'$;
      2.2.3 **if** $\mathbf{m}_1 \notin V$
      then $V := V \cup \{\mathbf{m}_1\}$, wobei $\mathbf{m}_1$ ein ungefärbter Knoten sei. ;
      2.2.4 $E := E \cup \{\langle \mathbf{m}, t, \mathbf{m}_1 \rangle\}$;
3. Der Algorithmus terminiert mit Ergebnis. ($G(\mathcal{N})$ ist der Überdeckungsgraph.)



$$
\begin{array}{ccc}
a & b & c
\end{array}
$$
$$
\begin{pmatrix} \boxed{-1} & 1 & \boxed{0} \\ \boxed{1} & -1 & \boxed{0} \\ \boxed{0} & -1 & \boxed{1} \end{pmatrix}
$$

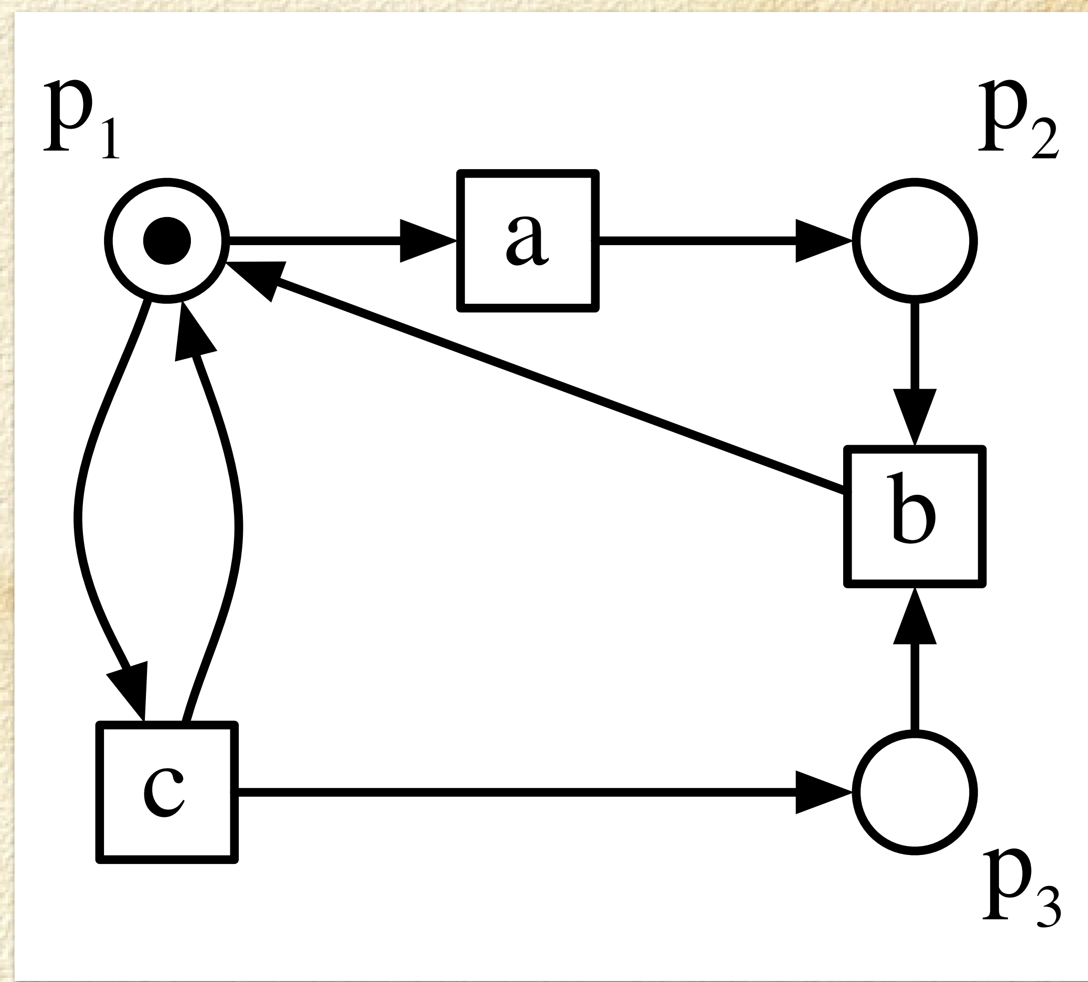
$$
\begin{tikzcd}
 & (1,0,0) \arrow[dl, "a"'] \arrow[dr, "c"] & \\
(0,1,0) & & \cancel{(1,0,1)} \\
 & & (1,0,\omega)
\end{tikzcd}
$$

ω?

ω?

$$
\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \boxed{\begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}
$$

$$
\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \boxed{\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}
$$

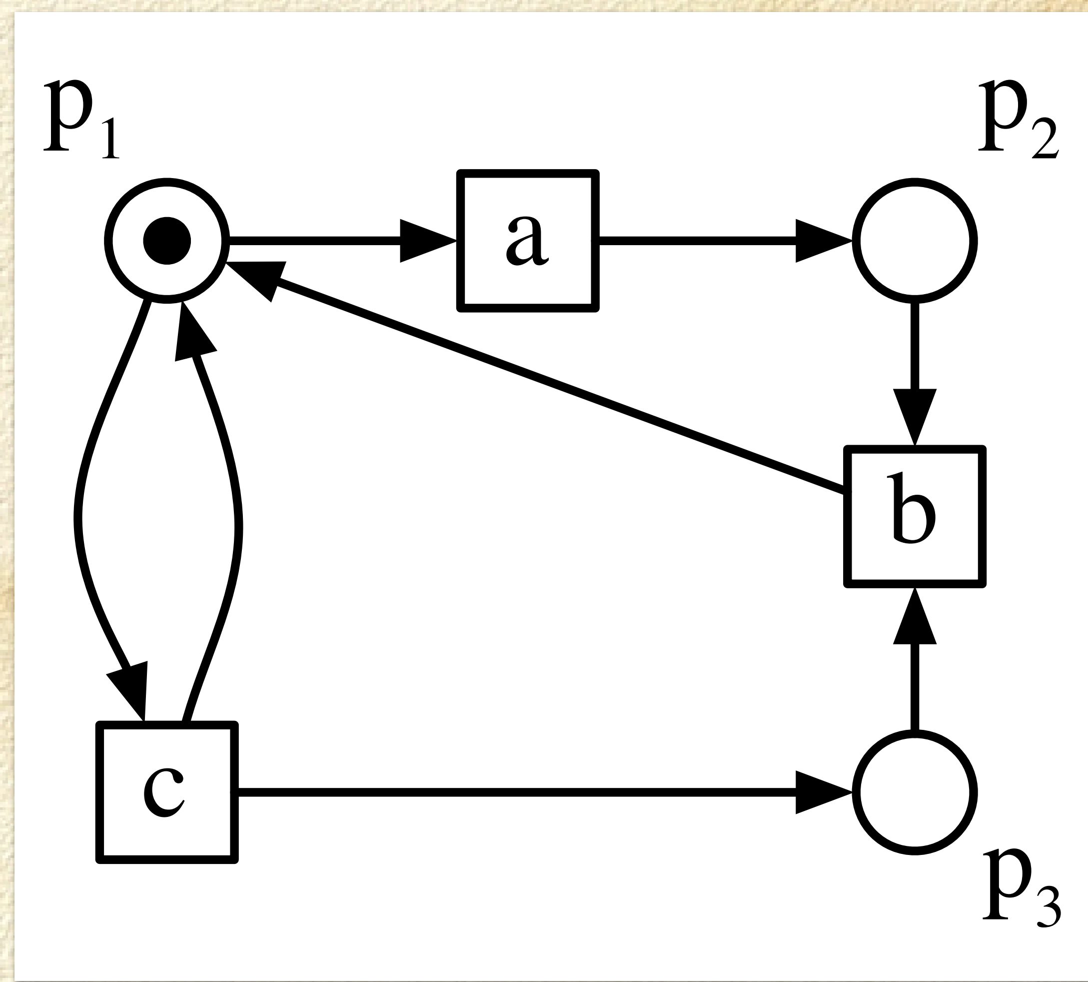


$$
\begin{array}{ccc}
a & b & c \\
\left(\begin{array}{ccc}
-1 & 1 & 0 \\
1 & -1 & 0 \\
0 & -1 & 1
\end{array}\right)
\end{array}
$$

$$
\begin{tikzcd}
 & (1,0,0) \arrow[dl, "a"'] \arrow[dr, "c"] & & \\
(0,1,0) & & \begin{matrix}\cancel{(1,0,1)} \\ (1,0,\omega)\end{matrix} \arrow[dl, "a"'] \arrow[dr, "c"] & \\
 & (0,1,\omega) & & (1,0,\omega)
\end{tikzcd}
$$

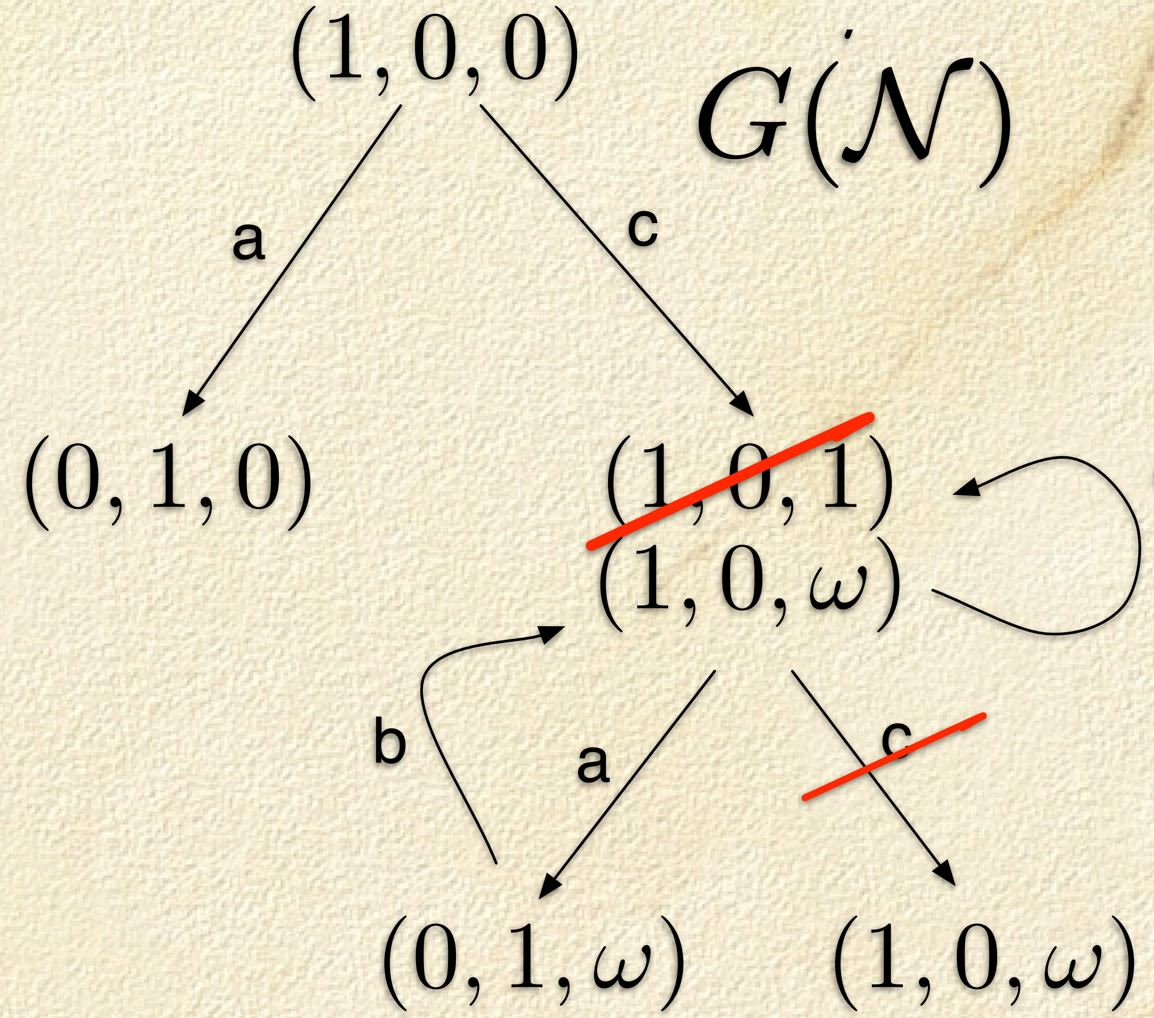
$$\begin{pmatrix}1\\0\\\omega\end{pmatrix} + \begin{pmatrix}-1\\1\\0\end{pmatrix} = \begin{pmatrix}0\\1\\\omega\end{pmatrix}$$

$$\begin{pmatrix}1\\0\\\omega\end{pmatrix} + \begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}1\\0\\\omega\end{pmatrix}$$

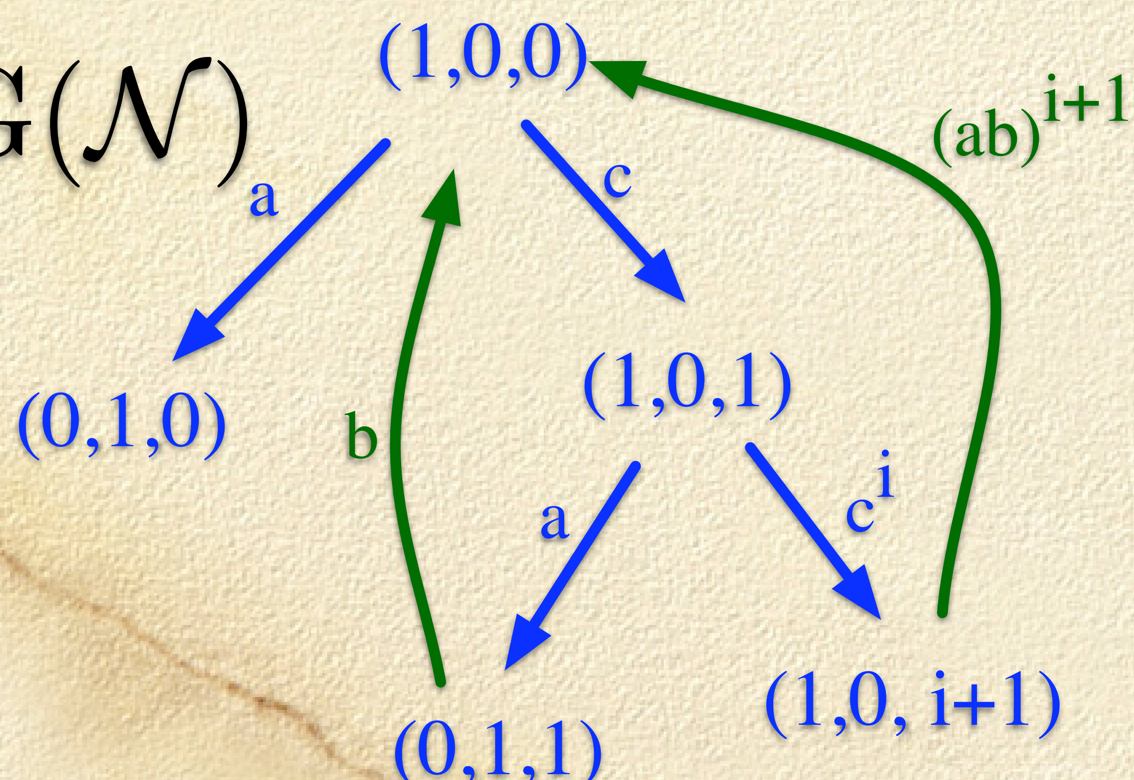


$$
\begin{array}{ccc}
a & b & c \\
\end{array}
$$
$$
\begin{pmatrix}
-1 & 1 & 0 \\
1 & -1 & 0 \\
0 & -1 & 1
\end{pmatrix}
$$

$G(\mathcal{N})$



$\mathrm{RG}(\mathcal{N})$

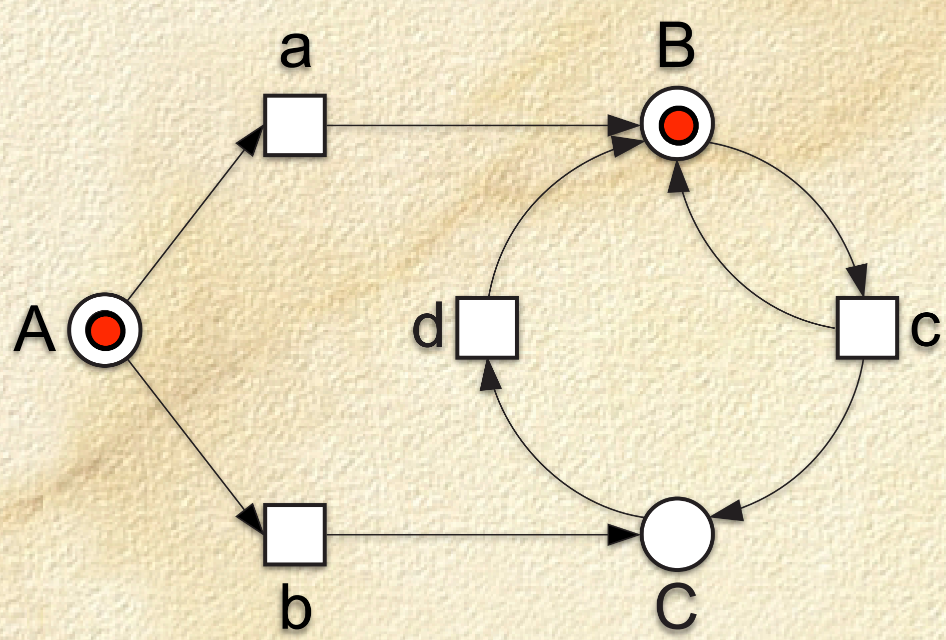


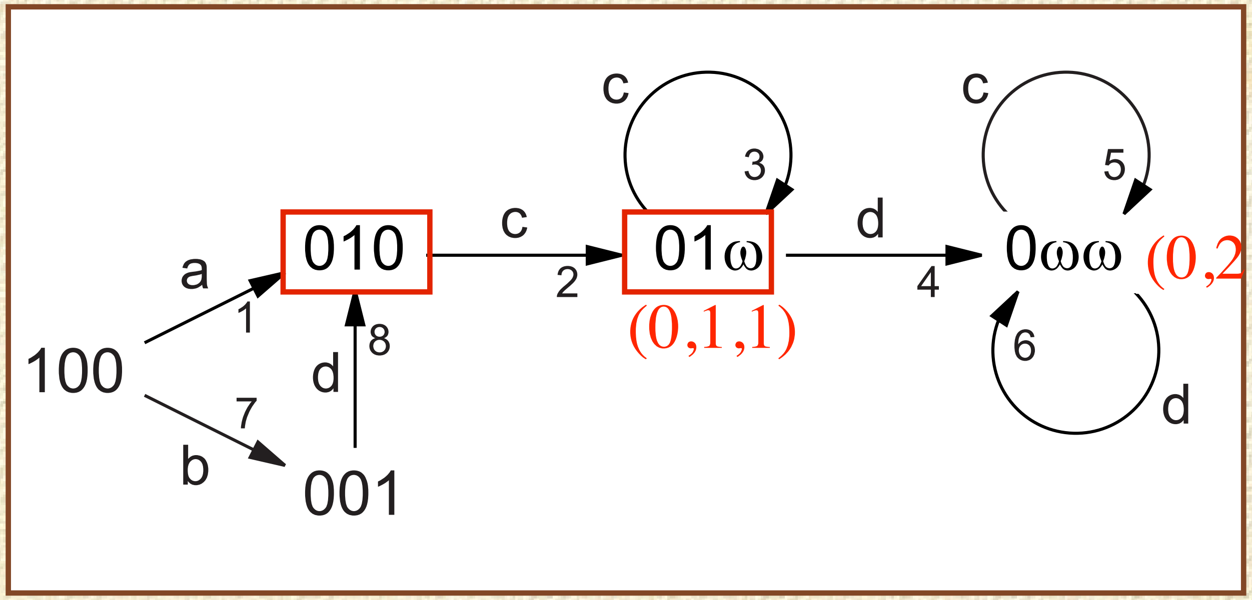
Algorithmus 3.4 (Berechnung des Überdeckungsgraph)

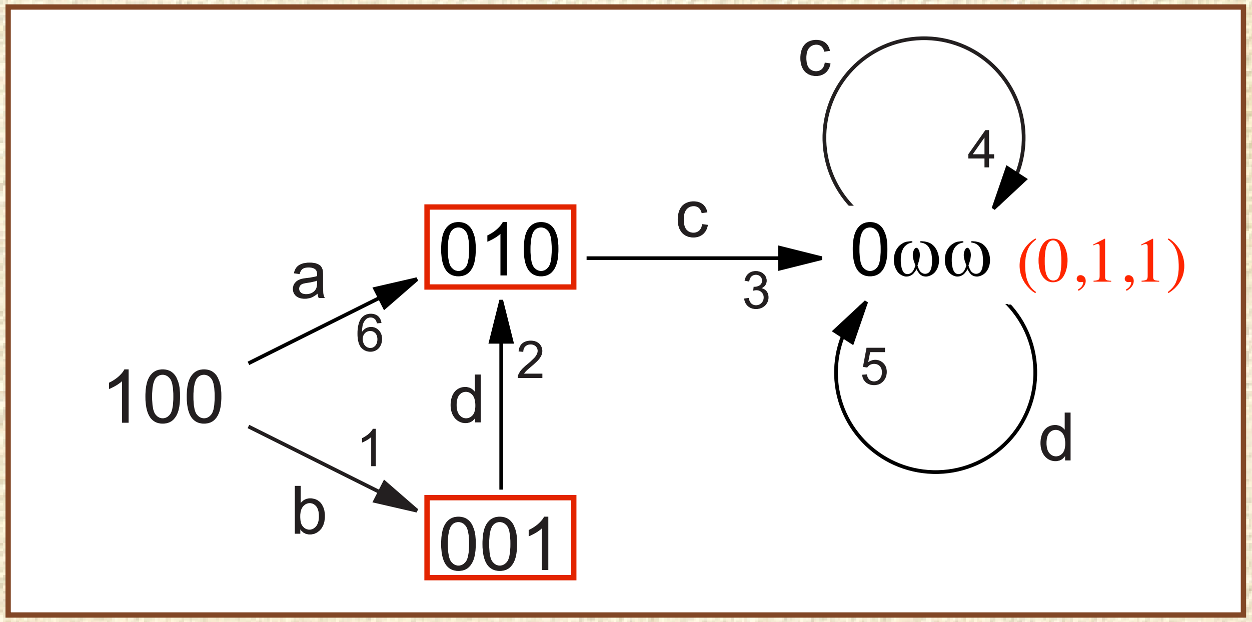
Input - Das P/T-Netz $\mathcal{N} = \langle P, T, F, W, \mathbf{m}_0 \rangle$
Output - Der Überdeckungsgraph Graph $G(\mathcal{N}) = (V, E)$.

1. Initialisiere $G(\mathcal{N}) = (\{\mathbf{m_0}\}, \emptyset)$; $\mathbf{m_0}$ sei ungefärbt;
2. **while** Es gibt ungefärbte Knoten in V **do**
   2.1 Wähle einen ungefärbte Knoten $\mathbf{m} \in V$ und färbe ihn.
   2.2 **for** Für jede in $\mathbf{m}$ aktivierte Transition t **do**
      2.2.1 Berechne $\mathbf{m}'$ mit $\mathbf{m} \xrightarrow{t} \mathbf{m}'$ und $X(\mathbf{m}') := \{\mathbf{m}'' \in V | \mathbf{m}'' \leq \mathbf{m}'$ und $\mathbf{m}'' \xrightarrow{*} \mathbf{m}'\}$;
      2.2.2 **if** $X(\mathbf{m}') \neq \emptyset$ **then** $\mathbf{m}_1(p) := \begin{cases} \omega, & \exists \mathbf{m}'' \in X(\mathbf{m}') : \mathbf{m}''(p) < \mathbf{m}'(p) \\ \mathbf{m}'(p), & \text{sonst.} \end{cases}$
      else $\mathbf{m}_1 := \mathbf{m}'$;
      2.2.3 **if** $\mathbf{m}_1 \notin V$
      then $V := V \cup \{\mathbf{m}_1\}$, wobei $\mathbf{m}_1$ ein ungefärbter Knoten sei. ;
      2.2.4 $E := E \cup \{\langle \mathbf{m}, t, \mathbf{m}_1 \rangle\}$;
3. Der Algorithmus terminiert mit Ergebnis. ($G(\mathcal{N})$ ist der Überdeckungsgraph.)

der Überdeckungsgraph oder ein Überdeckungsgraph?





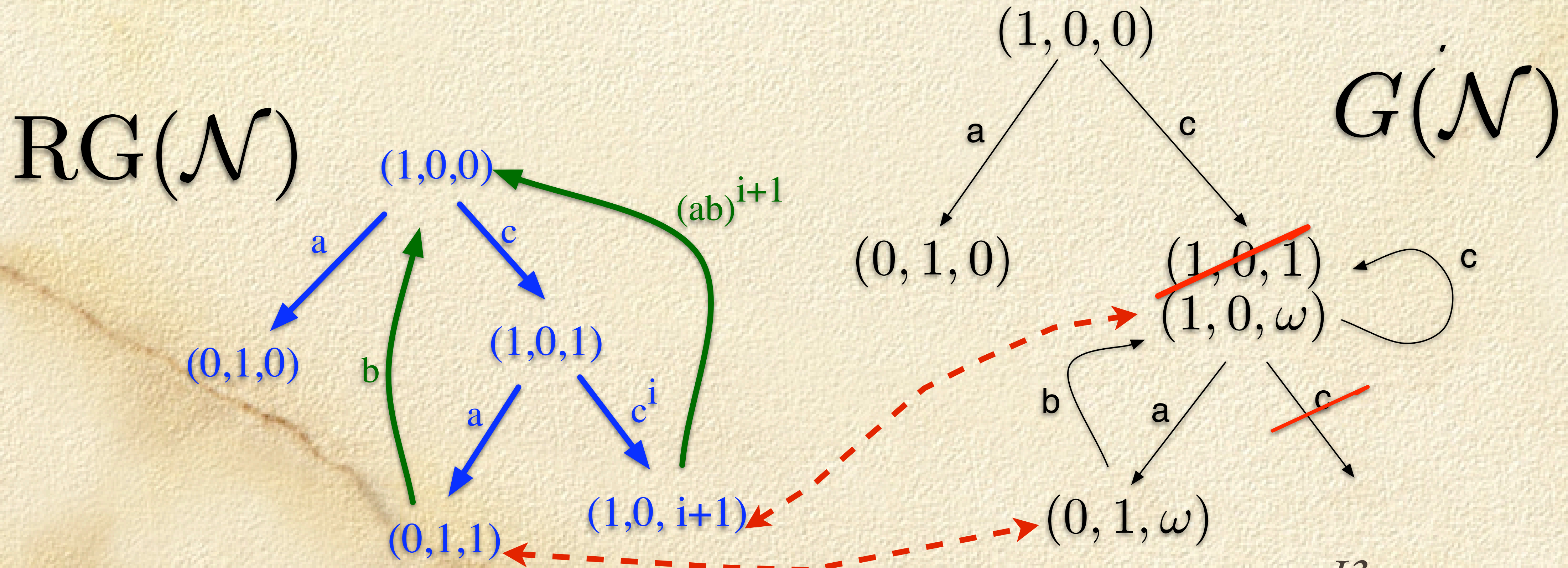


$X(\mathbf{m}')$

Die Bedeutung des Überdeckungsgraphen ergibt sich aus folgenden Eigenschaften:

Satz 3.23 *Sei $\mathcal{N} = \langle P, T, F, W, \mathbf{m}_0 \rangle$ ein P/T-Netz und $G(\mathcal{N}) = (V, E)$ ein Überdeckungsgraph zu $\mathcal{N}$, dann ist ein Platz $p \in P$ genau dann* beschränkt, *wenn es keinen Knoten $\mathbf{m} \in V$ mit $\mathbf{m}(p) = \omega$ gibt.*

Gilt $\mathbf{m}_0 \xrightarrow{w} \mathbf{m}$ im Netz $\mathcal{N}$ für ein Wort $w \in T^*$, so gibt es in $G(\mathcal{N})$ einen Knoten $\mathbf{m}_1$ mit $\mathbf{m}_1 \geq \mathbf{m}$ und $\mathbf{m}_0 \xrightarrow[w]{*} \mathbf{m}_1$.



Satz 3.24 *Der Algorithmus 3.4 zur Konstruktion von* $G(\mathcal{N})$ *terminiert.*

Satz 3.26 *Für ein P/T-Netz* $\mathcal{N}$ *ist* $\mathrm{RG}(\mathcal{N})$ *genau dann endlich, wenn kein Knoten von* $G(\mathcal{N})$ *eine* ω*-Komponente besitzt.*

3.5.2 Turing-Mächtigkeit *von Petrinetzen*

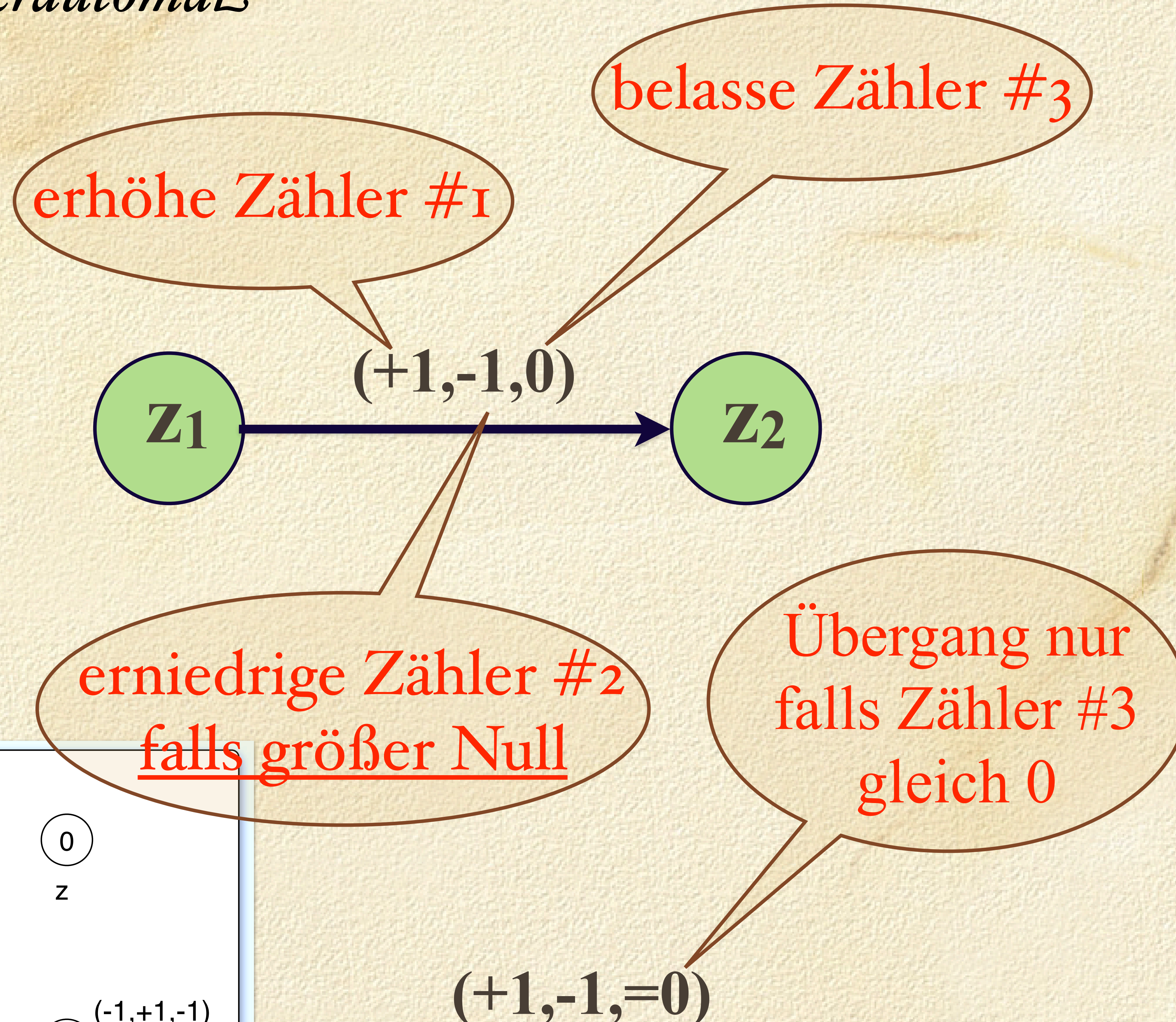
Turing-Maschine

(2-)Zählerautomat

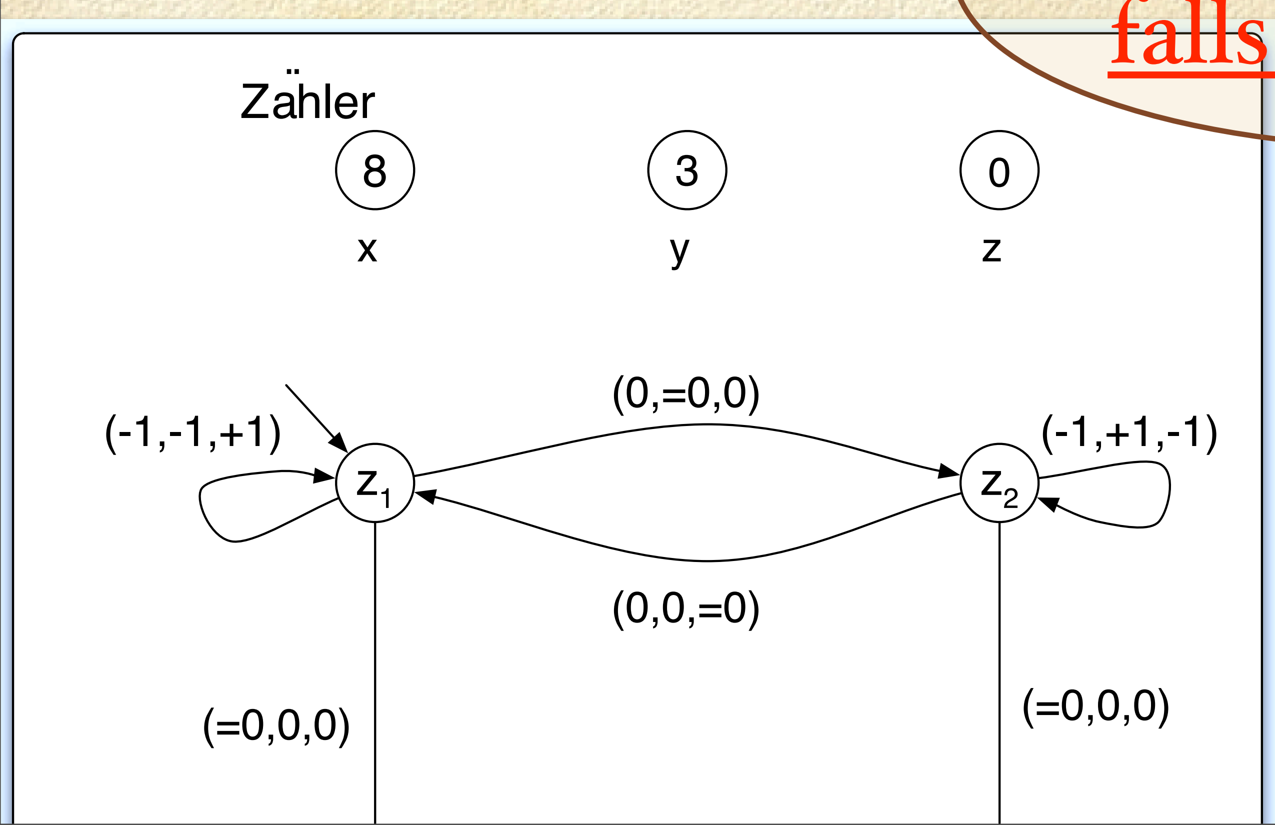
?

P/T-Netz

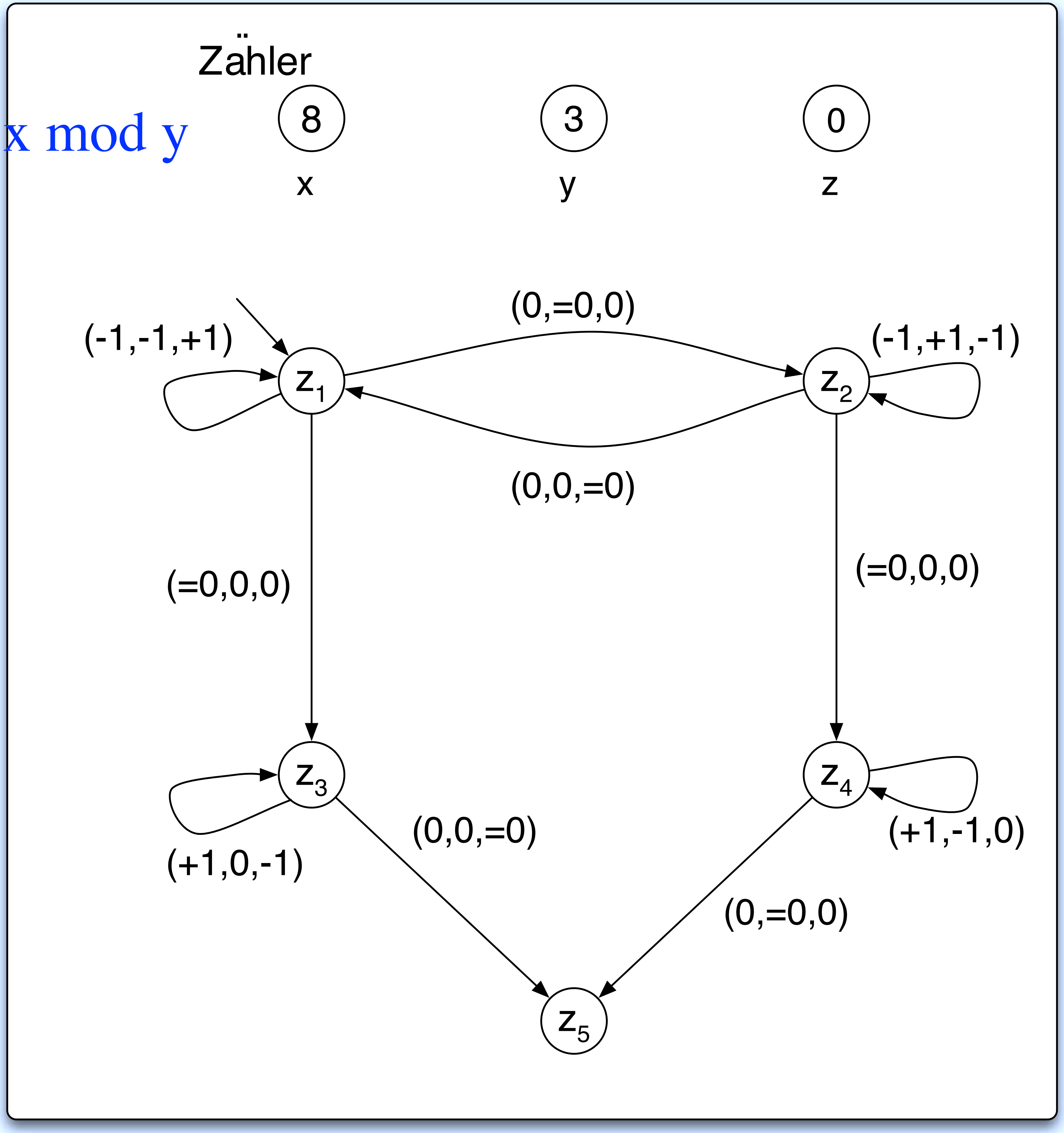
3-Zählerautomat



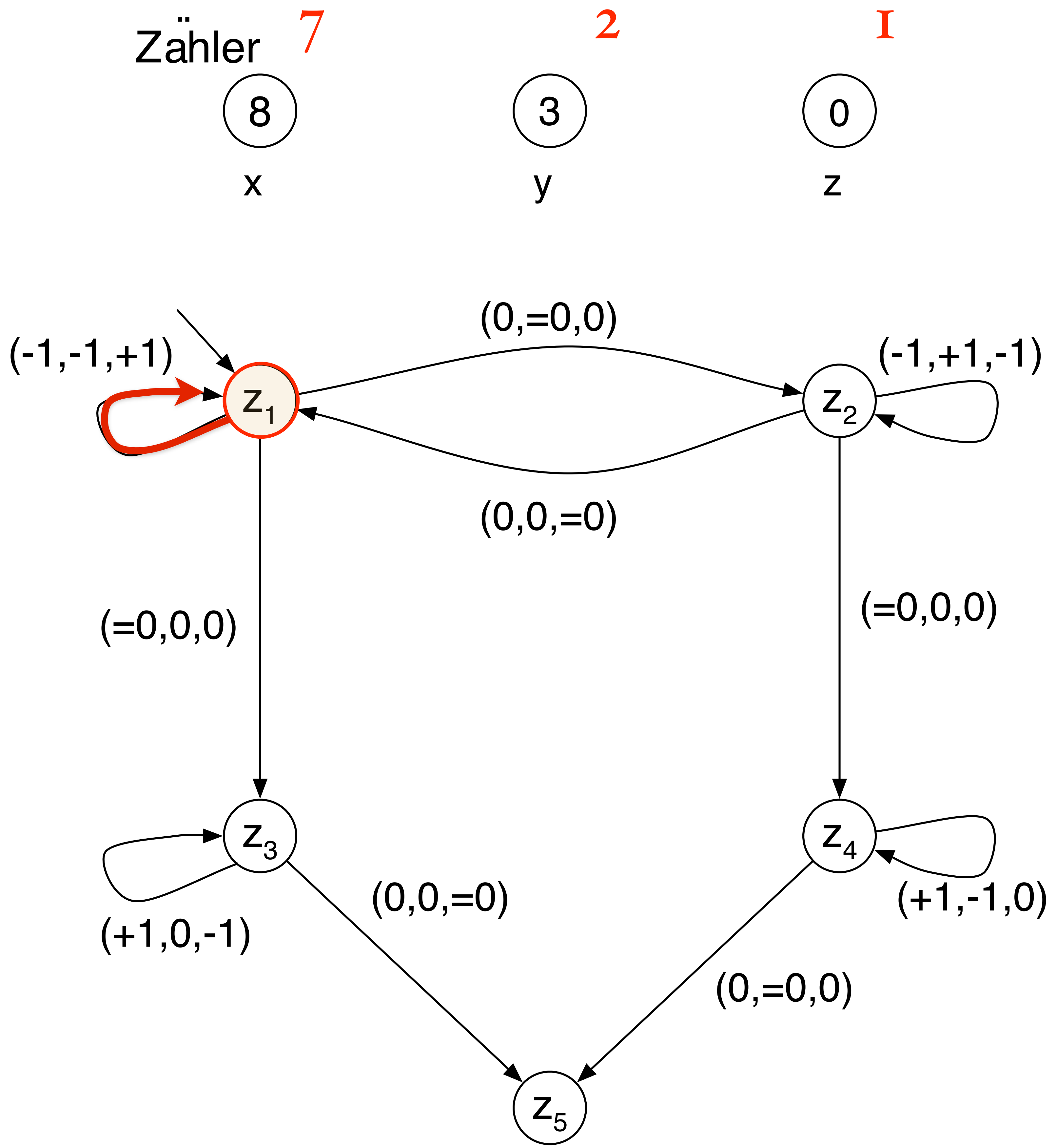
$x := x \bmod y$



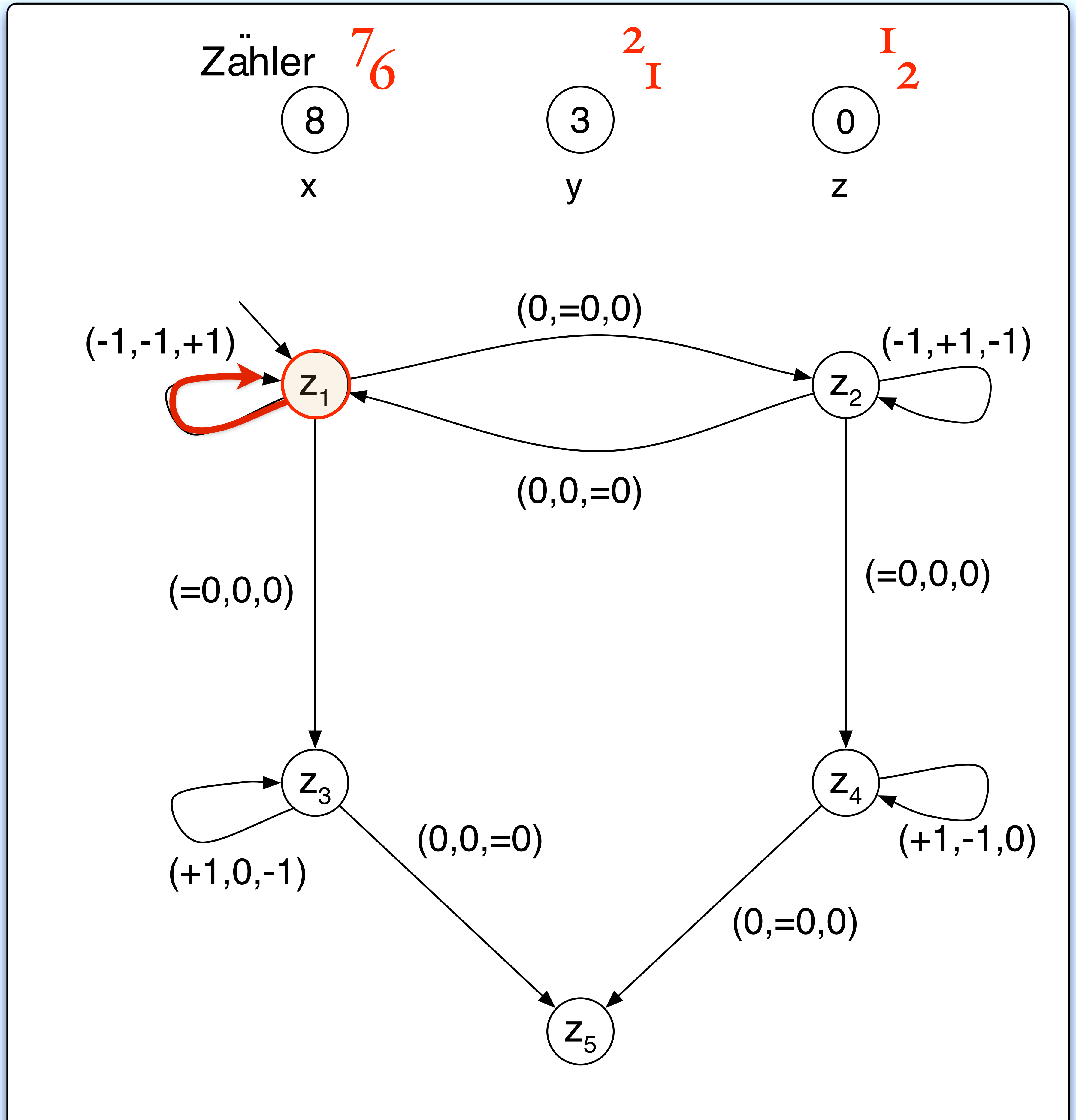
x := x mod y

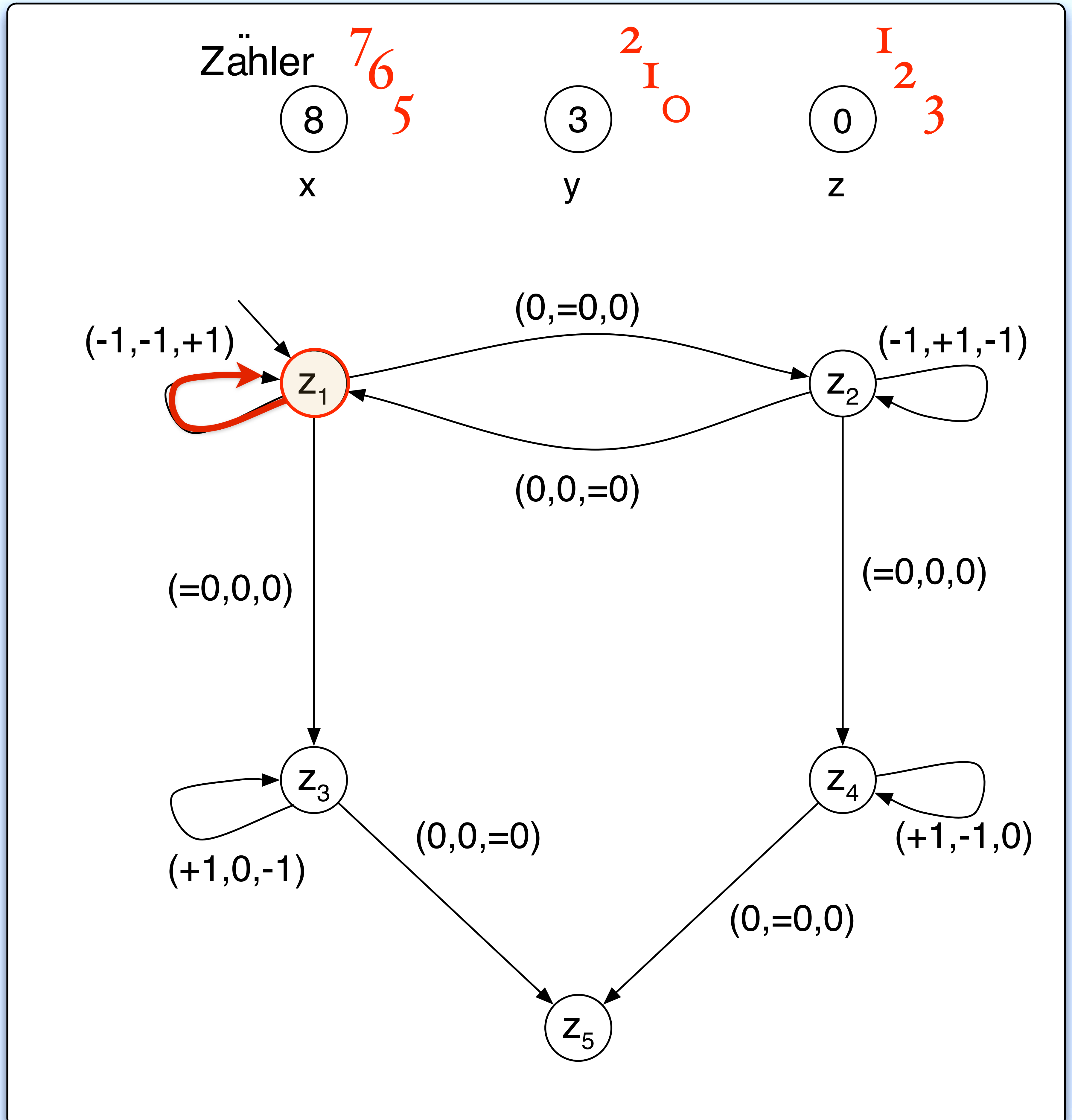


Zähler
8
3
0
x
y
z
(-1,-1,+1)
(0,=0,0)
(-1,+1,-1)
z_1
z_2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z_3
z_4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z_5

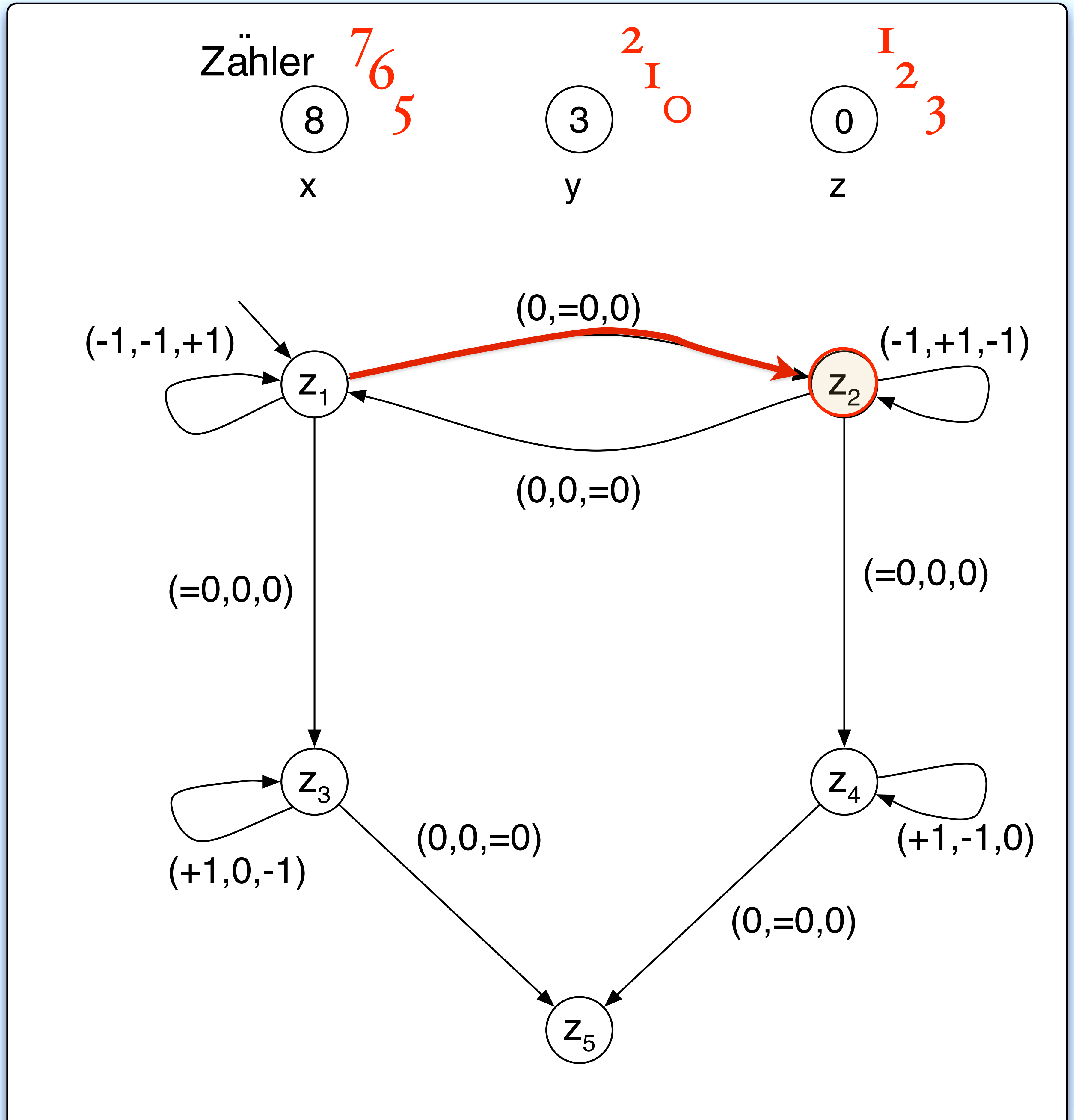


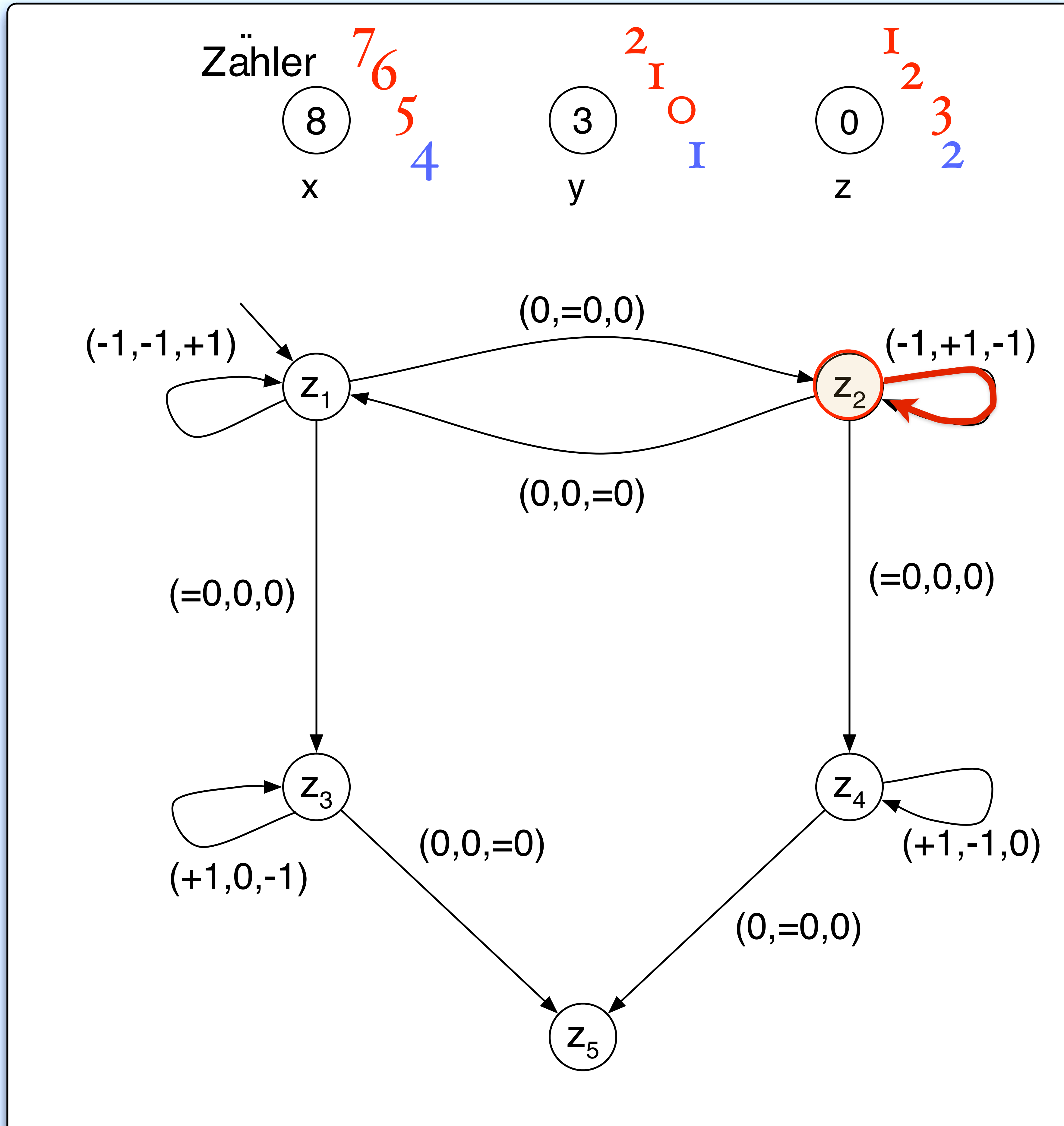
Zähler
7
2
1
8
3
0
x
y
z
(0,=0,0)
(-1,-1,+1)
(-1,+1,-1)
z_1
z_2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z_3
z_4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z_5

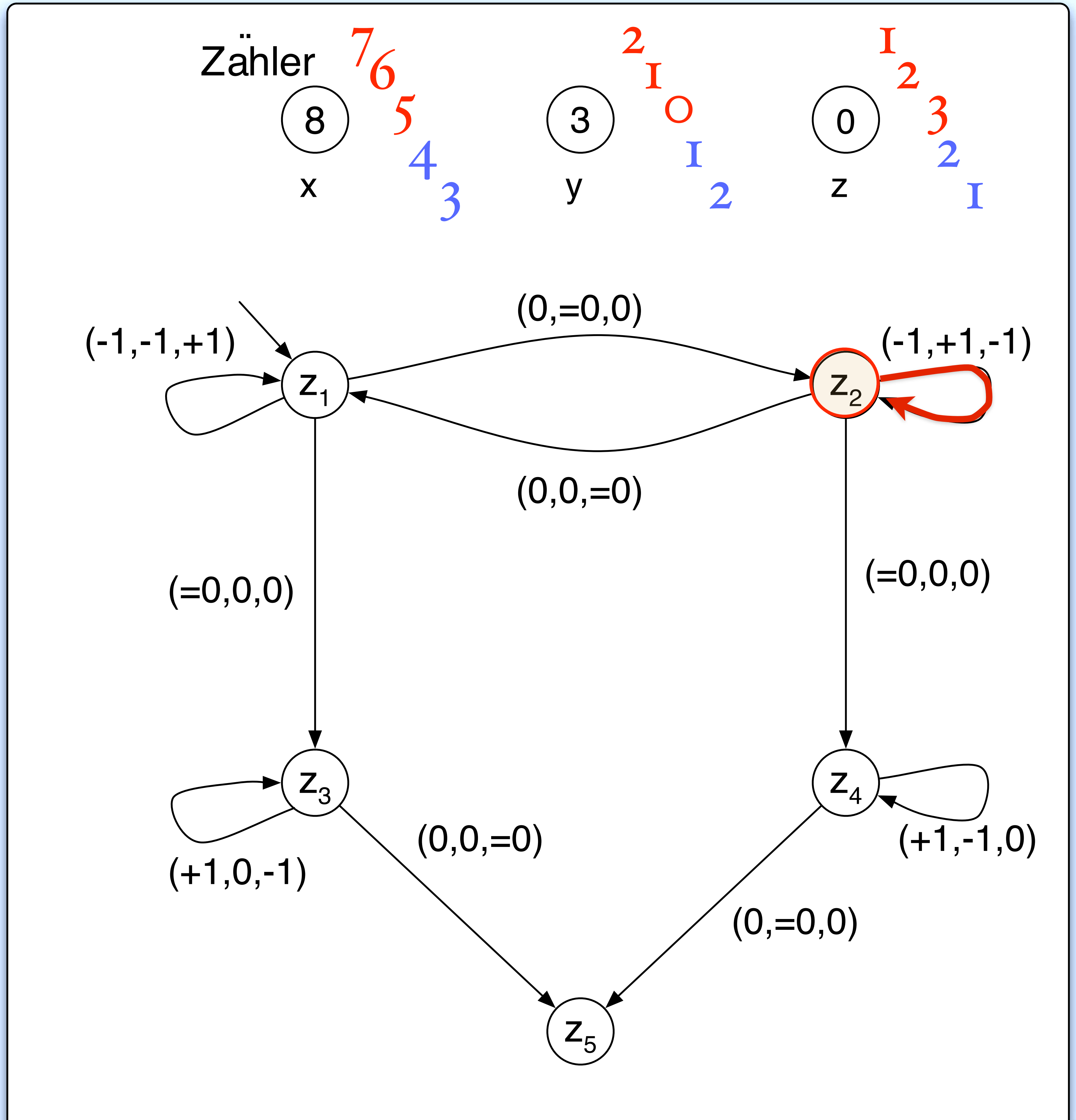

Zähler
7 6
2 1
1 2
8
3
0
x
y
z
(0,=0,0)
(-1,-1,+1)
(-1,+1,-1)
z_1
z_2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z_3
z_4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z_5

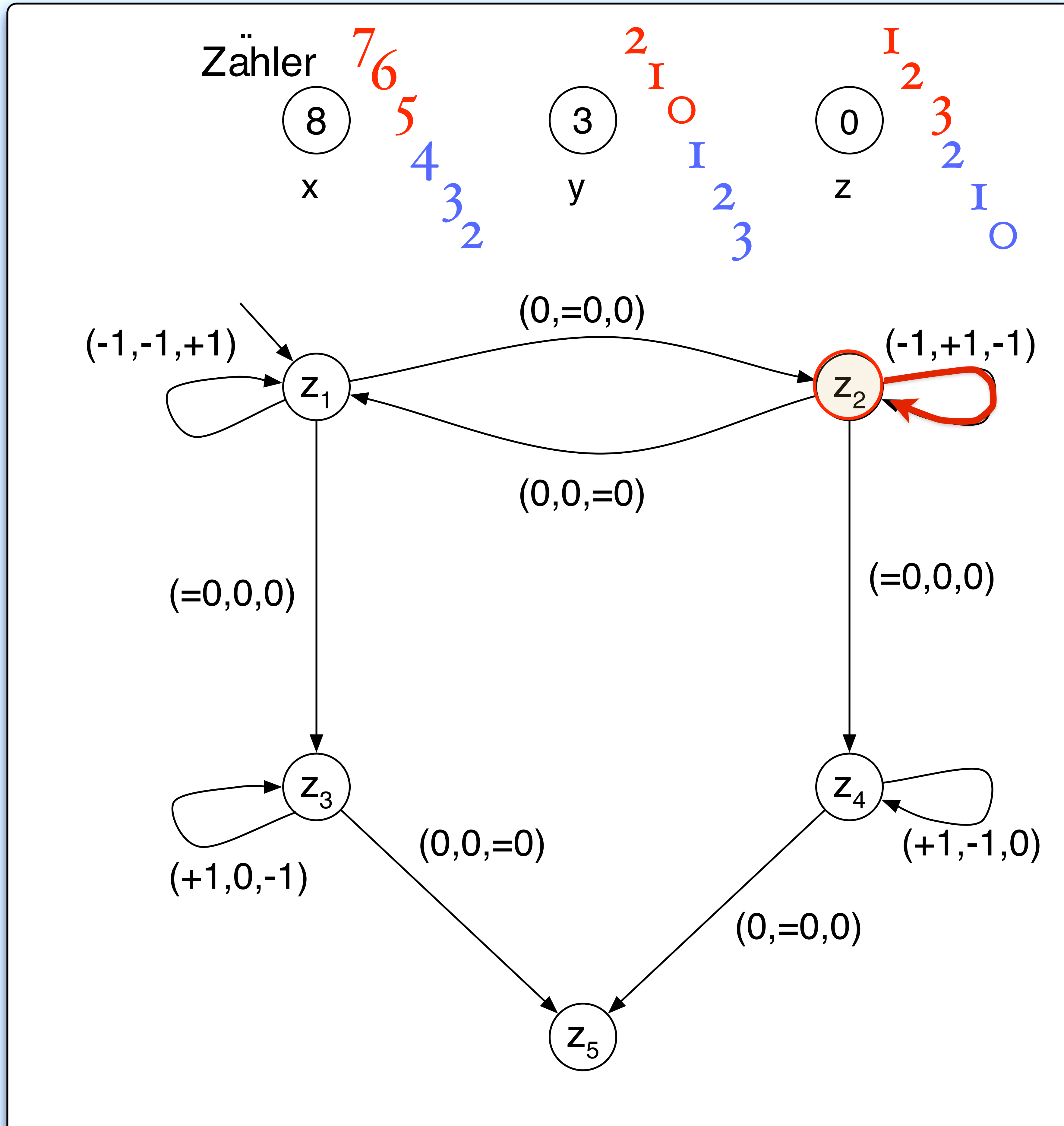


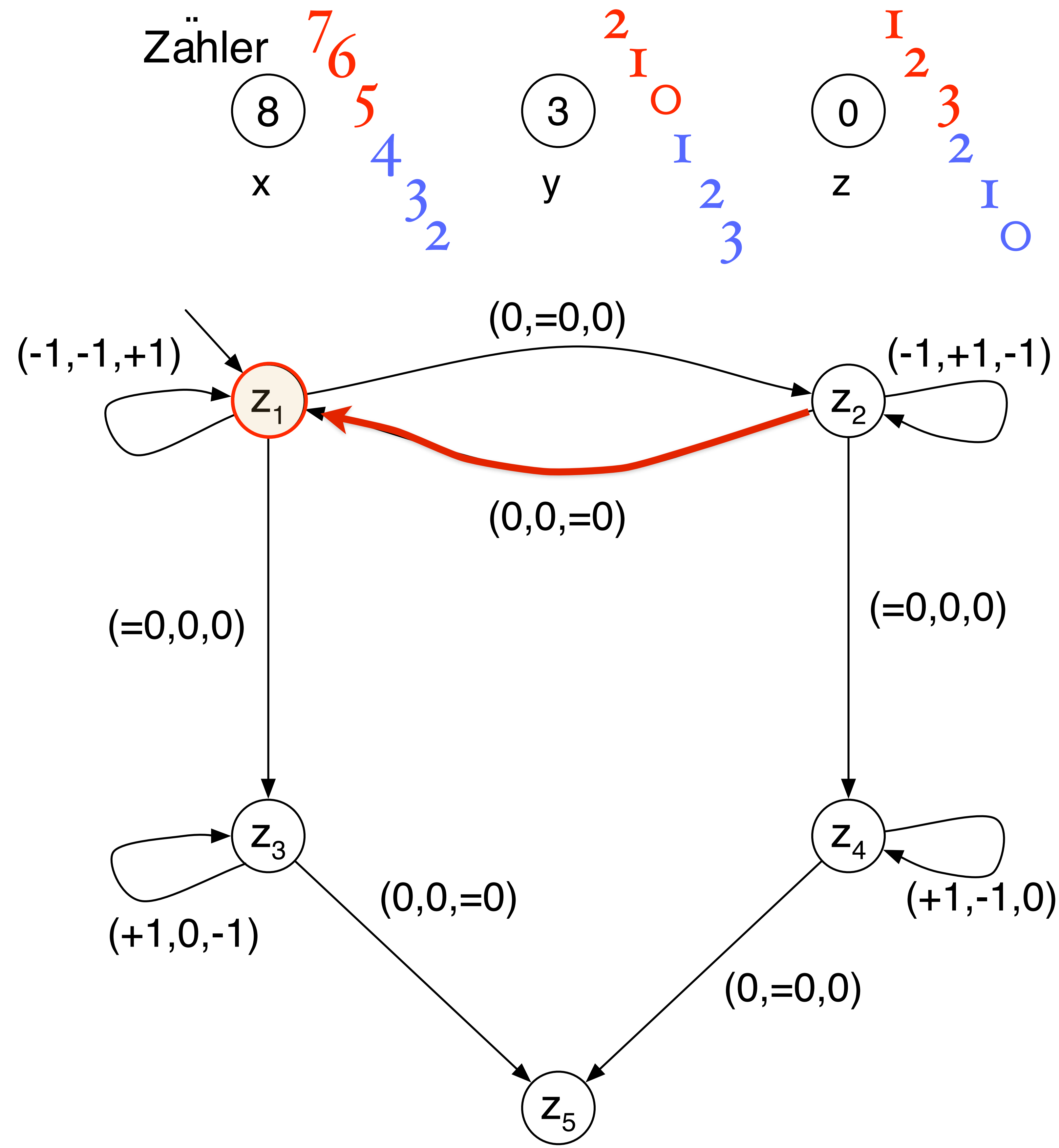
Zähler
8
3
0
x
y
z
7
6
5
2
1
0
1
2
3
(0,=0,0)
(-1,-1,+1)
(-1,+1,-1)
z1
z2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z3
z4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z5

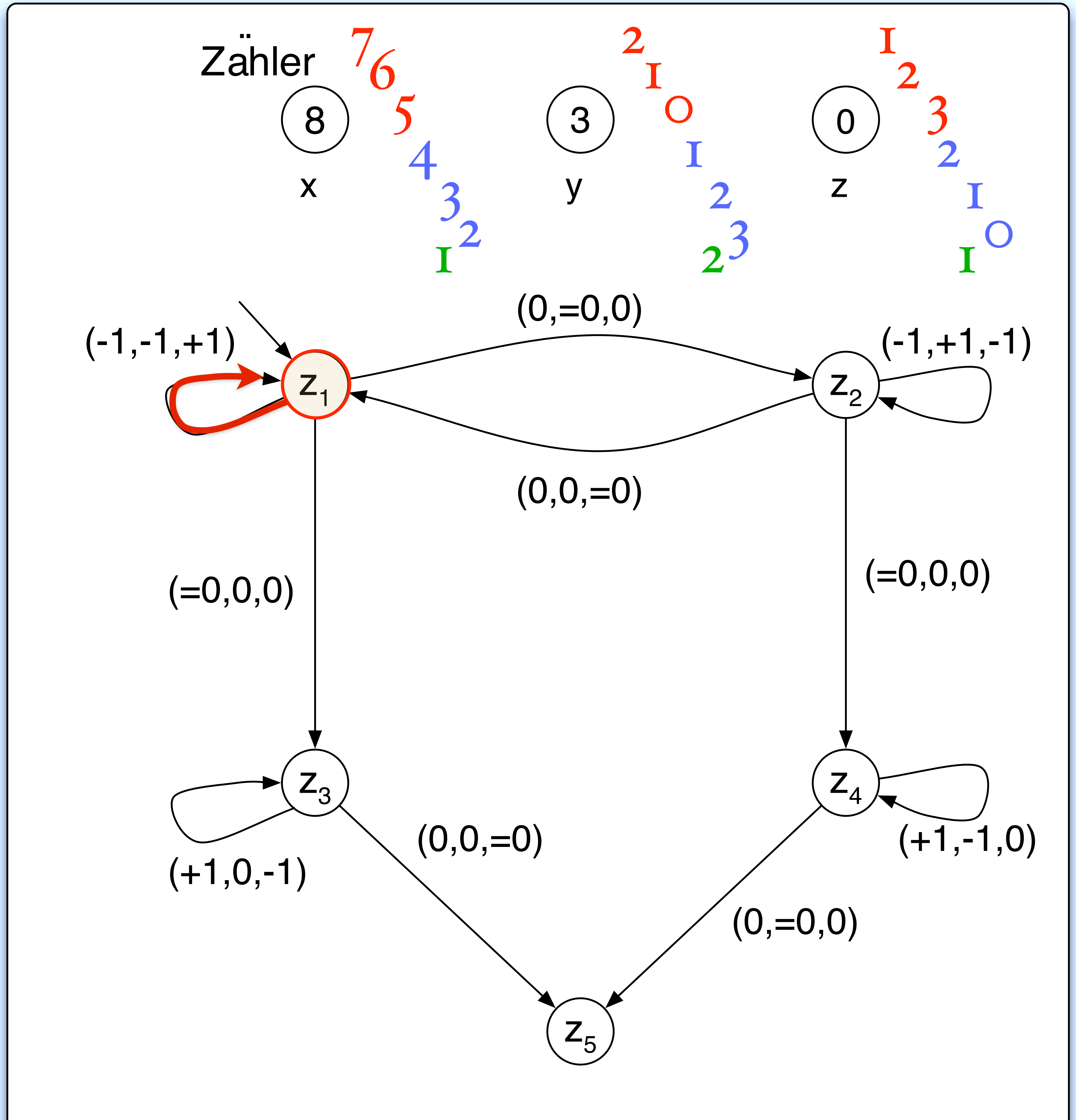

Zähler
8
x
3
y
0
z
7 6 5
2 1 0
1 2 3
(0,=0,0)
(-1,-1,+1)
(-1,+1,-1)
z_1
z_2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z_3
z_4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z_5

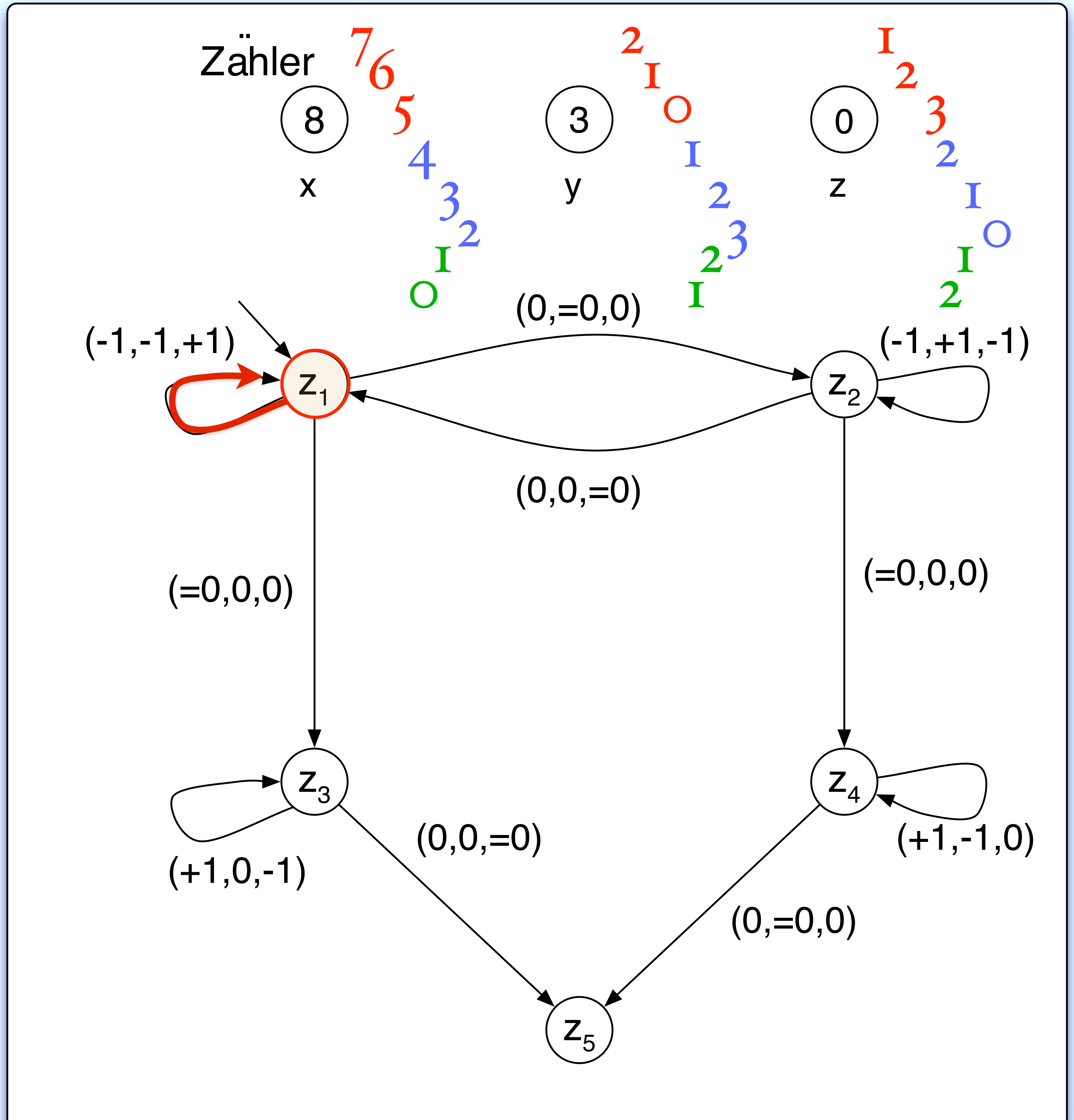

Zähler
8
x
3
y
0
z
7 6 5 4
2 1 0 1
1 2 3 2
(0,=0,0)
(-1,-1,+1)
(-1,+1,-1)
z1
z2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z3
z4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z5


Zähler
8
x
7 6 5 4 3
3
y
2 1 0 1 2
0
z
1 2 3 2 1
(0,=0,0)
(-1,-1,+1)
(-1,+1,-1)
z_1
z_2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z_3
z_4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z_5

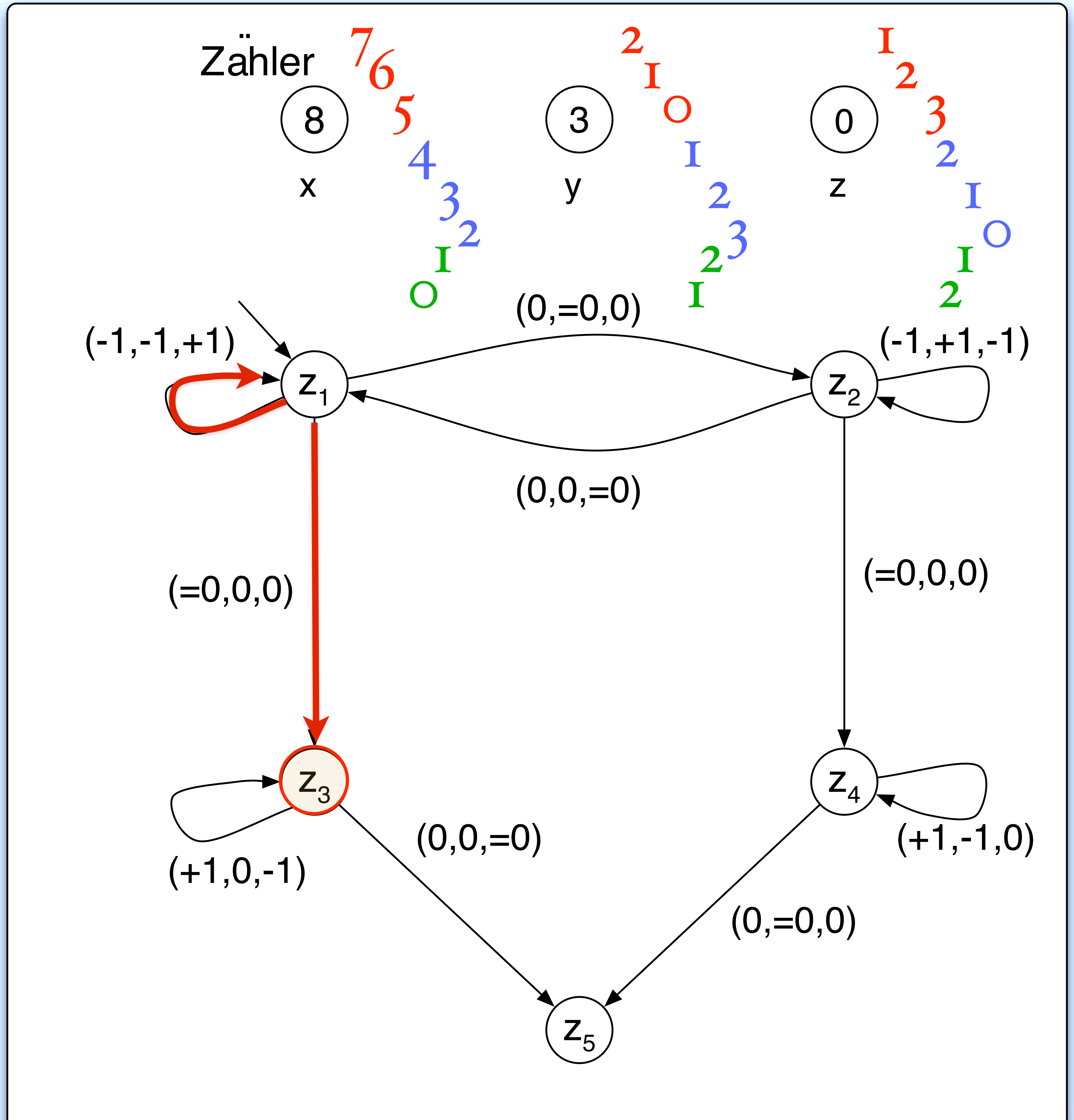

Zähler
8
x
3
y
0
z
7 6 5 4 3 2
2 1 0 1 2 3
1 2 3 2 1 0
(-1,-1,+1)
(0,=0,0)
(-1,+1,-1)
z_1
z_2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z_3
z_4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z_5


Zähler
8
x
3
y
0
z
7 6 5 4 3 2
2 1 0 1 2 3
1 2 3 2 1 0
(-1,-1,+1)
(0,=0,0)
(-1,+1,-1)
z_1
z_2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z_3
z_4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z_5

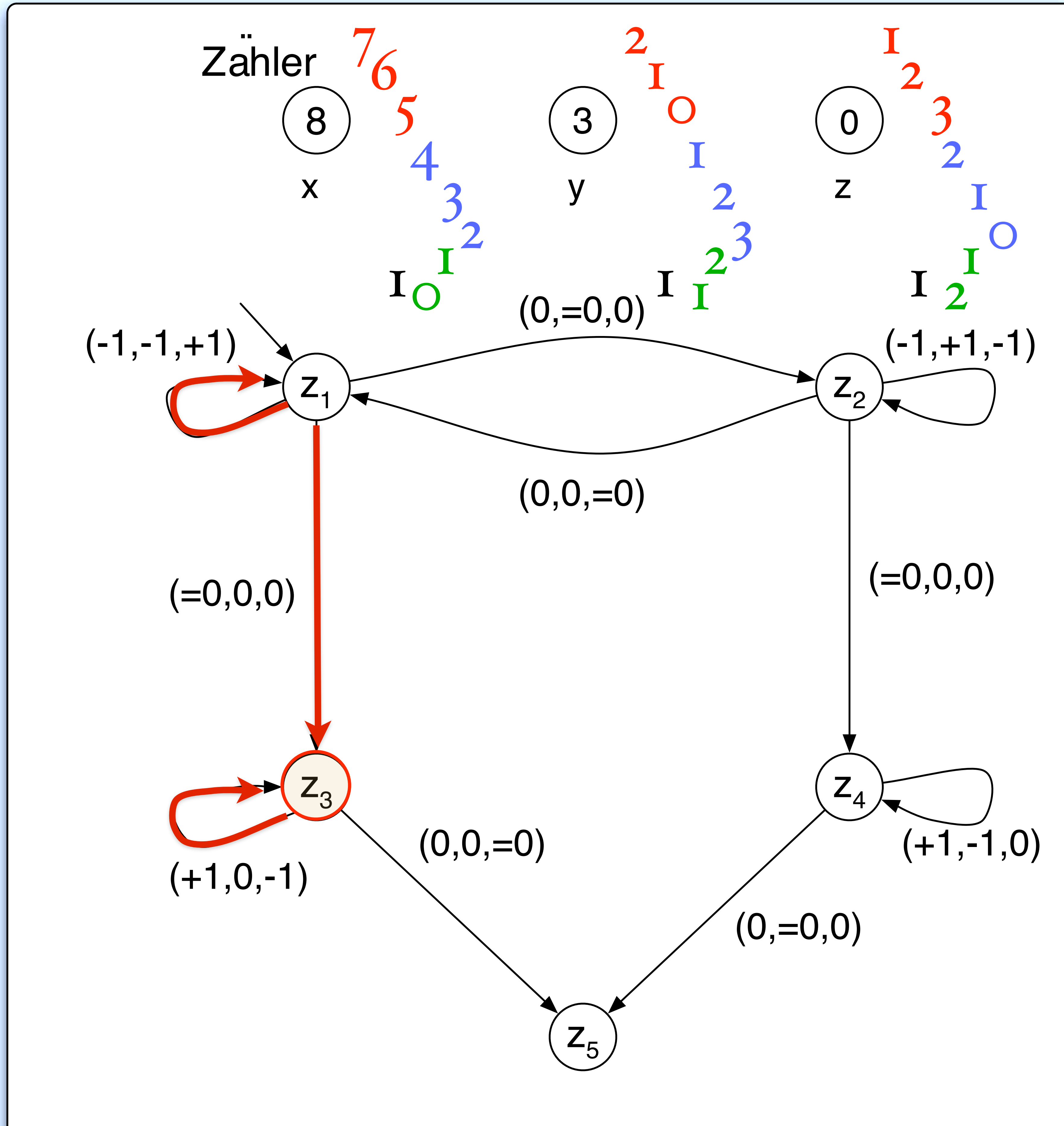

Zähler
8
x
7 6 5 4 3 2 1
3
y
2 1 0 1 2 3 2
0
z
1 2 3 2 1 0 1
(-1,-1,+1)
(0,=0,0)
(-1,+1,-1)
z_1
z_2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z_3
z_4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z_5

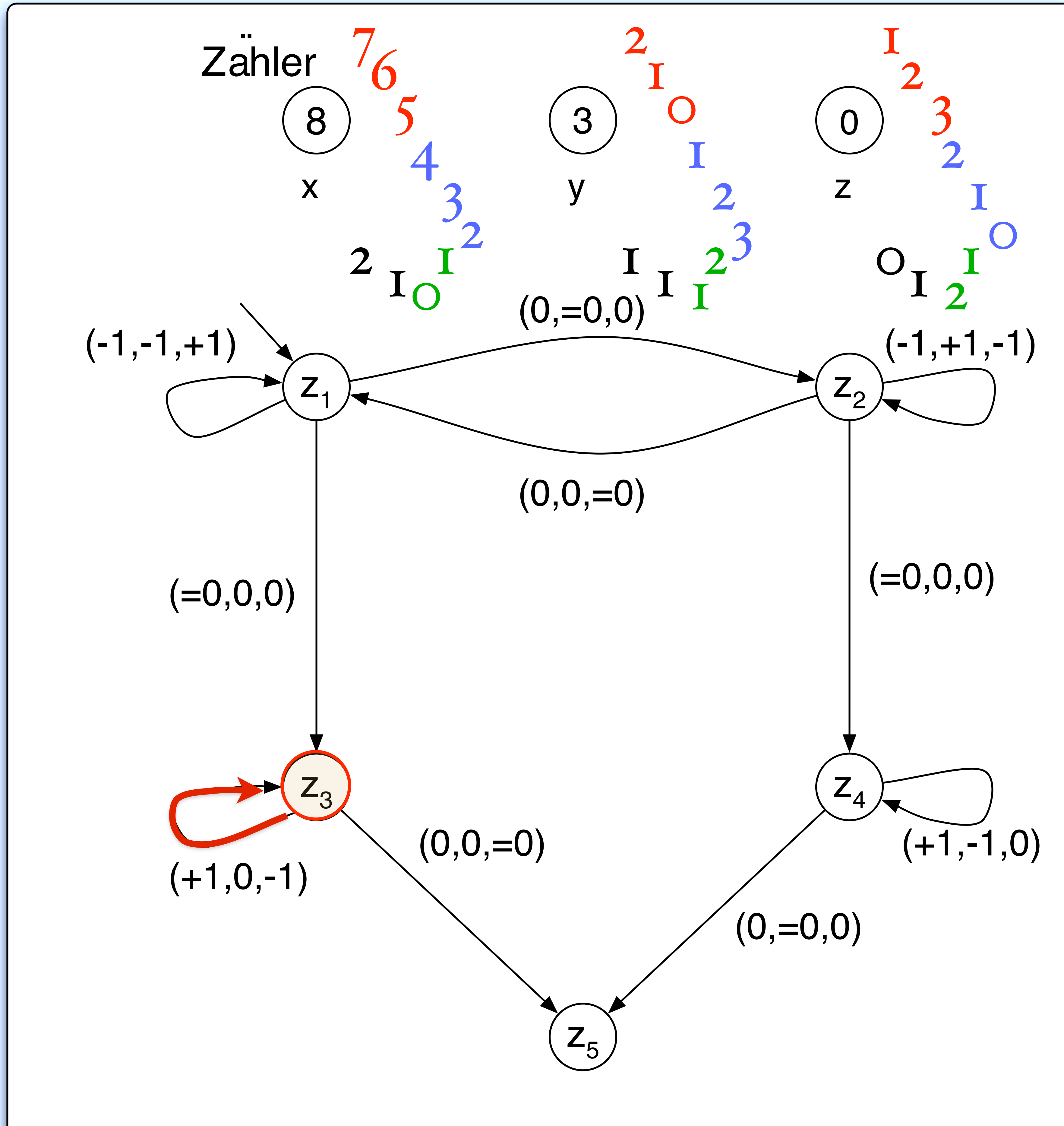


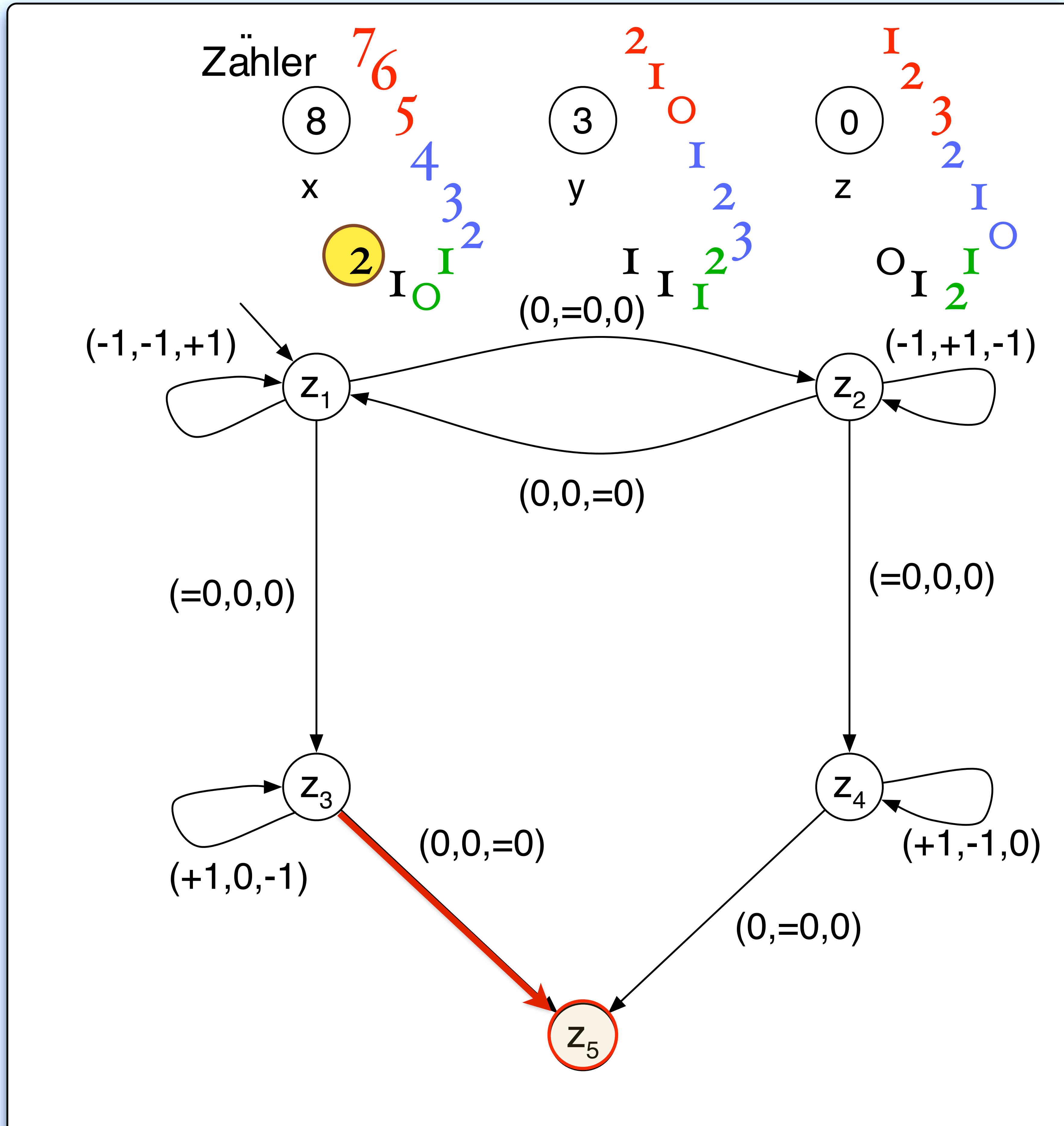
Zähler
8
x
7 6 5 4 3 2 1 0
3
y
2 1 0 1 2 3 2 1
0
z
1 2 3 2 1 0 1 2
(0,=0,0)
(-1,-1,+1)
(-1,+1,-1)
z_1
z_2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z_3
z_4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z_5



Zähler
8
x
3
y
0
z
7 6 5 4 3 2 1 0
2 1 0 1 2 3 2 1
1 2 3 2 1 0 1 2
(-1,-1,+1)
(0,=0,0)
(-1,+1,-1)
z_1
z_2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z_3
z_4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z_5


Zähler
8
x
3
y
0
z
7 6 5 4 3 2 1 0 1
2 1 0 1 2 3 2 1 1
1 2 3 2 1 0 1 2 1
(-1,-1,+1)
(0,=0,0)
(-1,+1,-1)
z_1
z_2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z_3
z_4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z_5


Zähler
8
x
3
y
0
z
(-1,-1,+1)
(0,=0,0)
(-1,+1,-1)
z_1
z_2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z_3
z_4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z_5


Zähler
8
x
3
y
0
z
2
(-1,-1,+1)
(0,=0,0)
(-1,+1,-1)
z_1
z_2
(0,0,=0)
(=0,0,0)
(=0,0,0)
z_3
z_4
(0,0,=0)
(+1,-1,0)
(+1,0,-1)
(0,=0,0)
z_5

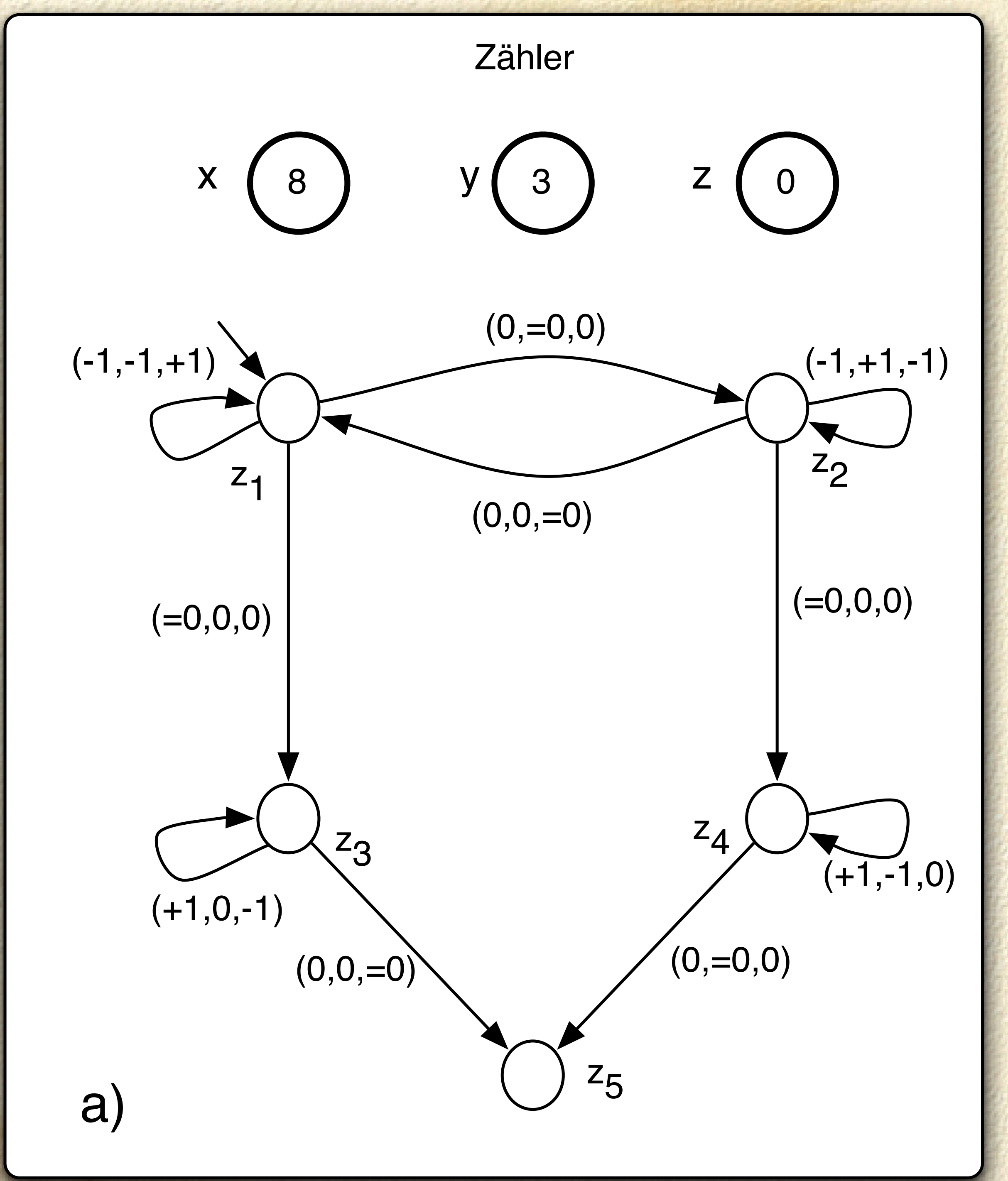
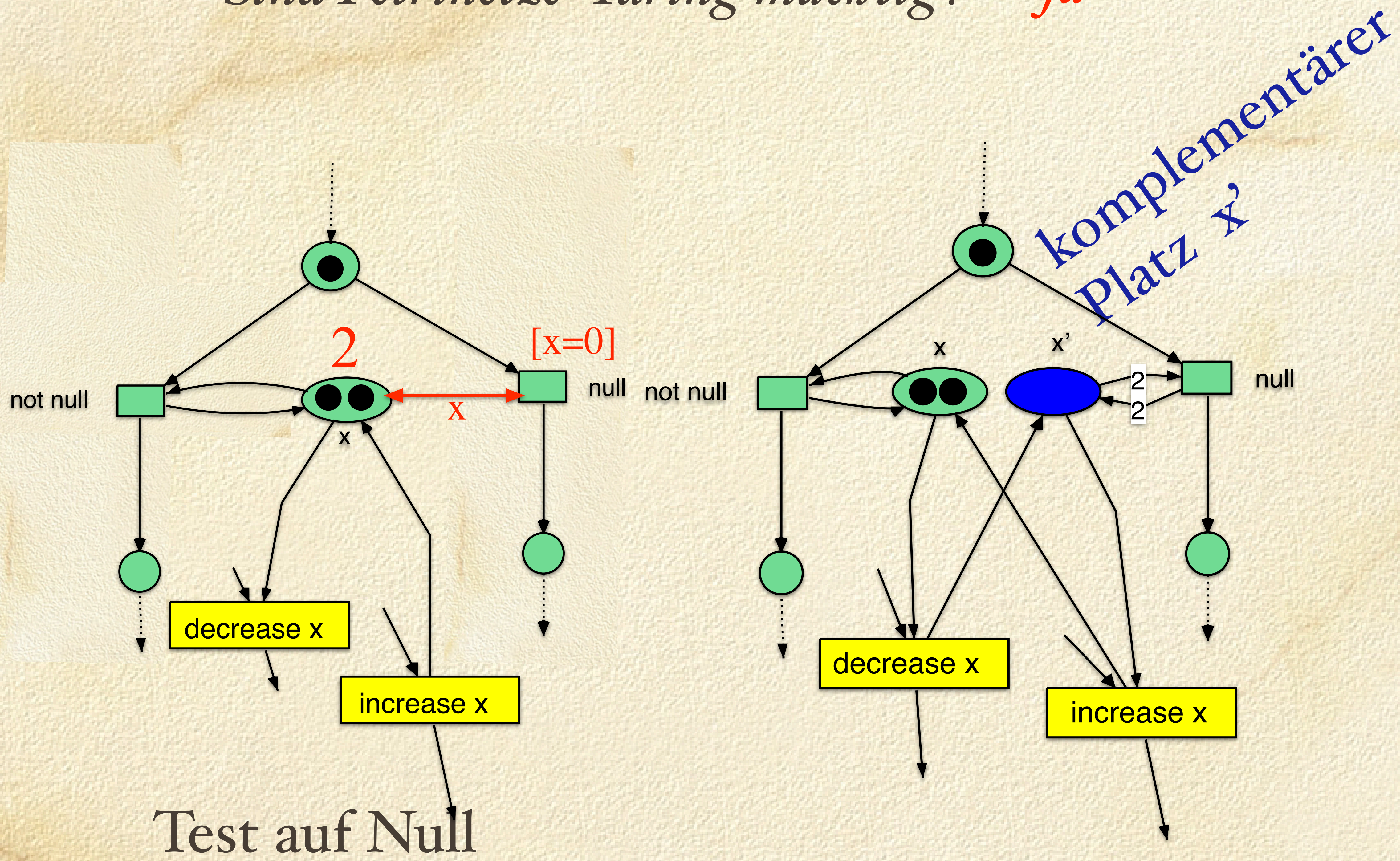


Abbildung 3.18: 3-Zählerautomat und P/T-Netz für $x := x \texttt{ mod } y$

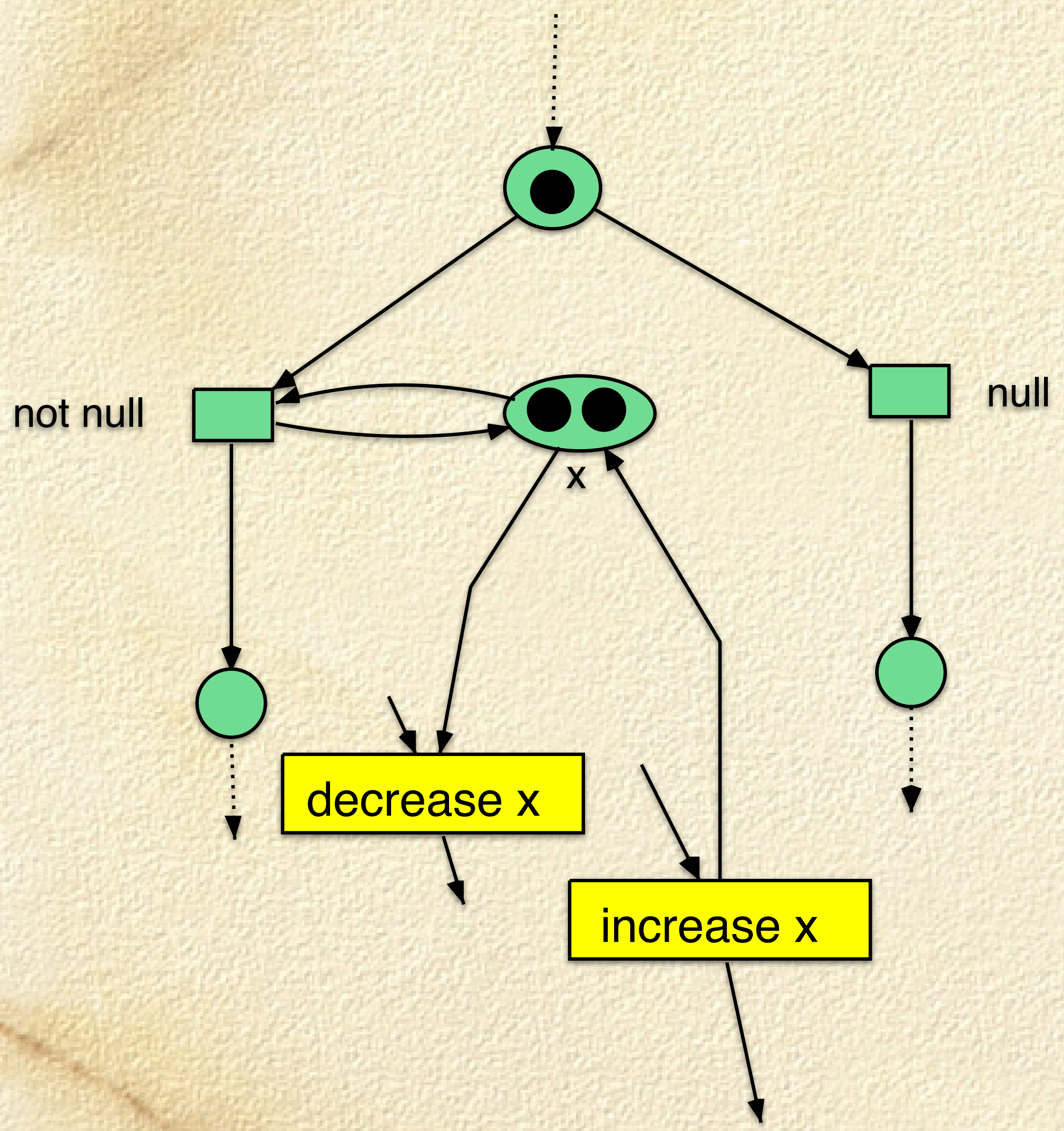
Sind P/T-Netze Turing-mächtig? nein
Sind Petrinetze Turing-mächtig? ja

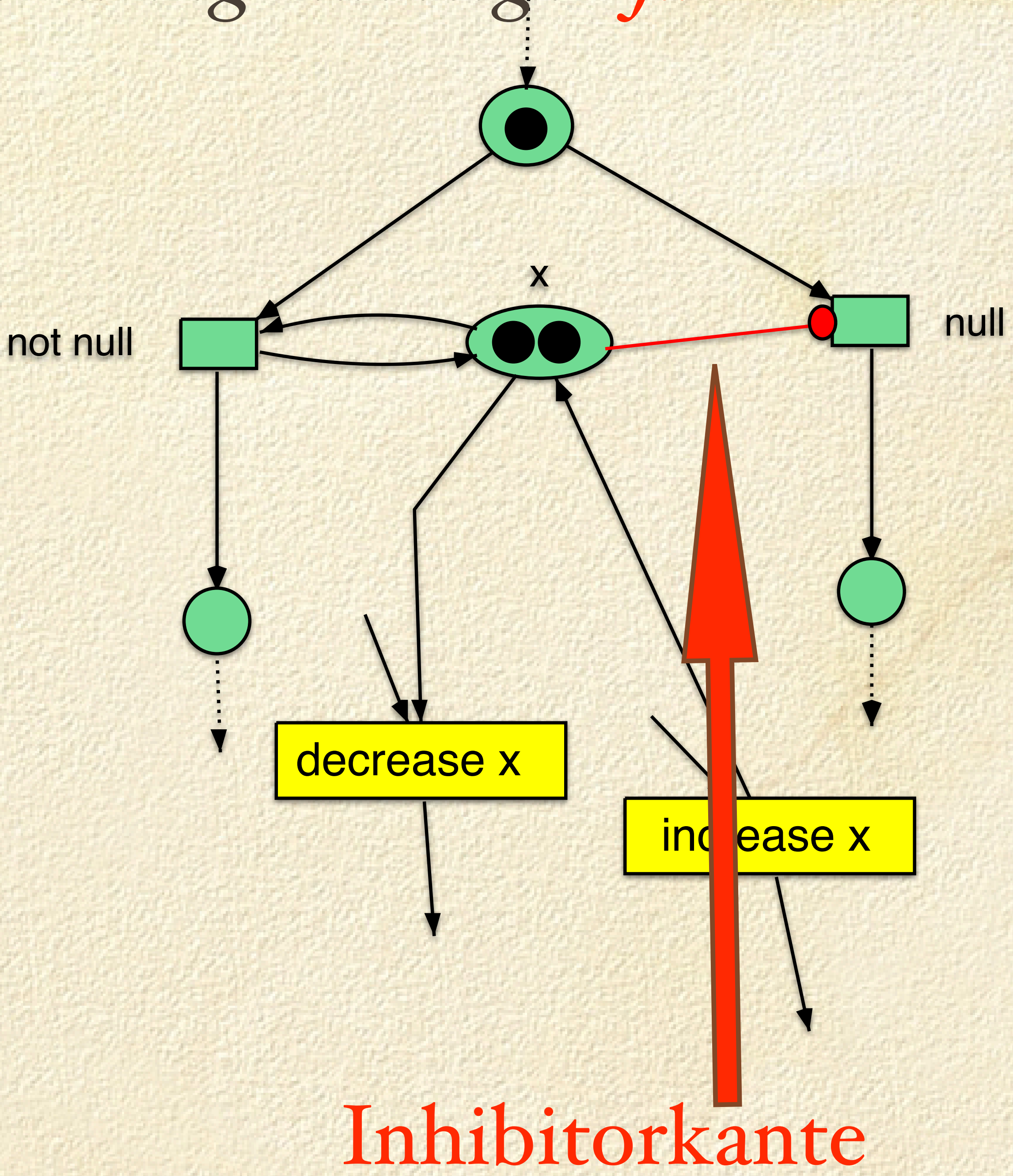


Test auf Null

Platz-Invariante: $\mathbf{m}(x) + \mathbf{m}(x') = k = 2$

Sind Petrinetze Turing-mächtig? ja
Sind P/T-Netze Turing-mächtig? nein
Sind P/T-Netze mit Inhibitorkanten Turing-mächtig? ja





	Turing-mächtig?
P/T-Netze	nein
Inhibitor-Netze	ja
gefärbte Netze mit endlichen Farbmengen	nein
gefärbte Netze mit beliebigen Farbmengen	ja

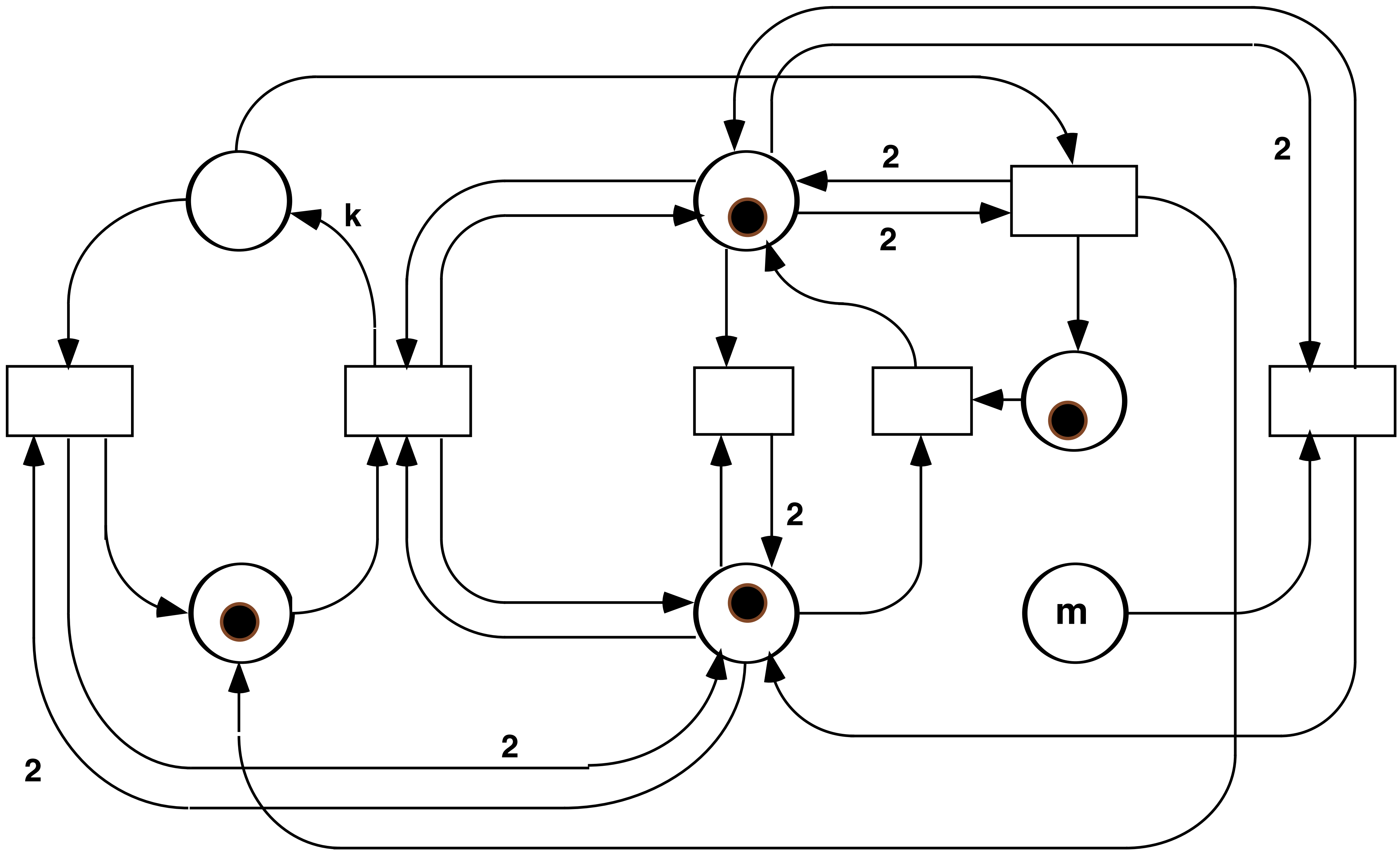
Tabelle 3.2: Turing-Mächtigkeit von Petrinetzen

3.5.3 Komplexität

Es gibt Petri-Netze mit im Verhältnis zu ihrer Größe sehr großer Erreichbarkeitsmenge, wie das folgende Ergebnis zeigt. Diese Erscheinung ist unter dem Begriff *Zustandsexplosion* bekannt.

Satz 3.27 *Es gibt eine unendliche Folge von beschränkten Petri-Netzen* $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \ldots$, *deren Größe* $(|P| + |T| + |F|)$ linear *wächst, jedoch die Größe der Erreichbarkeitsmengen* $|\mathbf{R}(\mathcal{N}_1)|, |\mathbf{R}(\mathcal{N}_2)|, |\mathbf{R}(\mathcal{N}_3)|, \ldots$ *wächst schneller als jede primitiv rekursive Funktion.*

$$2^{2^n} \qquad 2^{2^{2^n}} \qquad 2^{2^{2^{2^{2^{2^{2^{2^{2^{2^{2^{2^{2^n}}}}}}}}}}}}$$


2
2
2
k
2
2
2
m
2
2

Das P/T-Netz $\mathcal{N}_7$ hat 6 Plätzen, 6 Transitionen, $30+k$ Kanten und $m+4$ Marken in der eingezeichneten Anfangsmarkierung $\mathbf{m}_0$. Die Größen $k \geq 2$ und $m \geq 0$ sind festlegbar und die maximale Zahl von möglichen Marken bestimmt sich durch die Funktion

$$\max(m,k) := k \cdot f_k(m) + 2,$$

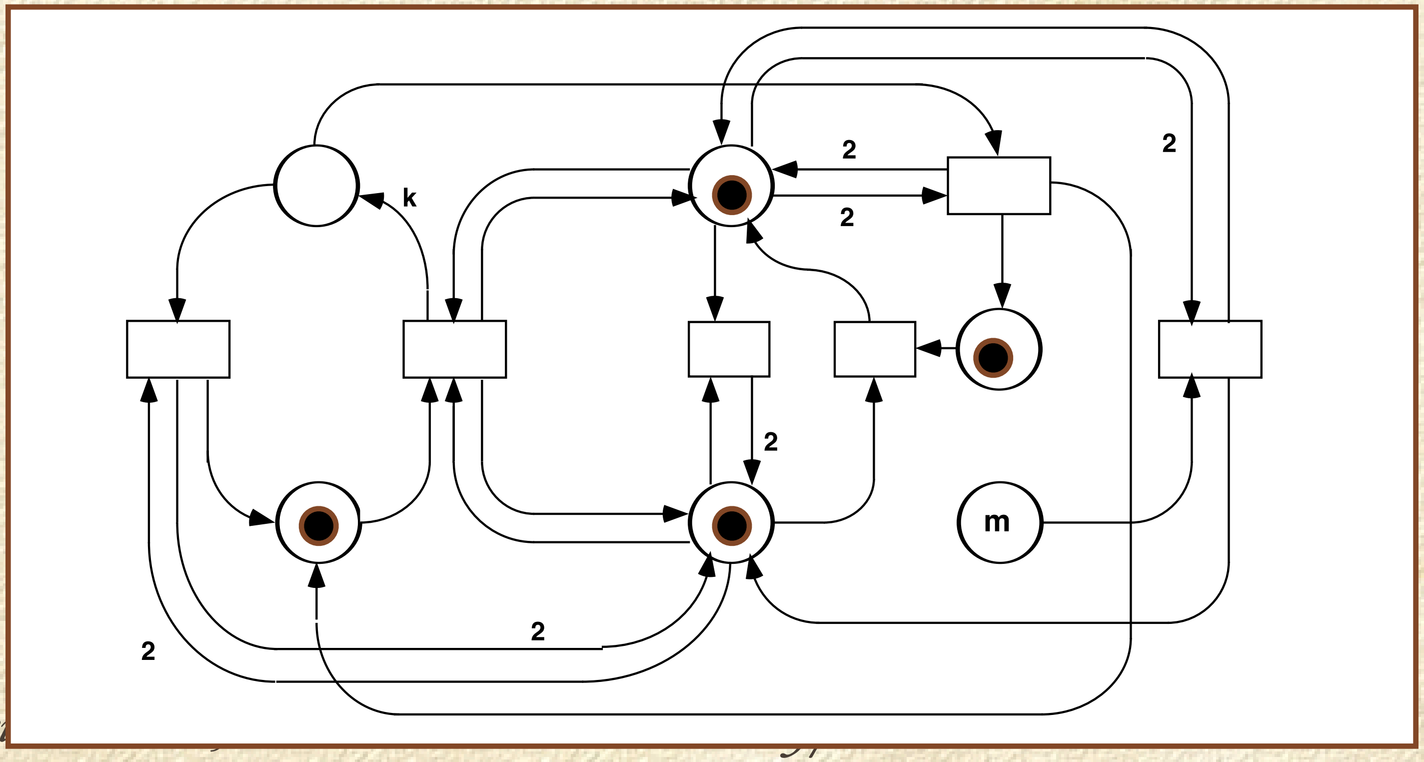
wobei f_k wie folgt definiert ist:

$$f_k(m) := \texttt{ IF } m = 0 \texttt{ THEN } k \texttt{ ELSE } f_k(m-1) \cdot k^{f_k(m-1)} \texttt{ FI}.$$

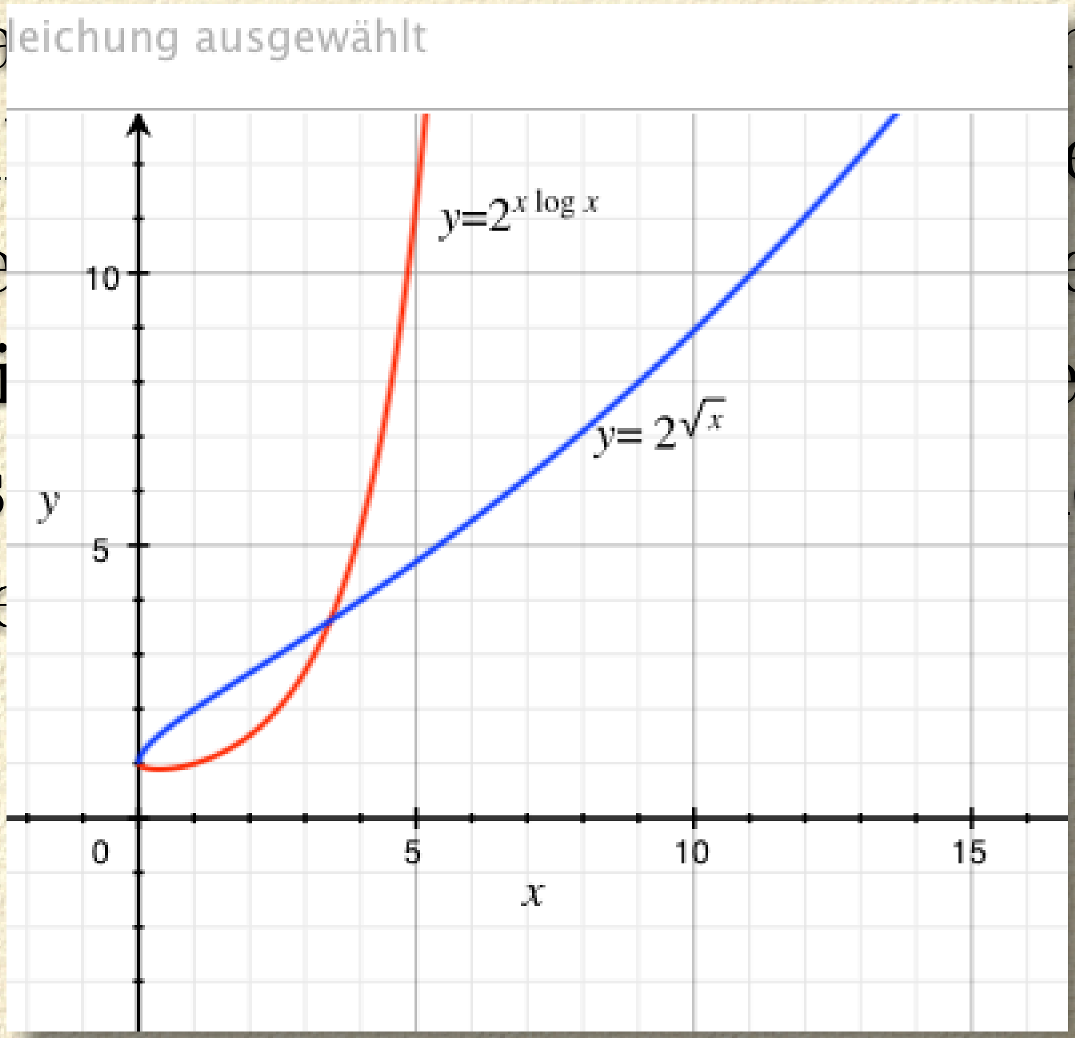
Es gilt: $\max(2,2) = 4098$

$$\max(3,2) = 1048576^{103} + 2$$

$$\max(2,3) = 3^{86} + 2.$$



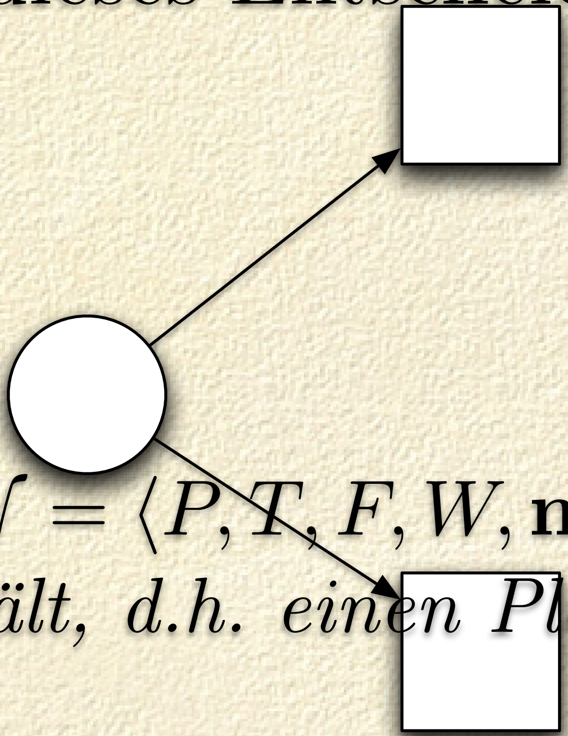
Aus diesem Ergebnis folgt, da … nt-
scheidungsverfahren (Konstru … eit
der Erreichbarkeitsmenge eine … er-
deckungsgraphen nicht primiti … ht
die Frage, ob es vielleicht bess … en
dieses Problem entschieden we …



Satz 3.28 (Rackoff (1978)) *Das Beschränktheitsproblem ist mit $O(2^{c \cdot n \cdot log(n)})$ Platzbedarf entscheidbar.*

Satz 3.29 (Lipton (1976)) *Das Beschränktheitsproblem benötigt für seine Entscheidung mindestens $O(2^{c \cdot \sqrt{n}})$ Platzbedarf.*

Erst für spezielle Teilklassen der Petrinetze kann man eine niedrigere Komplexität für dieses Entscheidungsproblem erwarten.



Definition 3.31 *Ein P/T-Netz* $\mathcal{N} = \langle P, T, F, W, \mathbf{m}_0 \rangle$ *heißt* konfliktfrei, *falls es keinen* Konfliktplatz *enthält, d.h. einen Platz* p *mit* $|p^{\bullet}| > 1$.

Satz 3.32 (Rosier et. al. (1987)) *Das Beschränktheitsproblem kann für konfliktfreie Petrinetze mit* $O(n^{1,5})$ *Platzbedarf entschieden werden.*

Definition 3.36 (Erreichbarkeitsproblem)

Gegeben: *Ein P/T-Netz* $\mathcal{N} := (P, T, F, W, \mathbf{m}_0)$ *und eine Markierung* $\mathbf{m} \in \mathbb{N}^P$.

Frage: *Gilt* $\mathbf{m} \in \mathbf{R}(\mathcal{N})$ *?*

Satz 3.39 *Das Erreichbarkeitsproblem für* beschränkte ~~*endliche*~~ *P/T-Netze ist entscheidbar,*
benötigt jedoch mindestens exponentiell viel Platz.

Eine untere Schranke $NSpace(2^{O(n)})$ wurde von Lipton 1976 bewiesen. Einen guten Überblick über weitere Komplexitätsresultate zu Algorithmen und Problemen bei Petrinetzen ist bei Esparza und Nielsen (1994) zu finden.

Definition 3.36 (Erreichbarkeitsproblem)

Gegeben: *Ein P/T-Netz* $\mathcal{N} := (S, T, W, K, \mathbf{m}_0)$ *und eine Markierung* $\mathbf{m} \in \mathbb{N}^P$.

Frage: *Gilt* $\mathbf{m} \in \mathrm{RG}(\mathcal{N})$ *?*

Definition 3.37 (Überdeckbarkeitsproblem)

Gegeben: *Ein P/T-Netz* $\mathcal{N} := (S, T, W, K, \mathbf{m}_0)$ *und eine Markierung* $\mathbf{m} \in \mathbb{N}^P$.

Frage: *Gibt es* $\mathbf{m}' \in \mathrm{RG}(\mathcal{N})$ *mit* $\mathbf{m}' \geq \mathbf{m}$ *?*

Satz 3.38 *Das Erreichbarkeitsproblem für 1-konservative P/T-Netze ist* $\mathcal{PSPACE}$*-vollständig.*

$$\forall t \in T : |t^\bullet| = |{}^\bullet t|$$