

4-Wege-Automaten und Tiling-Systeme

Angelika Schwarz

6.2.2012

Seminar über Angewandte Automatentheorie

<http://www.automata.rwth-aachen.de/teaching/ws1112/seminar.en>

Motivation

- Reguläre eindimensionale Sprachen
 - werden durch **endliche Automaten** erkannt
 - können durch **MSO-Formeln** definiert werden
 - können als **regulärer Ausdruck** angegeben werden
- Verallgemeinerung von 1D nach 2D
 - Wie können die Modelle erweitert werden?
 - Bleiben Abschlusseigenschaften erhalten?
 - Entscheidbarkeit von Leerheitsproblem,....?

Definition

- Ein *picture* ist eine Matrix mit Einträgen aus Σ .
- Σ^{**} ist die Menge aller pictures über Σ .
- Die $0 \times n$ bzw. $n \times 0$ Matrix heißt *leeres picture* und wird mit λ bezeichnet.

Beispiel

$$\Sigma = \{a, b\}$$

$$L = \{p \in \Sigma^{**} \mid l_1(p) = 2 \cdot l_2(p) \text{ und } p(i, 1) = a \text{ für } 1 \leq i \leq l_1(p) \}$$

a	a	a
a	b	b
a	a	a
a	b	a
a	a	b
a	a	a

a	b
a	a
a	a
a	a

a	b
a	b
a	b
a	b

Definition

Ein picture $p \in \Sigma^{**}$ mit Rahmenbegrenzung und Begrenzungssymbol $\# \notin \Sigma$:

$$\hat{p} = \begin{array}{|c|c|c|} \hline \# & \# & \# \\ \hline \# & p & \# \\ \hline \# & \# & \# \\ \hline \end{array}$$

Beispiel:

$$p = \begin{array}{|c|c|} \hline a & b \\ \hline a & a \\ \hline a & a \\ \hline \end{array} \rightsquigarrow \hat{p} = \begin{array}{|c|c|c|c|} \hline \# & \# & \# & \# \\ \hline \# & a & b & \# \\ \hline \# & a & a & \# \\ \hline \# & a & a & \# \\ \hline \# & \# & \# & \# \\ \hline \end{array}$$

Definition

- *Spaltenkonkatenation*: Falls pictures p und q bezüglich ihrer Zeilenanzahl übereinstimmen ist

$$p \oplus q = \begin{array}{|ccc|ccc|} \hline p_{11} & \dots & p_{1n} & q_{11} & \dots & q_{1n'} \\ \vdots & & \vdots & \vdots & & \vdots \\ p_{m1} & \dots & p_{mn} & q_{m1} & \dots & q_{mn'} \\ \hline \end{array}$$

- *Zeilenkonkatenation*: Falls die Spaltenanzahl übereinstimmt:

$$p \ominus q = \begin{array}{|ccc|ccc|} \hline p_{11} & \dots & p_{1n} & & & \\ \vdots & & \vdots & & & \\ p_{m1} & \dots & p_{mn} & & & \\ \hline q_{11} & \dots & q_{1n} & & & \\ \vdots & & \vdots & & & \\ q_{m'1} & \dots & q_{m'n} & & & \\ \hline \end{array}$$

- Konkatenation mit dem leeren picture:

$$\lambda \oplus p = p = p \oplus \lambda$$

$$\lambda \ominus p = p = p \ominus \lambda$$

Definition

Die *Kleensche Hülle* für eine zweidimensionale Sprachen $L \subseteq \Sigma^{**}$ ist definiert als:

- Abschluss unter Spaltenkonkatenation.

$$L^{*\oplus} := \bigcup_{i \geq 0} L^{i\oplus}$$

$$\text{mit } L^{0\oplus} := \lambda, L^{1\oplus} := L \text{ und } L^{n\oplus} := L \oplus L^{(n-1)\oplus}$$

- Abschluss unter Zeilenkonkatenation.

$$L^{*\ominus} := \bigcup_{i \geq 0} L^{i\ominus}$$

$$\text{mit } L^{0\ominus} := \lambda, L^{1\ominus} := L \text{ und } L^{n\ominus} := L \ominus L^{(n-1)\ominus}$$

Übersicht

Jetzt:

- 2OTA
- Logische Formeln
- PCFRE

two-dimensional on-line tessellation automata (2OTA)

Definition

Ein *nichtdeterministischer 2OTA* ist ein Automat $\mathcal{A} = (\Sigma, Q, I, F, \delta)$ mit

- Σ Eingabealphabet
- Q Zustandsmenge
- $I \subseteq Q$ eine Menge von Startzuständen
- $F \subseteq Q$ eine Menge von Endzuständen
- $\delta : Q \times Q \times \Sigma \rightarrow 2^Q$ als Transitionsrelation

two-dimensional on-line tessellation automata (2OTA)

Definition

Ein *nichtdeterministischer 2OTA* ist ein Automat $\mathcal{A} = (\Sigma, Q, I, F, \delta)$ mit

- Σ Eingabealphabet
- Q Zustandsmenge
- $I \subseteq Q$ eine Menge von Startzuständen
- $F \subseteq Q$ eine Menge von Endzuständen
- $\delta : Q \times Q \times \Sigma \rightarrow 2^Q$ als Transitionsrelation

Bemerkung

Ein *deterministischer 2OTA* erkennt weniger als ein 2OTA.

#	#	#	#	#	#	#	#	#	#
#									#
#									#
#									#
#									#
#									#
#	#	#	#	#	#	#	#	#	#

#	#	#	#	#	#	#	#	#	#
#									#
#									#
#									#
#									#
#									#
#	#	#	#	#	#	#	#	#	#

$t = 0 : \delta(q_0, q_0, p(1, 1))$

#	#	#	#	#	#	#	#	#	#
#									#
#									#
#									#
#									#
#									#
#	#	#	#	#	#	#	#	#	#

$$t = 1 : \delta(q_0, q_{(1,1)}, p(1, 2))$$

$$\delta(q_{(1,1)}, q_0, p(2, 1))$$

#	#	#	#	#	#	#	#	#	#
#			↓						#
#		↓	→						#
#	→	↓							#
#									#
#									#
#	#	#	#	#	#	#	#	#	#

$$\begin{aligned}
 t = 2 : \quad & \delta(q_0, q_{(1,2)}, p(1, 3)) \\
 & \delta(q_{(1,2)}, q_{(2,1)}, p(2, 2)) \\
 & \delta(q_{(2,1)}, q_0, p(3, 1))
 \end{aligned}$$

#	#	#	#	#	#	#	#	#	#
#				↓					#
#			↓	→					#
#		↓	→						#
#	↓	→							#
#	→	↓							#
#									#
#	#	#	#	#	#	#	#	#	#

$t = 3$

#	#	#	#	#	#	#	#	#	#
#									#
#									#
#									#
#									#
#									#
#									#
#	#	#	#	#	#	#	#	#	#

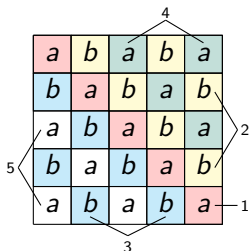


Wenn $(l_1(p), l_2(p)) \in F$, dann akzeptiert $\mathcal{A}$ das picture.

Schachbrettsprache

$$\left\{ \begin{array}{c} a \\ \hline \end{array}, \begin{array}{|c|c|} \hline a & b \\ \hline b & a \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline a & b & a \\ \hline b & a & b \\ \hline a & b & a \\ \hline \end{array}, \begin{array}{|c|c|c|c|} \hline a & b & a & b \\ \hline b & a & b & a \\ \hline a & b & a & b \\ \hline b & a & b & a \\ \hline \end{array}, \begin{array}{|c|c|c|c|c|} \hline a & b & a & b & a \\ \hline b & a & b & a & b \\ \hline a & b & a & b & a \\ \hline b & a & b & a & b \\ \hline a & b & a & b & a \\ \hline \end{array}, \dots \right\}$$

Beispiel Schachbrettsprache



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

$$I = \{0\}$$

$$F = \{1\}$$

 $\delta :$

	0		1		2		3		4		5	
	a	b	a	b	a	b	a	b	a	b	a	b
0	1		3				5				3	
1		2							2			
2	4				4							
3					1		5					
4		2							2			
5			3								3	

Logische Formeln

Definition

Ein picture p der Größe (m, n) kann mithilfe der Signatur $\underline{p} = (dom(p), S_1, S_2, (P_a)_{a \in \Sigma})$ dargestellt werden, wobei:

- $dom(p) = \{1, \dots, l_1(p)\} \times \{1, \dots, l_2(p)\}$
- S_1, S_2 binäre Nachfolgerrelationen; für zwei Positionen aus $dom(p)$ ist: $(i, j)S_1(i+1, j)$ und $(i, j)S_2(i, j+1)$ für $1 \leq i \leq m, 1 \leq j \leq n$
- $P_a = \{(i, j) \mid p(i, j) = a\}$ für $a \in \Sigma$

Logische Formeln

Definition

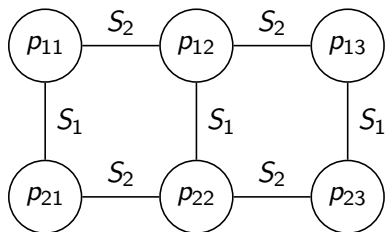
Ein picture p der Größe (m, n) kann mithilfe der Signatur $\underline{p} = (dom(p), S_1, S_2, (P_a)_{a \in \Sigma})$ dargestellt werden, wobei:

- $dom(p) = \{1, \dots, l_1(p)\} \times \{1, \dots, l_2(p)\}$
- S_1, S_2 binäre Nachfolgerrelationen; für zwei Positionen aus $dom(p)$ ist: $(i, j)S_1(i+1, j)$ und $(i, j)S_2(i, j+1)$ für $1 \leq i \leq m, 1 \leq j \leq n$
- $P_a = \{(i, j) \mid p(i, j) = a\}$ für $a \in \Sigma$

Anschauung:

$p =$

p_{11}	p_{12}	p_{13}
p_{21}	p_{22}	p_{23}



Beispiel Schachbrettsprache

Randeigenschaften:

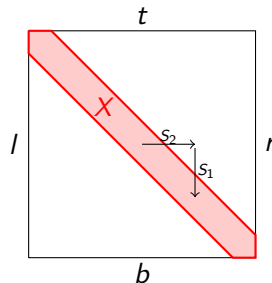
$$\varphi_t(x) = \neg \exists y (y S_1 x)$$

$$\varphi_l(x) = \neg \exists y (y S_2 x)$$

$$\varphi_{tl}(x) = \varphi_t(x) \wedge \varphi_l(x)$$

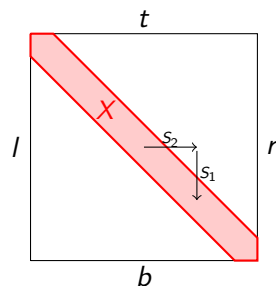
$\vdots$

Idee:



Beispiel Schachbrettsprache (fortgesetzt)

$$\begin{aligned}
 \varphi = & \exists X (\exists x (\varphi_{tl}(x) \wedge X(x)) \wedge \forall x (X(x) \rightarrow P_a(x)) \\
 & \wedge \forall x \forall y \forall z (X(x) \wedge xS_1y \wedge yS_2z \rightarrow X(z)) \\
 & \wedge \forall ((\varphi_b(x) \vee \varphi_r(x)) \rightarrow (\neg X(x) \vee \varphi_{br}(x)))) \\
 & \wedge \forall x \exists y (P_a(x) \wedge \neg \varphi_r(x) \rightarrow xS_2y \wedge P_b(y)) \\
 & \wedge \forall x \exists y (P_a(x) \wedge \neg \varphi_b(x) \rightarrow xS_1y \wedge P_b(y)) \\
 & \wedge \forall x \exists y (P_b(x) \wedge \neg \varphi_r(x) \rightarrow xS_2y \wedge P_a(y)) \\
 & \wedge \forall x \exists y (P_b(x) \wedge \neg \varphi_b(x) \rightarrow xS_1y \wedge P_a(y))
 \end{aligned}$$



reguläre Ausdrücke (RE)

Definition

Ein *regulärer Ausdruck* für zweidimensionale Sprachen über dem Alphabet Σ lässt sich aufbauen als:

- $\emptyset$ und jedes $a \in \Sigma$ sind reguläre Ausdrücke
- Für zwei reguläre Ausdrücke α, β sind ebenfalls regulär:
 - $\alpha \otimes \beta$ mit $\otimes \in \{\oplus, \ominus, \cup, \cap\}$
 - $\bar{\alpha}$
 - $(\alpha)^{* \oplus}$ bzw. $(\alpha)^{* \ominus}$

projection of complement-free regular expressions (PCFRE)

Definition

Eine *Projektion* ist eine Abbildung $\pi : \Gamma^{**} \rightarrow \Sigma^{**}$. Ein picture $p \in \Gamma^{**}$ wird Eintrag für Eintrag abgebildet: $p'(i, j) = \pi(p(i, j))$ für $1 \leq i \leq l_1(p), 1 \leq j \leq l_2(p)$.

Definition

- RE über $\{\oplus, \ominus, *^{\oplus}, *^{\ominus}, \cup, \cap\}$ heißen *complementation-free regular expressions* (CFRE).
- Ist eine Sprache $L \subseteq \Sigma^{**}$ durch Projektion π einer Sprache $L' \in L(\text{CFRE})$ und $\pi(L') = L$ entstanden, ist $L \in L(\text{PCFRE})$.

Beispiel Schachbrettsprache

Schachbretter ungerader Seitenlänge

a_0	b_2	a_2	b_2	a_2	b_2	a_2
b_1	a_0	b_2	a_2	b_2	a_2	b_2
a_1	b_1	a_0	b_2	a_2	b_2	a_2
b_1	a_1	b_1	a_0	b_2	a_2	b_2
a_1	b_1	a_1	b_1	a_0	b_2	a_2
b_1	a_1	b_1	a_1	b_1	a_0	b_2
a_1	b_1	a_1	b_1	a_1	b_1	a_0

Beispiel Schachbrettsprache

Schachbretter ungerader Seitenlänge

a_0	b_2	a_2	b_2	a_2	b_2	a_2
b_1	a_0	b_2	a_2	b_2	a_2	b_2
a_1	b_1	a_0	b_2	a_2	b_2	a_2
b_1	a_1	b_1	a_0	b_2	a_2	b_2
a_1	b_1	a_1	b_1	a_0	b_2	a_2
b_1	a_1	b_1	a_1	b_1	a_0	b_2
a_1	b_1	a_1	b_1	a_1	b_1	a_0

$$((a_2 \ominus b_2)^{* \ominus}) \ominus a_0 \ominus ((b_1 \ominus a_1)^{* \ominus})$$

Beispiel Schachbrettsprache

Schachbretter ungerader Seitenlänge

a_0	b_2	a_2	b_2	a_2	b_2	a_2
b_1	a_0	b_2	a_2	b_2	a_2	b_2
a_1	b_1	a_0	b_2	a_2	b_2	a_2
b_1	a_1	b_1	a_0	b_2	a_2	b_2
a_1	b_1	a_1	b_1	a_0	b_2	a_2
b_1	a_1	b_1	a_1	b_1	a_0	b_2
a_1	b_1	a_1	b_1	a_1	b_1	a_0

$$\beta_v^u = ((a_2 \ominus b_2)^{\ast\ominus}) \ominus a_0 \ominus ((b_1 \ominus a_1)^{\ast\ominus})$$

$$\cup b_2 \ominus ((a_2 \ominus b_2)^{\ast\ominus}) \ominus a_0 \ominus ((b_1 \ominus a_1)^{\ast\ominus}) \ominus b_1$$

Beispiel Schachbrettsprache

Schachbretter ungerader Seitenlänge

a_0	b_2	a_2	b_2	a_2	b_2	a_2
b_1	a_0	b_2	a_2	b_2	a_2	b_2
a_1	b_1	a_0	b_2	a_2	b_2	a_2
b_1	a_1	b_1	a_0	b_2	a_2	b_2
a_1	b_1	a_1	b_1	a_0	b_2	a_2
b_1	a_1	b_1	a_1	b_1	a_0	b_2
a_1	b_1	a_1	b_1	a_1	b_1	a_0

$$\beta_v^u = ((a_2 \ominus b_2)^{\ast\ominus}) \ominus a_0 \ominus ((b_1 \ominus a_1)^{\ast\ominus}) \\ \cup b_2 \ominus ((a_2 \ominus b_2)^{\ast\ominus}) \ominus a_0 \ominus ((b_1 \ominus a_1)^{\ast\ominus}) \ominus b_1$$

$$\beta_h^u = ((a_1 \oplus b_1)^{\ast\oplus}) \oplus a_0 \oplus ((b_2 \oplus a_2)^{\ast\oplus}) \\ \cup b_1 \oplus ((a_1 \oplus b_1)^{\ast\oplus}) \oplus a_0 \oplus ((b_2 \oplus a_2)^{\ast\oplus}) \oplus b_2$$

Beispiel Schachbrettsprache

Schachbretter ungerader Seitenlänge

a_0	b_2	a_2	b_2	a_2	b_2	a_2
b_1	a_0	b_2	a_2	b_2	a_2	b_2
a_1	b_1	a_0	b_2	a_2	b_2	a_2
b_1	a_1	b_1	a_0	b_2	a_2	b_2
a_1	b_1	a_1	b_1	a_0	b_2	a_2
b_1	a_1	b_1	a_1	b_1	a_0	b_2
a_1	b_1	a_1	b_1	a_1	b_1	a_0

$$\beta_v^u = ((a_2 \ominus b_2)^{* \ominus}) \ominus a_0 \ominus ((b_1 \ominus a_1)^{* \ominus}) \\ \cup b_2 \ominus ((a_2 \ominus b_2)^{* \ominus}) \ominus a_0 \ominus ((b_1 \ominus a_1)^{* \ominus}) \ominus b_1$$

$$\beta_h^u = ((a_1 \oplus b_1)^{* \oplus}) \oplus a_0 \oplus ((b_2 \oplus a_2)^{* \oplus}) \\ \cup b_1 \oplus ((a_1 \oplus b_1)^{* \oplus}) \oplus a_0 \oplus ((b_2 \oplus a_2)^{* \oplus}) \oplus b_2$$

$$L^u = ((\beta_v^u)^{* \oplus}) \cap ((\beta_h^u)^{* \ominus})$$

Beispiel Schachbrettsprache

- Schachbretter ungerader Länge über $\{a_0, a_1, a_2, b_1, b_2\}$

$$L^u = \left((\beta_v^u)^{* \oplus} \right) \cap \left((\beta_h^u)^{* \ominus} \right)$$

- Schachbretter gerader Länge über $\{a'_0, a'_1, a'_2, b'_1, b'_2\}$

$$L^g = \left((\beta_v^g)^{* \oplus} \right) \cap \left((\beta_h^g)^{* \ominus} \right)$$

- CFRE für Schachbretter:

$$L = L^u \cup L^g$$

- Projektion:

$$\pi(a_0) = \pi(a_1) = \pi(a_2) = \pi(a'_0) = \pi(a'_1) = \pi(a'_2) = a$$

$$\pi(b_1) = \pi(b_2) = \pi(b'_1) = \pi(b'_2) = b$$

$$L_{\text{Schachbrett}} = \pi(L)$$

Übersicht

Bisher gesehen:

- 2OTA
- Logische Formeln
- PCFRE

Jetzt:

- Tiling-Systeme
- Domino-Systeme

Definition

- Ein *Block* der Größe (h, w) ist ein Ausschnitt eines pictures der Größe (m, n) mit $h \leq m, w \leq n$. $B_{h,w}(p)$ bezeichnet die Menge aller Blöcke von p der Größe (h, w) .
- Ein *tile* ist ein Block der Größe $(2, 2)$.

Definition

Eine Sprache $L \subseteq \Gamma^{**}$ heißt *lokal*, falls es eine endliche Menge Θ von tiles gibt, so dass $L = \{p \in \Gamma^{**} \mid B_{2,2}(\hat{p}) \subseteq \Theta\}$. LOC bezeichnet die Menge aller lokalen Sprachen.

λ wird mit

#	#
#	#

 identifiziert.

Beispiel für eine lokale Sprache

$$\Gamma = \{a, b\}$$

$$L = \{p \in \Gamma^{**} \mid p(i, 1) = a \text{ und } p(i, 2) = b \text{ für } 1 \leq i \leq l_1(p) \\ \text{und } l_2(p) = 2\}$$

#	#	#	#
#	a	b	#
#	a	b	#
⋮	⋮	⋮	⋮
#	a	b	#
#	#	#	#

Beispiel für eine lokale Sprache

$$\Gamma = \{a, b\}$$

$$L = \{p \in \Gamma^{**} \mid p(i, 1) = a \text{ und } p(i, 2) = b \text{ für } 1 \leq i \leq l_1(p) \\ \text{und } l_2(p) = 2\}$$

#	#	#	#
#	a	b	#
#	a	b	#
:	:	:	:
#	a	b	#
#	#	#	#

 $\Theta = \left\{ \begin{array}{c} \begin{array}{|c|c|} \hline \# & \# \\ \hline \# & a \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline a & b \\ \hline \end{array}, \begin{array}{|c|c|} \hline a & b \\ \hline a & b \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & a \\ \hline \# & a \\ \hline \end{array}, \\ \\ \begin{array}{|c|c|} \hline b & \# \\ \hline b & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & a \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline b & \# \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline b & \# \\ \hline \end{array}, \\ \\ \begin{array}{|c|c|} \hline a & b \\ \hline \# & \# \\ \hline \end{array} \end{array} \right\}$

Tiling-Systeme (TS)

Definition

Ein *Tiling-System* $\mathcal{T}$ ist ein Quadrupel $\mathcal{T} = (\Sigma, \Gamma, \Theta, \pi)$ mit

- Σ, Γ zwei Alphabete
- Θ eine Menge von tiles über $\Gamma \cup \{\#\}$
- $\pi : \Gamma \rightarrow \Sigma$ eine Projektion

Ein Tiling-System erkennt eine Sprache $L \subseteq \Sigma^{**}$, falls $L = \pi(L(\Theta))$.

$$L(TS) = \{L \mid L \text{ ist durch ein Tiling-System erkennbar} \}$$

Beispiel Schachbrettsprache

- Die Schachbrettsprache ist keine lokale Sprache.

Beispiel Schachbrettsprache

- Die Schachbrettsprache ist keine lokale Sprache.
- stattdessen: Struktur durch lokale Sprache vorgeben

#	#	#	#	...	#	#	#
#	a_0	b_1	a_1	...			#
#	b_1	a_0	b_1				#
#	a_1	b_1	a_0				#
:	:			...			:
#						b_1	#
#					b_1	a_0	#
#	#	#	#	...	#	#	#

Beispiel Schachbrettsprache

- Die Schachbrettsprache ist keine lokale Sprache.
- stattdessen: Struktur durch lokale Sprache vorgeben

#	#	#	#	...	#	#	#
#	a_0	b_1	a_1	...			#
#	b_1	a_0	b_1				#
#	a_1	b_1	a_0				#
:	:			...			:
#						b_1	#
#					b_1	a_0	#
#	#	#	#	...	#	#	#

- Mit Projektion auf das „richtige“ Alphabet abbilden.

Beispiel Schachbrettsprache

lokale Sprache:

$$\Theta = \left\{ \begin{array}{l} \begin{array}{|c|c|} \hline \# & \# \\ \hline \# & a_0 \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline a_0 & b_1 \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline b_1 & a_1 \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline a_1 & b_1 \\ \hline \end{array}, \\ \\ \begin{array}{|c|c|} \hline \# & \# \\ \hline a_1 & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline b_1 & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline a_0 & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & a_0 \\ \hline \# & b_1 \\ \hline \end{array}, \\ \\ \begin{array}{|c|c|} \hline \# & b_1 \\ \hline \# & a_1 \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & a_1 \\ \hline \# & b_1 \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & a_1 \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & b_1 \\ \hline \# & \# \\ \hline \end{array}, \\ \\ \begin{array}{|c|c|} \hline \# & a_0 \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline a_1 & b_1 \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline b_1 & a_1 \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline a_0 & \# \\ \hline \# & \# \\ \hline \end{array}, \\ \\ \begin{array}{|c|c|} \hline b_1 & a_0 \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline a_1 & \# \\ \hline b_1 & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline b_1 & \# \\ \hline a_1 & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline b_1 & \# \\ \hline a_0 & \# \\ \hline \end{array}, \\ \\ \begin{array}{|c|c|} \hline a_0 & b_1 \\ \hline b_1 & a_0 \\ \hline \end{array}, \begin{array}{|c|c|} \hline b_1 & a_1 \\ \hline a_0 & b_1 \\ \hline \end{array}, \begin{array}{|c|c|} \hline b_1 & a_1 \\ \hline a_1 & b_1 \\ \hline \end{array}, \begin{array}{|c|c|} \hline a_1 & b_1 \\ \hline b_1 & a_1 \\ \hline \end{array}, \\ \\ \begin{array}{|c|c|} \hline b_1 & a_0 \\ \hline a_1 & b_1 \\ \hline \end{array} \end{array} \right\}$$

Projektion:

$$\begin{array}{ccc} \pi : & \Gamma & \rightarrow \Sigma \\ & \parallel & \parallel \\ & \{a_0, a_1, b_1\} & \{a, b\} \end{array} \quad \text{mit} \quad \begin{array}{l} \pi(a_0) = \pi(a_1) = a \\ \pi(b_1) = b \end{array}$$

Abschlusseigenschaften von $L(TS)$

Satz

- 1.) $L(TS)$ ist unter Projektion abgeschlossen.
- 2.) $L(TS)$ ist unter $\oplus$ und $\ominus$ abgeschlossen.
- 3.) $L(TS)$ ist unter $\oplus^*$ und $\ominus^*$ abgeschlossen.
- 4.) $L(TS)$ ist unter $\cup$ und $\cap$ abgeschlossen.
- 5.) $L(TS)$ ist **nicht** unter $\neg$ abgeschlossen.

Beweis von (5.)

$$\Sigma = \{a, b\}$$

$$L = \{p \in \Sigma^{**} \mid s \text{ ist quadratisches picture, } p = s \oplus s\}$$

$$p = \begin{array}{|c|} \hline s \\ \hline s \\ \hline \end{array}$$

Zeige:

(1) $L \notin L(TS)$ aber

(2) $\bar{L} \in L(TS)$

(1) $L \notin L(TS)$

- Angenommen, $L \in L(TS)$. $\Rightarrow$ es ex. lokale Sprache L' und eine Projektion π mit $\pi(L') = L$

(1) $L \notin L(TS)$

- Angenommen, $L \in L(TS)$. $\Rightarrow$ es ex. lokale Sprache L' und eine Projektion π mit $\pi(L') = L$
- $L_n = \left\{ p \in \Sigma^{**} \mid p = \begin{array}{|c|} \hline s \\ \hline s \\ \hline \end{array} \right\} 2n \}$
 - Wie viele pictures gibt es in L_n ?

(1) $L \notin L(TS)$

- Angenommen, $L \in L(TS)$. $\Rightarrow$ es ex. lokale Sprache L' und eine Projektion π mit $\pi(L') = L$
- $L_n = \left\{ p \in \Sigma^{**} \mid p = \begin{array}{|c|} \hline s \\ \hline s \\ \hline \end{array} \right\} 2n \}$
 - Wie viele pictures gibt es in L_n ? $|\Sigma|^{n^2}$

(1) $L \notin L(TS)$

- Angenommen, $L \in L(TS)$. $\Rightarrow$ es ex. lokale Sprache L' und eine Projektion π mit $\pi(L') = L$
- $L_n = \left\{ p \in \Sigma^{**} \mid p = \begin{array}{|c|} \hline s \\ \hline s \\ \hline \end{array} \right\} 2n \}$
 - Wie viele pictures gibt es in L_n ? $|\Sigma|^{n^2}$
- $L'_n = \{ p' \in \Gamma^{**} \mid \pi(p') = p, p \in L_n \}$
 - Wie stehen $|\Gamma|$ und $|\Sigma|$ zueinander? $|\Sigma| \leq |\Gamma|$ (o.B.d.A.)
 - Wie viele Möglichkeiten gibt es, die n -te und $(n+1)$ -te Zeile zu belegen?

(1) $L \notin L(TS)$

- Angenommen, $L \in L(TS)$. $\Rightarrow$ es ex. lokale Sprache L' und eine Projektion π mit $\pi(L') = L$

- $L_n = \left\{ p \in \Sigma^{**} \mid p = \begin{bmatrix} s \\ s \end{bmatrix} \right\} 2n \}$

➤ Wie viele pictures gibt es in L_n ? $|\Sigma|^{n^2}$

- $L'_n = \{ p' \in \Gamma^{**} \mid \pi(p') = p, p \in L_n \}$

➤ Wie stehen $|\Gamma|$ und $|\Sigma|$ zueinander? $|\Sigma| \leq |\Gamma|$ (o.B.d.A.)

➤ Wie viele Möglichkeiten gibt es, die n -te und $(n+1)$ -te Zeile zu belegen? $|\Gamma|^{2n}$

$\Rightarrow |\Sigma|^{n^2} > |\Gamma|^{2n}$ für große n

(1) $L \notin L(TS)$ (fortgesetzt)

- $|\Sigma|^{n^2} > |\Gamma|^{2n}$ für große n
- Sei n nun hinreichend groß.
- $p = \frac{s_p}{s_p} \in L_n, q = \frac{s_q}{s_q} \in L_n$ mit $s_p \neq s_q$

(1) $L \notin L(TS)$ (fortgesetzt)

- $|\Sigma|^{n^2} > |\Gamma|^{2n}$ für große n
- Sei n nun hinreichend groß.

- $p = \begin{bmatrix} s_p \\ s_p \end{bmatrix} \in L_n, q = \begin{bmatrix} s_q \\ s_q \end{bmatrix} \in L_n$ mit $s_p \neq s_q$

- zugehörige pictures in der lokalen Sprache:

$$p' = \begin{bmatrix} s'_p \\ s''_p \end{bmatrix} \in L'_n, q' = \begin{bmatrix} s'_q \\ s''_q \end{bmatrix} \in L'_n \text{ und die } n\text{-te und } (n+1)\text{-te} \\ \text{Zeile seien gleich.}$$

(1) $L \notin L(TS)$ (fortgesetzt)

- $|\Sigma|^{n^2} > |\Gamma|^{2n}$ für große n
- Sei n nun hinreichend groß.

- $p = \begin{array}{|c|} \hline s_p \\ \hline s_p \\ \hline \end{array} \in L_n, q = \begin{array}{|c|} \hline s_q \\ \hline s_q \\ \hline \end{array} \in L_n$ mit $s_p \neq s_q$

- zugehörige pictures in der lokalen Sprache:

$p' = \begin{array}{|c|} \hline s'_p \\ \hline s''_p \\ \hline \end{array} \in L'_n, q' = \begin{array}{|c|} \hline s'_q \\ \hline s''_q \\ \hline \end{array} \in L'_n$ und die n -te und $(n+1)$ te Zeile seien gleich.

$$\Rightarrow v' = \begin{array}{|c|} \hline s'_p \\ \hline s''_q \\ \hline \end{array} \in L'_n$$

(1) $L \notin L(TS)$ (fortgesetzt)

- $|\Sigma|^{n^2} > |\Gamma|^{2n}$ für große n
- Sei n nun hinreichend groß.

- $p = \begin{bmatrix} s_p \\ s_p \end{bmatrix} \in L_n, q = \begin{bmatrix} s_q \\ s_q \end{bmatrix} \in L_n$ mit $s_p \neq s_q$

- zugehörige pictures in der lokalen Sprache:

$p' = \begin{bmatrix} s'_p \\ s''_p \end{bmatrix} \in L'_n, q' = \begin{bmatrix} s'_q \\ s''_q \end{bmatrix} \in L'_n$ und die n -te und $(n+1)$ te Zeile seien gleich.

$$\Rightarrow v' = \begin{bmatrix} s'_p \\ s''_q \end{bmatrix} \in L'_n$$

$$\Rightarrow \pi(v') = \begin{bmatrix} s_p \\ s_q \end{bmatrix} \in L \not\vdash$$

(1) $L \notin L(TS)$ (fortgesetzt)

- $|\Sigma|^{n^2} > |\Gamma|^{2n}$ für große n
- Sei n nun hinreichend groß.

- $p = \begin{bmatrix} s_p \\ s_p \end{bmatrix} \in L_n, q = \begin{bmatrix} s_q \\ s_q \end{bmatrix} \in L_n$ mit $s_p \neq s_q$

- zugehörige pictures in der lokalen Sprache:

$p' = \begin{bmatrix} s'_p \\ s''_p \end{bmatrix} \in L'_n, q' = \begin{bmatrix} s'_q \\ s''_q \end{bmatrix} \in L'_n$ und die n -te und $(n+1)$ te Zeile seien gleich.

$$\Rightarrow v' = \begin{bmatrix} s'_p \\ s''_q \end{bmatrix} \in L'_n$$

$$\Rightarrow \pi(v') = \begin{bmatrix} s_p \\ s_q \end{bmatrix} \in L \nsubseteq$$

$$\Rightarrow L = \{p \in \Sigma^{**} \mid s \text{ ist quadratisches picture, } p = s \ominus s\} \notin L(TS)$$

(2) $\bar{L} \in L(TS)$

$$L = \{p \in \Sigma^{**} \mid s \text{ quadratisches picture, } p = s \ominus s\}$$

$$\bar{L} = \underbrace{\{p \in \Sigma^{**} \mid l_1(p) \neq 2 \cdot l_2(p)\}}_{L_1}$$

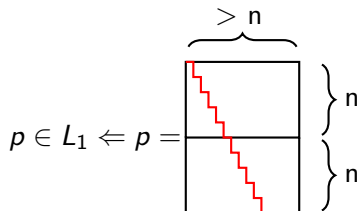
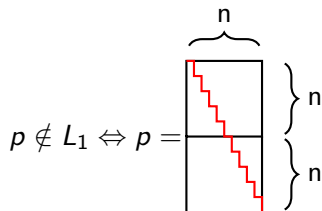
$$\cup \underbrace{\{p \in \Sigma^{**} \mid l_1(p) = 2 \cdot l_2(p), p = \begin{array}{|c|} \hline s \\ \hline t \\ \hline \end{array}, s \neq t\}}_{L_2}$$

Zeige, dass L_1 und L_2 durch TS erkennbar sind.

(2) $\bar{L} \in L(TS)$ (fortgesetzt)

$$L_1 = \{p \in \Sigma^{**} \mid l_1(p) \neq 2 \cdot l_2(p)\}$$

TS für L_1 (Idee):



(2) $\bar{L} \in L(TS)$ (fortgesetzt)

$$\begin{aligned}
 L_2 &= \{p \in \Sigma^{**} \mid l_1(p) = 2 \cdot l_2(p), p = \boxed{\frac{s}{t}}, s \neq t\} \\
 &= L_3 \cap (\Sigma^{**} \ominus (L_4 \cap (\Sigma^{**} \oplus L_5 \oplus \Sigma^{**}))) \ominus \Sigma^{**})
 \end{aligned}$$

mit

$$L_3 = \{p \in \Sigma^{**} \mid l_1(p) = 2l_2(p)\}$$

$$L_4 = \{p \in \Sigma^{**} \mid l_1(p) = l_2(p) + 1\}$$

$$L_5 = \{p \in \Sigma^{**} \mid l_2(p) = 1 \text{ und } p(1, 1) \neq p(1, l_1(p))\}$$

Wenn L_3 , L_4 und L_5 durch TS erkennbar sind, dann auch L_2 .

(2) $\bar{L} \in L(TS)$ (fortgesetzt)

$$L_2 = L_3 \cap (\Sigma^{**} \ominus (L_4 \cap (\Sigma^{**} \oplus L_5 \oplus \Sigma^{**})) \ominus \Sigma^{**})$$

$$L_3 = \{p \in \Sigma^{**} \mid l_1(p) = 2l_2(p)\}$$

$$L_4 = \{p \in \Sigma^{**} \mid l_1(p) = l_2(p) + 1\}$$

$$L_5 = \{p \in \Sigma^{**} \mid l_2(p) = 1 \text{ und } p(1,1) \neq p(1, l_1(p))\}$$

	a	
	*	
*	:	*
	:	
	*	
	b	

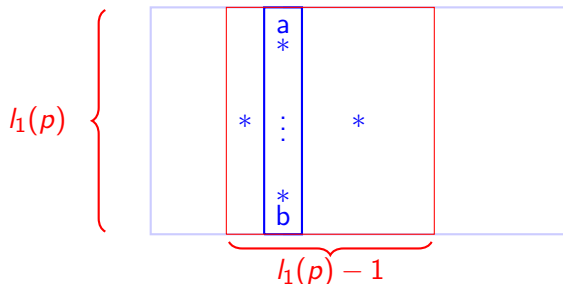
(2) $\bar{L} \in L(TS)$ (fortgesetzt)

$$L_2 = L_3 \cap (\Sigma^{**} \ominus (L_4 \cap (\Sigma^{**} \oplus L_5 \oplus \Sigma^{**})) \ominus \Sigma^{**})$$

$$L_3 = \{p \in \Sigma^{**} \mid l_1(p) = 2l_2(p)\}$$

$$L_4 = \{p \in \Sigma^{**} \mid l_1(p) = l_2(p) + 1\}$$

$$L_5 = \{p \in \Sigma^{**} \mid l_2(p) = 1 \text{ und } p(1, 1) \neq p(1, l_1(p))\}$$



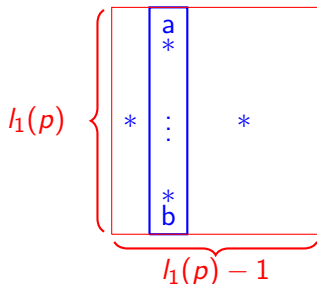
(2) $\bar{L} \in L(TS)$ (fortgesetzt)

$$L_2 = L_3 \cap (\Sigma^{**} \ominus (L_4 \cap (\Sigma^{**} \oplus L_5 \oplus \Sigma^{**})) \ominus \Sigma^{**})$$

$$L_3 = \{p \in \Sigma^{**} \mid l_1(p) = 2l_2(p)\}$$

$$L_4 = \{p \in \Sigma^{**} \mid l_1(p) = l_2(p) + 1\}$$

$$L_5 = \{p \in \Sigma^{**} \mid l_2(p) = 1 \text{ und } p(1, 1) \neq p(1, l_1(p))\}$$

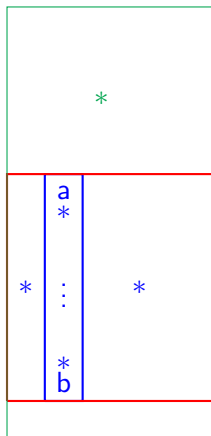


(2) $\bar{L} \in L(TS)$ (fortgesetzt)

$$L_2 = L_3 \cap (\Sigma^{**} \ominus (L_4 \cap (\Sigma^{**} \oplus L_5 \oplus \Sigma^{**}))) \ominus \Sigma^{**})$$

$$L_3 = \{p \in \Sigma^{**} \mid l_1(p) = 2l_2(p)\}$$

$$L_4 = \{p \in \Sigma^{**} \mid l_1(p) = l_2(p) + 1\}$$

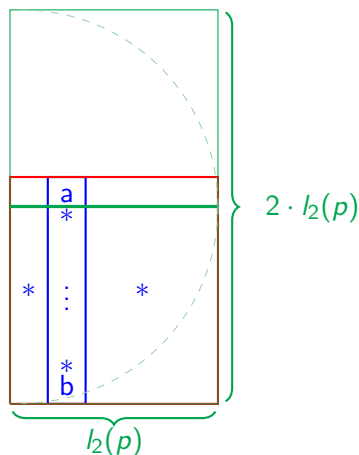


(2) $\bar{L} \in L(TS)$ (fortgesetzt)

$$L_2 = L_3 \cap (\Sigma^{**} \ominus (L_4 \cap (\Sigma^{**} \oplus L_5 \oplus \Sigma^{**}))) \ominus \Sigma^{**})$$

$$L_3 = \{p \in \Sigma^{**} \mid l_1(p) = 2l_2(p)\}$$

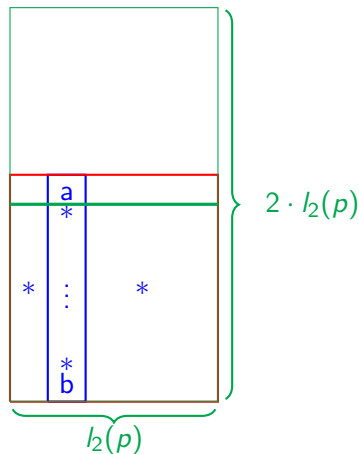
$$L_4 = \{p \in \Sigma^{**} \mid l_1(p) = l_2(p) + 1\}$$



(2) $\bar{L} \in L(TS)$ (fortgesetzt)

$$L_2 = \{p \in \Sigma^{**} \mid l_1(p) = 2 \cdot l_2(p), p = \begin{array}{|c|} \hline s \\ \hline t \\ \hline \end{array}, s \neq t\}$$

$$= L_3 \cap (\Sigma^{**} \ominus (L_4 \cap (\Sigma^{**} \oplus L_5 \oplus \Sigma^{**}))) \ominus \Sigma^{**})$$



Definition

Ein *Domino* ist ein Block der Größe $(1, 2)$ bzw. $(2, 1)$.

Definition

Eine Sprache $L \subseteq \Gamma^{**}$ heißt *hv-lokal*, falls es eine endliche Menge Δ von Dominos über $\Gamma \cup \{\#\}$ gibt, so dass $L = \{p \in \Gamma^{**} \mid (B_{1,2}(\hat{p}) \cup B_{2,1}(\hat{p})) \subseteq \Delta\}$.

$$\begin{array}{|c|c|} \hline \# & \# \\ \hline \end{array} \in L \text{ und } \begin{array}{|c|} \hline \# \\ \hline \# \\ \hline \end{array} \in L \Leftrightarrow \lambda \in L.$$

Domino-Systeme (DS)

Definition

Ein *Domino-System* $\mathcal{D}$ ist ein Quadrupel $\mathcal{D} = (\Sigma, \Gamma, \Delta, \pi)$ mit

- Σ, Γ zwei Alphabete
- Δ eine Menge von Dominos über $\Gamma \cup \{\#\}$
- $\pi : \Gamma \rightarrow \Sigma$ ein Projektion

Ein Domino-System erkennt eine Sprache $L \subseteq \Sigma^{**}$, falls $L = \pi(L(\Gamma))$.

$$L(DS) = \{L \mid L \text{ ist durch ein Domino-System erkennbar} \}$$

Beispiel Schachbrettsprache

Idee:

#	#	#	#	#	#	#
#	a_0	b_2	a_2	b_2	a_2	#
#	b_1	a_0	b_2	a_2	b_2	#
#	a_1	b_1	a_0	b_2	a_2	#
#	b_1	a_1	b_1	a_0	b_2	#
#	a_1	b_1	a_1	b_1	a_0	#
#	#	#	#	#	#	#

Beispiel Schachbrettsprache (fortgesetzt)

hv-lokale Sprache:

#	#	#	#	#	#	#
#	a_0	b_2	a_2	b_2	a_2	#
#	b_1	a_0	b_2	a_2	b_2	#
#	a_1	b_1	a_0	b_2	a_2	#
#	b_1	a_1	b_1	a_0	b_2	#
#	a_1	b_1	a_1	b_1	a_0	#
#	#	#	#	#	#	#

$$\Theta = \left\{ \begin{array}{llll} \begin{array}{|c|c|} \hline \# & \# \\ \hline \end{array}, & \begin{array}{|c|c|} \hline \# & a_0 \\ \hline \end{array}, & \begin{array}{|c|c|} \hline a_0 & b_2 \\ \hline \end{array}, & \begin{array}{|c|c|} \hline b_2 & a_2 \\ \hline \end{array}, \\ \begin{array}{|c|c|} \hline a_2 & b_2 \\ \hline \end{array}, & \begin{array}{|c|c|} \hline a_2 & \# \\ \hline \end{array}, & \begin{array}{|c|c|} \hline b_2 & \# \\ \hline \end{array}, & \begin{array}{|c|c|} \hline b_1 & a_0 \\ \hline \end{array}, \\ \begin{array}{|c|c|} \hline a_1 & b_1 \\ \hline \end{array}, & \begin{array}{|c|c|} \hline \# & b_1 \\ \hline \end{array}, & \begin{array}{|c|c|} \hline \# & a_1 \\ \hline \end{array}, & \begin{array}{|c|c|} \hline a_0 & \# \\ \hline \end{array}, \\ \begin{array}{|c|c|} \hline b_1 & a_1 \\ \hline \end{array}, & \begin{array}{|c|} \hline \# \\ \hline \# \\ \hline \end{array}, & \begin{array}{|c|} \hline \# \\ \hline a_0 \\ \hline \end{array}, & \begin{array}{|c|} \hline a_0 \\ \hline b_1 \\ \hline \end{array}, \\ \begin{array}{|c|} \hline b_2 \\ \hline a_0 \\ \hline \end{array}, & \begin{array}{|c|} \hline a_2 \\ \hline b_2 \\ \hline \end{array}, & \begin{array}{|c|} \hline b_2 \\ \hline a_2 \\ \hline \end{array}, & \begin{array}{|c|} \hline \# \\ \hline b_2 \\ \hline \end{array}, \\ \begin{array}{|c|} \hline \# \\ \hline a_2 \\ \hline \end{array}, & \begin{array}{|c|} \hline b_1 \\ \hline a_1 \\ \hline \end{array}, & \begin{array}{|c|} \hline a_1 \\ \hline \# \\ \hline \end{array}, & \begin{array}{|c|} \hline b_1 \\ \hline \# \\ \hline \end{array}, \\ \begin{array}{|c|} \hline a_0 \\ \hline \# \\ \hline \end{array}, & \begin{array}{|c|} \hline a_1 \\ \hline b_1 \\ \hline \end{array} & & \end{array} \right\}$$

Projektion:

$$\begin{aligned} \pi(a_0) &= \pi(a_1) = \pi(a_2) = a \\ \pi(b_1) &= \pi(b_2) = b \end{aligned}$$

Satz

Es sei L eine zweidimensionale Sprache. Dann sind die folgenden Aussagen äquivalent:

- 1.) L wird durch eine Tiling-System erkannt.
- 2.) L wird von einem 2OTA erkannt.
- 3.) L wird durch ein Domino-System erkannt.
- 4.) L ist durch eine EMSO-Formel definierbar.
- 5.) L ist darstellbar als komplementfreier regulärer Ausdruck mit Projektion, d.h. $L \in PCFRE$.

Definition

Die so definierten Sprachen werden mit *REC*, *recognizable two-dimensional languages* bezeichnet.

$$L(2OTA) = L(TS)$$

Die verschiedenen Modelle werden auf Tiling-Systeme zurückgeführt.

Zuerst:

- $L(2OTA) \subseteq L(TS)$
- $L(TS) \subseteq L(2OTA)$

$L(2OTA) \subseteq L(TS)$

$L \subseteq \Sigma^{**}$, 2OTA $\mathcal{A} = (\Sigma, Q, I, F, \delta)$ mit $L = L(\mathcal{A})$.

Konstruiere TS $\mathcal{T} = (\Sigma, \Gamma, \Theta, \pi)$ mit $L(\mathcal{T}) = L$

- $\Gamma = \Sigma \cup \{\#\} \times Q$
- $\Theta = \Theta_m \cup \Theta_l \cup \Theta_t \cup \Theta_r \cup \Theta_b$
 $\cup \Theta_{tl} \cup \Theta_{tr} \cup \Theta_{bl} \cup \Theta_{br}$

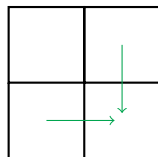
$$\triangleright \Theta_m = \left\{ \left(\begin{array}{cc} (a, r) & (b, s) \\ (c, t) & (d, q) \end{array} \right) \mid \begin{array}{l} a, b, c, d \neq \#, \\ q \in \delta(s, t, d) \end{array} \right\}$$

$$\triangleright \Theta_l = \left\{ \left(\begin{array}{cc} (\#, q_0) & (b, s) \\ (\#, q_0) & (d, q) \end{array} \right) \mid \begin{array}{l} b, d \neq \#, q_0 \in I, \\ q \in \delta(s, q_0, d) \end{array} \right\}$$

$\triangleright \dots$

- $\pi(a, q) = a$

Transitionen
eines 2OTAS



$$\delta : Q \times Q \times \Sigma \rightarrow 2^Q$$

$L(TS) \subseteq L(2OTA)$

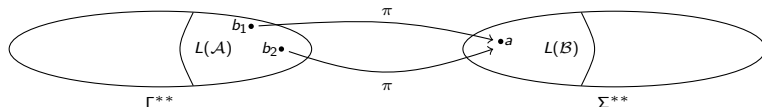
Einschub:

Lemma

$L(2OTA)$ ist unter Projektionen abgeschlossen.

Beweis:

$\pi : \Gamma \rightarrow \Sigma$ eine Projektion, $L \subseteq \Gamma^{**}$. 2OTA $\mathcal{A} = (\Gamma, Q, I, F, \delta)$ erkenne L . Konstruiere 2OTA $\mathcal{B}$, der $\pi(L)$ erkennt.



$$\mathcal{B} = (\Sigma, Q, I, F, \delta') \text{ mit}$$

$$\delta'(p, q, a) = \bigcup_{b \in \pi^{-1}(a)} \delta(p, q, b)$$

$L(TS) \subseteq L(2OTA)$ (fortgesetzt)

$L \subseteq \Sigma^{**}$, TS $\mathcal{T} = (\Sigma, \Gamma, \Theta, \pi)$ mit $L = L(\mathcal{T}) = \pi(L(\Theta))$.

$L(2OTA)$ unter Projektionen abgeschlossen $\Rightarrow$ Konstruiere 2OTA $\mathcal{A}$ mit $L(\mathcal{A}) = L(\Theta)$.

2OTA $\mathcal{A} = (\Gamma, Q, I, F, \delta)$ mit

- $Q = \Theta$
- $I = \left\{ \begin{array}{|c|c|} \hline \# & b \\ \hline c & d \\ \hline \end{array} \mid b, c, d \in \Gamma \cup \{\#\} \text{ und } \begin{array}{|c|c|} \hline \# & b \\ \hline c & d \\ \hline \end{array} \in \Theta \right\}$
- $F = \left\{ \begin{array}{|c|c|} \hline a & \# \\ \hline \# & \# \\ \hline \end{array} \mid \begin{array}{|c|c|} \hline a & \# \\ \hline \# & \# \\ \hline \end{array} \in \Theta \right\}$

$L(TS) \subseteq L(2OTA)$ (fortgesetzt)

- $\delta : Q \times Q \times \Sigma \rightarrow 2^Q$ so dass für alle $\begin{array}{|c|c|} \hline x & y \\ \hline a & b \\ \hline \end{array}, \begin{array}{|c|c|} \hline z & a \\ \hline t & c \\ \hline \end{array} \in Q$:

$$\delta \left(\begin{array}{|c|c|} \hline x & y \\ \hline a & b \\ \hline \end{array}, \begin{array}{|c|c|} \hline z & a \\ \hline t & c \\ \hline \end{array}, a \right) = \left\{ \begin{array}{|c|c|} \hline a & b \\ \hline c & d \\ \hline \end{array} \parallel \begin{array}{|c|c|} \hline a & b \\ \hline c & d \\ \hline \end{array} \in Q \right\}$$

simuliert einen Tiling-Schritt:

	x	y
z	a	b
t	c	d

$$L(TS) = L(DS)$$

- Zeige:

$$L(DS) \subseteq L(TS)$$

$$L(TS) \subseteq L(DS)$$

- Erinnerung:

$$\text{Domino-System} \quad \mathcal{D} = (\Sigma, \Gamma_D, \Delta, \pi_D), \quad \pi_D : \Gamma_D \rightarrow \Sigma$$

$$\text{Tiling-System} \quad \mathcal{T} = (\Sigma, \Gamma_T, \Theta, \pi_T), \quad \pi_T : \Gamma_T \rightarrow \Sigma$$

- Zeige, dass die lokalen Sprachen $L(\Theta)$ und $L(\Delta)$ ineinander überführbar sind.
- $\Rightarrow L(DS) = L(TS)$

$$L(DS) \subseteq L(TS)$$

- $\mathcal{D} = (\Sigma, \Gamma, \Delta, \pi)$ Domino-System
- Δ Menge von Dominos ,  und $L(\Delta)$ die zugehörige hv-lokale Sprache.
- Zeige: Es lässt sich eine Menge Θ von tiles konstruieren, so dass $L(\Delta) = L(\Theta)$
- damit direkt $\pi(L(\Delta)) = \pi(L(\Theta))$

$L(DS) \subseteq L(TS)$ (fortgesetzt)

Idee: Anlegeschritte von Dominos können durch tiles simuliert werden

$$\Delta = \left\{ \begin{array}{|c|} \hline a \\ \hline c \\ \hline \end{array}, \begin{array}{|c|} \hline b \\ \hline d \\ \hline \end{array}, \begin{array}{|c|} \hline a \\ \hline e \\ \hline \end{array}, \begin{array}{|c|} \hline b \\ \hline f \\ \hline \end{array}, \begin{array}{|c|c|} \hline a & b \\ \hline \end{array}, \begin{array}{|c|c|} \hline c & d \\ \hline \end{array}, \begin{array}{|c|c|} \hline e & f \\ \hline \end{array} \right\}$$

- Welche pictures können durch Anlegeschritte von Dominos erstellt werden?

$L(DS) \subseteq L(TS)$ (fortgesetzt)

Idee: Anlegeschritte von Dominos können durch tiles simuliert werden

$$\Delta = \left\{ \begin{array}{|c|c|} \hline a & b \\ \hline c & d \\ \hline \end{array}, \begin{array}{|c|c|} \hline b & a \\ \hline d & e \\ \hline \end{array}, \begin{array}{|c|c|} \hline a & b \\ \hline e & f \\ \hline \end{array}, \begin{array}{|c|c|} \hline a & b \\ \hline c & d \\ \hline \end{array}, \begin{array}{|c|c|} \hline c & d \\ \hline e & f \\ \hline \end{array} \right\}$$

- Welche pictures können durch Anlegeschritte von Dominos erstellt werden?

$$\begin{array}{|c|c|} \hline a & b \\ \hline c & d \\ \hline \end{array}, \begin{array}{|c|c|} \hline a & b \\ \hline e & f \\ \hline \end{array}$$

- $\Theta = \left\{ \theta \in (\Gamma \cup \{\#\})^{2 \times 2} \mid \theta \neq \begin{array}{|c|c|} \hline \# & \# \\ \hline \# & \# \\ \hline \end{array}, (B_{1,2}(\theta) \cup B_{2,1}(\theta)) \subseteq \Delta \right\}$
- $\mathcal{T} = (\Sigma, \Gamma, \Theta, \pi)$

$$L(TS) \subseteq L(DS)$$

Warum geht der Beweis nicht auch in die umgekehrte Richtung?

$$L(TS) \subseteq L(DS)$$

Warum geht der Beweis nicht auch in die umgekehrte Richtung?
 Beispiel (spezielles Schachbrett):

$$\begin{aligned}
 L &= \left\{ \begin{array}{|c|c|c|c|} \hline \# & \# & \# & \# \\ \hline \# & a & b & \# \\ \hline \# & b & a & \# \\ \hline \# & \# & \# & \# \\ \hline \end{array} \right\} \\
 \Theta &= \left\{ \begin{array}{|c|c|} \hline \# & \# \\ \hline \# & a \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline a & b \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline b & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & a \\ \hline \# & b \\ \hline \end{array}, \begin{array}{|c|c|} \hline a & b \\ \hline b & a \\ \hline \end{array}, \right. \\
 &\quad \left. \begin{array}{|c|c|} \hline b & \# \\ \hline a & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & b \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline b & a \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline a & \# \\ \hline \# & \# \\ \hline \end{array} \right\} \\
 \Delta &= \left\{ \begin{array}{|c|c|} \hline a_1 & a_2 \\ \hline \end{array}, \begin{array}{|c|} \hline a_1 \\ \hline a_2 \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|} \hline \# \\ \hline \# \\ \hline \end{array} \mid a_1, a_2 \in \{a, b, \#\}, a_1 \neq a_2, \right\}
 \end{aligned}$$

$$L(TS) \subseteq L(DS)$$

Warum geht der Beweis nicht auch in die umgekehrte Richtung?
Beispiel (spezielles Schachbrett):

$$L = \left\{ \begin{array}{|c|c|c|c|} \hline \# & \# & \# & \# \\ \hline \# & a & b & \# \\ \hline \# & b & a & \# \\ \hline \# & \# & \# & \# \\ \hline \end{array} \right\}$$

$$\Theta = \left\{ \begin{array}{|c|c|} \hline \# & \# \\ \hline \# & a \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline a & b \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline b & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & a \\ \hline \# & b \\ \hline \end{array}, \begin{array}{|c|c|} \hline a & b \\ \hline b & a \\ \hline \end{array}, \right. \\ \left. \begin{array}{|c|c|} \hline b & \# \\ \hline a & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & b \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline b & a \\ \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline a & \# \\ \hline \# & \# \\ \hline \end{array} \right\}$$

$$\Delta = \left\{ \begin{array}{|c|c|} \hline a_1 & a_2 \\ \hline \end{array}, \begin{array}{|c|} \hline a_1 \\ \hline a_2 \\ \hline \end{array}, \begin{array}{|c|c|} \hline \# & \# \\ \hline \end{array}, \begin{array}{|c|} \hline \# \\ \hline \# \\ \hline \end{array} \mid a_1, a_2 \in \{a, b, \#\}, a_1 \neq a_2, \right\}$$

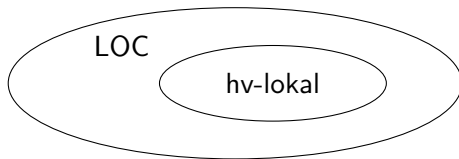
Dann ist z.B.

#	#	#	#	#
#	a	b	a	#
#	#	#	#	#

durch Dominos konstruierbar.

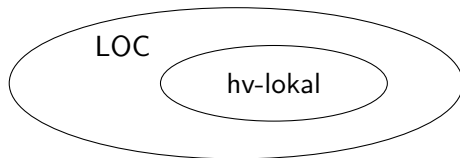
$L(TS) \subseteq L(DS)$ (fortgesetzt)

- Die Struktur (Quadrat) kann nicht durch Dominos kodiert werden.



$L(TS) \subseteq L(DS)$ (fortgesetzt)

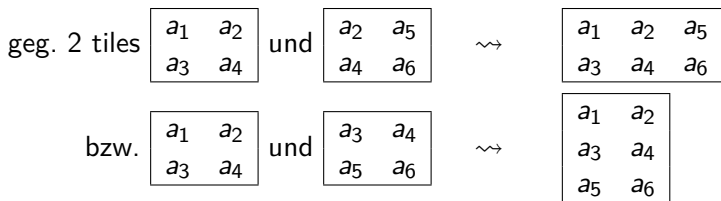
- Die Struktur (Quadrat) kann nicht durch Dominos kodiert werden.



- Idee:
 - Konstruiere Dominos, die Tiling-Schritte simulieren
 - nutze Abbildung, so dass $\pi(L(\Delta)) = L(\Theta)$

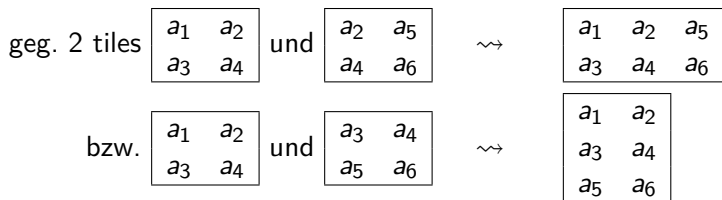
$L(TS) \subseteq L(DS)$ (fortgesetzt)

- Wie funktioniert Tiling?

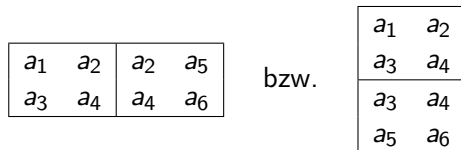


$L(TS) \subseteq L(DS)$ (fortgesetzt)

- Wie funktioniert Tiling?



- nehme alle möglichen Tiling-Schritte als Dominos



$L(TS) \subseteq L(DS)$ (fortgesetzt)

$$\Gamma := \Theta$$

$$\Delta := \left\{ \begin{array}{c} \begin{array}{|c|c|c|c|} \hline a_1 & a_2 & b_1 & b_2 \\ \hline a_3 & a_4 & b_3 & b_4 \\ \hline \end{array}, \begin{array}{|c|c|} \hline c_1 & c_2 \\ \hline c_3 & c_4 \\ \hline d_1 & d_2 \\ \hline d_3 & d_4 \\ \hline \end{array} \mid \begin{array}{c} a_2 = b_1, a_4 = b_3, c_3 = d_1, c_4 = d_2 \\ \begin{array}{|c|c|} \hline a_1 & a_2 \\ \hline a_3 & a_4 \\ \hline \end{array}, \begin{array}{|c|c|} \hline a_1 & a_2 \\ \hline a_3 & a_4 \\ \hline \end{array}, \begin{array}{|c|c|} \hline a_1 & a_2 \\ \hline a_3 & a_4 \\ \hline \end{array}, \begin{array}{|c|c|} \hline a_1 & a_2 \\ \hline a_3 & a_4 \\ \hline \end{array} \in \Gamma \end{array} \right\}$$

$$\cup \left\{ \begin{array}{c} \begin{array}{|c|c|c|c|} \hline b_2 & \# & \# & \# \\ \hline b_4 & \# & \# & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline d_3 & d_4 \\ \hline \# & \# \\ \hline \# & \# \\ \hline \# & \# \\ \hline \end{array} \mid \begin{array}{c} \begin{array}{|c|c|} \hline b_2 & \# \\ \hline b_4 & \# \\ \hline \end{array}, \begin{array}{|c|c|} \hline d_3 & d_4 \\ \hline \# & \# \\ \hline \end{array} \in \Gamma \end{array} \right\}$$

Beispiel:

$$\hat{p} = \begin{array}{|c|c|c|c|} \hline \# & \# & \# & \# \\ \hline \# & p_1 & p_2 & \# \\ \hline \# & \# & \# & \# \\ \hline \end{array} \rightsquigarrow \hat{p}' = \begin{array}{|c|c|c|c|} \hline \# & \# & \# & \# \\ \hline \# & p_1 & p_1 & p_2 \\ \hline \# & p_1 & p_1 & p_2 \\ \hline \# & \# & \# & \# \\ \hline \end{array}$$

$L(TS) \subseteq L(DS)$ (fortgesetzt)

$$\pi : \Gamma \rightarrow \Sigma$$

$$\begin{array}{|c|c|} \hline a_1 & a_2 \\ \hline a_3 & a_4 \\ \hline \end{array} \mapsto a_1$$

Beispiel (fortgesetzt):

$$\hat{p} = \begin{array}{|c|c|c|c|} \hline \# & \# & \# & \# \\ \hline \# & p_1 & p_2 & \# \\ \hline \# & \# & \# & \# \\ \hline \end{array} \rightsquigarrow \hat{p}' = \begin{array}{|c|c|c|c|c|c|} \hline \# & \# & \# & \# & \# & \# \\ \hline \# & p_1 & p_1 & p_2 & p_2 & \# \\ \hline \# & p_1 & p_1 & p_2 & p_2 & \# \\ \hline \# & \# & \# & \# & \# & \# \\ \hline \# & \# & \# & \# & \# & \# \\ \hline \end{array}$$

$$L(TS) = L(EMSO)$$

- $L(TS) \subseteq L(EMSO)$
- $L(EMSO) \subseteq L(TS)$

$$L(TS) \subseteq L(EMSO)$$

Übersicht:

Um ein TS $\mathcal{T} = (\Sigma, \Gamma, \Theta, \pi)$ durch eine EMSO-Formel auszudrücken, muss

- lokale Sprache $L(\Theta)$ dargestellt werden
 - Blockgröße
 - tiles, die in Θ enthalten sind
- $L = \pi(L(\Theta)) \subseteq \Sigma^{**}$ dargestellt werden

$L(TS) \subseteq L(EMSO)$

- $\mathcal{T} = (\Sigma, \Gamma, \Theta, \pi)$ TS mit $L = L(\mathcal{T})$ und
 - $\Gamma = \Sigma \times Q$
 - $\pi : \Sigma \times Q \rightarrow \Sigma$
- $p' \in L(\Theta)$ ist von der Form $p' = p \times c$ (p und c sind pictures derselben Größe)
- Die lokale Sprache von $\mathcal{T}$ wird in der Komponente Q kodiert
 - $Q = \{q_1, \dots, q_k\}$ „lokales Alphabet“
 - Verwende Mengenvariablen $X_1, \dots, X_k$, um Q zu kodieren:
 $X_I(x)$ bedeutet $c(x) = q_I$

$p \in L \Leftrightarrow$ es gibt ein passendes picture c mit $p' = p \times c, p' \in L(\Theta)$
 $\Leftrightarrow$ es gibt ein passendes picture c mit $p' = p \times c, B_{2,2}(\hat{p}') \subseteq \Theta$

$L(TS) \subseteq L(EMSO)$ (fortgesetzt)

$p \in L \Leftrightarrow$ es gibt ein passendes picture c mit $p' = p \times c$, $B_{2,2}(\hat{p}') \subseteq \Theta$
 $\Leftrightarrow \exists X_1 \dots \exists X_k (\varphi_{\text{Partition}}(X_1, \dots, X_k))$ und das picture c wird
 kodiert als $c(i, j) = q_l \Leftrightarrow (i, j) \in X_l$ und für $p' = p \times c$,
 $B_{2,2}(\hat{p}') \subseteq \Theta$

- $\varphi_{\text{Partition}}(X_1, \dots, X_k)$ wie in AAT
- $B_{2,2}(\hat{p}')$

► Format eines 2×2 Blocks:

$$\Theta = \Theta_m \cup \Theta_t \cup \Theta_b \cup \Theta_l \cup \Theta_r \cup \Theta_{tl} \cup \Theta_{tr} \cup \Theta_{bl} \cup \Theta_{br}$$

x_1	x_2
x_3	x_4

$$\psi_m(x_1, x_2, x_3, x_4) = x_1 S_1 x_3 \wedge x_1 S_2 x_2 \wedge x_3 S_2 x_4 \wedge x_2 S_1 x_4$$

#	#
x_3	x_4

$$\psi_t(x_3, x_4) = x_3 S_2 x_4 \wedge \neg \exists x (x S_1 x_3) \wedge \neg \exists x (x S_1 x_4)$$

⋮

$L(TS) \subseteq L(EMSO)$ (fortgesetzt)

- $B_{2,2}(\hat{p}')$

- Format eines 2×2 Blocks ✓

- Inhalt eines 2×2 Blocks

θ_1	θ_2
θ_3	θ_4

mit θ_i von der Form (a, q_i) :

$$\varphi_{\theta_i}(x) = P_a(x) \wedge X_i(x) \text{ für } i \in \{1, 2, 3, 4\}$$

$L(TS) \subseteq L(EMSO)$ (fortgesetzt)

- $B_{2,2}(\hat{p}')$

➤ Format eines 2×2 Blocks ✓

➤ Inhalt eines 2×2 Blocks

θ_1	θ_2
θ_3	θ_4

mit θ_i von der Form (a, q_i) :

$$\varphi_{\theta_i}(x) = P_a(x) \wedge X_i(x) \text{ für } i \in \{1, 2, 3, 4\}$$

➤ tile („Format und Inhalt“)

$$\chi_m : \underbrace{\psi_m(x_1, x_2, x_3, x_4)}_{\text{Format: } \begin{array}{|c|c|} \hline x_1 & x_2 \\ \hline x_3 & x_4 \\ \hline \end{array}} \rightarrow \bigvee \left(\begin{array}{|c|c|} \hline \theta_1 & \theta_2 \\ \hline \theta_3 & \theta_4 \\ \hline \end{array} \in \Theta_m \right) \bigwedge_{i=1}^4 \varphi_{\theta_i}(x_i)$$

$$\chi_t : \underbrace{\psi_t(x_3, x_4)}_{\text{Format: } \begin{array}{|c|c|} \hline \# & \# \\ \hline x_3 & x_4 \\ \hline \end{array}} \rightarrow \bigvee \left(\begin{array}{|c|c|} \hline \# & \# \\ \hline \theta_3 & \theta_4 \\ \hline \end{array} \in \Theta_t \right) (\varphi_{\theta_3}(x_3) \wedge \varphi_{\theta_4}(x_4))$$

⋮

$L(TS) \subseteq L(EMSO)$ (fortgesetzt)

Erinnerung:

$p \in L \Leftrightarrow \exists X_1 \dots \exists X_k (\varphi_{\text{Partition}}(X_1, \dots, X_k))$ und das picture c wird kodiert als $c(i, j) = q_l \Leftrightarrow (i, j) \in X_l$ und für $p' = p \times c$,
 $B_{2,2}(\hat{p}') \subseteq \Theta$

Mit den eben definierten Formeln lässt sich eine EMSO-Formel angeben, die L definiert.

$$\exists X_1 \dots \exists X_k (\varphi_{\text{Partition}} \wedge \forall x_1 \dots \forall x_4 (\chi_m \wedge \chi_t \wedge \chi_b \wedge \chi_l \wedge \chi_r \wedge \chi_{tl} \wedge \chi_{tr} \wedge \chi_{br} \wedge \chi_{bl}))$$

verbleibende Inklusionen (ohne Beweis):

- $L(EMSO) \subseteq L(TS)$
- $L(PCFRE) \subseteq L(TS)$
- $L(TS) \subseteq L(PCFRE)$

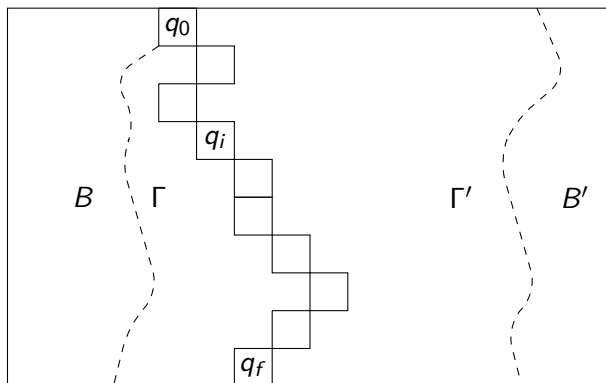
Leerheitsproblem

Satz

Das Leerheitsproblem für REC ist unentscheidbar.

Beweis (Idee)

- TS simuliert TM $\mathcal{M}$ auf Eingabe w
- Konfiguration der TM stellt Zeile in picture p_w dar
- Akzeptierender Lauf von $\mathcal{M}$ simulierbar durch endliche Menge von tiles Θ_w .
- Die Menge aller akzeptierenden Läufe ist kodierbar durch Menge von tiles $\Theta_{\mathcal{M}}$
- $L(\Theta) = \emptyset \Leftrightarrow L(\mathcal{M}) = \emptyset$



- i -te Konfiguration hat die Form $w_i = uqv'$
- eine Zeile hat die Form $B^*\Gamma^*q\Gamma'^*B'^*$, $q \in Q_M$

Zusammenfassung

- REC wird erkannt durch
 - Tiling-Systeme
 - 2OTAs
 - EMSO-Formeln
 - Domino-Systeme
 - PCFRE
- REC ist abgeschlossen unter Projektionen, $\oplus, \ominus, \oplus^*, \ominus^*, \cup, \cap$
- REC ist nicht unter Komplement abgeschlossen.
- Das Leerheitsproblem für REC ist unentscheidbar.