

Technische Informatik (RO)

Informationskodierung (1)

Boolesche Algebren: BMA, BAA (2,3)

Kombinatorische Schaltungen (4)

Automaten (5,6)

Datenkodierung (7,8)

Fortsetzung Teil Rechnerarchitektur,

Prof. Fengler ab 04. Dezember 2018

Bonusklausur am 20.11.2018

- **Spielregeln:**
- Bis zu 10% Bonus zum Ergebnis der Prüfung addiert
- z.B. 40 Punkte Prüfung = 100%
- \Rightarrow 10% Bonus = 4 Prüfungspunkte
- nicht da $>$ kein Nachholen $>$ kein Bonus
- $>$ kein Problem, da $>$ keine Prüfungsvoraussetzung
- Wiederholer starten neu, d.h. neue Boni, neue Prüfung
- keine Prüfungsanmeldung notwendig

- **Inhalt:**
- Addition/Subtraktion von Dualzahlen
- Kombinatorische Funktionen
- (Wertetabelle $\langle \rangle$ Ausdruck $\langle \rangle$ Schaltung)
- Boolesche Algebra (Kürzen, Erweitern)
- Kombinatorische Strukturen

Berechnung der ID-Nummer

- Die ID-Nummer wird für Leistungen genutzt, bei denen die Ergebnisse nicht mittels thoska abgerufen werden können, z. B. für Leistungen die im Rahmen von Bonuspunkten oder alternativen Prüfungsleistungen abgelegt werden.
- Die ID-Nummer besteht aus 8 Zeichen und setzt sich wie folgt zusammen: ID_letzte drei Stellen
Matrikelnummer_letzte drei Stellen
„UB-Nummer“ (thoska).
- z. B.: ID_ xx345_ xxxxxxxxxxxx321 -> ID345321

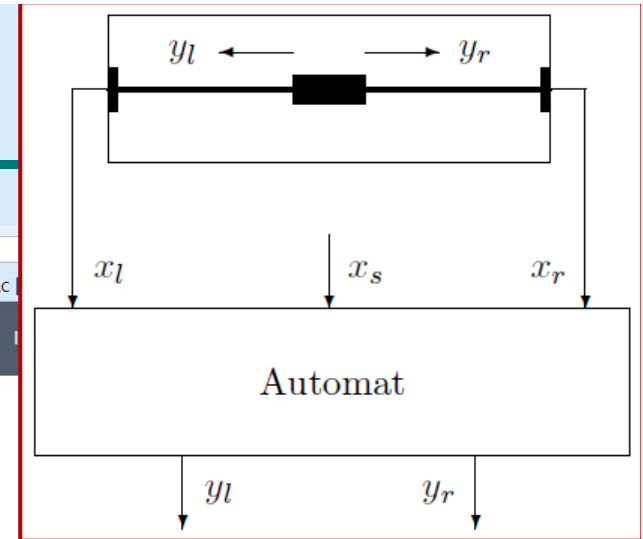
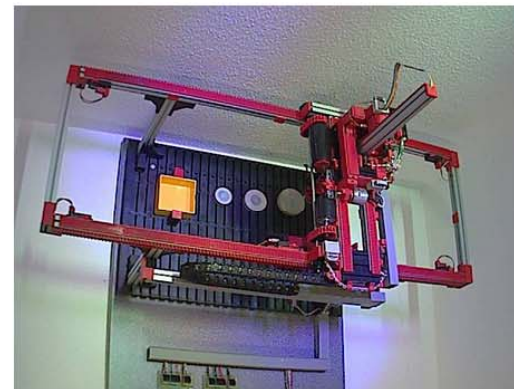
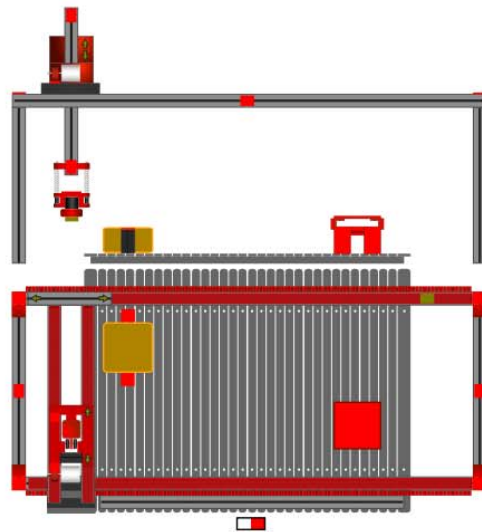
Technische Informatik (RO)

Automatentypen (Mealy/Moore)

**Synthese sequentieller Strukturen
(z- und y- Gleichungen)**

**Verifikation (Vollständigkeit
& Widerspruchsfreiheit)**

Beispiel



y01 =

y02 =

y03 =

y04 =

y05 =

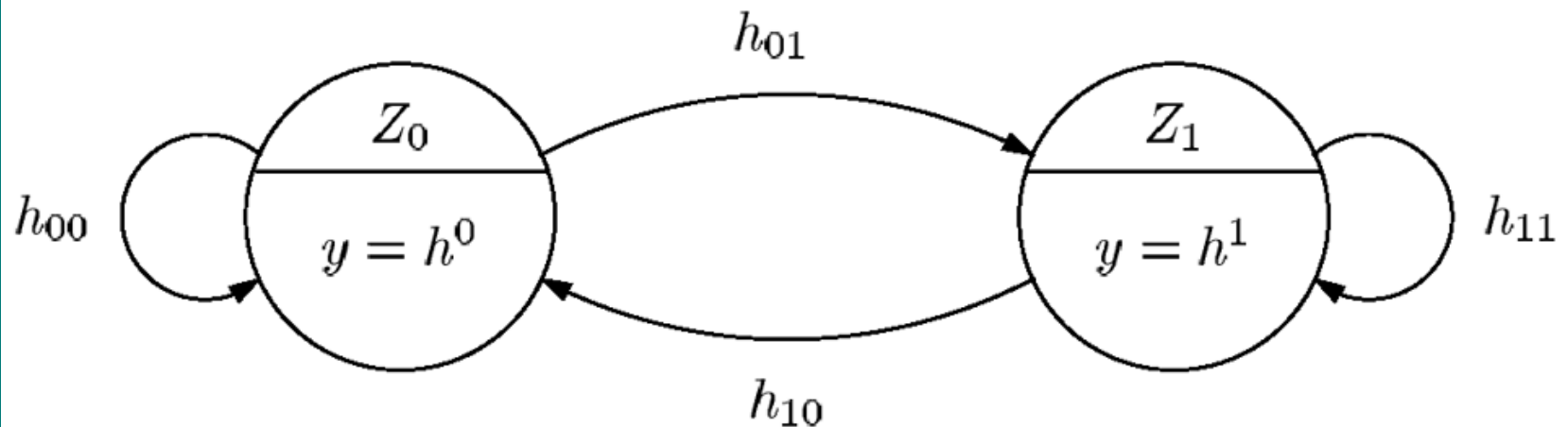
y06 =

www.goldi-labs.net

Automatengraph – aus Automatentabelle

s	i, j	δ						Situationsbeschreibung \Rightarrow Folgezustand	
		${}^aZ \times X$				Y			nZ
		z_0	x_2	x_1	x_0	y_1	y_0	z_0	
0	0,0	0	0	0	0	0	0	0	Bewegung war links, Band stopp \Rightarrow halten links
1	0,1	0	0	0	1	0	0	1	Bewegung war links, Band stopp, links angekommen \Rightarrow rechts?
2	0,2	0	0	1	0	0	0	0	Bewegung war links, Band stopp, noch rechtes Ende \Rightarrow links!
3	0,3	0	0	1	1	*	*	*	rechtes und linkes Ende \Rightarrow don't care
4	0,4	0	1	0	0	0	1	0	Bewegung war links, kein Bandende \Rightarrow links weiter
5	0,5	0	1	0	1	1	0	1	Bewegung war links, linkes Ende erreicht \Rightarrow Wechsel rechts
6	0,6	0	1	1	0	0	1	0	Bewegung war links, noch rechtes Ende \Rightarrow links weiter
7	0,7	0	1	1	1	*	*	*	rechtes und linkes Ende \Rightarrow don't care
8	1,0	1	0	0	0	0	0	1	Bewegung war rechts, Band stopp \Rightarrow halten rechts
9	1,1	1	0	0	1	0	0	1	Bewegung war rechts, Band stopp, noch linkes Ende \Rightarrow rechts!
10	1,2	1	0	1	0	0	0	0	Bewegung war rechts, Band stopp, rechts angekommen \Rightarrow links?
11	1,3	1	0	1	1	*	*	*	rechtes und linkes Ende \Rightarrow don't care
12	1,4	1	1	0	0	1	0	1	Bewegung war rechts, kein Bandende erreicht \Rightarrow rechts weiter
13	1,5	1	1	0	1	1	0	1	Bewegung war rechts, noch linkes Ende \Rightarrow rechts!
14	1,6	1	1	1	0	0	1	0	Bewegung war rechts, rechtes Ende erreicht \Rightarrow Wechsel links
15	1,7	1	1	1	1	*	*	*	rechtes und linkes Ende \Rightarrow don't care

$$\begin{aligned} X_2 &= X_s \\ X_1 &= X_r \\ X_0 &= X_l \end{aligned}$$



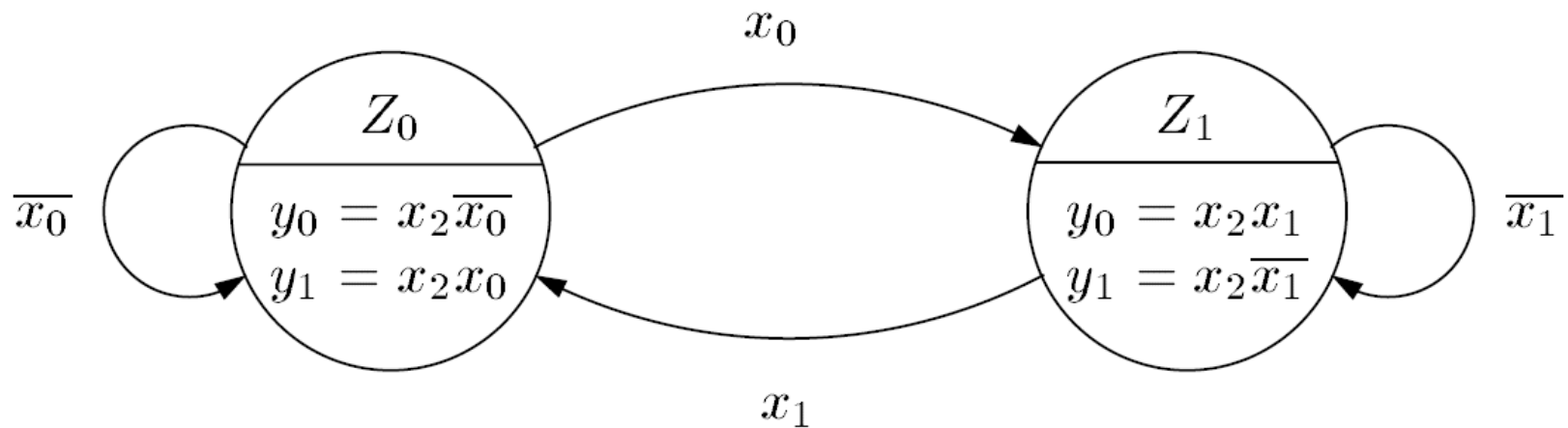
Automatengraph – aus Automatentabelle

s	i, j	δ						Situationsbeschreibung \Rightarrow Folgezustand	
		${}^aZ \times X$				Y			nZ
		z_0	x_2	x_1	x_0	y_1	y_0	z_0	
0	0,0	0	0	0	0	0	0	0	Bewegung war links, Band stopp \Rightarrow halten links
1	0,1	0	0	0	1	0	0	1	Bewegung war links, Band stopp, links angekommen \Rightarrow rechts?
2	0,2	0	0	1	0	0	0	0	Bewegung war links, Band stopp, noch rechtes Ende \Rightarrow links!
3	0,3	0	0	1	1	*	*	*	rechtes und linkes Ende \Rightarrow don't care
4	0,4	0	1	0	0	0	1	0	Bewegung war links, kein Bandende \Rightarrow links weiter
5	0,5	0	1	0	1	1	0	1	Bewegung war links, linkes Ende erreicht \Rightarrow Wechsel rechts
6	0,6	0	1	1	0	0	1	0	Bewegung war links, noch rechtes Ende \Rightarrow links weiter
7	0,7	0	1	1	1	*	*	*	rechtes und linkes Ende \Rightarrow don't care
8	1,0	1	0	0	0	0	0	1	Bewegung war rechts, Band stopp \Rightarrow halten rechts
9	1,1	1	0	0	1	0	0	1	Bewegung war rechts, Band stopp, noch linkes Ende \Rightarrow rechts!
10	1,2	1	0	1	0	0	0	0	Bewegung war rechts, Band stopp, rechts angekommen \Rightarrow links?
11	1,3	1	0	1	1	*	*	*	rechtes und linkes Ende \Rightarrow don't care
12	1,4	1	1	0	0	1	0	1	Bewegung war rechts, kein Bandende erreicht \Rightarrow rechts weiter
13	1,5	1	1	0	1	1	0	1	Bewegung war rechts, noch linkes Ende \Rightarrow rechts!
14	1,6	1	1	1	0	0	1	0	Bewegung war rechts, rechtes Ende erreicht \Rightarrow Wechsel links
15	1,7	1	1	1	1	*	*	*	rechtes und linkes Ende \Rightarrow don't care

$$X_2 = X_s$$

$$X_1 = X_r$$

$$X_0 = X_l$$



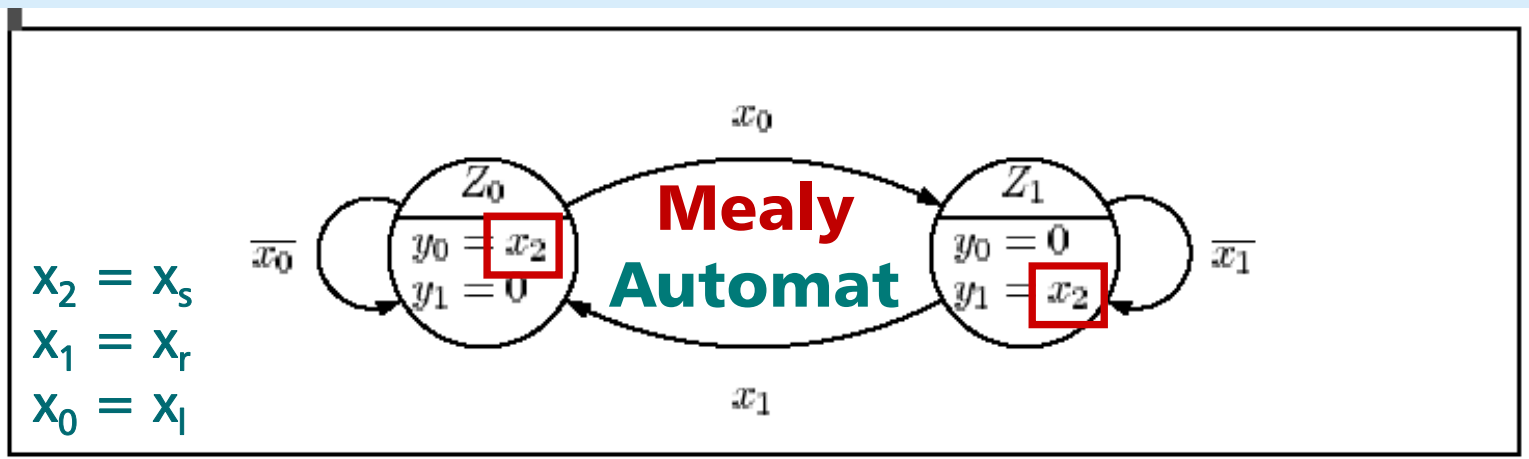
Automatentypen (Mealy/Moore)

**Synthese sequentieller Strukturen
(z- und y- Gleichungen)**

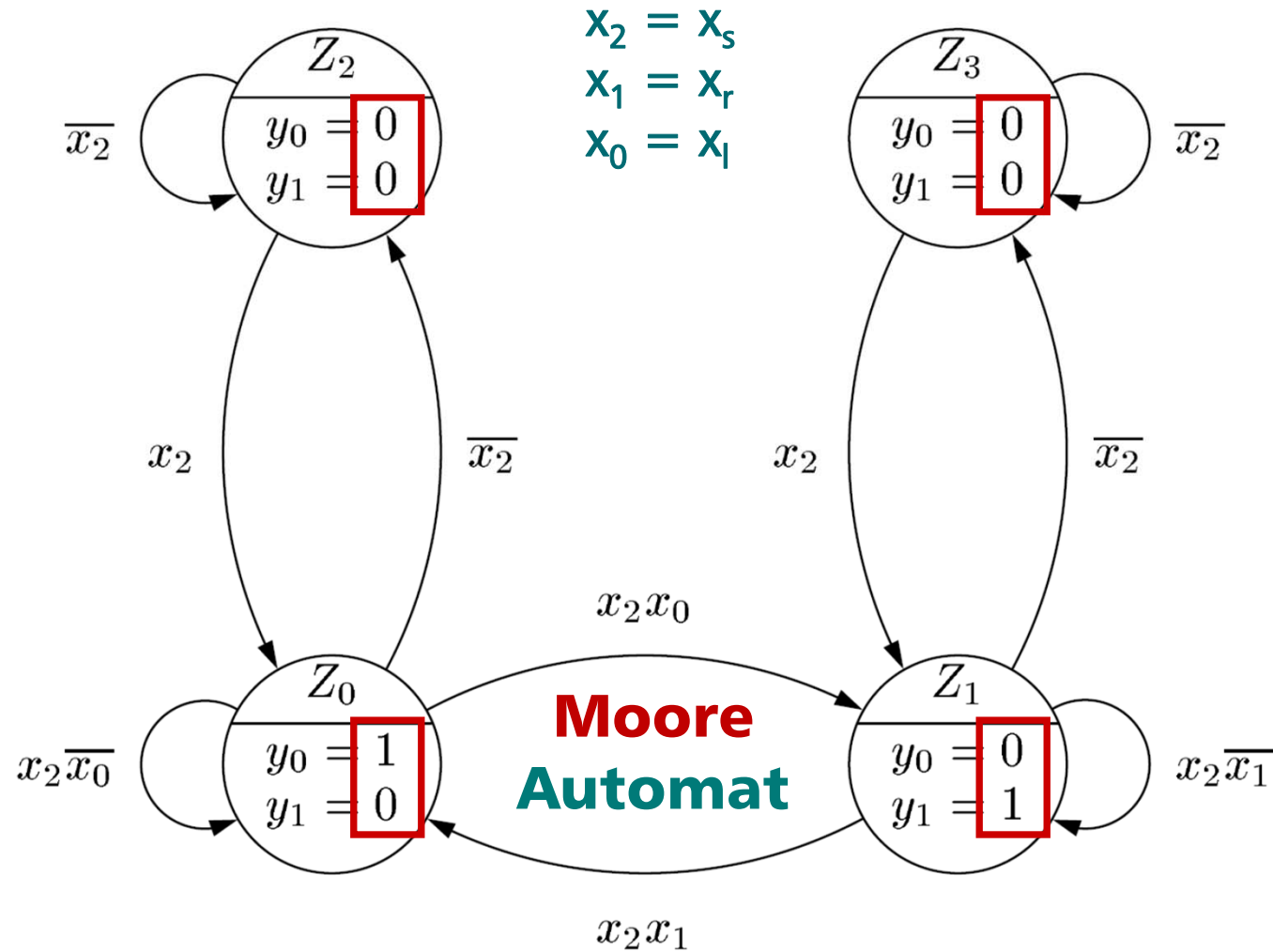
**Verifikation (Vollständigkeit
& Widerspruchsfreiheit)**

Automatengraph – Intuitiver Entwurf

Beispiel: Zustand $Z_0 \Rightarrow$ links fahren
 Zustand $Z_1 \Rightarrow$ rechts fahren
 Zustand behalten, falls Rand nicht erreicht \Rightarrow z.B. $h_{00} = \overline{x_0}$
 fahren, falls $x_2 = 1$: $y_k = x_2 \wedge \dots$



Automatengraph – Intuitiver Entwurf



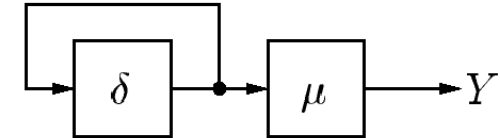
Automatentypen

Autonomer Automat

$$A = [Y, Z, \delta, \mu]$$

$$\delta : Z \Rightarrow Z$$

$$\mu : Z \Rightarrow Y$$

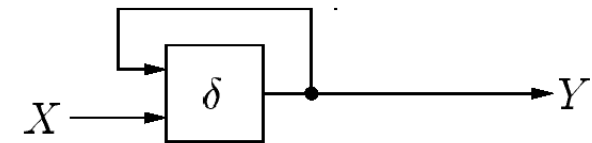


Medwedjew-Automat

$$A = [X, Y, Z, \delta]$$

$$\delta : Z \times X \Rightarrow Z$$

$$Y = Z$$

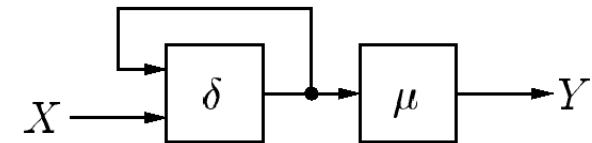


Moore-Automat

$$A = [X, Y, Z, \delta, \mu]$$

$$\delta : Z \times X \Rightarrow Z$$

$$\mu : Z \Rightarrow Y$$

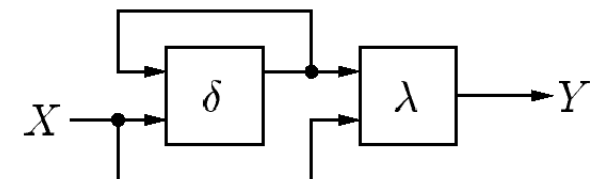


Mealy-Automat

$$A = [X, Y, Z, \delta, \lambda]$$

$$\delta : Z \times X \Rightarrow Z$$

$$\lambda : Z \times X \Rightarrow Y$$



Technische Informatik (RO)

Automatentypen (Mealy/Moore)

**Synthese sequentieller Strukturen
(z- und y- Gleichungen)**

**Verifikation (Vollständigkeit
& Widerspruchsfreiheit)**

Zustandsüberföhrungsfunktion:

Zur DNF-Realisierung **1-Belegungen**
der **z-Variablenwerte** in den
Zustandskodierungen suchen und
Bedingungen notieren:

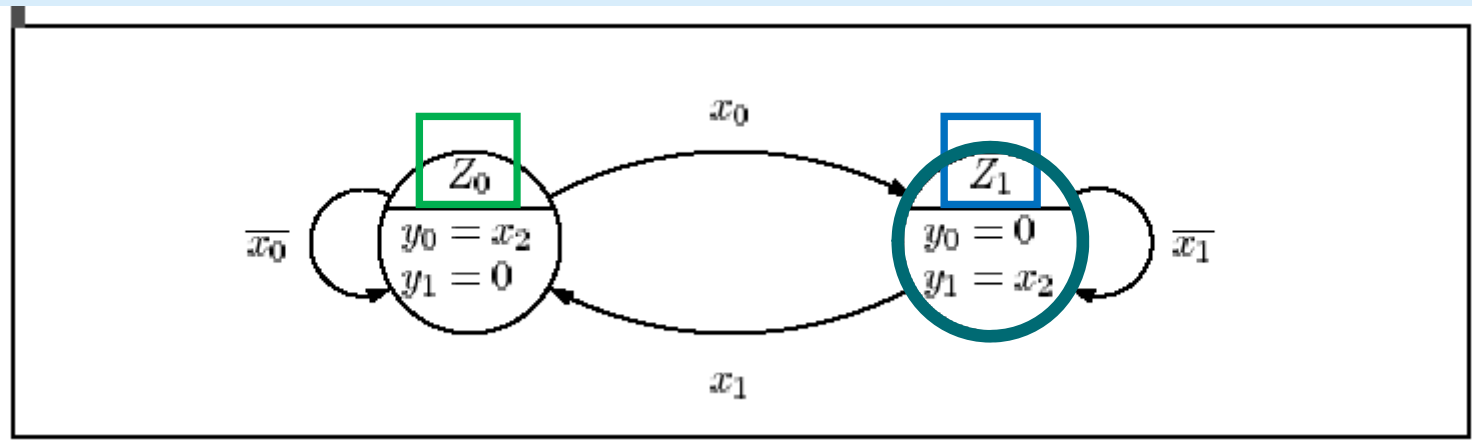
z.B. **1 = Belegung von z_0 in Z_1**

z-Gleichungen

Zustandsüberföhrungsfunktion:

$$Z_s := \bigvee_{j \in M_s} \left(\bigvee_{i=0}^{2^P-1} k_i(z) \wedge h_{ij}(x) \right)$$

$$z_0 := \bar{z}_0 \wedge x_0 \vee z_0 \wedge \bar{x}_1$$



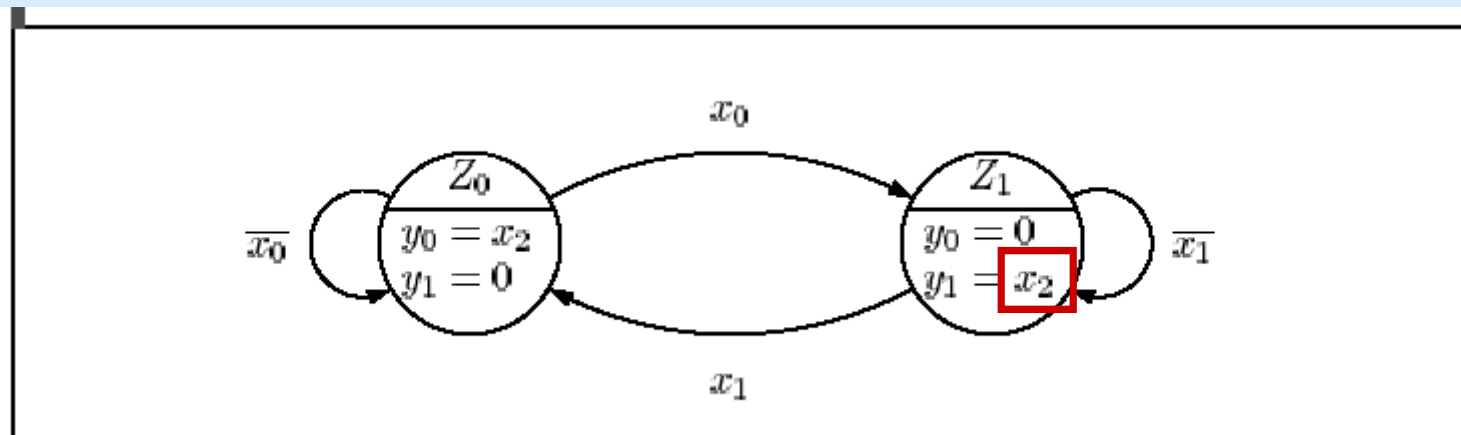
- 1-Belegung von z_0 in Z_1
- **hinföhrende Kanten**

y-Gleichungen

Ausgabefunktion:

$$y_k = \bigvee_{i=0}^{2^p-1} k_i(z) \wedge h_k^i(x) = h_k(z, x)$$

$$y_1 = z_0 \wedge x_2$$



Struktur-Gleichungen

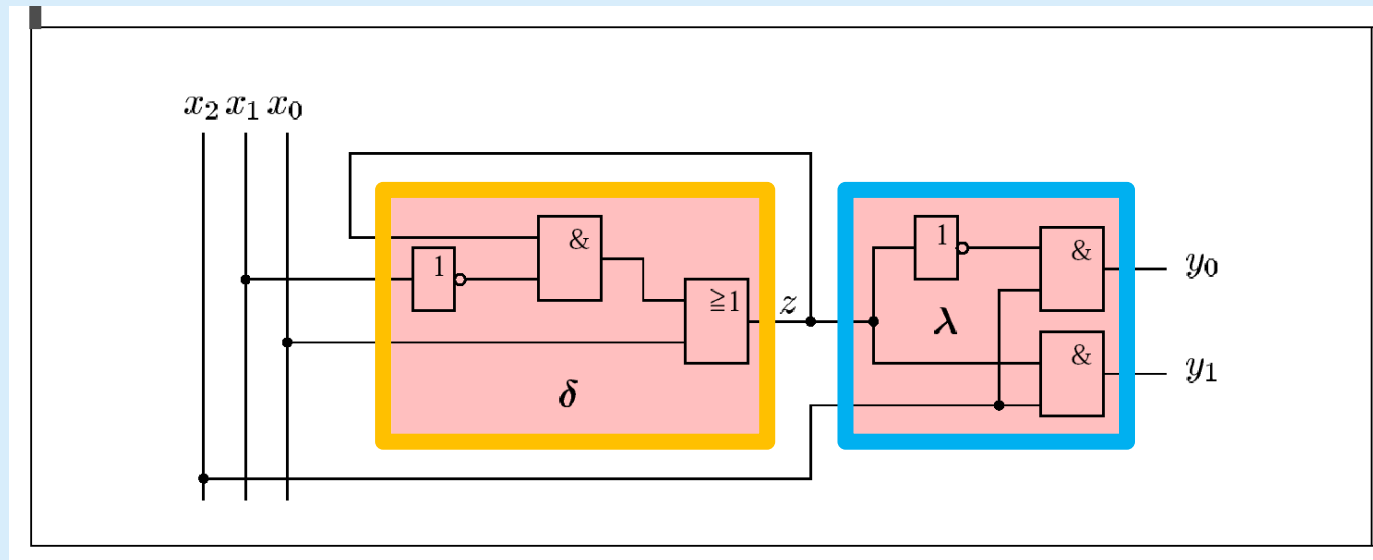
Zustandsüberföhrungsfunktion:

$$z_0 := x_0 \vee z_0 \wedge \bar{x}_1$$

Ausgabefunktion:

$$y_0 = \bar{z}_0 \wedge x_2$$

$$y_1 = z_0 \wedge x_2$$



Technische Informatik (RO)

Automatentypen (Mealy/Moore)

**Synthese sequentieller Strukturen
(z- und y- Gleichungen)**

**Verifikation (Vollständigkeit
& Widerspruchsfreiheit)**

Korrektur Entwurf?

= > formale Verifikation

Prüfung auf

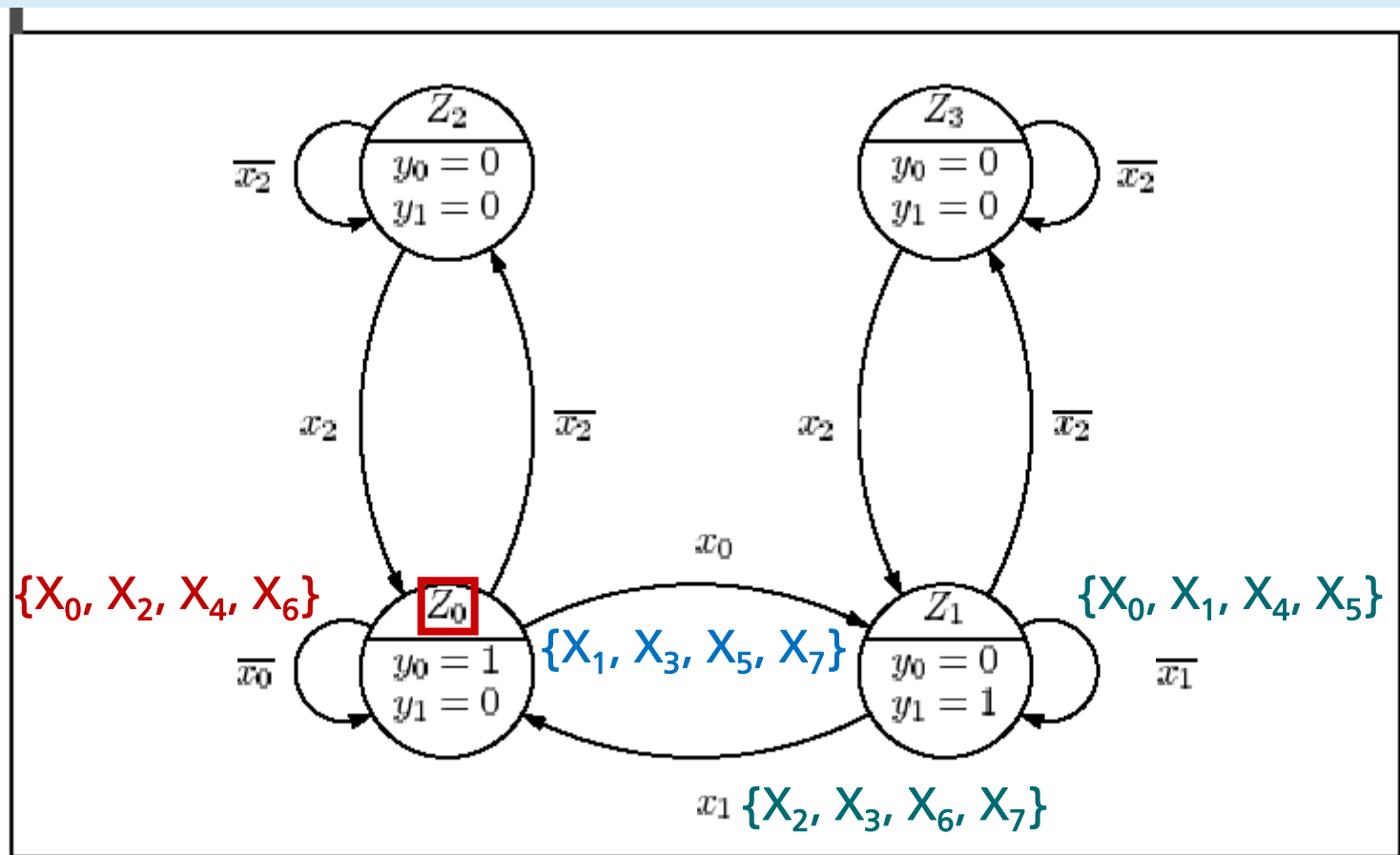
Vollständigkeit und

Widerspruchsfreiheit

Vollständigkeit

BAA => BMA je Zustand alle X_k

vollständig



Vollständigkeit

BMA

$$\{x_0, x_2, x_4, x_6\} \cup \{x_1, x_3, x_5, x_7\} = X$$

BAA

$$\overline{x_0} \vee x_0 = 1$$

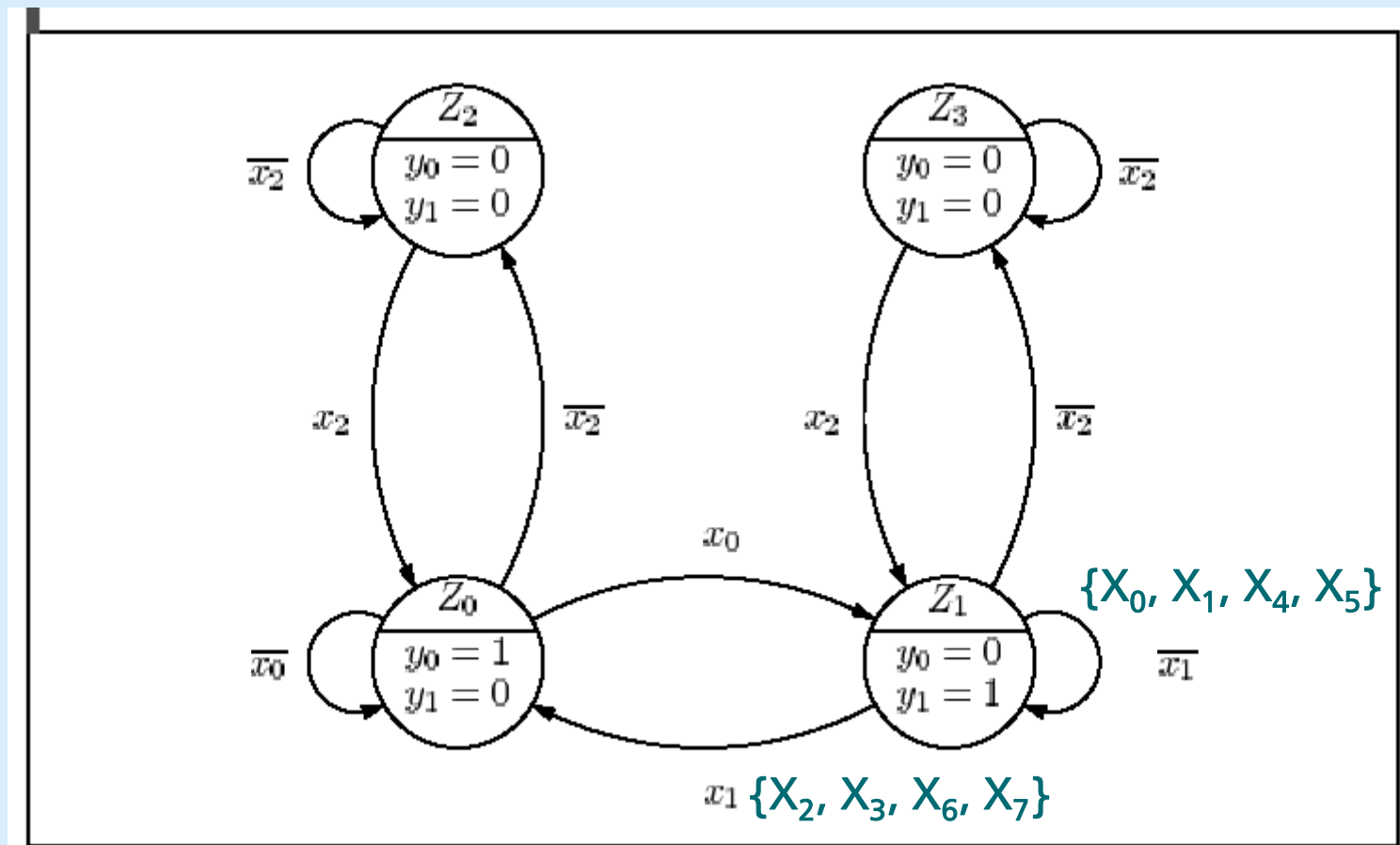
allgemein

$$\forall i \left(\bigvee_{j=0}^{2^p-1} h_{ij}(x) = 1 \right)$$

Für jeden Zustand einzeln testen !

Widerspruchsfreiheit

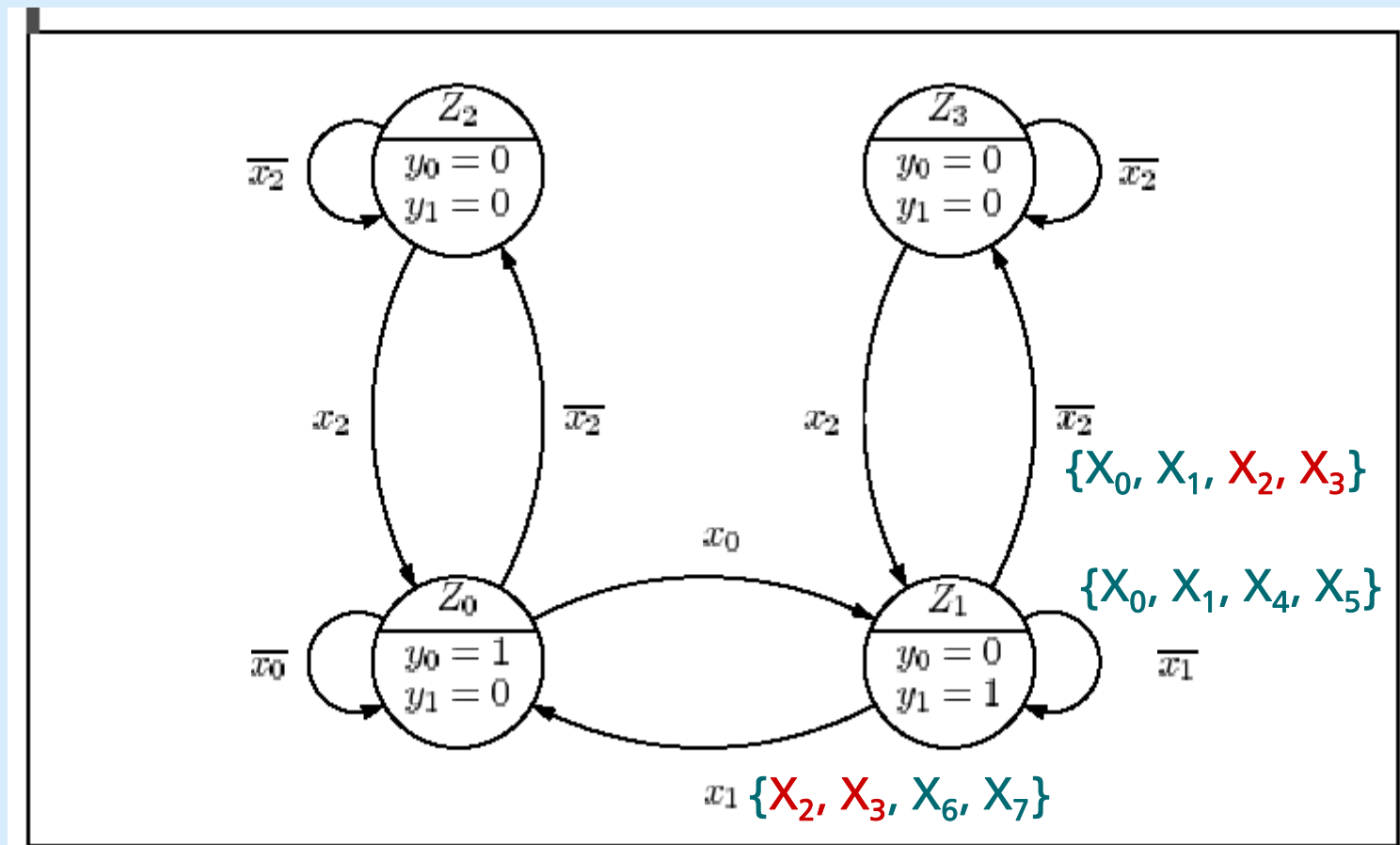
BAA => BMA je Kantenpaar
keine gleichen Belegungen



Widerspruchsfreiheit

BAA => BMA

Widerspruch



Widerspruchsfreiheit

BAA \Rightarrow BMA

Widerspruch

BMA: paarweise Schnittbildung

$$\{x_2, x_3, x_6, x_7\} \cap \{x_0, x_1, x_2, x_3\} = \{x_2, x_3\} \Rightarrow \neq \emptyset \dots \text{Widerspruch !}$$

BAA: paarweise Konjunktion

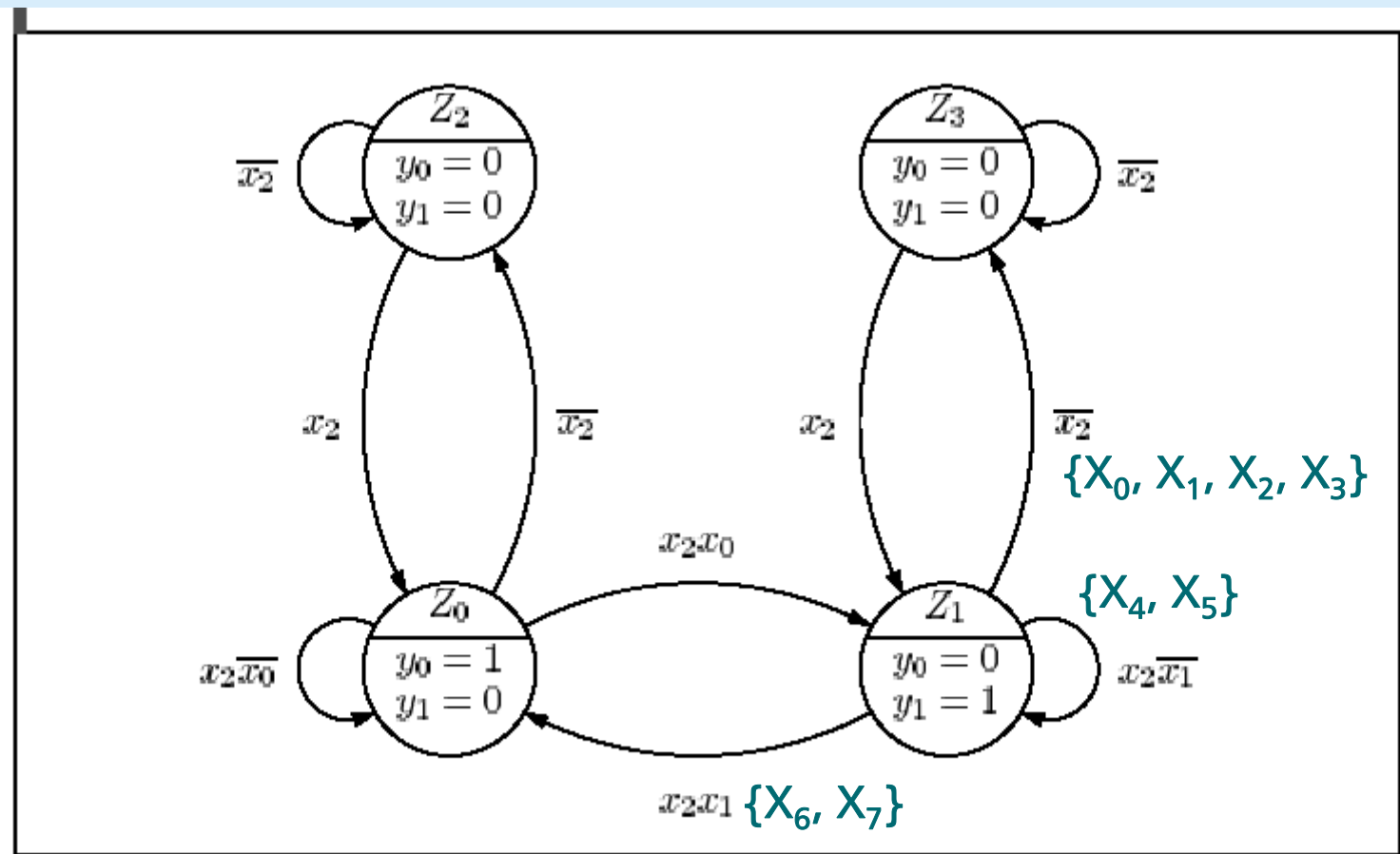
$$x_1 \wedge \overline{x_2} \neq 0 \dots \text{Widerspruch !}$$

BAA allgemein

$$\forall i \left(\bigvee_{\substack{j,k=0 \\ j \neq k}}^{2^p - 1} h_{ij}(x) \wedge h_{ik}(x) = 0 \right)$$

Widerspruchsfreiheit

Vergleich mit **Aufgabe** und **Widerspruch auflösen**



Das war's für heute

Viel Spaß beim Wiederholen!



Kap. 5.2.4, 5.2.5

Bis nächsten Dienstag um 15.00 ...