# 6.2 Datenkonsistenz

Begriffe:

(schematisches) Auftragssystem

Serialisierbarkeit

Funktionalität

$$P_2 \quad : \quad a_0 \; : \; z := 1;$$
$$\underline{con}\; A: \; z := z+1 \,||\, B: \; z := z+2 \; \underline{noc}$$

$$\boxed{\begin{array}{lll} P_1 & : & a_0 \; : \; z := 1 \\ & & \underline{con}\; a_1 \; : \; x := z+1; \; a_2 : \; z := x \; || \\ & & \qquad b_1 \; : \; y := z+2; \; b_2 : \; z := y \; \underline{noc} \end{array}}$$

*x, y sind "lokale" Variable*

$$
\begin{aligned}
P_1 \quad : \quad & a_0 \; : \; z := 1 \\
& \underline{con}\; a_1 \; : \; x := z + 1; \; a_2 : \; z := x \; || \\
& \qquad b_1 \; : \; y := z + 2; \; b_2 : \; z := y \; \underline{noc}
\end{aligned}
$$

*x, y sind "lokale" Variable*

Zuerst ist $z := 1$, dann einige Ausführungsfolgen:

*seriell* ← *„inkonsistent"* → *seriell*

| $a_1a_2b_1b_2$ mit | $a_1b_1a_2b_2$ mit | $a_1b_1b_2a_2$ mit | $b_1a_1a_2b_2$ mit | $b_1b_2\,a_1a_2$ |
|---|---|---|---|---|
| x := z+1 | x := z+1 | x := z+1 | y := z+2 | |
| z := x | y := z+2 | y := z+2 | x := z+1 | |
| y := z+2 | z := x | z := y | z := x | |
| z := y | z := y | z := x | z := y | |
| x : 2 | x : 2 | x : 2 | y : 3 | |
| z : 2 | y : 3 | y : 3 | x : 2 | |
| y : 4 | z : 2 | z : 3 | z : 2 | |
| z : 4 | z : 3 | z : 2 | z : 3 | z : 4 |

## *ein anderes Beispiel:*

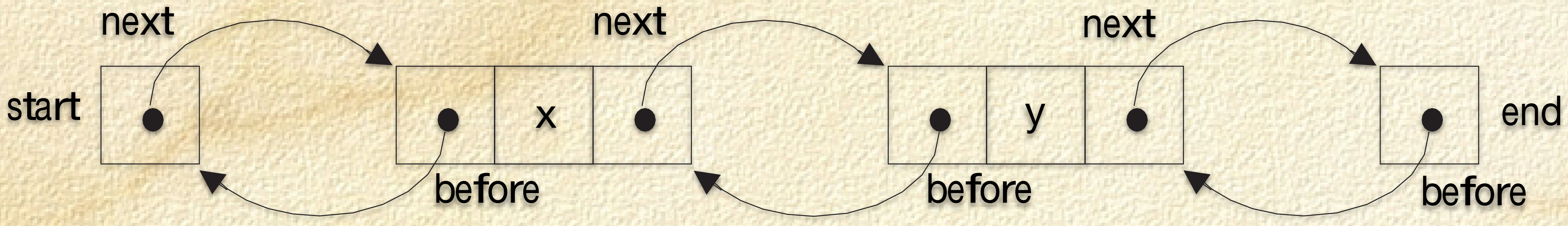



Abbildung 4.28: Doppelt verkettete Liste

$$
\begin{aligned}
&delete(p) \qquad (4.3)\\
&\quad \underline{con}\ a_1 \ :\ p_1 := p.\text{before} \ ||\ b_1 \ :\ p_2 := p.\text{next}\ \underline{noc}\ ;\\
&\quad \underline{con}\ a_2 \ :\ p_1.\text{next} := p_2 \ ||\ b_2 \ :\ p_2.\text{before} := p_1\ \underline{noc}
\end{aligned}
$$

$$p_3 :\ \underline{con}\ delete(x)\ ||\ delete(y)\ \underline{noc}$$

## *ein anderes Beispiel:*

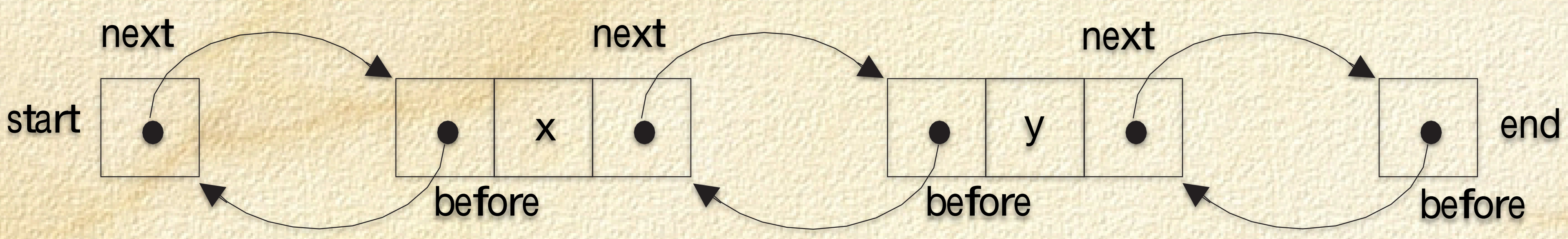
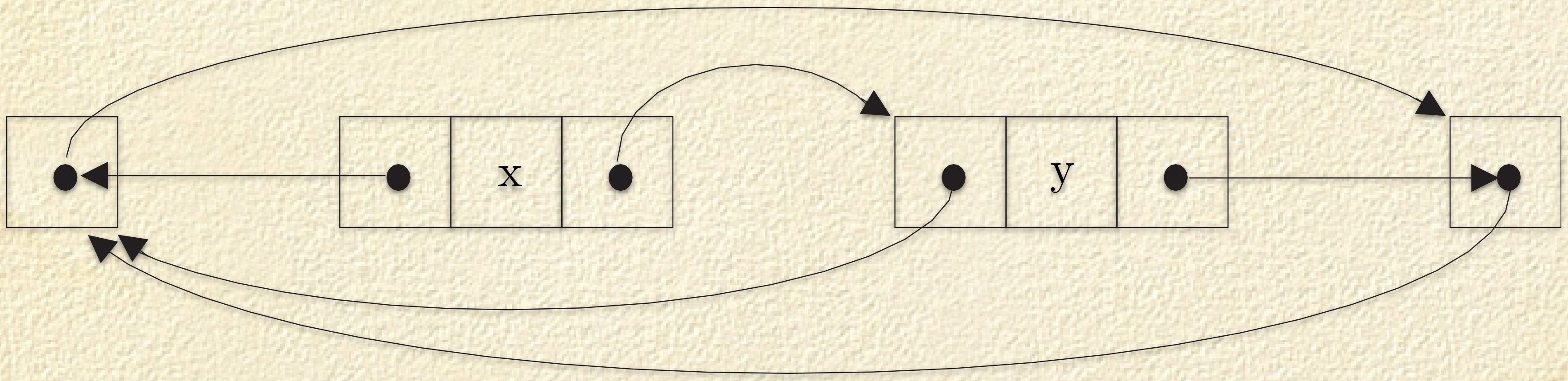



Abbildung 4.28: Doppelt verkettete Liste

## *serielle Ausführung*

a)




$$p_3 : \ \underline{con}\ delete(x) \ || \ delete(y) \ \underline{noc}$$

## ein anderes Beispiel:

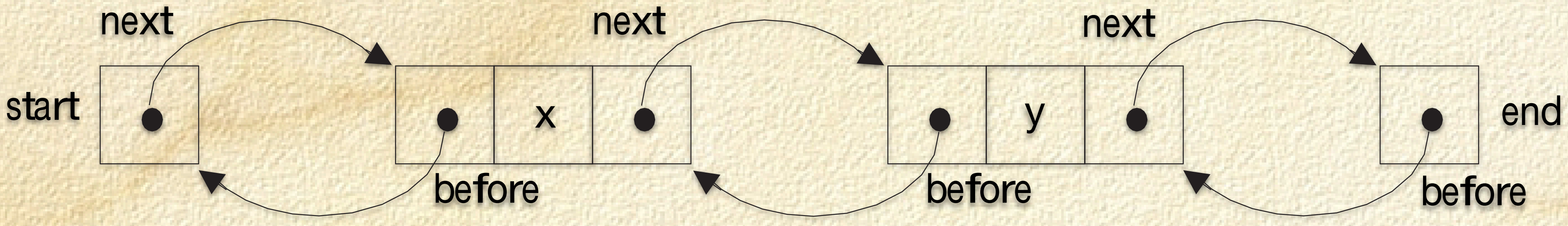
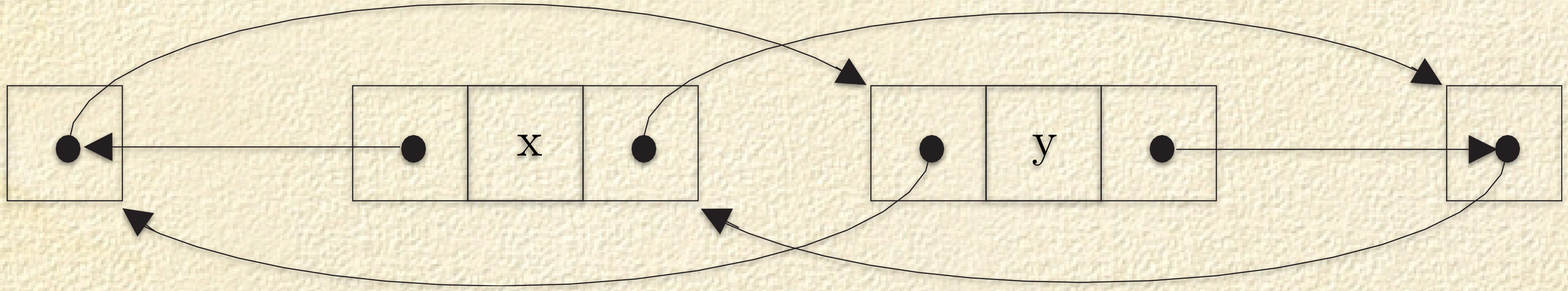



Abbildung 4.28: Doppelt verkettete Liste

## nebenläufige Ausführung

„inkonsistent“ ≈ „nicht serialisierbar“

b)

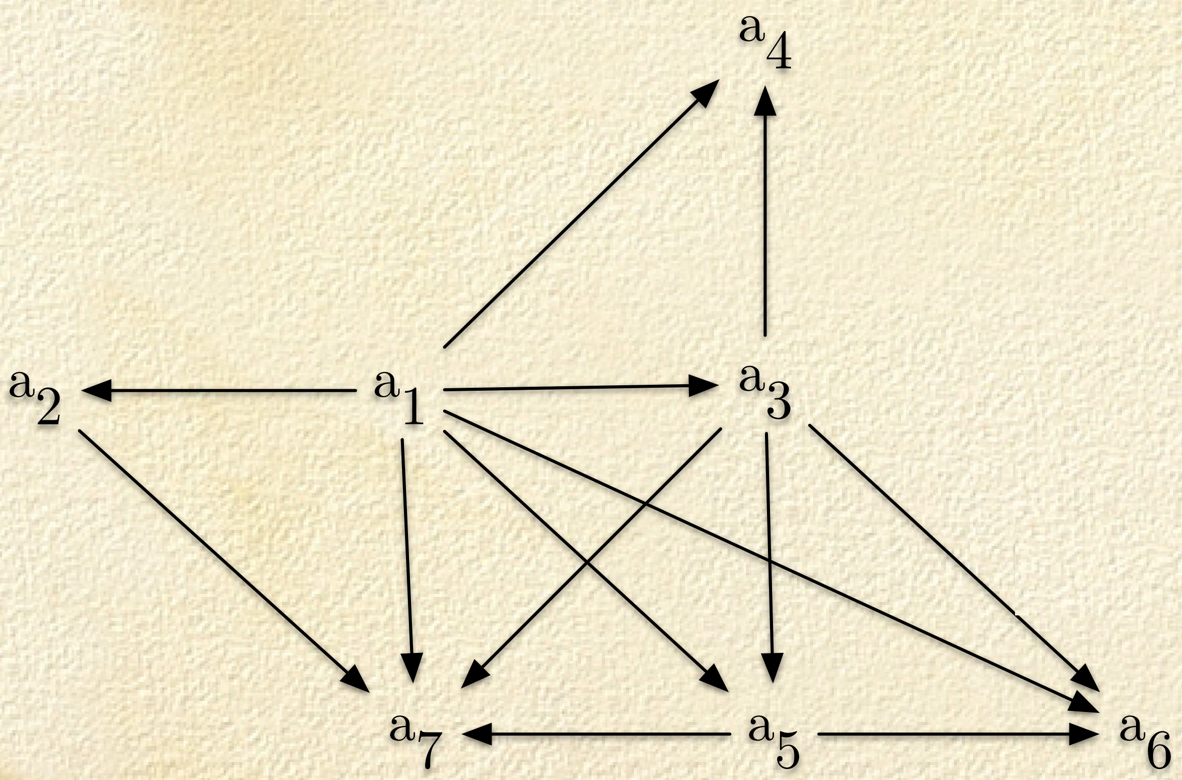



$$p_3 : \ \underline{con} \ delete(x) \ || \ delete(y) \ \underline{noc}$$

Auftragssystem:

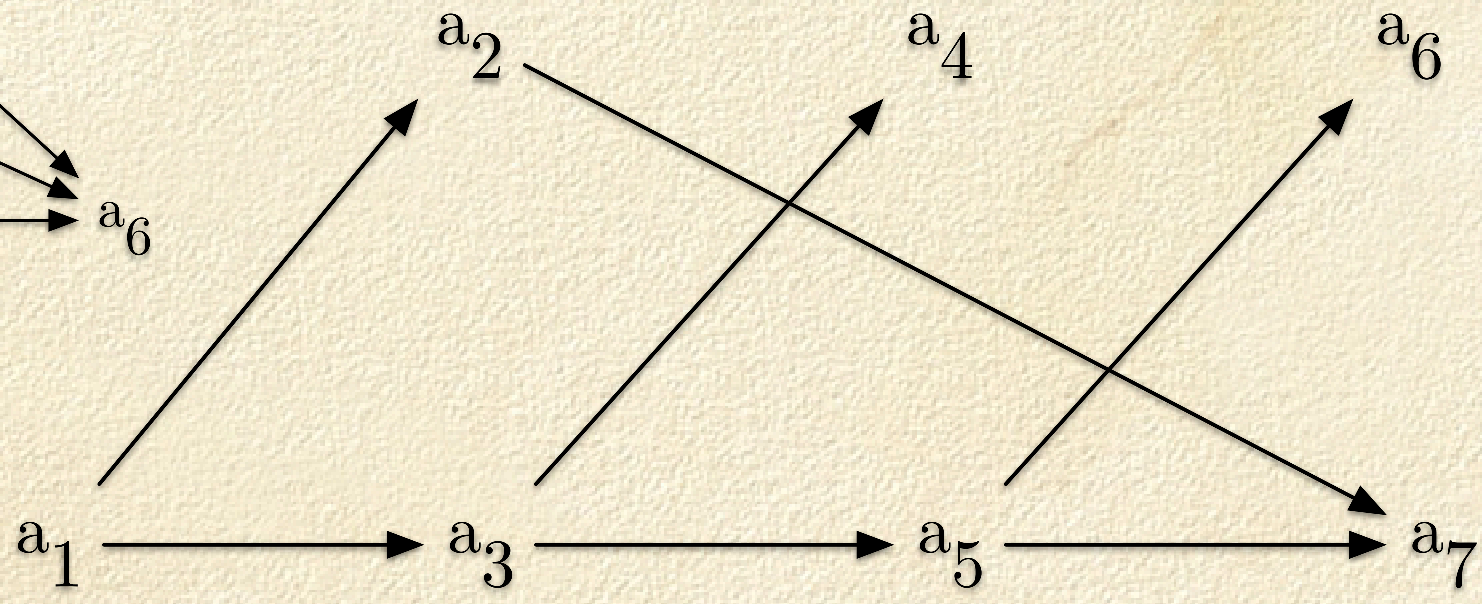
a) endliche Menge von Aufträgen

b) irreflexible, transitive Relation $<$ ( Striktordnung! )

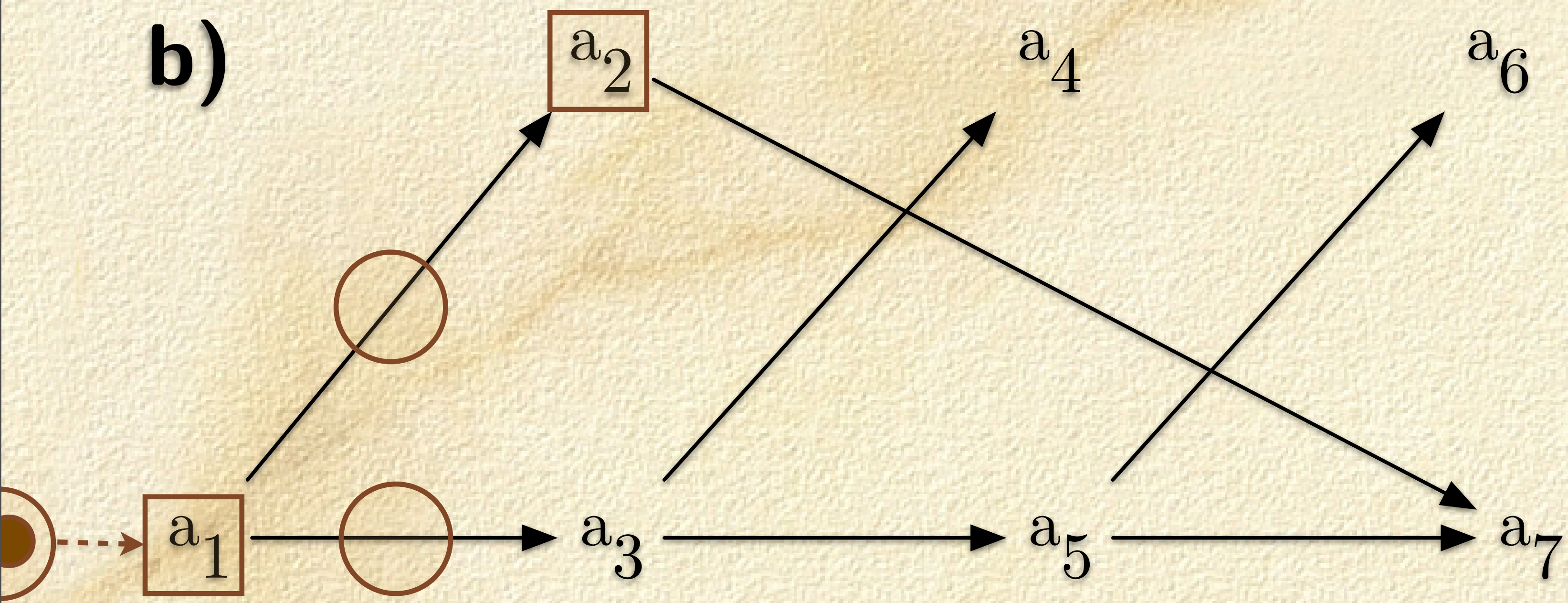



zugehörige **Präzedenzrelation** $\lessdot$

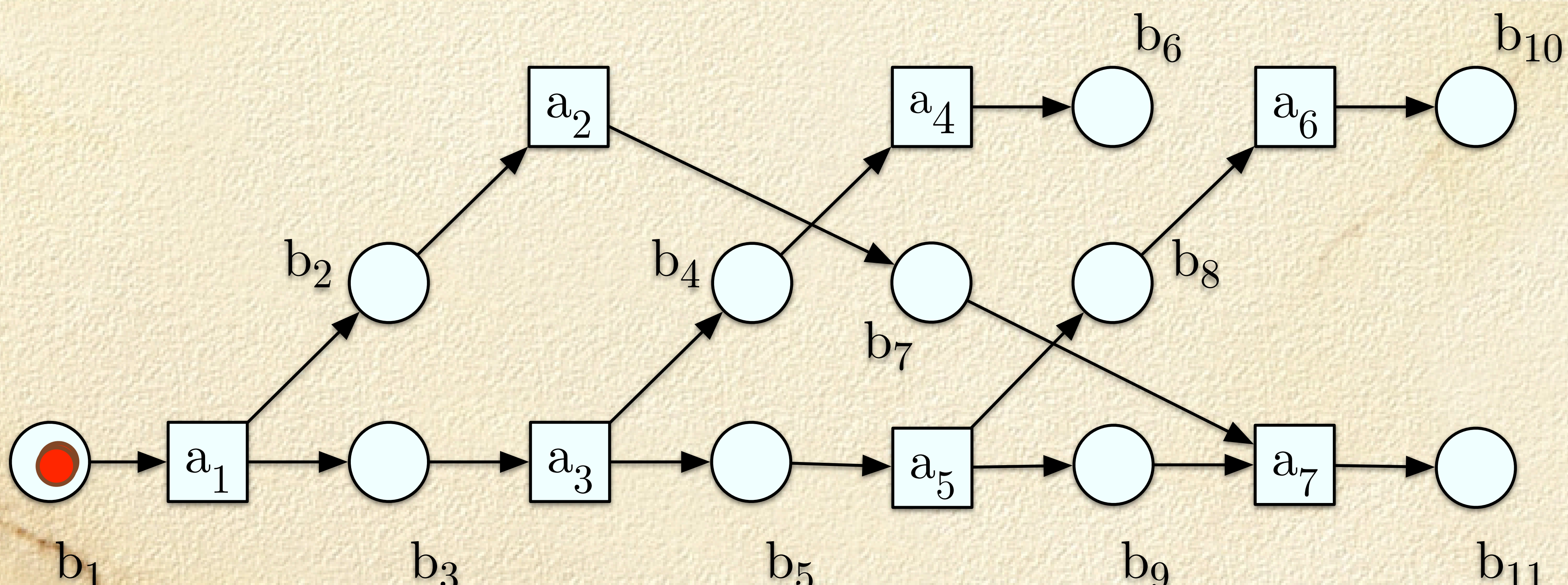
( nur direkte Nachfolger )



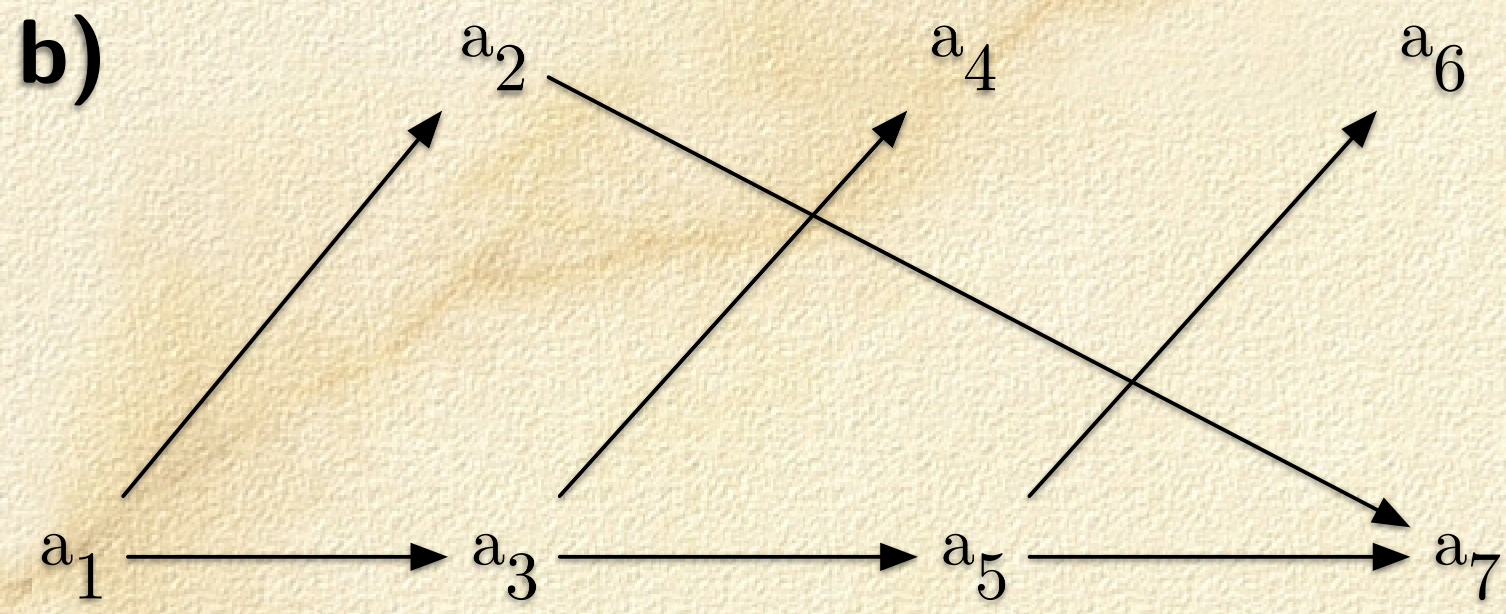
**b)**



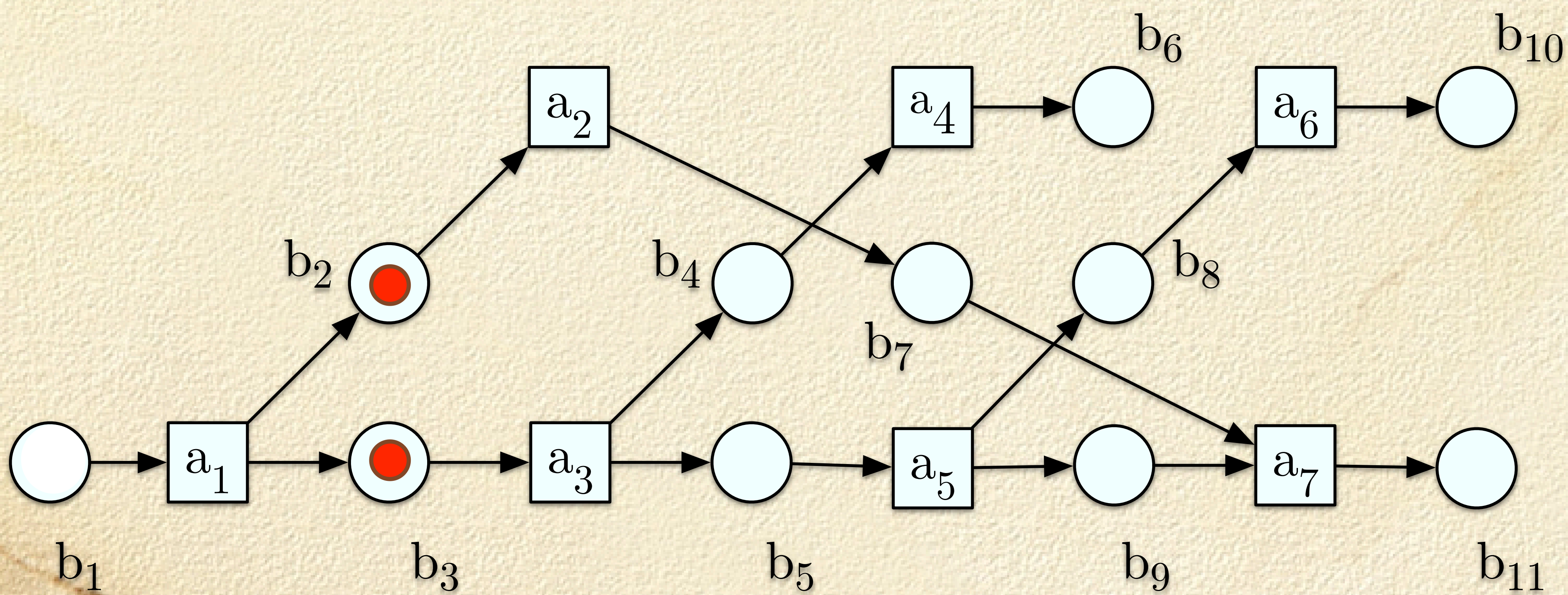

*als (Kausal-) Petrinetz*



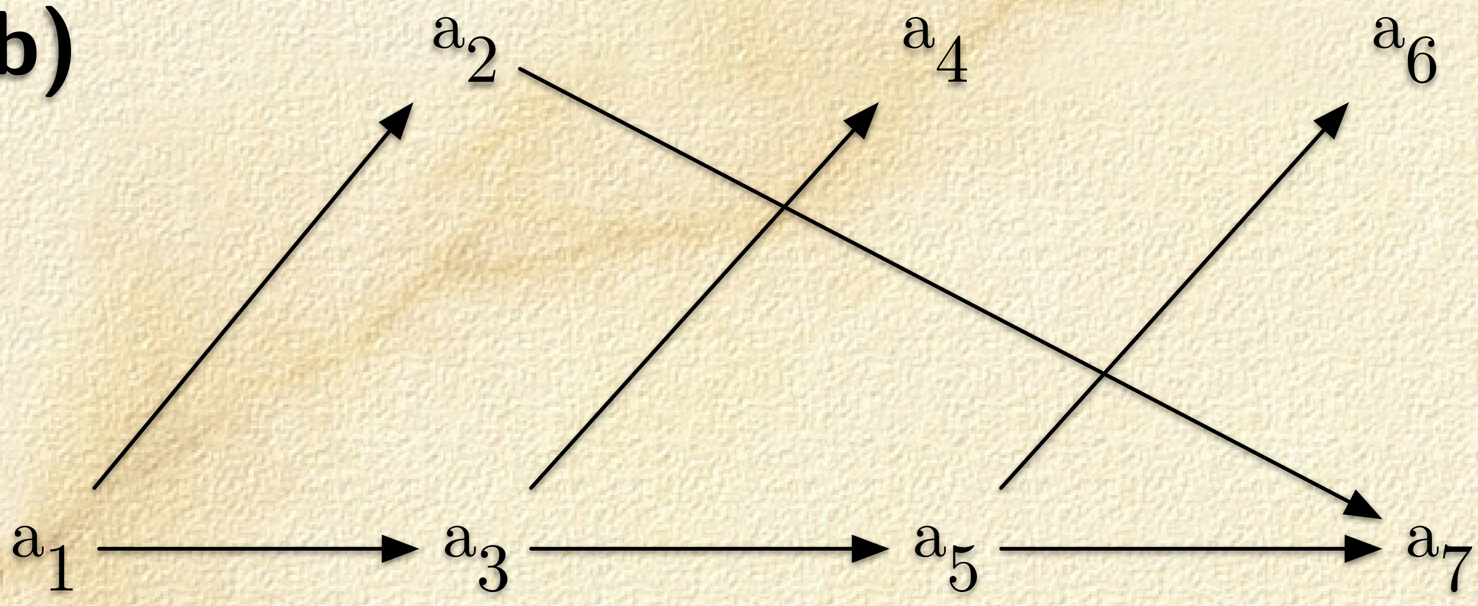
**b)**



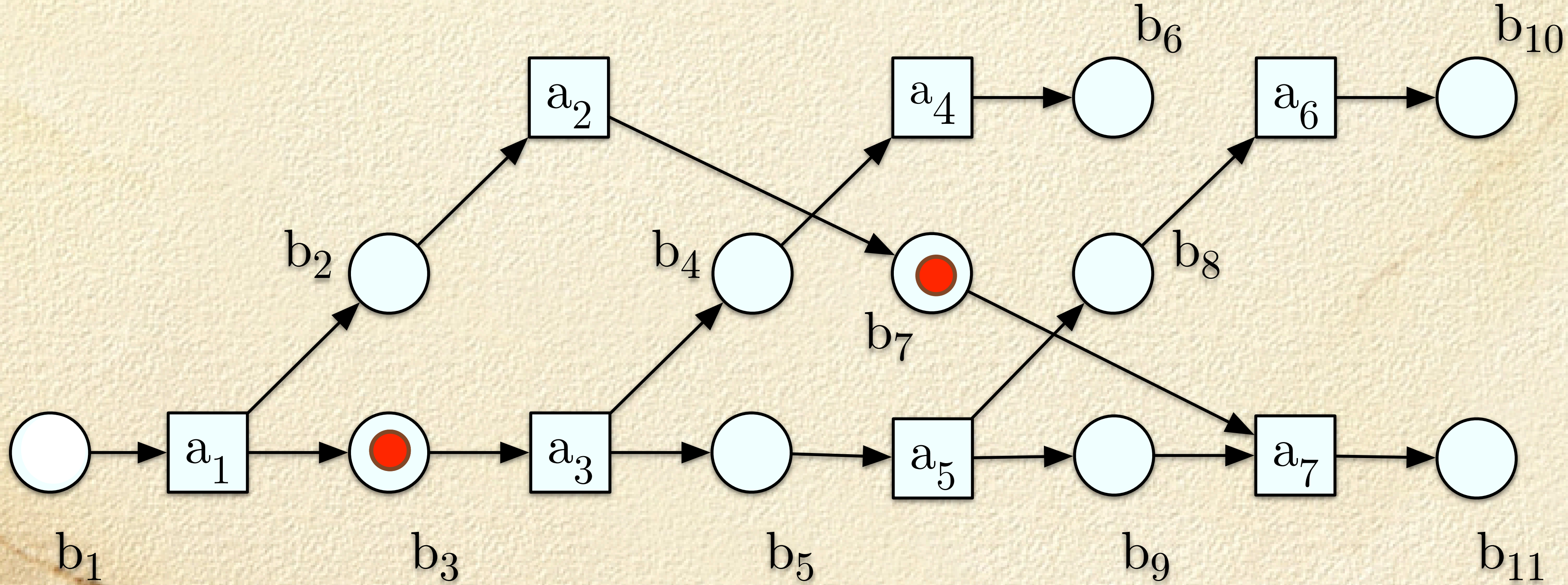

*als (Kausal-) Petrinetz*



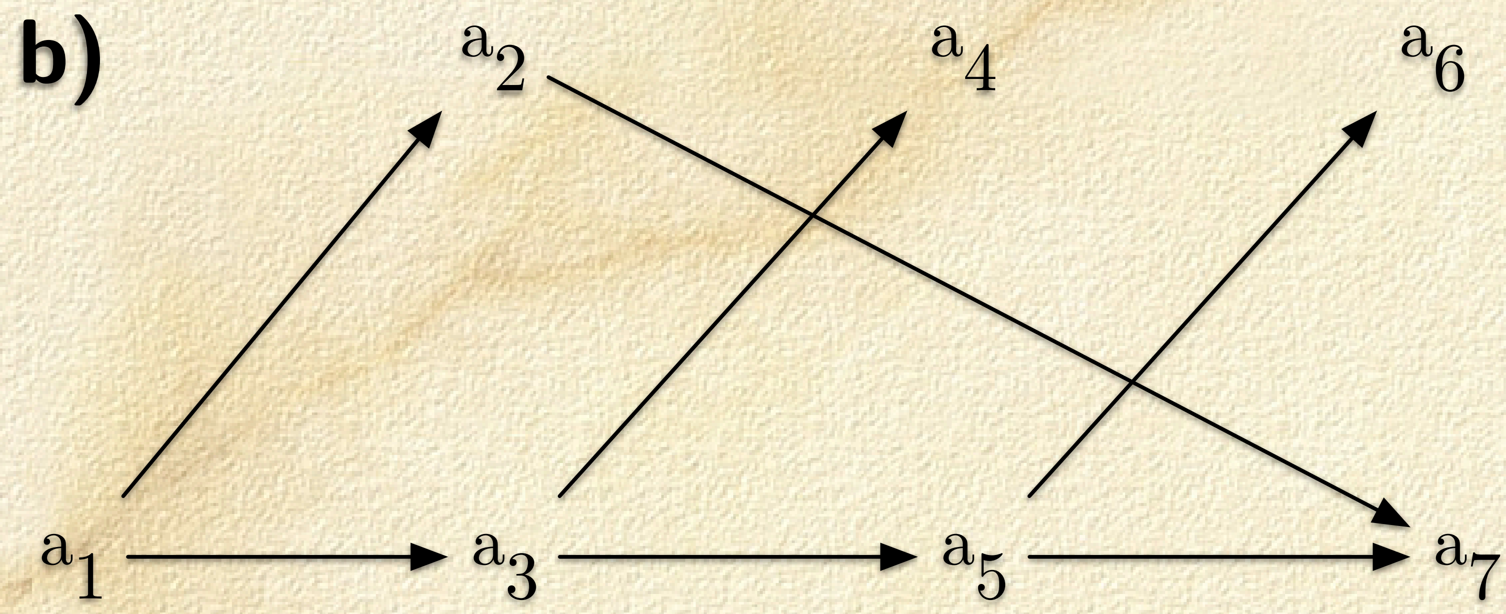
**b)**



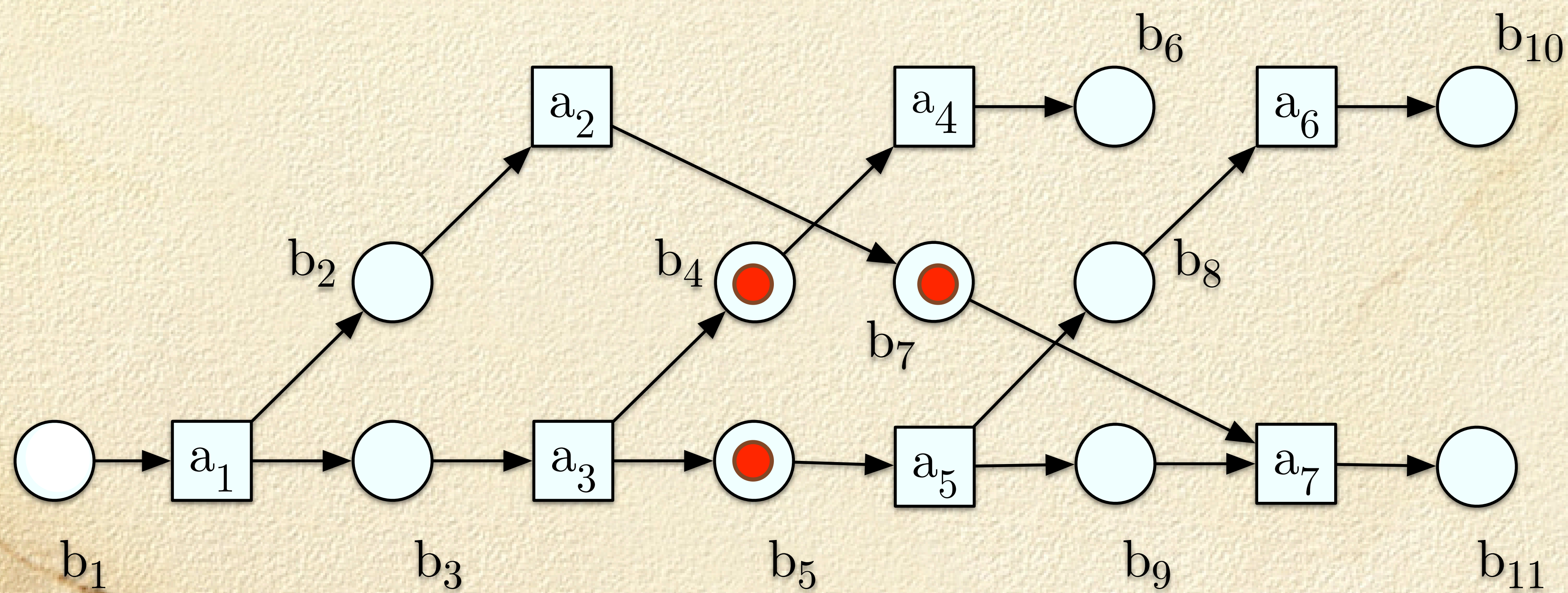

*als (Kausal-) Petrinetz*



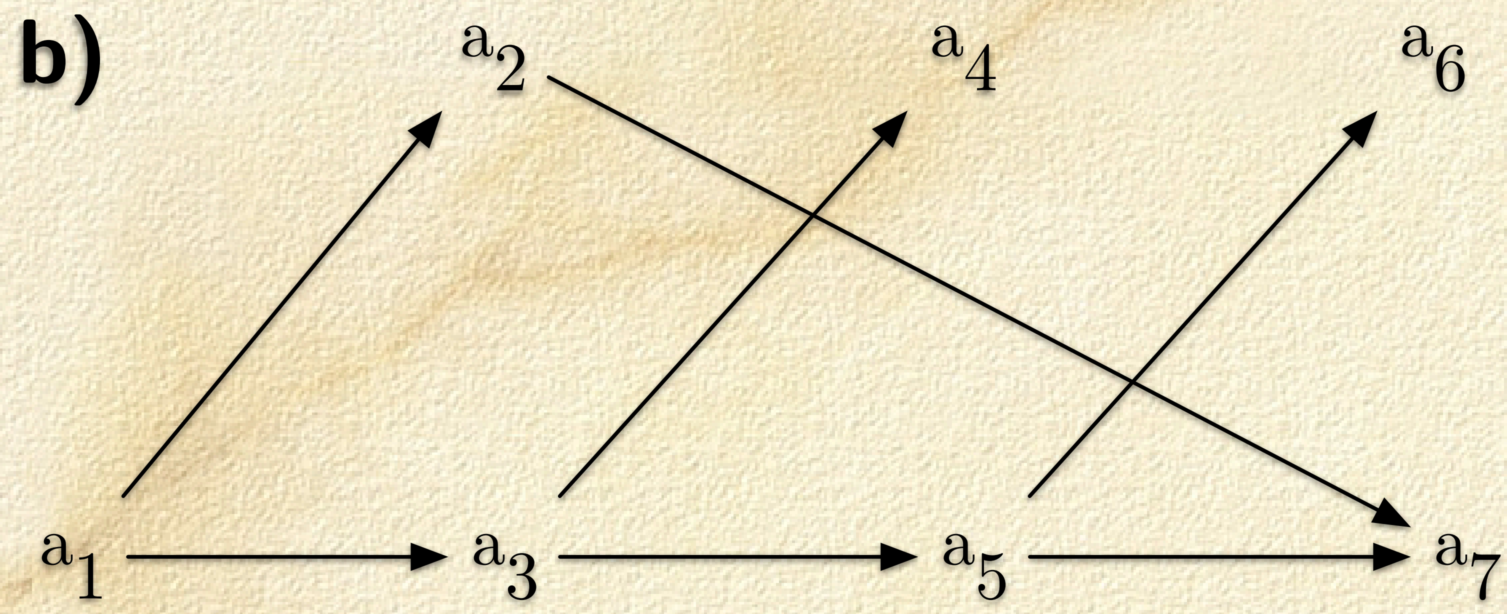
**b)**



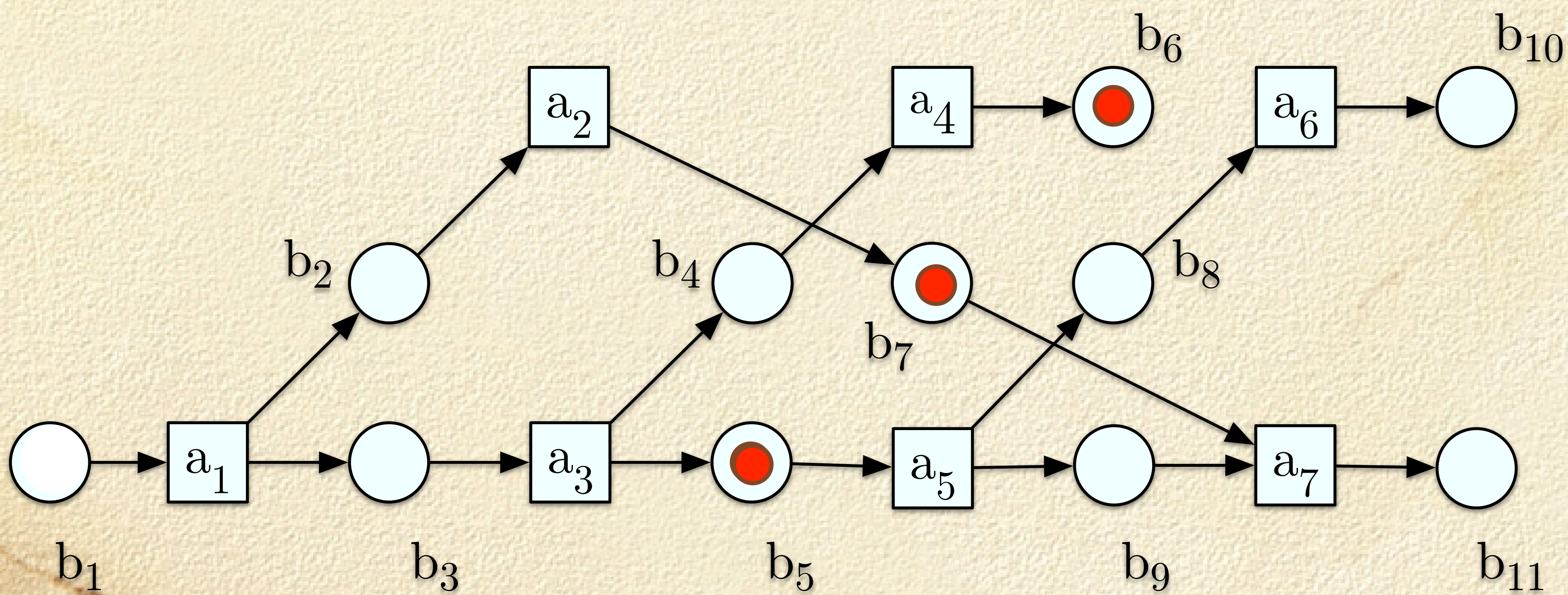

*als (Kausal-) Petrinetz*



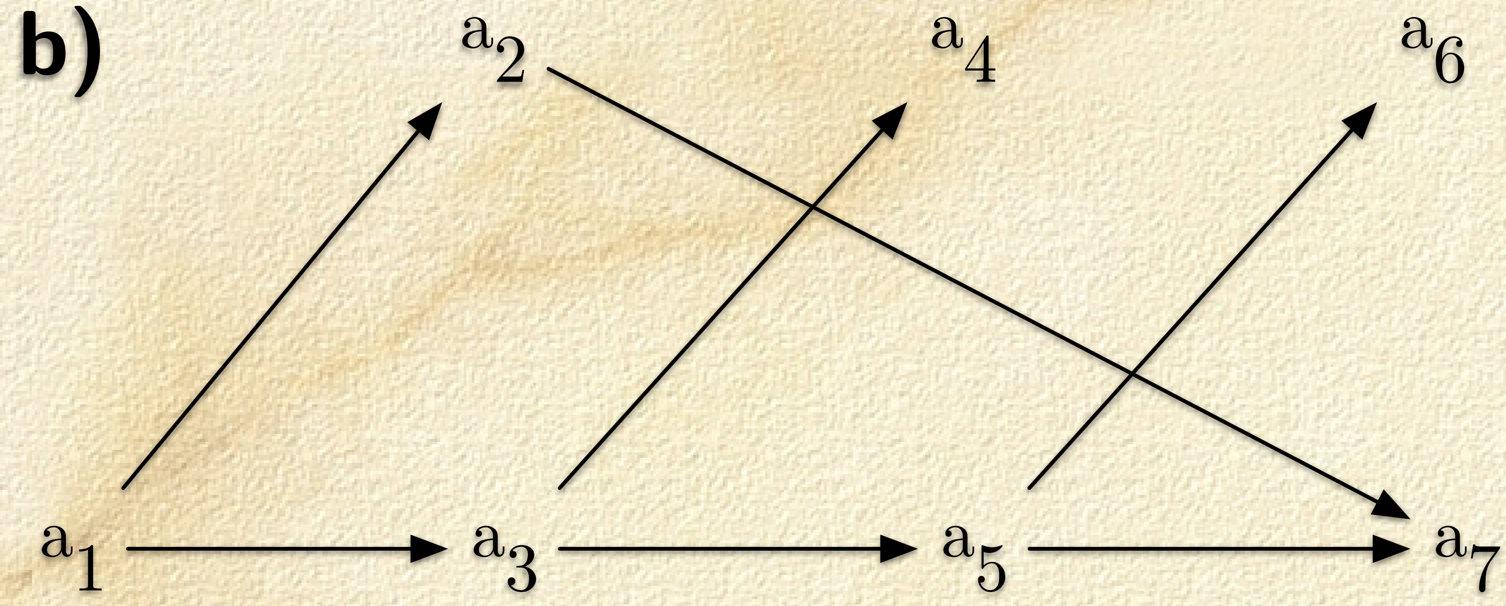
**b)**



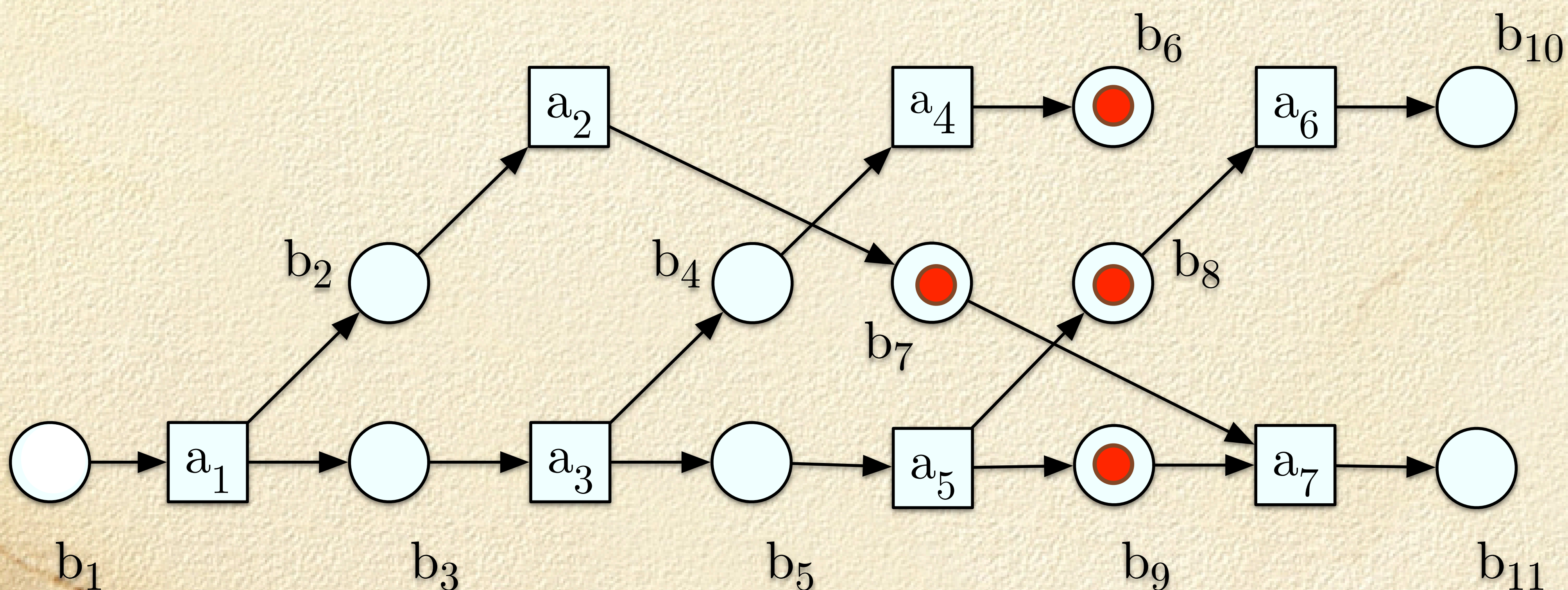

*als (Kausal-) Petrinetz*



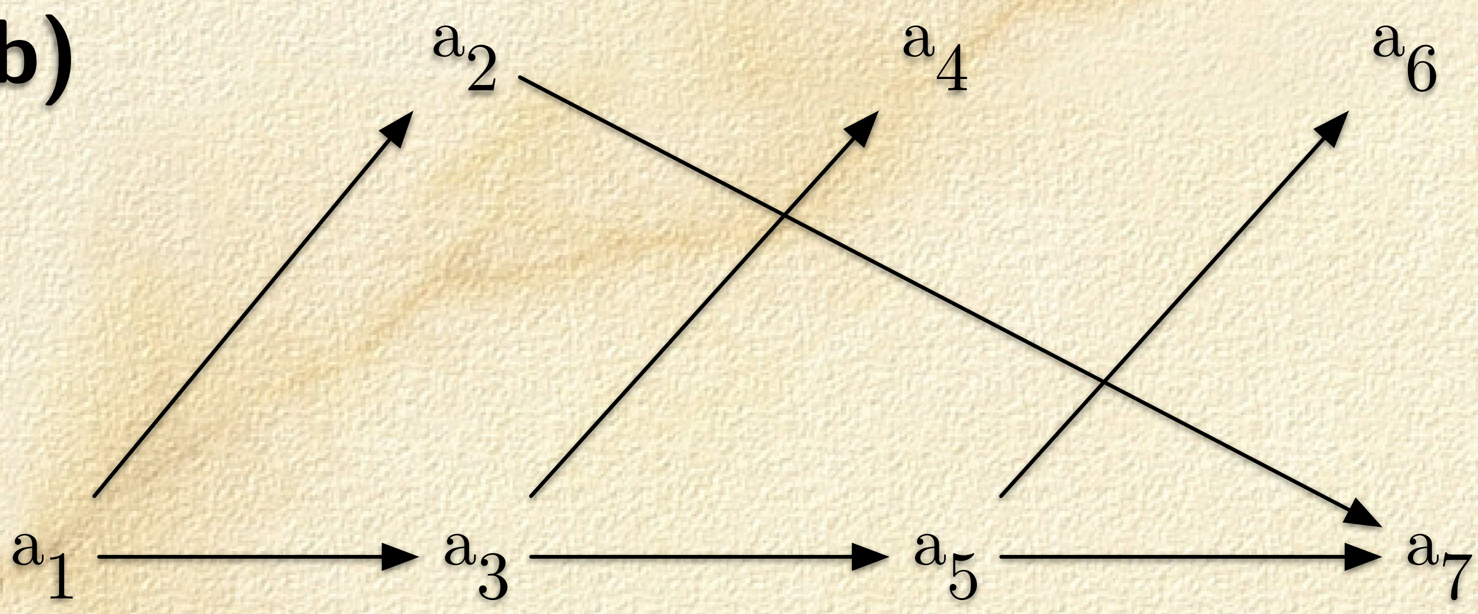
**b)**



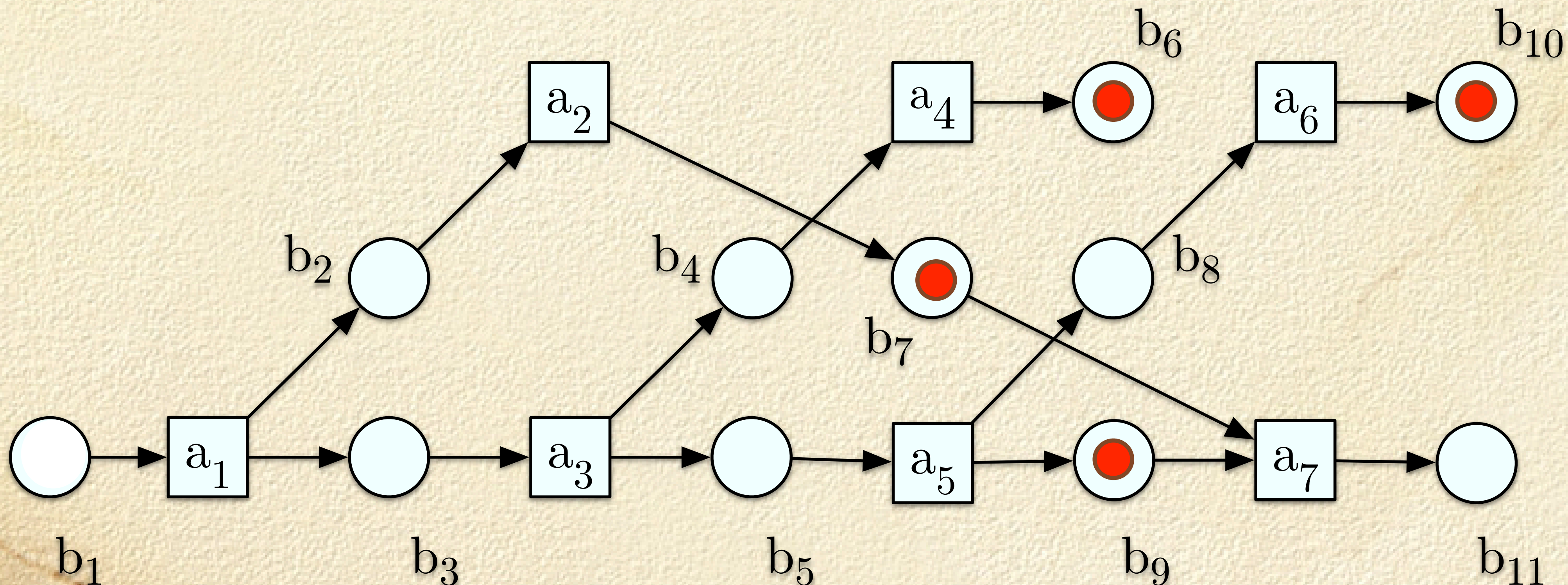

*als (Kausal-) Petrinetz*



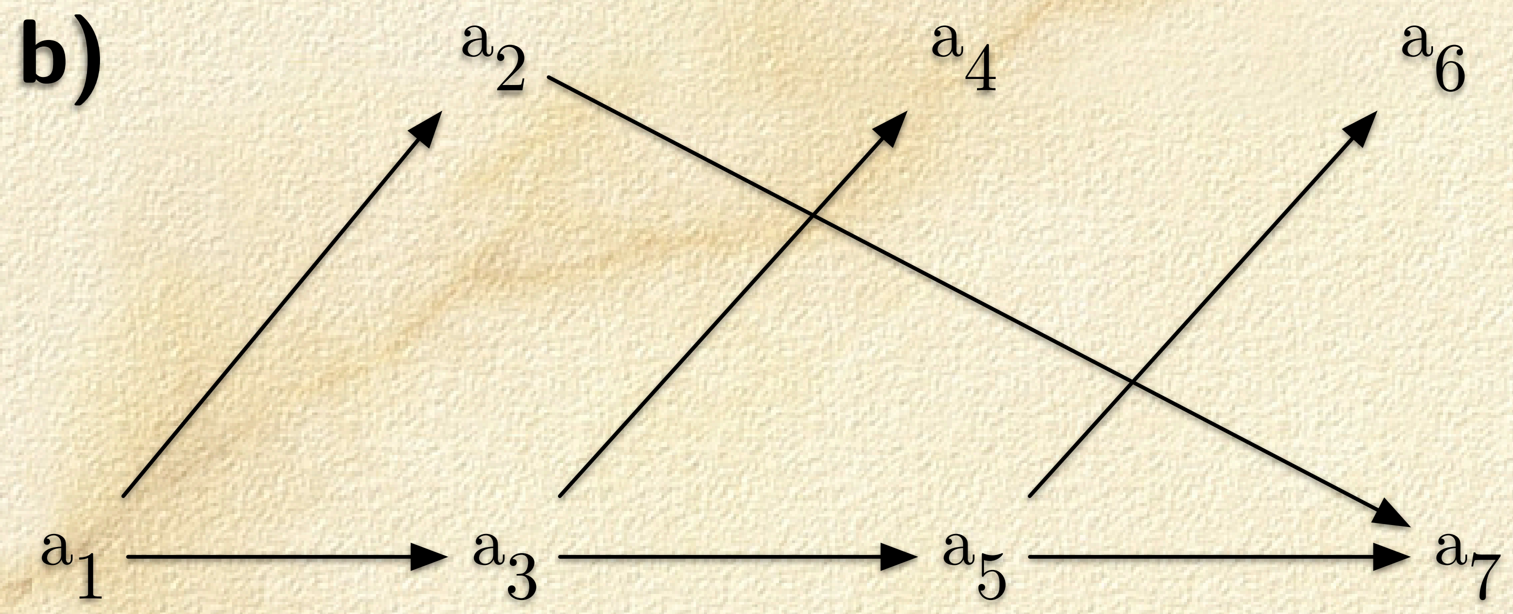
**b)**



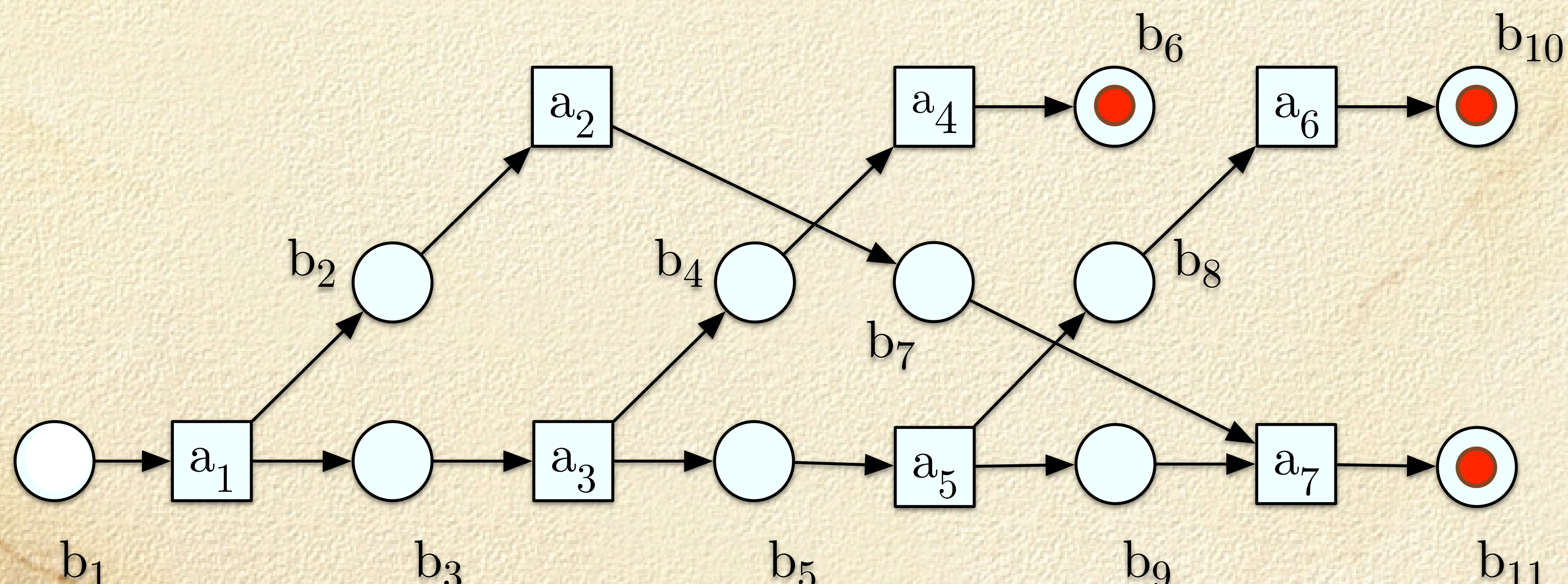

*als (Kausal-) Petrinetz*



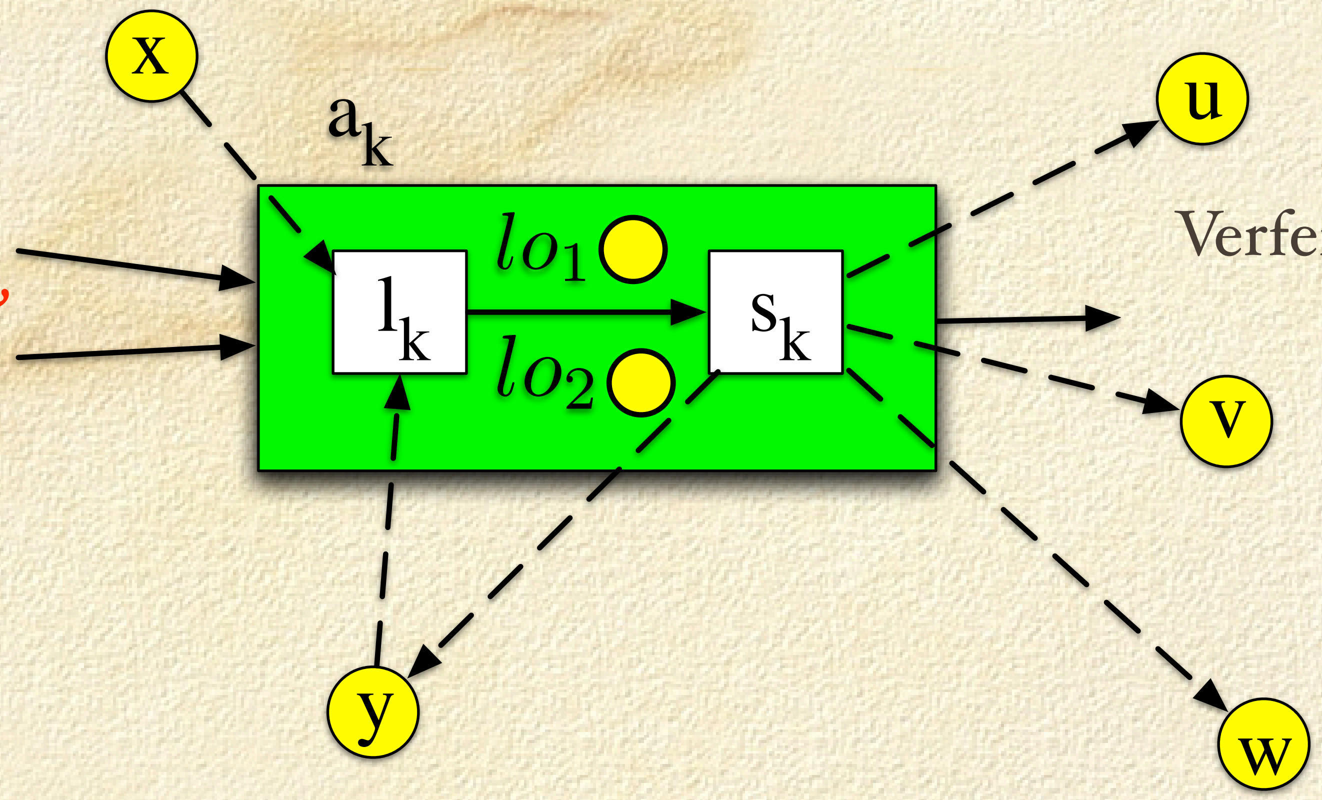
**b)**



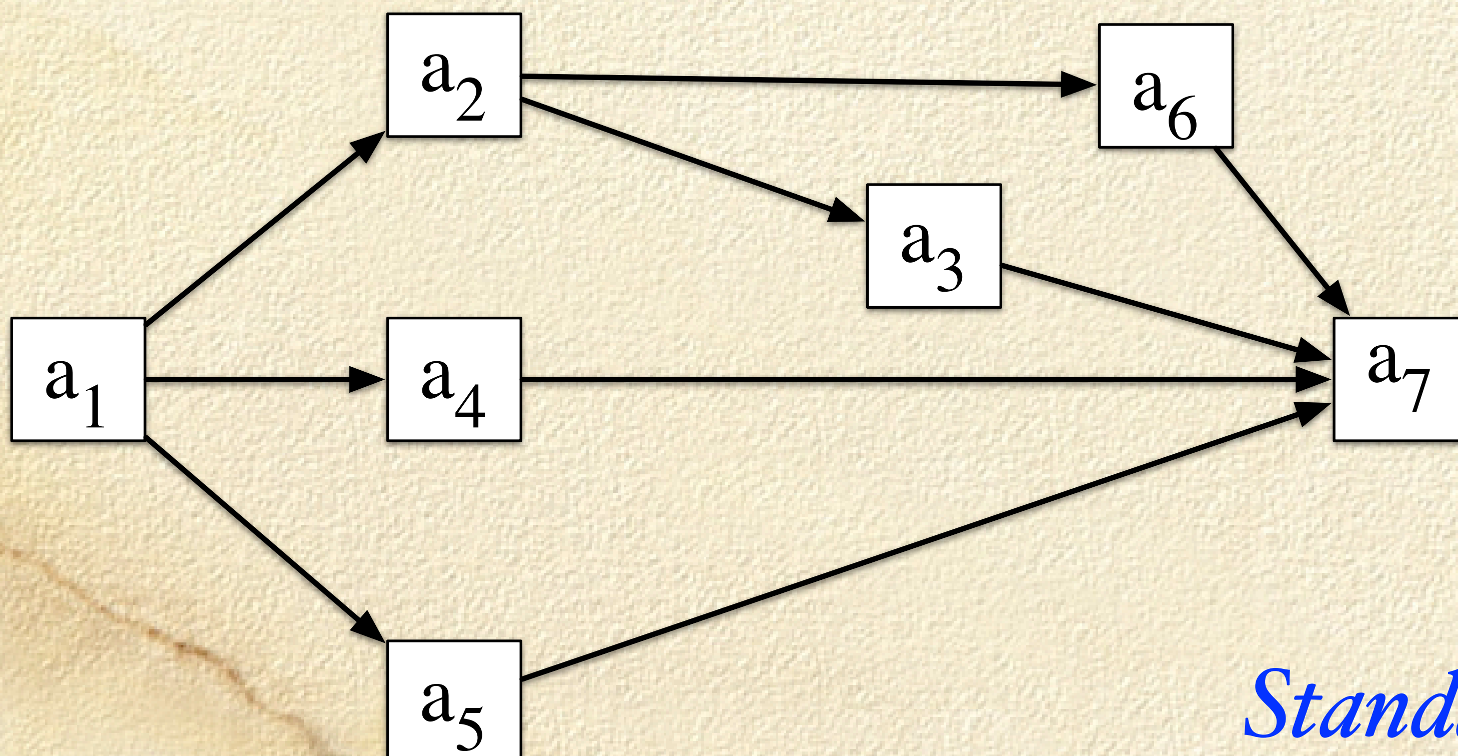

*als (Kausal-) Petrinetz*



*schematisches Auftragssystem*

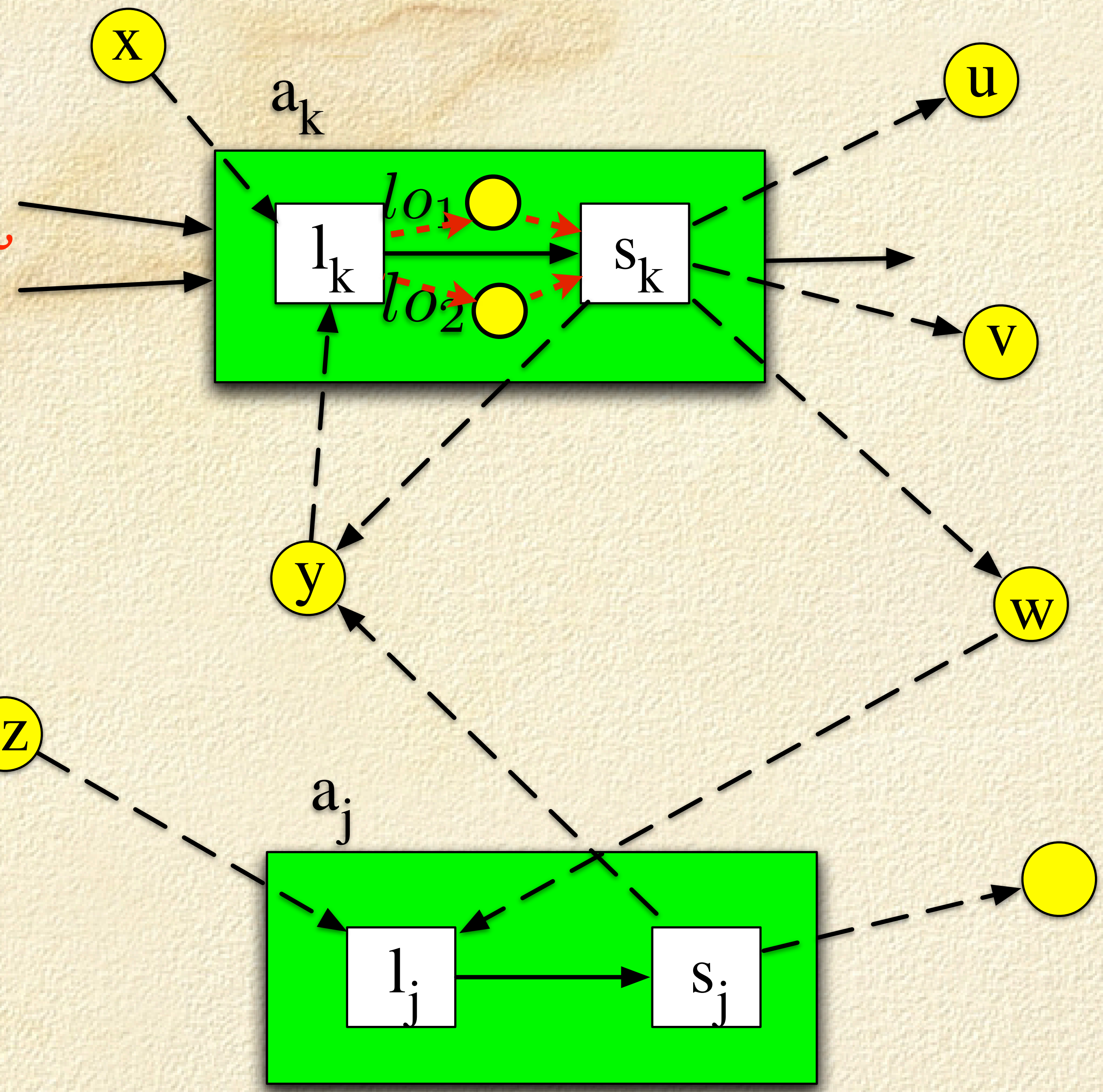


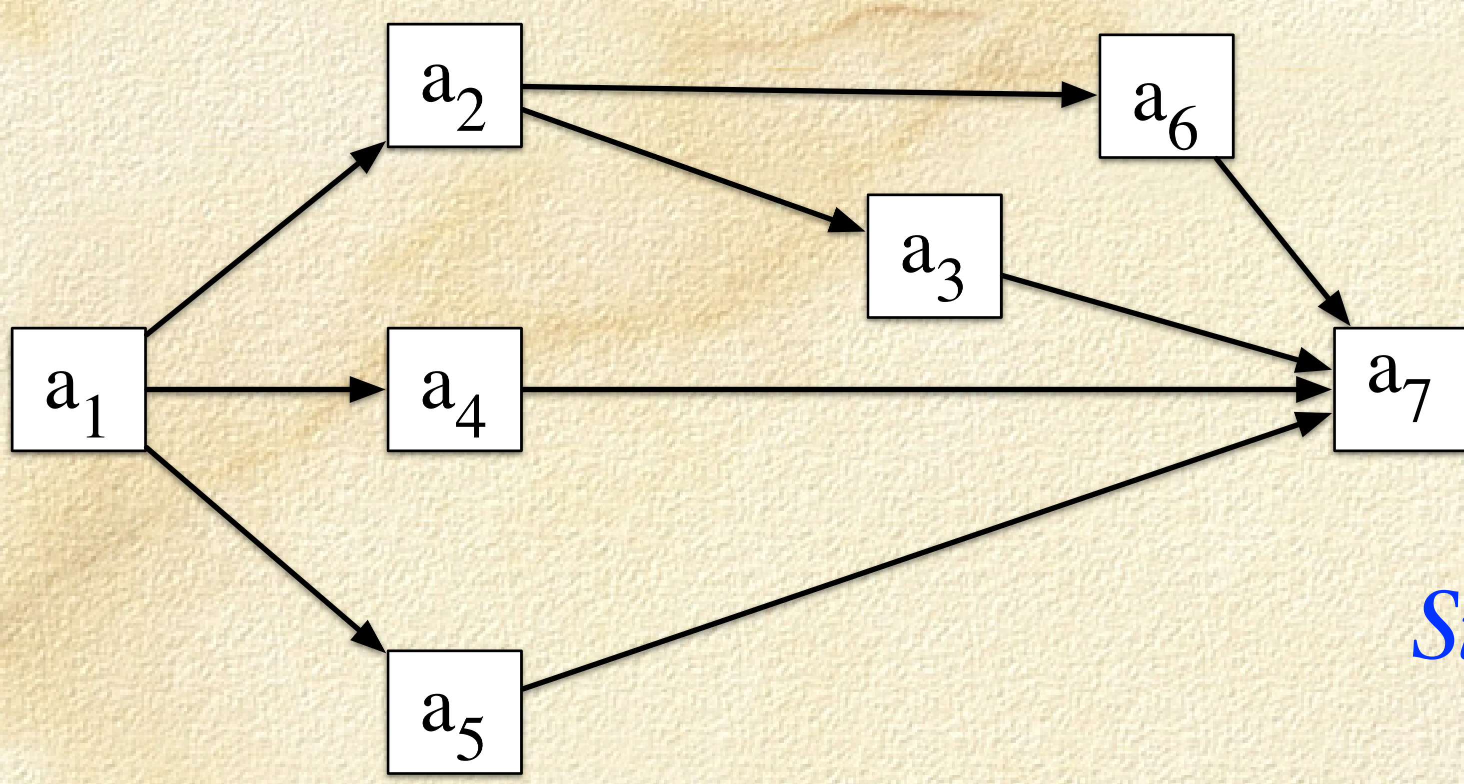

Verfeinerung in Lese- und Schreibaufträge



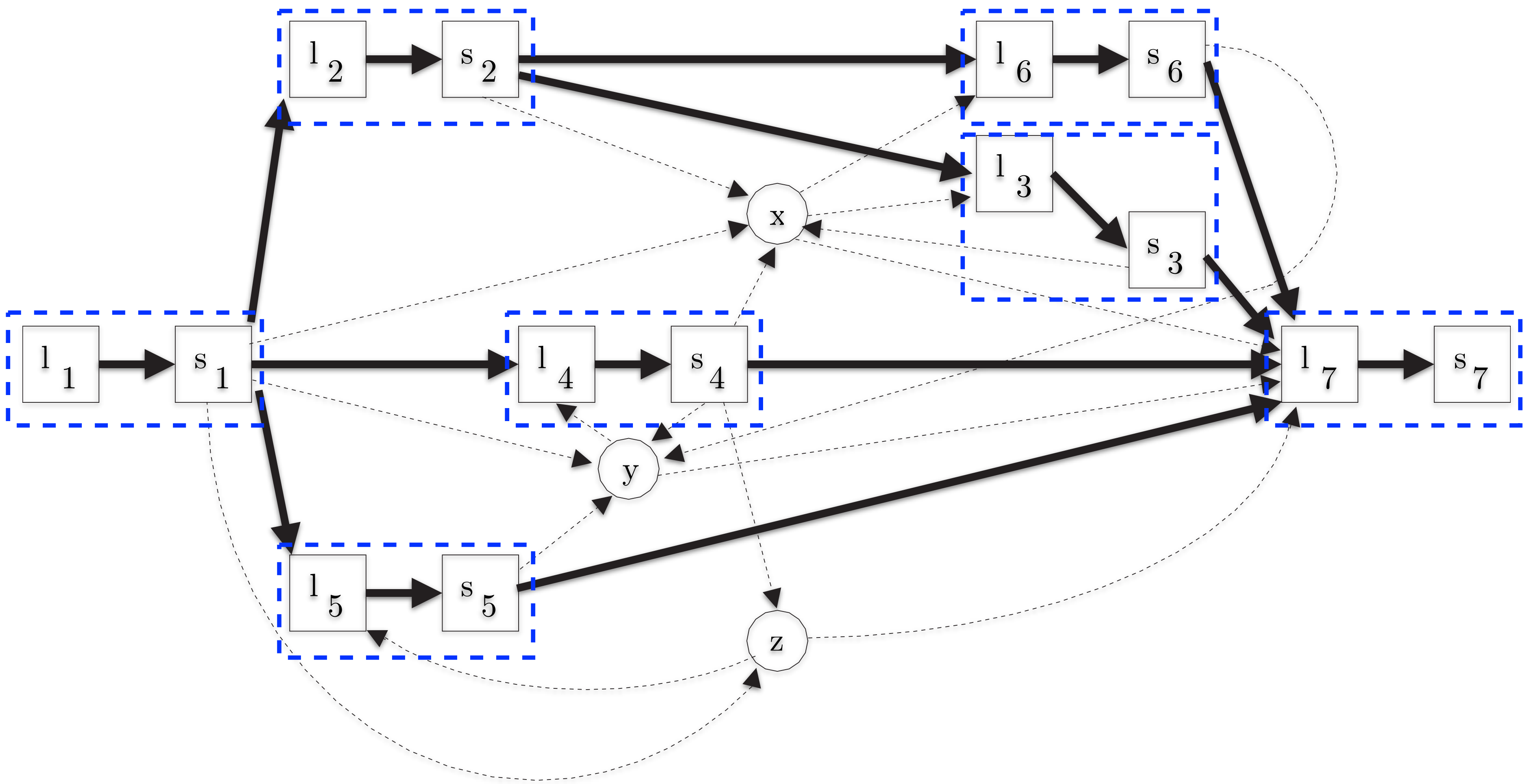

*Standardbeispiel*

*schematisches Auftragssystem*






*Standardbeispiel*



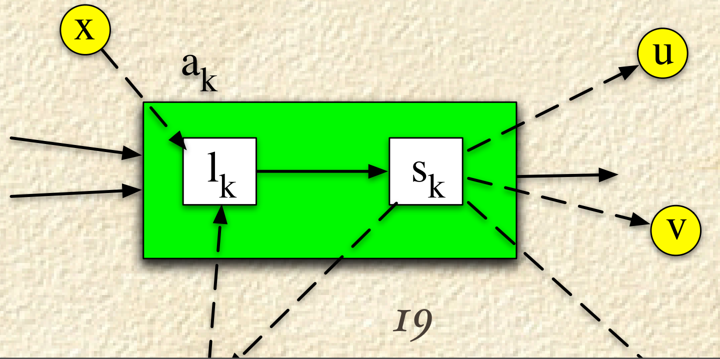

Verfeinertes Auftragssystem $AS'$ mit Darstellung der Schreib/ und Lese-Zugriffe

**Definition 6.10** *Ein Auftragssystem $AS = (A, <)$ (Def. 4.25) heißt* schematisch *oder* uninterpretiert, *wenn die Auftragsmenge die Form*

$$A = \{l_1, s_1, \ldots, l_n, s_n\} \quad (n \geq 1)$$

*hat. Gefordert werden auf jeden Fall die direkten Präzedenzen* $l_i \lessdot s_i$ $(1 \leq i \leq n)$. *Weiter können weitere direkte Präzedenzen der Form* $s_i \lessdot l_j$ $(1 \leq i, j \leq n)$ $(i \neq j)$ *bestehen.* $l_i$ *bzw.* $s_i$ *heißen Lese- bzw Schreib-Auftrag. Zusammen bilden sie den Auftrag* $a_i = (l_i, s_i)$. *Ferner sei eine endliche Menge* $V = \{v_1, v_2, \ldots, v_p\}$ *von Variablen gegeben. Zu jedem* $a_i$ $(1 \leq i \leq n)$ *ist dabei*

- *eine Menge* $ein_i := \{e_1, \ldots, e_{p_i}\} \subseteq V$ *von* *Eingangsvariablen* *und*
- *eine Menge* $aus_i := \{u_1, \ldots, u_{q_i}\} \subseteq V$ *von* *Ausgangsvariablen* *festgelegt.*

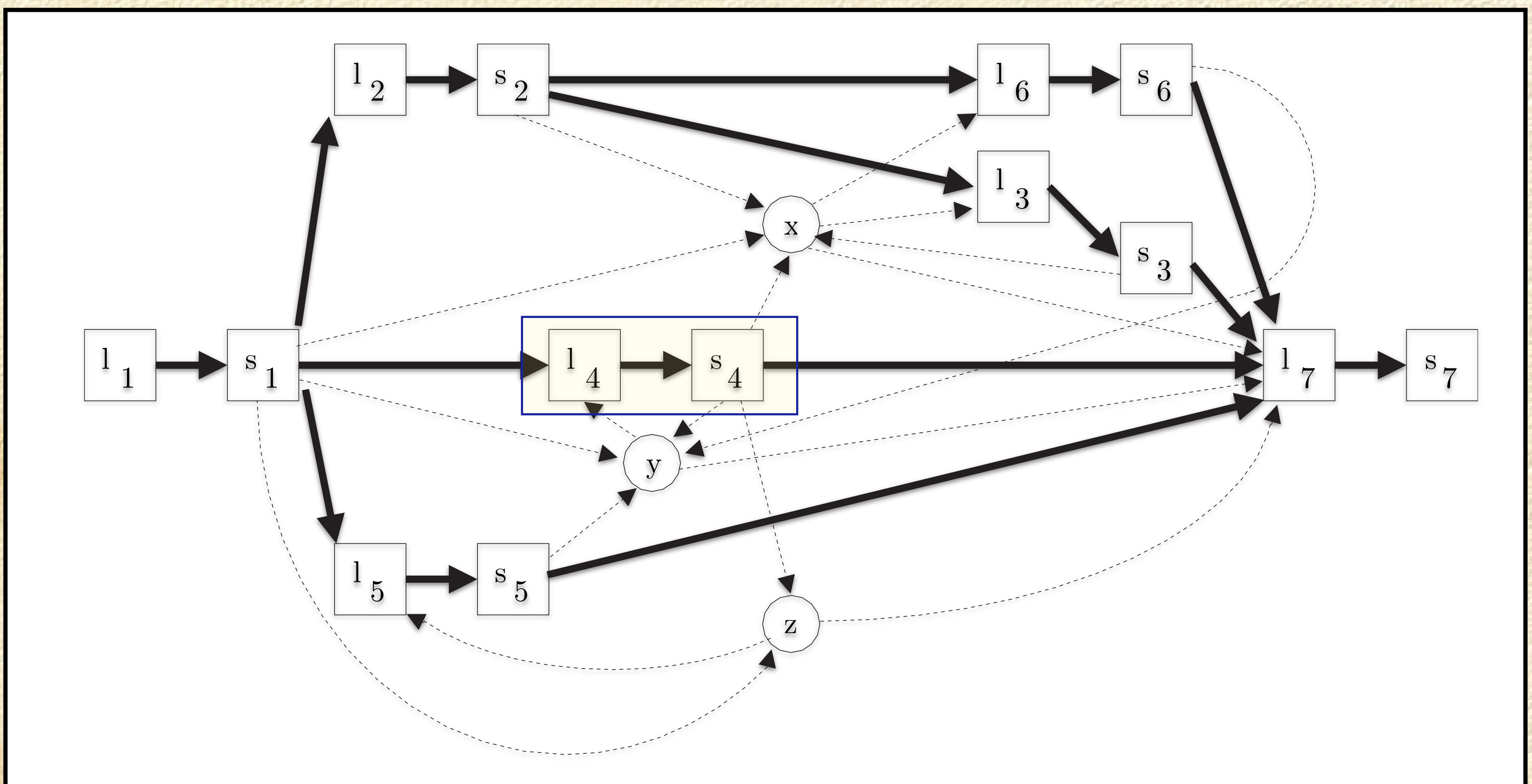


*AS kann somit eindeutig durch* $\lessdot$ *und die Schreibweise* $A = \boxed{\{l_1[ein_1], s_1[aus_1], \ldots, l_n[ein_n], s_n[aus_n]\}}$ *dargestellt werden. Ist* $A' =$

*erweiterte Darstellung*

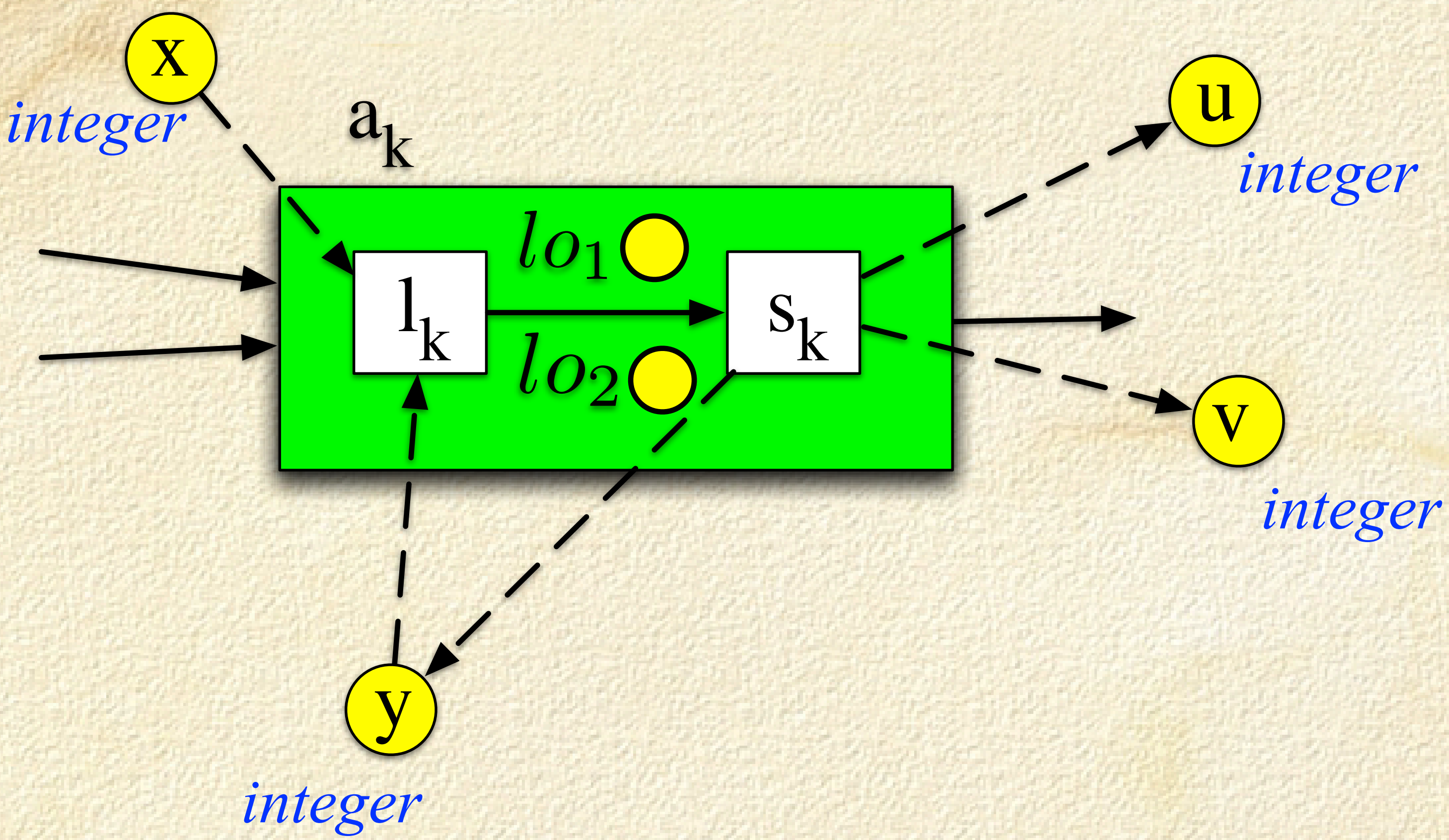
$$AS = (A, <)$$

$$A = \{l_1, s_1[xyz], l_2, s_2[x], l_3[x], s_3[x], \boxed{l_4[y], s_4[xyz]}, l_5[z], s_5[y], l_6[x], s_6[y], l_7[xyz], s_7\}$$



*„Interpretation"*

*„Modell"*




*„atomar"*

$$u := x + y$$
$$v := 2 \cdot x$$
$$y := x^y + y$$

*„geteilt"*

$$\frac{lo_1 := x \quad lo_2 := y}{\begin{array}{l} u := lo_1 + lo_2 \\ v := 2 \cdot lo_1 \\ y := lo_1^{lo_2} + lo_2 \end{array}}$$

*„schematisch"*

$$\frac{lo_1 := x \quad lo_2 := y}{\begin{array}{l} u := f_k^1(lo_1, lo_2) \\ v := f_k^2(lo_1, lo_2) \\ y := f_k^3(lo_1, lo_2) \end{array}}$$

**Definition 6.14** *Eine* $\boxed{\text{Interpretation } I}$ *eines schematischen Auftragssystems AS mit Bezeichnung wie in Definition 4.27 ist durch eine* $\boxed{\textit{Wertemenge } D_I}$ *(kurz $D$) und für jedes $1 \le i \le n$ durch* $\boxed{\text{Funktionen}}$

$$f_i^j : D^{p_i} \to D \quad (1 \le j \le q_i)$$

*sowie durch einen* $\boxed{\textit{Anfangszustand } d_0 \in D^p}$ *gegeben. Ist $p_i = 0$, dann ist $f_i^j \in D$ eine Konstante, ist $q_i = 0$, dann entfällt $f_i^j$.*

*Ein Zustand ist ein Vektor $d \in D^p$ der Länge $p$ und ordnet jeder Variablen $v_i \in V$ einen Wert $d_i \in D$ zu.*

$$u := f_k^1(lo_1, lo_2)$$
$$v := f_k^2(lo_1, lo_2)$$
$$w := f_k^3(lo_1, lo_2)$$

*Zu einer Ausführungsfolge* $w = w_1 w_2 \ldots w_{2n} \in F_E(AS)$ *ist eine Zustandsfolge* $d_0, d_1, d_2, \ldots, d_{2n}$ *mit* $d_i \in D^p$ *folgendermaßen definiert:*

- $d_0$ *ist der Anfangszustand*
- *Ist* $w_j = l_i$ *ein Leseauftrag* $(1 \leq j \leq 2n)$, *dann ist* $d_j = d_{j-1}$ *und für eine nur in* $a_i$ *vorkommende Menge von lokalen Variablen* $lok_i := \{lo_1, \ldots, lo_{p_i}\}$ *erhalten wir die Werte:*

$$lo_1, lo_2, \ldots, lo_{p_i} := e_1, e_2, \ldots, e_{p_i}$$

$lo_1 := x$
$lo_2 := y$

- *Ist* $w_j = s_i$ *ein Schreibauftrag, dann entsteht* $d_j$ *aus* $d_{j-1}$ *durch die Zuweisung:*

$$u_1, \ldots, u_{q_i} := f_i^1(lo_1, \ldots, lo_{p_i}), \ldots, f_i^{q_i}(lo_1, \ldots, lo_{p_i})$$

$u := lo_1 + lo_2$
$v := 2 \cdot lo_1$
$w := lo_1^{lo_2} + lo_2$

*Mit* $res(w, I) := d_{2n}$ *bezeichnen wir das Resultat von* $w$ *bezüglich der Interpretation* $I$.

*Zwei Ausführungsfolgen* $w_1, w_2 \in F_E(AS)$ *heißen äquivalent, wenn*

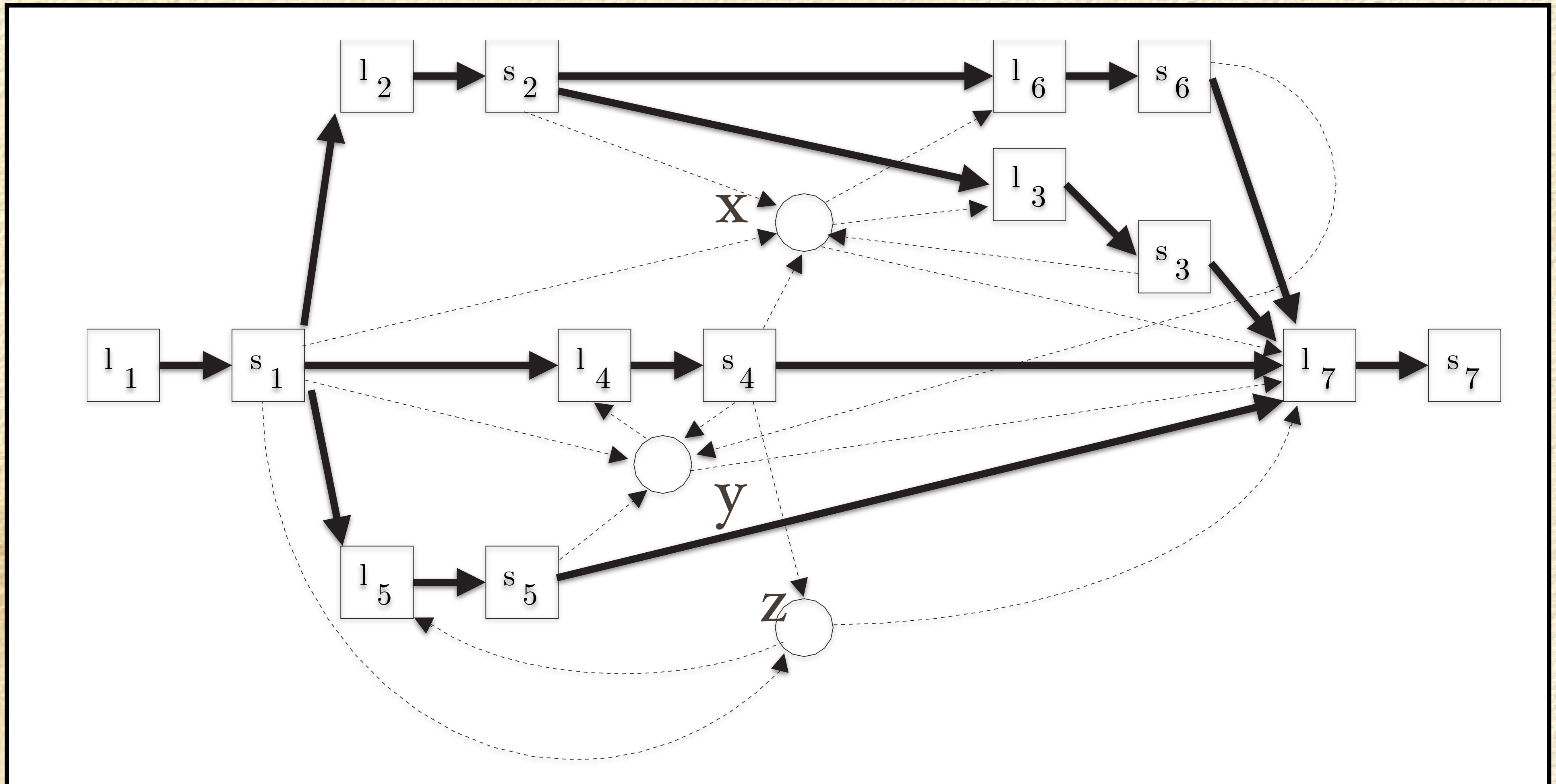
$$res(w_1, I) = res(w_2, I)$$

*für* alle *Interpretationen* $I$ *von* $AS$ *gilt.*

Beispiel **6.15** die Interpretation $I$ mit $D_I = \mathbb{Z}$ und $d_0 = (0, 0, 0)$

$$
\begin{aligned}
f_1^1 &= 0, f_1^2 = 1, f_1^3 = 2, \\
f_2^1 &= 20 \\
f_3^1(lo_3) &= lo_3 + 5 \\
f_4^1(lo_4) &= lo_4 - 1, f_4^2(lo_4) = lo_4 + 10, f_4^3(lo_4) = lo_4 \\
f_5^1(lo_5) &= lo_5 \\
f_6^1(lo_6) &= 2 \cdot lo_6
\end{aligned}
$$

$w = l_1 \; s_1 \; l_2 \; s_2 \; l_4 \; l_3 \; l_6 \; s_4 \; l_5 \; s_3 \; s_6 \; s_5 \; l_7 \; s_7$

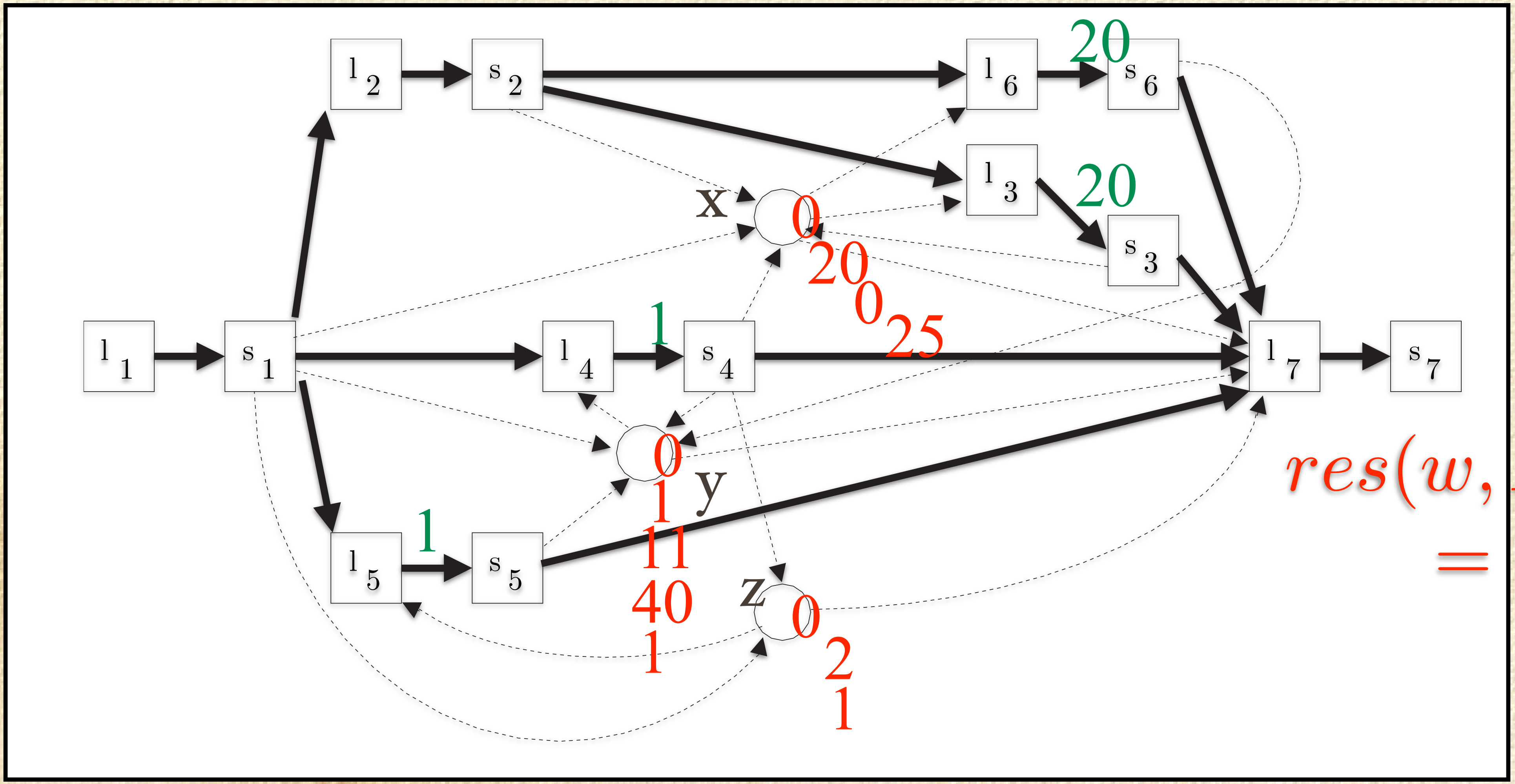


Beispiel **6.15** die Interpretation $I$ mit $D_I = \mathbb{Z}$ und $d_0 = (0,0,0)$

$$
\begin{aligned}
f_1^1 &= 0, f_1^2 = 1, f_1^3 = 2, \\
f_2^1 &= 20 \\
f_3^1(lo_3) &= lo_3 + 5 \\
f_4^1(lo_4) &= lo_4 - 1, f_4^2(lo_4) = lo_4 + 10, f_4^3(lo_4) = lo_4 \\
f_5^1(lo_5) &= lo_5 \\
f_6^1(lo_6) &= 2 \cdot lo_6
\end{aligned}
$$

$w = l_1\ s_1\ \ l_2\ s_2\ \ l_4\ \ l_3\ \ l_6\ s_4\ \ l_5\ s_3\ \ s_6\ \ s_5\ \ l_7\ \ s_7$

| | $d_0$ | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ | $d_6$ | $d_7$ | $d_8$ | $d_9$ | $d_{10}$ | $d_{11}$ | $d_{12}$ | $d_{13}$ | $d_{14}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $x$ | 0 | 0 | 0 | 0 | 20 | 20 | 20 | 20 | 0 | 0 | 25 | 25 | 25 | 25 | 25 |
| $y$ | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 11 | 11 | 11 | 40 | 1 | 1 | 1 |
| $z$ | 0 | 0 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |




*Resultat*

$res(w, I) := d_{2n}$
$= (25, 1, 1)$

## Kriterien für Datenkonsistenz

- *inhaltlich (semantisch)*

  z.B. Einstellungsalter ≤ Lebensalter

- *formal*

  z.B. aufgrund des Ablaufes

*Ausführungsfolge:*

Resultat

$$w = l_1 \; s_1 \; l_2 \; s_2 \; l_4 \; l_3 \; l_6 \; s_4 \; l_5 \; s_3 \; s_6 \; s_5 \; l_7 \; s_7$$

$$res(w, I) := d_{2n} = (25, 1, 1)$$

*serielle Ausführungsfolge:*

$$w_2 = l_1 \; s_1 \; l_4 \; s_4 \; l_2 \; s_2 \; l_6 \; s_6 \; l_5 \; s_5 \; l_3 \; s_3 \; l_7 \; s_7$$

*konsistenter Zustand*

$$res(w_2, I) := d_{2n} = (25, 1, 1)$$

## *serielle Ausführungsfolge:*

$$w = \boxed{l_1\ s_1}\boxed{l_4\ s_4}\boxed{l_2\ s_2}\boxed{l_6\ s_6}\boxed{l_5\ s_5}\boxed{l_3\ s_3}\boxed{l_7\ s_7}$$

**Definition 6.21** *Eine Ausführungsfolge* $w = w_1 w_2 \ldots w_{2n} \in F(AS)$ *eines schematischen Auftragssystems* $AS = (A, <)$ *heißt* seriell, *wenn für alle* $1 \leq i < 2n$ *gilt:*

$$w_i = l_k \Rightarrow w_{i+1} = s_k$$

*d.h. Leseauftrag* $l_k$ *und Schreibauftrag* $s_k$ *eines Auftrages* $a_k$ *werden (ungeteilt) hintereinander ausgeführt.*

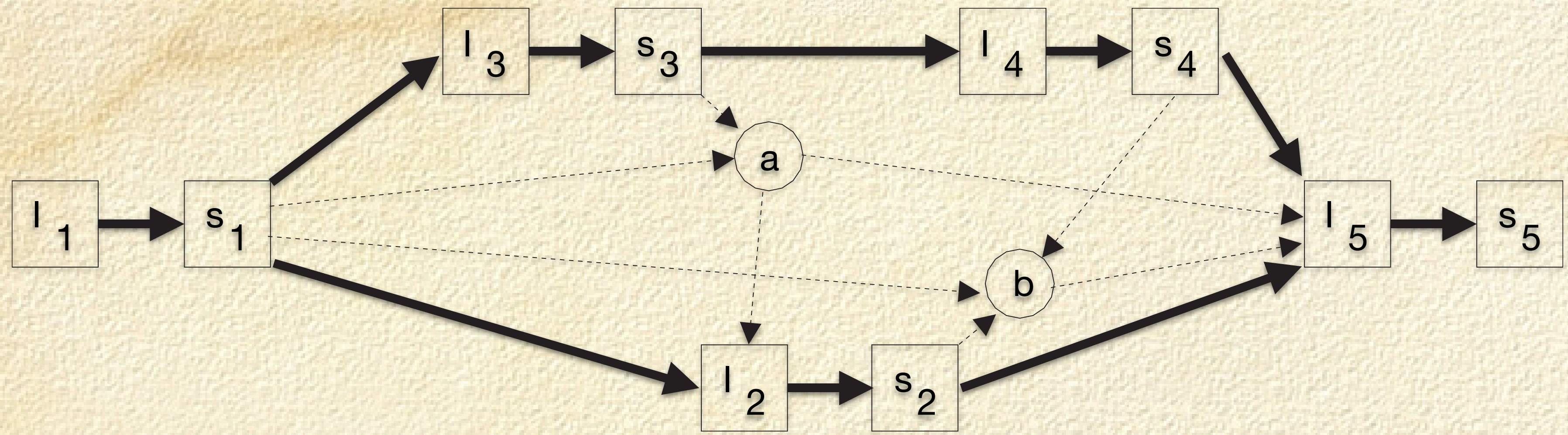
- $w \in F(AS)$ *heißt* Serialisierung *von* $w' \in F(AS)$, *wenn* $w$ *seriell und äquivalent zu* $w'$ *ist.*
- $w' \in F(AS)$ *heißt* serialisierbar *(serializable), wenn* $w'$ *eine Serialisierung* $w \in F(AS)$ *besitzt.*

## *Ausführungsfolge:*

$$w' = l_1\ s_1\ l_2\ s_2\ l_4\ l_3\ l_6\ s_4\ l_5\ s_3\ s_6\ s_5\ l_7\ s_7$$

# *Gegenbeispiel:*

$$w = \mathbf{l}_1 s_1[a,b]\ \mathbf{l}_2[a]\ \mathbf{l}_3 s_3[a]\ \mathbf{l}_4 s_4[b]\ s_2[b]\ \mathbf{l}_5[a,b]\ s_5$$




$$w = \mathbf{l}_1 s_1[a,b]\ \mathbf{l}_2[a]\ \mathbf{l}_3 s_3[a]\ \mathbf{l}_4 s_4[b]\ s_2[b]\ \mathbf{l}_5[a,b]\ s_5$$

Für jede Serialisierung $w''$ von $w$ muss gelten:

a) $\boxed{a_4 <_{w''} a_2, \text{ da beide auf } b \text{ schreiben}}$

b) $\boxed{a_2 <_{w''} a_3, \text{ da } a_2 \text{ den Wert von } a \text{ liest und } a_3 \text{ auf } a \text{ schreibt}}$

Aus a) und b) folgt $a_4 <_{w''} a_3$ im Widerspruch zur Präzedenz $a_3 \lessdot a_4$.
Also ist $w$ nicht serialisierbar.

# Ist das entscheidbar?

*Gibt es einen Algorithmus, der das prüft?*

- $w' \in F(AS)$ *heißt* $\boxed{\text{serialisierbar}}$ *(serializable), wenn* $w'$ *eine Serialisierung* $w \in F(AS)$ *besitzt.*

*Ausführungsfolge:*

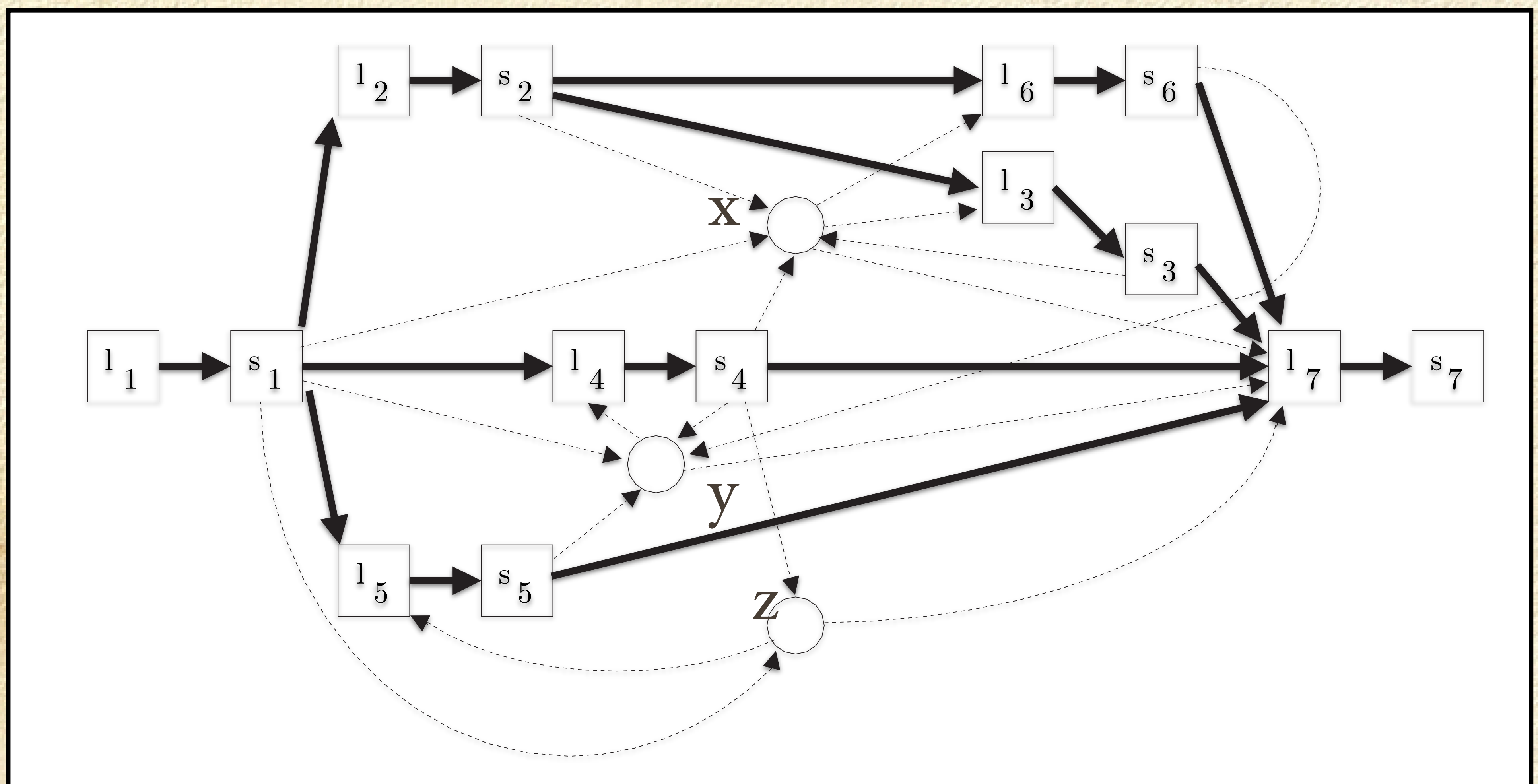
$$w = l_1\ s_1\ l_2\ s_2\ l_4\ l_3\ l_6\ s_4\ l_5\ s_3\ s_6\ s_5\ l_7\ s_7$$

## *Ausführungsfolge:*

$$w = l_1\ s_1\ l_2\ s_2\ l_4\ l_3\ l_6\ s_4\ l_5\ s_3\ s_6\ s_5\ l_7\ s_7$$

$$w = l_1 s_1[xyz]\ l_2 s_2[x]\ l_4[y]\ l_3[x]\ l_6[x]\ s_4[xyz]\ l_5[z]\ s_3[x]\ s_6[y]\ s_5[y]\ l_7[xyz]\ s_7$$

*Welchen Wert von y liest der Auftrag?*

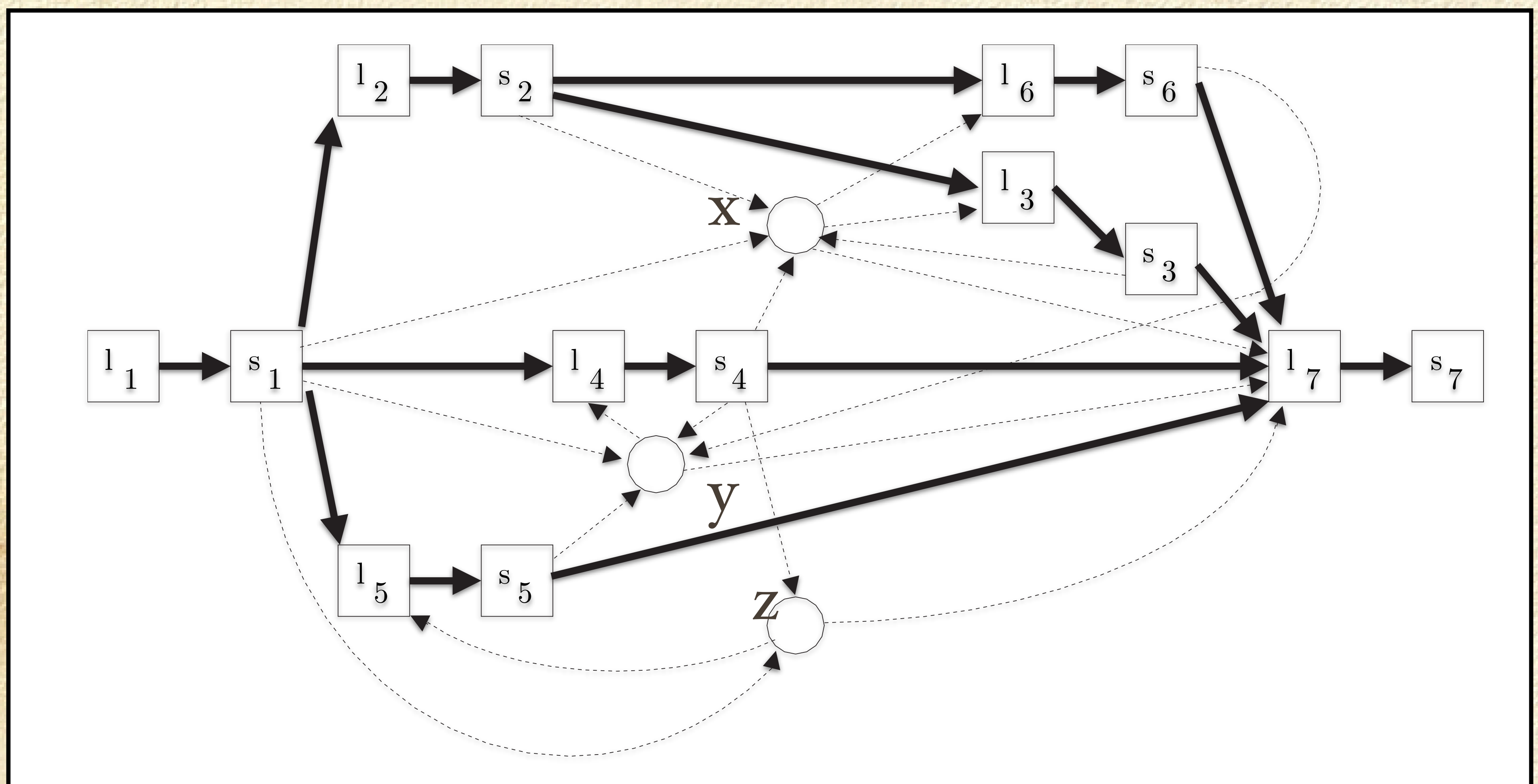


## *Ausführungsfolge:*

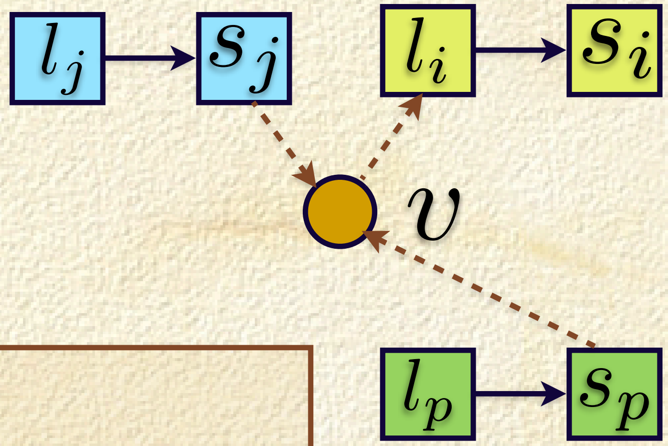
$$w = l_1\ s_1\ l_2\ s_2\ l_4\ l_3\ l_6\ s_4\ l_5\ s_3\ s_6\ s_5\ l_7\ s_7$$

$$w = \mathbf{l}_1 s_1[xyz]\ \mathbf{l}_2 s_2[x]\ \mathbf{l}_4[y]\ \mathbf{l}_3[x]\ \mathbf{l}_6[x]\ s_4[xyz]\ \mathbf{l}_5[z]\ s_3[x]\ s_6[y]\ s_5[y]\ \mathbf{l}_7[xyz]\ s_7$$

*"Wertübertragungsrelation"*



**Definition 6.16** *Es sei $AS = (A, <)$ wie in 1.30. Wir definieren für eine gegebene Ausführungsfolge $w \in F(AS)$ und $a_1, a_2 \in A$:*




$\boxed{a_1 <_w a_2,}$ *falls $a_1$ vor $a_2$ in $w$ vorkommt.*

$\boxed{l_i \text{ liest } v \text{ von } s_j \text{ in } w,}$ *falls*

a) $s_j <_w l_i$

b) $v \in aus_j \cap ein_i$ *($v$ wird von $s_j$ geschrieben und von $l_i$ gelesen)*

c) *für alle $a_p = (l_p, s_p), p \notin \{j, i\}$ mit $v \in aus_p$ gilt*

$s_p <_w s_j$ *oder* $l_i <_w s_p$ *(kein dritter Auftrag schreibt dazwischen).*

*Die Relation $W\ddot{U}(w) := \{(a_j, a_i) | \exists v \in V : l_i \text{ liest } v \text{ von } s_j \text{ in } w\}$ heißt* Werteübertragungsrelation .

$$w = l_1 s_1[xyz]\ l_2 s_2[x]\ l_4[y]\ l_3[x]\ l_6[x]\ s_4[xyz]\ l_5[z]\ s_3[x]\ s_6[y]\ s_5[y]\ l_7[xyz]\ s_7$$

*"Wertübertragungsrelation"*

**Definition 6.18** *Sei $AS = (A, <)$ ein schematisches und vollständiges Auftragssystem und $w \in F(AS)$ eine Ausführungsfolge. Die für $w$ relevanten Aufträge definiert:*

a) *Der Ausgabeauftrag ist relevant*

b) *Wenn $a_i \in A$ relevant ist und $(a_j, a_i) \in W\ddot{U}(w)$ gilt, dann ist auch $a_j$ relevant.*

c) *Nur nach a) und b) erhaltene Aufträge heißen relevant für $w$, die anderen nutzlos für $w$.*

$$w = l_1 s_1[xyz]\; l_2 s_2[x]\; l_4[y]\; l_3[x]\; l_6[x]\; s_4[xyz]\; l_5[z]\; s_3[x]\; s_6[y]\; s_5[y]\; l_7[xyz]\; s_7$$

$$\begin{tikzcd}
a_1 \arrow[r] & a_4 \arrow[r] & a_5 \arrow[dr] & \\
& & & a_7 \\
a_2 \arrow[rr] & & a_3 \arrow[ur] &
\end{tikzcd}$$

$a_6$ ? nicht relevant !

**Satz 6.20** *Zwei Ausführungsfolgen $w_1, w_2 \in F(AS)$ eines schematischen Auftragssystems $AS$ sind* genau dann *äquivalent, wenn für ihre Vervollständigung $w_1', w_2'$ gilt:*

*a) $w_1', w_2'$ haben die gleichen Mengen von relevanten Aufträgen und*

*b) für alle relevanten Aufträge $a_i, a_j$ und jedes $v \in V$ gilt:*
*$a_j$ liest $v$ von $a_i$ in $w_1'$ $\Leftrightarrow$ $a_j$ liest $v$ von $a_i$ in $w_2'$*

die Wertübertragungsrelation ist gleich

*Beispiel*

$$w = l_1 s_1[xyz]\ l_2 s_2[x]\ l_4[y]\ l_3[x]\ l_6[x]\ s_4[xyz]\ l_5[z]\ s_3[x]\ s_6[y]\ s_5[y]\ l_7[xyz]\ s_7$$

*serielle Ausführungsfolge:* $w_2 = l_1\ s_1\ l_4\ s_4\ l_2\ s_2\ l_6\ s_6\ l_5\ s_5\ l_3\ s_3\ l_7\ s_7$

$$w_2 = l_1\ s_1[xyz]\ l_4[y]\ s_4[xyz]\ l_2\ s_2[x]\ l_6[x]\ s_6[y]\ l_5[z]\ s_5[y]\ l_3[x]\ s_3[x]\ l_7[xyz]\ s_7$$

# *Beweismethode: Herbrand-Interpretation H*

$f_5^1(f_4^3(f_1^2))$ , $f_4^3(f_1^2)$

Wertemenge: $D_H$ := Menge aller Terme

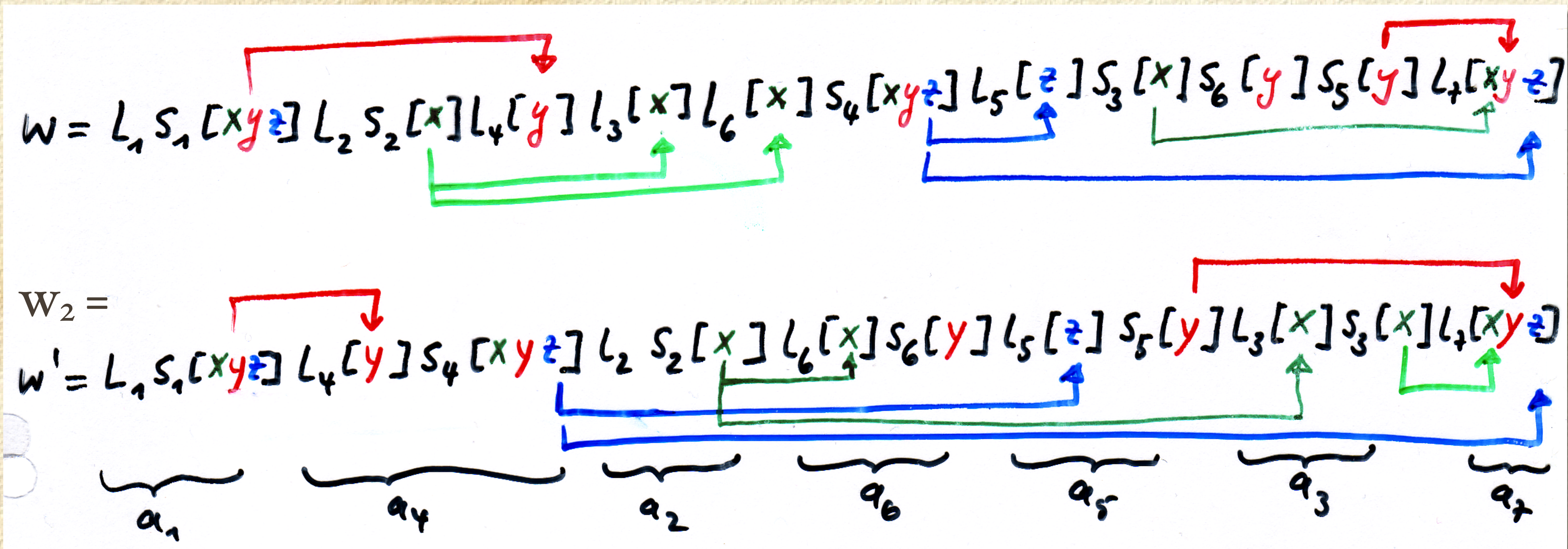
zweistelliges Funktionszeichen $f_i$

$$f_i^H : D_H \times D_H \to D_H$$

$$f_i^H(term_1, term_2) := ?$$

$$f_i(term_1, term_2)$$

$$w = l_1 s_1[xyz]\ l_2 s_2[x]\ l_4[y]\ l_3[x]\ l_6[x]\ s_4[xyz]\ l_5[z]\ s_3[x]\ s_6[y]\ s_5[y]\ l_7[xyz]\ s_7$$




$\text{res}(w, H) = \text{res}(w_2, H) =$

$$\left( f_3^1(f_2^1),\ f_5^1(f_4^3(f_1^2)),\ f_4^3(f_1^2) \right)$$

Auftrags-Nummer

$res(w,H) = res(w_2,H) =$

$$\big( f_3^1(f_2^1),\ f_5^1(f_4^3(f_1^2)),\ f_4^3(f_1^2) \big)$$

Auftrags-Nummer

$$f_3^1(lo_3) := lo_3 + 5$$

$$f_2^1 := 20$$

$$f_5^1(lo_5) := lo_5$$

$$f_4^3(lo_4) := lo_4$$

$$f_1^2 := 1$$

$$f_4^3(lo_4) := lo_4$$

$$f_1^2 := 1$$

$$(\,25,\ 1\ ,\ 1\,)$$

*daraus folgt:*

*Es ist entscheidbar, ob zwei Ausführungsfolgen* $w_1$ *und* $w_2$ *äquivalent sind.*

Zeitkomplexität ? polynomiell

*Es ist entscheidbar, ob es zu einer Ausführungsfolge* $w_1$ *eine serielle Ausführungsfolge* $w_2$ *gibt, die äquivalent zu* $w_1$ *ist.*

*Es ist entscheidbar, ob eine Ausführungsfolge* $w_1$ *serialisierbar ist.*

Zeitkomplexität ? NP - vollständig

# 6.2.3 Funktionalität

| | $A_0$ | $B_0$ | | $A$ | $B$ |
|---|---|---|---|---|---|
| *Beispiel:* | [{5,6} | {2,3,7}] | $\rightarrow$ | [{2,3} | {5,6,7}] |

(* $A = A_0 \subset \mathbb{Z}, B = B_0 \subset \mathbb{Z}, A$ und $B$ endlich und disjunkt *)

$\underline{var}\ A, B : \underline{set}\ \underline{of}\ \underline{integer},$

$\underline{var}\ max, min : \underline{integer};$

$\quad max, min := max(A), min(B);$

$\underline{do}\ max > min \rightarrow$

$\quad \underline{con}\ A := A \setminus \{max\}; A := A \cup \{min\}$

$\quad \| \ B := B \setminus \{min\}; B := B \cup \{max\}$

$\quad \underline{noc};$

$\quad max, min := max(A), min(B)$

$\underline{od}$

| A | max | B | min |
|---|---|---|---|
| {5,6} | 6 | {2,3,7} | 2 |
| {5,2} | | {6,3,7} | |
| | 5 | | 3 |
| {3,2} | | {6,5,7} | |
| | 3 | | 5 |

(* $A \cup B = A_0 \cup B_0, A \cap B = \emptyset, |A| = |A_0|, |B| = |B_0|, max(A) < min(B)$ *)

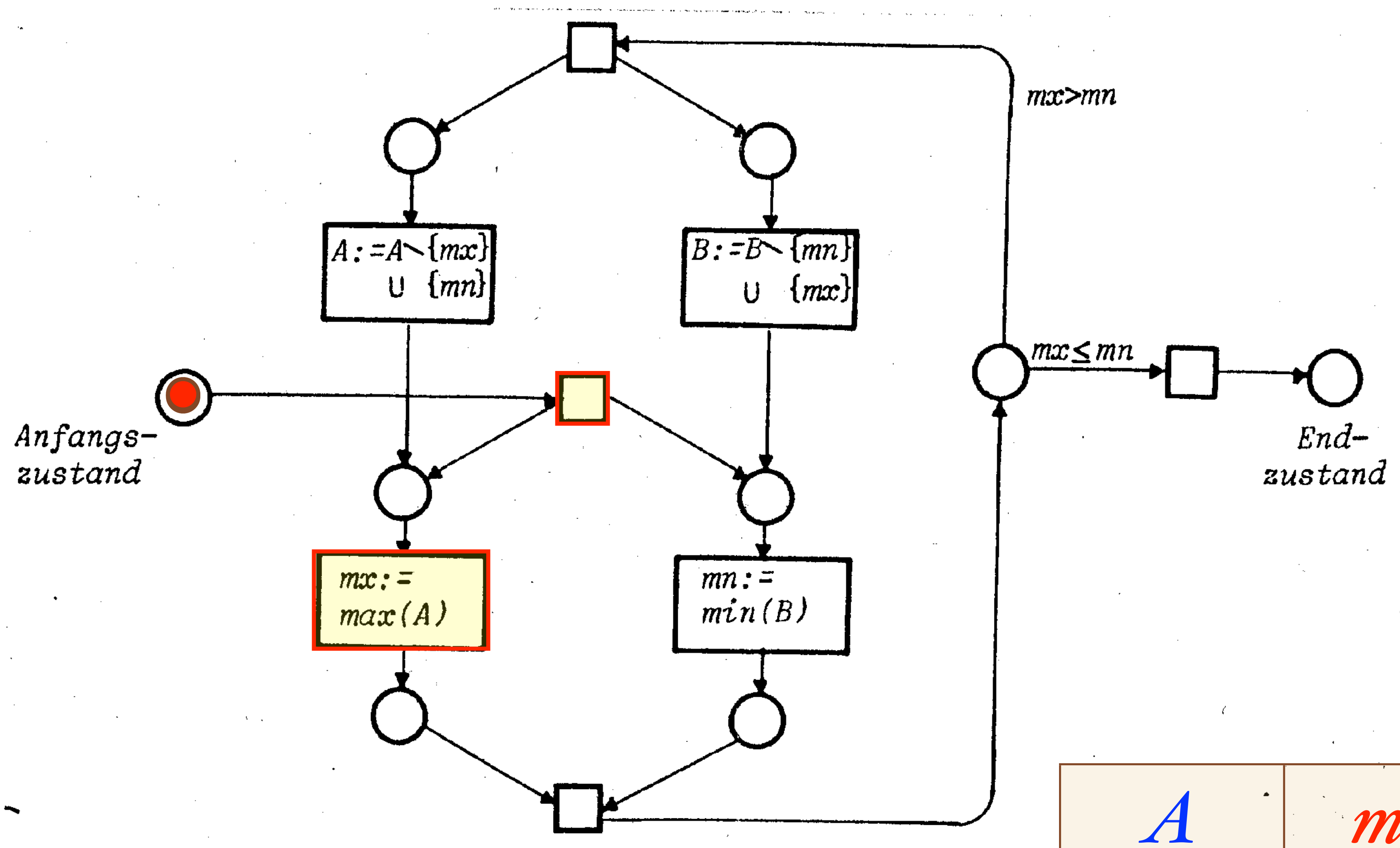



Abb. 15 Ein nichtsequentielles Programm zur Lösung des

| *A* | *mx* | *B* | *mn* |
|---|---|---|---|
| {1,6} | | {2,3,7} | |

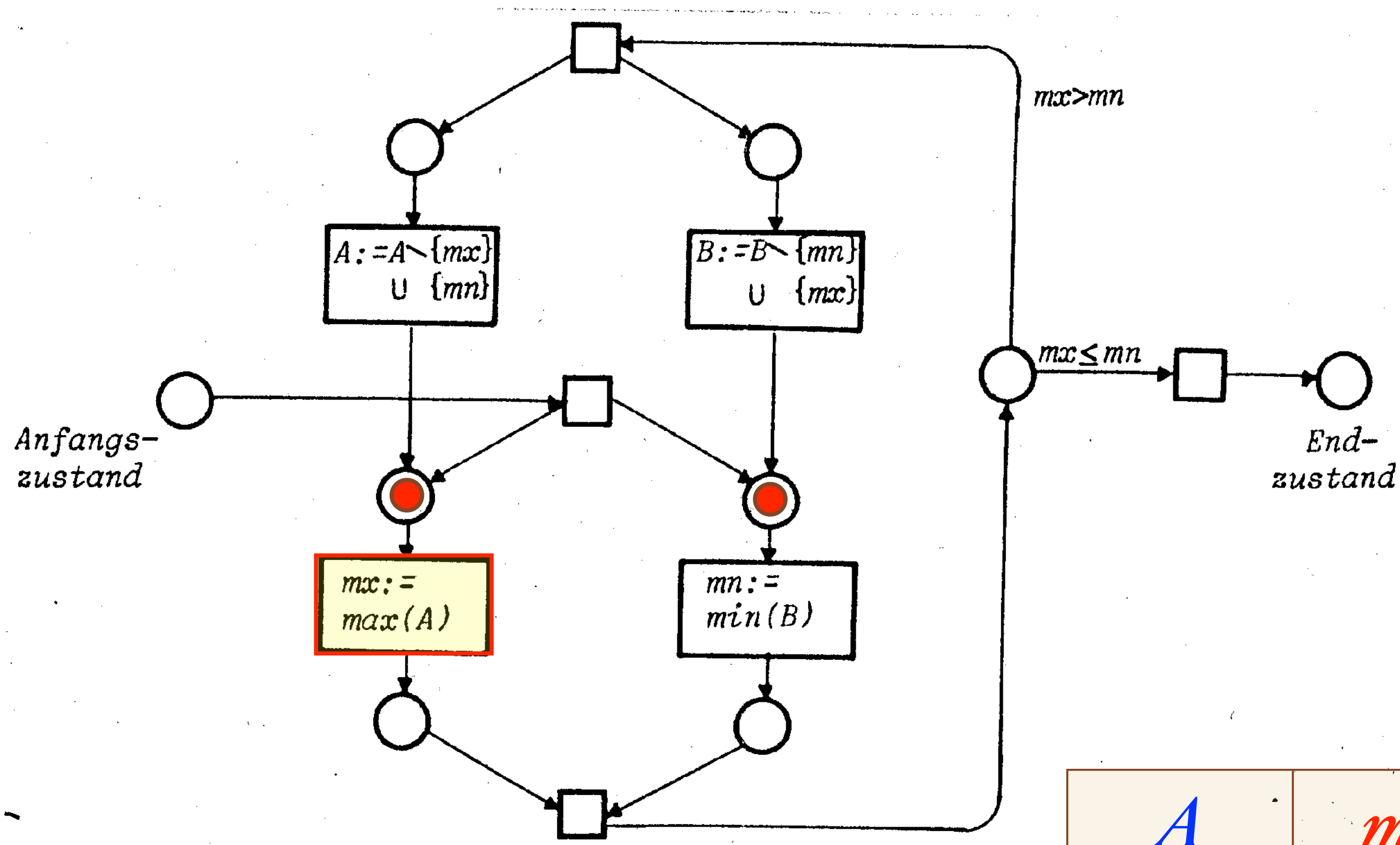



Abb. 15 Ein nichtsequentielles Programm zur Lösung des

| A | mx | B | mn |
|---|---|---|---|
| {1,6} | | {2,3,7} | |

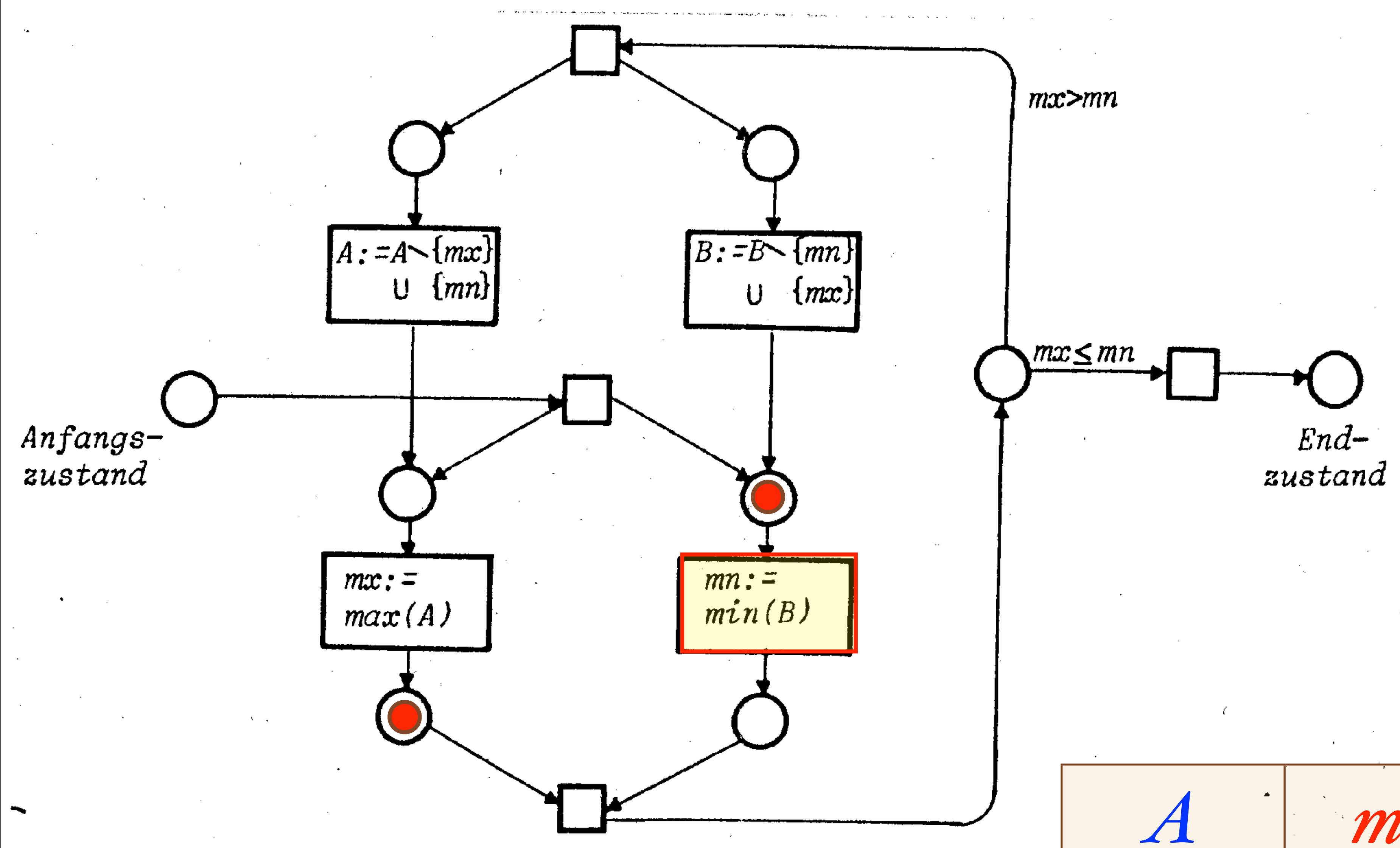



Abb. 15 Ein nichtsequentielles Programm zur Lösung des

| A | mx | B | mn |
|---|---|---|---|
| {1,6} | 6 | {2,3,7} | |

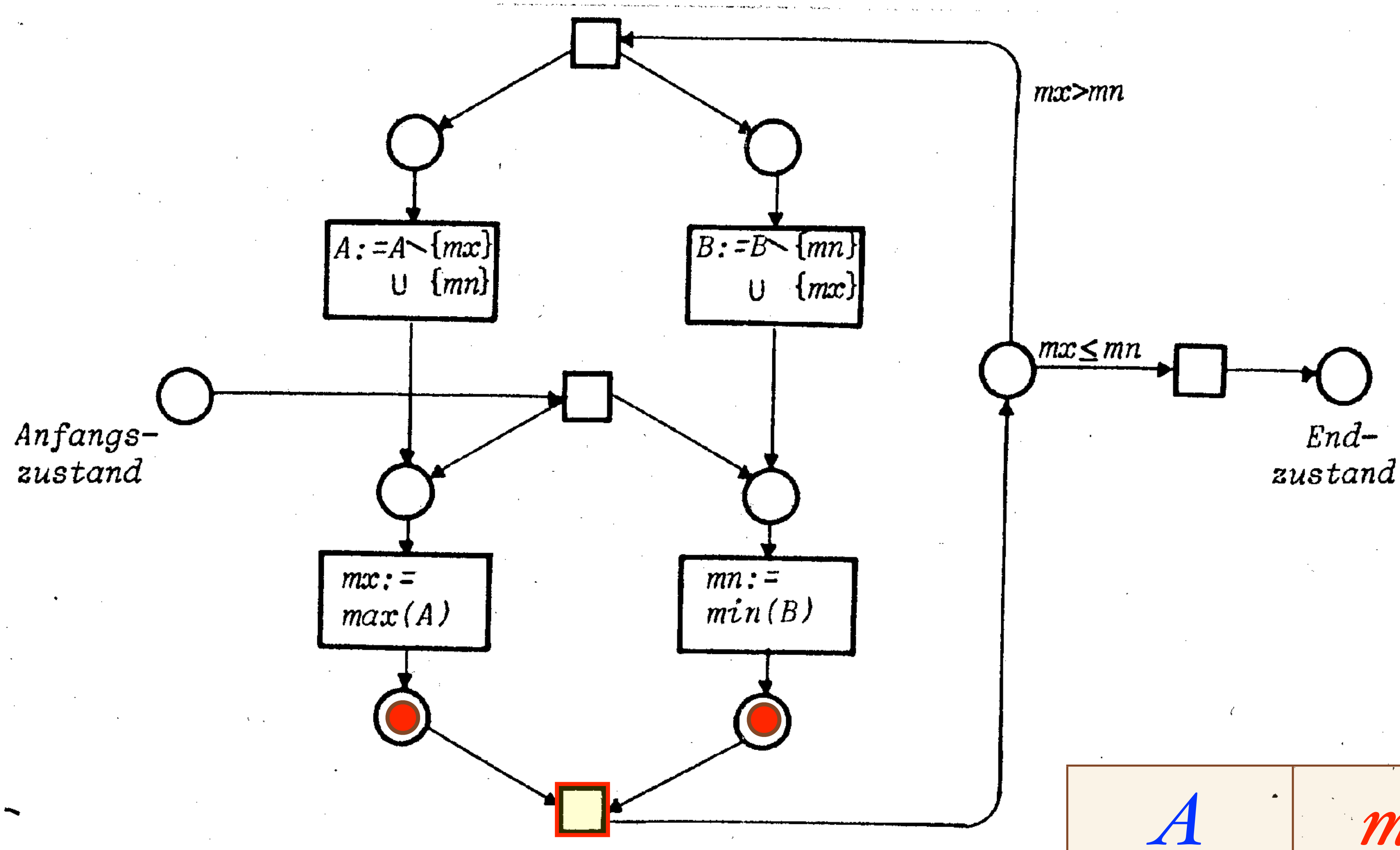



Abb. 15 Ein nichtsequentielles Programm zur Lösung des

| A | mx | B | mn |
|---|---|---|---|
| {1,6} | 6 | {2,3,7} | 2 |

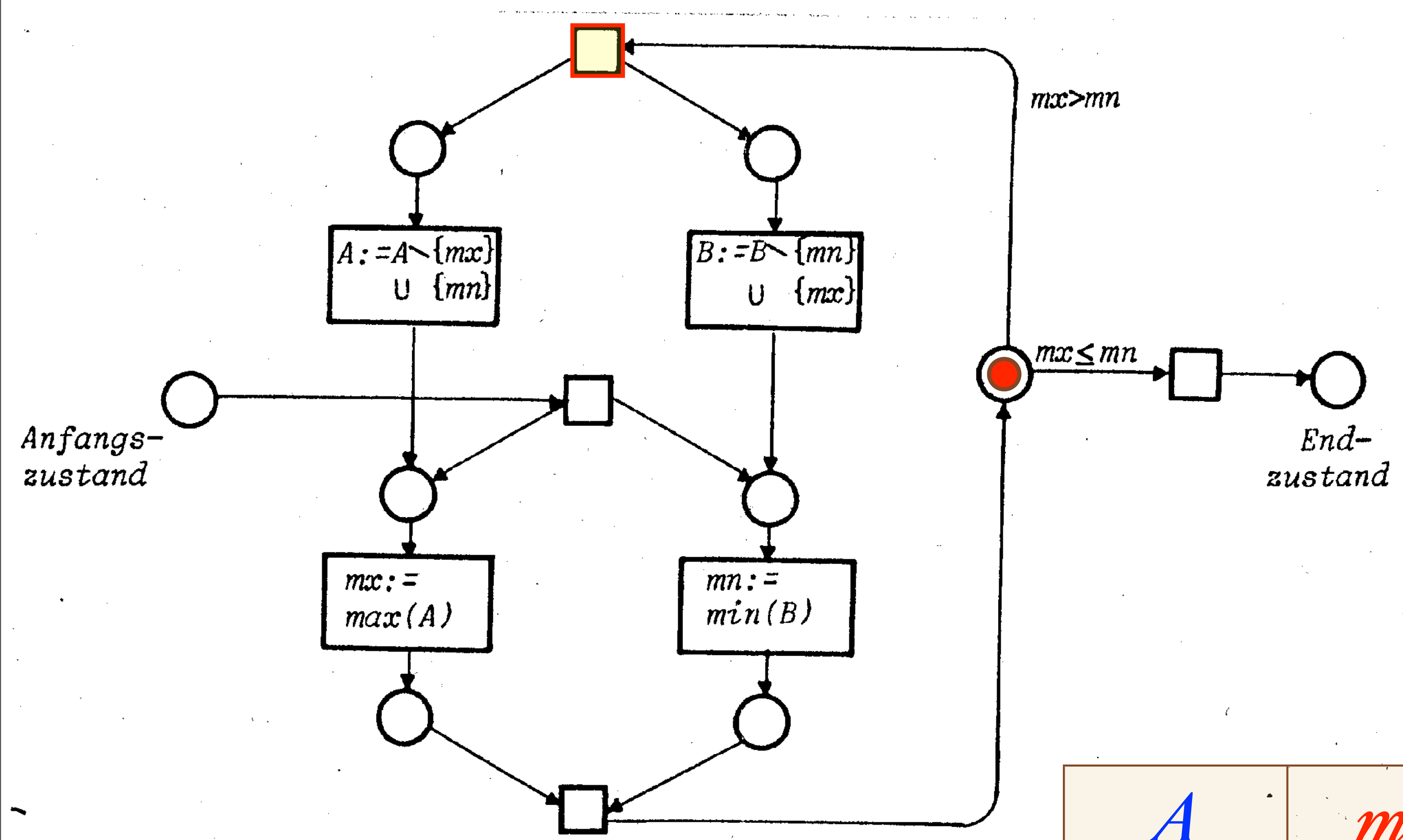



Abb. 15 Ein nichtsequentielles Programm zur Lösung des

| A | mx | B | mn |
|---|---|---|---|
| {1,6} | 6 | {2,3,7} | 2 |

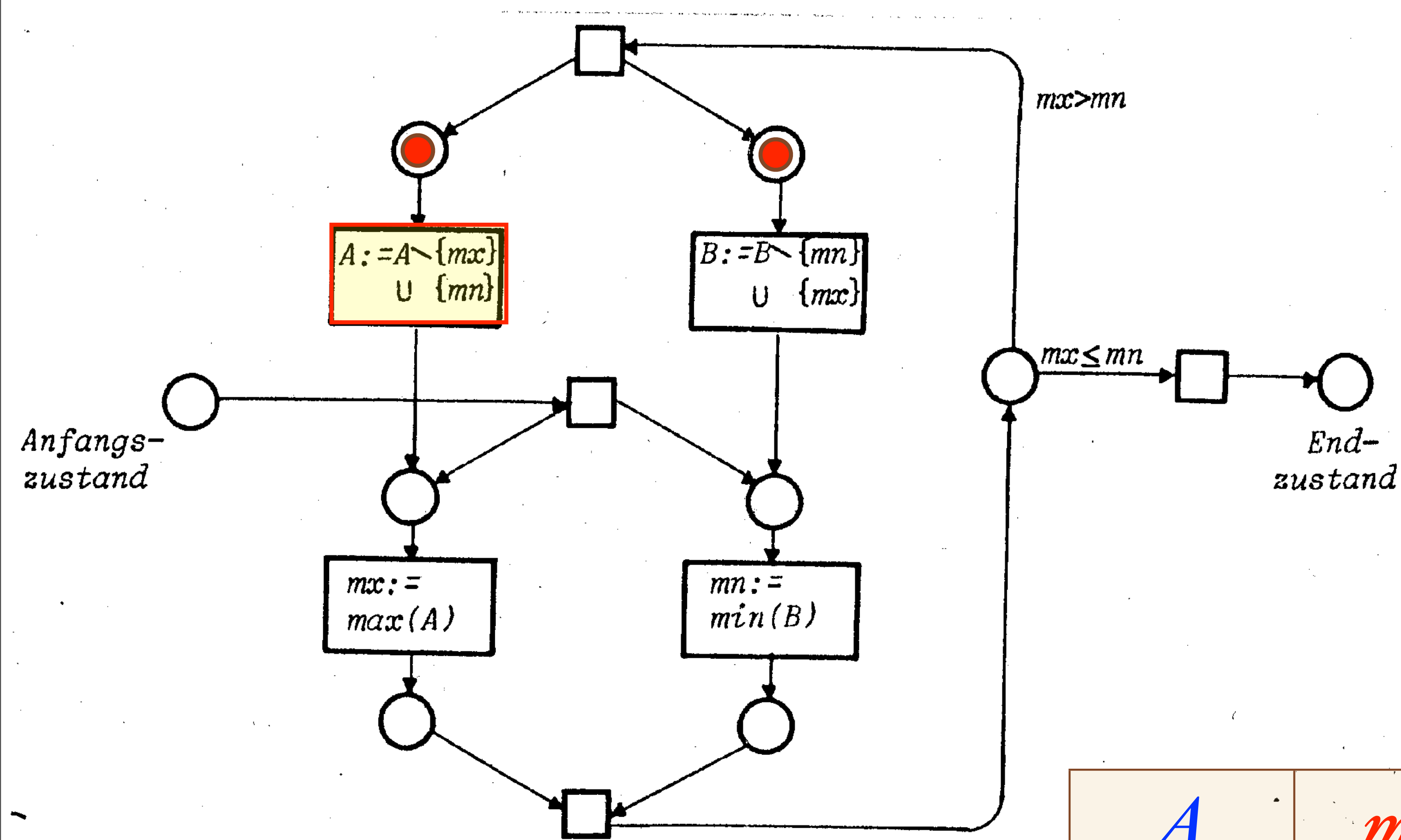



Abb. 15 Ein nichtsequentielles Programm zur Lösung des

| A | mx | B | mn |
|---|---|---|---|
| {1,6} | 6 | {2,3,7} | 2 |

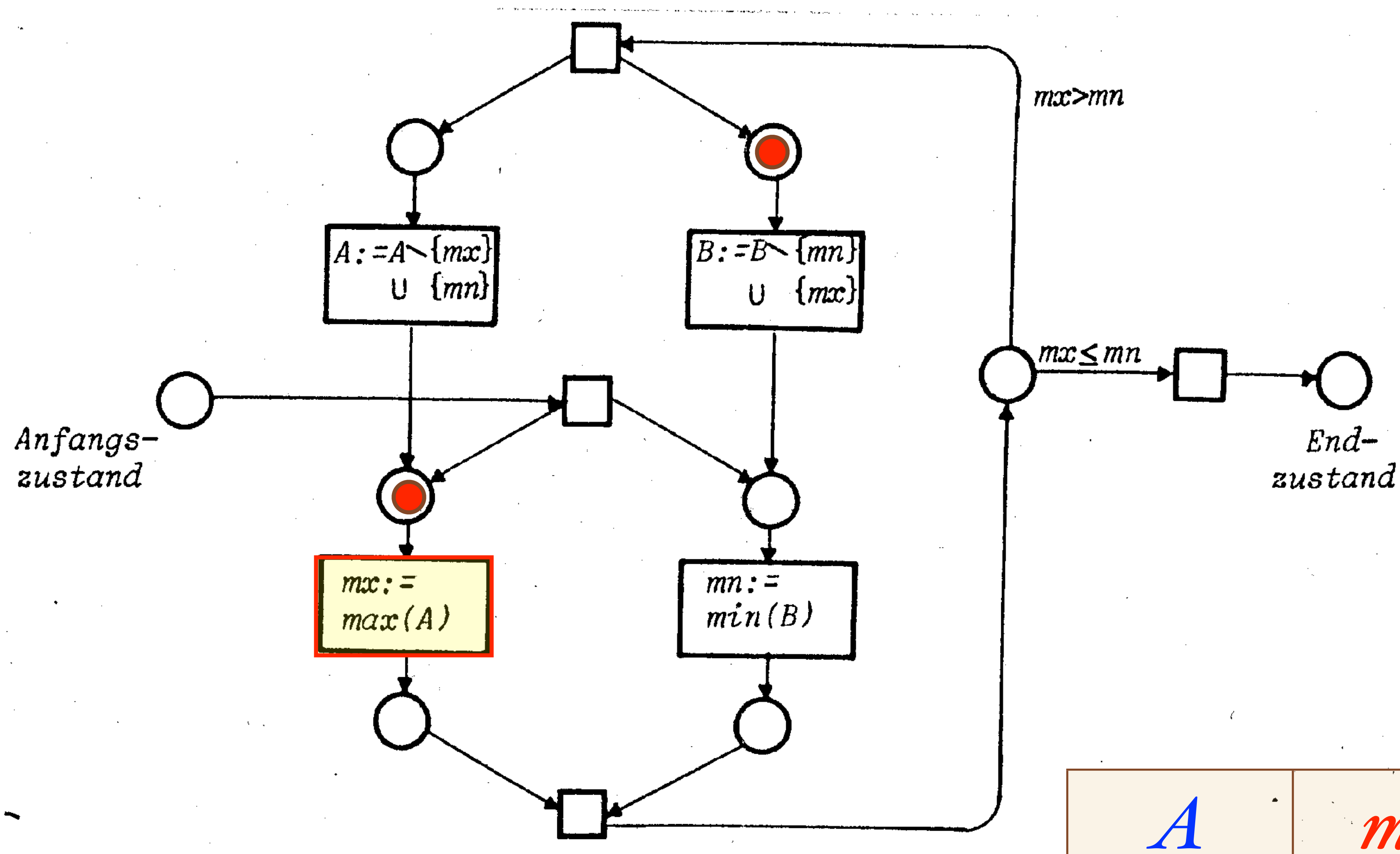



Abb. 15 Ein nichtsequentielles Programm zur Lösung des

| *A* | *mx* | *B* | *mn* |
|---|---|---|---|
| {1,6} | 6 | {2,3,7} | 2 |
| {1,2} | | | |

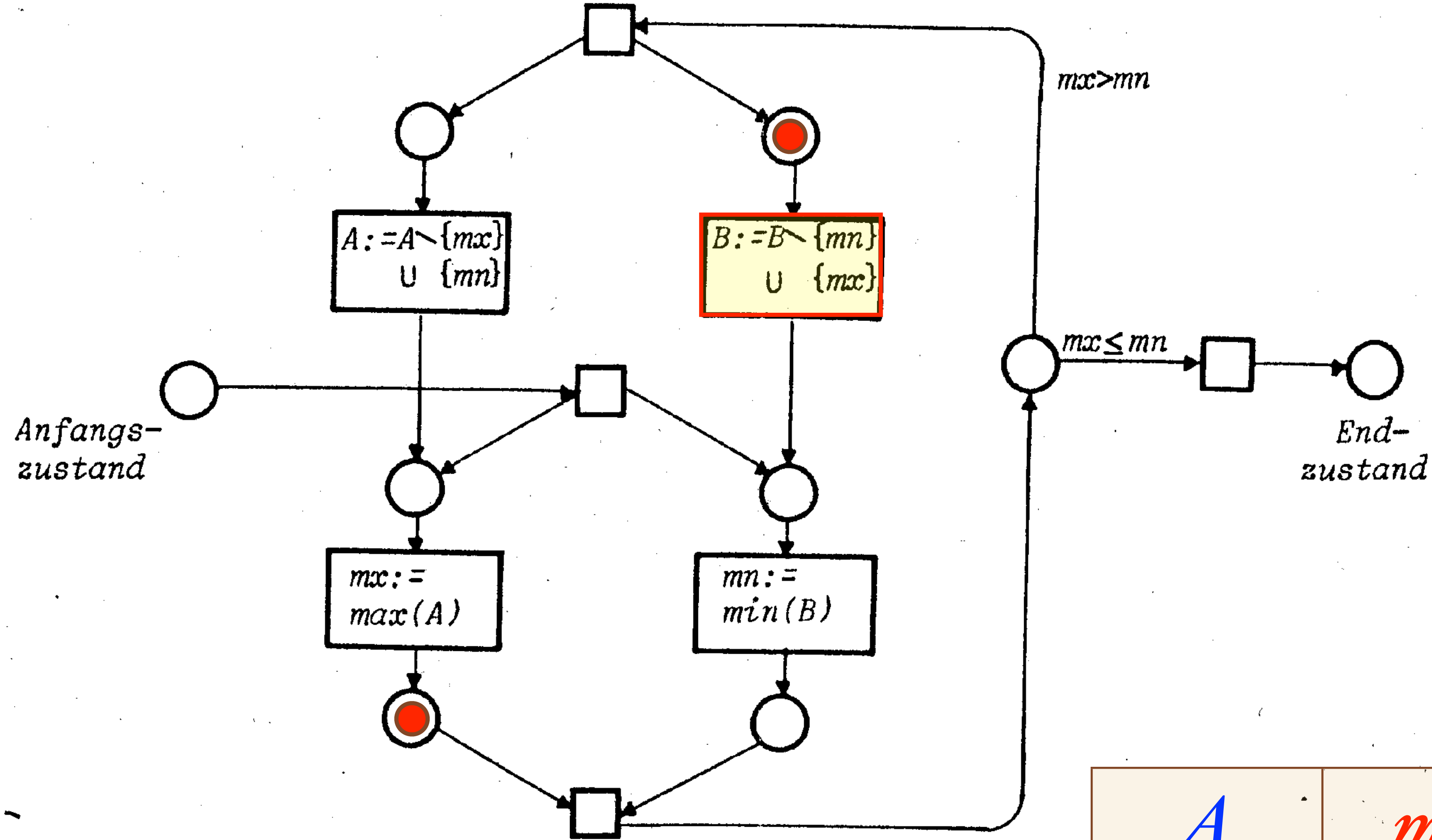



Abb. 15 Ein nichtsequentielles Programm zur Lösung des

| *A* | *mx* | *B* | *mn* |
|---|---|---|---|
| {1,6} | 6 | {2,3,7} | 2 |
| {1,2} | 2 | | |

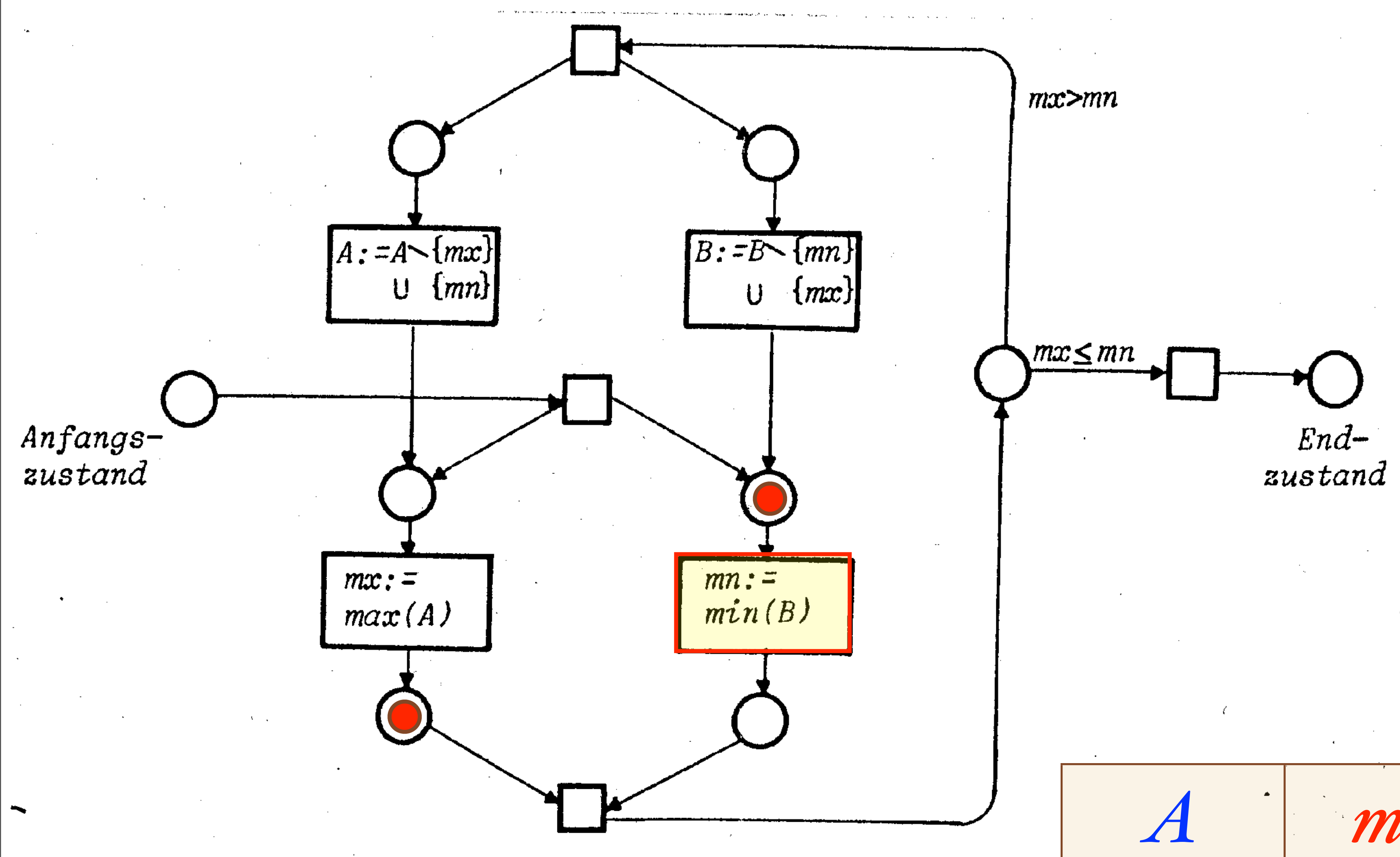



Abb. 15 Ein nichtsequentielles Programm zur Lösung des

| *A* | *mx* | *B* | *mn* |
|---|---|---|---|
| {1,6} | 6 | {2,3,7} | 2 |
| {1,2} | 2 | {2,3,7} | |

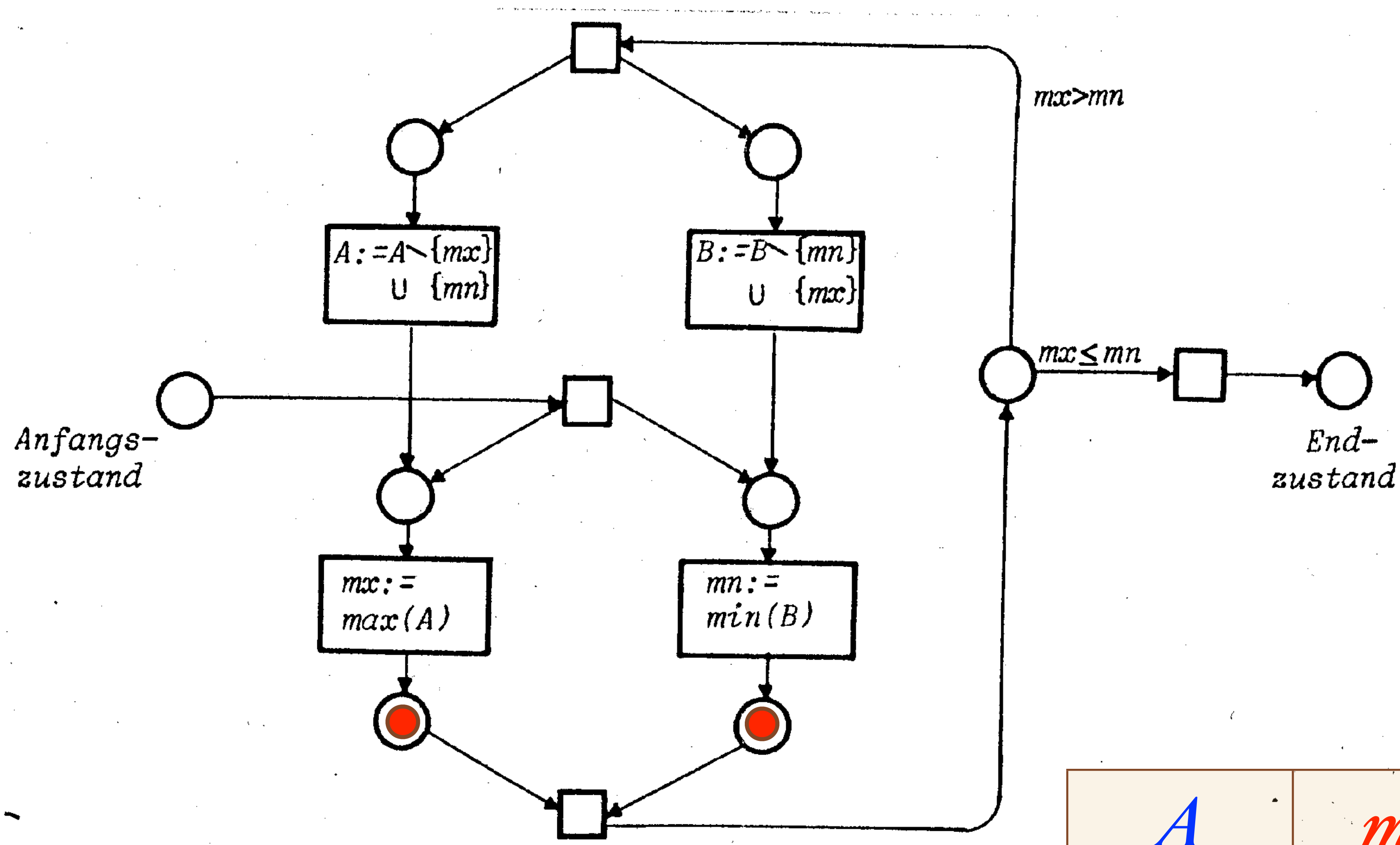



Abb. 15 Ein nichtsequentielles Programm zur Lösung des

| A | mx | B | mn |
|---|---|---|---|
| {1,6} | 6 | {2,3,7} | 2 |
| {1,2} | 2 | {2,3,7} | 2 |

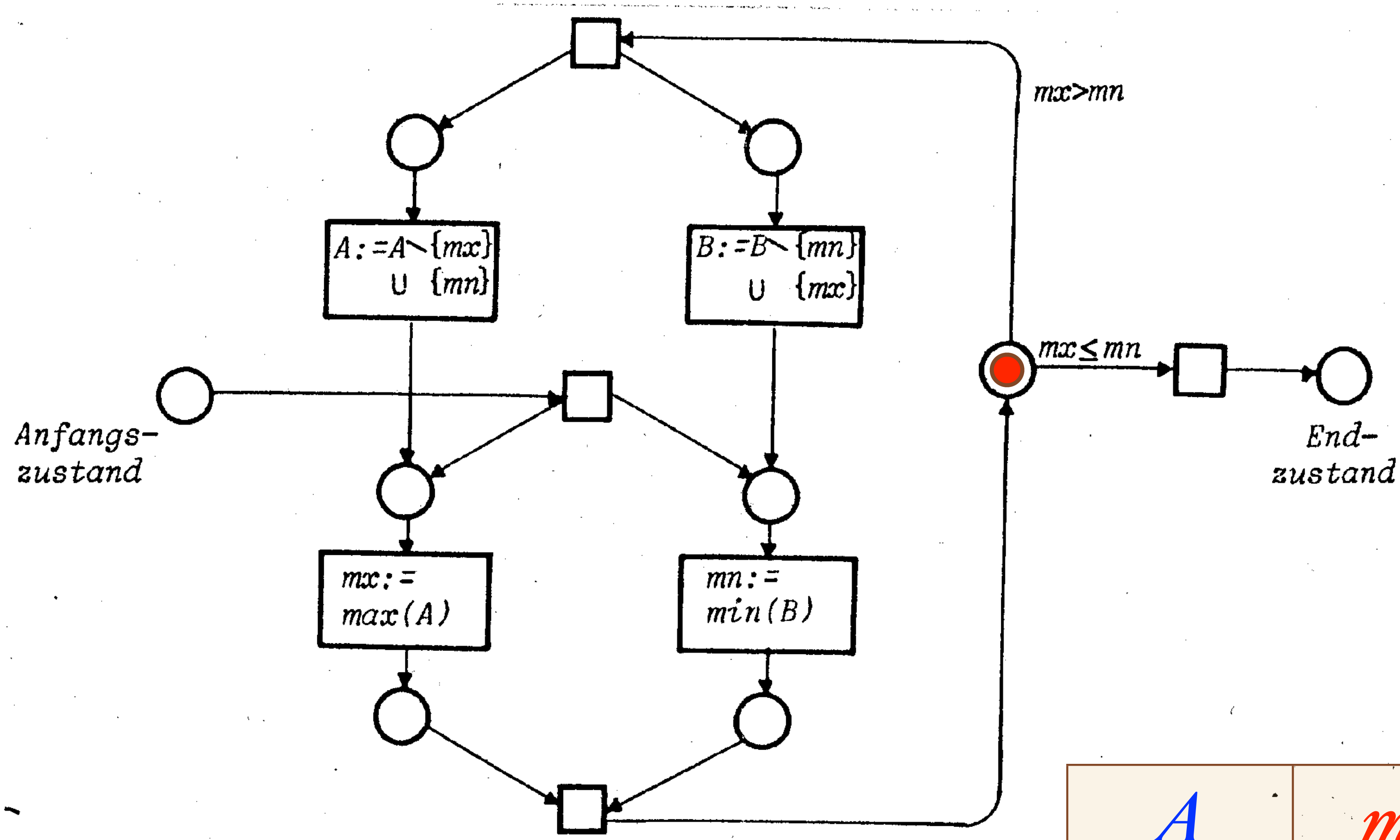



Abb. 15 Ein nichtsequentielles Programm zur Lösung des

| A | mx | B | mn |
|---|---|---|---|
| {1,6} | 6 | {2,3,7} | 2 |
| {1,2} | 2 | {2,3,7} | 2 |

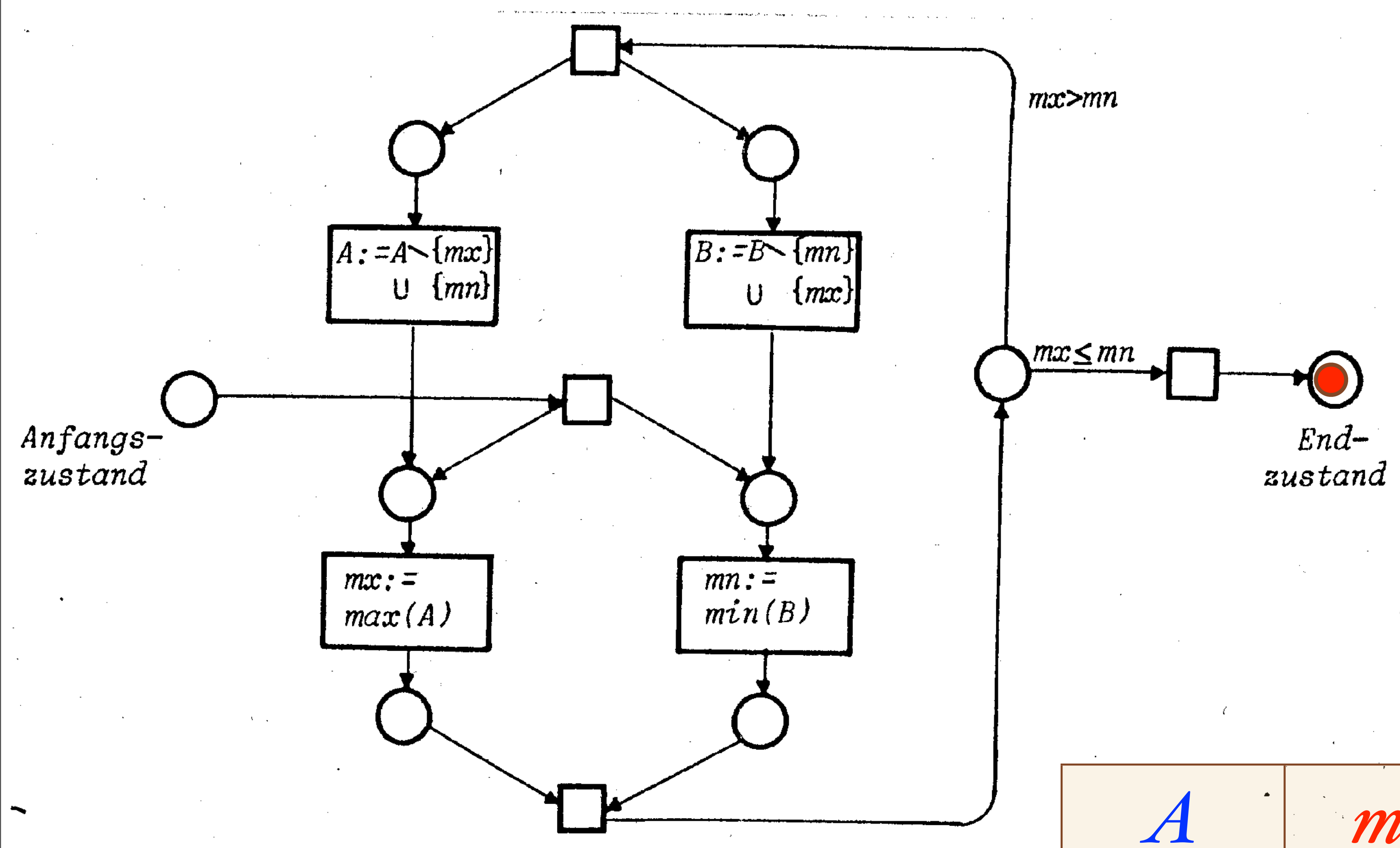



Abb. 15 Ein nichtsequentielles Programm zur Lösung des

| *A* | *mx* | *B* | *mn* |
|---|---|---|---|
| {1,6} | 6 | {2,3,7} | 2 |
| {1,2} | 2 | {2,3,7} | 2 |

*auch möglich:* {1,2} {3,6,7}

*also nicht funktional !*

“richtige” Präzedenzen setzen!

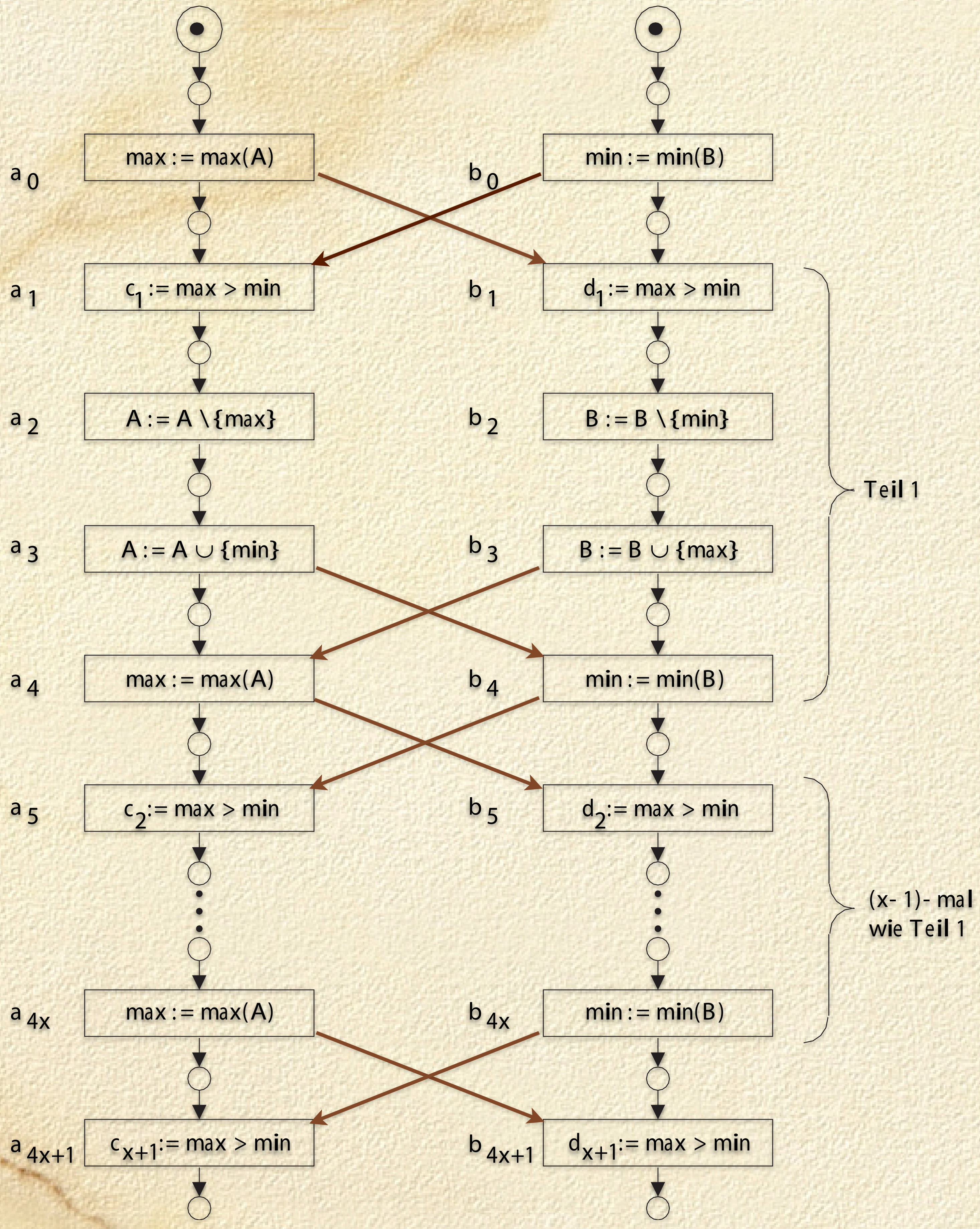



Abbildung 4.35: Interpretiertes Auftragssystem zu Beispiel 5.4

**Definition 4.41** *Sei AS ein schematisches Auftragssystem. AS heißt* funktional, *falls alle Ausführungsfolgen zueinander äquivalent sind.*

- *funktional*

**Satz 4.43** *Sei AS ein vollständiges schematisches Auftragssystem.*

alle Aufträge paarweise $\boxed{\text{störungsfrei}}$ $\Rightarrow$ AS funktional

$\Leftarrow$ falls alle Aufträge $\boxed{\text{relevant}}$

**Definition 4.42** *Sei AS ein vollständig schematisches Auftragssystem.*

(a) *Zwei Aufträge $a_i$ und $a_j$ von AS heißen* störungsfrei, *falls* $\boxed{a_i < a_j}$ *oder* $\boxed{a_j < a_i}$ *oder* $\boxed{dis(a_i, a_j)}$ *gilt, wobei*

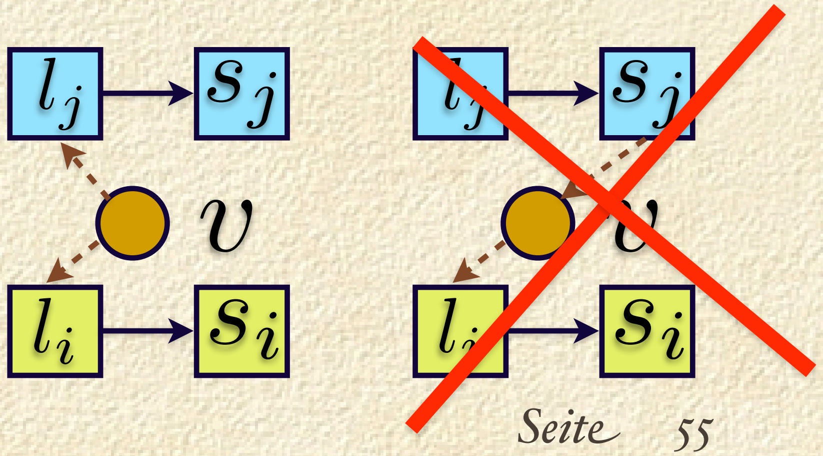
*präzedent* oder *disjunkt*

*disjunkt: Bernstein-Relation:*

$$dis(a_i, a_j) := (aus_i \cap aus_j = ein_i \cap aus_j = aus_i \cap ein_j = \emptyset)$$

*also nur erlaubt:* $$ein_i \cap ein_j \neq \emptyset$$

*"paralleles" Lesen erlaubt*



# *Beispiel*

$$
\begin{aligned}
AS = (\{\quad & l_1[v_1, v_2], & & s_1[v_3], \\
& l_2[v_1], & & s_2[v_4], \\
& l_3[v_3, v_4], & & s_3[v_1], \\
& l_4[v_3, v_4], & & s_4[v_5], \\
& l_5[v_4], & & s_5[v_2], \\
& l_6[v_5], & & s_6[v_5], \\
& l_7[v_1, v_2, v_4], & & s_7[v_4], \\
& l_8[v_1, v_3], & & s_8[v_5]\ \},\ \lessdot\ )
\end{aligned}
$$

*disjunkt* $\vee$ *präzedent*

*nicht disjunkt* $\Rightarrow$ *präzedent*

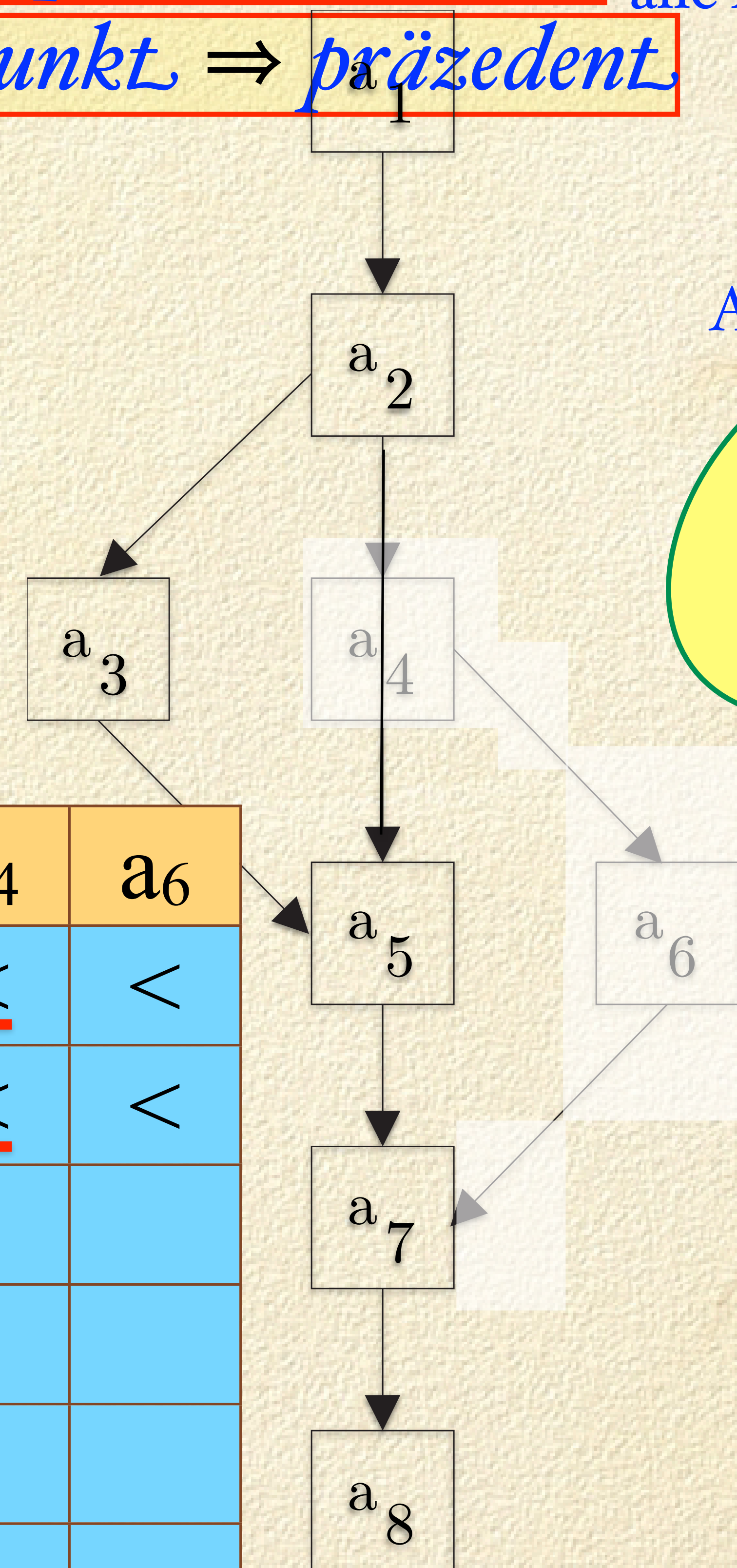
alle Aufträge paarweise

$\boxed{\text{störungsfrei}}$

$\Downarrow \quad \Uparrow$

AS funktional

falls alle Aufträge relevant



*relevant*

| | $a_1$ | $a_2$ | $a_3$ | $a_5$ | $a_7$ | $a_8$ | $a_4$ | $a_6$ |
|---|---|---|---|---|---|---|---|---|
| $a_1$ | | $<$ | $\underline{\leq}$ | $\underline{\leq}$ | $<$ | $\underline{\leq}$ | $\underline{\leq}$ | $<$ |
| $a_2$ | | | $\underline{\leq}$ | $\underline{\leq}$ | $\underline{\leq}$ | $<$ | $\underline{\leq}$ | $<$ |
| $a_3$ | | | | $<$ | $\underline{\leq}$ | $\underline{\leq}$ | | |
| $a_5$ | | | | | $\underline{\leq}$ | $<$ | | |
| $a_7$ | | | | | | $<$ | | |
| $a_8$ | | | | | | | | |
| $a_4$ | | | | $<$ | $\underline{\leq}$ | $\underline{\leq}$ | | $\underline{\leq}$ |
| $a_6$ | | | | | $<$ | $\underline{\leq}$ | | |

— *nicht disjunkt*

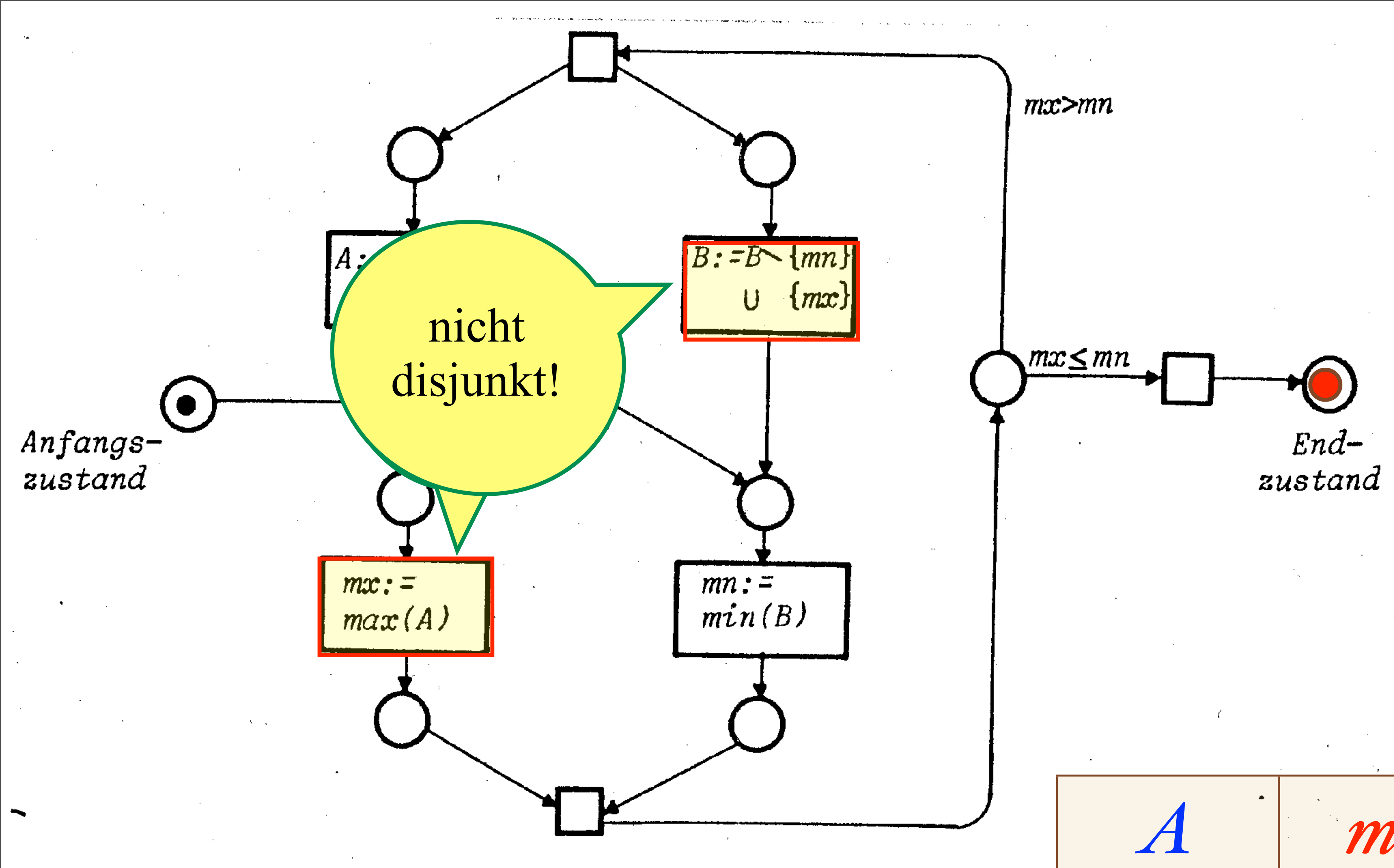



Abb. 15 Ein nichtsequentielles Programm zur Lösung des

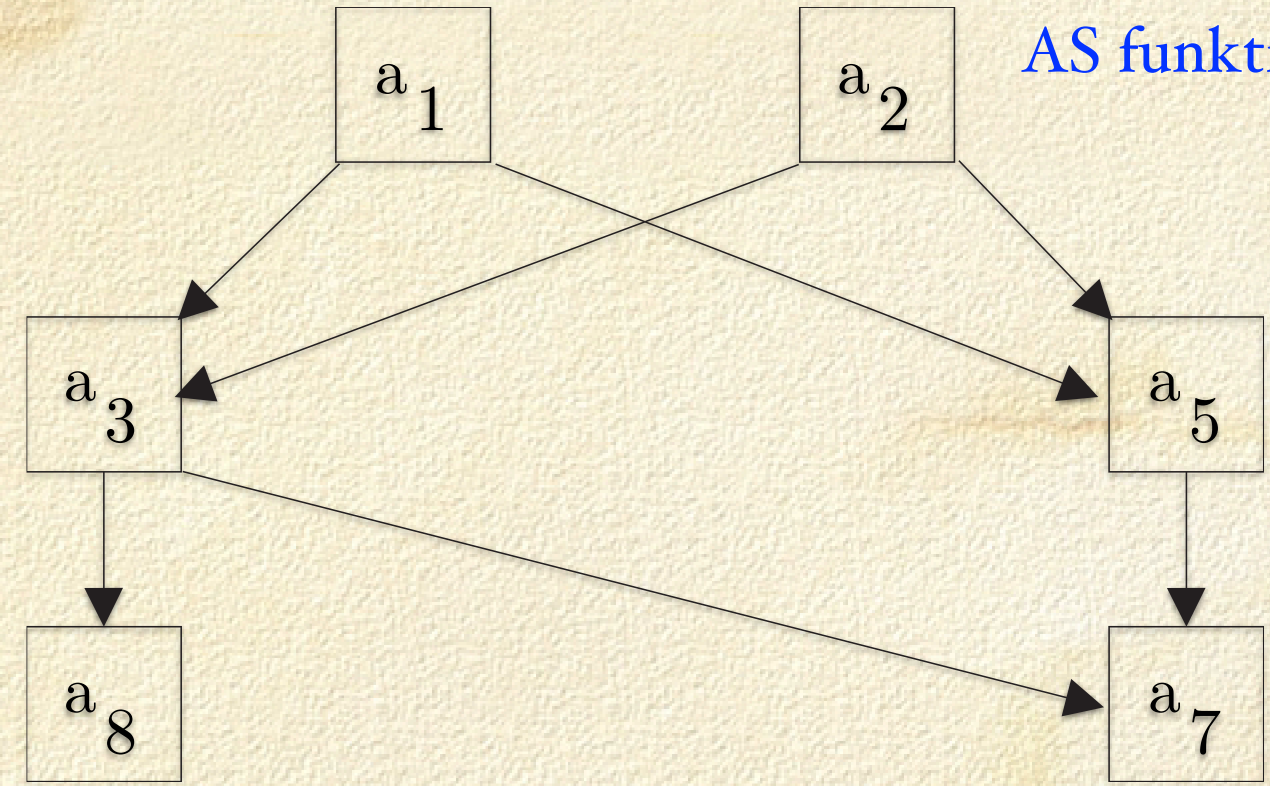
| *A* | *mx* | *B* | *mn* |
|---|---|---|---|
| {1,6} | 6 | {2,3,7} | 2 |
| {1,2} | 2 | {2,3,7} | 2 |

*auch möglich:* {1,2} {3,6,7}

*also nicht funktional*

## *Beispiel*

$$
\begin{array}{lll}
AS = (\{ & l_1[v_1, v_2], & s_1[v_3], \\
 & l_2[v_1], & s_2[v_4], \\
 & l_3[v_3, v_4], & s_3[v_1], \\
 & l_4[v_3, v_4], & s_4[v_5], \\
 & l_5[v_4], & s_5[v_2], \\
 & l_6[v_5], & s_6[v_5], \\
 & l_7[v_1, v_2, v_4], & s_7[v_4], \\
 & l_8[v_1, v_3], & s_8[v_5]\ \}, \ \lessdot\ )
\end{array}
$$




AS funktional

AS maximal nebenläufig
(für Funktionalität)

relevant

| | $a_1$ | $a_2$ | $a_3$ | $a_5$ | $a_7$ | $a_8$ | $a_4$ | $a_6$ |
|---|---|---|---|---|---|---|---|---|
| $a_1$ | | $<$ | $\leq$ | $\leq$ | $<$ | $\leq$ | $\leq$ | $<$ |
| $a_2$ | | | $\leq$ | $\leq$ | $\leq$ | $<$ | $\leq$ | $<$ |
| $a_3$ | | | | $<$ | $\leq$ | $\leq$ | | |
| $a_5$ | | | | | $\leq$ | $<$ | | |
| $a_7$ | | | | | | $<$ | | |
| $a_8$ | | | | | | | | |
| $a_4$ | | | | $<$ | $\leq$ | $\leq$ | | $\leq$ |
| $a_6$ | | | | | $<$ | $\leq$ | | |

*Welche Präzedenzen werden nicht gebraucht?*

*d.h. dadurch mehr Nebenläufigkeit*

**Definition 6.27** *Sei AS ein schematisches Auftragssystem. AS heißt* funktional, *falls alle Ausführungsfolgen zueinander äquivalent sind*.

- *funktional*

  Ergebnis

- *spurfunktional*

  alle Zuweisungen

$$w = l_1\ s_1\ l_2\ s_2\ l_4\ l_3\ l_6\ s_4\ l_5\ s_3\ s_6\ s_5\ l_7\ s_7$$

| | $d_0$ | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ | $d_6$ | $d_7$ | $d_8$ | $d_9$ | $d_{10}$ | $d_{11}$ | $d_{12}$ | $d_{13}$ | $d_{14}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $x$ | 0 | 0 | 0 | 0 | 20 | 20 | 20 | 20 | 0 | 0 | 25 | 25 | 25 | 25 | 25 |
| spur(w,y,I) $y$ | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 11 | 11 | 11 | 40 | 1 | 1 | 1 |
| $z$ | 0 | 0 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

$$spur(w, v, I) := d_{i_0}(v)d_{i_1}(v)\dots d_{i_k}(v)$$

*Zwei Ausführungsfolgen $w_1, w_2 \in F(AS)$ heißen* spuräquivalent*, falls $spur(w_1, I) = spur(w_2, I)$ für alle Interpretationen $I$ gilt.*

*AS heißt* spurfunktional *(oder determiniert, determinante), falls alle Ausführungsfolgen paarweise spuräquivalent sind.*

*spurfunktional*

| | x | y | z |
|---|---|---|---|
| | 1 | 2 | 3 |
| x := y+1 | 3 | 2 | 3 |
| z := y+2 | 3 | 2 | 4 |

| | x | y | z |
|---|---|---|---|
| | 1 | 2 | 3 |
| z := y+2 | 1 | 2 | 4 |
| x := y+1 | 3 | 2 | 4 |

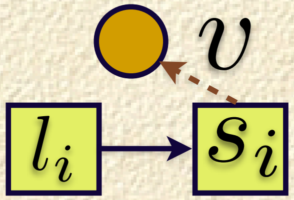
**Satz 6.29** *Sei AS ein vollständiges schematisches Auftragssystem.*

*(a) AS ist genau dann funktional, wenn alle Ausführungsfolgen die gleichen relevanten Aufträge haben und diese paarweise störungsfrei sind.*

*spurfunktional*

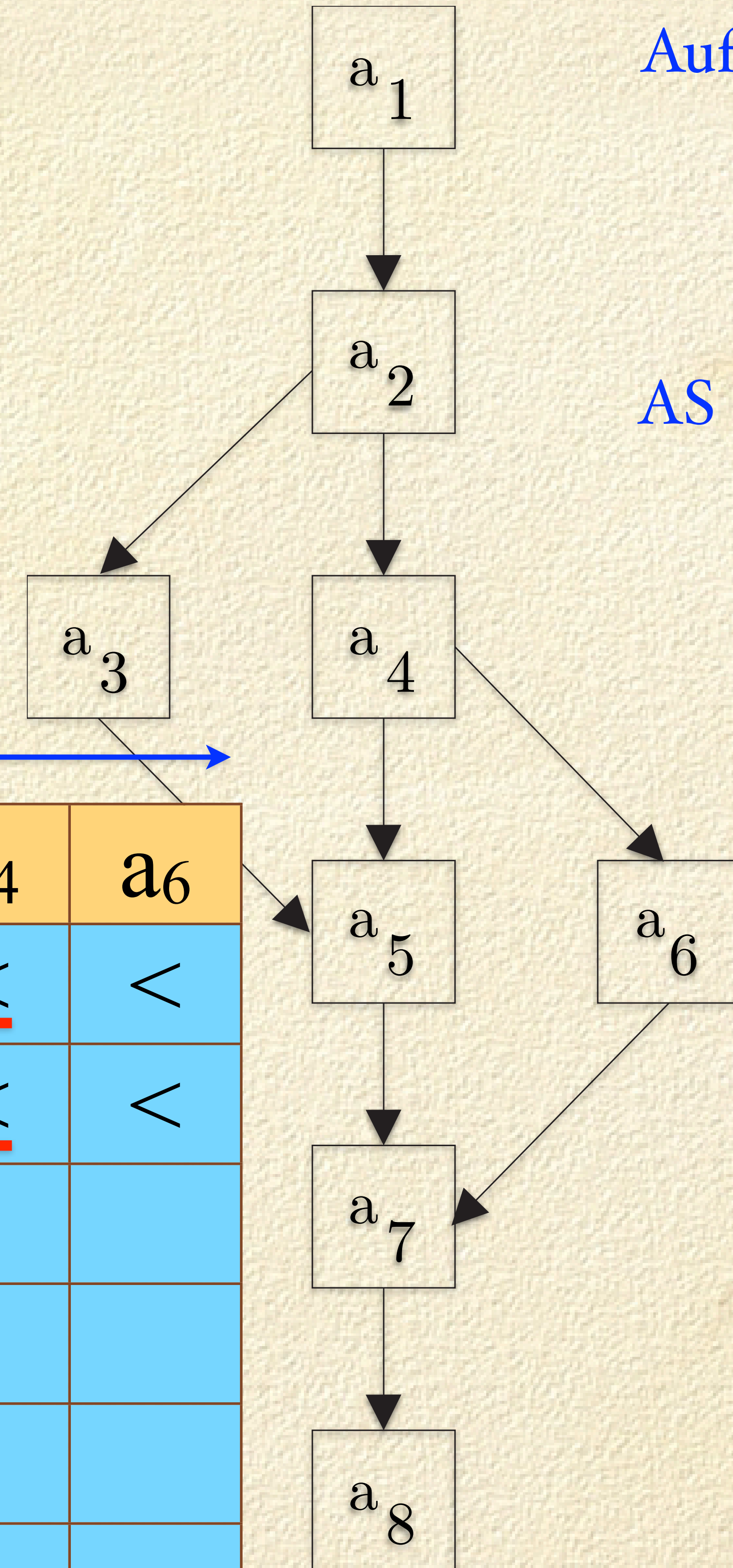
*(b) AS ist genau dann spurfunktional, wenn alle verlustfreien Aufträge von AS paarweise störungsfrei sind.*

*Ein Auftrag $a_i$ mit $aus_i \neq \emptyset$ heißt* verlustfrei. *AS heißt verlustfrei, falls alle Aufträge mit Ausnahme des Ausgabeauftrages verlustfrei*



## Beispiel

$$
\begin{aligned}
AS = (\{\ & l_1[v_1, v_2], & s_1[v_3],\\
& l_2[v_1], & s_2[v_4],\\
& l_3[v_3, v_4], & s_3[v_1],\\
& l_4[v_3, v_4], & s_4[v_5],\\
& l_5[v_4], & s_5[v_2],\\
& l_6[v_5], & s_6[v_5],\\
& l_7[v_1, v_2, v_4], & s_7[v_4],\\
& l_8[v_1, v_3], & s_8[v_5]\ \}, \lessdot\ )
\end{aligned}
$$




alle verlustfreien Aufträge paarweise störungsfrei

$\Downarrow\Uparrow$

AS spurfunktional

verlustfrei

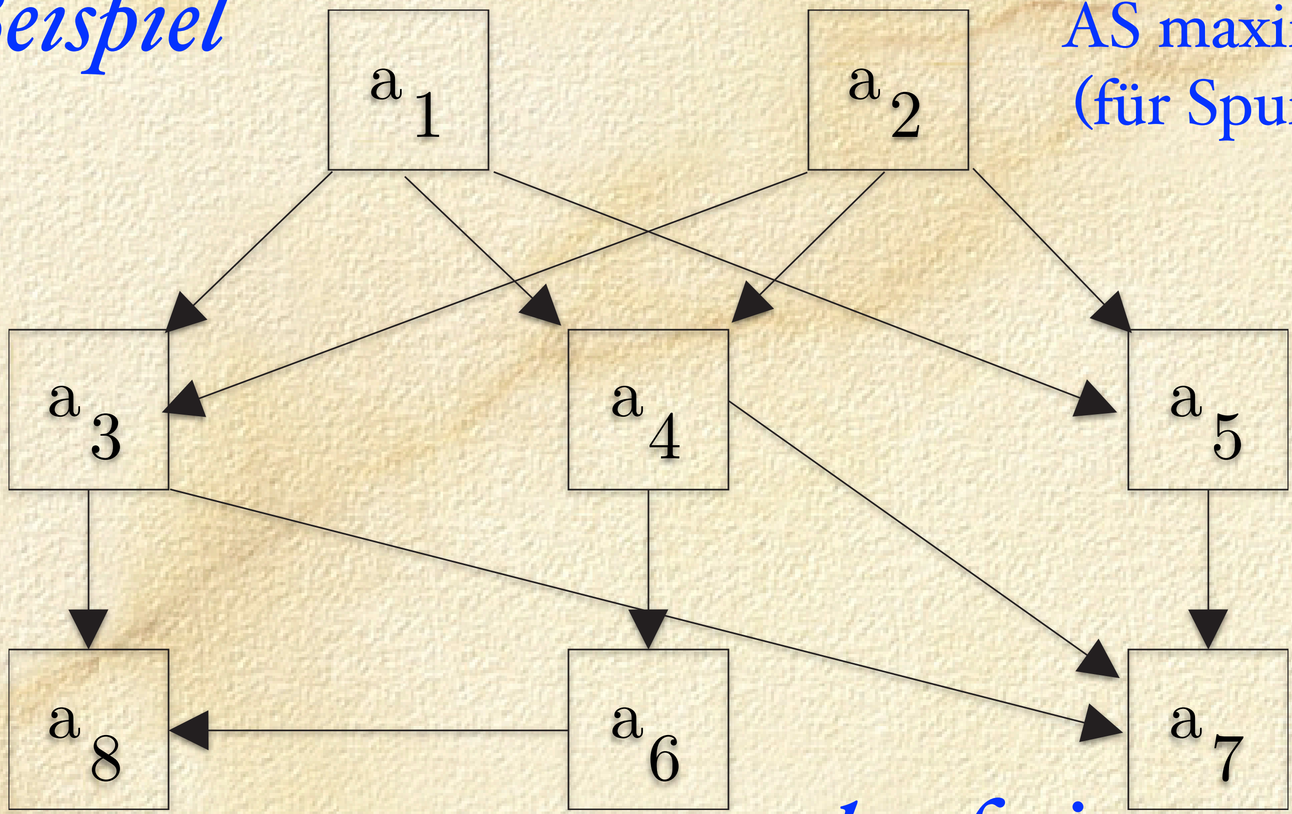
relevant

| | $a_1$ | $a_2$ | $a_3$ | $a_5$ | $a_7$ | $a_8$ | $a_4$ | $a_6$ |
|---|---|---|---|---|---|---|---|---|
| $a_1$ | | $<$ | $\underline{\leq}$ | $\underline{\leq}$ | $<$ | $\underline{\leq}$ | $\underline{\leq}$ | $<$ |
| $a_2$ | | | $\underline{\leq}$ | $\underline{\leq}$ | $\underline{\leq}$ | $<$ | $\underline{\leq}$ | $<$ |
| $a_3$ | | | | $<$ | $\underline{\leq}$ | $\underline{\leq}$ | | |
| $a_5$ | | | | | $\underline{\leq}$ | $<$ | | |
| $a_7$ | | | | | | $<$ | | |
| $a_8$ | | | | | | | | |
| $a_4$ | | | | $<$ | $\underline{\leq}$ | $\underline{\leq}$ | | $\underline{\leq}$ |
| $a_6$ | | | | | $<$ | $\underline{\leq}$ | | |

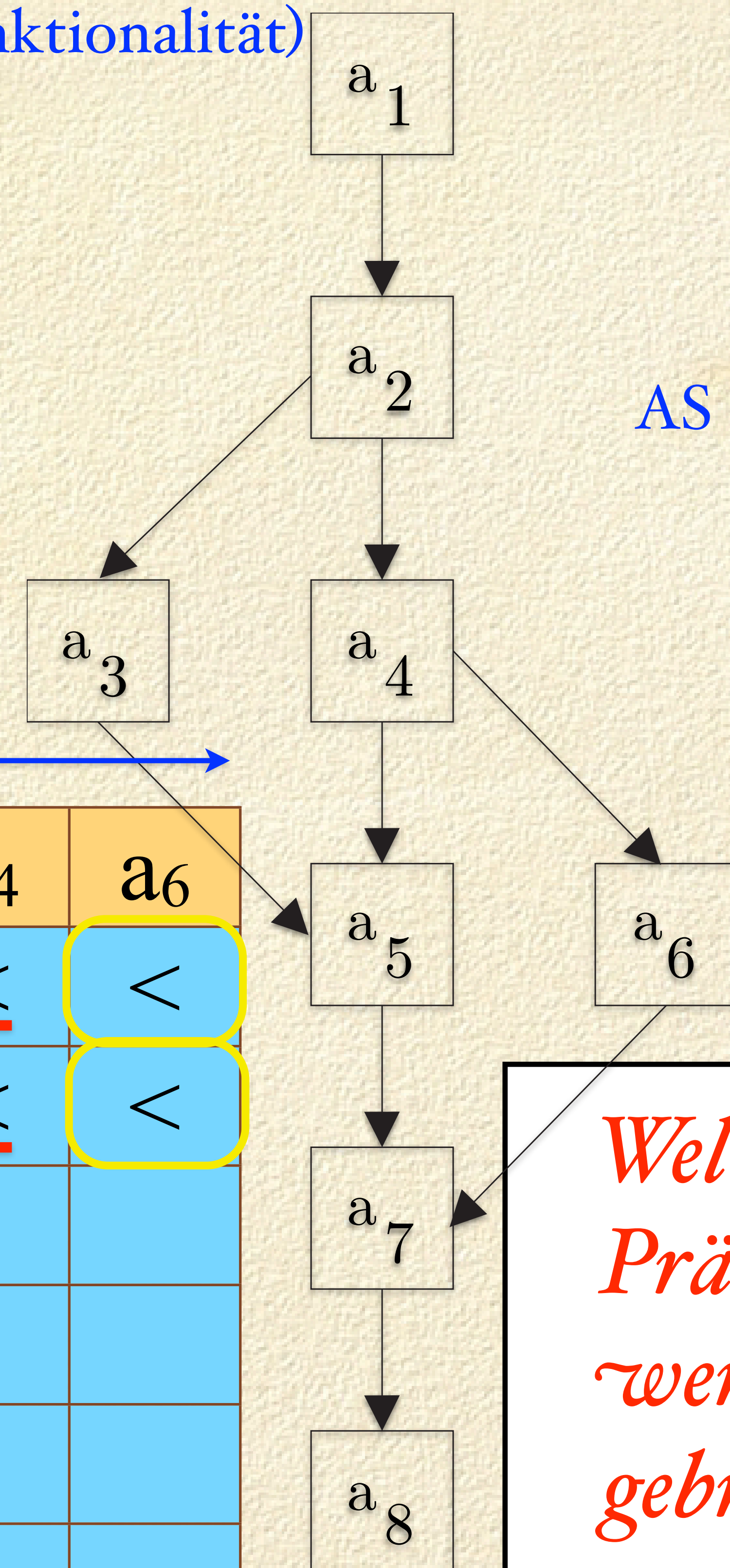
— nicht disjunkt

## Beispiel

AS maximal nebenläufig (für Spurfunktionalität)




AS spurfunktional




verlustfrei

relevant

| | $a_1$ | $a_2$ | $a_3$ | $a_5$ | $a_7$ | $a_8$ | $a_4$ | $a_6$ |
|---|---|---|---|---|---|---|---|---|
| $a_1$ | | $<$ | $\leq$ | $\leq$ | $<$ | $\leq$ | $\leq$ | $<$ |
| $a_2$ | | | $\leq$ | $\leq$ | $\leq$ | $<$ | $\leq$ | $<$ |
| $a_3$ | | | | $<$ | $\leq$ | $\leq$ | | |
| $a_5$ | | | | | $\leq$ | $<$ | | |
| $a_7$ | | | | | | $<$ | | |
| $a_8$ | | | | | | | | |
| $a_4$ | | | | $<$ | $\leq$ | $\leq$ | | $\leq$ |
| $a_6$ | | | | | $<$ | $\leq$ | | |

*Welche Präzedenzen werden nicht gebraucht?*

*d.h. mehr Nebenläufigkeit*

# Zusammenfassung