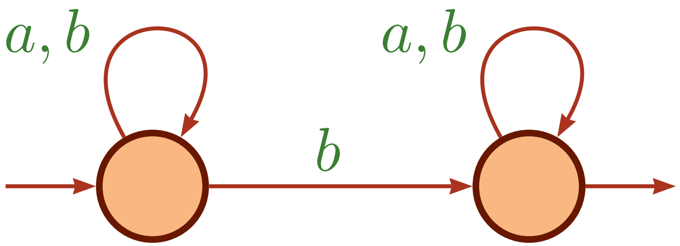
Conjugacy, Equivalence and Coverings of Automata

Sylvain Lombardy IGM - Université Marne-la-Vallée

Marie-Pierre Béal IGM - Université Marne-la-Vallée
Jacques Sakarovitch LTCI - ENST/CNRS

Automaton... with multiplicity

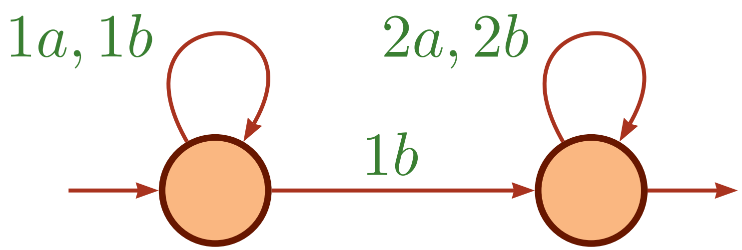
$\mathbb{B}$	"classical" automata
$\mathbb{N}$, $\mathbb{Z}$	counting paths
Fields	
$\mathrm{Rat}(B^*)$	transducers
max-plus, min-plus	distance or cost automata



Boolean: accepts words with at least one b.
Over $\mathbb{N}$: counts the number of b's.

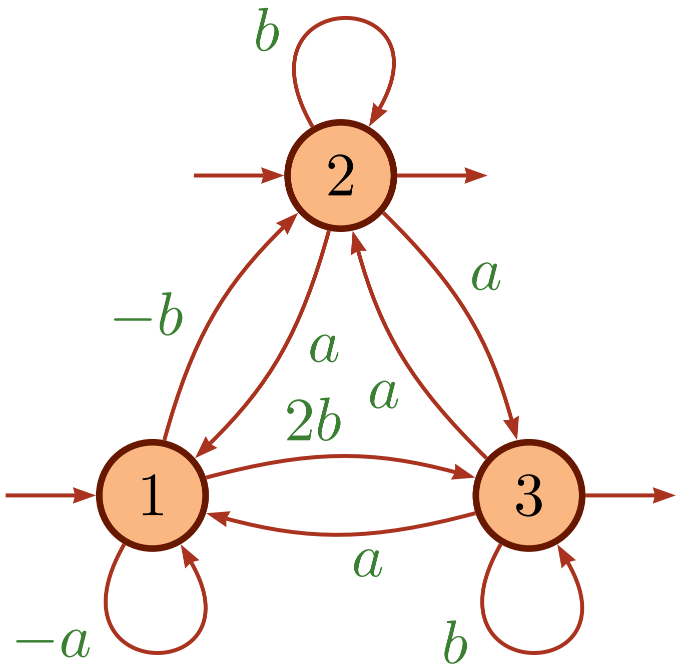
Automaton... with multiplicity

$\mathbb{B}$	“classical” automata
$\mathbb{N}$, $\mathbb{Z}$	counting paths
Fields	
Rat(B^*)	transducers
max-plus, min-plus	distance or cost automata



Over $\mathbb{N}$: value of the number written in base 2.
Over $(\mathbb{N}, \min, +)$:
length of the word + nber of a’s at the end

$\mathbb{K}$-Automaton / $\mathbb{K}$-Representation



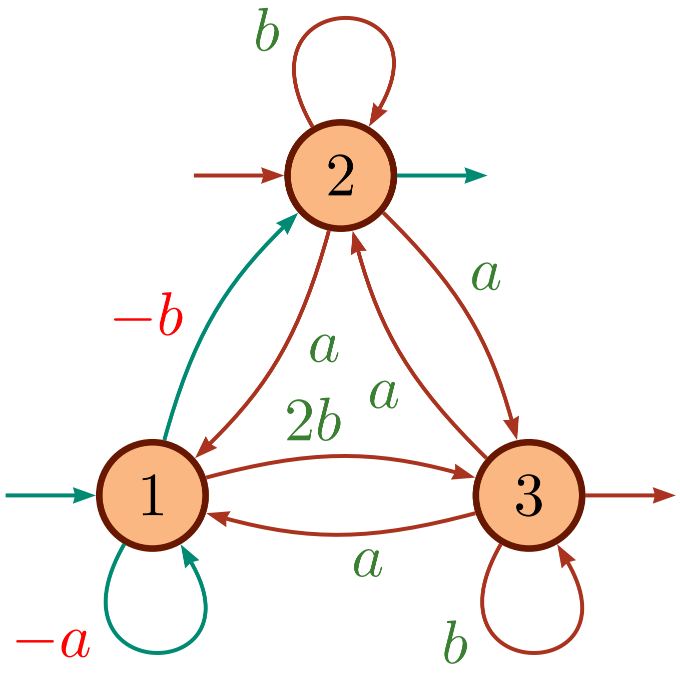
$$I = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

$$M = \begin{bmatrix} -a & -b & 2b \\ a & b & a \\ a & a & b \end{bmatrix}$$

$$T = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$\langle|\mathcal{A}|, ab\rangle =$

$\mathbb{K}$-Automaton / $\mathbb{K}$-Representation



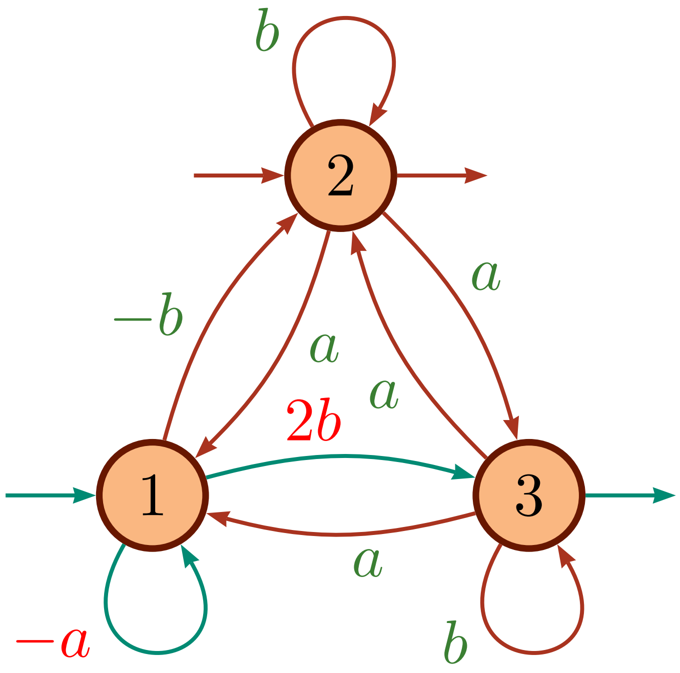
$$I = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

$$M = \begin{bmatrix} -a & -b & 2b \\ a & b & a \\ a & a & b \end{bmatrix}$$

$$T = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$\langle |\mathcal{A}|, ab \rangle = 1$

$\mathbb{K}$-Automaton / $\mathbb{K}$-Representation



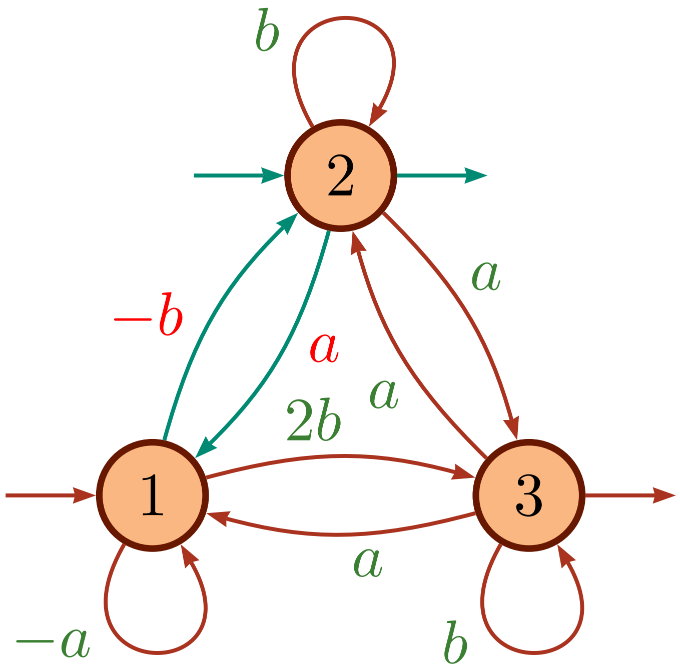
$\langle |\mathcal{A}|, ab \rangle = 1 - 2$

$$I = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

$$M = \begin{bmatrix} -a & -b & 2b \\ a & b & a \\ a & a & b \end{bmatrix}$$

$$T = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$\mathbb{K}$-Automaton / $\mathbb{K}$-Representation



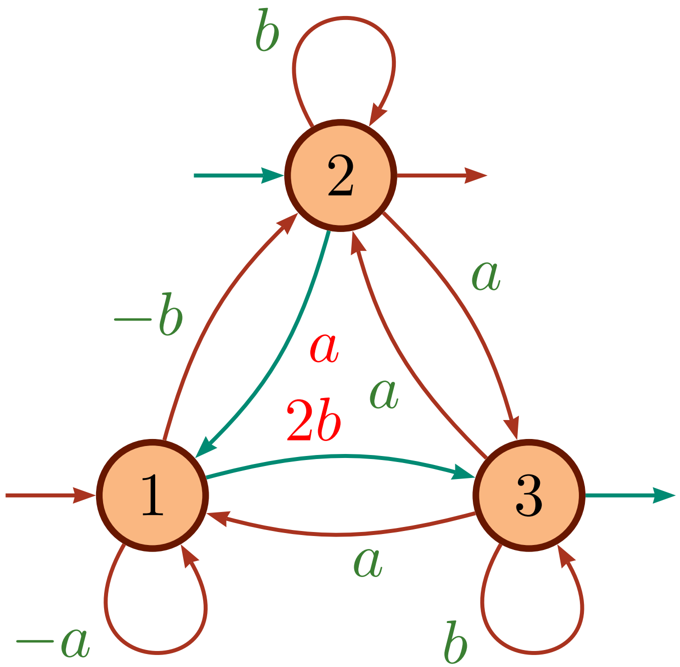
$\langle |\mathcal{A}|, ab\rangle = 1 - 2 - 1$

$$I = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

$$M = \begin{bmatrix} -a & -b & 2b \\ a & b & a \\ a & a & b \end{bmatrix}$$

$$T = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$\mathbb{K}$-Automaton / $\mathbb{K}$-Representation



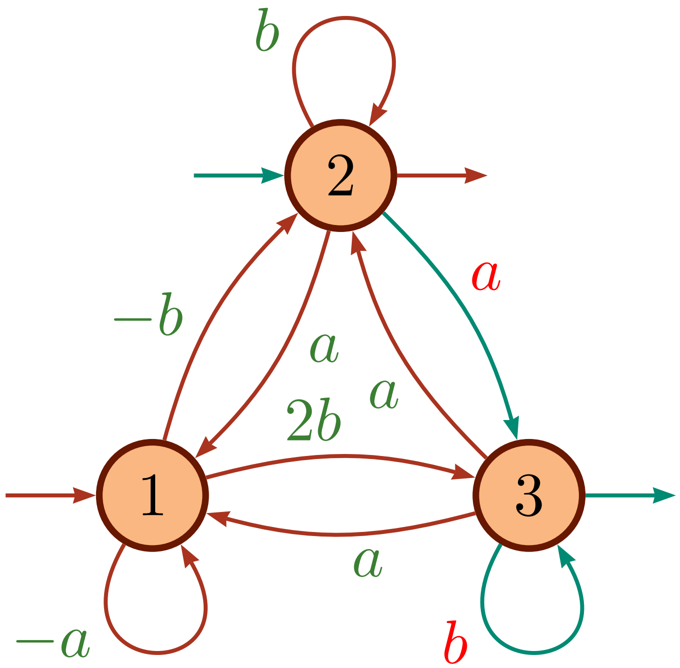
$\langle|\mathcal{A}|, ab\rangle = 1 - 2 - 1 + 2$

$$I = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

$$M = \begin{bmatrix} -a & -b & 2b \\ a & b & a \\ a & a & b \end{bmatrix}$$

$$T = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$\mathbb{K}$-Automaton / $\mathbb{K}$-Representation



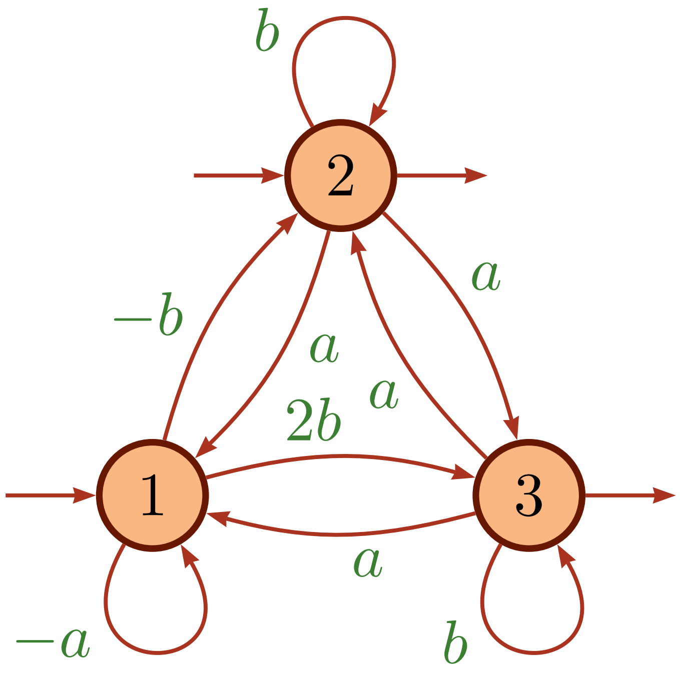
$$\langle |\mathcal{A}|, ab \rangle = 1 - 2 - 1 + 2 + 1 = 1$$

$$I = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

$$M = \begin{bmatrix} -a & -b & 2b \\ a & b & a \\ a & a & b \end{bmatrix}$$

$$T = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$\mathbb{K}$-Automaton / $\mathbb{K}$-Representation



$$I = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

$$\mu(a) = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \mu(b) = \begin{bmatrix} 0 & -1 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$\langle |\mathcal{A}|, ab \rangle = 1 - 2 - 1 + 2 + 1 = 1 = I\mu(a)\mu(b)T$$

Equivalence

Two automata are equivalent if they realize the same power series.
Decidability of equivalence depends on the semiring or on the particular form of automata:

Boolean	decidable		
Field	decidable		
Transducers	undecidable	functional	decidable
(min/max,+)	undecidable	unambiguous	decidable

Conjugacy

$\mathcal{A} = (I, M, T)$, $\mathcal{B} = (J, N, U)$. $\mathcal{A} \overset{X}{\Longrightarrow} \mathcal{B}$:

$$IX = J, \quad MX = XN, \quad \text{et} \quad T = XU.$$

Conjugacy

$\mathcal{A} = (I, M, T)$, $\mathcal{B} = (J, N, U)$. $\mathcal{A} \stackrel{X}{\Longrightarrow} \mathcal{B}$:

$$IX = J, \quad MX = XN, \quad \text{et} \quad T = XU.$$

For every w,

$$I\mu(w_1)...\mu(w_n)T = I\mu(w_1)...\mu(w_n)XU$$

Conjugacy

$\mathcal{A} = (I, M, T)$, $\mathcal{B} = (J, N, U)$. $\mathcal{A} \stackrel{X}{\Longrightarrow} \mathcal{B}$:

$$IX = J, \quad MX = XN, \quad \text{et} \quad T = XU.$$

For every w,

$$I\mu(w_1)...\mu(w_n)T = I\mu(w_1)...X\nu(w_n)U$$

$\mathcal{A} = (I, M, T)$, $\mathcal{B} = (J, N, U)$. $\mathcal{A} \stackrel{X}{\Longrightarrow} \mathcal{B}$:

$$IX = J, \quad MX = XN, \quad \text{et} \quad T = XU.$$

For every w,

$$I\mu(w_1)...\mu(w_n)T = IX\nu(w_1)...\nu(w_n)U$$

$\mathcal{A} = (I, M, T)$, $\mathcal{B} = (J, N, U)$. $\mathcal{A} \stackrel{X}{\Longrightarrow} \mathcal{B}$:

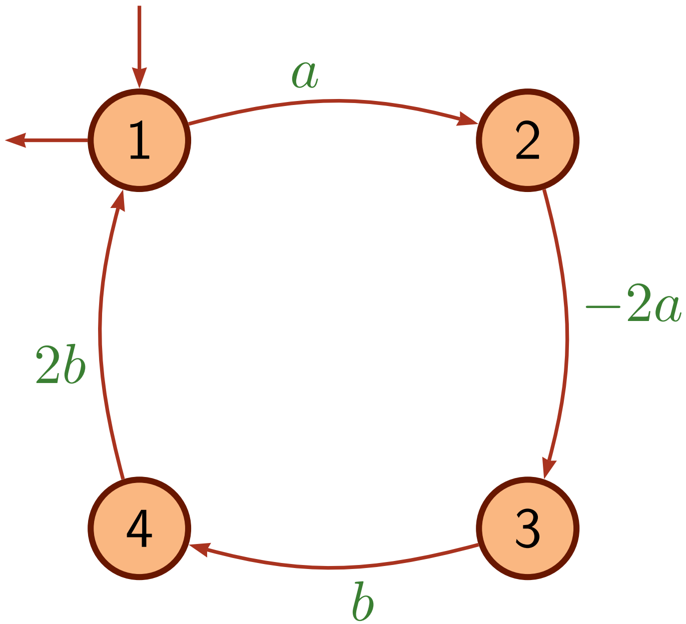
$$IX = J, \quad MX = XN, \quad \text{et} \quad T = XU.$$

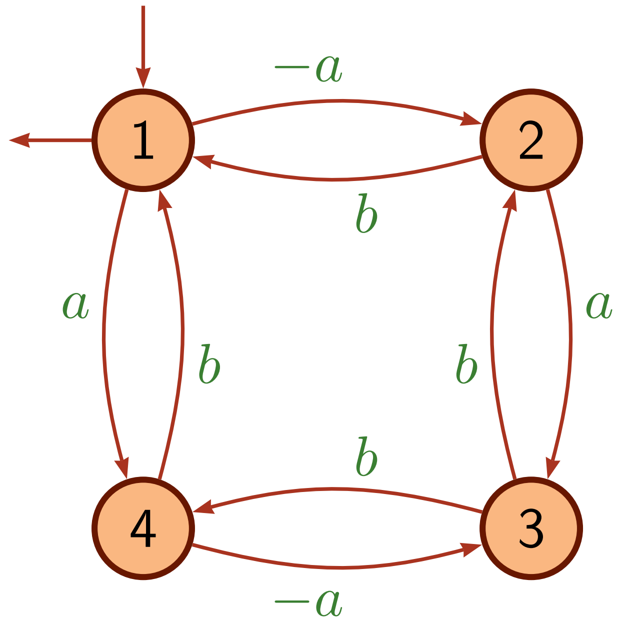
For every w,

$$I\mu(w_1)...\mu(w_n)T = J\mu(w_1)...\mu(w_n)U$$

$\Rightarrow \mathcal{A}$ and $\mathcal{B}$ are equivalent.

⚠ Conjugacy is not an equivalence relation.
It is a pre-order.

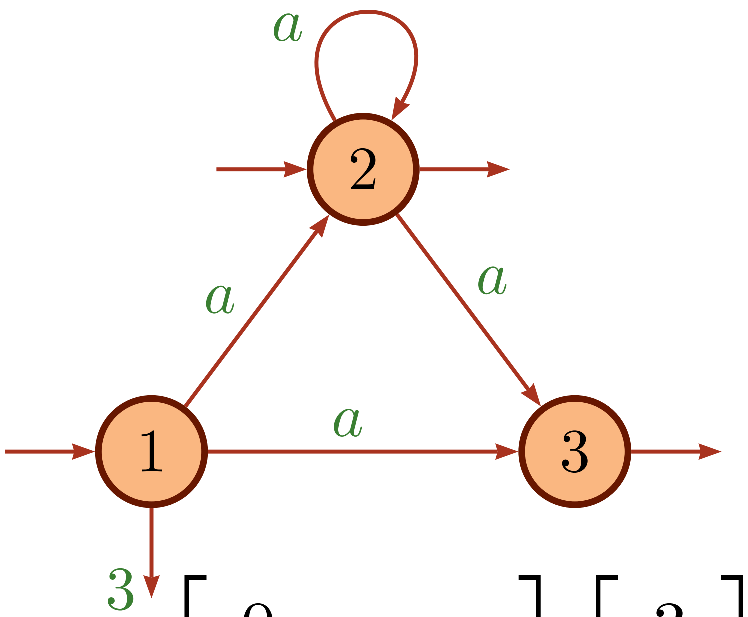




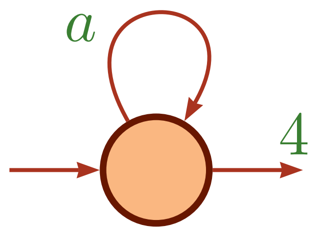
Conjugated in $\mathbb{Z}$:

$$\begin{bmatrix} 0 & a & 0 & 0 \\ 0 & 0 & -2a & 0 \\ 0 & 0 & 0 & b \\ 2b & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & -a & 0 & a \\ b & 0 & a & 0 \\ 0 & b & 0 & b \\ b & 0 & -a & 0 \end{bmatrix}$$

Two equivalent automata may be not conjugated.



$$[1\ 1\ 0]\begin{bmatrix}0 & a & a\\ 0 & a & a\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}3\\ 1\\ 1\end{bmatrix}$$



$$[1][a][4]$$

Conjugacy and equivalence

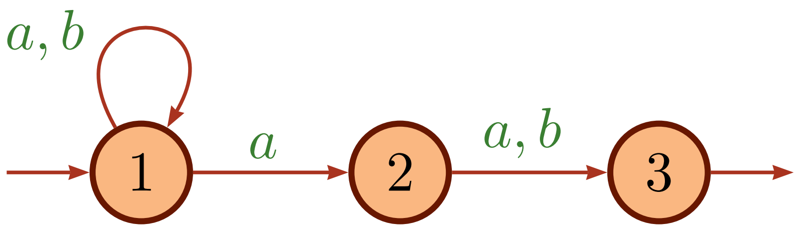
Theorem 1: Let $\mathcal{A}$ and $\mathcal{B}$ be

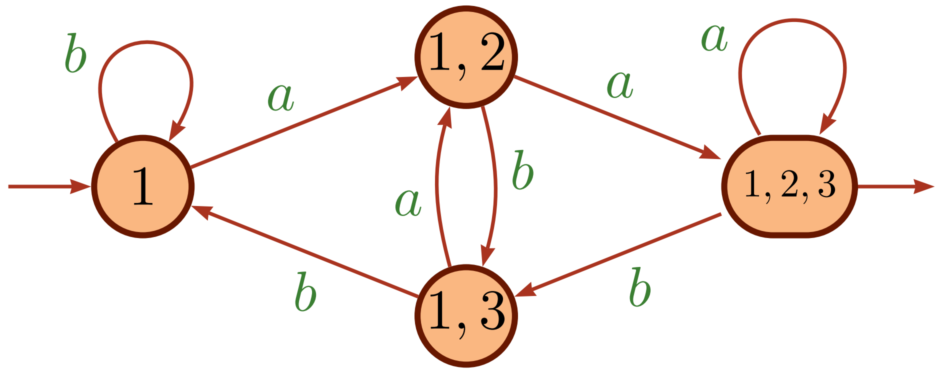
- two automata,
- two $\mathbb{N}$-automata,
- two $\mathbb{Z}$-automata,
- two $\mathbb{K}$-automata, with $\mathbb{K}$ field,
- two functional transducers.

If $\mathcal{A}$ and $\mathcal{B}$ are equivalent, there exists $\mathcal{C}$ such that $\mathcal{A} \stackrel{X}{\Longleftarrow} \mathcal{C} \stackrel{Y}{\Longrightarrow} \mathcal{B}$.

Boolean: conjugacy and determinization

$\mathsf{det}(\mathcal{A}) \stackrel{X}{\Longrightarrow} \mathcal{A}$





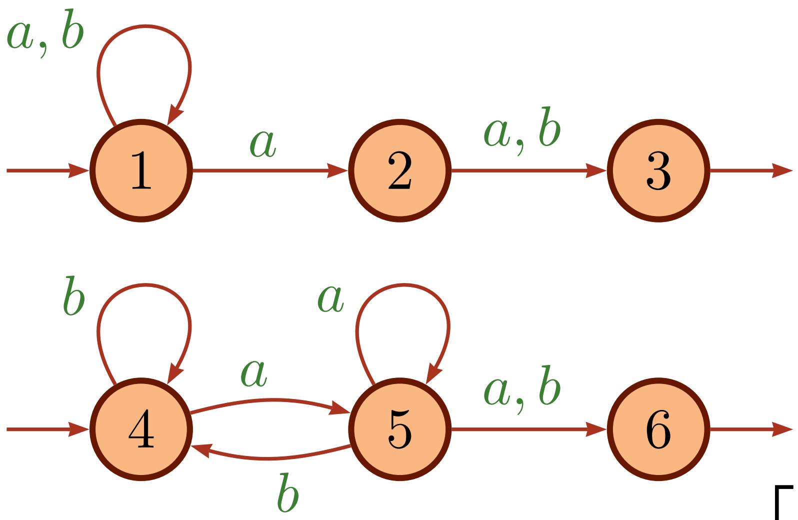
Every state of $\mathsf{det}(\mathcal{A})$ is a row vector $I\mu(w)$.

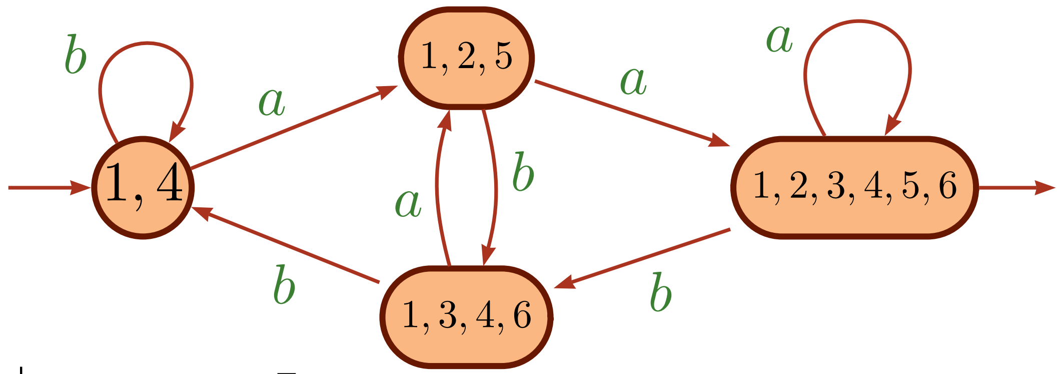
Conjugacy matrix: $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

Likewise: $\mathcal{A} \stackrel{X}{\Longrightarrow} \mathsf{codet}(\mathcal{A})$

Boolean: conjugacy and determinization

$\mathcal{A}$, $\mathcal{B}$ equivalent. Let $\mathcal{C} = \det(\mathcal{A} \cup \mathcal{B})$





$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]$$

Thus $\mathcal{C} \stackrel{[X|Y]}{\Longrightarrow} \mathcal{A} \cup \mathcal{B}$.

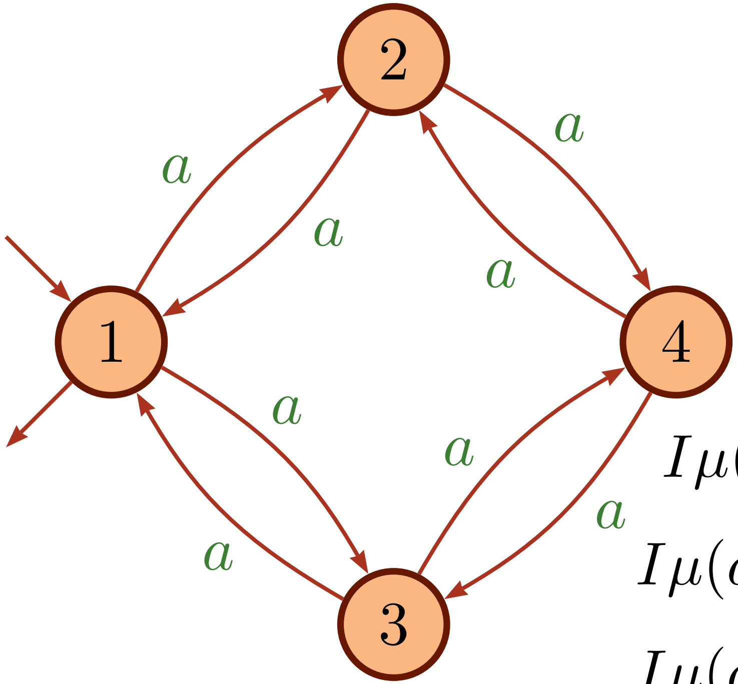
Finally, $\mathcal{C} \stackrel{X}{\Longrightarrow} \mathcal{A}$ and $\mathcal{C} \stackrel{Y}{\Longrightarrow} \mathcal{B}$.

$\longrightarrow$ Theorem 1 holds for Boolean.

Fields: reduction and conjugacy

$\mathcal{A} = (I, \mu, T)$

Left reduction : computing a basis of $\langle I\mu(w)\rangle$.



$I = [1\ 0\ 0\ 0] \longrightarrow$ state 1

$I\mu(a) = [0\ 1\ 1\ 0] \longrightarrow$ state 2

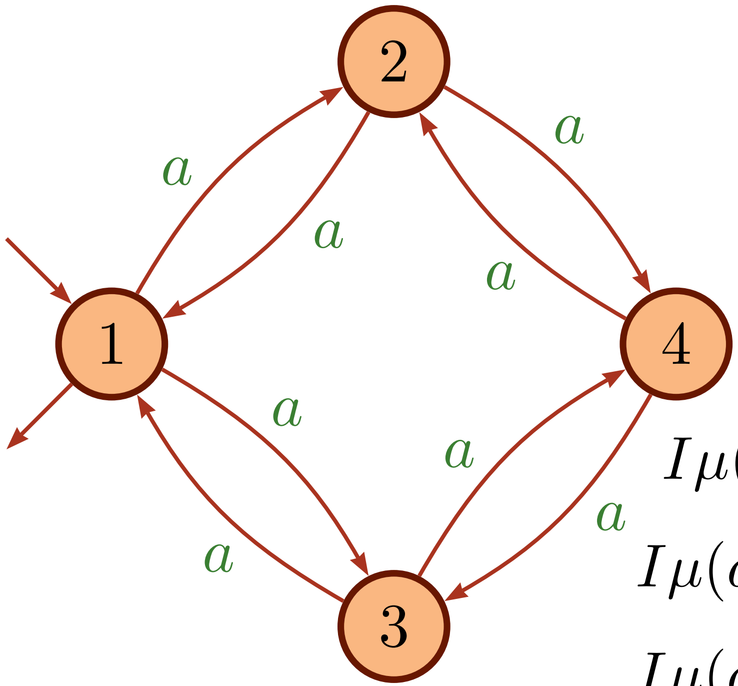
$I\mu(a^2) = 2[1\ 0\ 0\ 1] \longrightarrow$ state 3

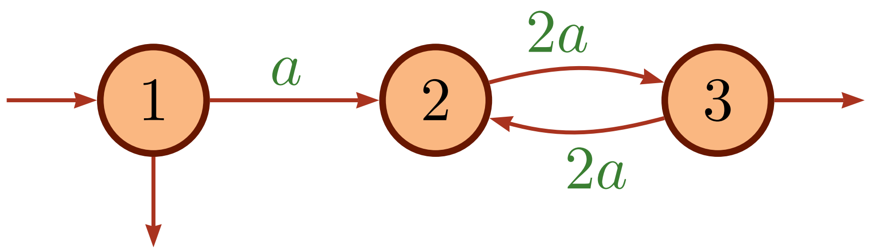
$I\mu(a^3) = 4[0\ 1\ 1\ 0]$

Fields: reduction and conjugacy

$\mathcal{A} = (I, \mu, T)$

Left reduction : computing a basis of $\langle I\mu(w) \rangle$.





$I = [1\ 0\ 0\ 0] \rightarrow$ state 1

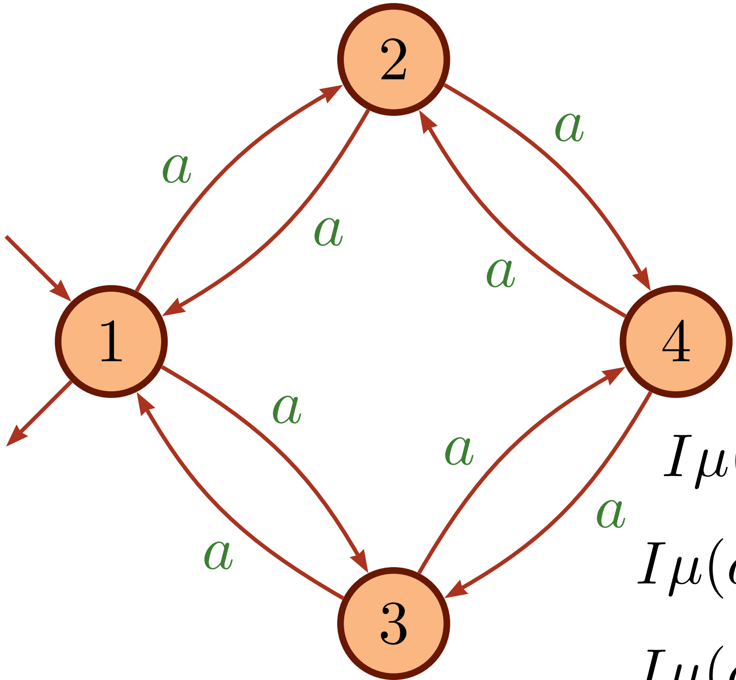
$I\mu(a) = [0\ 1\ 1\ 0] \rightarrow$ state 2

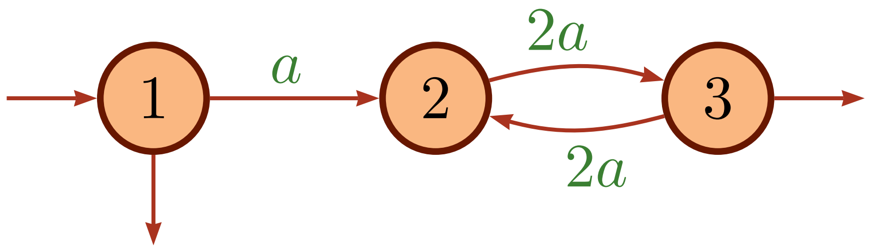
$I\mu(a^2) = 2[1\ 0\ 0\ 1] \rightarrow$ state 3

$I\mu(a^3) = 4[0\ 1\ 1\ 0]$

$\mathcal{A} = (I, \mu, T)$

Left reduction : computing a basis of $\langle I\mu(w)\rangle$.





$I = [1\ 0\ 0\ 0] \rightarrow$ state 1

$I\mu(a) = [0\ 1\ 1\ 0] \rightarrow$ state 2

$I\mu(a^2) = 2[1\ 0\ 0\ 1] \rightarrow$ state 3

$I\mu(a^3) = 4[0\ 1\ 1\ 0]$

$\text{red}_g(\mathcal{A}) \stackrel{X}{\Longrightarrow} \mathcal{A}$, with $X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$

Fields: reduction and conjugacy

Likewise $\mathcal{A} \stackrel{X}{\Longrightarrow} \mathsf{red}_d(\mathcal{A})$.

Remarks:

– $\mathsf{red}_g(\mathsf{red}_d(\mathcal{A}))$ is a reduced automaton (minimal number of states);

– all reduced automata are conjugated both ways with an invertible matrix (change of basis).

Fields: reduction and conjugacy

$\mathcal{A}$, $\mathcal{B}$ equivalent. Let $\mathcal{C} = \mathsf{red}_g(\mathcal{A} + \mathcal{B})$

Thus $\mathcal{C} \stackrel{[X|Y]}{\Longrightarrow} \mathcal{A} + \mathcal{B}$.

$\mathcal{C} = (I, M, T)$; set $\mathcal{C}' = (I, M, T/2)$.

Finally, $\mathcal{C}' \stackrel{X}{\Longrightarrow} \mathcal{A}$ and $\mathcal{C}' \stackrel{Y}{\Longrightarrow} \mathcal{B}$.

$\longrightarrow$ Theorem 1 holds on fields.

Integers case

Over $\mathbb{Z}$, every thing works like over fields: a basis of the $\mathbb{Z}$-module $\langle I\mu(w)\rangle$ is computed.

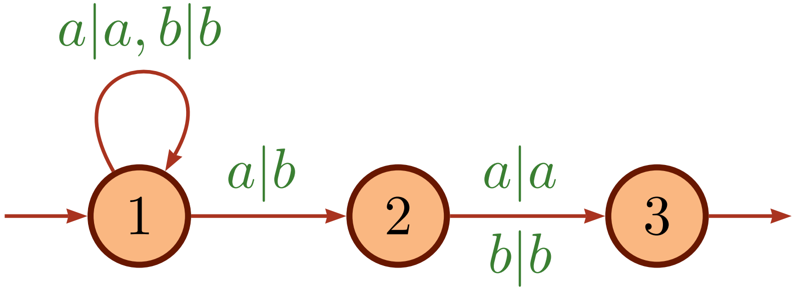
Over $\mathbb{N}$, a generator set of the $\mathbb{N}$-semi-module containing the vectors $I\mu(w)$ is computed using the good properties of $\mathbb{N}^k$.

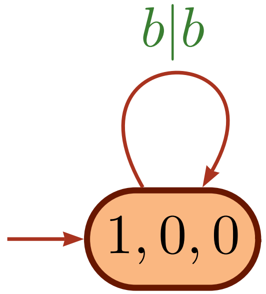
⚠ Let $\mathcal{A} = (I, \mu, T)$ and $\mathcal{B} = (J, \nu, U)$ be equivalent $\mathbb{N}$-automata with resp. dim. r and s.
During the common reduction of $\mathcal{A} + \mathcal{B}$, only vectors $[x \mid y]$ such that $x.T = y.U$ can be used in the generator set.

Functional transducers: sequentialization

$\mathcal{T} = (I, \mu, T)$ functional transducer

$$I = [1\ 0\ 0],\ \mu(a) = \begin{bmatrix} a & b & 0 \\ 0 & 0 & a \\ 0 & 0 & 0 \end{bmatrix},\ \mu(b) = \begin{bmatrix} b & 0 & 0 \\ 0 & 0 & b \\ 0 & 0 & 0 \end{bmatrix},\ T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

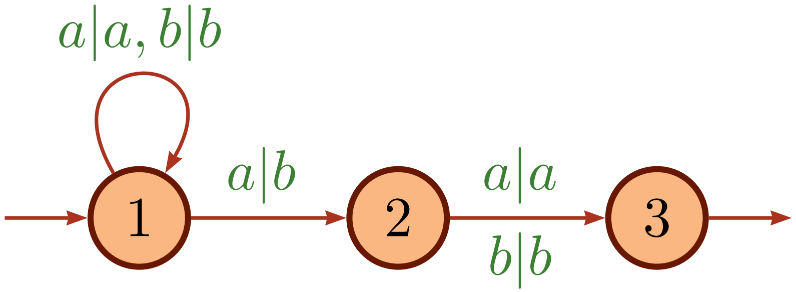


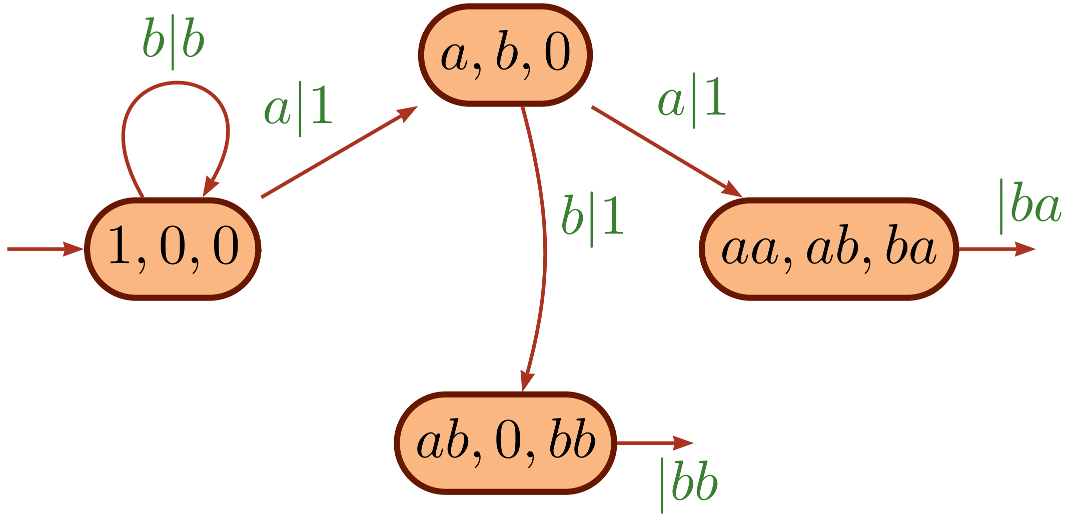


Functional transducers: sequentialization

$\mathcal{T} = (I, \mu, T)$ functional transducer

$$I = [1\ 0\ 0],\ \mu(a) = \begin{bmatrix} a & b & 0 \\ 0 & 0 & a \\ 0 & 0 & 0 \end{bmatrix},\ \mu(b) = \begin{bmatrix} b & 0 & 0 \\ 0 & 0 & b \\ 0 & 0 & 0 \end{bmatrix},\ T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

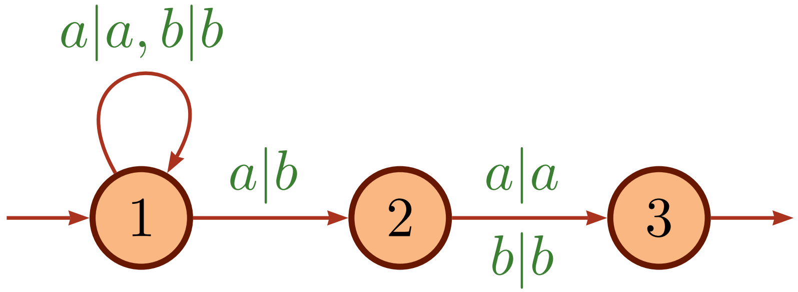


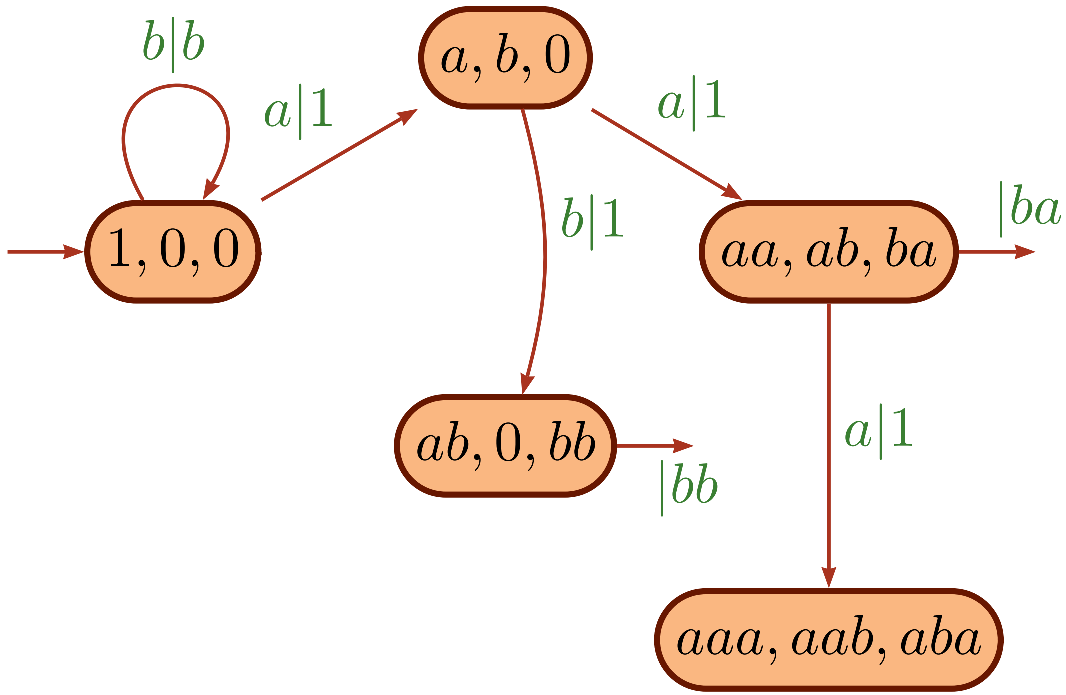


Functional transducers: sequentialization

$\mathcal{T} = (I, \mu, T)$ functional transducer

$$I = [1\ 0\ 0],\ \mu(a) = \begin{bmatrix} a & b & 0 \\ 0 & 0 & a \\ 0 & 0 & 0 \end{bmatrix},\ \mu(b) = \begin{bmatrix} b & 0 & 0 \\ 0 & 0 & b \\ 0 & 0 & 0 \end{bmatrix},\ T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

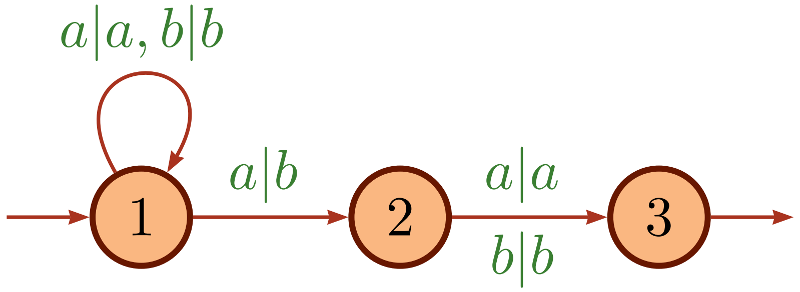


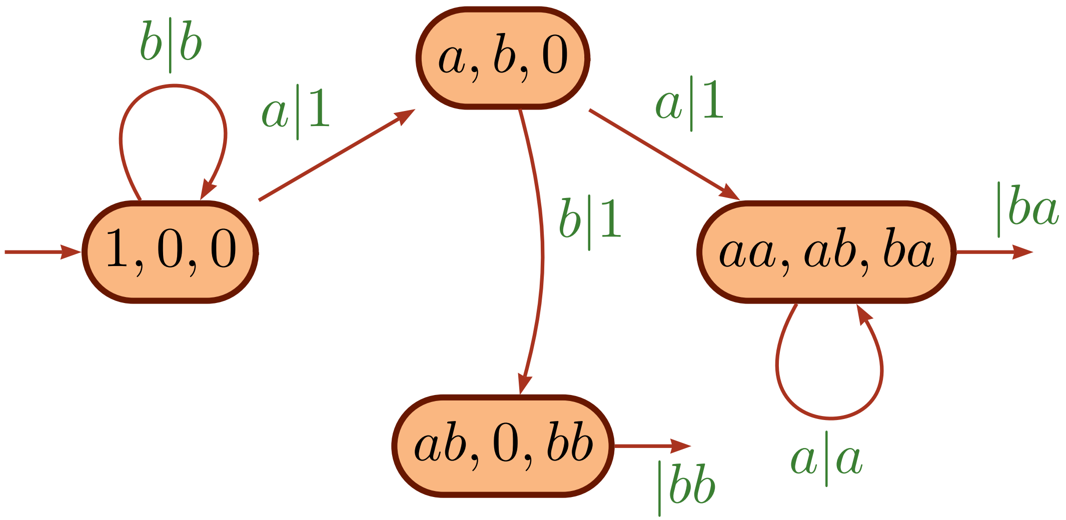


Functional transducers: sequentialization

$\mathcal{T} = (I, \mu, T)$ functional transducer

$$I = [1\ 0\ 0],\ \mu(a) = \begin{bmatrix} a & b & 0 \\ 0 & 0 & a \\ 0 & 0 & 0 \end{bmatrix},\ \mu(b) = \begin{bmatrix} b & 0 & 0 \\ 0 & 0 & b \\ 0 & 0 & 0 \end{bmatrix},\ T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

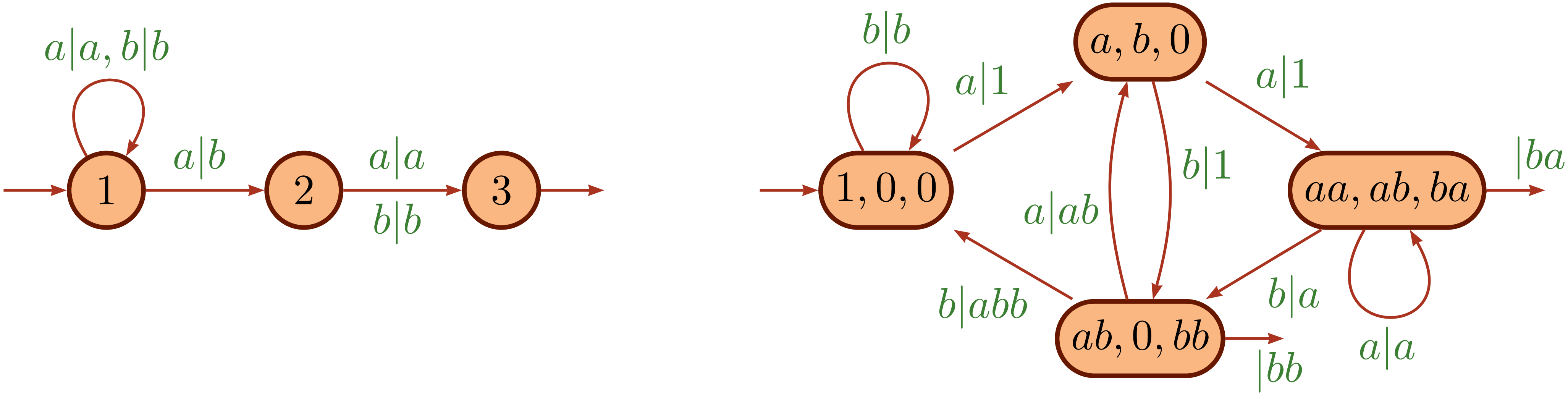




Functional transducers: sequentialization

$\mathcal{T} = (I, \mu, T)$ functional transducer

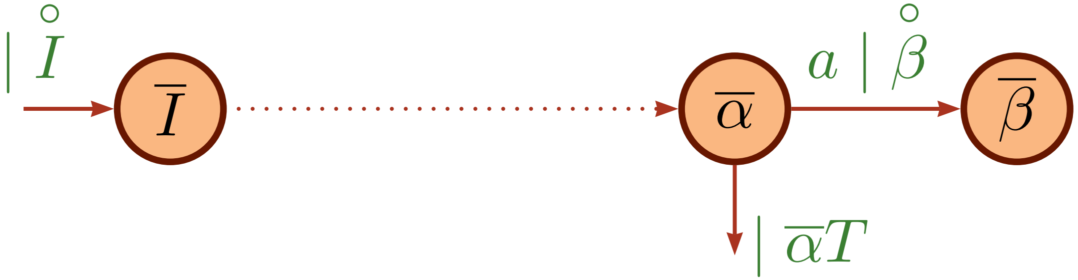
$$I = [1\ 0\ 0],\ \mu(a) = \begin{bmatrix} a & b & 0 \\ 0 & 0 & a \\ 0 & 0 & 0 \end{bmatrix},\ \mu(b) = \begin{bmatrix} b & 0 & 0 \\ 0 & 0 & b \\ 0 & 0 & 0 \end{bmatrix},\ T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$



Functional transducers: sequentialization

$\mathcal{T} = (I, \mu, T)$ functional transducer

Sequentialization: α words vector
$\mathring{\alpha}$: largest common prefix
$\overline{\alpha} = \mathring{\alpha}^{-1}\alpha$

seq$(\mathcal{T})$: 

, with $\beta = \overline{\alpha}\mu(a)$

⚠ $\mathcal{T}$ non sequentializable $\Longrightarrow$ seq$(\mathcal{T})$ infinite

Functional transducers: quasi-sequentialization

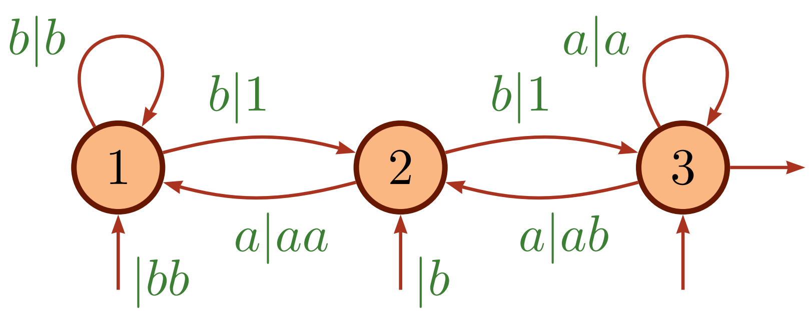
Idea: if components of α are too different, they cannot be used for the same words.

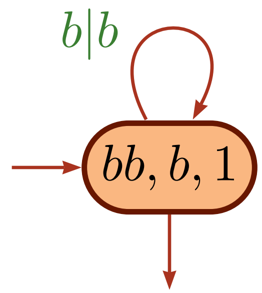
$\rightarrow$We require:

– every $\overline{\alpha}$ has to contain the empty word

– if α_i is non minimal, there exists α_j, prefix of α_i such that $|\alpha_i| - |\alpha_j| < K(\mathcal{T})$

Otherwise α is split into a union of disjoint support vectors that fit these properties. The transducer that follows is not sequential. $\rightarrow$qseq$(\mathcal{T})$





Functional transducers: quasi-sequentialization

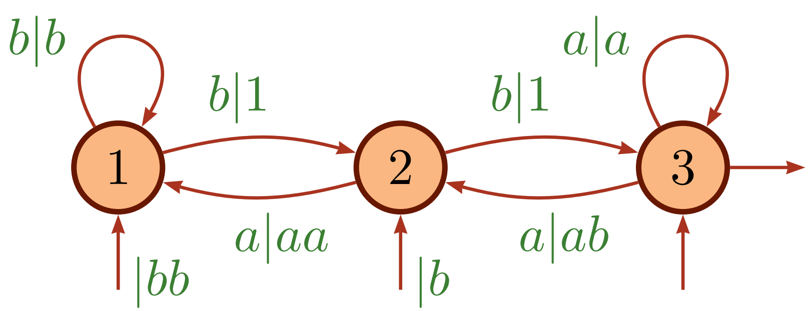
Idea: if components of α are too different, they cannot be used for the same words.

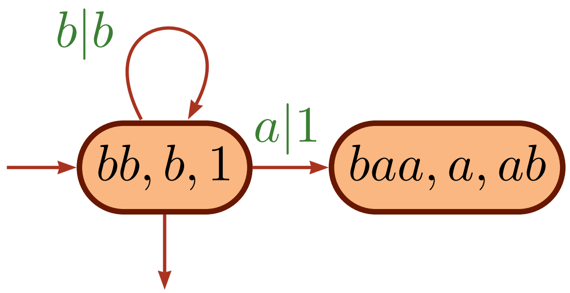
→We require:

– every $\overline{\alpha}$ has to contain the empty word

– if α_i is non minimal, there exists α_j, prefix of α_i such that $|\alpha_i| - |\alpha_j| < K(\mathcal{T})$

Otherwise α is split into a union of disjoint support vectors that fit these properties. The transducer that follows is not sequential. →qseq$(\mathcal{T})$





Functional transducers: quasi-sequentialization

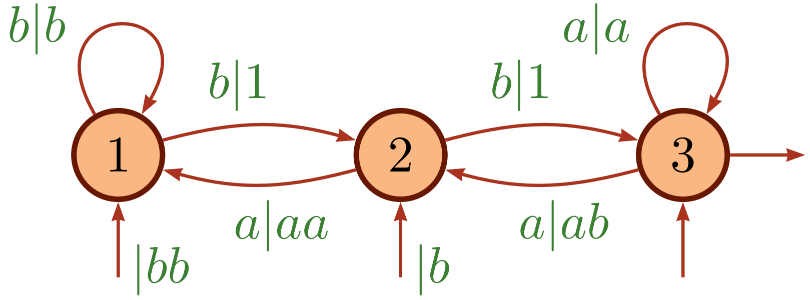
Idea: if components of α are too different, they cannot be used for the same words.

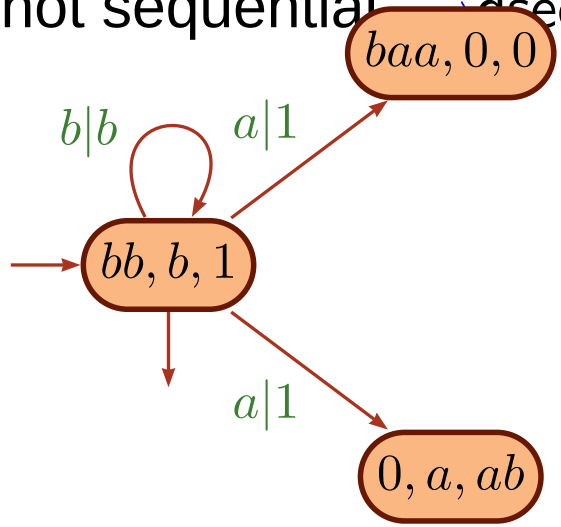
$\rightarrow$We require:

– every $\overline{\alpha}$ has to contain the empty word

– if α_i is non minimal, there exists α_j, prefix of α_i such that $|\alpha_i| - |\alpha_j| < K(\mathcal{T})$

Otherwise α is split into a union of disjoint support vectors that fit these properties. The transducer that follows is not sequential $\rightarrow$ qseq$(\mathcal{T})$





Functional transducers: quasi-sequentialization

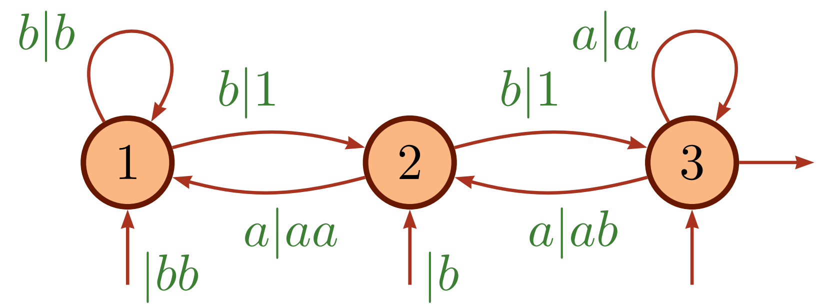
Idea: if components of α are too different, they cannot be used for the same words.

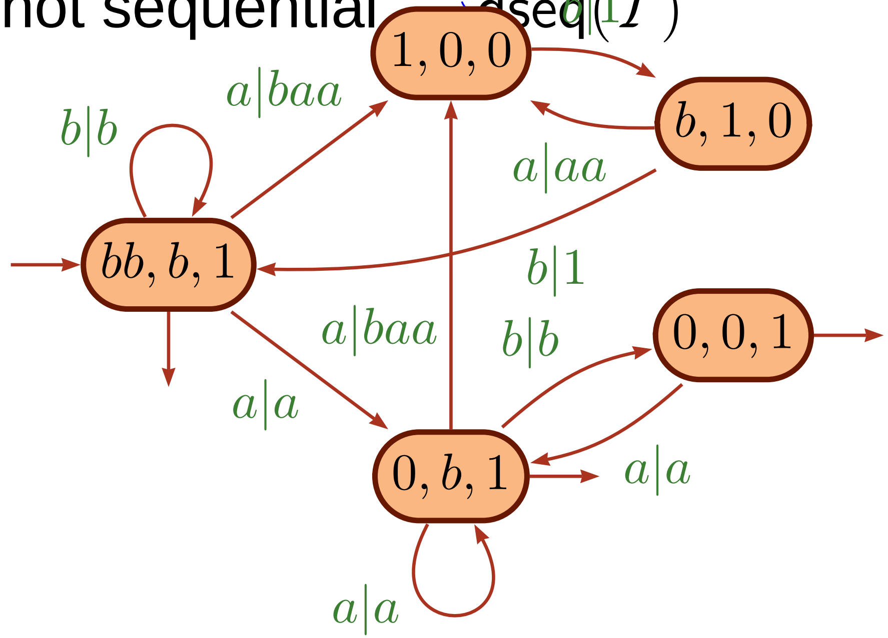
$\rightarrow$We require:

– every $\overline{\alpha}$ has to contain the empty word

– if α_i is non minimal, there exists α_j, prefix of α_i such that $|\alpha_i| - |\alpha_j| < K(\mathcal{T})$

Otherwise α is split into a union of disjoint support vectors that fit these properties. The transducer that follows is not sequential $\rightarrow$ qseq($\mathcal{T}$)





Functional transducers: quasi-sequentialization

Idea: if components of α are too different, they cannot be used for the same words.

$\rightarrow$We require:

– every $\overline{\alpha}$ has to contain the empty word

– if α_i is non minimal, there exists α_j, prefix of α_i such that $|\alpha_i| - |\alpha_j| < K(\mathcal{T})$

Otherwise α is split into a union of disjoint support vectors that fit these properties. The transducer that follows is not sequential. $\rightarrow \mathsf{qseq}(\mathcal{T})$

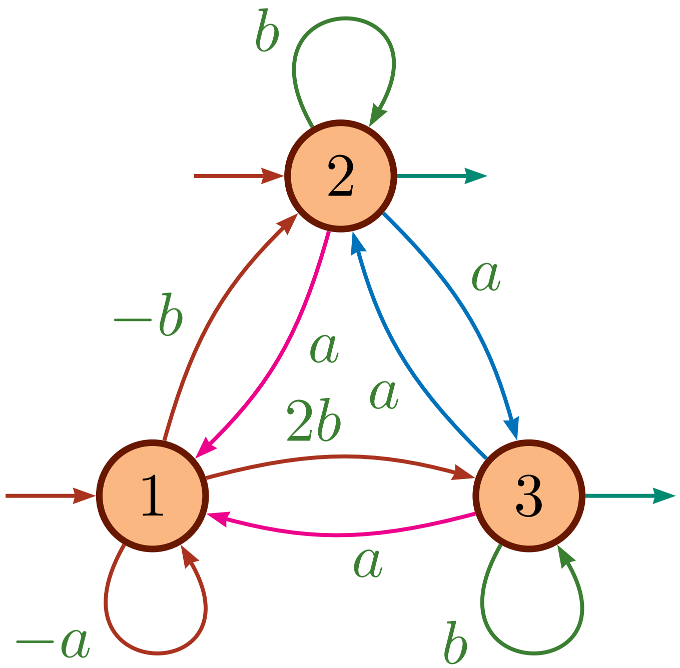
What is the mean ?

– The transducer is unambiguous

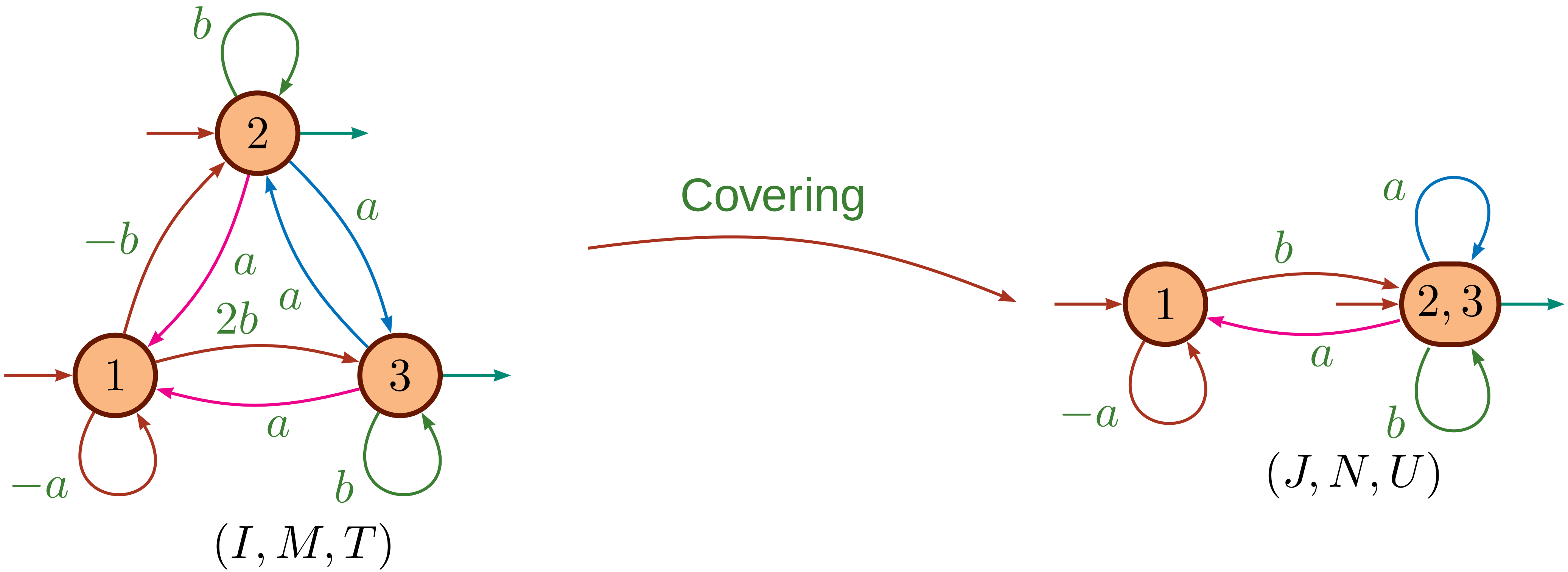
– $\mathsf{qseq}(\mathcal{T}) \overset{X}{\Longrightarrow} \mathcal{T}$

– $\mathsf{qseq}(\mathcal{T} \cup \mathcal{T}')$ is conjugated to $\mathcal{T}$ and à $\mathcal{T}'$ (if they are equivalent).

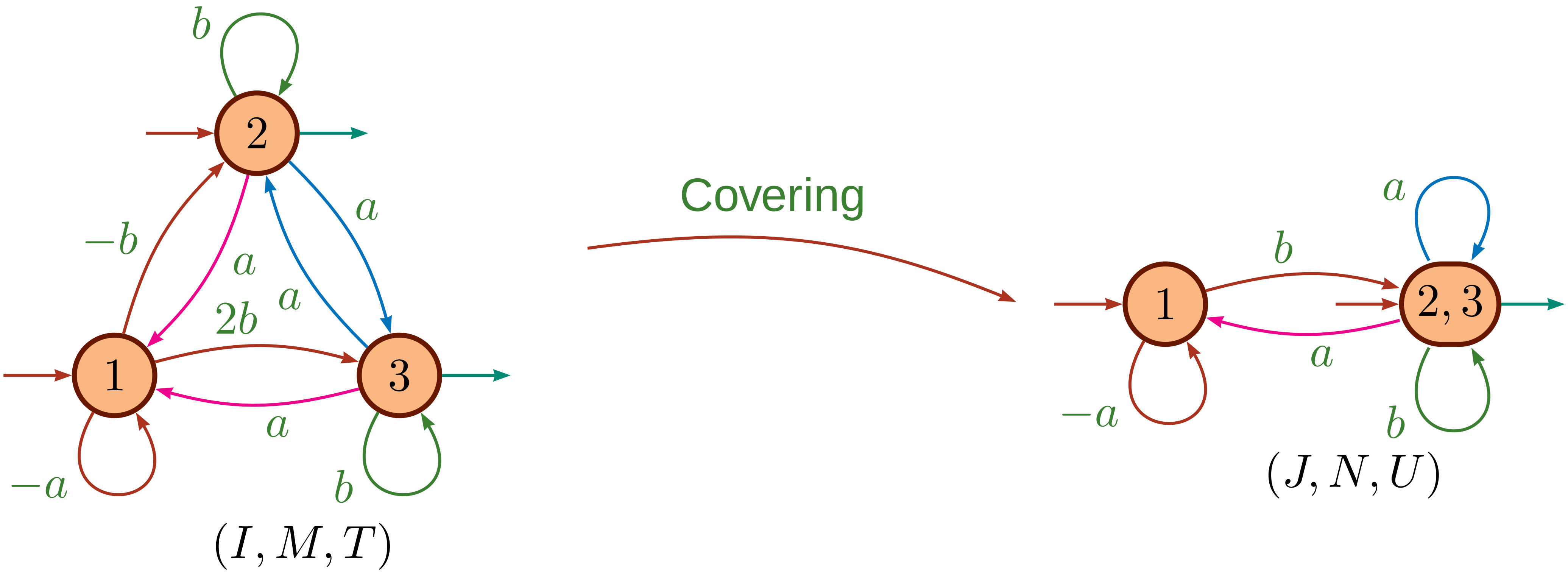
$\rightarrow$ Theorem 1 holds for functional transducers.



(I, M, T)


b
2
−b
a
a
a
2b
1
3
a
−a
b
(I, M, T)
Covering
a
b
1
2, 3
a
−a
b
(J, N, U)

Covering / Quotient



$$\left.\begin{bmatrix} -a & -b & 2b \\ a & b & a \\ a & a & b \end{bmatrix}\right\} \longrightarrow \begin{bmatrix} -a & b \\ a & a+b \end{bmatrix}$$

Covering / Quotient

$$\left.\begin{bmatrix} -a & -b & 2b \\ a & b & a \\ a & a & b \end{bmatrix}\right\} \longrightarrow \begin{bmatrix} -a & b \\ a & a+b \end{bmatrix}$$

— ∘ —

$$\begin{bmatrix} -a & -b & 2b \\ a & b & a \\ a & a & b \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -a & b \\ a & a+b \end{bmatrix}$$

$$\left.\begin{bmatrix} -a & -b & 2b \\ a & b & a \\ a & a & b \end{bmatrix}\right\} \longrightarrow \begin{bmatrix} -a & b \\ a & a+b \end{bmatrix}$$

$$- \circ -$$

$$\begin{bmatrix} -a & -b & 2b \\ a & b & a \\ a & a & b \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -a & b \\ a & a+b \end{bmatrix}$$

Initial: $\begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix}$, Final: $\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Covering / Quotient

Definition: $\mathcal{A} = (I, M, T)$ and $\mathcal{B} = (J, N, U)$, $\mathbb{Z}$-automata.

$\mathcal{A}$ is a *covering* of $\mathcal{B}$
$\mathcal{B}$ is a *quotient* of $\mathcal{A}$ if there exists an amalgamation matrix X such that

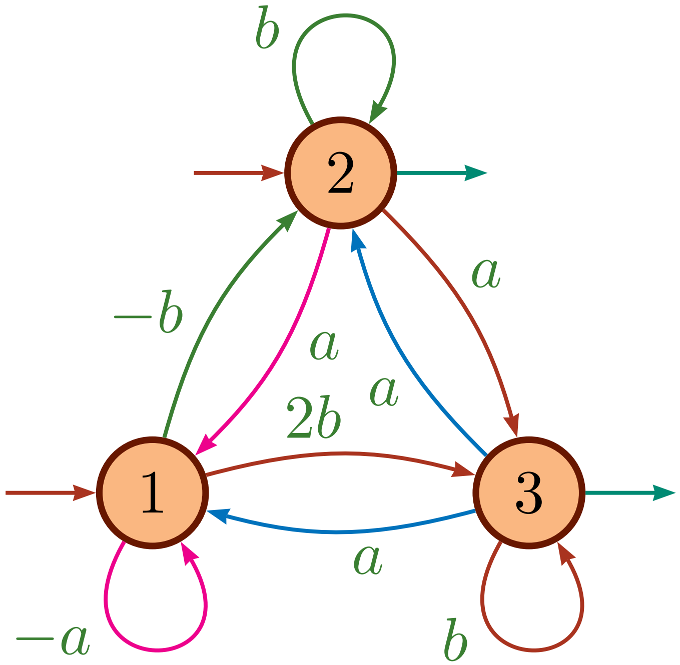
$$IX = J, \quad MX = XN, \quad \text{et } T = XU.$$

Definition: $\mathcal{A} = (I, M, T)$ and $\mathcal{B} = (J, N, U)$

$\mathcal{A}$ is a *co-covering* of $\mathcal{B}$
$\mathcal{B}$ is a *co-quotient* of $\mathcal{A}$ if there exists an amalgamation matrix X such that

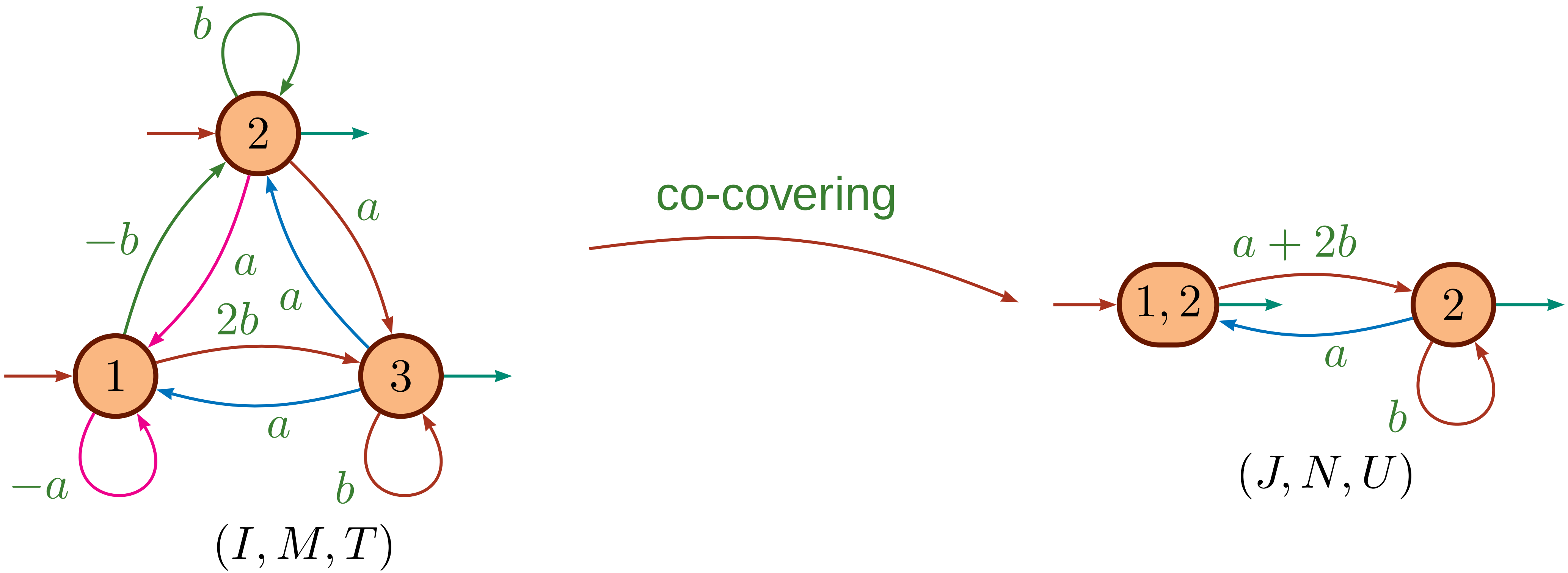
$$I = J\,{}^tX, \quad {}^tXM = N\,{}^tX, \quad \text{et } {}^tXT = U.$$

co-covering / co-quotient

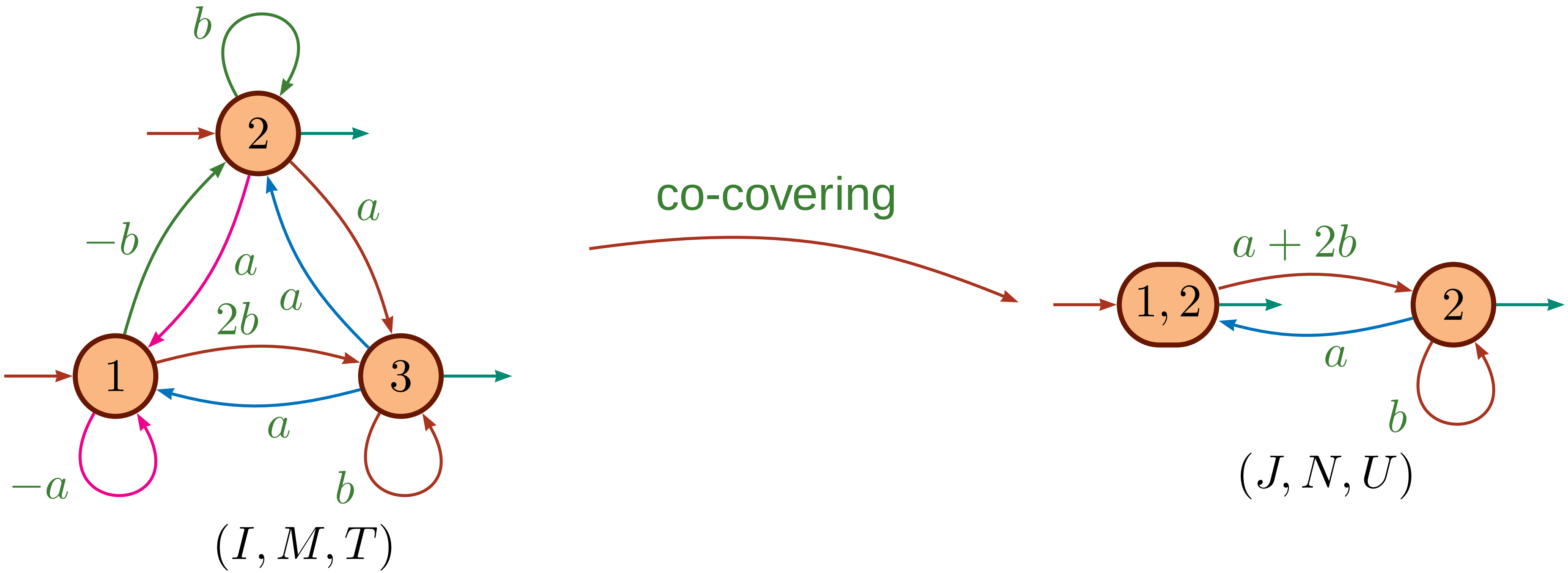


(I, M, T)

co-covering / co-quotient

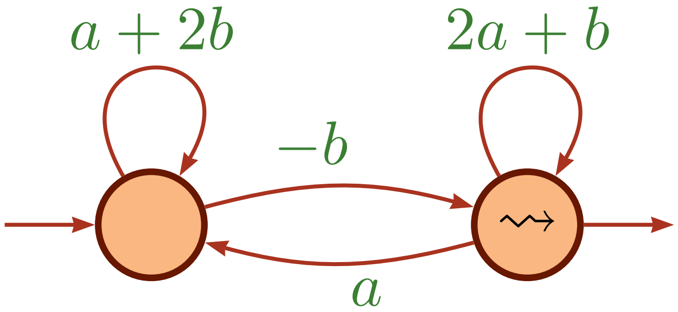


co-covering / co-quotient



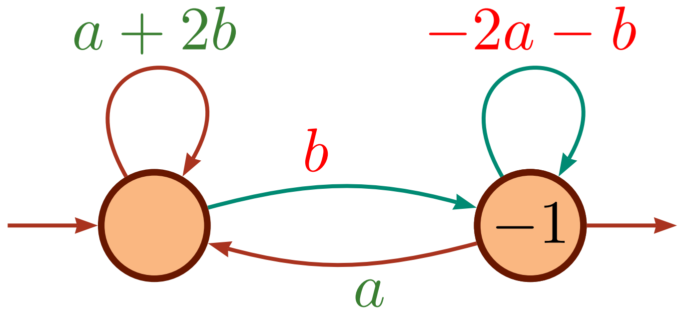
$$\begin{bmatrix} -a & -b & 2b \\ a & b & a \\ a & a & b \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & a+2b \\ a & b \end{bmatrix}$$

Circulation of invertible elements

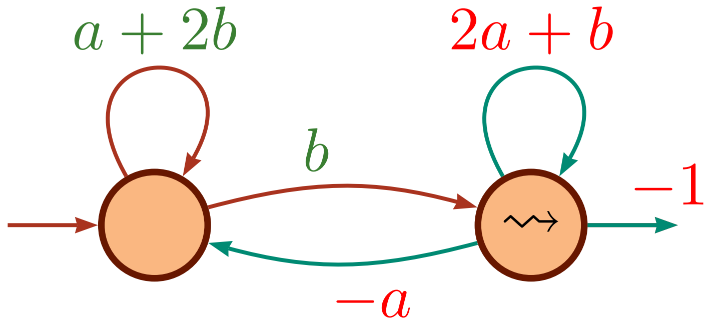


$$(I, M, T)$$

Circulation of invertible elements

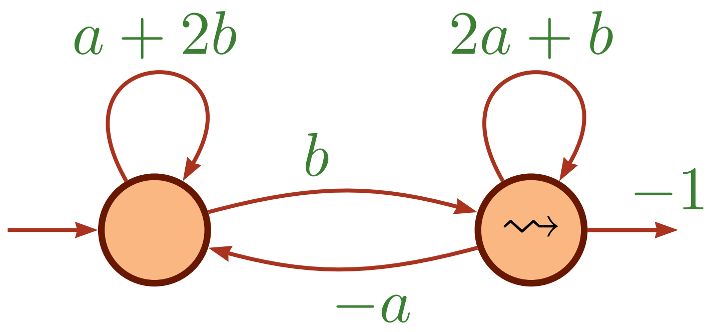


Circulation of invertible elements



(J, N, U)

Circulation of invertible elements



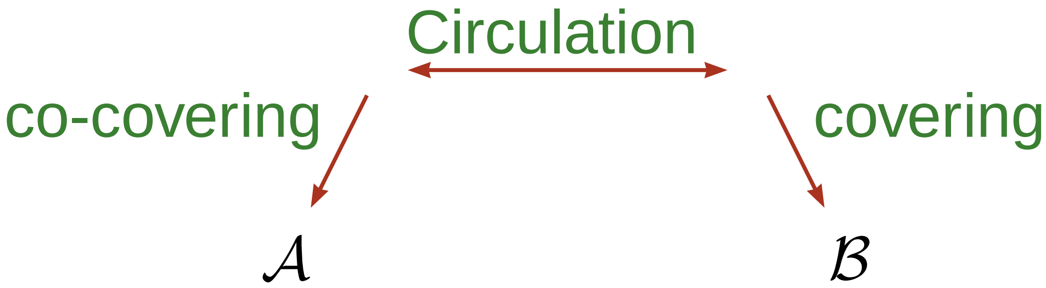
$$(J, N, U)$$

$$I\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} = J, \quad M\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}N, \quad T = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}U$$

Conjugacy and coverings

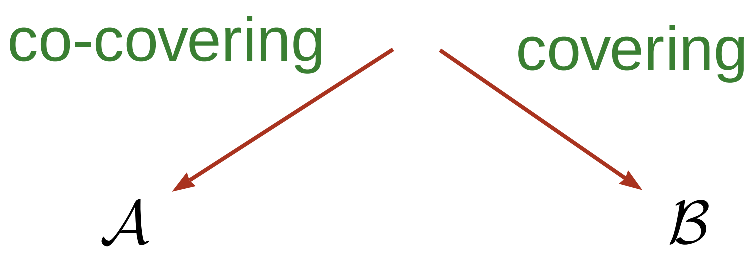
Theorem 2: Let $\mathcal{A}$ and $\mathcal{B}$ be
- two $\mathbb{Z}$-automata
- two $\mathbb{K}$-automata, where $\mathbb{K}$ field
- two functional trim transducers

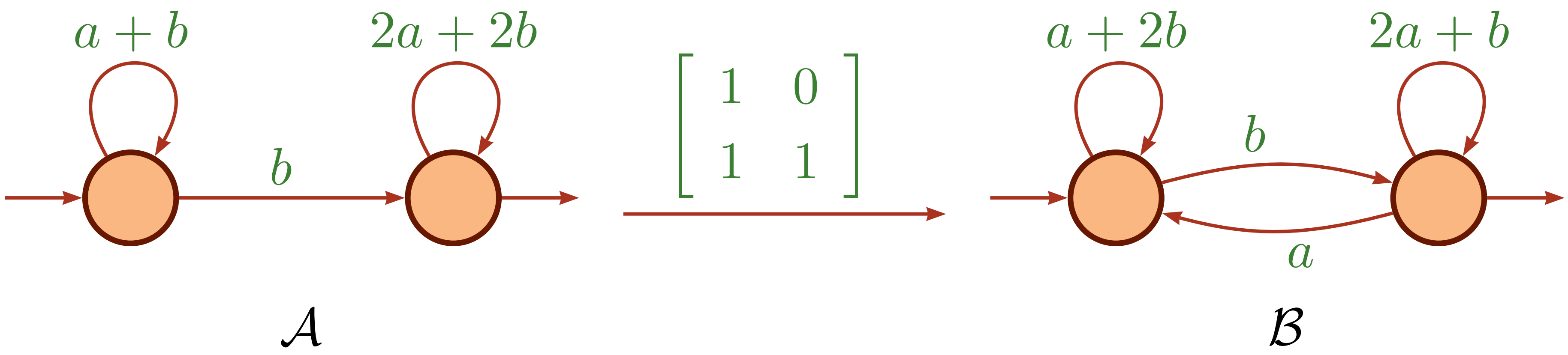
If $\mathcal{A} \stackrel{X}{\Longrightarrow} \mathcal{B}$, then



Theorem 2: Let $\mathcal{A}$ and $\mathcal{B}$ be | two trim automata
| two trim $\mathbb{N}$-automata

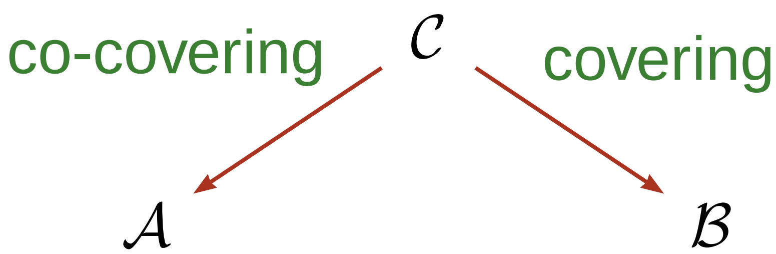
If $\mathcal{A} \stackrel{X}{\Longrightarrow} \mathcal{B}$, then

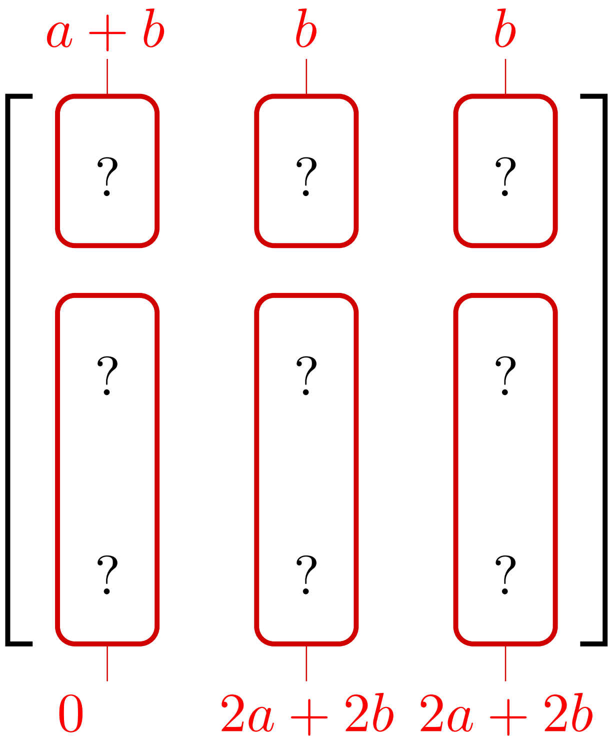




$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

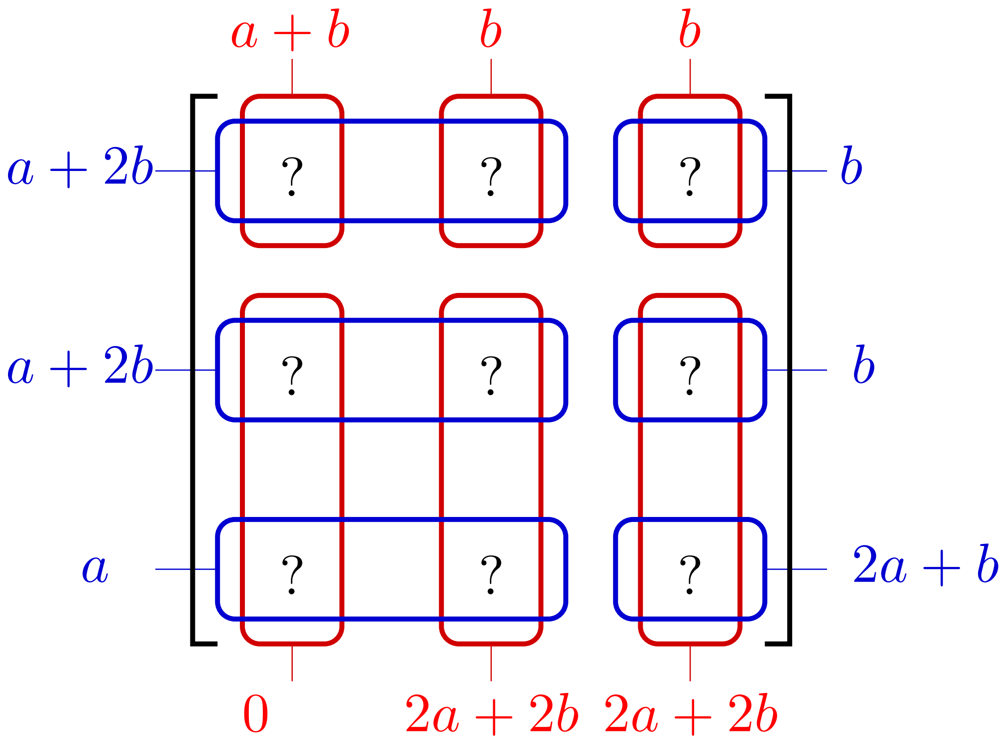
Let build $\mathcal{C} = (K, C, V)$ such that





$$\begin{bmatrix} a+b & b \\ 0 & 2a+2b \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} C$$

Conjugacy and coverings

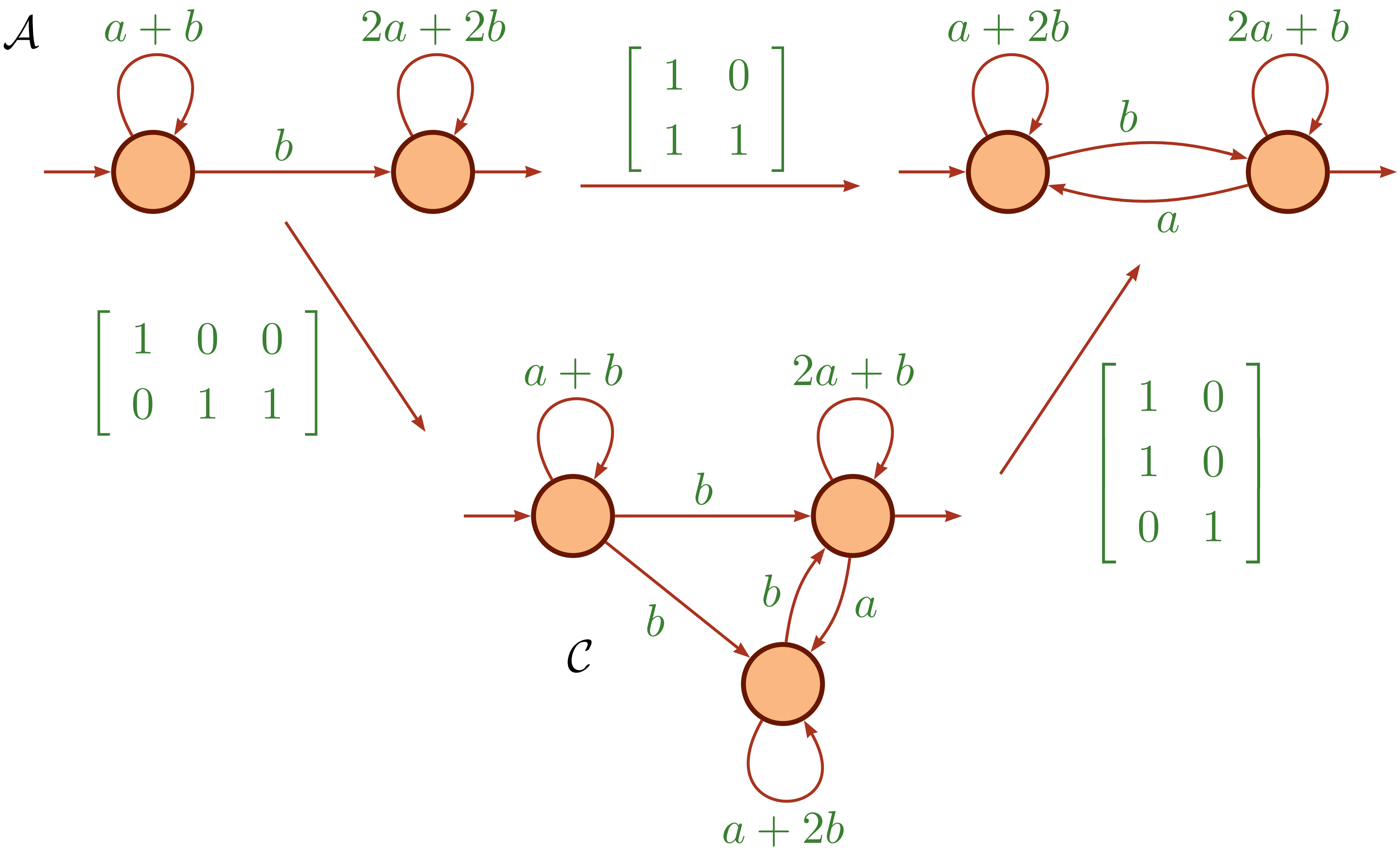


$$C\begin{bmatrix}1 & 0\\1 & 0\\0 & 1\end{bmatrix} = \begin{bmatrix}1 & 0\\1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}a+2b & b\\a & 2a+b\end{bmatrix}$$

Conjugacy and coverings

$$\begin{bmatrix} a+b & b & b \\ 0 & a+2b & b \\ 0 & a & 2a+2b \end{bmatrix}$$

Conjugacy and coverings



Equivalence and coverings

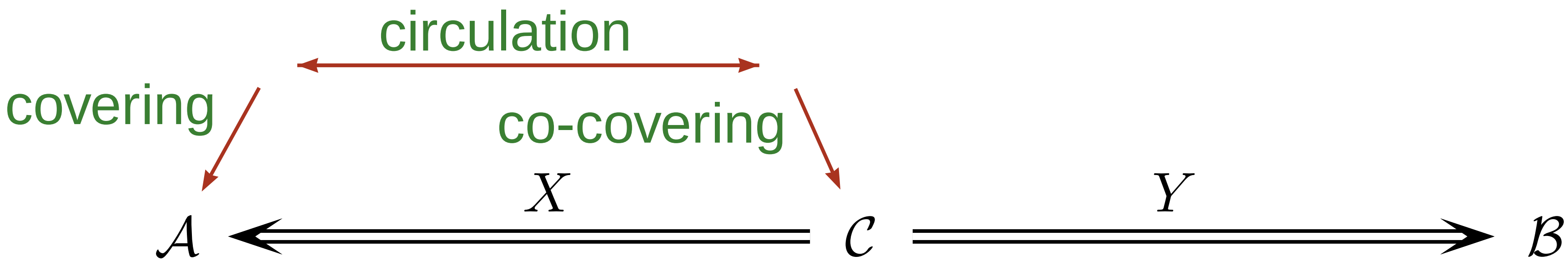
$|\mathcal{A}| = |\mathcal{B}|$

Theorem 1:

$$\mathcal{A} \xLeftarrow{\quad X \quad} \mathcal{C} \xRightarrow{\quad Y \quad} \mathcal{B}$$

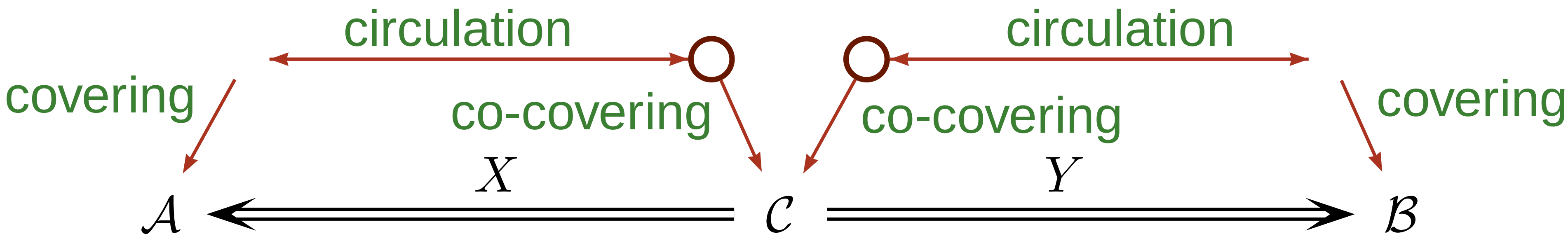
$|\mathcal{A}| = |\mathcal{B}|$

Theorem 2:



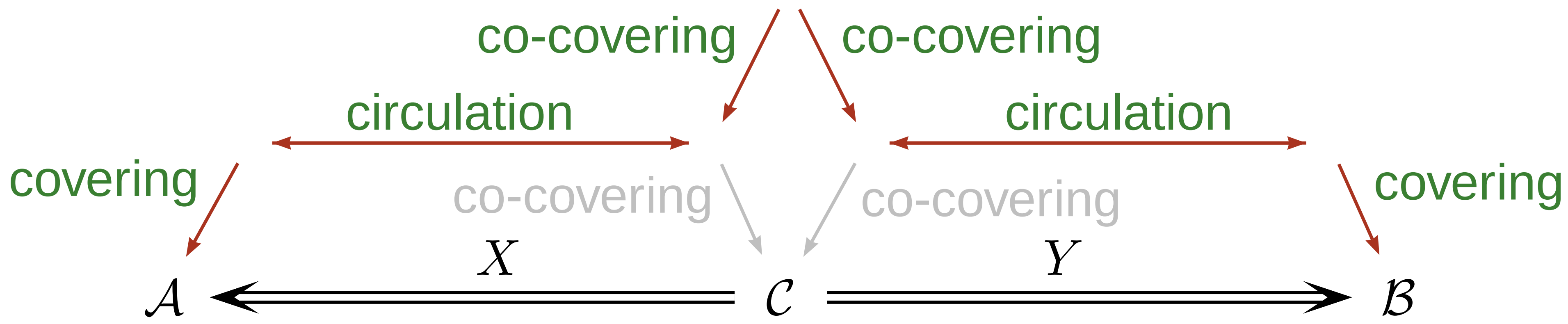
Equivalence and coverings

$|\mathcal{A}| = |\mathcal{B}|$



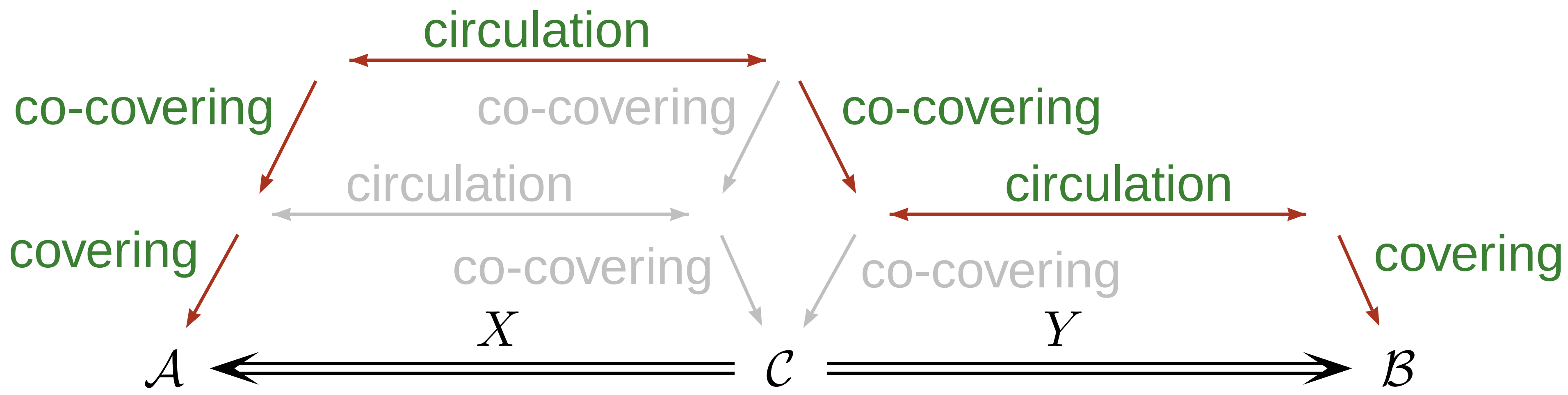
Equivalence and coverings

$|\mathcal{A}| = |\mathcal{B}|$



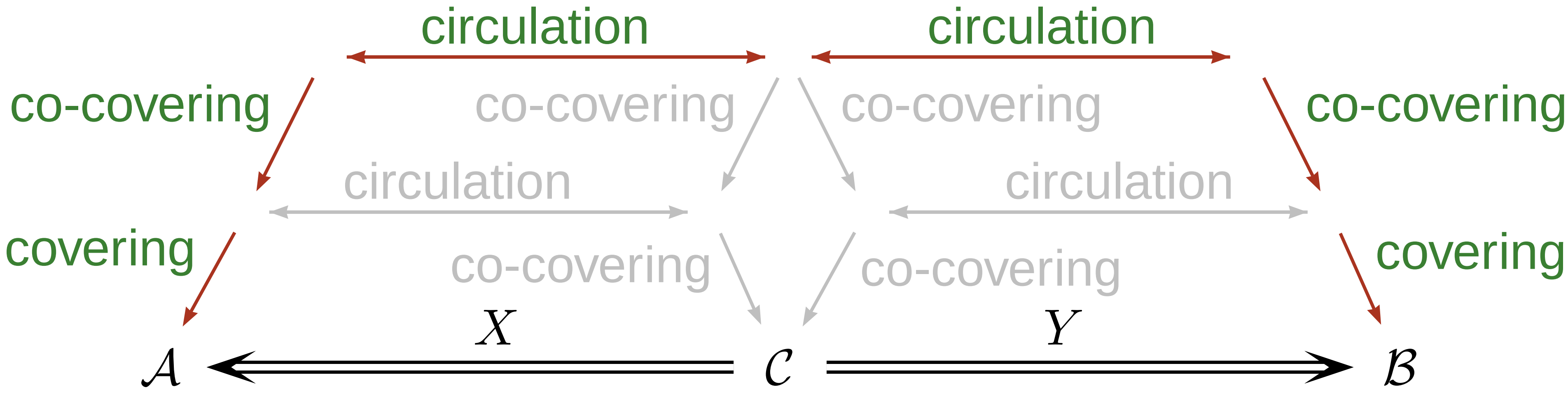
Equivalence and coverings

$|\mathcal{A}| = |\mathcal{B}|$



Equivalence and coverings

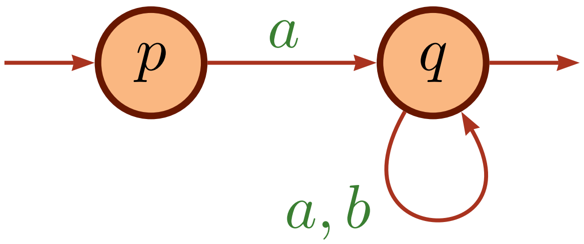
$|\mathcal{A}| = |\mathcal{B}|$

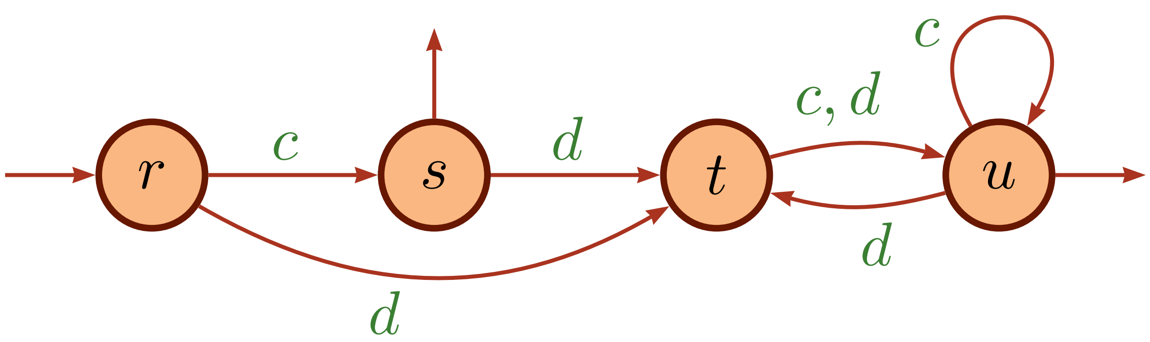


Application

Proposition: If two rational languages have the same growth function, there exists between them a letter-to-letter rational bijection.

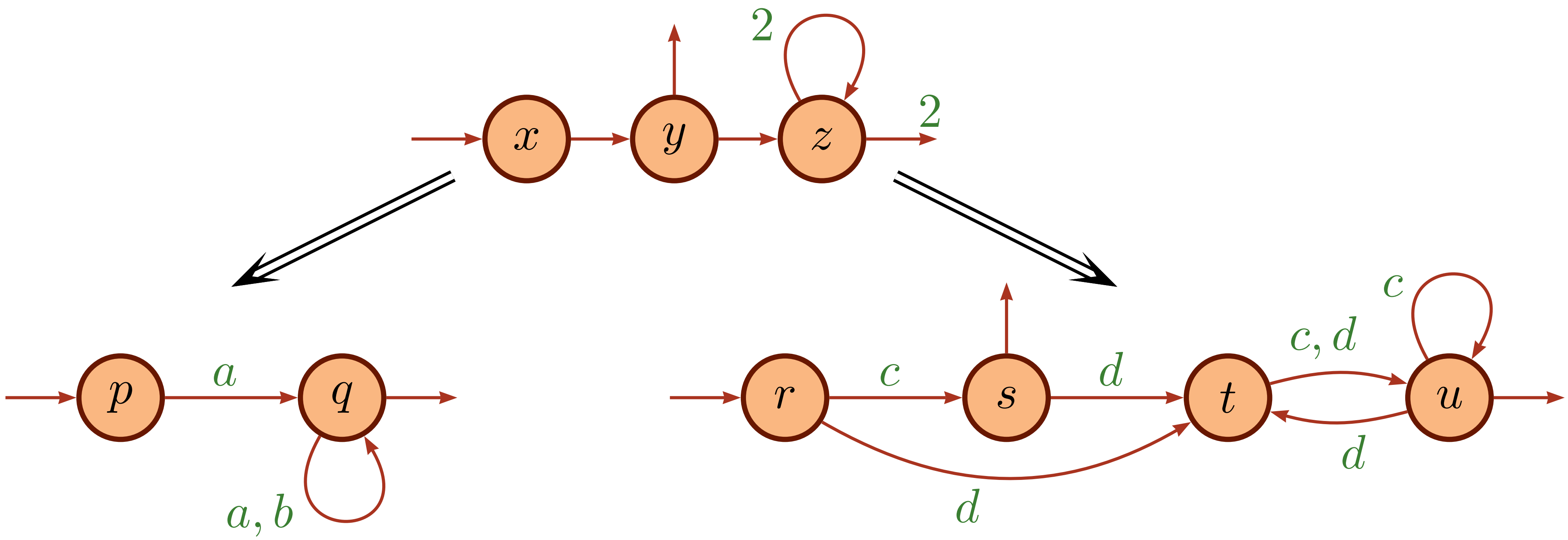
Example: $L_1 = a(a+b)^*$ and $L_2 = (c + dc + dd)^* \setminus cc(c+d)^*$:



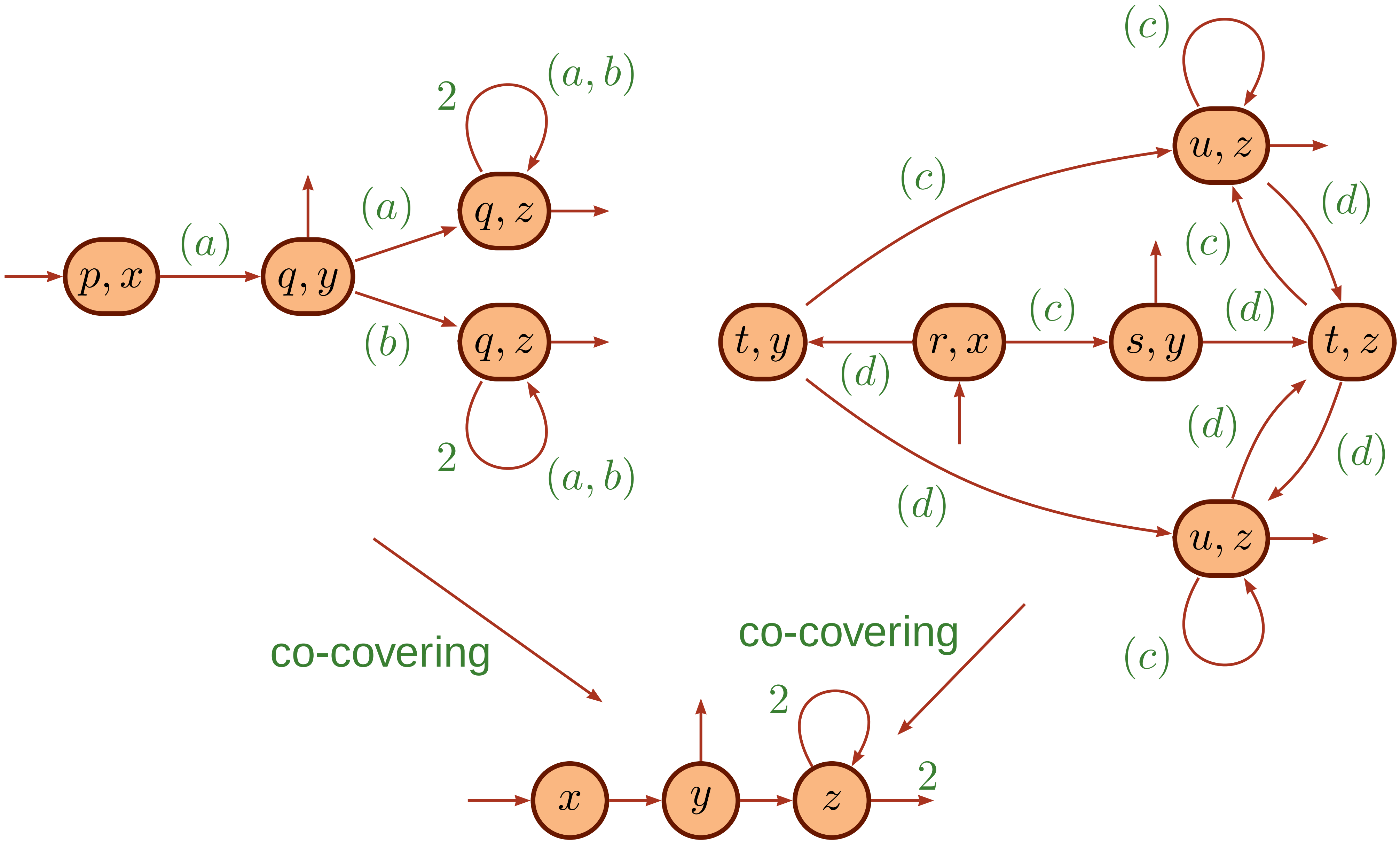


Proposition: If two rational languages have the same growth function, there exists between them a letter-to-letter rational bijection.

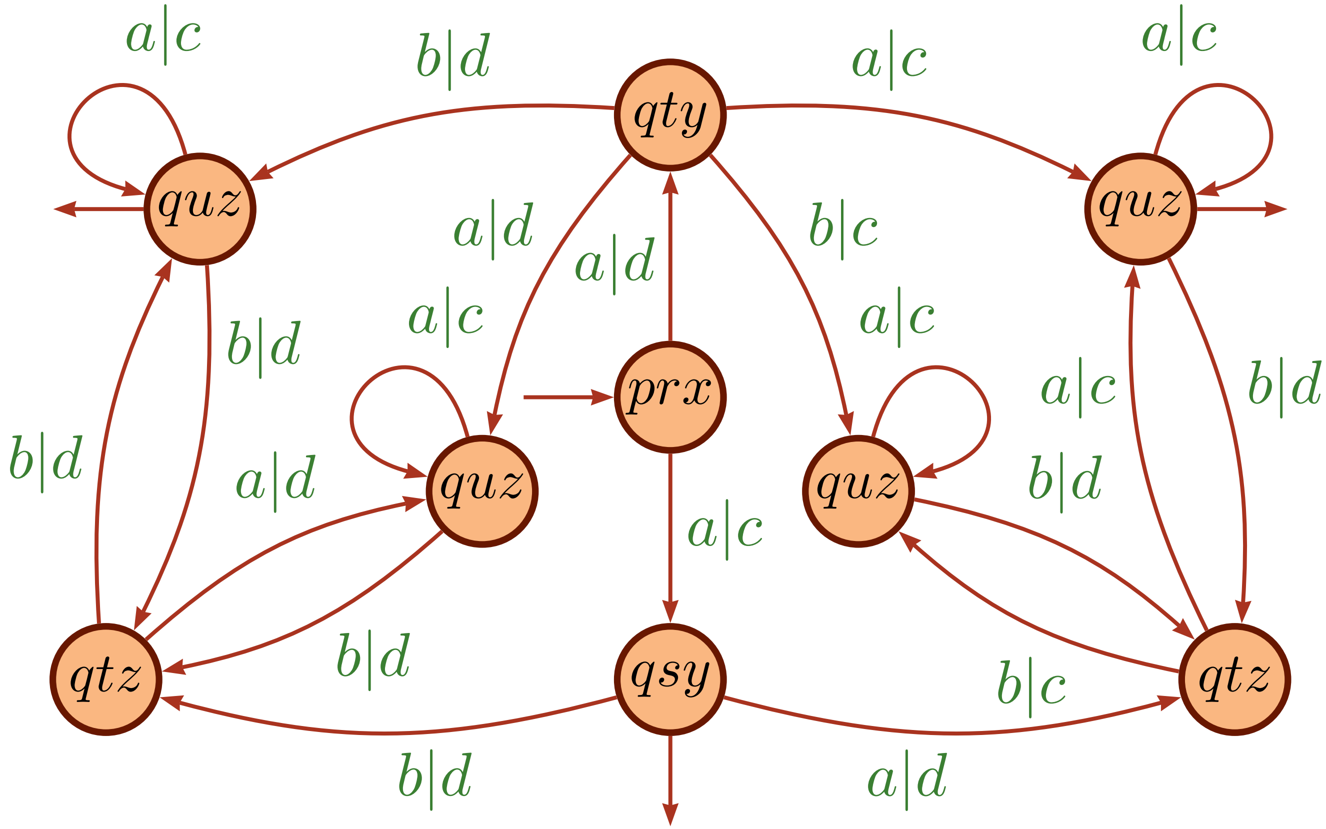
Example: $L_1 = a(a+b)^*$ and $L_2 = (c+dc+dd)^* \setminus cc(c+d)^*$:

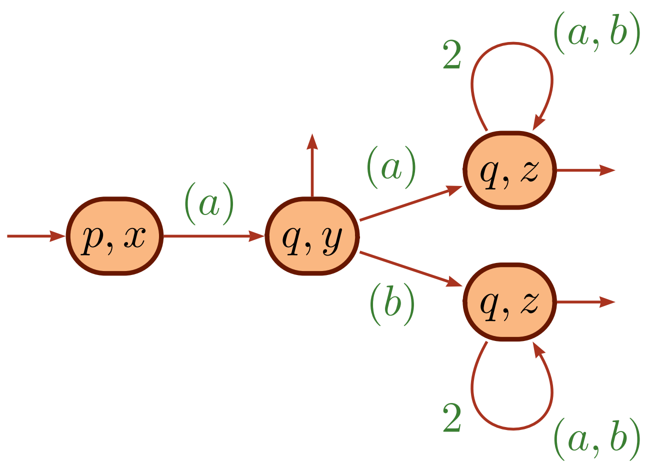


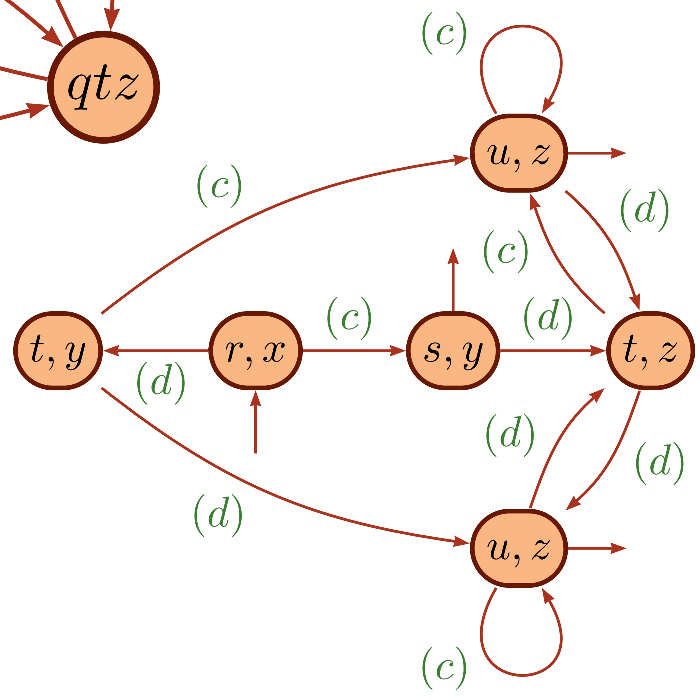
Application



Application







Conjugacy and dynamical systems

Finite Equivalence Theorem (Parry):
Two sofic subshifts are image by a *bloc-map finite-to-one* mapping of the same finite type subshift iff they have the same entropy.

proof (very sketchy):
Furstenberg Lemma: X, Y same entropy $\Rightarrow XF = FY$, $F \geqslant 0$, $F \neq 0$
$XF = FY \Rightarrow$ existence of *bloc-map finite-to-one* mappings

Conclusion

- Work in progress...
- What can we say about non functional transducers, (max/min,+) automata, *etc.* ?
- What is the link between the decidability of equivalence and the decidability of conjugacy ?