Synthesizing Asynchronous Automata from Fair Specifications

Béatrice Bérard
Sorbonne Univ.
Paris, France

Benjamin Monmege
Aix-Marseille Univ,
Marseille, France

B. Srivathsan
Chennai Mathematical Institute
India

Arnab Sur

CAALM Workshop '25
Paris

Outline

-1. Asynchronous Automata + Zeilonka's theorem

-2. Fair specifications (DFAs)

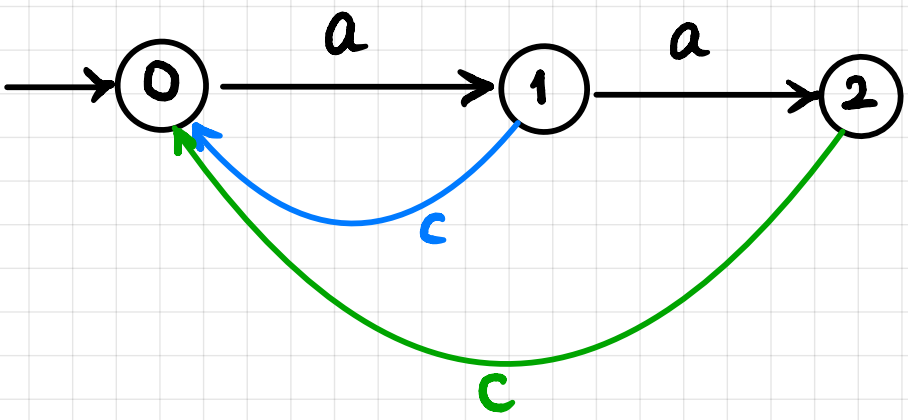
-3. Main construction: Fair DFAs to AA

-4. Generalization: Fair DFAs + acyclic architecture

COMING NEXT: Asynchronous Automata

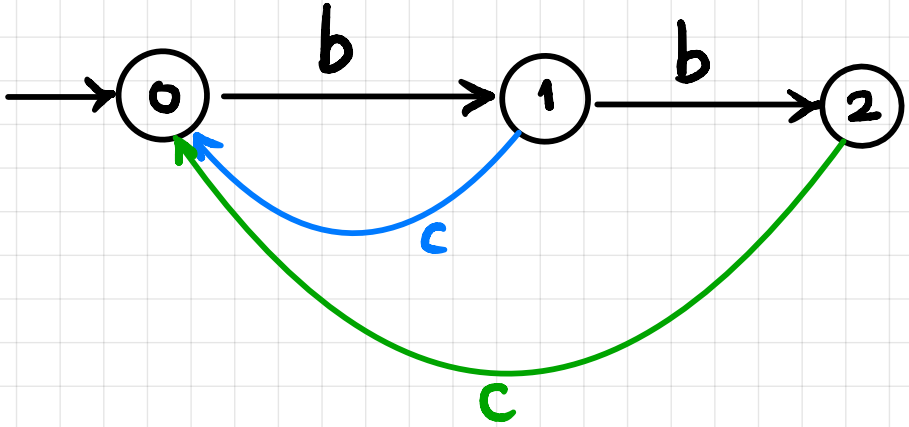
Source: Madhavan's notes!

Process P_1



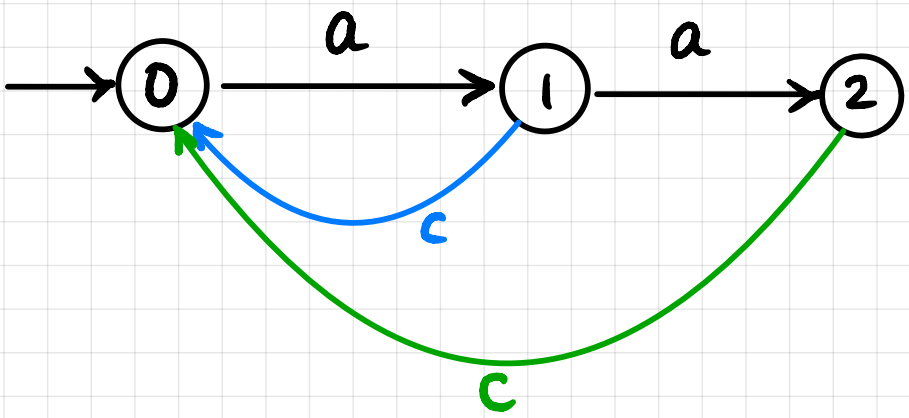
$\Sigma_1 = \{ a, c \}$

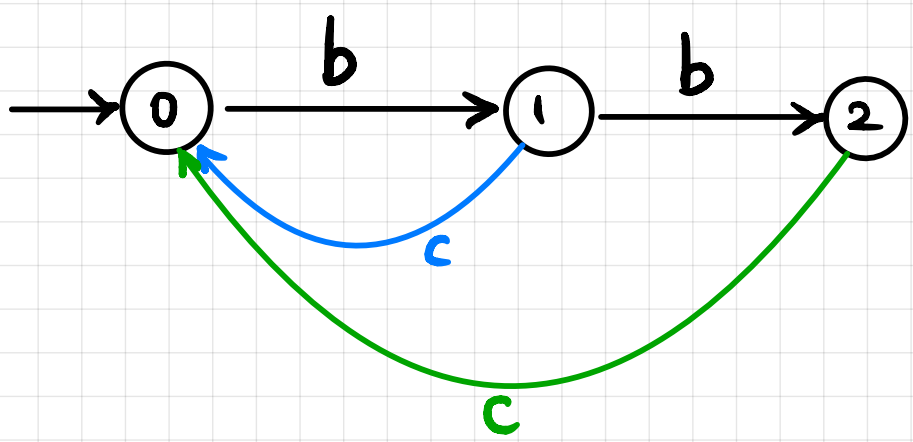
Process P_2



$\Sigma_2 = \{ b, c \}$

Accepting state: $(0, 0)$





Words in the language

a b c

b a c

a b a b c

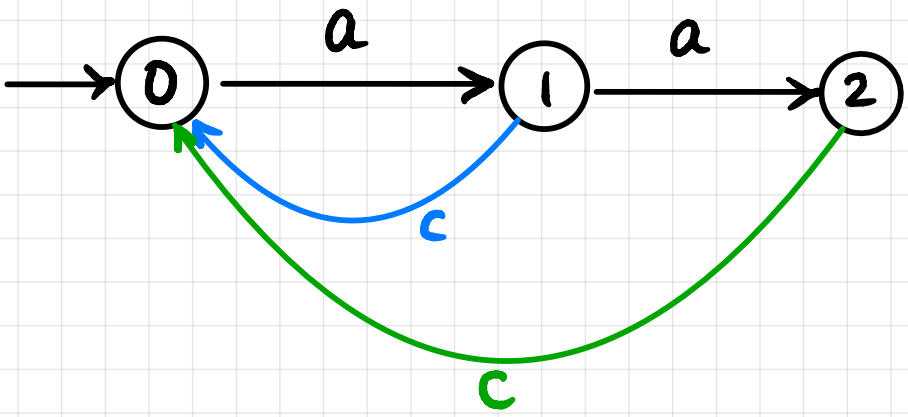
a a b b c

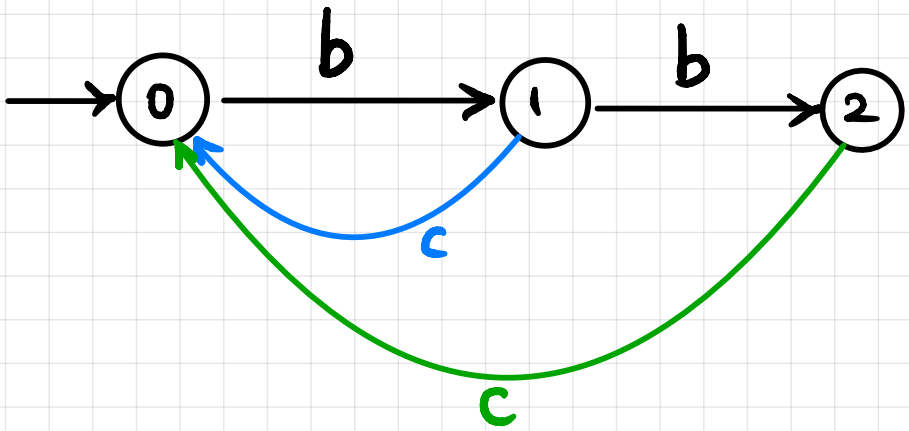
a b c b b a a c

Words not in the language

a c ✗

a a b c ✗



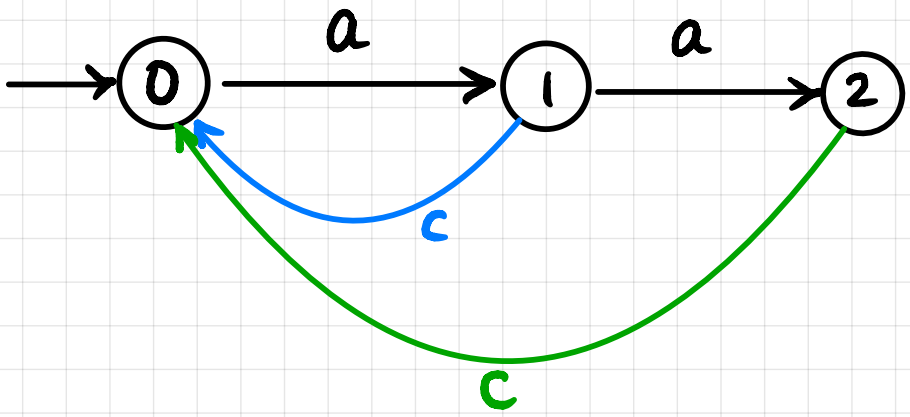


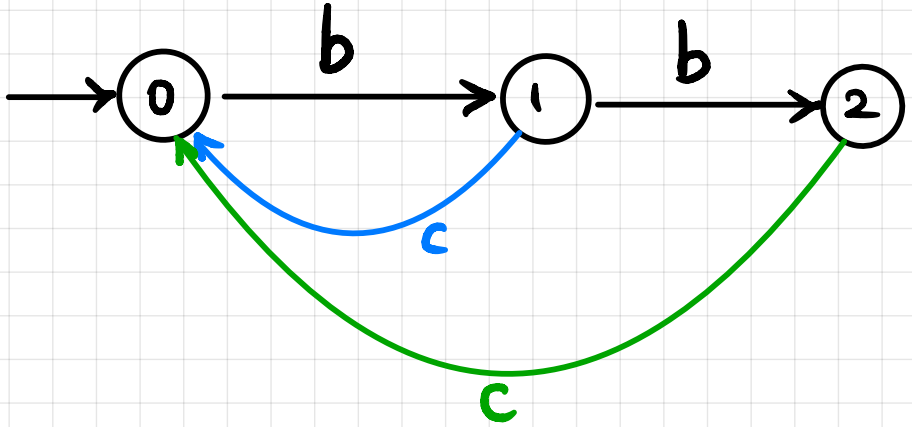
Language: $\left([\text{shuffle}(ab) + \text{shuffle}(aabb)] \cdot c \right)^*$

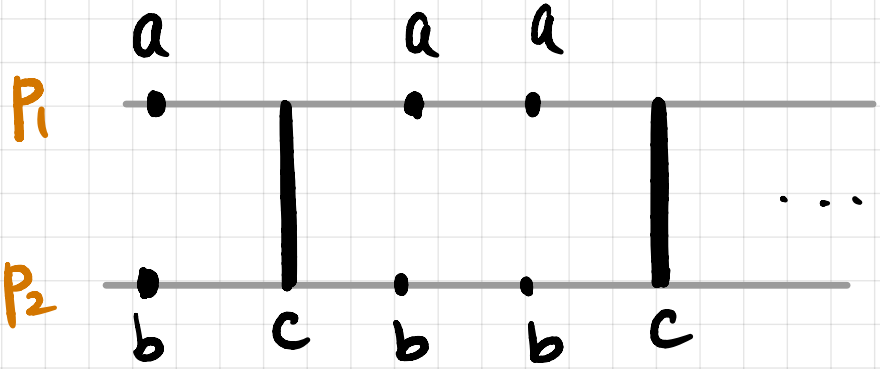
Behaviours of Asynchronous Automata

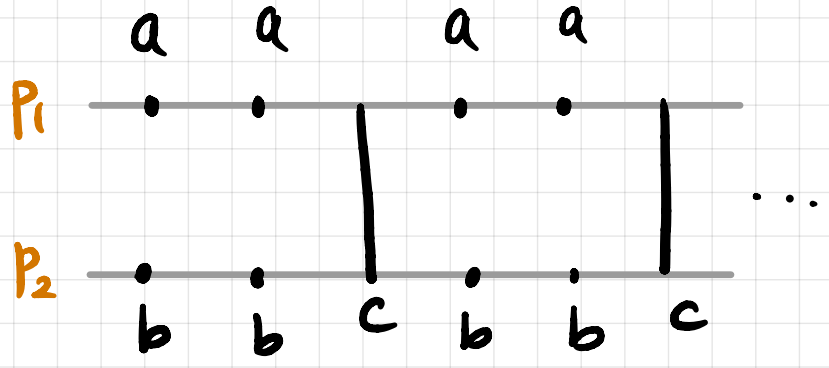
can be understood as certain

partial orders called TRACES



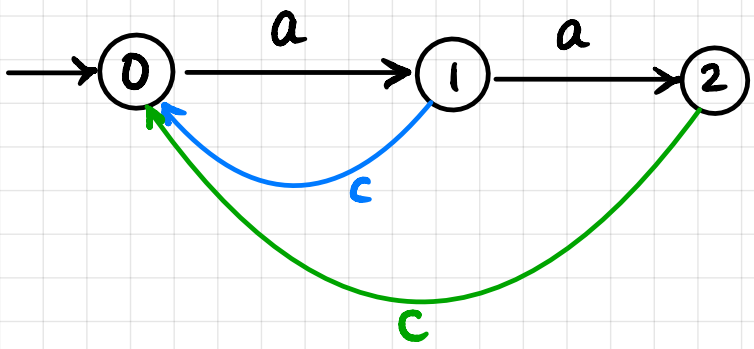


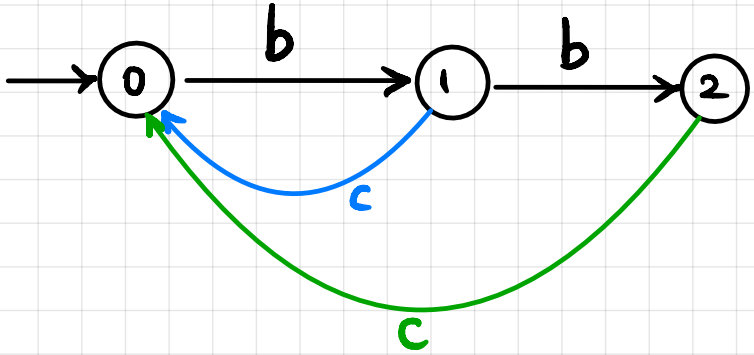


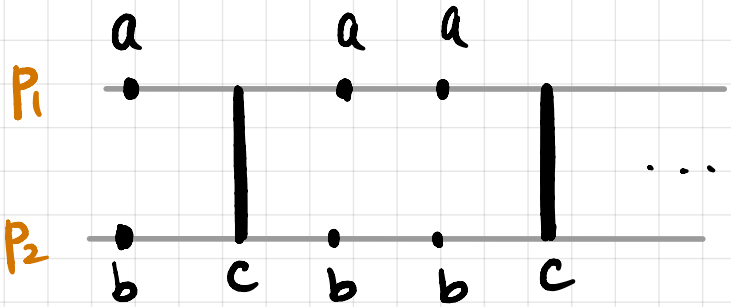


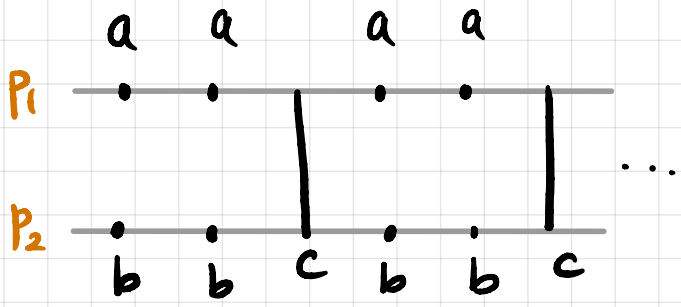
Trace

each trace corresponds to a set of words









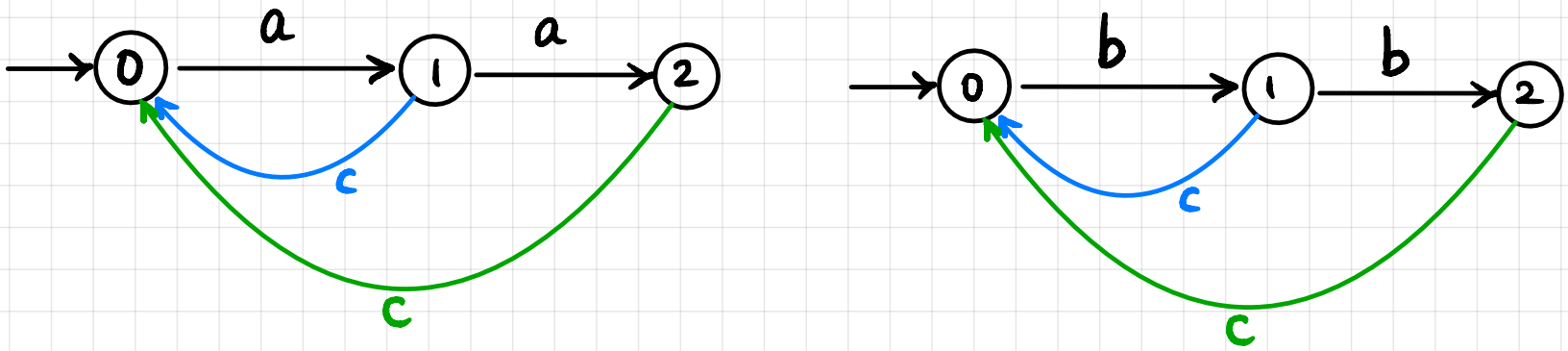
Trace

Language of AA can be seen as a set of traces

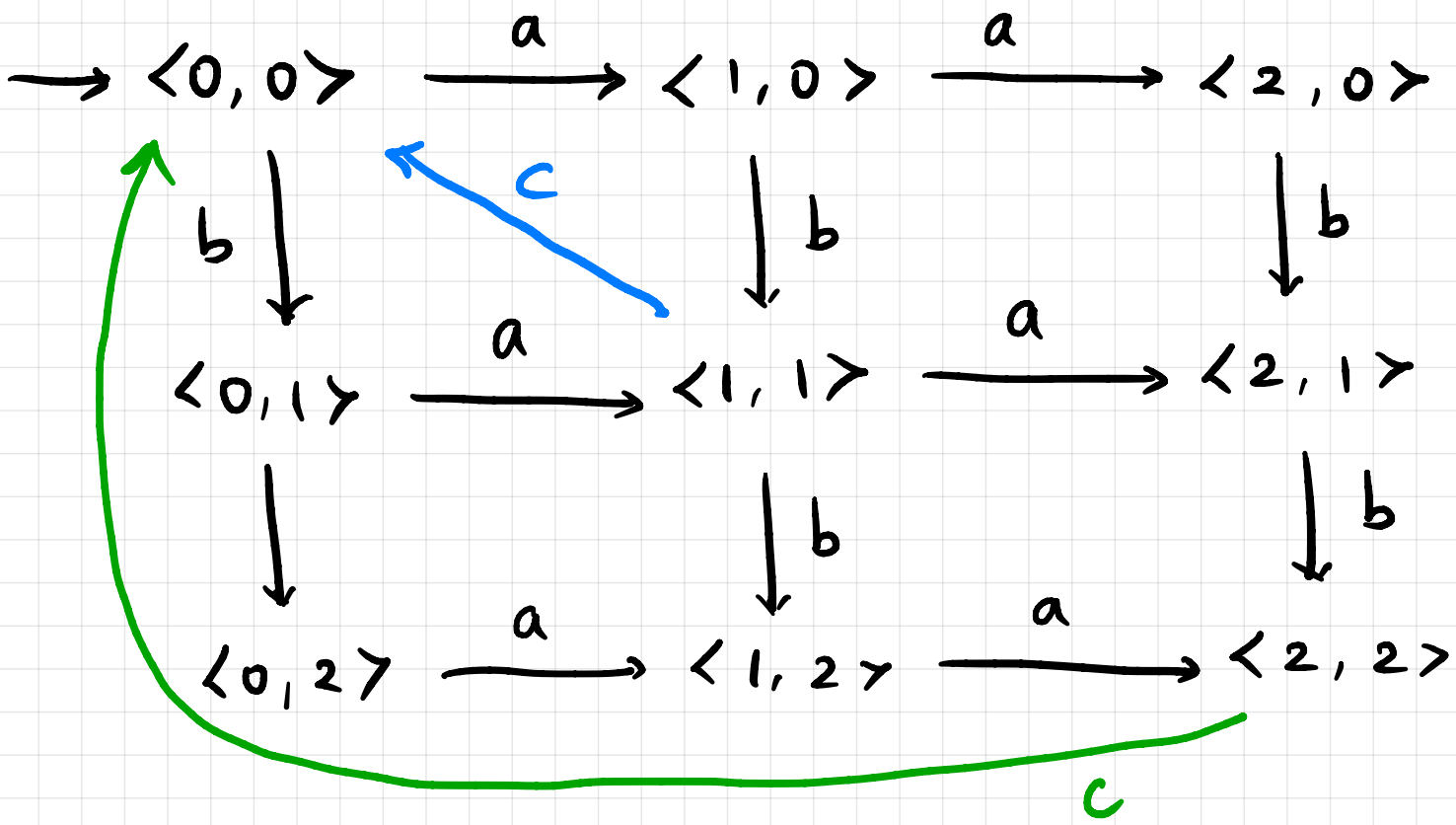
Observation: Language accepted by an AA

is regular

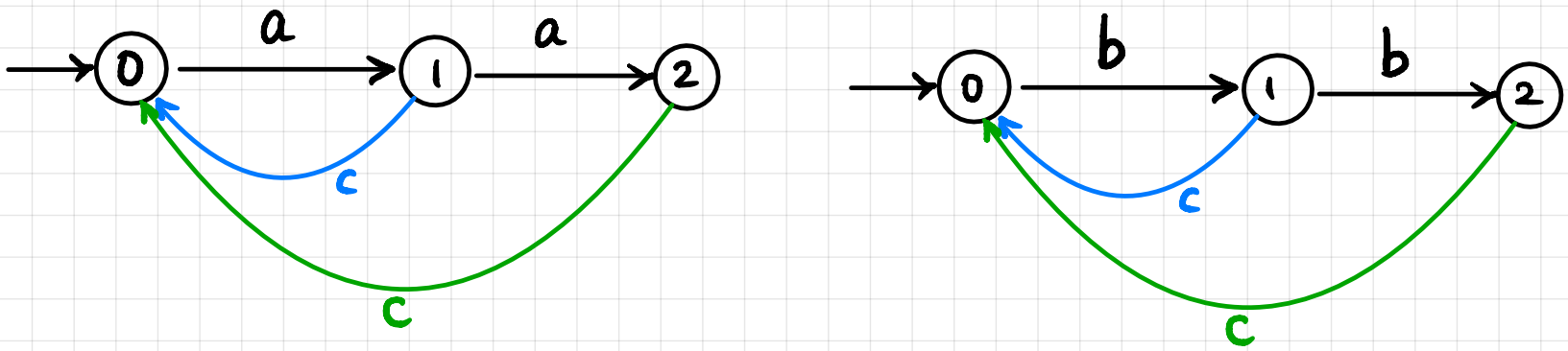
AA:



DFA:

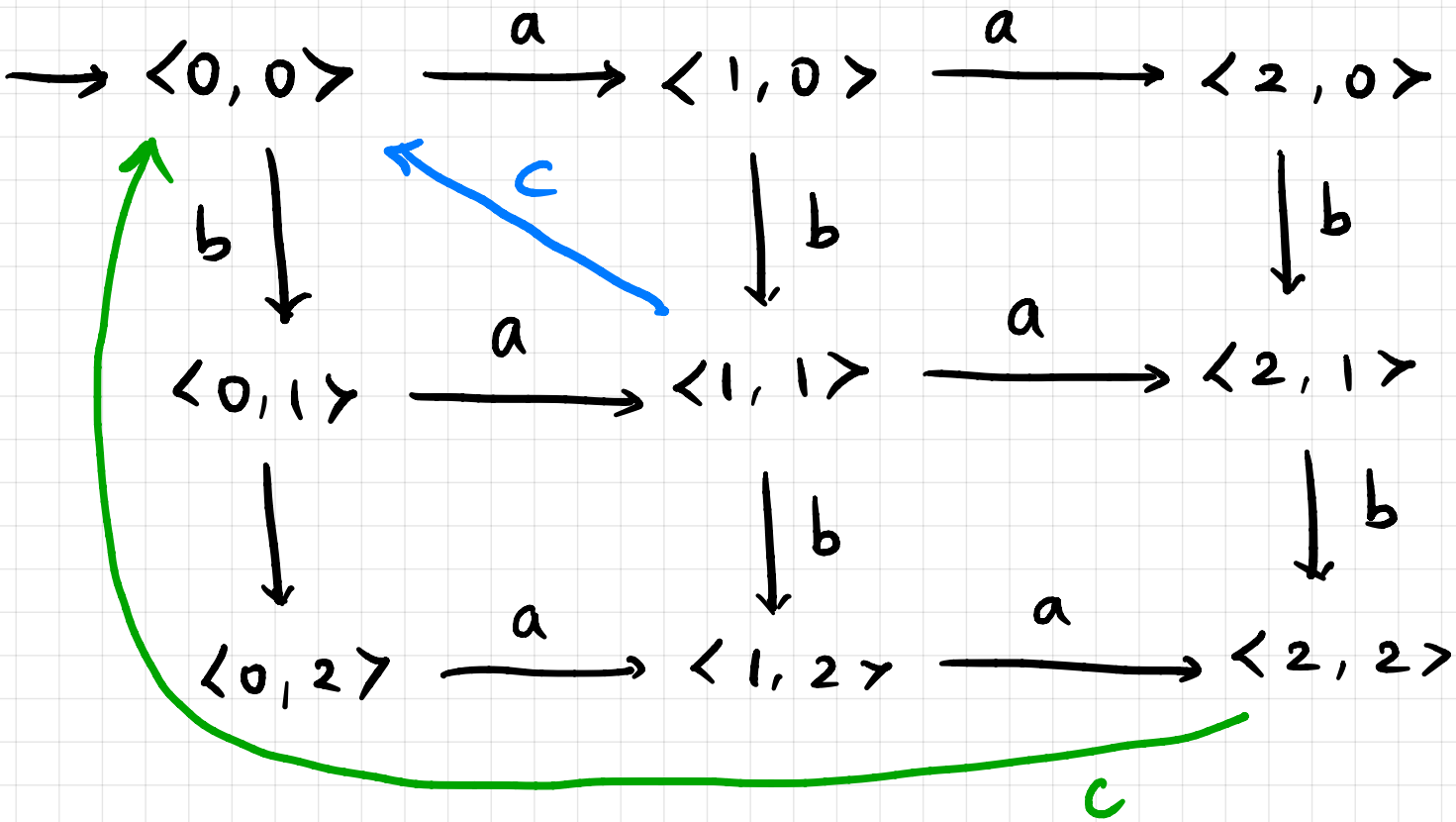


AA:



↑ ?

DFA:

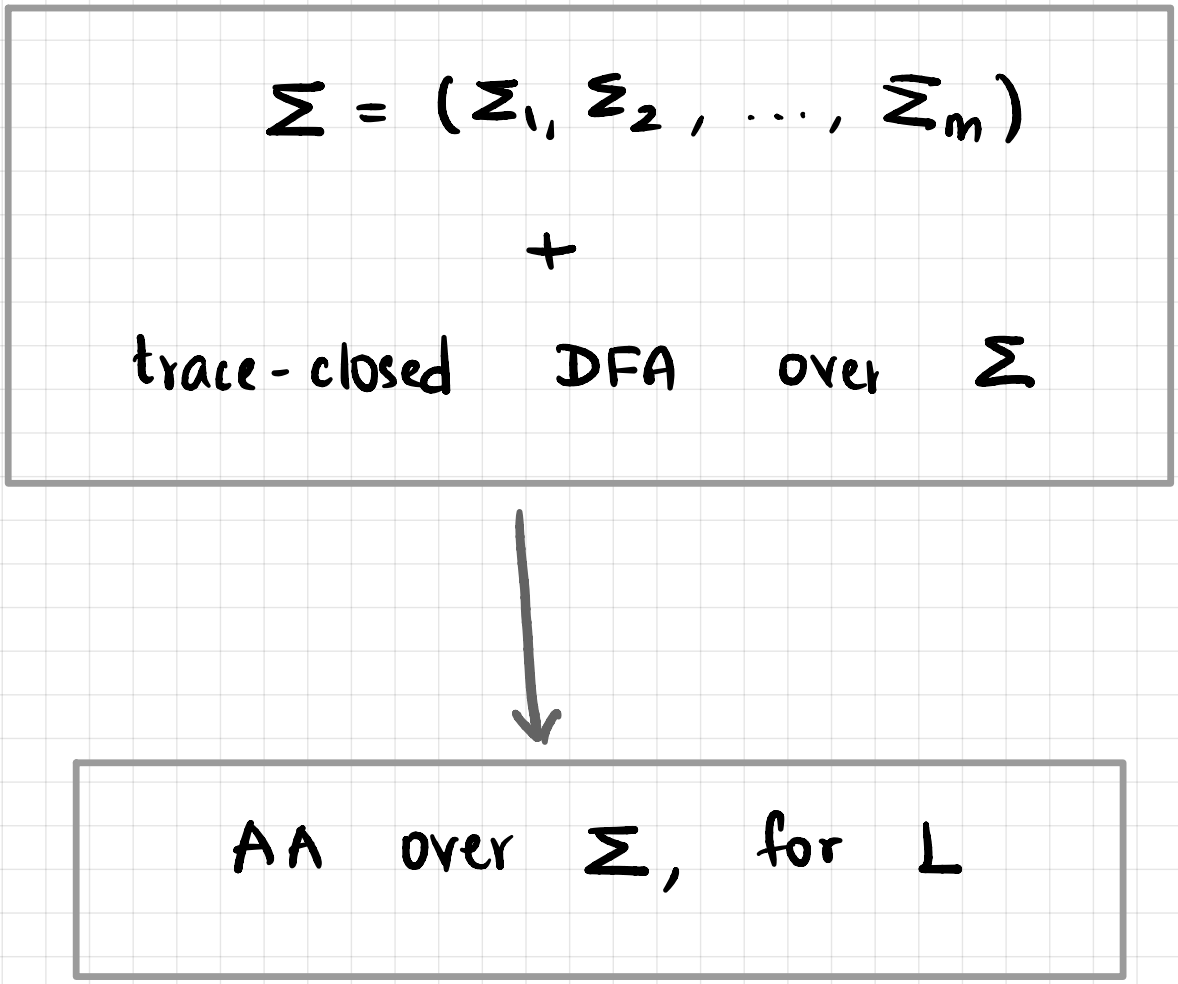


Zielonka's Theorem (1987):

Given a distributed alphabet $\Sigma = (\Sigma_1, \Sigma_2, \ldots, \Sigma_m)$.

For every regular trace-closed language L over Σ

there exists an AA over Σ accepting L



(Global)
Specification

(Distributed)
Implementation

Zielonka's result lays the foundation for synthesizing distributed systems from global specifications

Some history

- Cori, Métivier, Zielonka (1993) : $|\Sigma|^{|\Sigma|^2} \cdot |M|^{2^{|\Sigma|}}$
- Mukund, Sohoni (1994) : $|P|^{|P|^2} \cdot |A|^{|A| \cdot 2^{|P|}}$
- Genest, Muscholl (2006) : $2^{3|P|^3} \cdot |A|^{|A| \cdot |P|^2}$
- Genest, Gimbert, Muscholl, Walukiewicz (2010) : $4^{|P|^2} \cdot |A|^{|P|^2}$

Some history

- Adsul, Gastin, Kulkarni, Weil (2024) : regular trace language as a cascade product of localised AA
- Pighizzini (1993) : synthesize non-deterministic AA
- Baudru (2011) : compositional synthesis of non-dekrministic AA

Our Work

Impose a (fair) restriction on DFAs

&

study the AA synthesis problem

Goal: - conceptually simpler construction

- better complexity

Outline

-1. Asynchronous Automata + Zeilonka's theorem ✓

-2. Fair specifications (DFAs)

-3. Main construction: Fair DFAs to AA

-4. Generalization: Fair DFAs + acyclic architecture

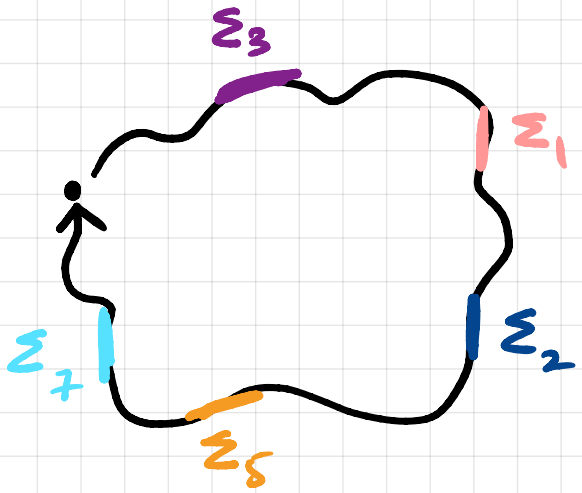
P_1 can do an arbitrary number of actions while starving P_2

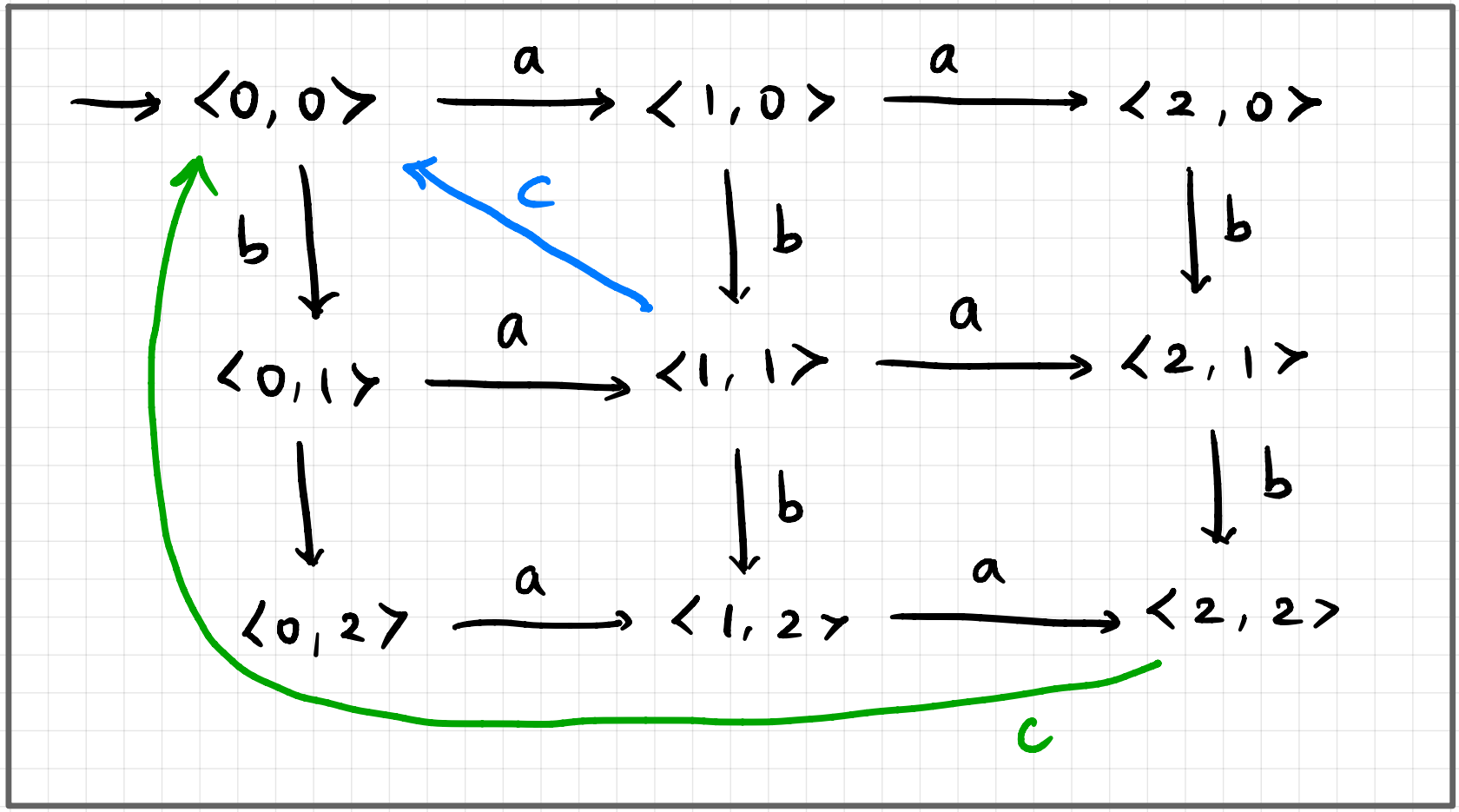
FAIR DFAs:

$\Sigma = (\Sigma_1, \Sigma_2, \ldots, \Sigma_m)$

DFA M

- Each loop of M contains an action from every Σ_i

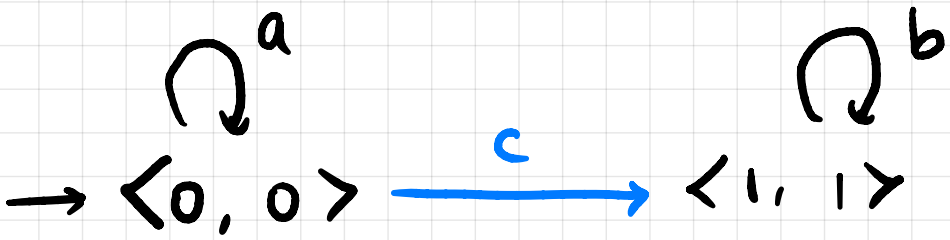




a fair DFA

$\Sigma_1 = \{a, c\}$

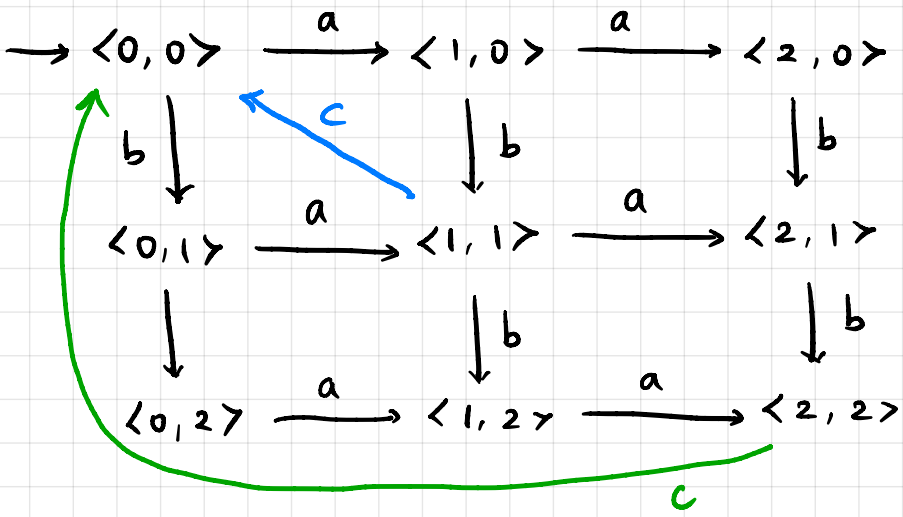
$\Sigma_2 = \{b, c\}$



↳ an unfair DFA

k - fairness :

DFA is k- fair if in every sequence of k- transitions, all processes participate



3 - fair, not 2- fair

For a fair DFA, we can compute the smallest k s.t. it is k-fair

k- fairness :

A language L is k- fair if for every $w \in L$,

every factor of w with length k

contains actions from all processes

$$\Sigma = (\Sigma_1, \Sigma_2, \ldots, \Sigma_n) \qquad \Sigma_i = \{a_i, c\}$$

$$L_n = \left[\bigcup_{1 \le i \le j \le n} (a_i a_j + a_j a_i) . c \right]^*$$

- 3- fair

Note: A DFA M is k-fair iff

$L(M)$ is k-fair

Main Theorem

For k-fair DFAs, an equivalent AA can be synthesized, where the number of states of each local automaton is bounded by:

$$O(n \cdot k \cdot |\Sigma|^{3k-3})$$

Outline

1. Asynchronous Automata + Zielonka's theorem ✓

2. Fair specifications (DFAs) ✓

3. Main construction: Fair DFAs to AA

4. Generalization: Fair DFAs + acyclic architecture

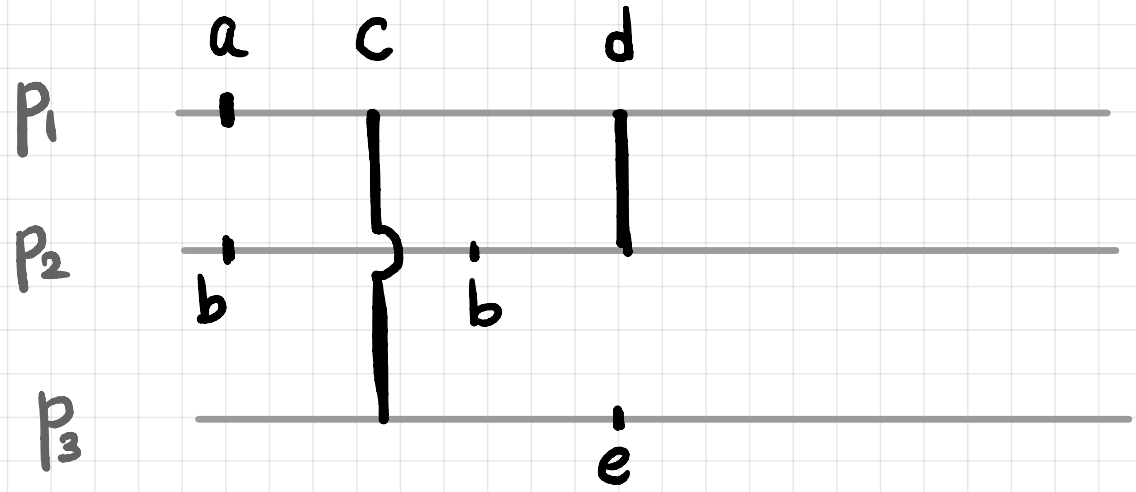
STEPS TO THE CONSTRUCTION

Step 0: Two notations

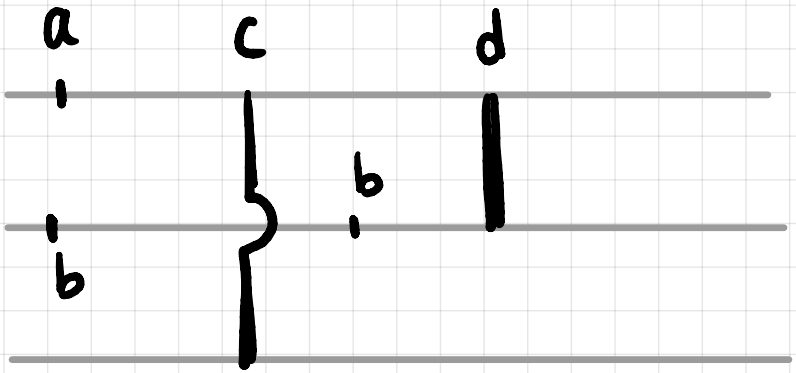
Step 1: An infinite AA maintaining local views

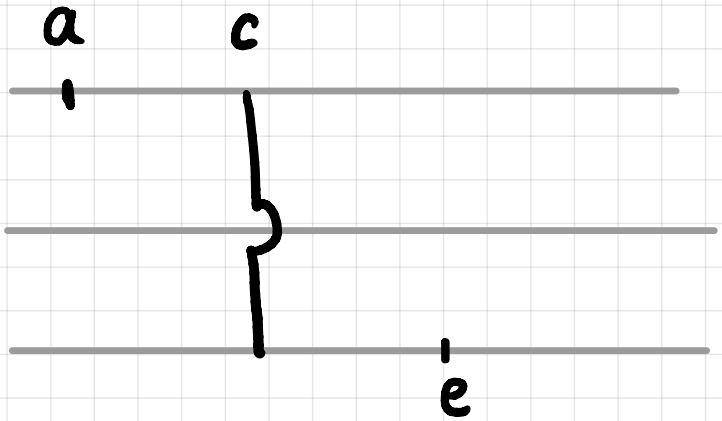
Step 2: A "better" infinite AA that maintains suffixes of views + an unbounded counter

Step 3: Making the construction finite by modulo counting



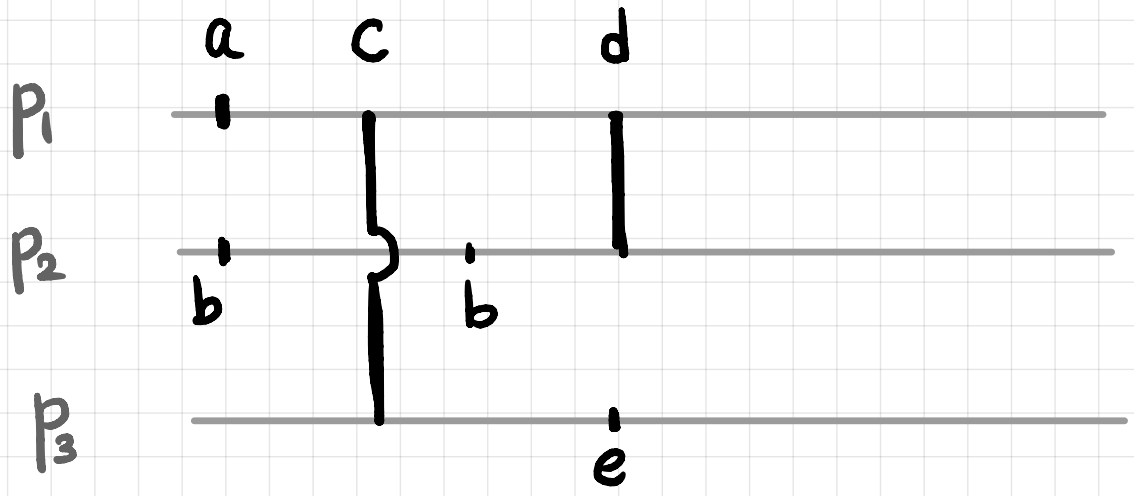
Trace t





$view_{P_1}(t) = view_{P_2}(t)$

$view_{P_3}(t)$



Trace t

Foata Normal form:

$$F(t) = \{a, b\}\ \{c, b\}\ \{d, e\}$$

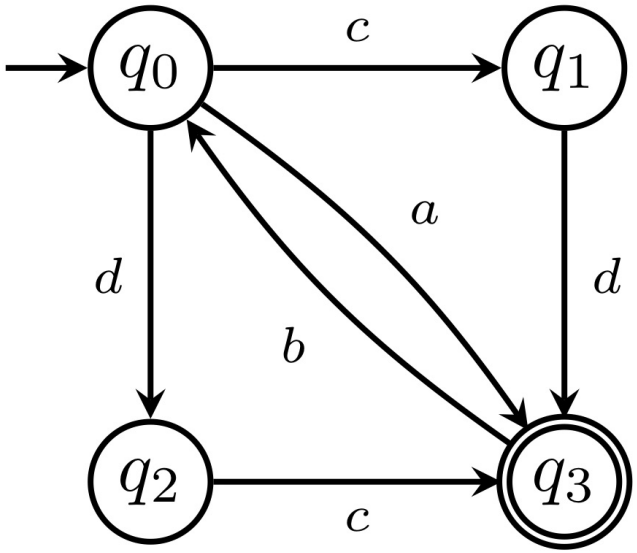
STEPS TO THE CONSTRUCTION

Step 0: Two notations ✓

Step 1: An infinite AA maintaining local views

Step 2: A "better" infinite AA that maintains suffixes of views + an unbounded counter

Step 3: Making the construction finite by modulo counting



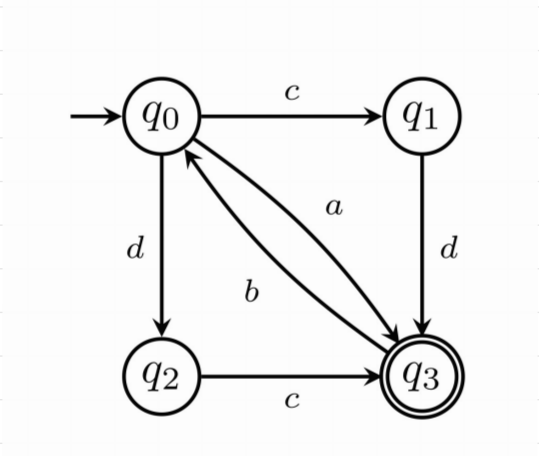
$\Sigma_1 = \{a, b, d\}$

$\Sigma_2 = \{a, c\}$

$\Sigma_3 = \{b, c\}$

DFA M

M is 4-fair, but not 3-fair



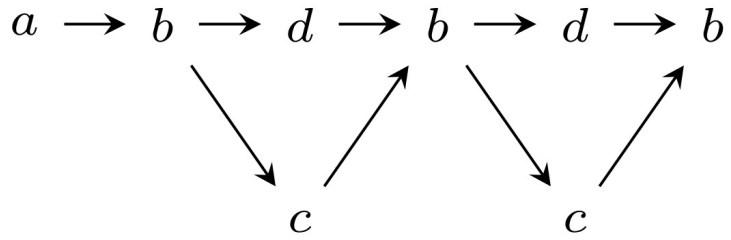
$\Sigma_1 = \{a, b, d\}$

$\Sigma_2 = \{a, c\}$

$\Sigma_3 = \{b, c\}$

Word w : $abcdbdcb$

Trace $[w]$:



$\mathsf{F}([w])$: $\{a\}\ \{b\}\ \{c,d\}\ \{b\}\ \{c,d\}\ \{b\}$

$\Sigma_1 = \{a, b, d\}$

$\Sigma_2 = \{a, c\}$

$\Sigma_3 = \{b, c\}$

$a \rightarrow \{p_1, p_2\}$

$b \rightarrow \{p_1, p_3\}$

$c \rightarrow \{p_2, p_3\}$

---

	a	b	c	d	b
p_1	$\{a\}$	$\{a\}\{b\}$	$\{a\}\{b\}$	$\{a\}\{b\}\{d\}$	$\{a\}\{b\}\{c,d\}\{b\}$
p_2	$\{a\}$	$\{a\}$	$\{a\}\{b\}\{c\}$	$\{a\}\{b\}\{c\}$	$\{a\}\{b\}\{c\}$
p_3	$\{\}$	$\{a\}\{b\}$	$\{a\}\{b\}\{c\}$	$\{a\}\{b\}\{c\}$	$\{a\}\{b\}\{c,d\}\{b\}$

Checking for acceptance:

	a	b	c	d	b
p_1	$\{a\}$	$\{a\}\{b\}$	$\{a\}\{b\}$	$\{a\}\{b\}\{d\}$	$\{a\}\{b\}\{c,d\}\{b\}$
p_2	$\{a\}$	$\{a\}$	$\{a\}\{b\}\{c\}$	$\{a\}\{b\}\{c\}$	$\{a\}\{b\}\{c\}$
p_3	$\{\}$	$\{a\}\{b\}$	$\{a\}\{b\}\{c\}$	$\{a\}\{b\}\{c\}$	$\{a\}\{b\}\{c,d\}\{b\}$

↓

synchronize all views

$\{a\}\ \{b\}\ \{c,d\}\ \{b\}$

↳

run this on the DFA M

STEPS TO THE CONSTRUCTION

Step 0: Two notations ✓

Step 1: An infinite AA maintaining local views ✓

Step 2: A "better" infinite AA that maintains suffixes of views + an unbounded counter

Step 3: Making the construction finite by modulo counting

Using k-fairness we can show:

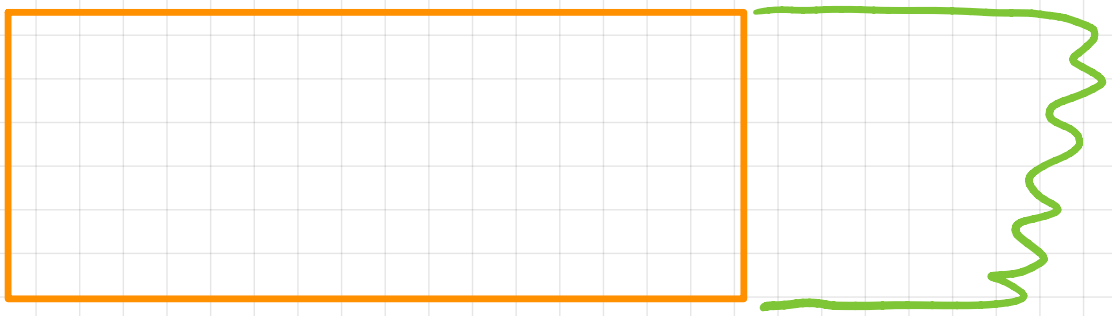
if the local view of a process p' contains more than $2k-2$ actions

$F(\text{view}_p(t))$

Let j := largest index s.t. $F()_j\ F()_{j+1} \cdots$ contains $2k-2$ elements.

The prefix $F[1 \ldots j-1]$ is present in $\text{view}_{p'}(t)\ \forall p'$

$view_p(t)$:



$view_{p'}(t)$:

It is not useful to maintain the "orange" prefix.

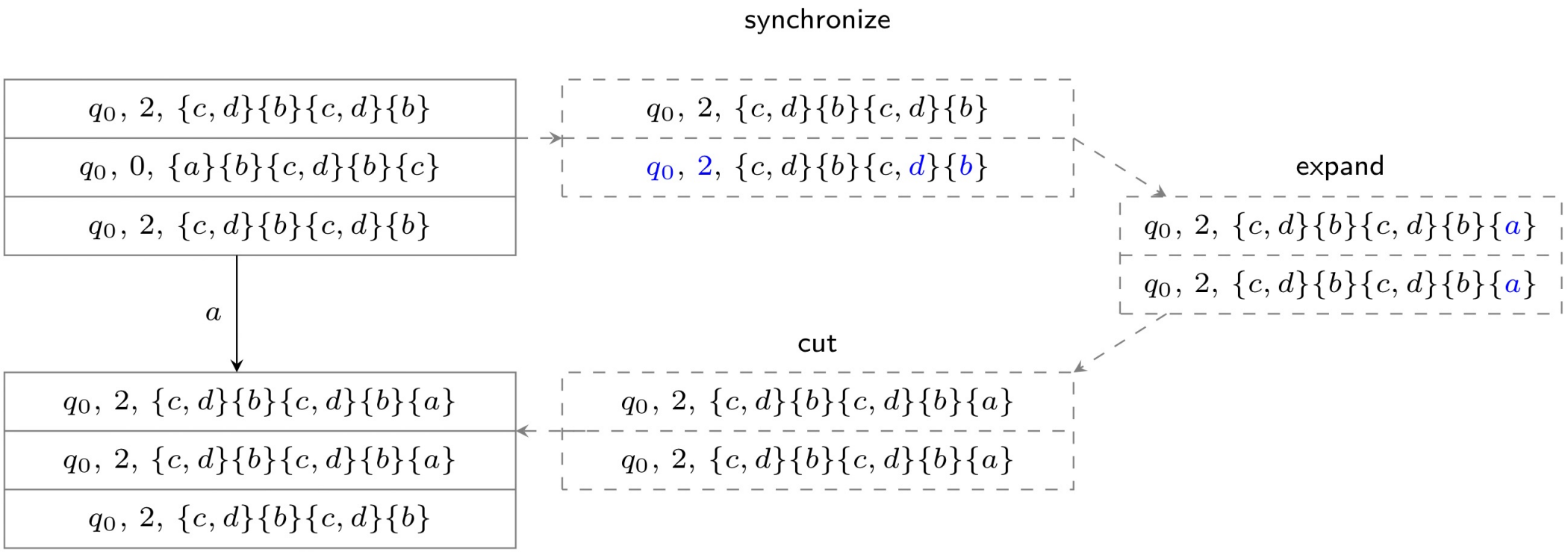
b

$\{a\}\{b\}\{c,d\}\{b\}\{c,d\}\{b\}$
$\{a\}\{b\}\{c,d\}\{b\}\{c\}$
$\{a\}\{b\}\{c,d\}\{b\}\{c,d\}\{b\}$

$k = 4$

$2k - 2 = 6$

p_1	$q_0, 2, \{c,d\}\{b\}\{c,d\}\{b\}$
p_2	$q_0, 0, \{a\}\{b\}\{c,d\}\{b\}\{c\}$
p_3	$q_0, 2, \{c,d\}\{b\}\{c,d\}\{b\}$


synchronize
$q_0,\ 2,\ \{c,d\}\{b\}\{c,d\}\{b\}$
$q_0,\ 0,\ \{a\}\{b\}\{c,d\}\{b\}\{c\}$
$q_0,\ 2,\ \{c,d\}\{b\}\{c,d\}\{b\}$
a
$q_0,\ 2,\ \{c,d\}\{b\}\{c,d\}\{b\}\{a\}$
$q_0,\ 2,\ \{c,d\}\{b\}\{c,d\}\{b\}\{a\}$
$q_0,\ 2,\ \{c,d\}\{b\}\{c,d\}\{b\}$
$q_0,\ 2,\ \{c,d\}\{b\}\{c,d\}\{b\}$
$q_0,\ 2,\ \{c,d\}\{b\}\{c,d\}\{b\}\{b\}$
expand
$q_0,\ 2,\ \{c,d\}\{b\}\{c,d\}\{b\}\{a\}$
$q_0,\ 2,\ \{c,d\}\{b\}\{c,d\}\{b\}\{a\}$
cut
$q_0,\ 2,\ \{c,d\}\{b\}\{c,d\}\{b\}\{a\}$
$q_0,\ 2,\ \{c,d\}\{b\}\{c,d\}\{b\}\{a\}$

Checking for acceptance:

- synchronize all views
- to get $(q, v,$ suffix of trace$)$
- Run the suffix on the DFA starting from q.

STEPS TO THE CONSTRUCTION

Step 0: Two notations ✓

Step 1: An infinite AA maintaining local views ✓

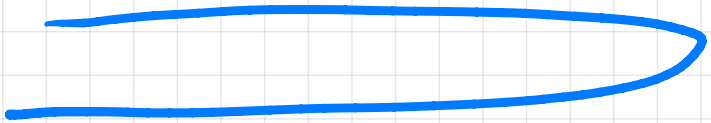
Step 2: A "better" infinite AA that maintains ✓
suffixes of views + an unbounded counter

Step 3: Making the construction finite by modulo counting

Using k- fairness, we can show:

$view_p(t)$

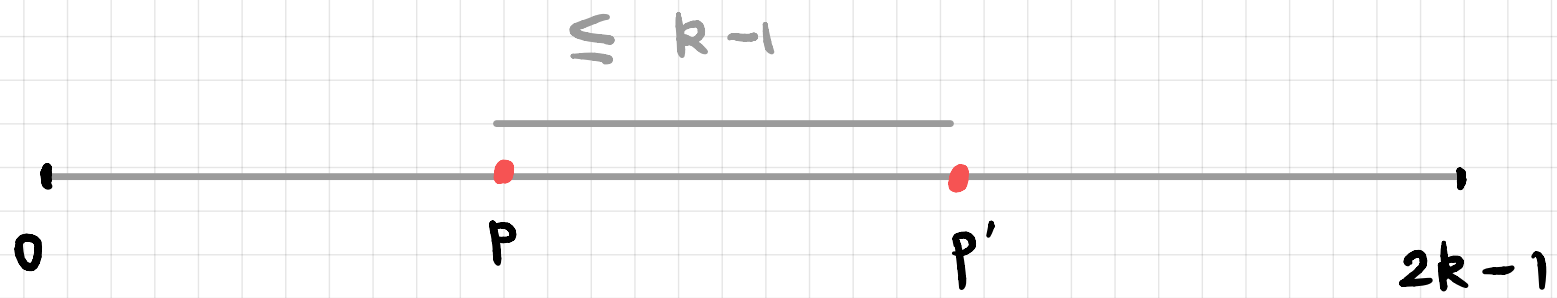
$view_{p'}(t)$



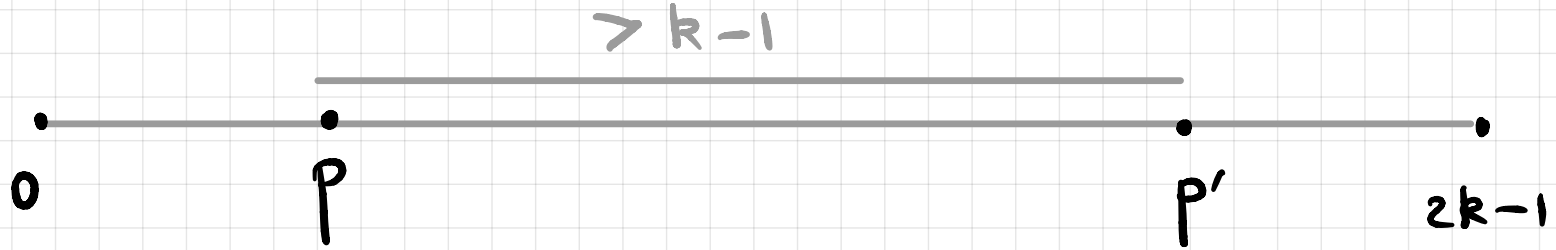
$$\left| \, |view_p(t)| - |view_{p'}(t)| \, \right| \leq k-1$$

$$\left| \, |view_p(t)| - |view_{p'}(t)| \, \right| \leq k-1$$

Use this observation to count modulo $2k$



P' is ahead of P



P is ahead of P'

STEPS TO THE CONSTRUCTION

Step 0: Two notations ✓

Step 1: An infinite AA maintaining local views ✓

Step 2: A "better" infinite AA that maintains ✓
suffixes of views + an unbounded counter

Step 3: Making the construction finite by modulo counting ✓

Main Theorem

For k-fair DFAs, an equivalent AA can be synthesized, where the number of states of each local automaton is bounded by:

$$O(n \cdot k \cdot |\Sigma|^{3k-3})$$

+ a matching lower bound

Outline

1. Asynchronous Automata + Zeilonka's theorem ✓

2. Fair specifications (DFAs) ✓

3. Main construction: Fair DFAs to AA ✓

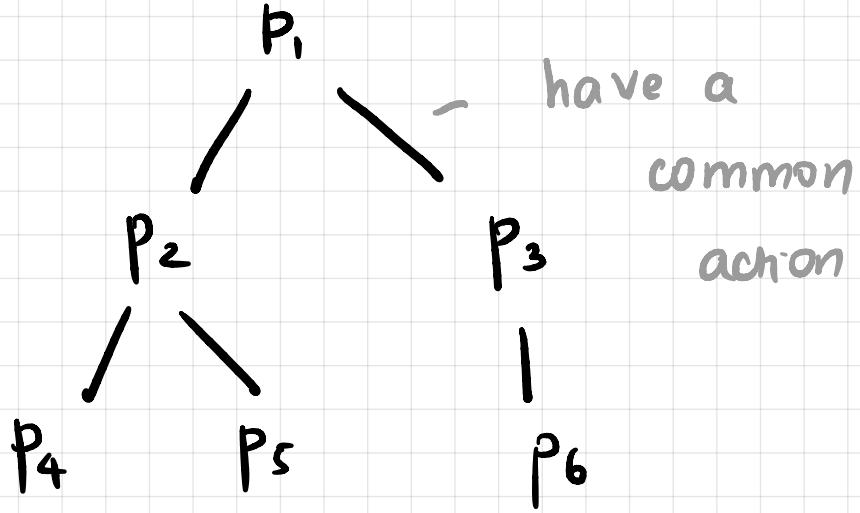
4. Generalization: Fair DFAs + acyclic architecture

Siddharth Krishna & Anca Muscholl.

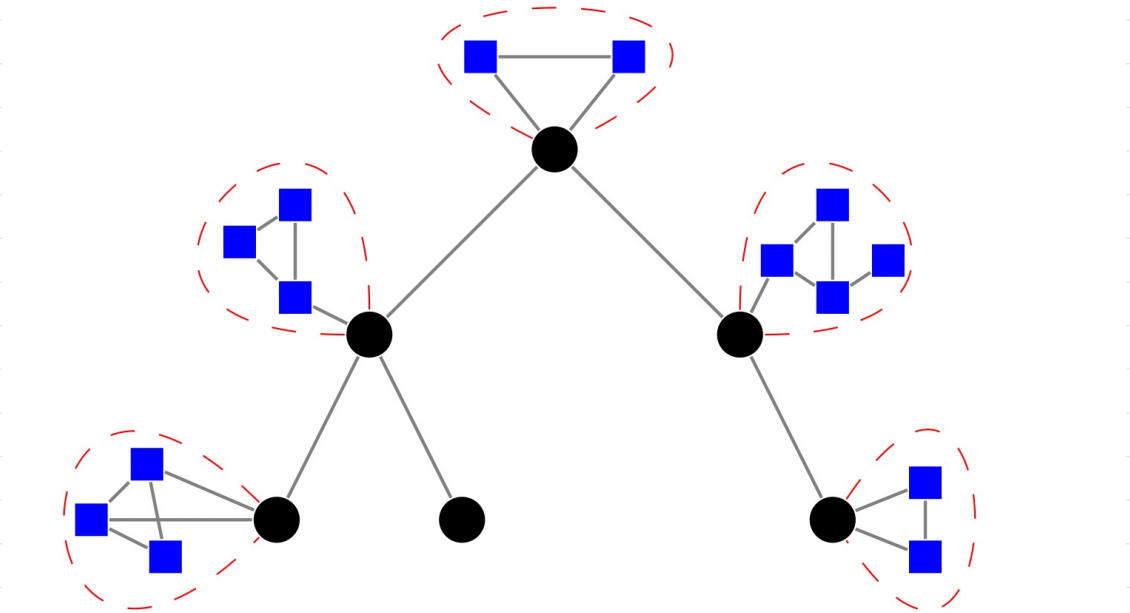
A quadratic construction for Zielonka automata with acyclic communication structure

TCS 2013

- communication graph is acyclic



Our generalization



Tree - of - bags architecture

→ Fair subsystems within a bag.

Outline

-1. Asynchronous Automata + Zeilonka's theorem ✓

-2. Fair specifications (DFAs) ✓

-3. Main construction: Fair DFAs to AA ✓

-4. Generalization: Fair DFAs + acyclic architecture ✓

Others: a matching lower bound for 3.

PERSPECTIVES

- Weaker notions of fairness
- Synthesizing bisimulation equivalent AA
- Fairness in the context of specifications stated in a logical formalism