Concurrency theory

2023/24

Lecture 4

Automata recognizing regular trace languages

Distributed alphabet $\Sigma_1, \dots, \Sigma_n$

$\Sigma = \Sigma_1 \cup \dots \cup \Sigma_n$

$D = \Sigma_1^2 \cup \dots \cup \Sigma_n^2 \subseteq \Sigma^2$

Monoid $\Sigma_1^* \times \dots \times \Sigma_n^*$

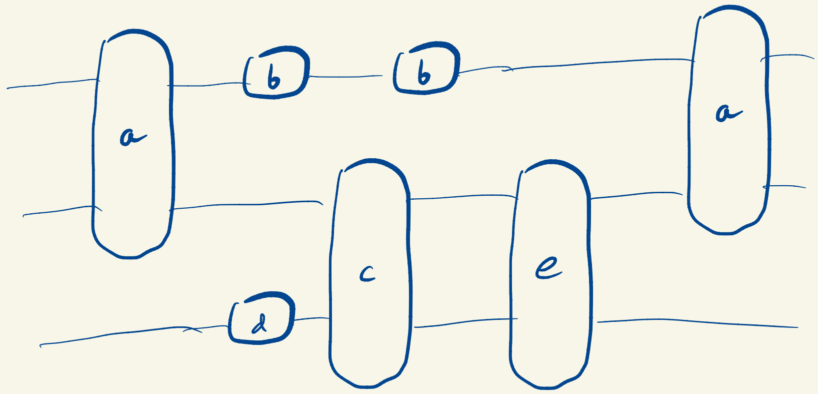
pointwise concatenation

Example:

$\{a,b,c,d,e\} = \{a,b\} \cup \{a,c,e\} \cup \{c,d,e\}$

(abba, acea, dce)

historia



~~(abba, cea, dce)~~

~~(abba, aeca, dce)~~

depends only on D?

$H_D \subseteq \Sigma_1^* \times \dots \times \Sigma_n^*$ historie

Lemma: $\Sigma = \Sigma_1 \cup \dots \cup \Sigma_n$, $D = \Sigma_1^2 \cup \dots \cup \Sigma_n^2$

$$(\Sigma^*/D, \cdot) \cong (H_D, \cdot)$$

Ustalmy alfabet rozproszony $\Sigma_1, \dots, \Sigma_n$

Asynchronous product I

$1 \le i \le n$

$$A_i = (Q_i,\ q_{0i},\ F_i,\ \delta_i \subseteq Q_i \times \Sigma_i \times Q_i)$$

$$A = A_1 \times \dots \times A_n = (Q, q_0, F, \delta)$$

$$Q = Q_1 \times \dots \times Q_n$$

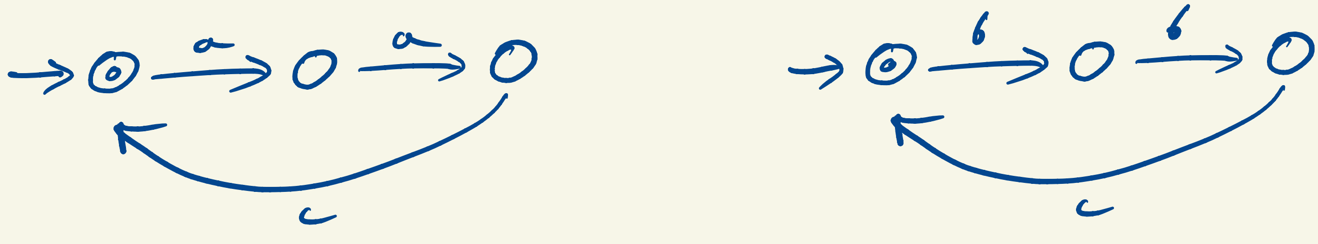
$$q_0 = (q_{01}, \dots, q_{0n})$$

$$F = F_1 \times \dots \times F_n$$

$$\delta(q_1 \dots q_n, a, q_1' \dots q_n') \iff$$

$$\forall i \begin{cases} a \in \Sigma_i \Rightarrow \delta_i(q_i, a, q_i') \\ a \notin \Sigma_i \Rightarrow q_i = q_i' \end{cases}$$

Example: $\Sigma_1 = \{a, c\}$ $\quad \Sigma_2 = \{b, c\}$



$$((aa \parallel bb)\, c)^*$$

Question: which languages are recognised?

Lemma: Fix $\Sigma_1, \dots, \Sigma_n$.
Asynchronous products I recognise exactly rectangular languages:

$$\underbrace{\left(\forall i \quad \pi_i(w) \in \pi_i(L)\right)}_{w \in \mathrm{rect}(L)} \Rightarrow w \in L$$

$\mathrm{rect}(L) = L$

$$\left(\forall i \quad \exists u_i \in L \quad \pi_i(w) = \pi_i(u_i)\right) \Rightarrow w \in L$$

Przykład: $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$

$a \parallel b \quad \cup \quad aa \parallel bb$

is not rectangular, $w = abb$

both components are

Asynchronous product II

$$F = \cancel{F_1 \times \ldots \times F_n} \qquad F \subseteq Q_1 \times \ldots \times Q_n$$

Question: which languages are recognised?

Lemma: Fix $\Sigma_1, \ldots, \Sigma_n$.

Asynchronous products I recognise exactly finite unions of rectangular languages

Example: $\Sigma_1 = \{a, c\}$ $\quad \Sigma_2 = \{b, c\}$

$$\big((a \parallel b \;\cup\; aa \parallel bb)\, c \big)^*$$

Asynchronous automaton:

$$\Sigma_1^2 \cup \dots \cup \Sigma_n^2 = D \quad (\subseteq D)$$

$$A = (\Sigma_1, \dots, \Sigma_n, Q_1, \dots, Q_n, q_0 \in Q = Q_1 \times \dots \times Q_n, F \subseteq Q, (\delta_a)_{a \in \Sigma})$$

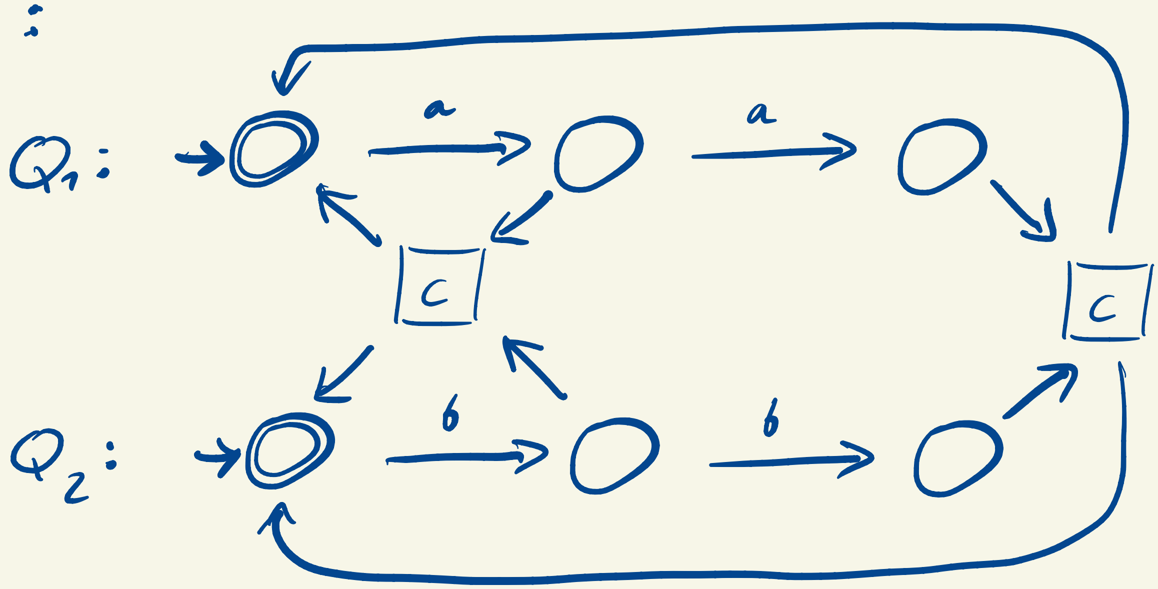
- $dom(a) = \{ i : a \in \Sigma_i \}$
- $\delta_a \subseteq \prod_{i \in dom(a)} Q_i \times \prod_{i \in dom(a)} Q_i$

$$\delta \subseteq Q \times \Sigma \times Q$$

$$\delta(q_1 \dots q_n, a, q_1' \dots q_n') \Longleftrightarrow$$

- $\delta_a(q_1 \dots q_n|_{dom(a)}, q_1' \dots q_n'|_{dom(a)})$
- $q_1 \dots q_n|_{\overline{dom(a)}} = q_1' \dots q_n'|_{\overline{dom(a)}}$

Example:



Theorem: Fix (Σ, D).

Asynchronous automata recognise exactly regular trace languages.

any distributed alphabet corresponding to (Σ, D)

Proof: [assuming that D is acyclic]

ordered tree

(minimal) DFA

$\Big\downarrow$ states P

deterministic asynchronous automaton

$$
\begin{tikzcd}
 & {} \arrow[d, no head, "f : P \to P"] & \\
 & a \arrow[dl, no head, "g"'] \arrow[d, no head, "h"] \arrow[dr, no head, "k"] & \\
b & c & d
\end{tikzcd}
\qquad f_a : P \longrightarrow P
$$

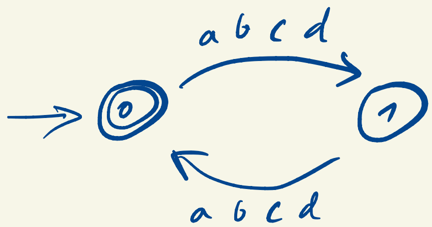
$$(f, g, h, k) \longmapsto (f\, g\, h\, k\, f_a, Id, Id, Id)$$

Invariant: after reading w, composition of all functions $P \to P$ (pre-order) yields f_w

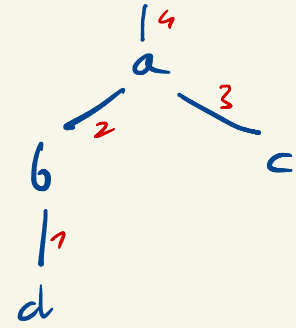
Acceptance: composition(initial state) $\in$ final states

Example : $\Sigma = \{a, b, c, d\}$

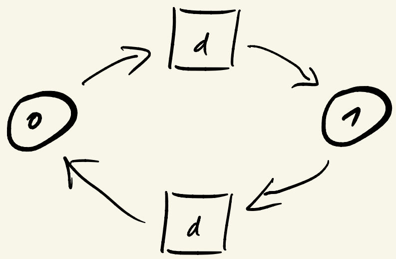
L = even length



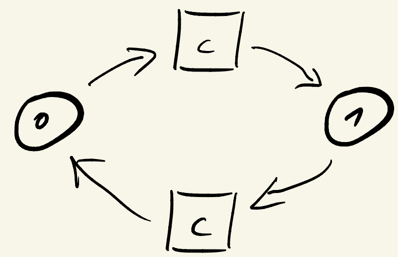
$D =$



1 :

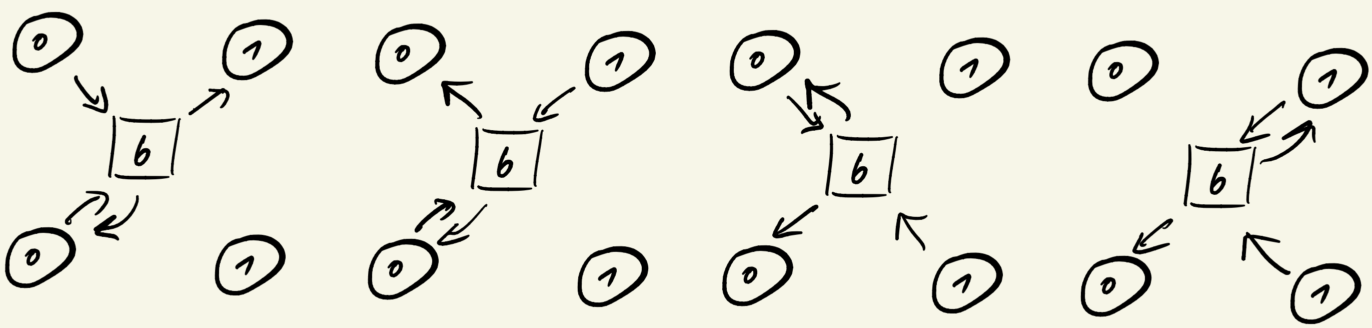


3 :



2 :

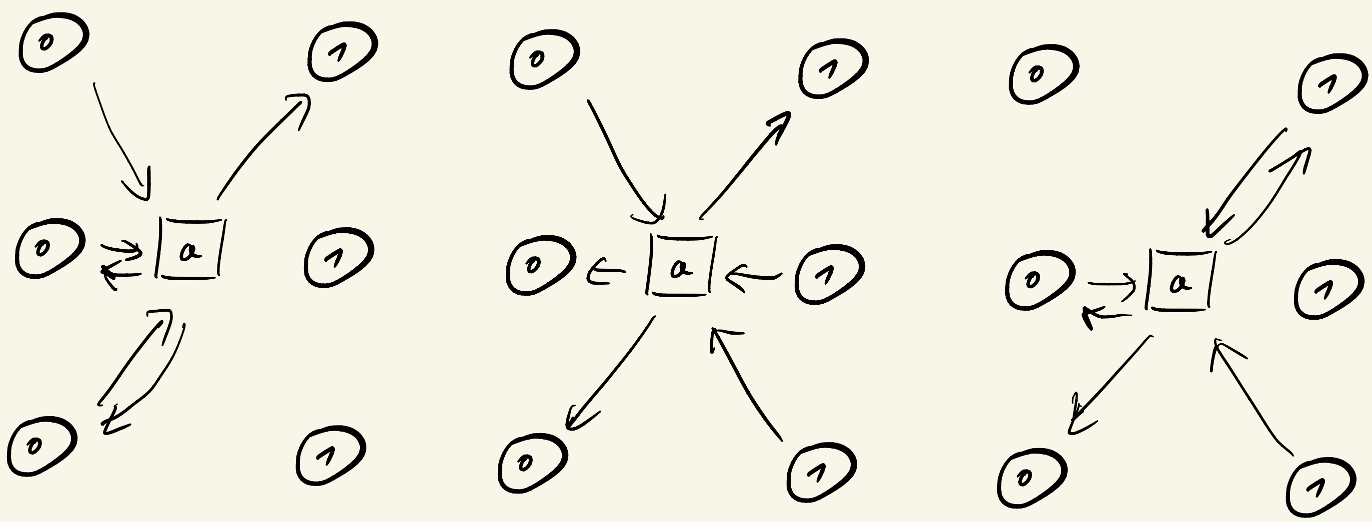
1 :



4 :

3 :

2 :



$$F = \{ (p, q, r, s) : p + q + r + s \equiv 0 \mod 2 \}$$