

Automata based verification over linearly ordered data domains

to be presented at STACS '11

Luc Segoufin
INRIA and ENS Cachan

Szymon Toruńczyk
University of Warsaw

Motivation

Motivation

Given a system which refers to a database and data values, verify properties of the system

AVIS

Car Hire Home Page Return to Booking Help

Car Hire Offers Avis Locations Products & Services Join Avis Preferred Business Rentals Help and Contacts UK Fleet Home Delivery Newsletter

1 When and Where 2 Your Car Choice 3 Price and Extras 4 Checkout

Avis car hire quote and booking

Find pickup location by
Search for

Return location
Same as pickup location [Change](#)

Rental Start Date/Time 09 00

Rental End Date/Time 09 00

Number of days

Avis Worldwide Discount (AWD) No.

[View, modify or cancel a booking](#) [Get a quote ▶](#)

Sign in [Create an account](#)

[Sign In](#) to your Online account/Avis Preferred

**Car hire delivered
to your front door**

**Click here to book
Home Delivery! ▶**

View, modify or cancel a booking [Get a quote ▶](#)



Car hire from £17 per day
Enjoy great discounts this Autumn. Get on the road and explore the wonderful places the UK has to offer.
Book Now!



Car hire from £15 per day
Explore Europe this Autumn with great discounts which will take you further so you can see more.
Book Now!



Car hire from £17 per day
Explore the world by taking advantage of our amazing Autumn Sale prices.
Book Now!

Motivation

Given a system which refers to a database and data values, verify properties of the system

AVIS

Car Hire Home Page Return to Booking Help

Car Hire Offers Avis Locations Products & Services Join Avis Preferred Business Rentals Help and Contacts UK Fleet Home Delivery Newsletter

1 When and Where 2 Your Car Choice 3 Price and Extras 4 Checkout Sign in Create an account

Avis car hire quote and booking

Find pickup location by
Search for

Return location
Same as pickup location [Change](#)

Rental Start Date/Time 09 00

Rental End Date/Time 09 00

Number of days

Avis Worldwide Discount (AWD) No.

[View, modify or cancel a booking](#)

Get a quote ▶

input values

change state

Car hire delivered to your front door

UK from £17 per day

Car hire from £17 per day

Enjoy great discounts this Autumn. Get on the road and explore the wonderful places the UK has to offer.

Book Now!

Europe £15 per day

Car hire from £15 per day

Explore Europe this Autumn with great discounts which will take you further so you can see more.

Book Now!

Car hire from £17 per day

Explore the world by taking advantage of our amazing Autumn Sale prices.

Book Now!

Motivation

Given a system which refers to a database and data values, verify properties of the system

AVIS

Car Hire Home Page Return to Booking Help

Car Hire Offers Avis Locations Products & Services Join Avis Preferred Business Rentals Help and Contacts UK Fleet Home Delivery Newsletter

1 When and Where 2 Your Car Choice 3 Price and Extras 4 Checkout

You are viewing vehicles available at Edinburgh Airport

Sign in Create an account

Sign in to your Online account/Avis Preferred

Your Booking

1 WHEN AND WHERE Change

Pickup : Edinburgh Airport 23/09/2010 19:00
Return : Edinburgh Airport 28/09/2010 19:00
Rental Days : 5

YOUR CAR CHOICE

3 PRICE AND EXTRAS

4 CHECKOUT

Empty Basket

Small Medium Large Select Series

Price Size Number of Luggage

 Small : Economy (Example of this range : Peugeot 207) Hide Info

Best price £155.65 per rental
Available to book now

Vehicle Features Air Bag - Driver Short Wheel Base Radio/Cassette

Driver age requirements
You must be at least 23 years old to hire this vehicle If you are under 25 years old a Young Driver Surcharge will apply Young Driver surcharge is £11/day + VAT; max £110

Credit card requirements
The number of Credit Cards required when you pick up this vehicle is: 1.

 Medium : Economy (Example of this range : Nissan Note 1.4) More Info

Best price £181.37 per rental
Available to book now

Motivation

Given a system which refers to a database and data values, verify properties of the system

AVIS

Car Hire Home Page Return to Booking Help

Car Hire Offers Avis Locations Products & Services Join Avis Preferred Business Rentals Help and Contacts UK Fleet Home Delivery Newsletter

1 When and Where 2 Your Car Choice 3 Price and Extras 4 Checkout

You are viewing vehicles available at Edinburgh Airport

Small Medium Large Select Series

Price Size Number of Luggage

 Small : Economy (Example of this range : Peugeot 207)
Best price £155.65 per rental
Available to book now

[Hide Info](#)

Vehicle Features
Air Bag - Driver
Short Wheel Base

Radio/Cassette

Driver age requirements
You must be at least 23 years old to hire this vehicle If you are under 25 years old a Young Driver Surcharge will apply Young Driver surcharge is £11/day + VAT; max £110

Credit card requirements
The number of Credit Cards required when you pick up this vehicle is: 1.

Medium : Economy (Example of this range : Nissan Note 1.4)

 Best price £181.37 per rental
Available to book now

[More Info](#)

Sign in [Create an account](#)

[Sign in](#) to your Online account/Avis Preferred

Your Booking

1 WHEN AND WHERE [Change](#)

Pickup : Edinburgh Airport 23/09/2010 19:00

Return : Edinburgh Airport 28/09/2010 19:00

Rental Days : 5

YOUR CAR CHOICE

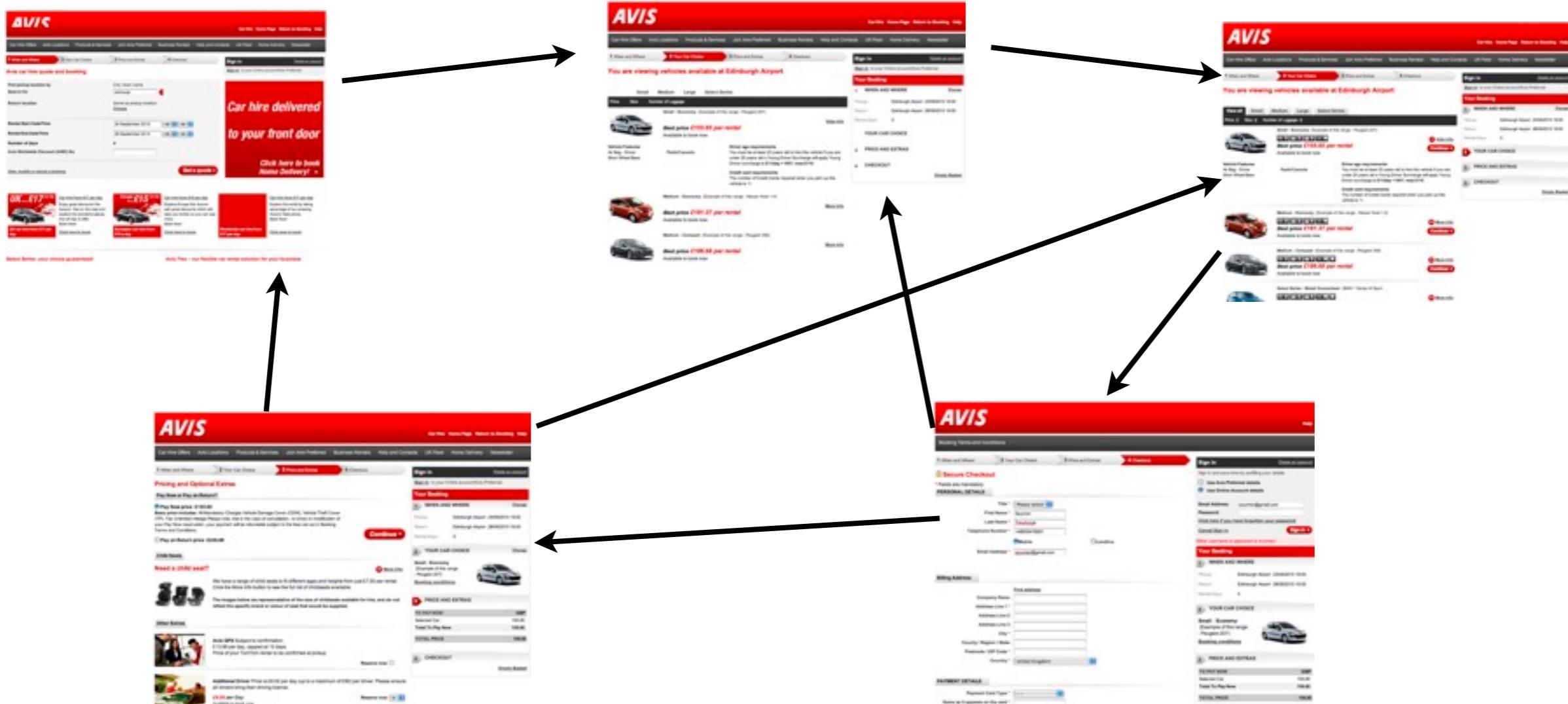
3 PRICE AND EXTRAS

4 CHECKOUT

[Empty Basket](#)

Motivation

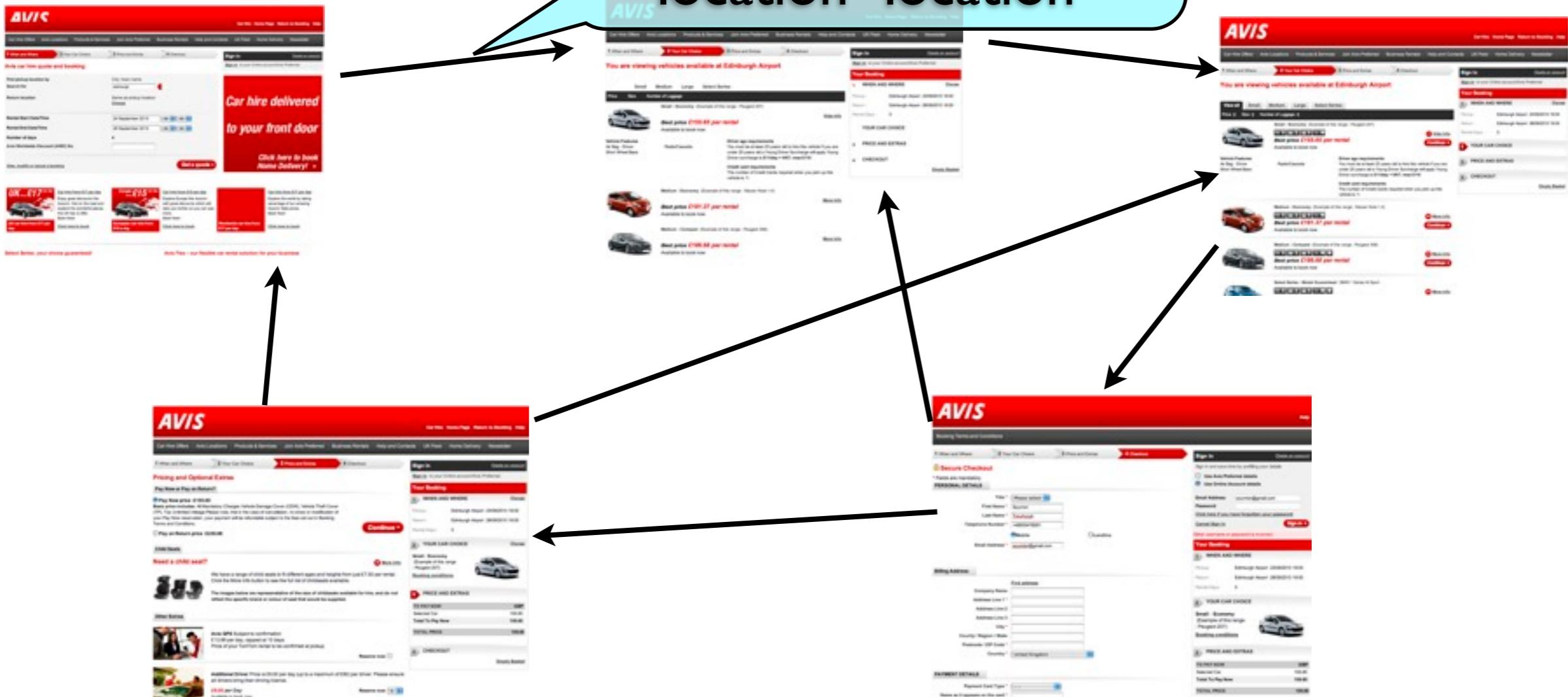
Given a system which refers to a database and data values, verify properties of the system



Motivation

Given a system which refers to a database and data values, verify properties of the system

start_date=start_date'
end_date=end_date'
location=location'



Motivation

Given a system which refers to a database and data values, verify properties of the system

$\text{start_date} = \text{start_date}'$
 $\text{end_date} = \text{end_date}'$
 $\text{location} = \text{location}'$

This screenshot shows the Avis website interface. It displays a search form for car hire, including fields for pickup location (Edinburgh), drop-off location (Edinburgh), and dates (24 September 2014 - 25 September 2014). A prominent red banner states "Car hire delivered to your front door". Below the form, there are three car models listed with their prices: Small (£199.85 per day), Medium (£209.37 per day), and Large (£209.89 per day). At the bottom, there are links for "Secure Checkout" and "Check Out".

This screenshot shows the Avis website displaying vehicles available at Edinburgh Airport. It lists three car models: Small (Small Hatchback, £199.85 per day), Medium (Medium Hatchback, £209.37 per day), and Large (Large Hatchback, £209.89 per day). Each listing includes a "View Details" button. The page also features sections for "Vehicles & Services" and "Business Rates".

This screenshot shows the Avis website with a large blue callout box containing the inequality $\text{pricemin} \leq \text{price} \leq \text{pricemax}$. The page displays a list of vehicles with their respective daily rates: Small (£199.85), Medium (£209.37), and Large (£209.89). The background shows the Avis logo and some vehicle images.

This screenshot shows the Avis website's "Pricing and Optional Extras" section. It includes a "Pay Now or Pay on Return?" checkbox, a "Need a CDW?" checkbox, and a "Additional Driver" section. Below these, there is a "PRICE AND EXTRAS" table with rows for "CDW INSURANCE" (Small: £10.00, Medium: £10.00, Large: £10.00) and "TOTAL PRICE" (£209.85). At the bottom, there are "CheckOut" and "Secure Checkout" buttons.

This screenshot shows the Avis website's "Secure Checkout" process. It includes a "PERSONAL DETAILS" form with fields for First Name, Last Name, Middle Name, Home Phone Number, and Email Address. Below this is a "Billing Address" form with fields for Company Name, Address Line 1, Address Line 2, Address Line 3, City, County, Postcode, and Country. At the bottom, there is a "PAYMENT DETAILS" section with a "Payment Card Type" dropdown set to "Credit or Debit Card".

Motivation

Given a system which refers to a database and data values, verify properties of the system

$\text{start_date} = \text{start_date}'$
 $\text{end_date} = \text{end_date}'$
 $\text{location} = \text{location}'$

$\text{pricemin} \leq \text{price}$
 $\leq \text{pricemax}$

CAR(car_id, price)

This screenshot shows the Avis website interface for searching vehicles. It includes fields for 'From' and 'To' dates, 'Location', and 'Car Type'. A prominent red button at the bottom right says 'Search Now!'. Below the search bar, there's a message: 'Car hire delivered to your front door'.

This screenshot shows the results of a car search. It lists several vehicle options with their respective prices: Small (Basic) at £199.85 per day, Medium (Economy) at £219.37 per day, and Large (Executive) at £239.89 per day. Each listing includes a small image of the car and a 'View Details' link.

This screenshot shows a detailed view of a car rental offer. It includes the car type ('Small (Basic)'), price ('£199.85 per day'), and a summary of included services ('Basic (Basic) includes 100 miles'). There are also links for 'Check Out' and 'Book Now'.

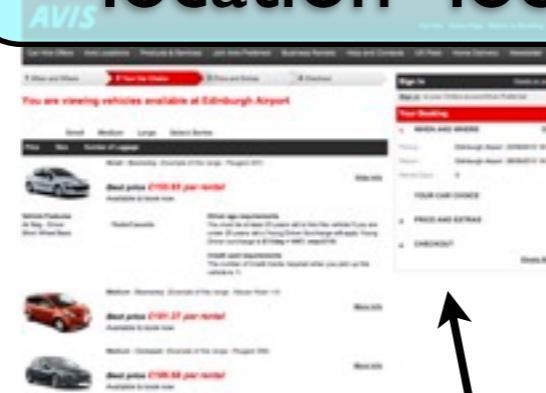
This screenshot shows the 'Pricing and Optional Extras' step of the booking process. It displays the base price (£199.85), optional extras like 'Pay Now or Pay at Return', and other service options like 'Need a CHUFF seat?'.

This screenshot shows the final booking summary. It includes the total price (£239.89), payment method ('Credit Card'), and a breakdown of the cost ('£199.85 per day', '£40.00 Total', '£0.00 Total'). The payment section shows a card icon and the amount '£239.89'.

Motivation

Given a system which refers to a database and data values, verify properties of the system

$\text{start_date} = \text{start_date}'$
 $\text{end_date} = \text{end_date}'$
 $\text{location} = \text{location}'$



$\text{CAR}(\text{car_id}, \text{price})$

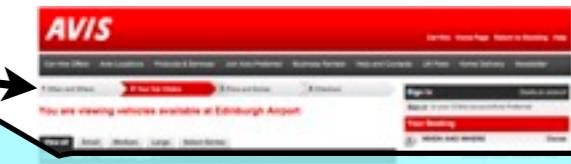


$\text{PAYMENT}(\text{user_id}, \text{car_id},$
 $\text{price}, \text{order_id})$

Motivation

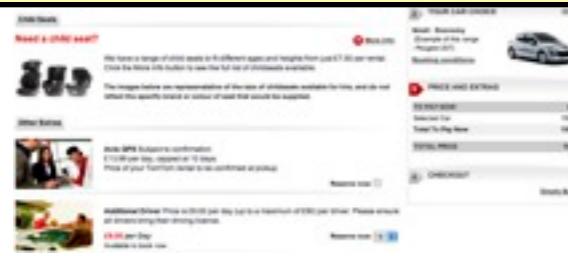
Given a system which refers to a database and data values, verify properties of the system

$\text{start_date} = \text{start_date}'$
 $\text{end_date} = \text{end_date}'$
 $\text{location} = \text{location}'$



$\text{pricemin} < \text{price}$

finally, a car is rented and

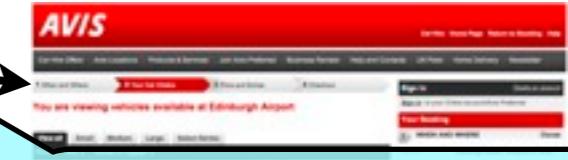
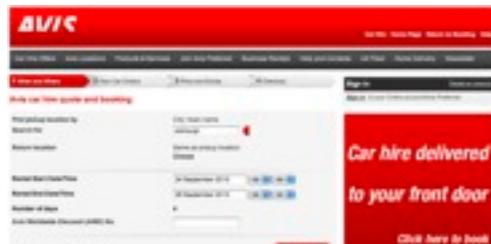


$\text{PAYMENT}(\text{user_id}, \text{car_id}, \text{price}, \text{order_id})$

Motivation

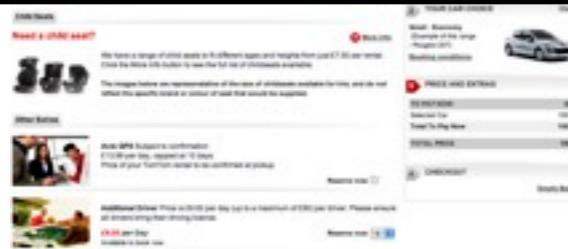
Given a system which refers to a database and data values, verify properties of the system

$\text{start_date} = \text{start_date}'$
 $\text{end_date} = \text{end_date}'$
 $\text{location} = \text{location}'$



$\text{pricemin} < \text{price}$

finally, a car is rented and
• no payment was received

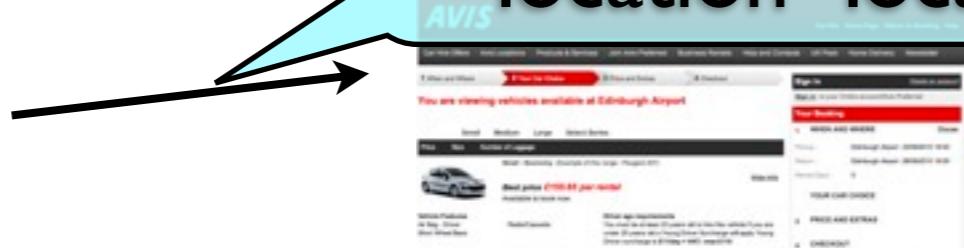


PAYMENT(user_id,car_id,
price,order_id)

Motivation

Given a system which refers to a database and data values, verify properties of the system

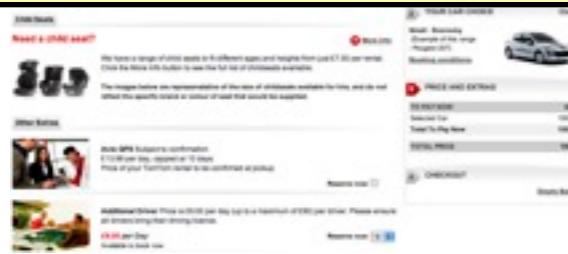
$\text{start_date} = \text{start_date}'$
 $\text{end_date} = \text{end_date}'$
 $\text{location} = \text{location}'$



$\text{pricemin} < \text{price}$

finally, a car is rented and

- no payment was received
- a payment was received but $\text{payed} = 0$

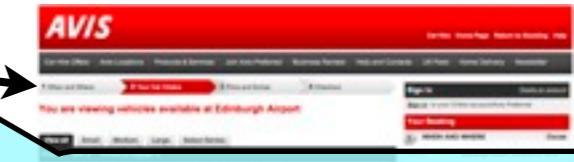


$\text{PAYMENT}(\text{user_id}, \text{car_id}, \text{price}, \text{order_id})$

Motivation

Given a system which refers to a database and data values, verify properties of the system

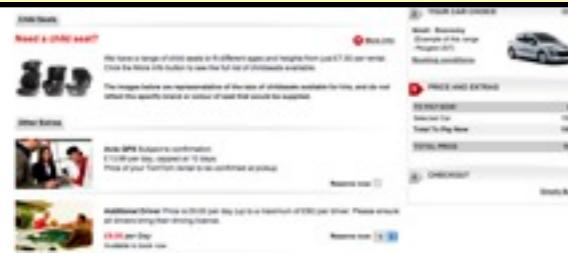
$\text{start_date} = \text{start_date}'$
 $\text{end_date} = \text{end_date}'$
 $\text{location} = \text{location}'$



$\text{pricemin} < \text{price}$

finally, a car is rented and

- no payment was received
- a payment was received but $\text{payed} = 0$
- there was a payment, but $\text{payed} < 100 \text{ & } \text{end_date} > 2050$

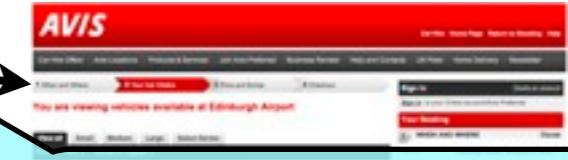
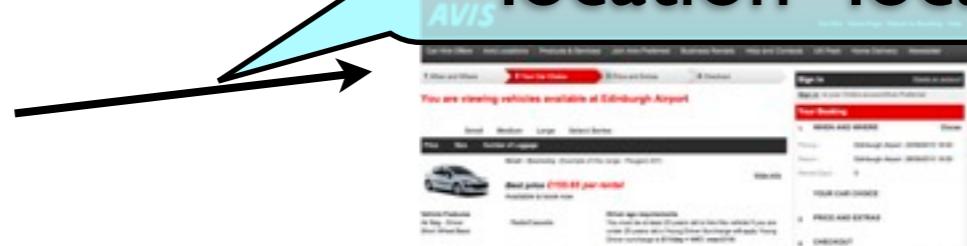


$\text{PAYMENT}(\text{user_id}, \text{car_id}, \text{price}, \text{order_id})$

Motivation

Given a system which refers to a database and data values, verify properties of the system

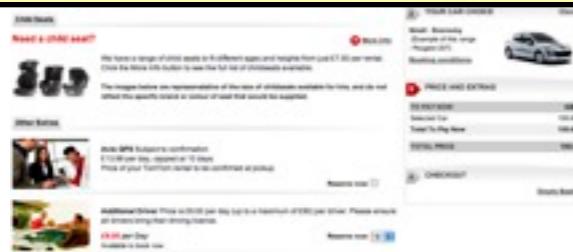
$\text{start_date} = \text{start_date}'$
 $\text{end_date} = \text{end_date}'$
 $\text{location} = \text{location}'$



$\text{pricemin} < \text{price}$

finally, a car is rented and

- no payment was received
- a payment was received but $\text{payed} = 0$
- there was a payment, but $\text{payed} < 100$ & $\text{end_date} > 2050$
- there was a payment, but $\text{payed} < \text{car_price}$



$\text{PAYMENT}(\text{user_id}, \text{car_id}, \text{price}, \text{order_id})$

Outline

1. Define automata model
2. Analyze automata over dense orders
3. Analyze automata over discrete orders
4. Add database and infinite runs

Extended automaton

a definition that captures timed automata, or vector addition systems

D – fixed domain

Q – finite set of states

X – finite set of variables

$D^X = \{ v: X \rightarrow D \}$ – space of variable valuations

$Q \times D^X$ – space of configurations

restricted
form

$\delta \subseteq (Q \times D^X) \times (Q \times D^X)$ – set of allowed transitions

$I \subseteq (Q \times D^X)$ – set of initial configurations

$F \subseteq (Q \times D^X)$ – set of final configurations

Examples

- *Vector Addition System:* $D = \mathbb{N}$

$$(q, v) \xrightarrow{v' = v + w} (q', v')$$

- *Timed Automata:* $D = \mathbb{R}$

$$(q, v) \xrightarrow{\begin{array}{l} c_1 < 2 \\ c_2 = 0 \end{array}} (q', v')$$

- *(Lossy) channel system:* $D = \{a+b\}^*$

$$(q, v) \xrightarrow{\begin{array}{l} first_a(c_1) \\ c_1 = c_1 \cdot b \end{array}} (q', v')$$

Our setting (without the database)

The domain:

$$\mathcal{D} = \langle D, <, P_1, P_2, P_3, \dots, P_l \rangle$$

\begin{array}{ll} \text{linearly ordered set} & \text{unary predicates} \\[-1ex] \swarrow & \searrow \\[-1ex] & \text{(subsets of } D) \end{array}

Examples

$$\langle \mathbb{N}, <, 0, 100, P_{even}, P_{prime} \rangle$$

$$\langle \mathbb{Q}, <, 0, 100, P_{integer}, P_{<\pi} \rangle$$

$$\langle \{a+b\}^*, <_{lex}, P_{(ab)^*} \rangle$$

Transitions:

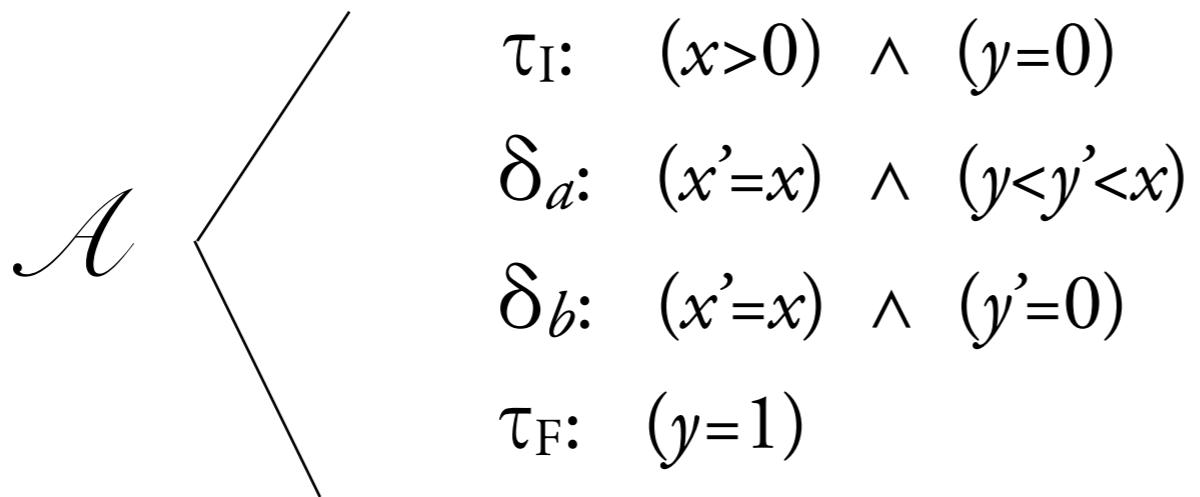
$$I, F \subseteq (Q \times D^X), \quad \delta \subseteq (Q \times D^X) \times (Q \times D^X)$$

are specified by quantifier free formulas over \mathcal{D}

Example D -automaton \mathcal{A}

x,y – variables of \mathcal{A}

no states



Example D -automaton \mathcal{A}

\mathcal{A}

x, y – variables of \mathcal{A}

no states



y

x

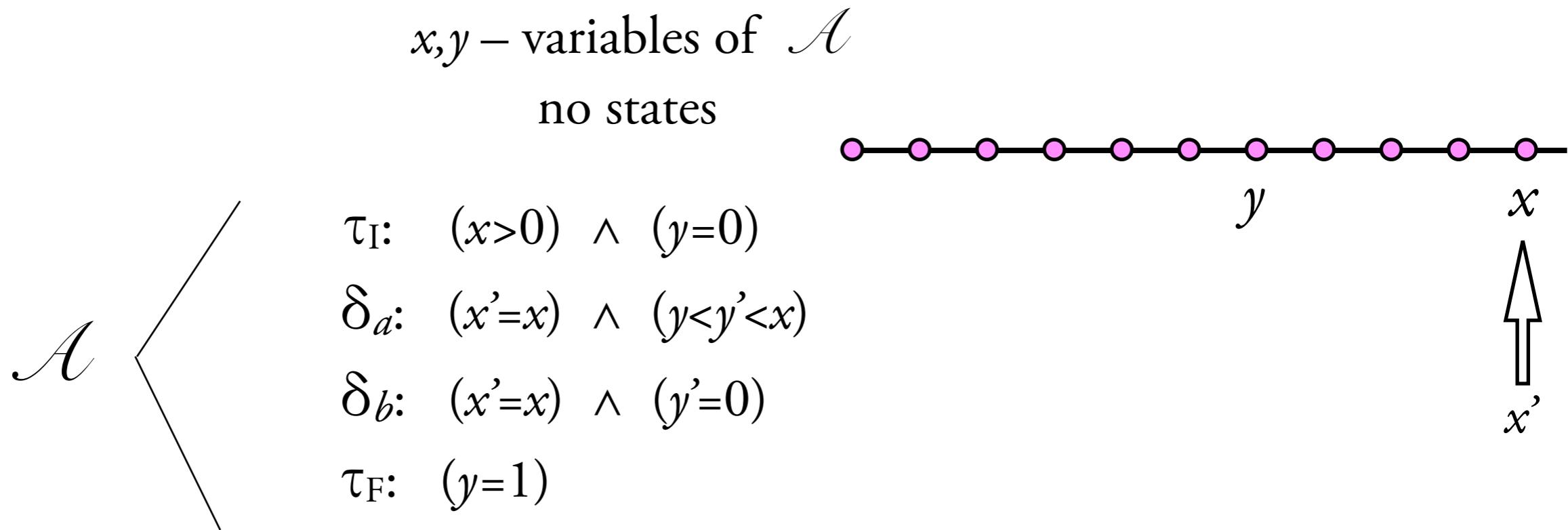
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

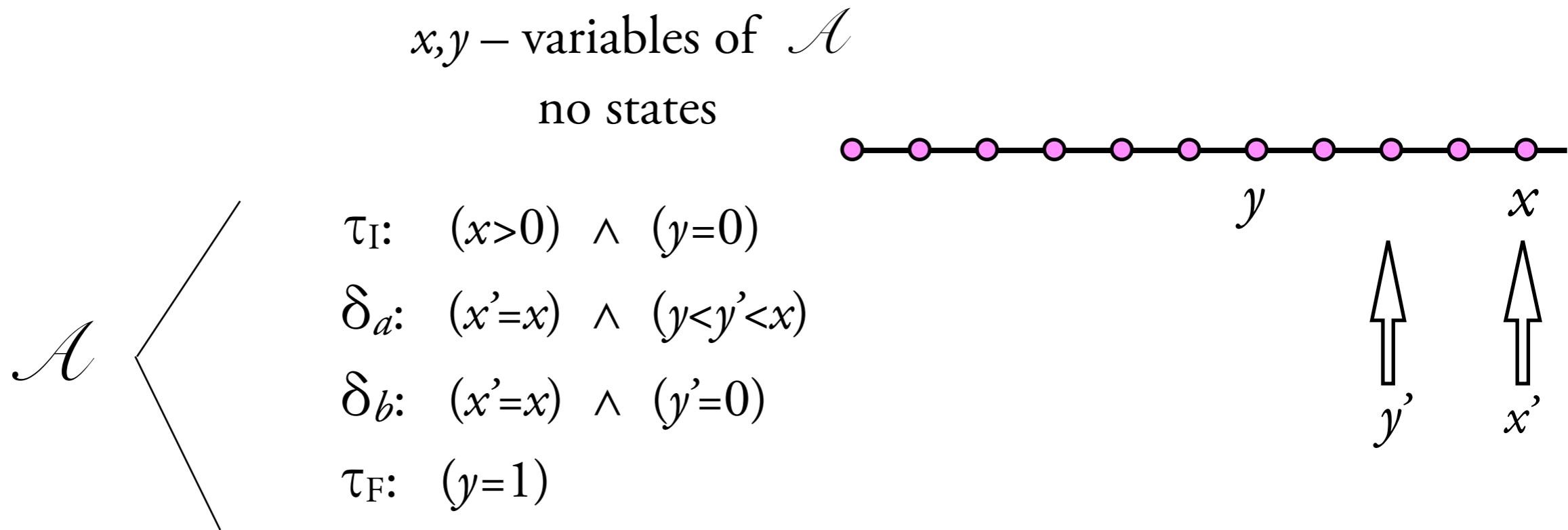
$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

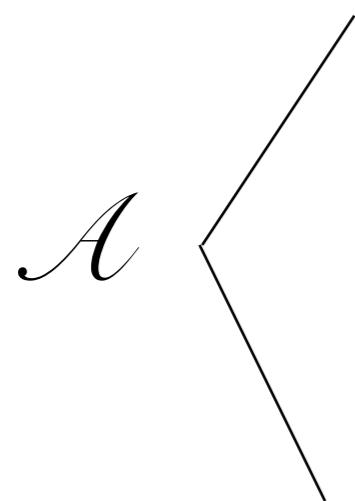
Example D -automaton \mathcal{A}



Example D -automaton \mathcal{A}



Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

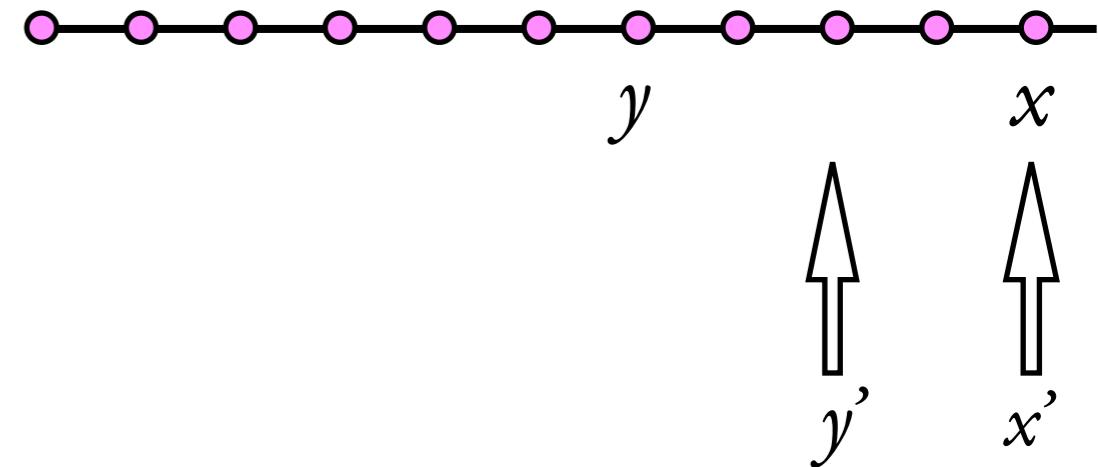
no states

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

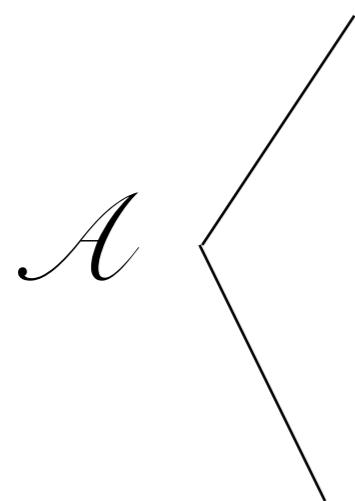
$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



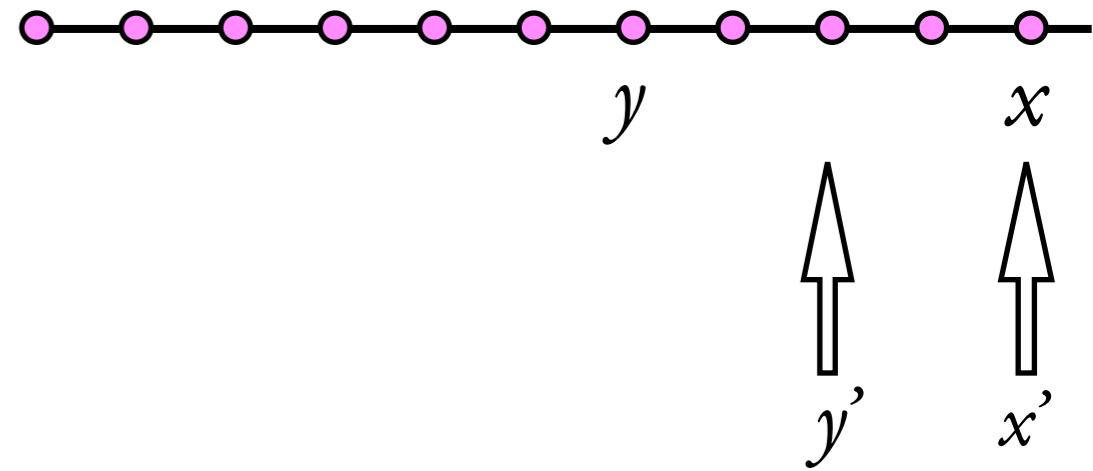
a finite run over \mathbb{Q} :

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

no states



$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

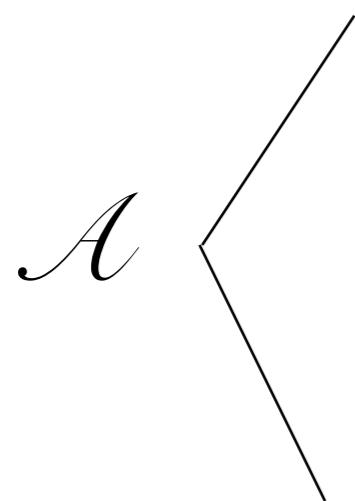
$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

a finite run over \mathbb{Q} :

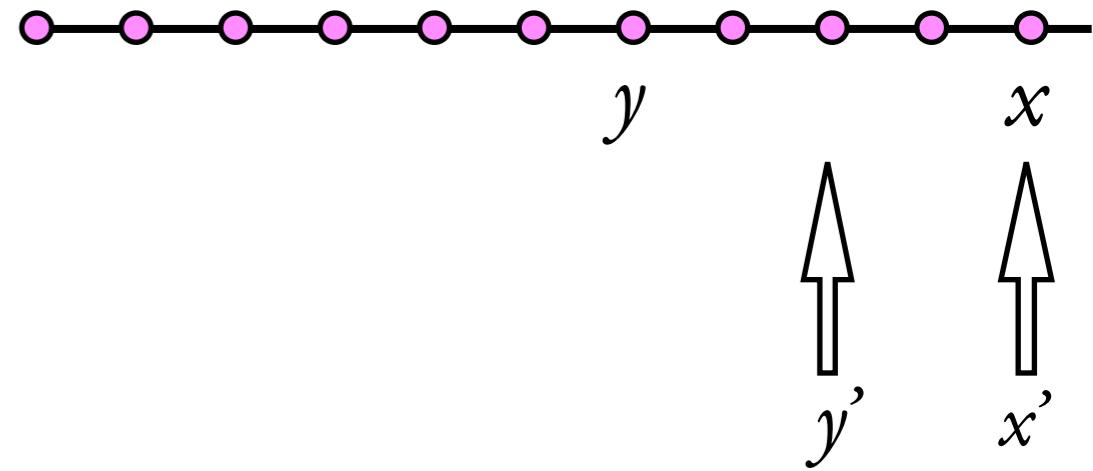
$a a a b a b a a a a a$

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

no states



$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

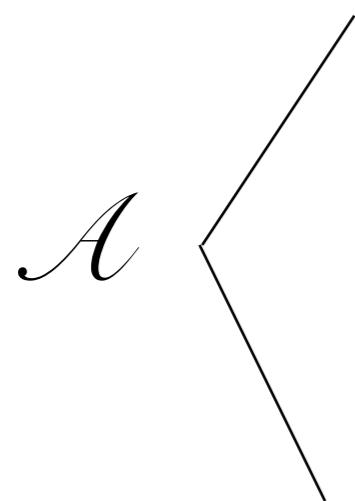
a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

x

y

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

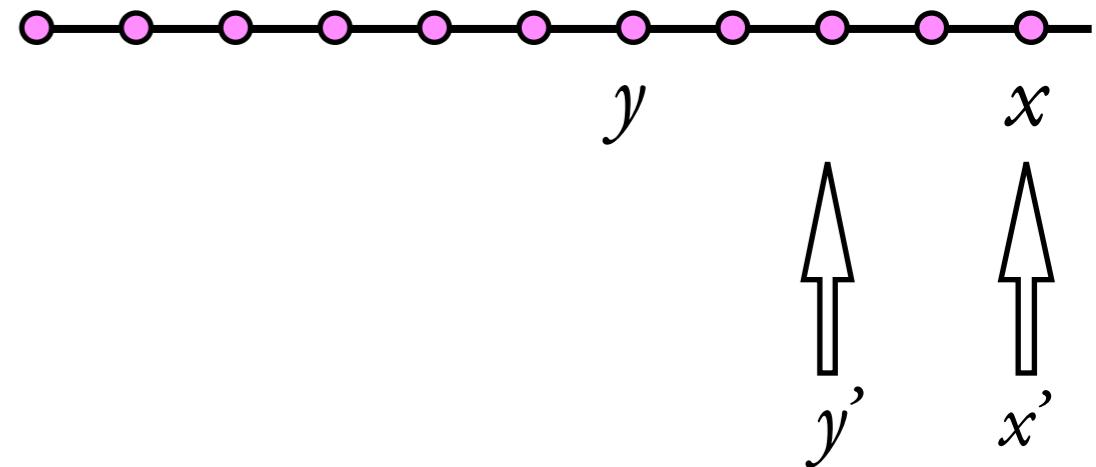
no states

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



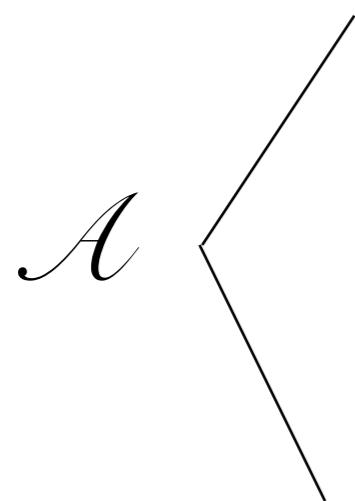
a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

x 5

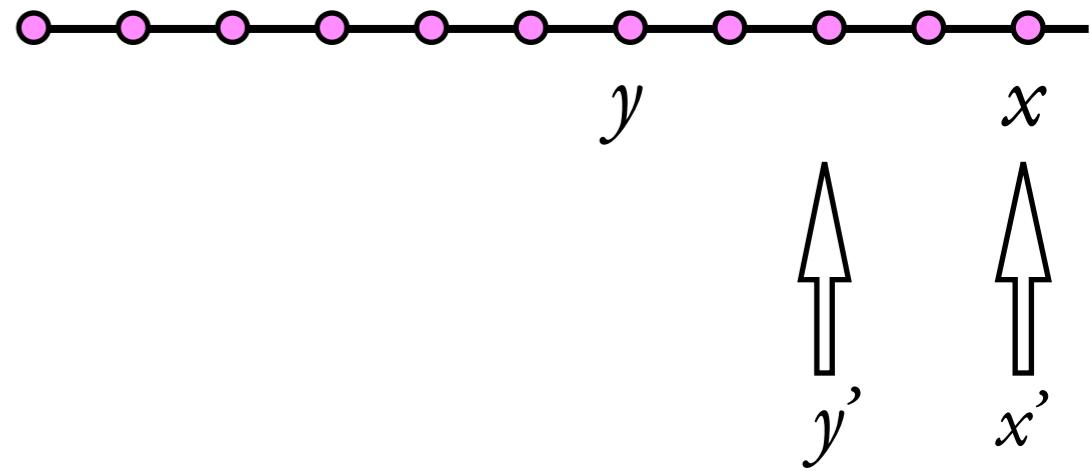
y 0

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

no states



$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

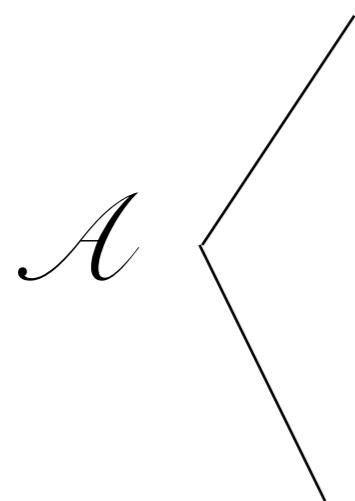
a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

$x \quad 5 \quad 5$

$y \quad 0 \quad 1$

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

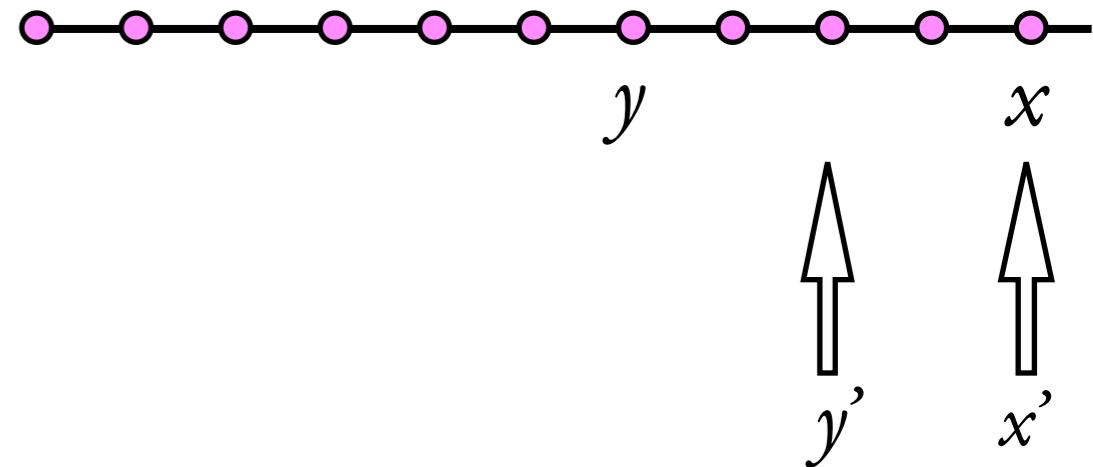
no states

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



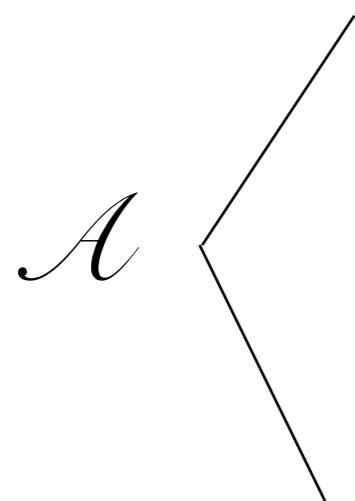
a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

$x \quad 5 \ 5 \ 5$

$y \quad 0 \ 1 \ 2$

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

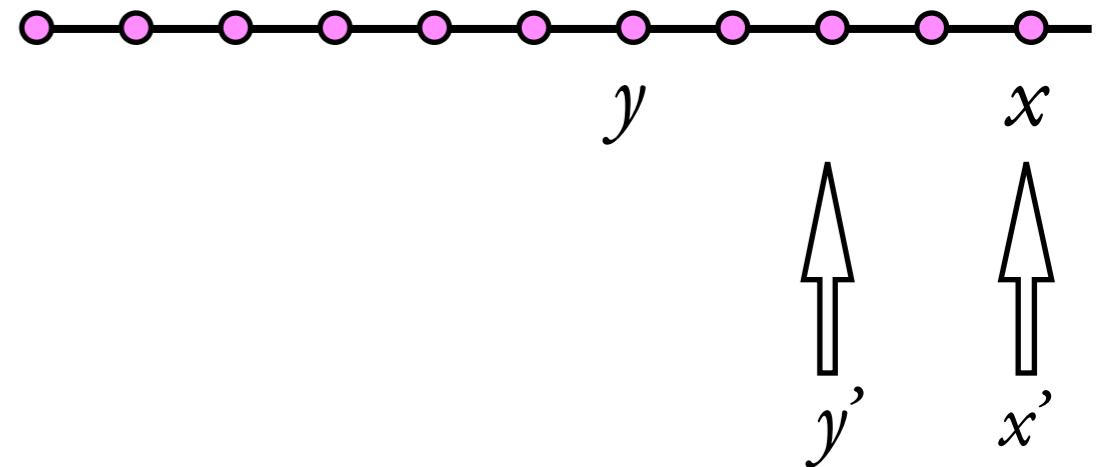
no states

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



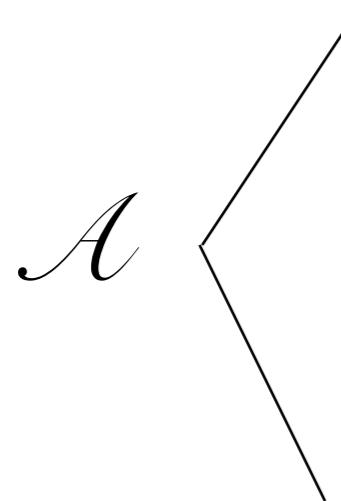
a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

$x \quad 5 \ 5 \ 5 \ 5$

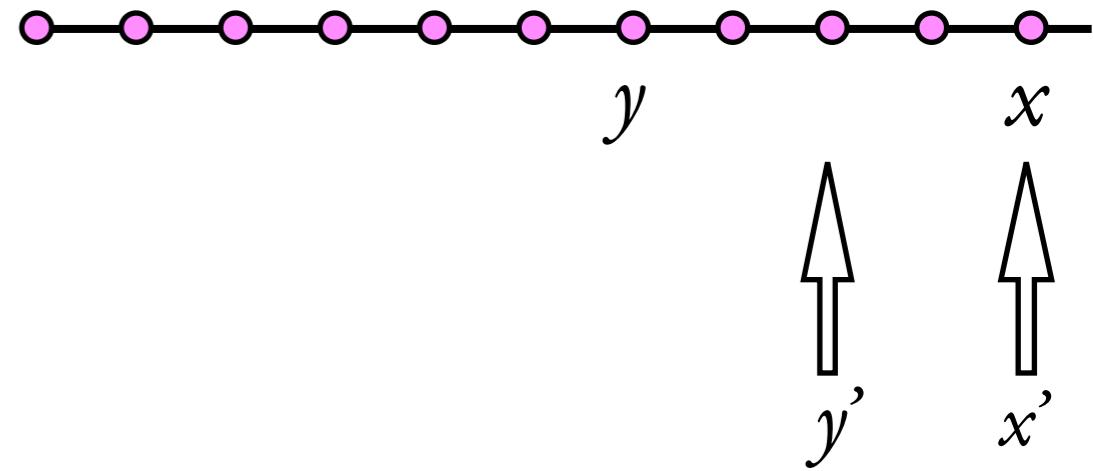
$y \quad 0 \ 1 \ 2 \ 4$

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

no states



$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

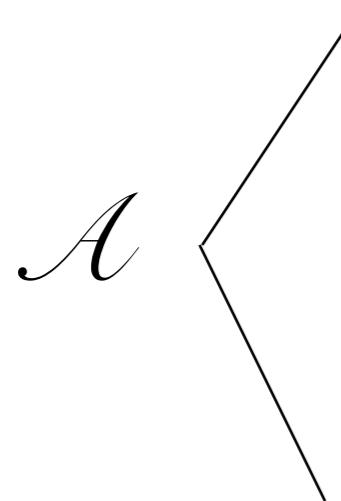
a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

$x \quad 5 \ 5 \ 5 \ 5 \ 5$

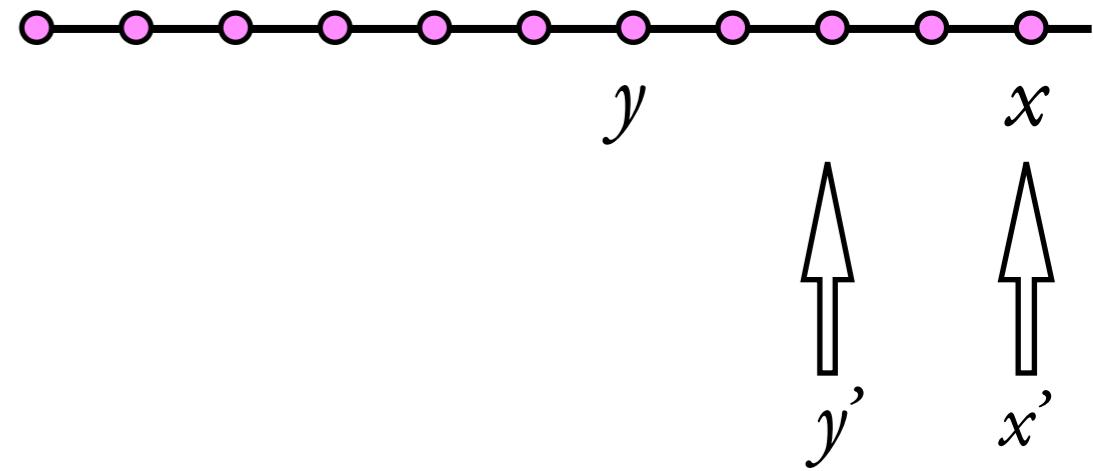
$y \quad 0 \ 1 \ 2 \ 4 \ 0$

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

no states



$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

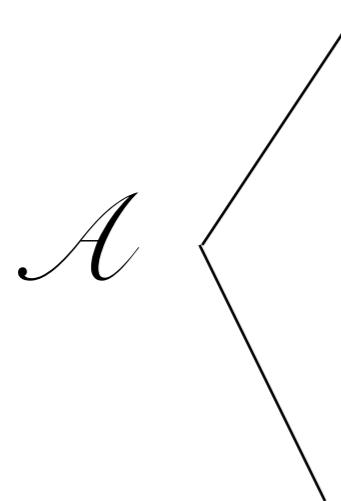
a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

$x \quad 5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5$

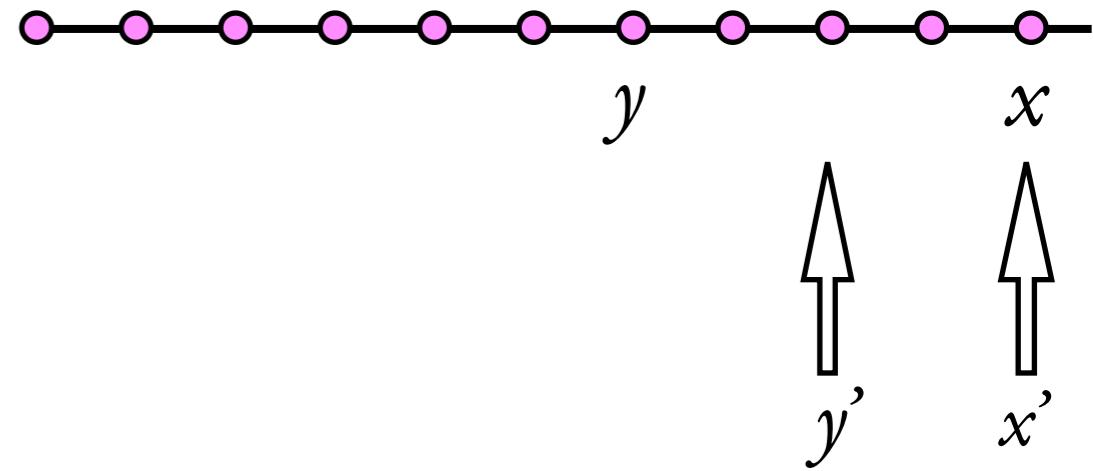
$y \quad 0 \ 1 \ 2 \ 4 \ 0 \ 3$

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

no states



$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

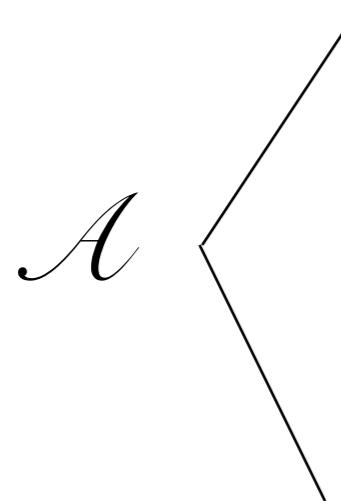
a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

$x \quad 5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5$

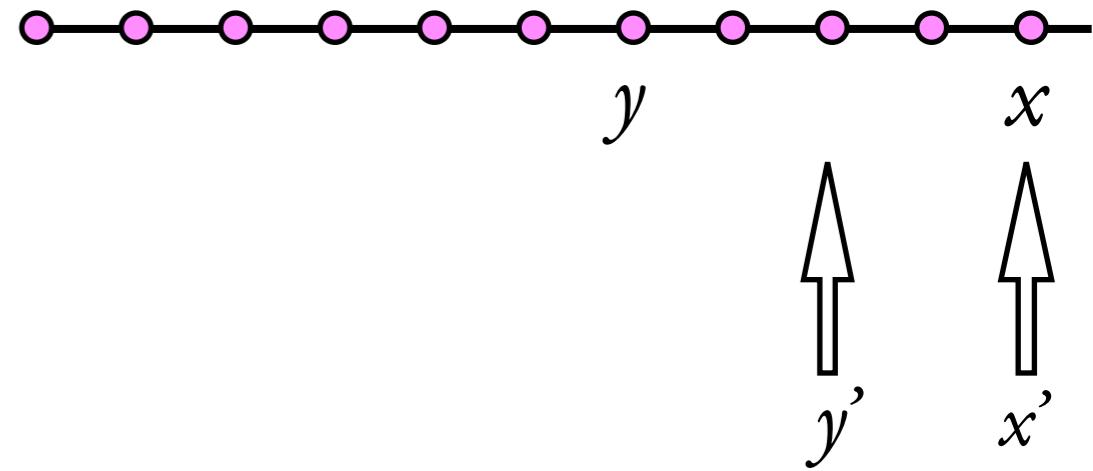
$y \quad 0 \ 1 \ 2 \ 4 \ 0 \ 3 \ 0$

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

no states



$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

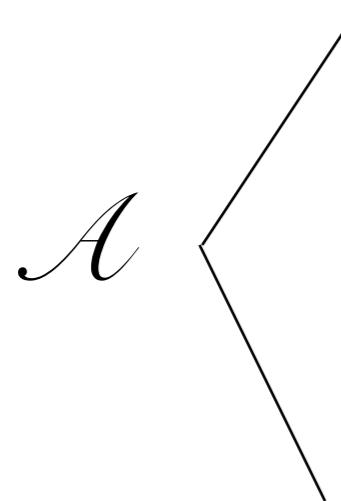
a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

$x \quad 5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5$

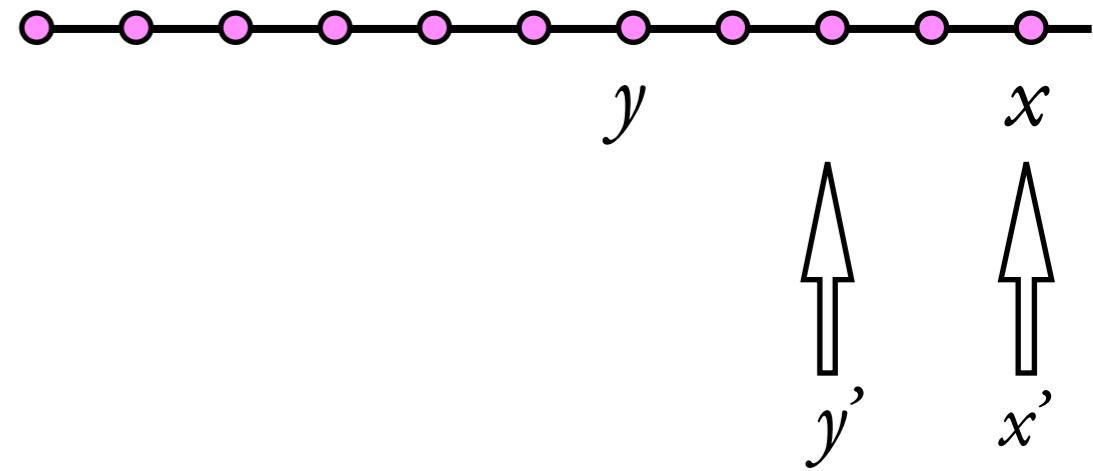
$y \quad 0 \ 1 \ 2 \ 4 \ 0 \ 3 \ 0 \ \frac{1}{4}$

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

no states



$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

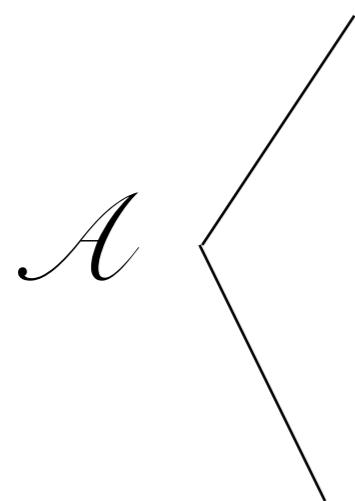
a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

x	5	5	5	5	5	5	5	5
-----	---	---	---	---	---	---	---	---

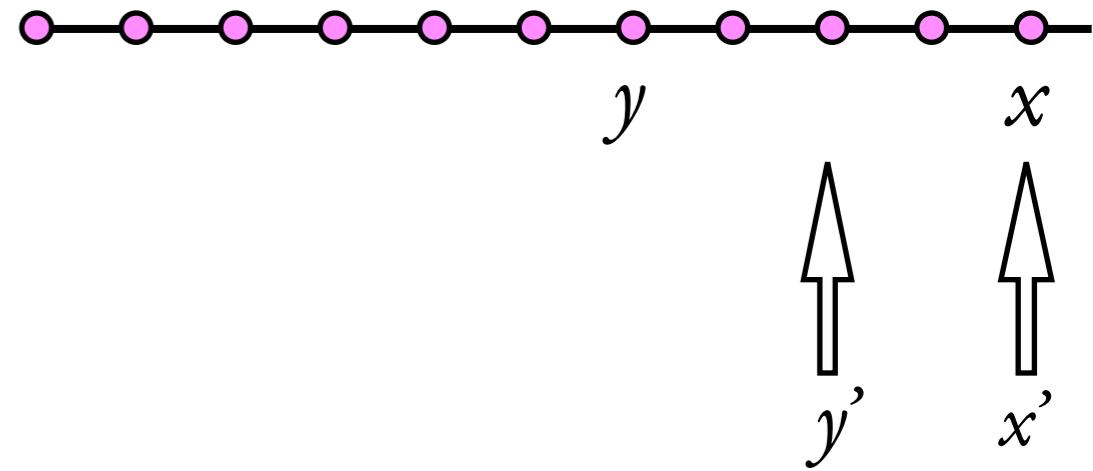
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$
-----	---	---	---	---	---	---	---	---------------	---------------

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

no states



$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

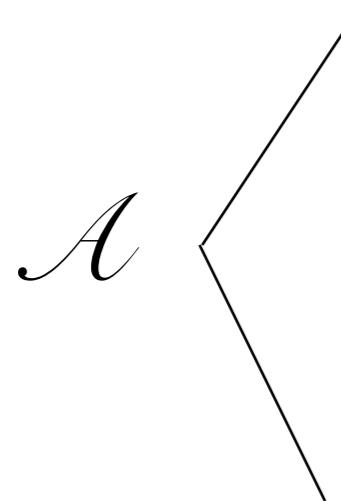
a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

x	5	5	5	5	5	5	5	5	5
-----	---	---	---	---	---	---	---	---	---

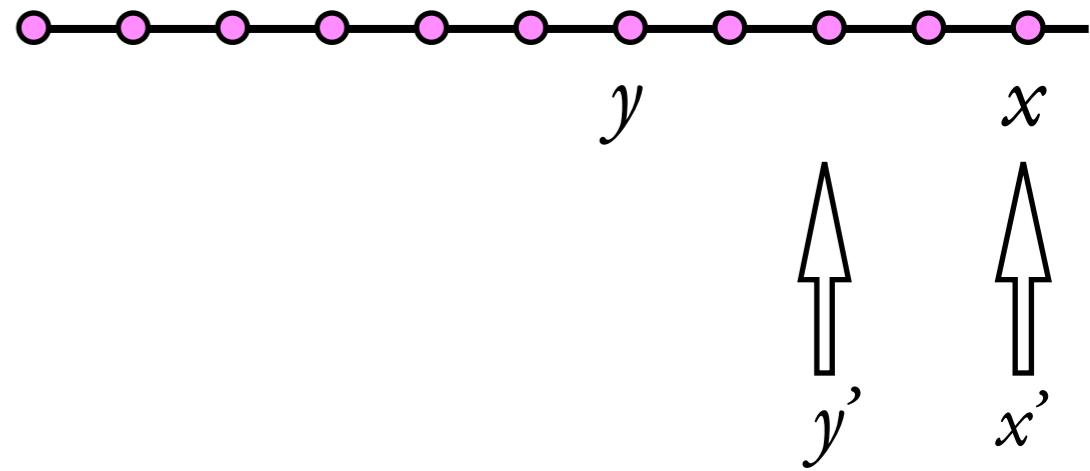
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{2}{3}$
-----	---	---	---	---	---	---	---	---------------	---------------	---------------

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

no states

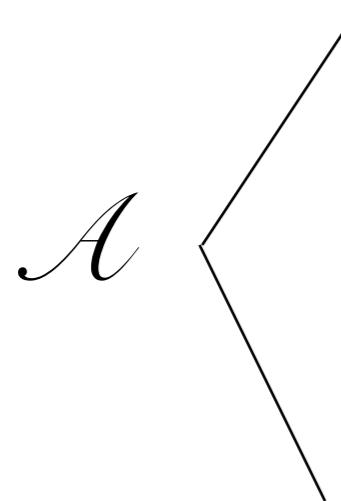


a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

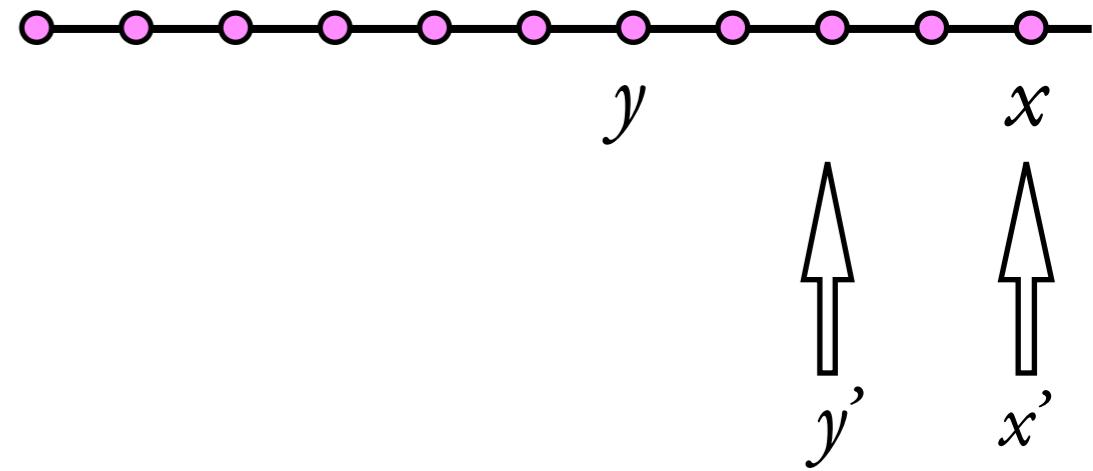
x	5	5	5	5	5	5	5	5	5	5	
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{4}{5}$

Example D -automaton \mathcal{A}



x, y – variables of \mathcal{A}

no states



$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

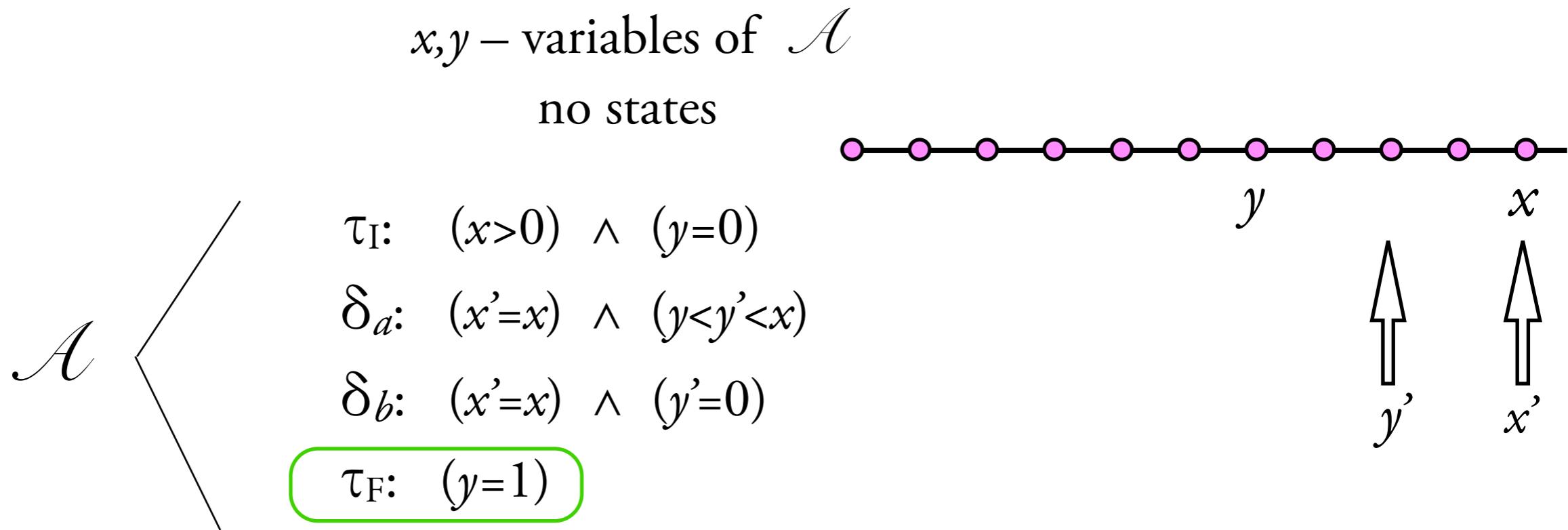
$$\tau_F: (y = 1)$$

a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

x	5	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{2}{3}$

Example D -automaton \mathcal{A}



a finite run over \mathbb{Q} :

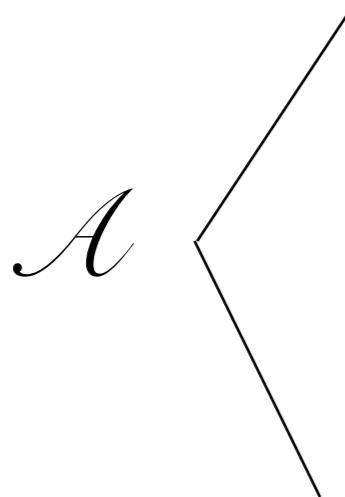
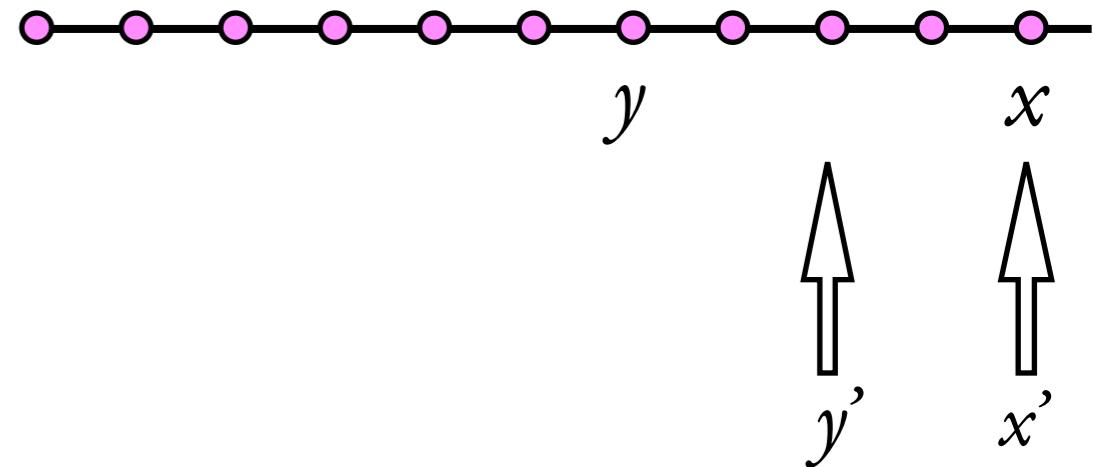
$a a a b a b a a a a a$

x	5	5	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{4}{5}$

Example D -automaton \mathcal{A}

x, y – variables of \mathcal{A}

no states



$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

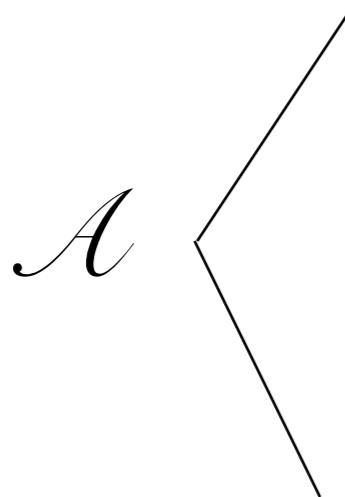
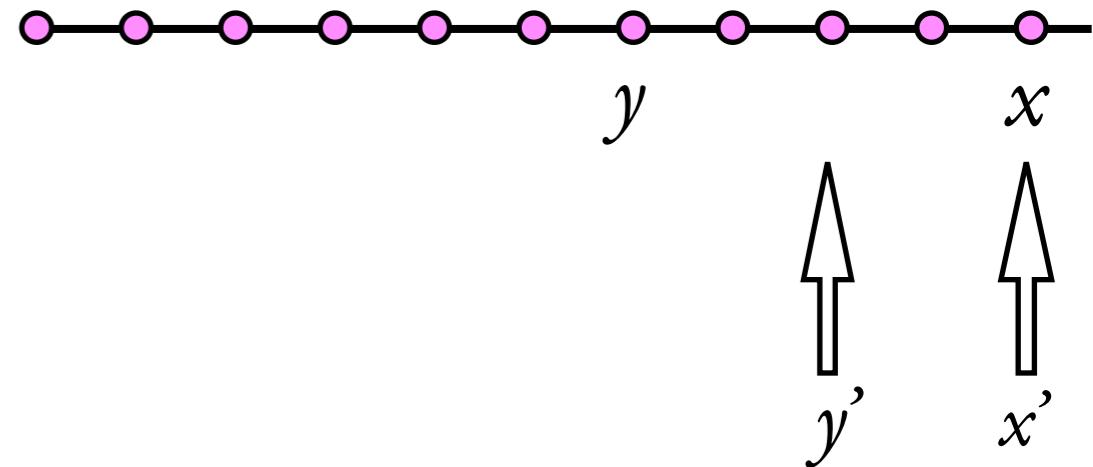
x	5	5	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{4}{5}$

accepted language: $(a+b)^*a$

Example D -automaton \mathcal{A}

x, y – variables of \mathcal{A}

no states



$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

a finite run over \mathbb{Q} :

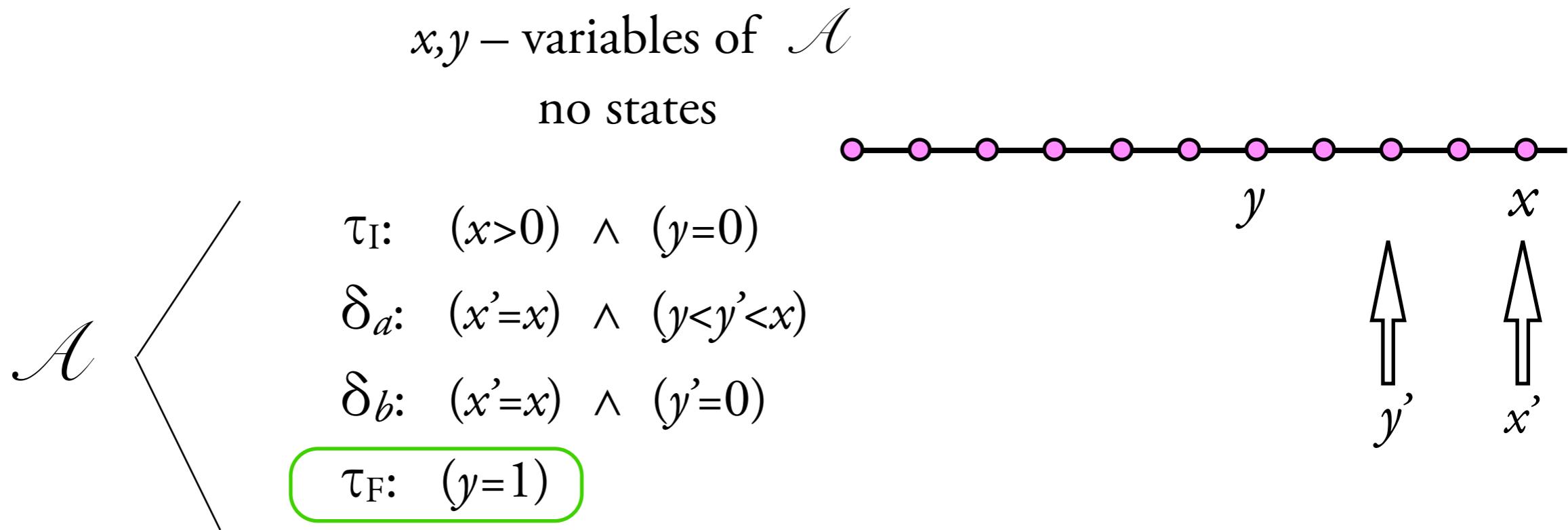
$a a a b a b a a a a a$

x	5	5	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{4}{5}$

a finite run over \mathbb{N} :

accepted language: $(a+b)^*a$

Example D -automaton \mathcal{A}



a finite run over \mathbb{Q} :

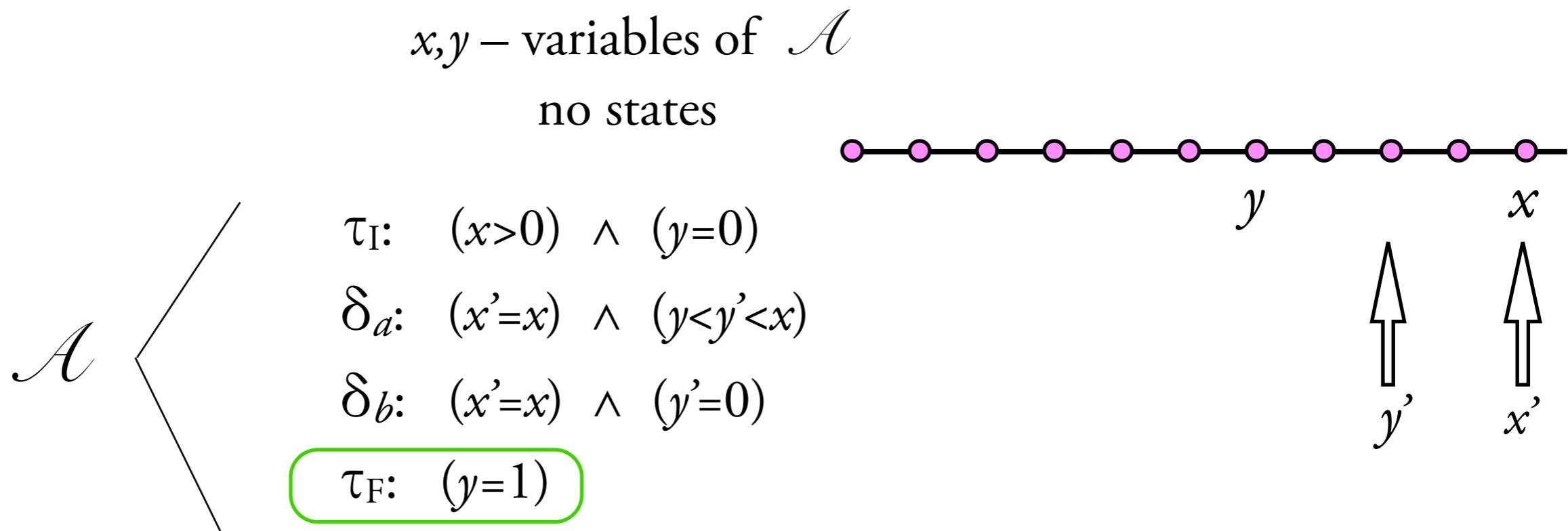
a	a	a	b	a	b	a	a	a	a
x	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$

accepted language: $(a+b)^*a$

a finite run over \mathbb{N} :

a a a b a b a a a a

Example D -automaton \mathcal{A}



a finite run over \mathbb{Q} :

a	a	a	b	a	b	a	a	a	a
x	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$

accepted language: $(a+b)^*a$

a finite run over \mathbb{N} :

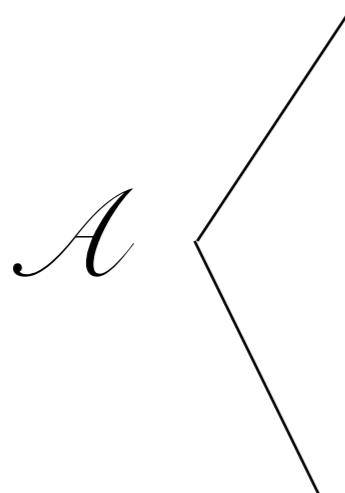
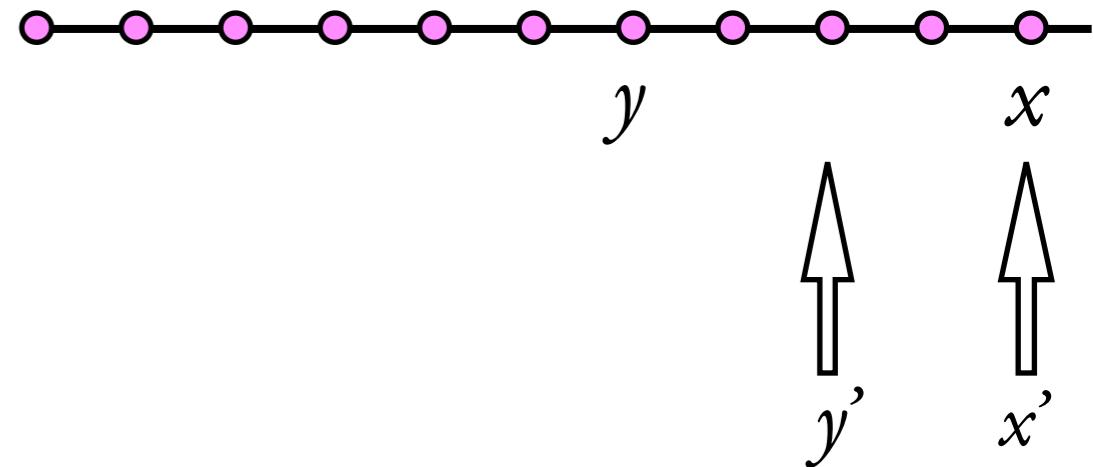
a a a b a b a a a a

x
 y

Example D -automaton \mathcal{A}

x, y – variables of \mathcal{A}

no states



$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

a finite run over \mathbb{Q} :

$a a a b a b a a a a a$

x	5	5	5	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{4}{5}$	1

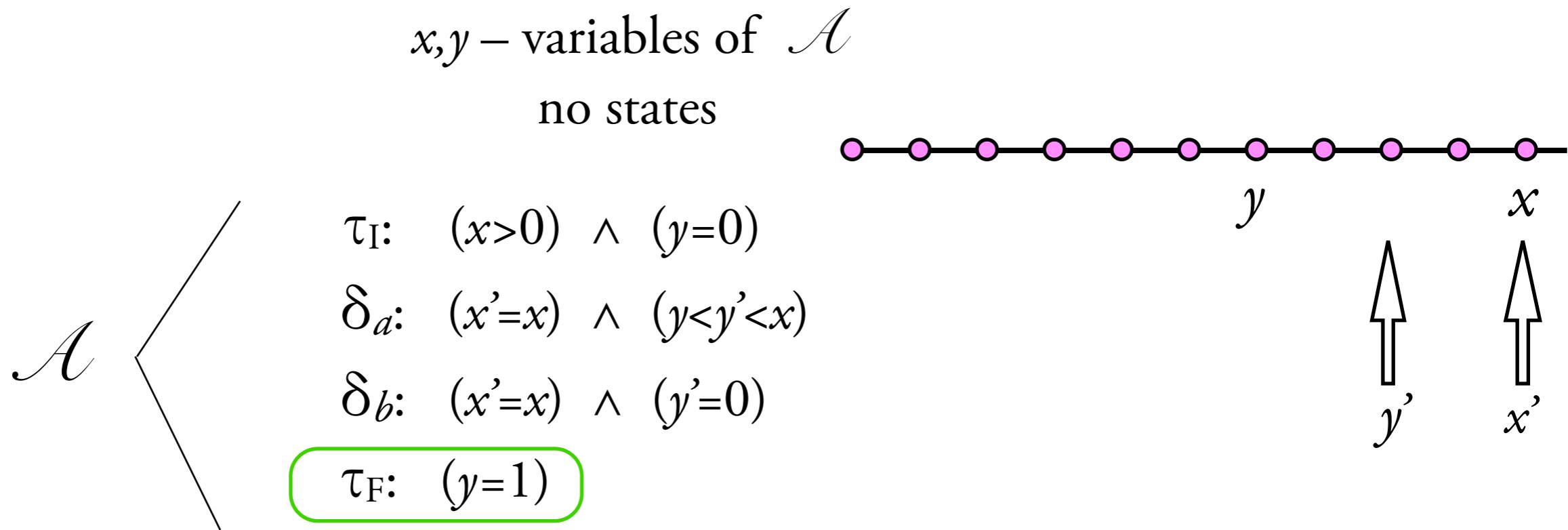
a finite run over \mathbb{N} :

$a a a b a b a a a a a$

x	5	5	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	1	2	3	4

accepted language: $(a+b)^*a$

Example D -automaton \mathcal{A}



a finite run over \mathbb{Q} :

a	a	a	b	a	b	a	a	a	a
x	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$

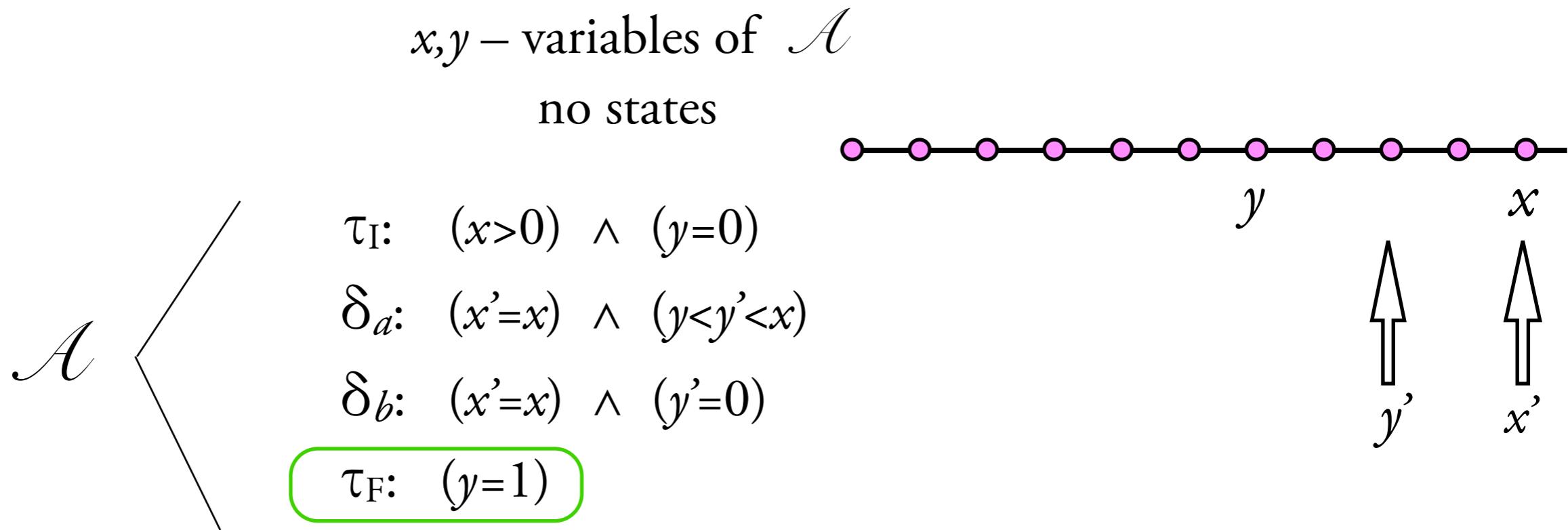
accepted language: $(a+b)^*a$

a finite run over \mathbb{N} :

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

accepted language: $(a^*b)^*a$

Example D -automaton \mathcal{A}



a finite run over \mathbb{Q} :

a	a	a	b	a	b	a	a	a	a
x	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$

accepted language: $(a+b)^*a$

a finite run over \mathbb{N} :

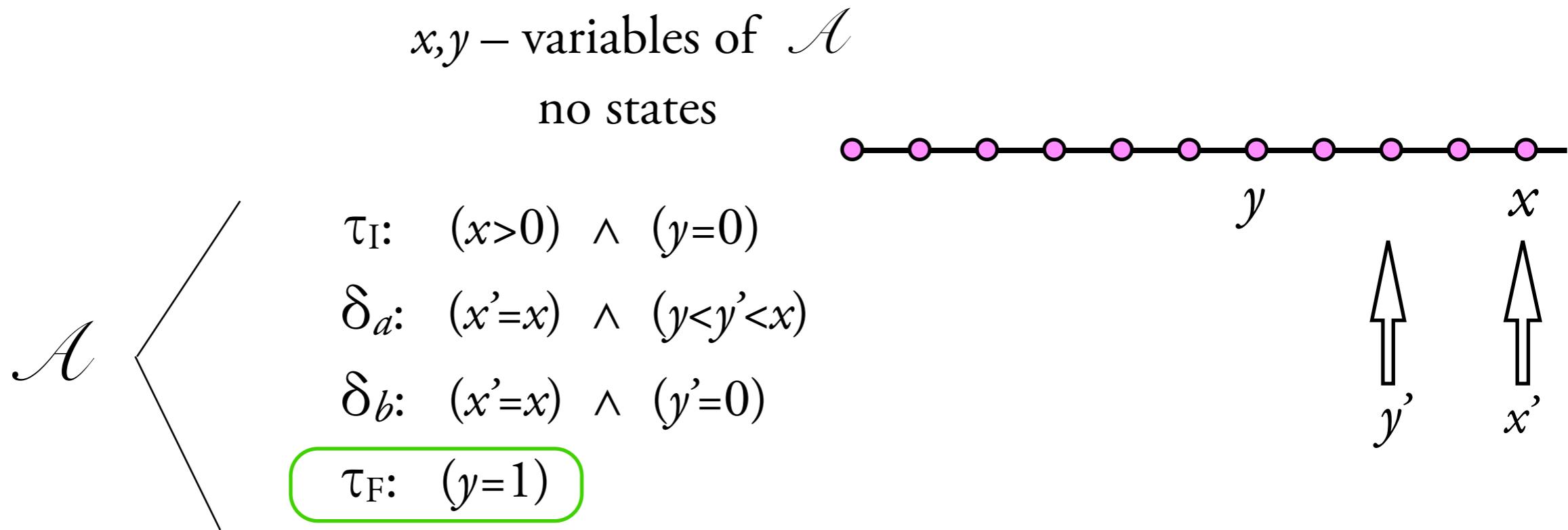
a	a	a	b	a	b	a	a	a
x	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	1

accepted language: $(a^*b)^*a$

an infinite run over \mathbb{N} :

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

Example D -automaton \mathcal{A}



a finite run over \mathbb{Q} :

a	a	a	b	a	b	a	a	a	a
x	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$

accepted language: $(a+b)^*a$

a finite run over \mathbb{N} :

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

accepted language: $(a^*b)^*a$

an infinite run over \mathbb{N} :

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

accepted language: $(a^B b)^\omega$

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

EMPTINESS: is there a *finite* database M , a word w ,
and an accepting run over w consistent with M ?

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

EMPTINESS: is there a *finite* database M , a word w , and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

EMPTINESS: is there a *finite* database M , a word w , and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

$a a a b a b a$

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

EMPTINESS: is there a *finite* database M , a word w , and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

$a a a b a b a$

$\} \exists$

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

EMPTINESS: is there a *finite* database M , a word w , and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

$a a a b a b a$

$R = \{(0), (1), (2), (3), (4)\}$

$\} \exists$

$S = \{\}$

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

EMPTINESS: is there a *finite* database M , a word w , and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

$a a a b a b a$

x	0	2	3	4	0	3	0	1	}
	$R = \{(0), (1), (2), (3), (4)\}$	$S = \{\}$							

\exists

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

EMPTINESS: is there a *finite* database M , a word w , and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

$a a a b a b a$
$x \quad 0 \ 2 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1$
$R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

$\} \exists$

accepted language: $(a^*b)^*a$

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

EMPTINESS: is there a *finite* database M , a word w , and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

$a a a b a b a$
 $x \quad 0 \ 2 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1 \quad \left. \right\} \exists$
 $R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^*b)^*a$

an infinite run over \mathbb{N} :

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

EMPTINESS: is there a *finite* database M , a word w , and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

$aaababaa$
 $x \quad 0 \ 2 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1 \quad \left. \right\} \exists$
 $R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^*b)^*a$

an infinite run over \mathbb{N} :

$aaababaa...baaa...$

Adding database constraints

x, y, z, \dots – a finite set of variables
 R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

EMPTINESS: is there a *finite* database M , a word w , and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

$a a a b a b a$
 $x \quad 0 \ 2 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1 \quad \left. \right\} \exists$
 $R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^*b)^*a$

an infinite run over \mathbb{N} :

$a a a b a b a a b a a \dots$
 $R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

EMPTINESS: is there a *finite* database M , a word w , and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

$aaababaa$
 $x \quad 0 \ 2 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1 \quad \} \exists$
 $R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^*b)^*a$

an infinite run over \mathbb{N} :

$aaababaa\ldots$
 $x \quad 0 \ 1 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1 \ 4 \ 0 \ 1 \ 3 \ldots$
 $R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x' > x) \wedge R(x) \wedge \neg S(x, y')$

$\delta_b: (x' = 0)$

$\tau_F: (x=1)$

EMPTINESS: is there a *finite* database M , a word w , and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

$aaababaa$
 $x \quad 0 \ 2 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1 \quad \} \exists$
 $R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^*b)^*a$

an infinite run over \mathbb{N} :

$aaababaa\ldots$
 $x \quad 0 \ 1 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1 \ 4 \ 0 \ 1 \ 3 \ldots$
 $R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^B b)^\omega$

Adding database constraints

x, y, z, \dots – a finite set of variables
 R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

$\delta_a: (x > x) \wedge R(x) \wedge \neg S(x, y)$
 $\delta_b: (x = 0)$

Decide emptiness

$\tau_F: (x = 1)$

of D -automata

EMPTINESS: is there a finite database M , a word w ,
over ω -words, with databases
and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

$a a a b a b a$
 $x \quad 0 \ 2 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1 \quad \} \exists$
 $R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^*b)^*a$

an infinite run over \mathbb{N} :

$a a a b a b a a b a a \dots$
 $x \quad 0 \ 1 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1 \ 4 \ 0 \ 1 \ 3 \dots$
 $R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^B b)^\omega$

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

I. infinite, ordered data

$\delta_a: (x' > x) \wedge R(x) \rightarrow S(x, y')$

$\delta_b: (x' < x) \wedge R(x) \rightarrow S(y, x')$

Decide emptiness

$\tau_F: (x = 1)$

EMPTINESS: is there a finite database M , a word w ,
over ω -words, with databases
and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

$aaababaa$
$x \quad 0 \ 2 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1$
$R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^*b)^*a$

an infinite run over \mathbb{N} :

$aaabababaa...$
$x \quad 0 \ 1 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1 \ 4 \ 0 \ 1 \ 3$
$R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^B b)^\omega$

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

I. infinite, ordered data

Decide emptiness

$\tau_F: (x=1)$

of D -automata

EMPTINESS: is there a finite database M , a word w ,
over ω -words, with databases
and an accepting run over w consistent with M ?

a finite run over \mathbb{N} :

$a a a b a b a$
$x \quad 0 \ 2 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1$
$R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^*b)^*a$

an infinite run over \mathbb{N} :

$a a a b a b a a b a a \dots$
$x \quad 0 \ 1 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1 \ 4 \ 0 \ 1 \ 3$
$R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^B b)^\omega$

Adding database constraints

x, y, z, \dots – a finite set of variables

R, S, \dots – a finite set of relational symbols

$\tau_I: (x=0)$

I. infinite, ordered data

Decide emptiness
of D -automata

EMPTINESS: is there a finite database M , a word w ,
over ω -words, with databases
and an accepting run over w consistent with M ?

3. infinite runs

a finite run over \mathbb{N} :

$a a a b a b a$
 $x \quad 0 \ 2 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1 \ \} \exists$
 $R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^*b)^*a$

2. unbounded database
an infinite run over \mathbb{N} :

$a a a b a b a b a a \dots$
 $x \quad 0 \ 1 \ 3 \ 4 \ 0 \ 3 \ 0 \ 1 \ 4 \ 0 \ 1 \ 3 \ \dots$
 $R=\{(0),(1),(2),(3),(4)\} \quad S=\{\}$

accepted language: $(a^B b)^\omega$

Known results deciding emptiness (for ω -runs)

	no database	with database
$\mathbb{N}, =$	Kaminski, Francez (94)	Deutsch, Sui, Vianu, Zhou (06) PSPACE
$\mathbb{Q}, <$	Čerans (94) PSPACE	Deutsch, Hull, Patrizi, Vianu (09) PSPACE
$\mathbb{N}, <$	Čerans (94) NONPRIMITIVE	?

Known results deciding emptiness (for ω -runs)

	no database	with database
$\mathbb{N}, =$	Kaminski, Francez (94)	Deutsch, Sui, Vianu, Zhou (06) PSPACE
$\mathbb{Q}, <$	Čerans (94) PSPACE	Deutsch, Hull, Patrizi, Vianu (09) PSPACE
$\mathbb{N}, <$	Čerans (94) NONPRIMITIVE	?
$\mathcal{D}, <$	PSPACE	PSPACE

Known results deciding emptiness (for ω -runs)

	no database	with database
$\mathbb{N}, =$	Kaminski, Francez (94)	Deutsch, Sui, Vianu, Zhou (06) PSPACE
$\mathbb{Q}, <$	Čerans (94) PSPACE	Deutsch, Hull, Patrizi, Vianu (09) PSPACE
$\mathbb{N}, <$	Čerans (94) NONPRIMITIVE	PSPACE
$\mathcal{D}, <$	PSPACE	PSPACE

Known results deciding emptiness (for ω -runs)

	no database	with database
$\mathbb{N}, =$	Kaminski, Francez (94)	Deutsch, Sui, Vianu, Zhou (06) PSPACE
$\mathbb{Q}, <$	Čerans (94) PSPACE	Deutsch, Hull, Patrizi, Vianu (09) PSPACE
$\mathbb{N}, <$	PSPACE	PSPACE
$\mathcal{D}, <$	PSPACE	PSPACE

Outline

1. Define automata model
2. Analyze automata over dense orders
3. Analyze automata over discrete orders
4. Add database and infinite runs

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, 0 < x < 1 = y, \dots$ The region on construction $0 < 1 < y < x, 0 < 1 = y < x, 0 = y < x, 0 < y < 1 = x, \dots$

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, \quad 0 < x < 1 = y, \quad 0 \leq 1 < y \leq x, \quad 0 < 1 \leq y < x, \quad 0 \leq y < x, \quad 0 = y < x, \quad \dots$



Dense case

- $D = \mathbb{Q}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, \quad 0 < x < 1 = y, \quad 0 \leq 1 < y < x, \quad 0 < 1 \leq y < x, \quad 0 = y < 1 < x, \quad 0 < y < 1 = x, \dots$

Ψ
 $\exists(x, y)$

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, \quad 0 < x < 1 = y, \quad 0 \leq 1 < y < x, \quad 0 < 1 \leq y < x, \quad 0 = y < 1 < x, \quad 0 < y < 1 = x, \dots$

$$\Psi(x', y') \quad \Psi(x, y)$$

\curvearrowleft_a

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, \quad 0 < x < 1 = y, \quad 0 \leq 1 < y \leq x, \quad 0 < 1 \leq y < x, \quad 0 \leq y < x, \quad 0 = y < 1 = x, \dots$

The region on construction

$$\exists(x', y'): \quad \exists(x, y): \quad \delta_a(x, y, x', y')$$

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

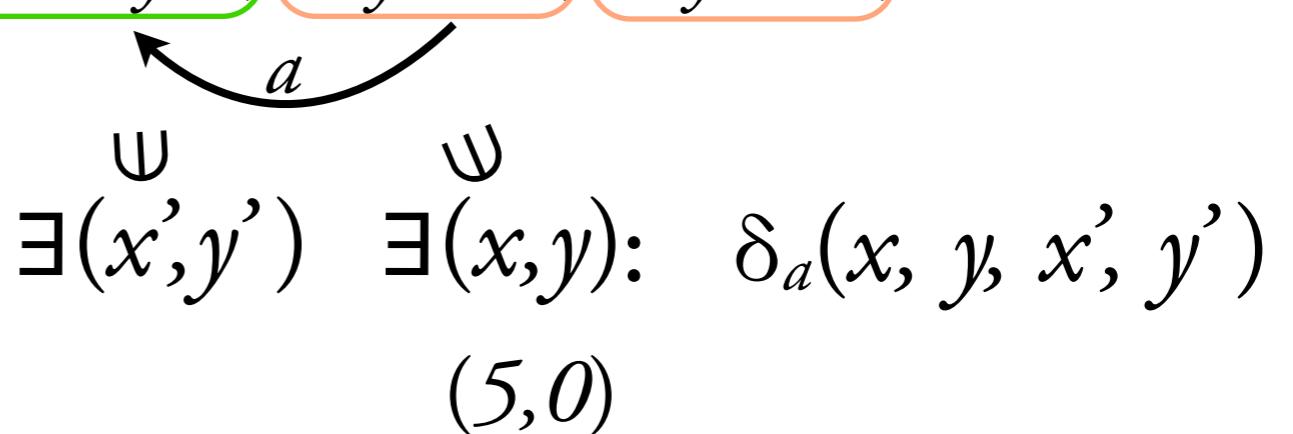
$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, \quad 0 < x < 1 = y, \quad 0 \leq 1 < y \leq x, \quad 0 < 1 \leq y < x, \quad 0 \leq y < x, \quad 0 = y < 1 = x, \dots$

The region on construction


$$\Psi(x', y') \quad \Psi(x, y): \quad \delta_a(x, y, x', y')$$
$$(5, 0)$$

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, \quad 0 < x < 1 = y, \quad 0 \leq 1 < y < x, \quad 0 < 1 = y < x, \quad 0 = y < 1 < x, \quad \dots$

The region on construction

$$\exists(x', y'): \quad \exists(x, y): \quad \delta_a(x, y, x', y')$$

$$(5, 1)$$

$$(5, 0)$$

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, \quad 0 < x < 1 = y, \quad 0 \leq 1 < y \leq x, \quad 0 < 1 \leq y < x, \quad 0 \leq y < x, \quad 0 = y < x, \quad \dots$



Dense case

- $D = \mathbb{Q}$
- finite words
- no database

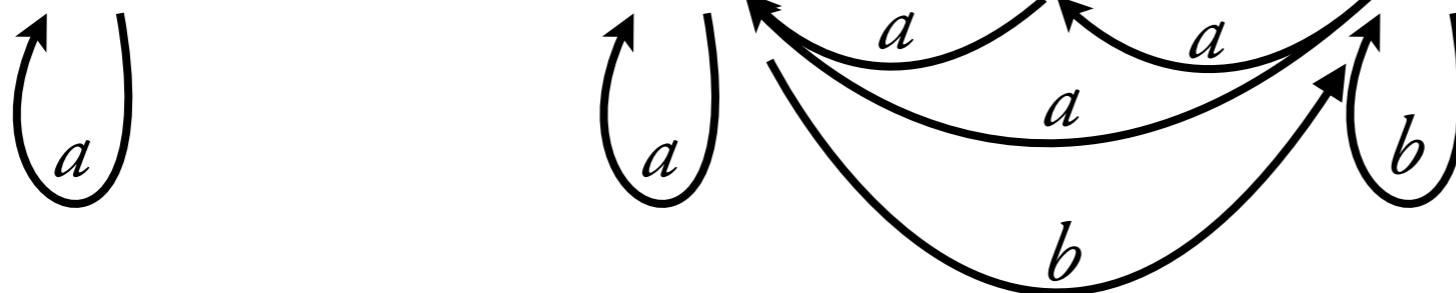
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, 0 < x < 1 = y, 0 \leq 1 < y < x, 0 < 1 \leq y < x, 0 \leq y < 1 < x, 0 = y < 1 = x, \dots$



Dense case

- $D = \mathbb{Q}$
- finite words
- no database

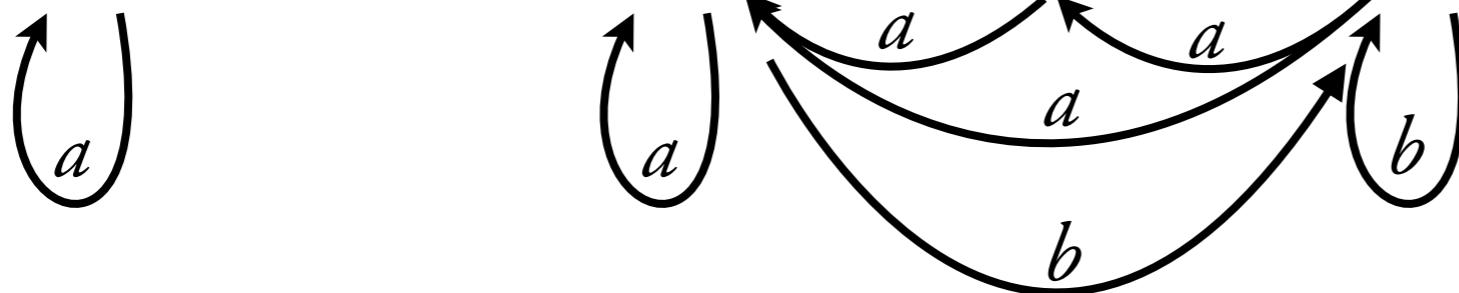
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, 0 < x < 1 = y, 0 \leq 1 < y < x, 0 < 1 = y < x, 0 = y < 1 < x, 0 = y < 1 = x, \dots$



bisimulation

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

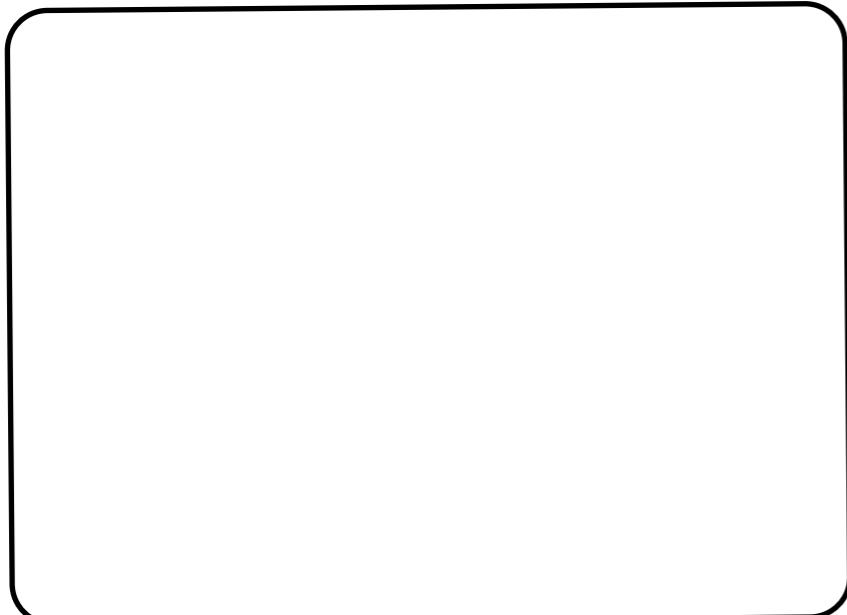
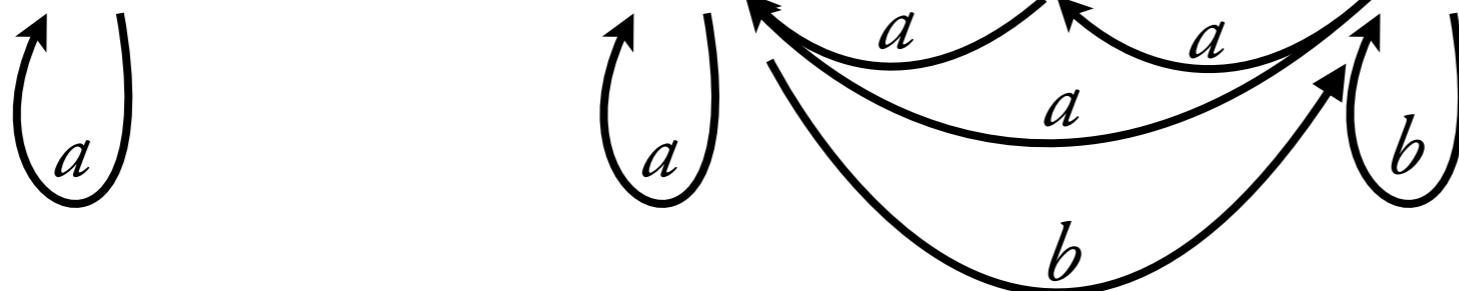
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, \quad 0 < x < 1 = y, \quad 0 \leq 1 < y < x, \quad 0 < 1 = y < x, \quad 0 = y < 1 < x, \quad 0 = y < 1 = x, \dots$



bisimulation

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

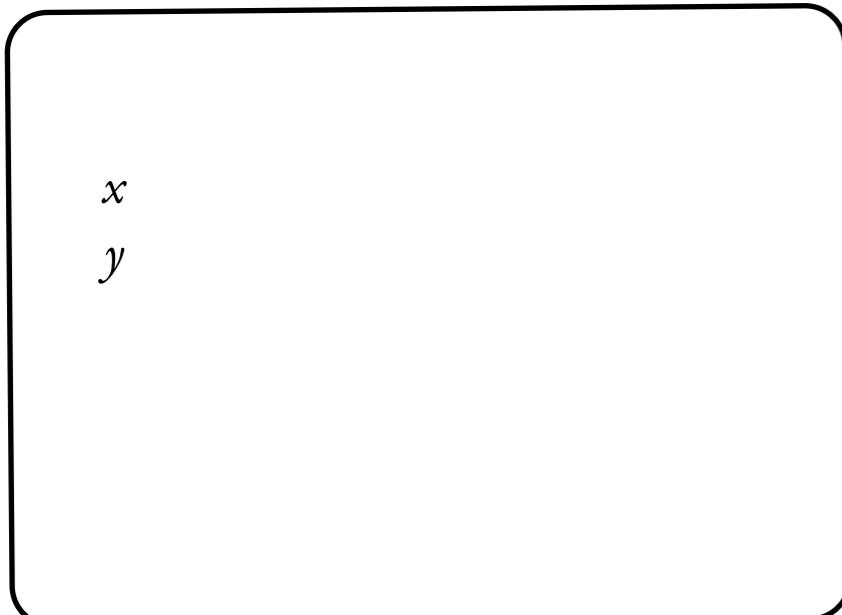
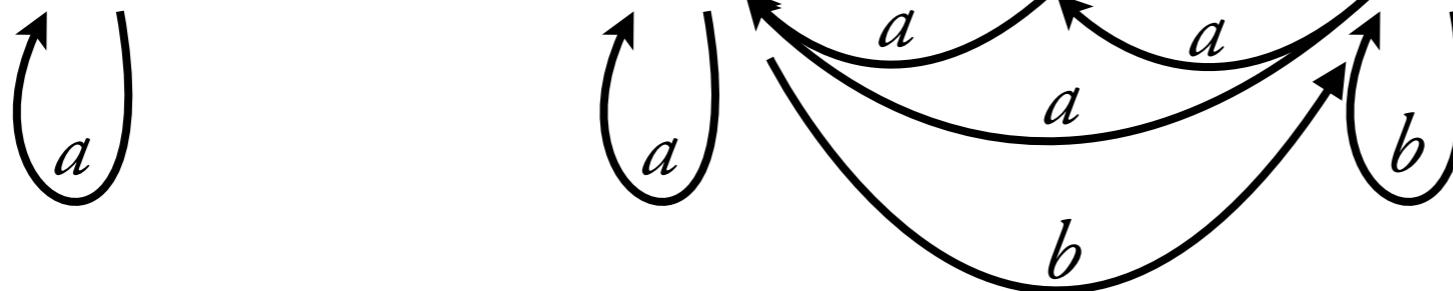
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, 0 < x < 1 = y, 0 \leq 1 < y < x, 0 < 1 = y < x, 0 = y < 1 < x, 0 = y < 1 = x, \dots$



bisimulation

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

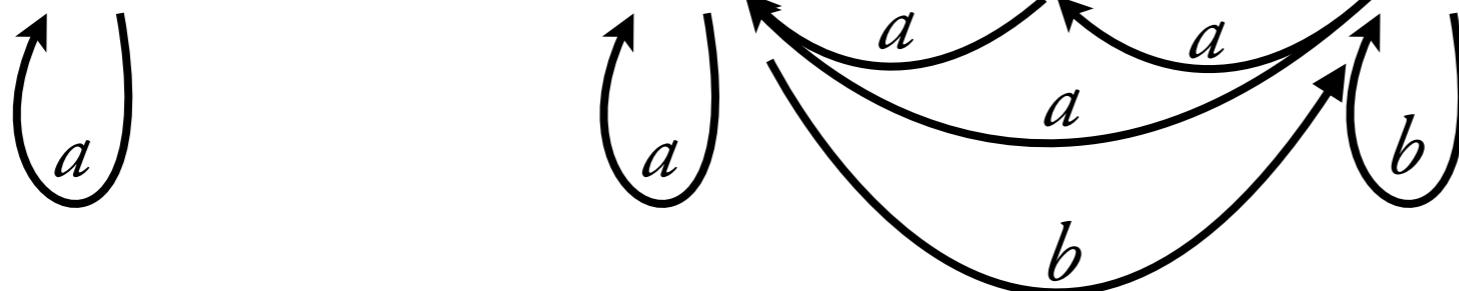
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, 0 < x < 1 = y, 0 \leq 1 < y < x, 0 < 1 = y < x, 0 = y < x, 0 - y < 1 = x, \dots$



		<i>a</i>	<i>a</i>	<i>a</i>	<i>b</i>
<i>x</i>	5	5	5	5	5
<i>y</i>	0	1	2	4	0

bisimulation

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

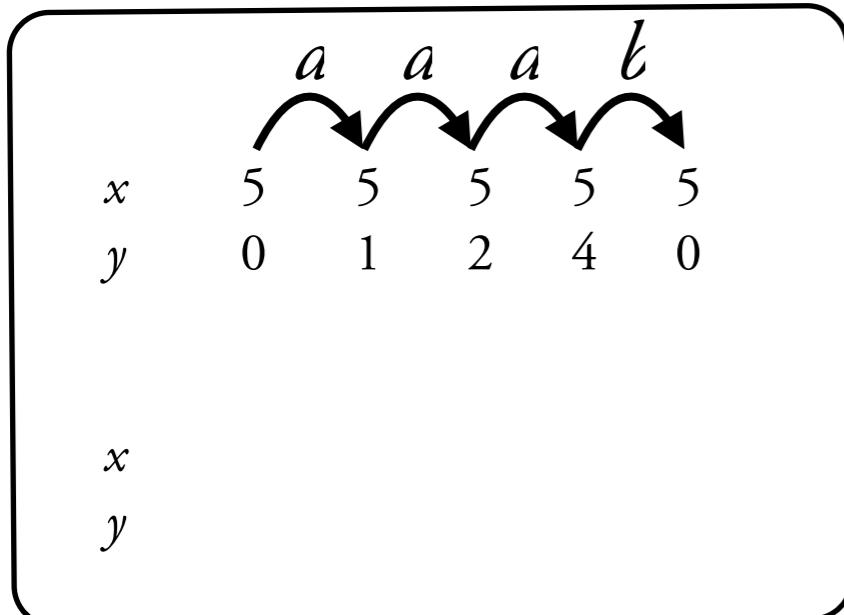
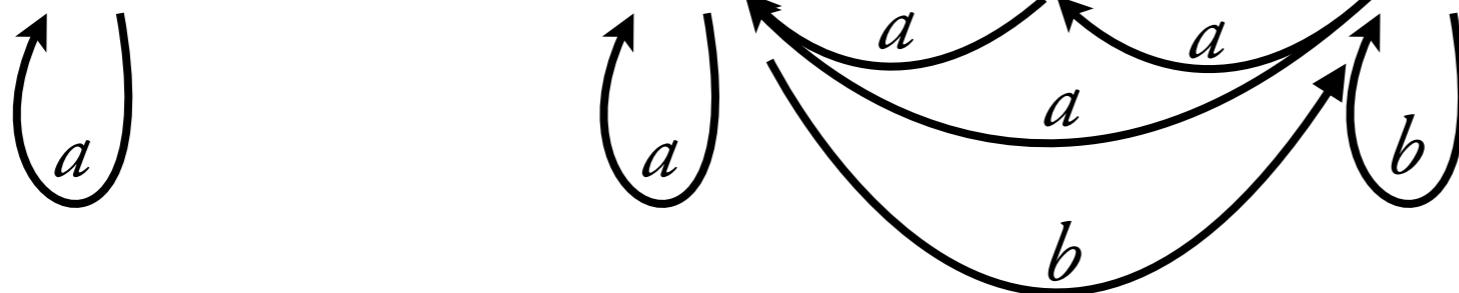
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, 0 < x < 1 = y, \dots$ The region on construction \dots



bisimulation

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

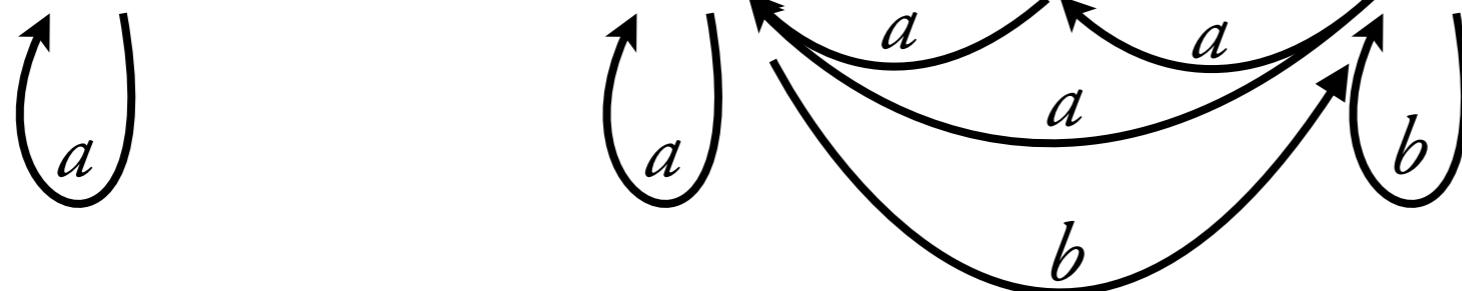
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, 0 < x < 1 = y, 0 \leq 1 < y < x, 0 < 1 = y < x, 0 = y < x, 0 - y < 1 = x, \dots$



	a	a	a	b
x	5	5	5	5
y	0	1	2	4
	ℓ			
x	2			
y	0			

bisimulation

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

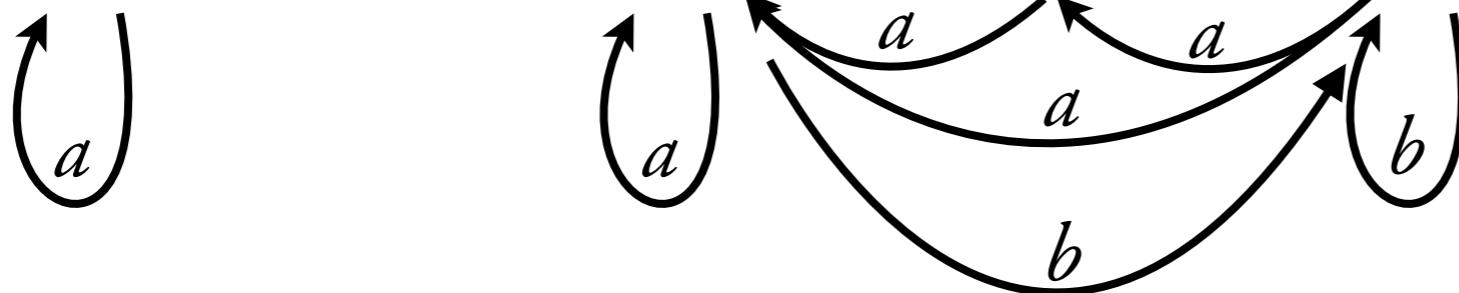
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, 0 < x < 1 = y, \dots$ The region on construction $\vdash_1 y < x, 0 < 1 = y < x, 0 < y < x, 0 = y < x, 0 < y < 1 = x, \dots$



		<i>a</i>	<i>a</i>	<i>a</i>	<i>b</i>
<i>x</i>	5	5	5	5	5
<i>y</i>	0	1	2	4	0
	l	l			
<i>x</i>	2	2			
<i>y</i>	0	1			

bisimulation

Dense case

- $D = \mathbb{Q}$
- finite words
- no database

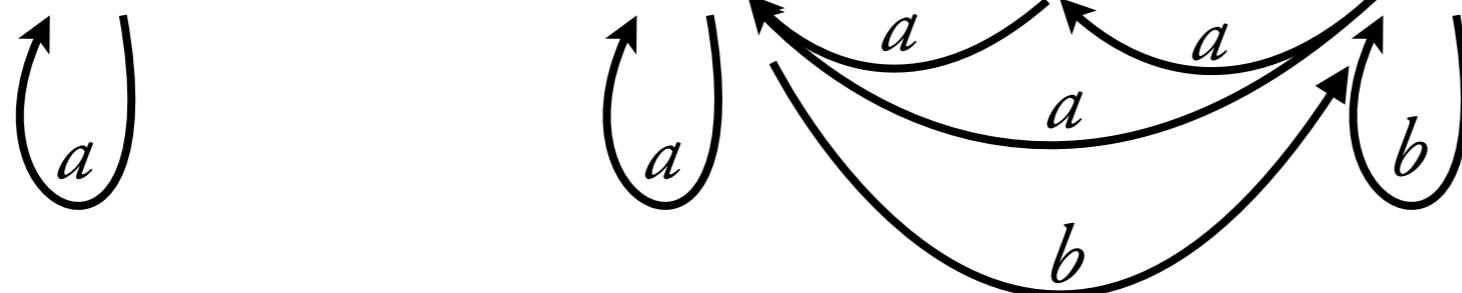
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, 0 < x < 1 = y, 0 \leq 1 < y < x, 0 < 1 = y < x, 0 = y < x, 0 = y < 1 = x, \dots$



	<i>a</i>	<i>a</i>	<i>a</i>	<i>b</i>
<i>x</i>	5	5	5	5
<i>y</i>	0	1	2	4
	l	l	l	l
<i>x</i>	2	2	2	2
<i>y</i>	0	1	1.2	1.3

bisimulation

Dense case

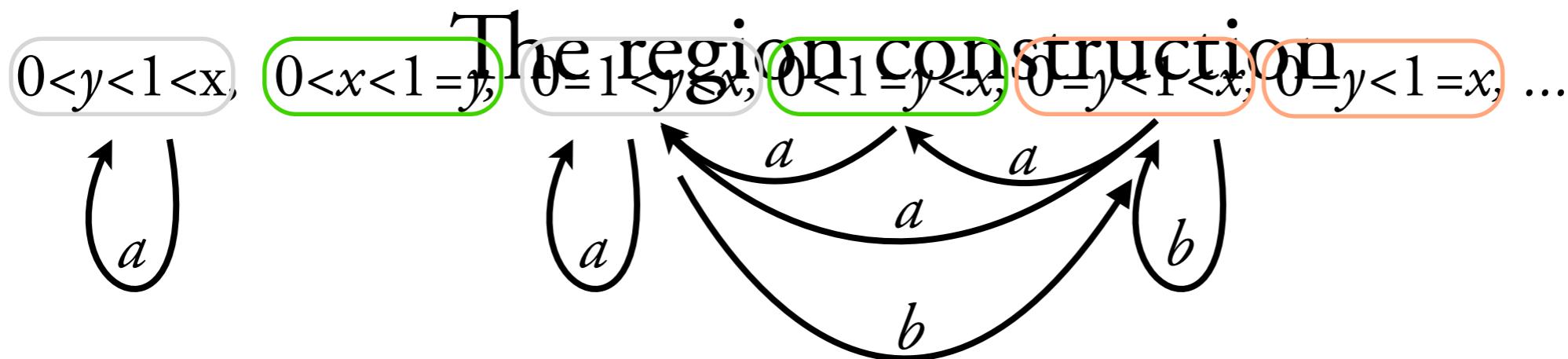
- $D = \mathbb{Q}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



		a	a	a	b
x	5	5	5	5	5
y	0	1	2	4	0
	l	l	l	l	l
x	2	2	2	2	2
y	0	1	1.2	1.3	0

bisimulation \rightarrow runs of the region
automaton correspond to runs of \mathcal{A}

Dense case

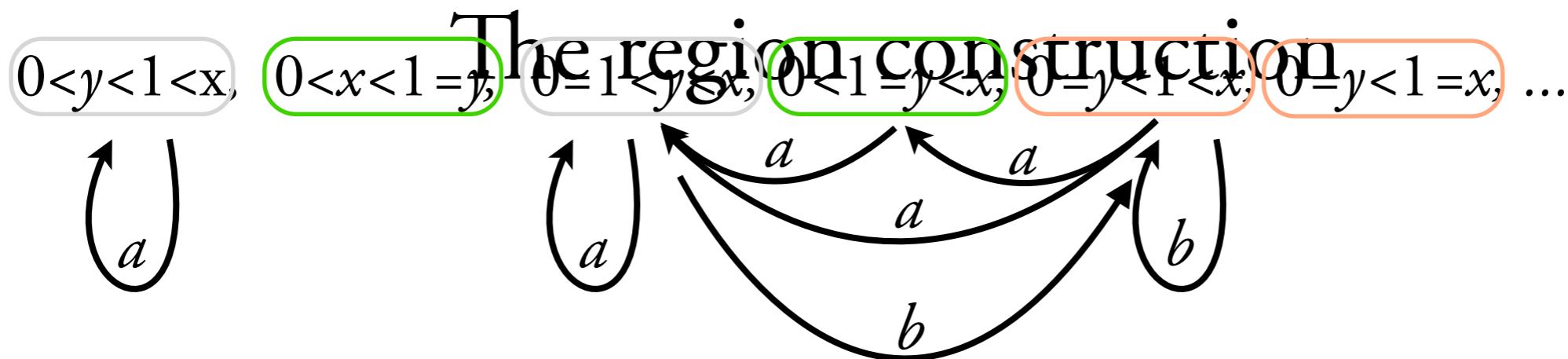
- $D = \mathbb{Q}$
- finite words
- no database

$$\tau_I: (x>0) \wedge (y=0)$$

$$\delta_a: (x'=x) \wedge (y < y' < x)$$

$$\delta_b: (x'=x) \wedge (y'=0)$$

$$\tau_F: (y=1)$$



	<i>a</i>	<i>a</i>	<i>a</i>	<i>b</i>
<i>x</i>	5	5	5	5
<i>y</i>	0	1	2	4
	l	l	l	l
<i>x</i>	2	2	2	2
<i>y</i>	0	1	1.2	1.3

bisimulation \rightarrow runs of the region
automaton correspond to runs of \mathcal{A}

accepted language: $(a+b)^*a$

- $D = \mathbb{Q}$
- finite words
- no database

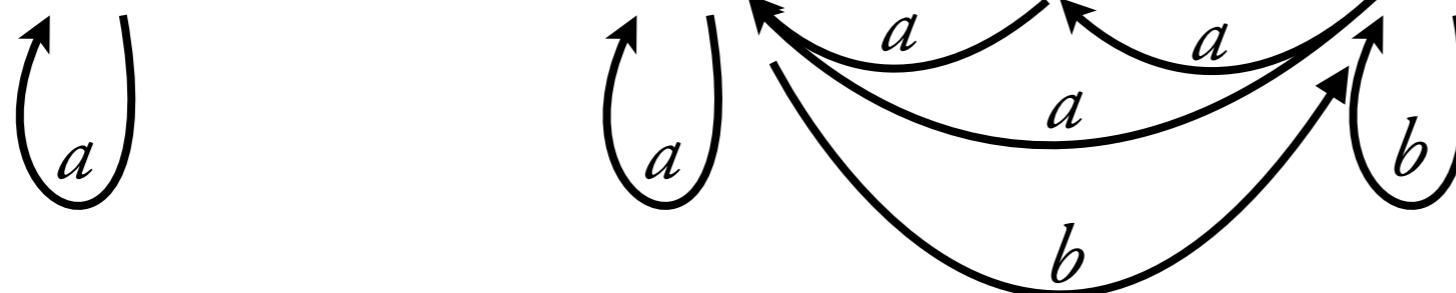
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, 0 < x < 1 = y, 0 \leq 1 < y < x, 0 < 1 = y < x, 0 = y < x, 0 - y < 1 = x, \dots$



		<i>a</i>	<i>a</i>	<i>a</i>	<i>b</i>
<i>x</i>	5	5	5	5	5
<i>y</i>	0	1	2	4	0
	l	l	l	l	l
<i>x</i>	2	2	2	2	2
<i>y</i>	0	1	1.2	1.3	0

bisimulation → runs of the region
automaton correspond to runs of \mathcal{A}

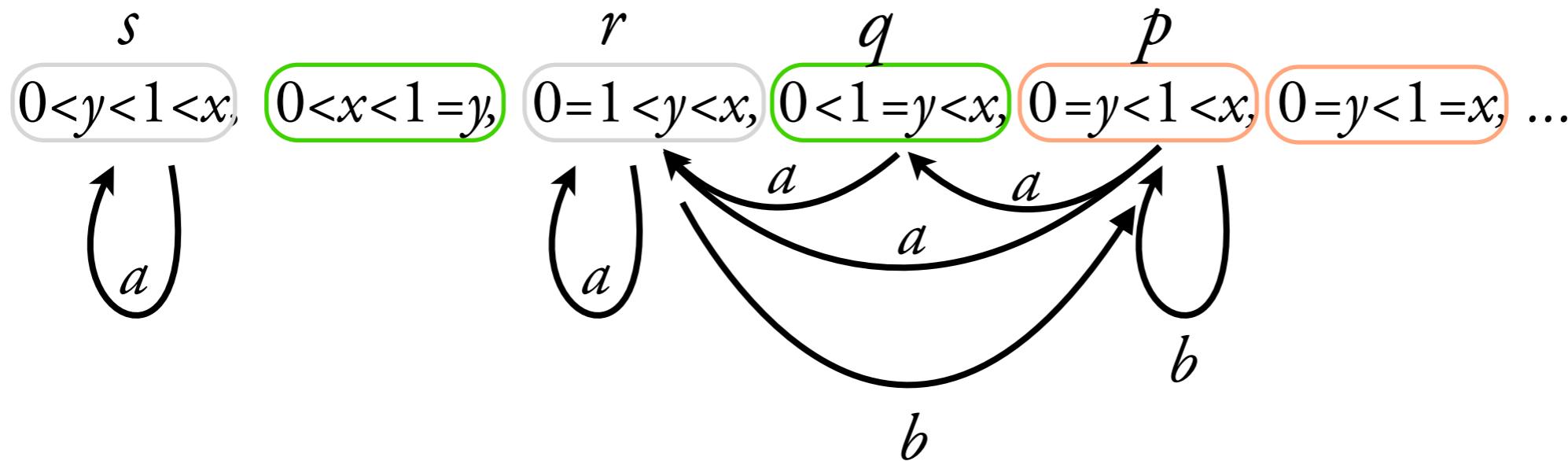
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x>0) \wedge (y=0)$$

$$\delta_a: (x'=x) \wedge (y < y' < x)$$

$$\delta_b: (x'=x) \wedge (y'=0)$$

$$\tau_F: (y=1)$$



bisimulation \rightarrow runs of the region automaton correspond to runs of \mathcal{A}

Discrete case

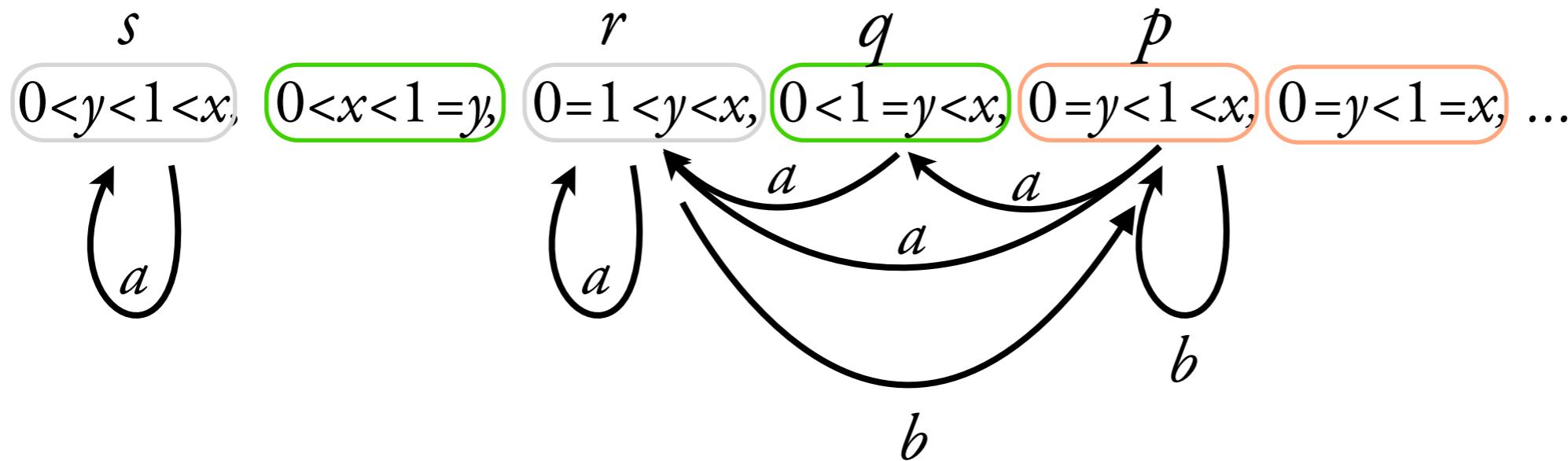
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



bisimulation \rightarrow runs of the region
automaton correspond to runs of \mathcal{A}

Discrete case

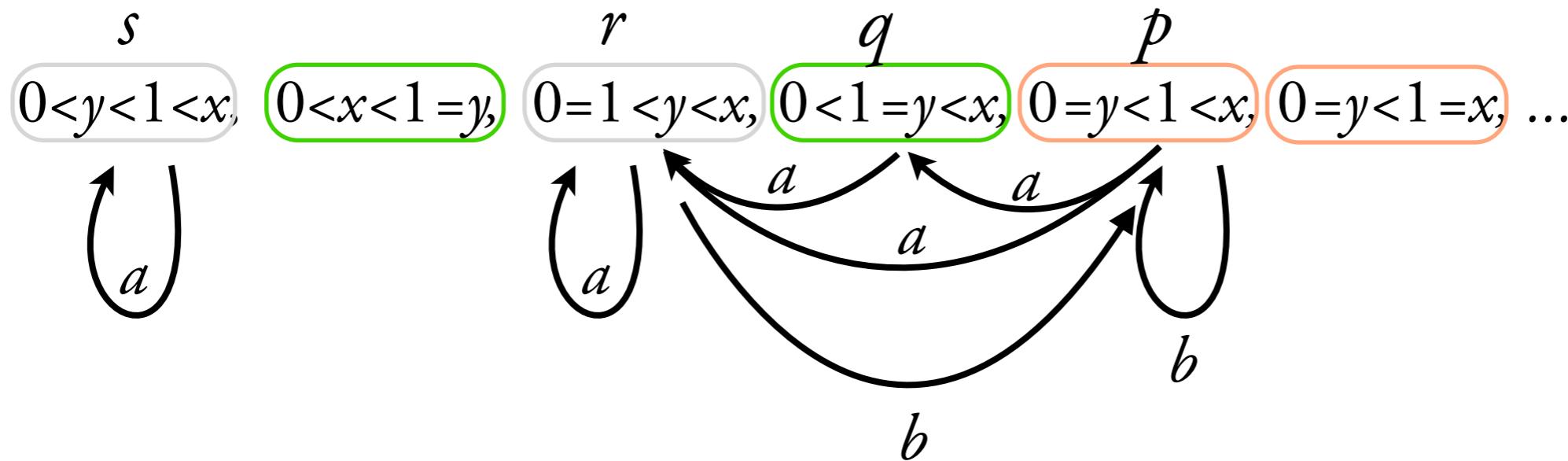
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



~~bisimulation~~ → runs of the region
automaton correspond to runs of \mathcal{A}

Discrete case

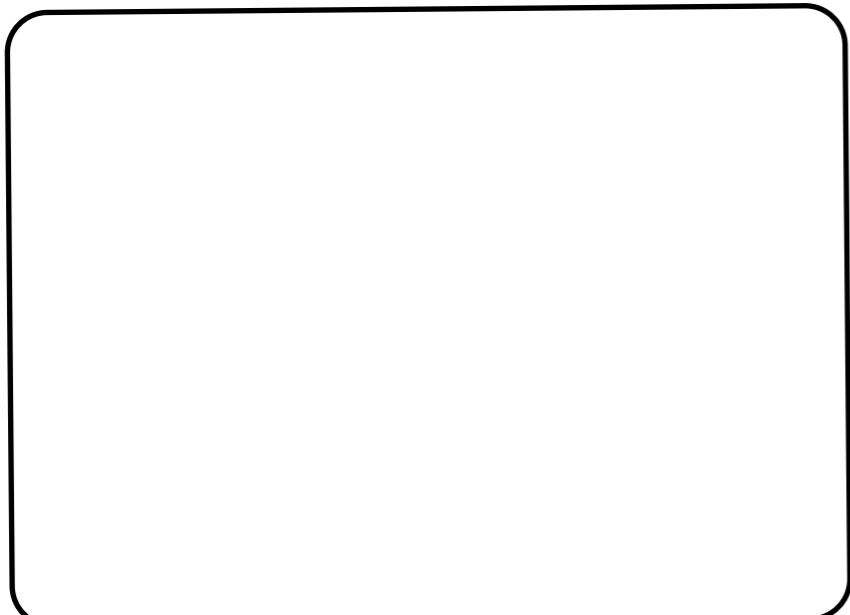
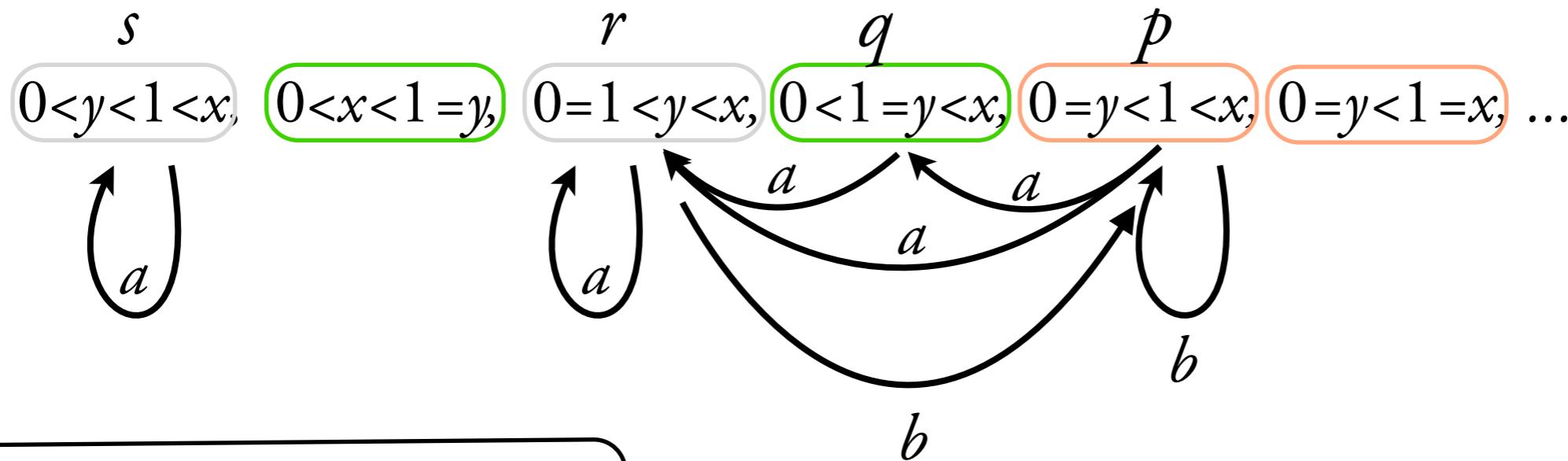
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



~~bisimulation~~ → runs of the region
automaton correspond to runs of \mathcal{A}

Discrete case

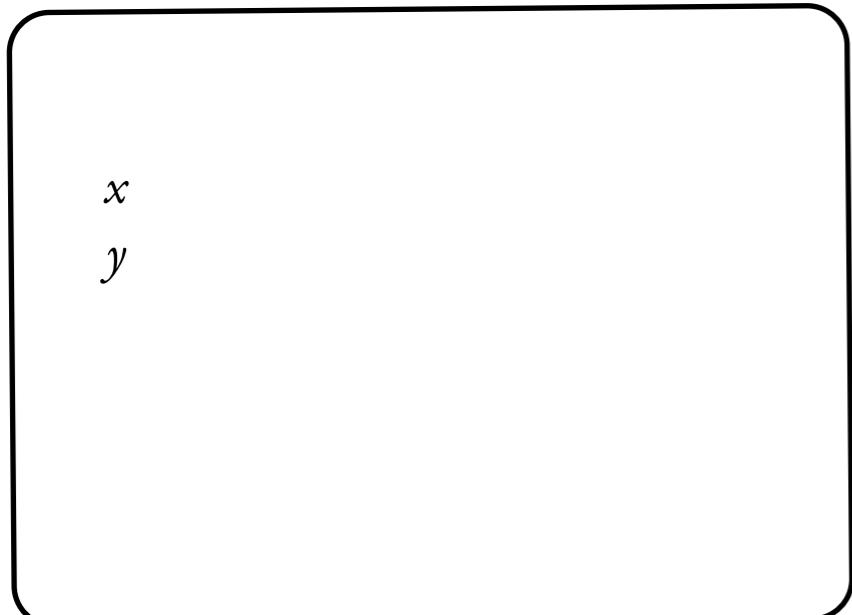
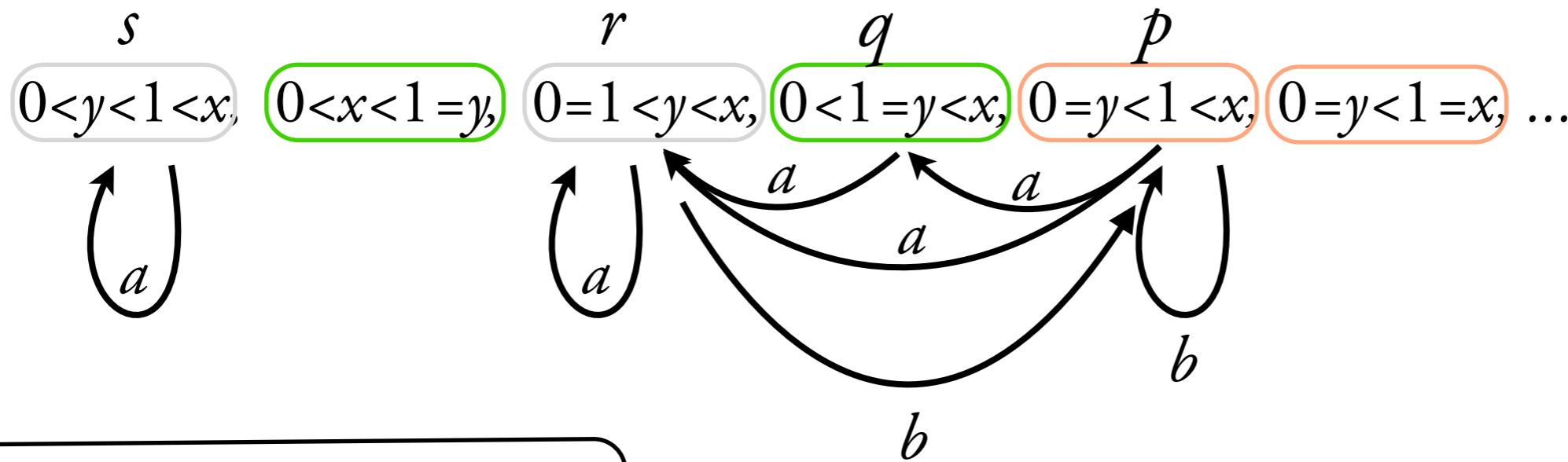
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



~~bisimulation~~ → runs of the region
automaton correspond to runs of \mathcal{A}

Discrete case

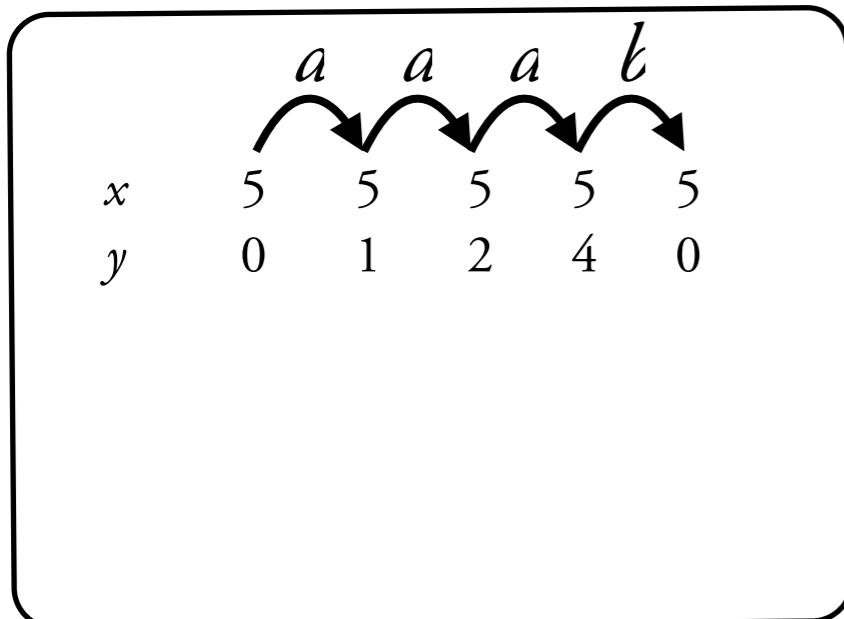
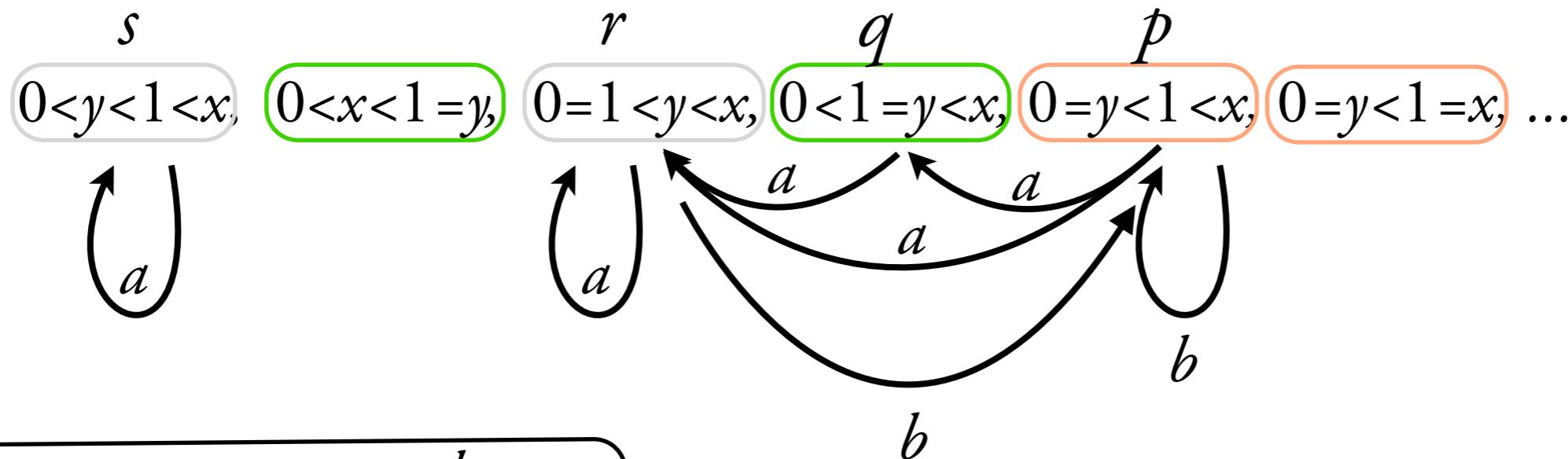
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



~~bisimulation~~ → runs of the region
automaton correspond to runs of \mathcal{A}

Discrete case

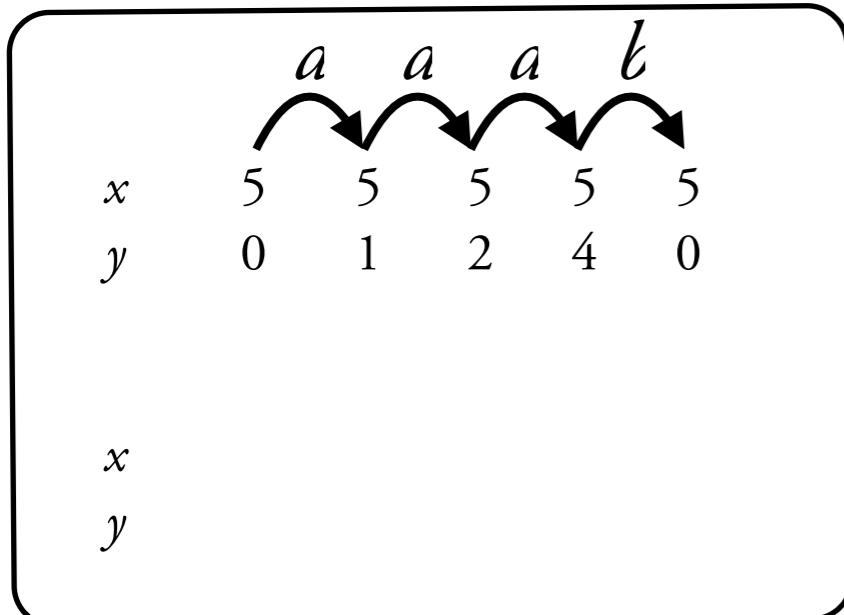
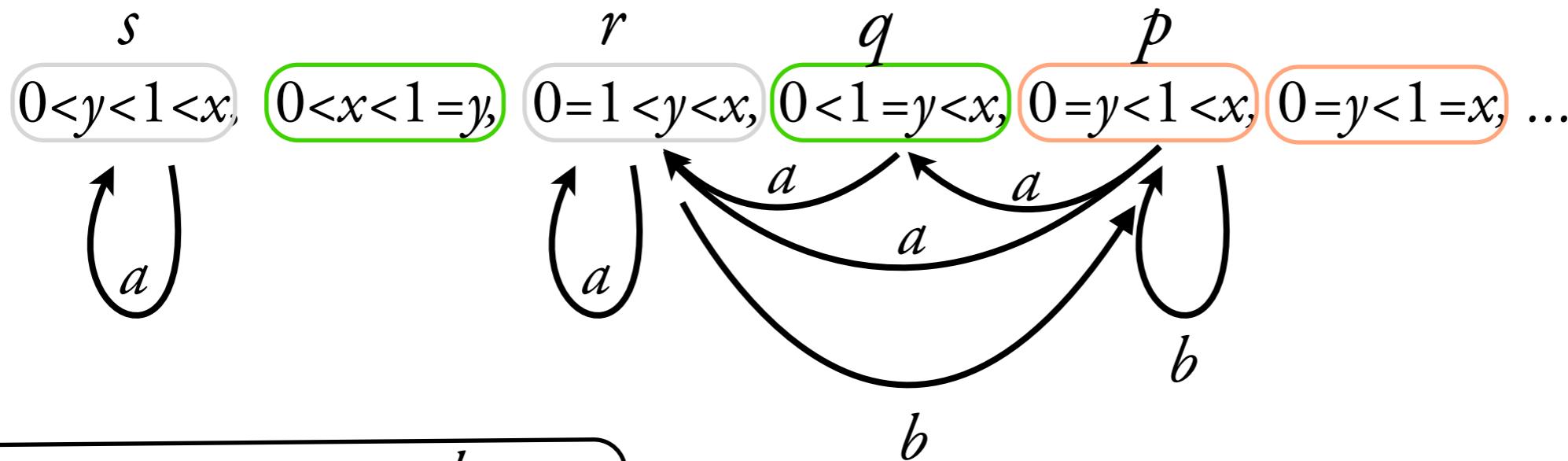
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



~~bisimulation~~ → runs of the region
automaton correspond to runs of \mathcal{A}

Discrete case

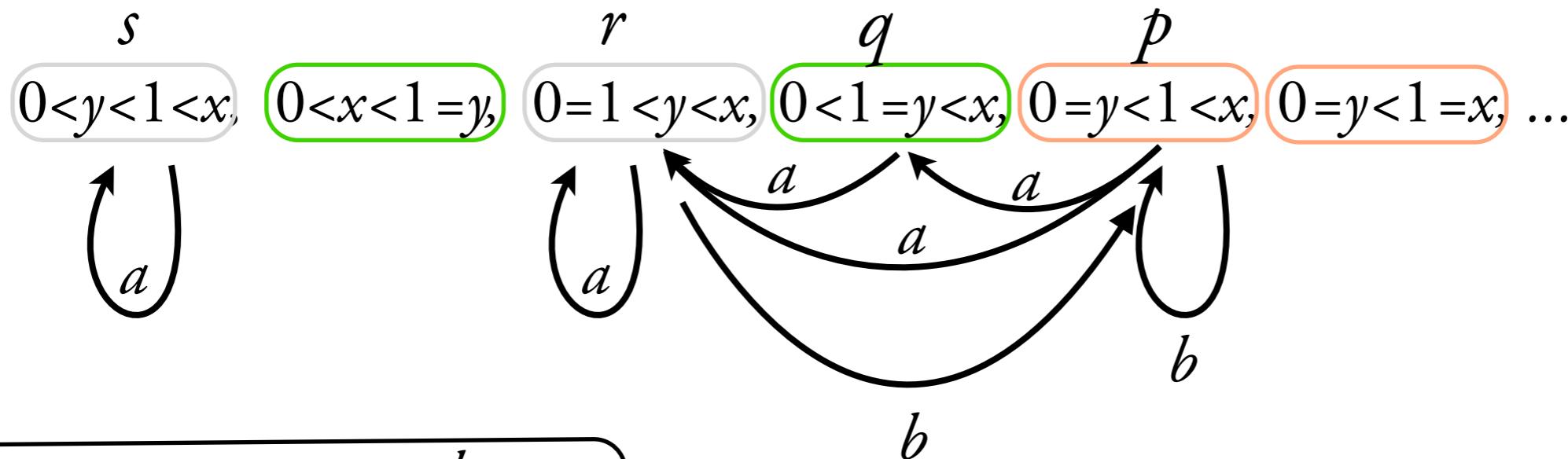
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



	a	a	a	ℓ
x	5	5	5	5
y	0	1	2	4
ℓ				0
x	2			
y	0			

~~bisimulation~~ → runs of the region
automaton correspond to runs of \mathcal{A}

Discrete case

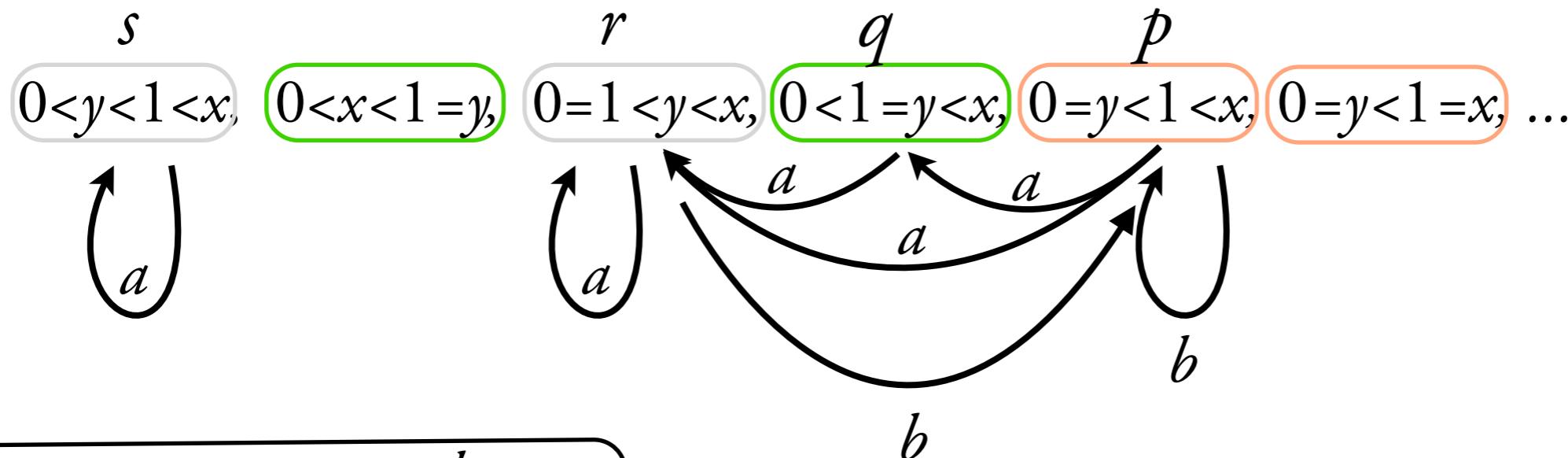
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



	a	a	a	ℓ
x	5	5	5	5
y	0	1	2	4
	\wr	\wr		
x	2	2		
y	0	1		

~~bisimulation~~ → runs of the region
automaton correspond to runs of \mathcal{A}

Discrete case

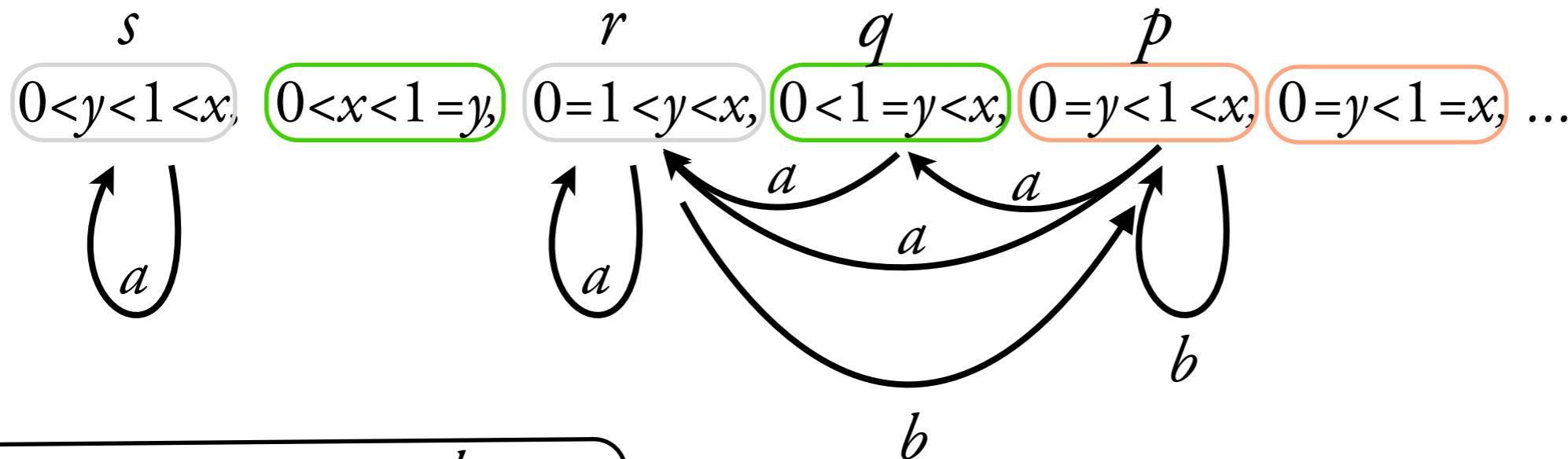
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 1)$$



	a	a	a	ℓ
x	5	5	5	5
y	0	1	2	4
	?	?	?	?
x	2	2	2	
y	0	1	?	

~~bisimulation~~ → runs of the region
automaton correspond to runs of \mathcal{A}

Discrete case

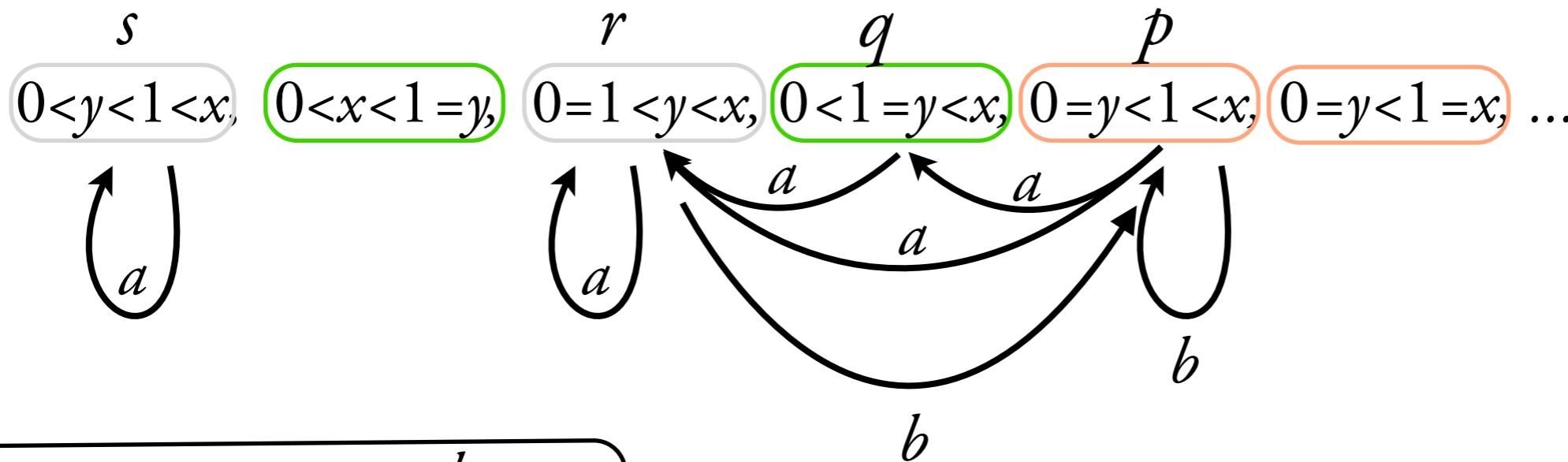
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x>0) \wedge (y=0)$$

$$\delta_a: (x'=x) \wedge (y < y' < x)$$

$$\delta_b: (x'=x) \wedge (y'=0)$$

$$\tau_F: (y=1)$$



	a	a	a	ℓ
x	5	5	5	5
y	0	1	2	4
	?	?	?	?
x	2	2	2	
y	0	1	?	

but: in any cell and any n we can find configurations which are *sufficiently good* for all runs of length $\leq n$
 → *finite* runs of the region automaton correspond to runs of \mathcal{A}

Discrete case

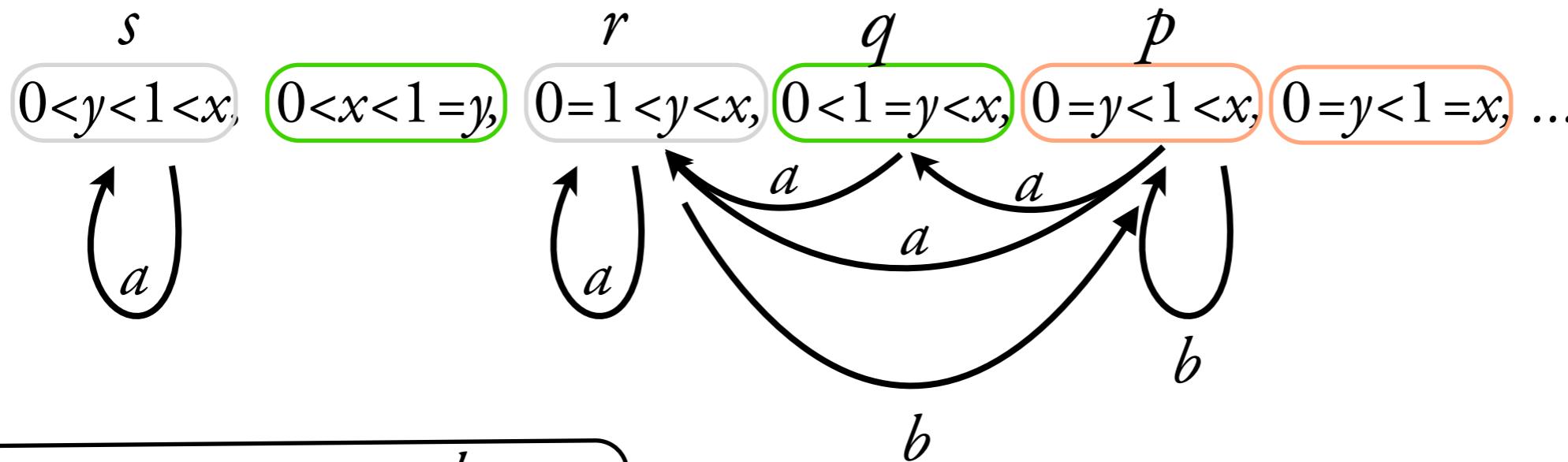
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 3)$$



	a	a	a	ℓ
x	5	5	5	5
y	0	1	2	4
	?	?	?	
x	2	2	2	
y	0	1	?	

but: in any cell and any n we can find configurations which are *sufficiently good* for all runs of length $\leq n$
 → *finite* runs of the region automaton correspond to runs of \mathcal{A}

Discrete case

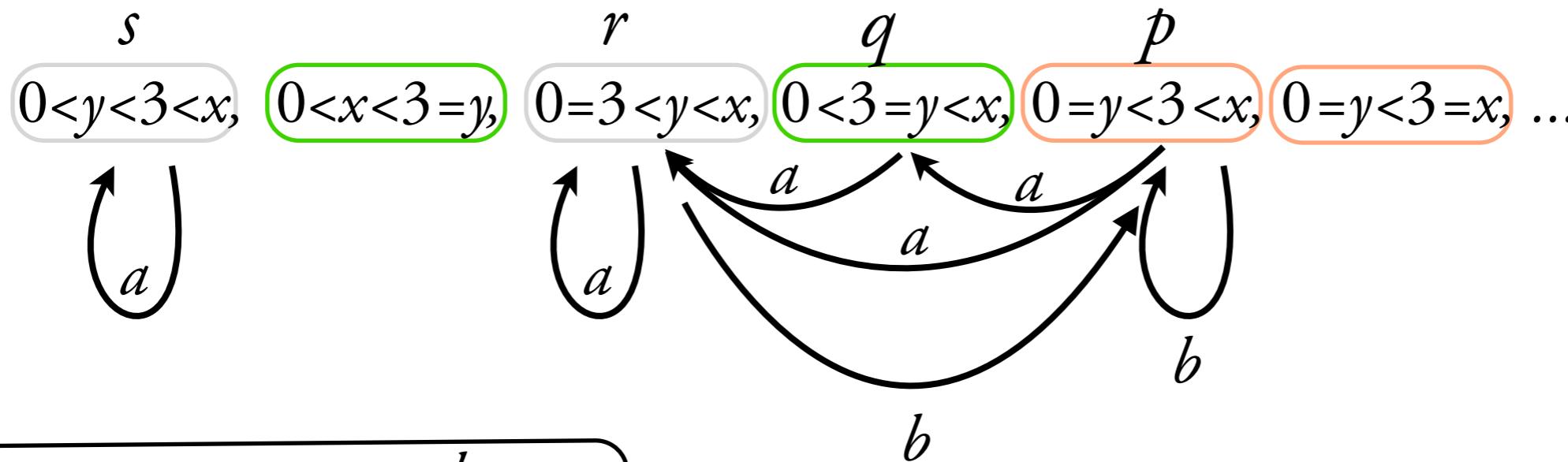
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 3)$$



	a	a	a	ℓ
x	5	5	5	5
y	0	1	2	4
	?	?	?	
x	2	2	2	
y	0	1	?	

but: in any cell and any n we can find configurations which are *sufficiently good* for all runs of length $\leq n$
 → *finite* runs of the region automaton correspond to runs of \mathcal{A}

Discrete case

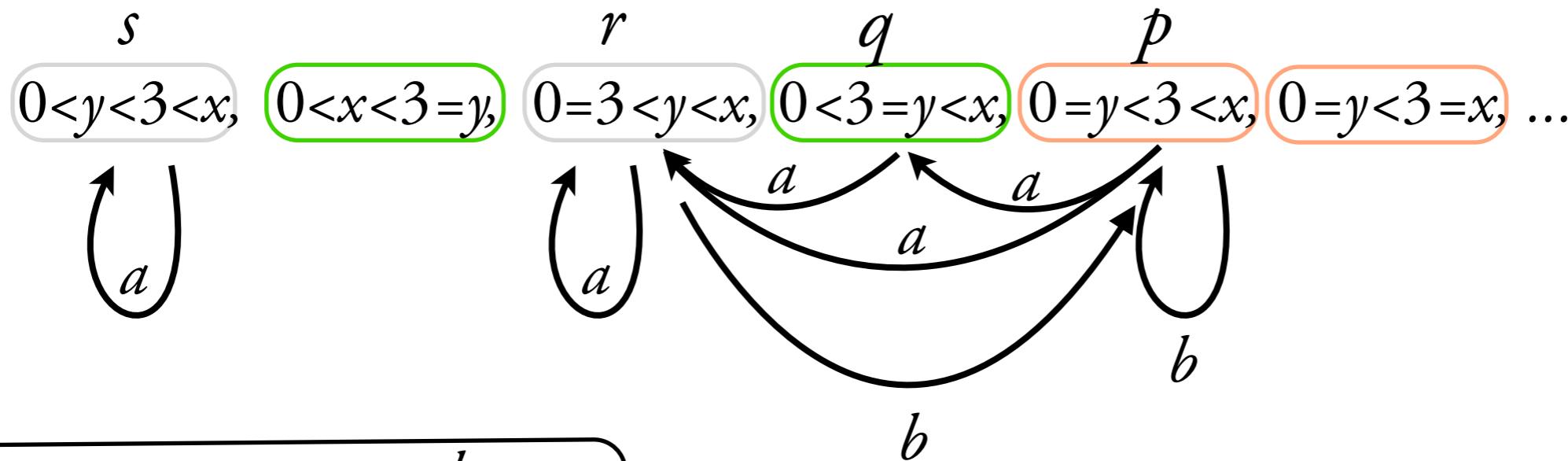
- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 3)$$



	a	a	a	ℓ
x	5	5	5	5
y	0	1	2	4
	?	?	?	0
x	2	2	2	
y	0	1	?	

but: in any cell and any n we can find configurations which are *sufficiently good* for all runs of length $\leq n$
 \rightarrow *finite* runs of the region automaton correspond to runs of \mathcal{A}

Discrete case

- $D = \mathbb{N}$
- finite words
- no database

$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 3)$$

Solution: consider 0,1,2,3 as special values (constants)

Discrete case

- $D = \mathbb{N}$
- finite words
- no database

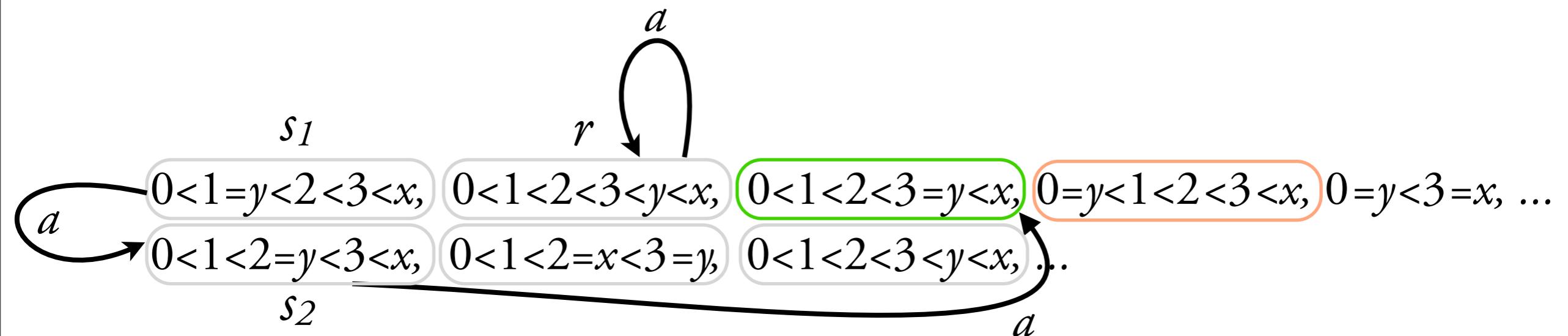
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x)$$

$$\delta_b: (x' = x) \wedge (y' = 0)$$

$$\tau_F: (y = 3)$$

Solution: consider 0,1,2,3 as special values (constants)



Discrete case

- $D = \mathbb{N}$
- finite words
- no database

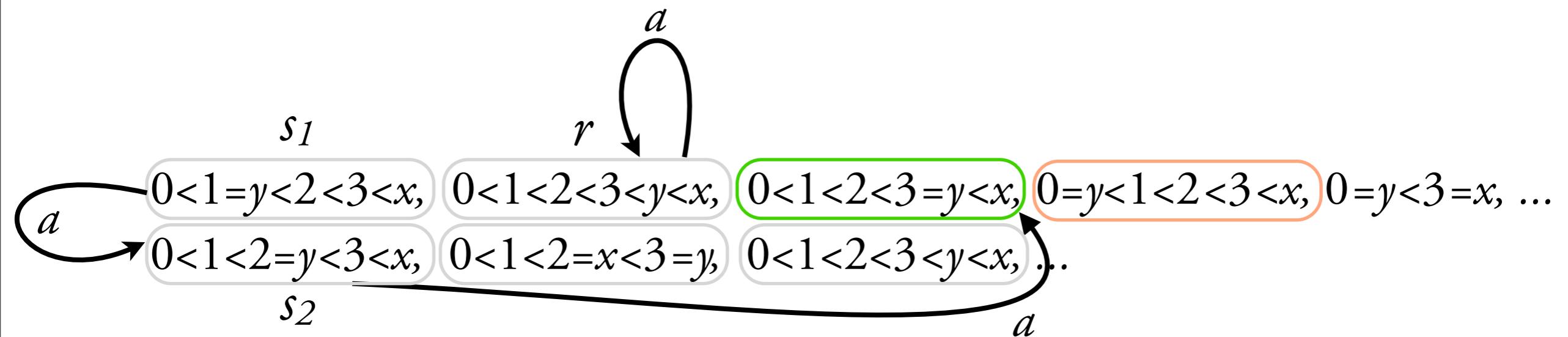
$$\tau_I: (x>0) \wedge (y=0)$$

$$\delta_a: (x'=x) \wedge (y < y' < x)$$

$$\delta_b: (x'=x) \wedge (y'=0)$$

$$\tau_F: (y=3)$$

Solution: consider 0,1,2,3 as special values (constants)



Then the obtained cell automaton simulates \mathcal{A} fatefully

Discrete case

- $D = \mathbb{N}$
- finite words
- no database

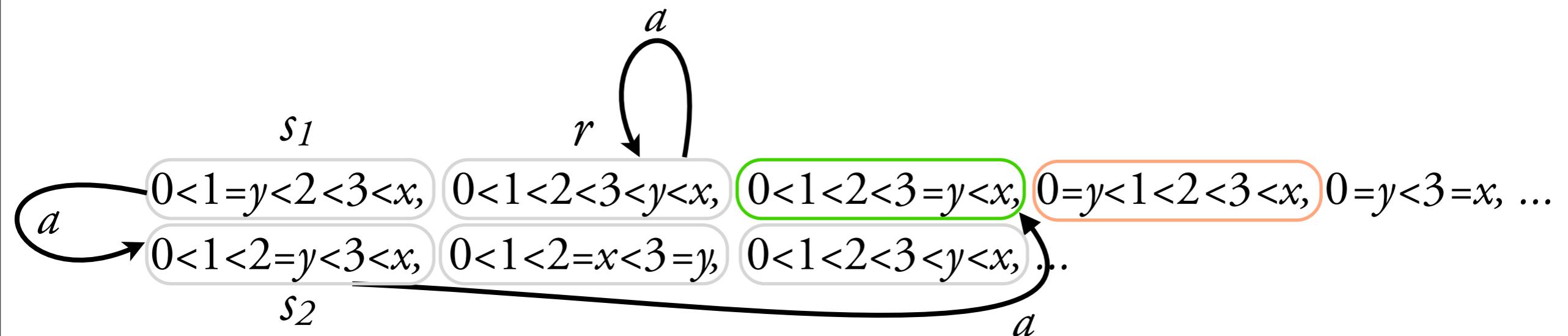
$$\tau_I: (x>0) \wedge (y=0)$$

$$\delta_a: (x'=x) \wedge (y < y' < x)$$

$$\delta_b: (x'=x) \wedge (y'=0)$$

$$\tau_F: (y=3)$$

Solution: consider 0,1,2,3 as special values (constants)



Then the obtained cell automaton simulates \mathcal{A} fatefully

Theorem. For any linearly ordered structure there is a finite number of special values

This solves the case of finite runs, with no database.

This solves the case of finite runs, with no database.

Theorem. For all \mathcal{D} , \mathcal{D} -automata over finite words accept regular languages.

Theorem. Emptiness of \mathcal{D} -automata is decidable* in PSPACE

**For any reasonable linearly ordered structure* - there should be an efficient algorithm which can compute the special values, given some initial constants

Adding a database

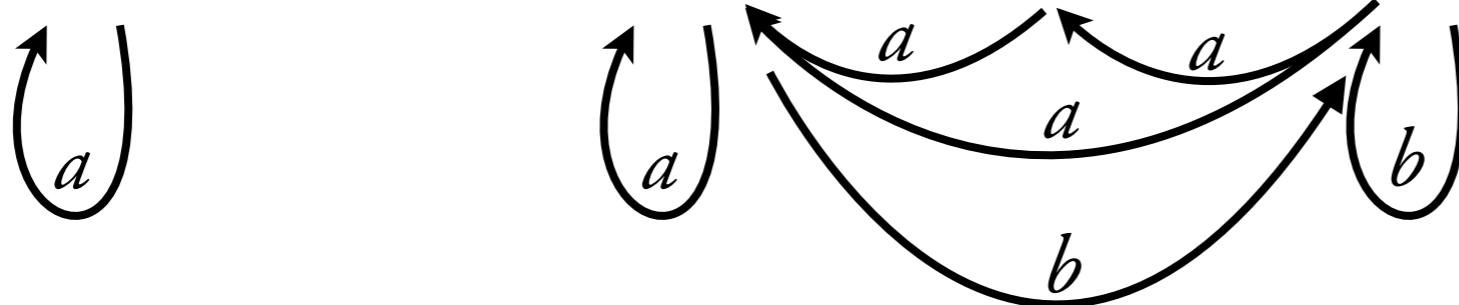
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x) \wedge R(x, y)$$

$$\delta_b: (y' = 0) \wedge \neg R(x, y)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, \quad 0 < x < 1 = y, \quad 0 = 1 < y < x, \quad 0 < 1 = y < x, \quad 0 = y < 1 < x, \quad 0 = y < 1 = x, \dots$



can be guaranteed
to be new

is forced to be zero

Adding a database

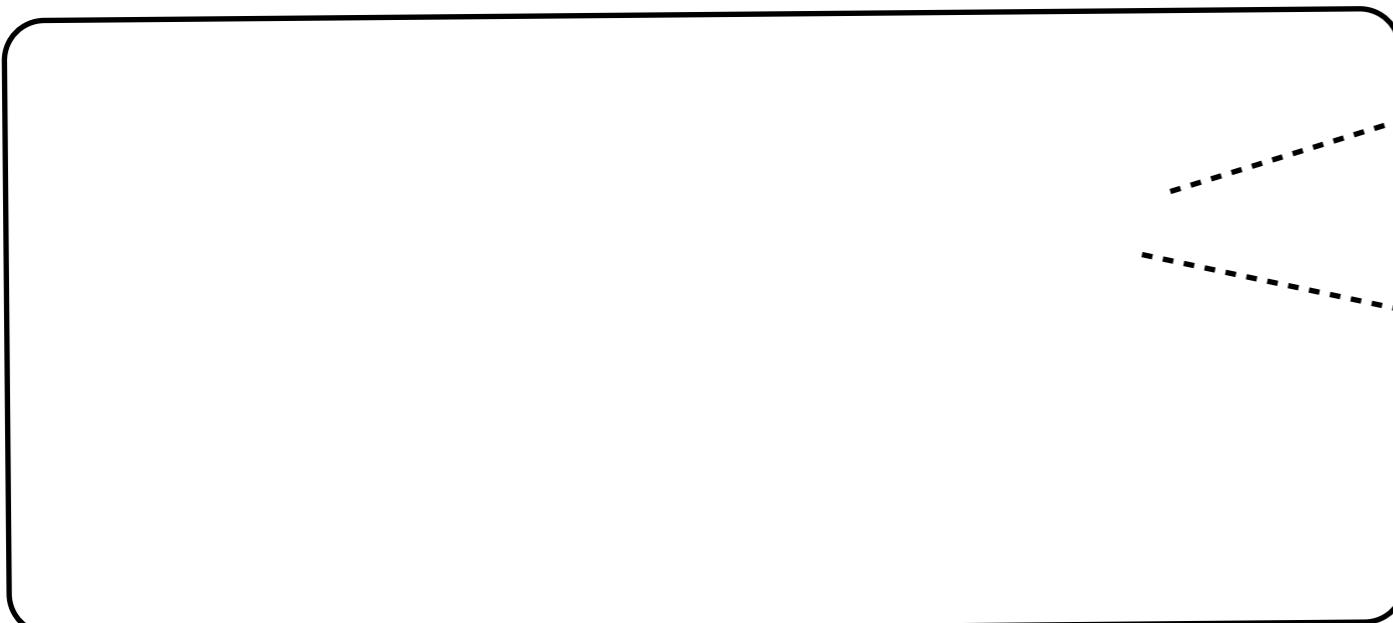
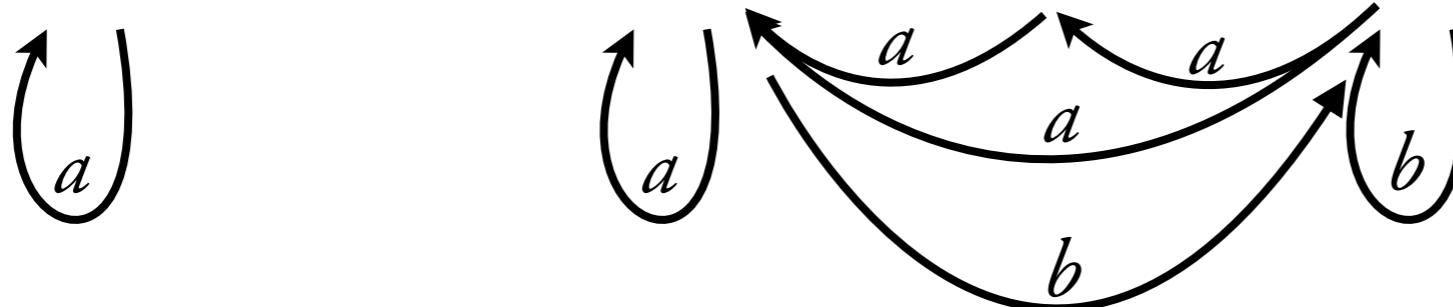
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x) \wedge R(x, y)$$

$$\delta_b: (y' = 0) \wedge \neg R(x, y)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, \quad 0 < x < 1 = y, \quad 0 = 1 < y < x, \quad 0 < 1 = y < x, \quad 0 = y < 1 < x, \quad 0 = y < 1 = x, \dots$



can be guaranteed
to be new

is forced to be zero

Adding a database

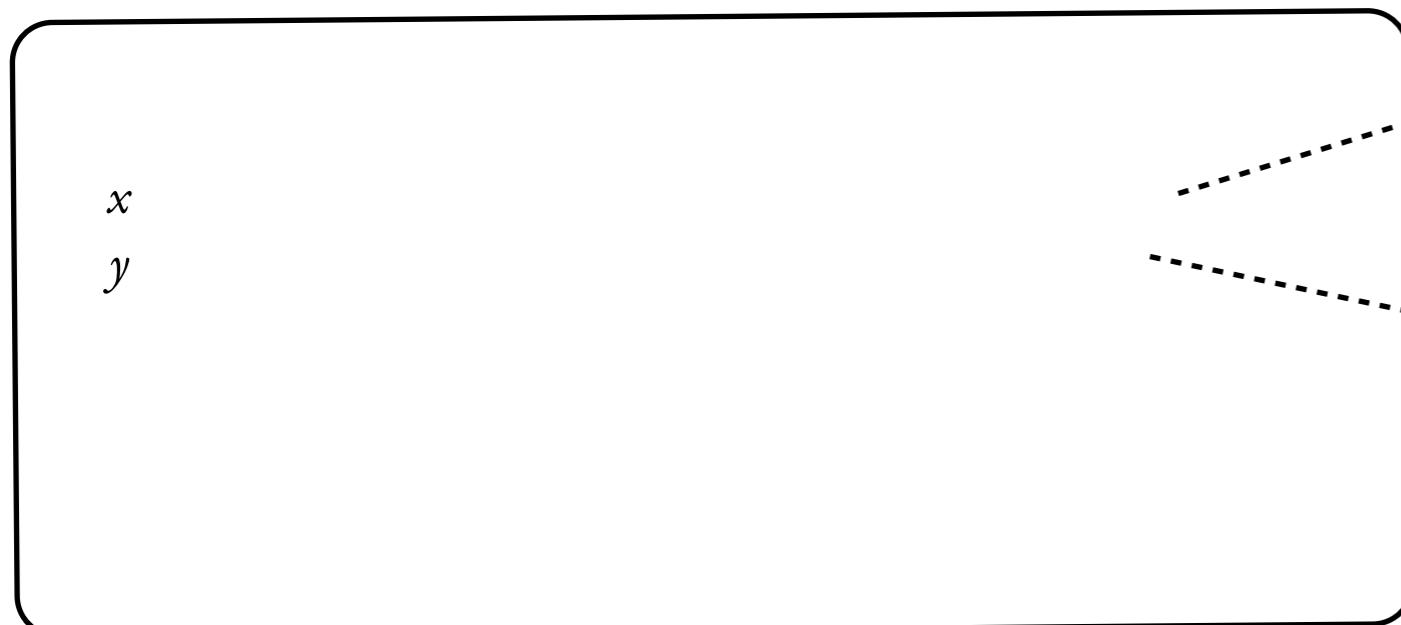
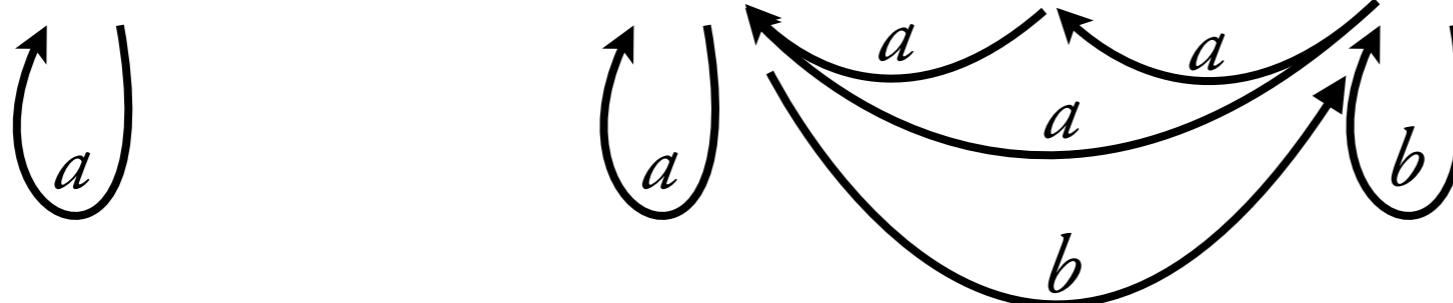
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x) \wedge R(x, y)$$

$$\delta_b: (y' = 0) \wedge \neg R(x, y)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, \quad 0 < x < 1 = y, \quad 0 = 1 < y < x, \quad 0 < 1 = y < x, \quad 0 = y < 1 < x, \quad 0 = y < 1 = x, \dots$

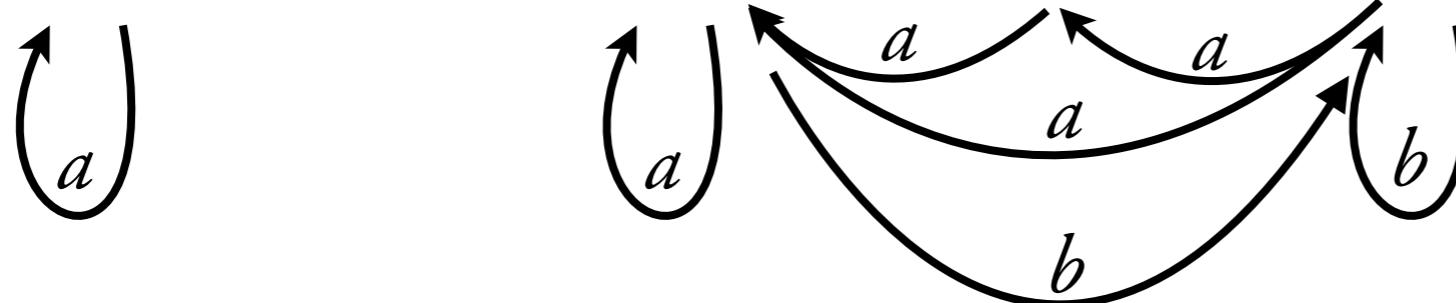


can be guaranteed
to be new

is forced to be zero

$\tau_I: (x > 0) \wedge (y = 0)$ $\delta_a: (x' = x) \wedge (y < y' < x) \wedge R(x, y)$ $\delta_b: (y' = 0) \wedge \neg R(x, y)$ $\tau_F: (y = 1)$

$0 < y < 1 < x$, $0 < x < 1 = y$, $0 = 1 < y < x$, $0 < 1 = y < x$, $0 = y < 1 < x$, $0 = y < 1 = x$, ...



x	5	5	5	5	6
y	0	1	2	4	0

A table showing a sequence of points (x, y) . The x-values are 5, 5, 5, 5, 6. The y-values are 0, 1, 2, 4, 0. Curved arrows labeled 'a' connect the first four points sequentially. A curved arrow labeled 'b' connects the fourth point (x=5, y=4) back to the curve at the fifth point (x=6, y=0).

can be guaranteed
to be new

is forced to be zero

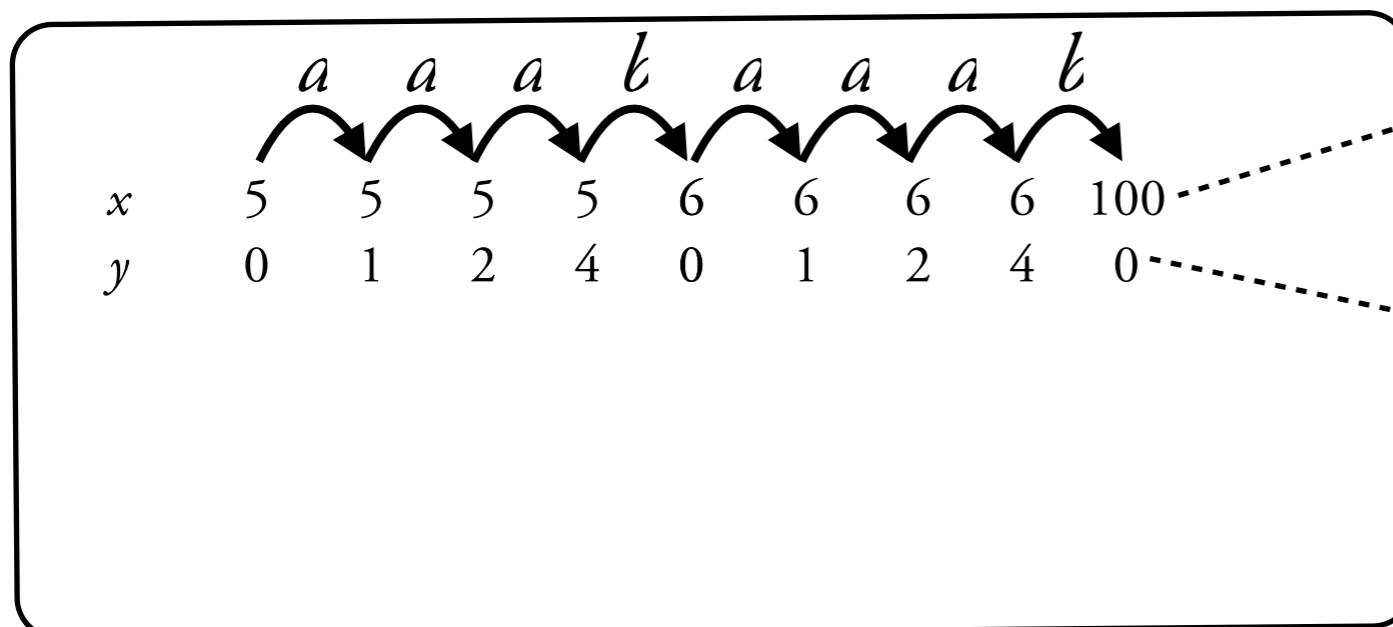
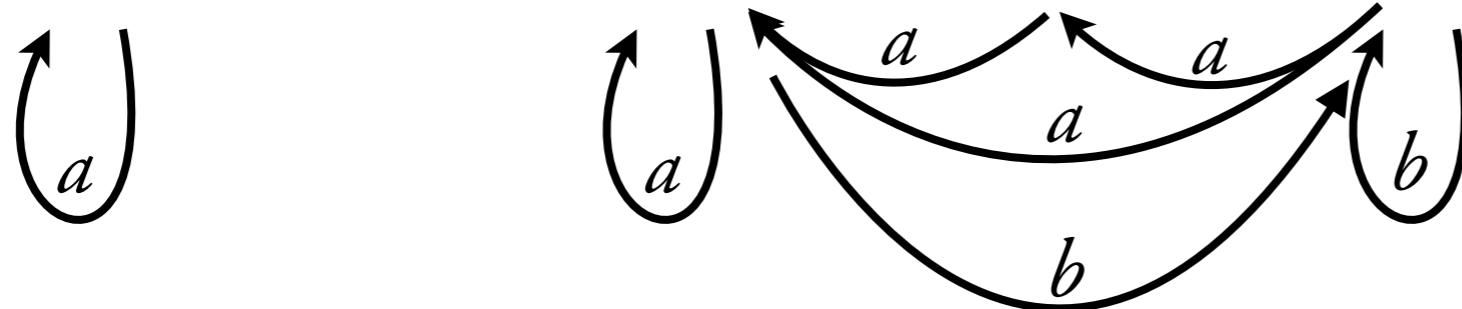
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x) \wedge R(x, y)$$

$$\delta_b: (y' = 0) \wedge \neg R(x, y)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x$, $0 < x < 1 = y$, $0 = 1 < y < x$, $0 < 1 = y < x$, $0 = y < 1 < x$, $0 = y < 1 = x$, ...



can be guaranteed
to be new

is forced to be zero

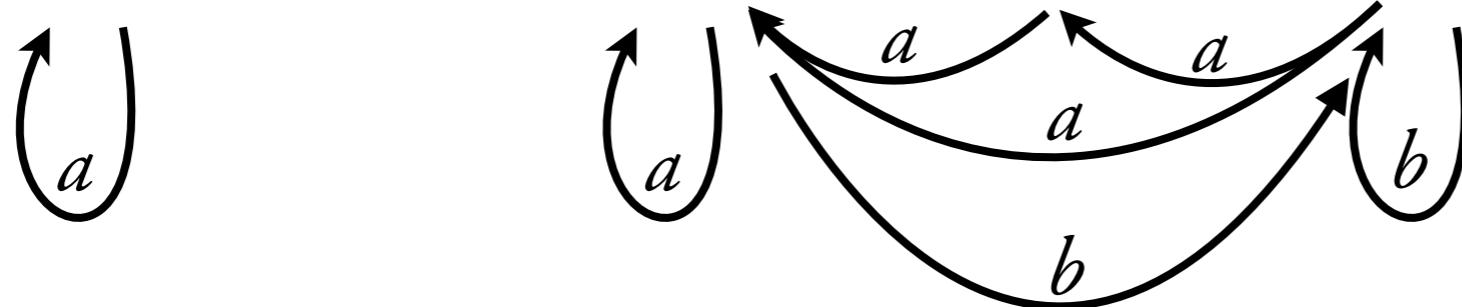
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x) \wedge R(x, y)$$

$$\delta_b: (y' = 0) \wedge \neg R(x, y)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, 0 < x < 1 = y, 0 = 1 < y < x, 0 < 1 = y < x, 0 = y < 1 < x, 0 = y < 1 = x, \dots$



	a	a	a	ℓ	a	a	a	ℓ
x	5	5	5	5	6	6	6	100
y	0	1	2	4	0	1	2	0

\cap_R

can be guaranteed
to be new

is forced to be zero

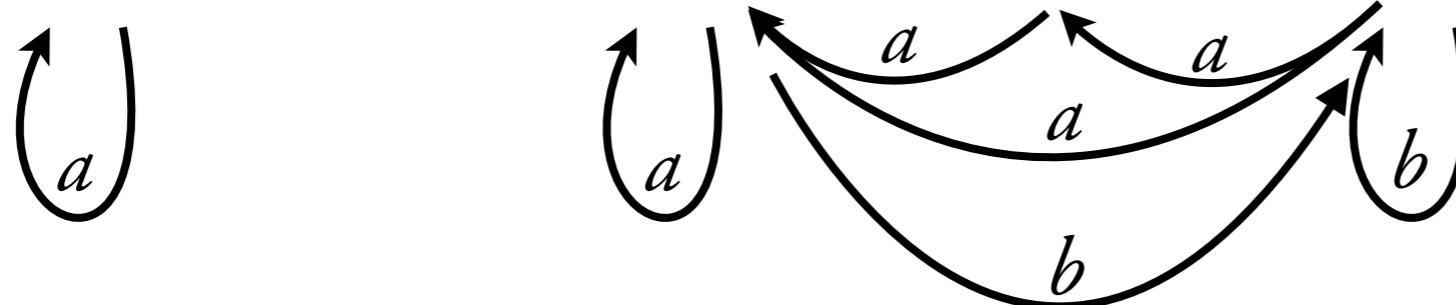
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x) \wedge R(x, y)$$

$$\delta_b: (y' = 0) \wedge \neg R(x, y)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x, \quad 0 < x < 1 = y, \quad 0 = 1 < y < x, \quad 0 < 1 = y < x, \quad 0 = y < 1 < x, \quad 0 = y < 1 = x, \dots$



can be guaranteed
to be new

	a	a	a	ℓ	a	a	a	ℓ
x	5	5	5	5	6	6	6	100
y	0	1	2	4	0	1	2	4
\cap	R	R						

is forced to be zero

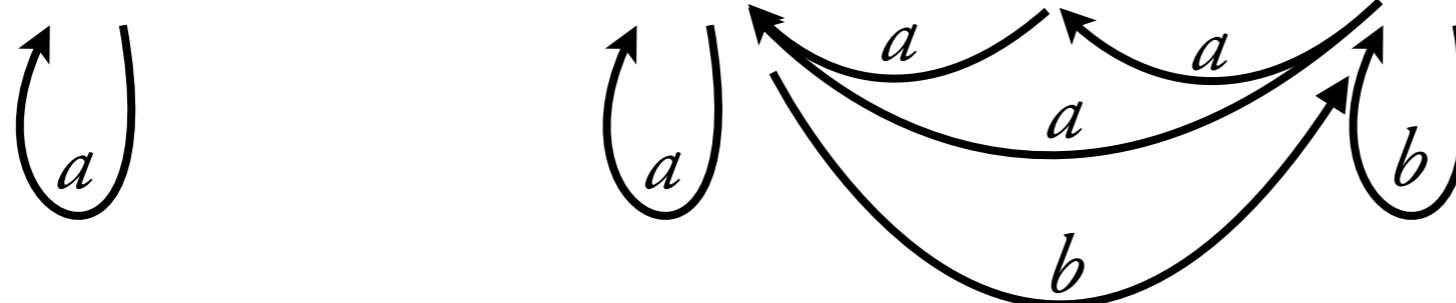
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x) \wedge R(x, y)$$

$$\delta_b: (y' = 0) \wedge \neg R(x, y)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x$, $0 < x < 1 = y$, $0 = 1 < y < x$, $0 < 1 = y < x$, $0 = y < 1 < x$, $0 = y < 1 = x$, ...



can be guaranteed
to be new

	a	a	a	ℓ	a	a	a	ℓ
x	5	5	5	5	6	6	6	100
y	0	1	2	4	0	1	2	0
\cap	R	R	R					

is forced to be zero

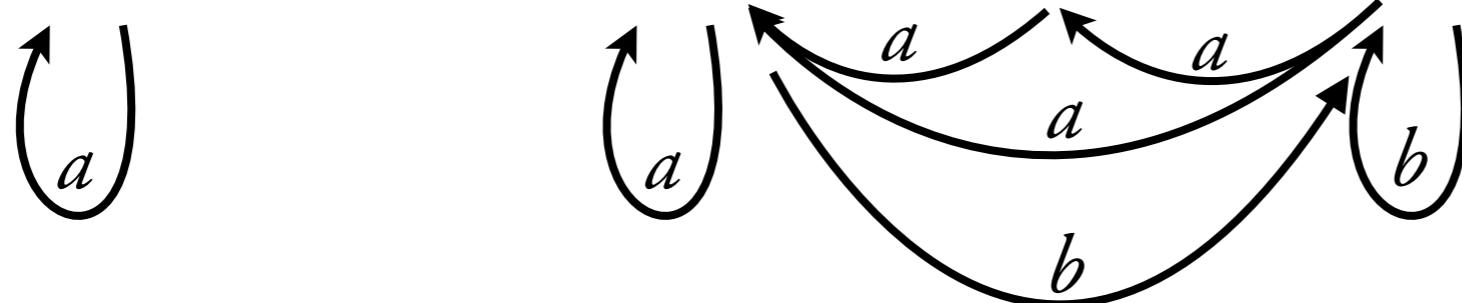
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x) \wedge R(x, y)$$

$$\delta_b: (y' = 0) \wedge \neg R(x, y)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x$, $0 < x < 1 = y$, $0 = 1 < y < x$, $0 < 1 = y < x$, $0 = y < 1 < x$, $0 = y < 1 = x$, ...



can be guaranteed
to be new

	a	a	a	ℓ	a	a	a	ℓ
x	5	5	5	5	6	6	6	100
y	0	1	2	4	0	1	2	4
\cap	R	R	R	R				

is forced to be zero

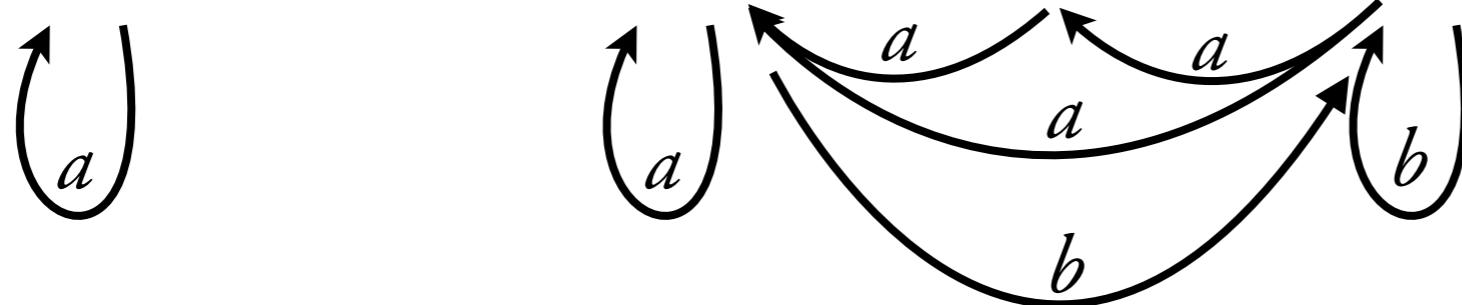
$$\tau_I: (x > 0) \wedge (y = 0)$$

$$\delta_a: (x' = x) \wedge (y < y' < x) \wedge R(x, y)$$

$$\delta_b: (y' = 0) \wedge \neg R(x, y)$$

$$\tau_F: (y = 1)$$

$0 < y < 1 < x$, $0 < x < 1 = y$, $0 = 1 < y < x$, $0 < 1 = y < x$, $0 = y < 1 < x$, $0 = y < 1 = x$, ...



can be guaranteed
to be new

	a	a	a	ℓ	a	a	a	ℓ
x	5	5	5	5	6	6	6	100
y	0	1	2	4	0	1	2	4
\cap	R	R	R	R	$\not R$			

is forced to be zero

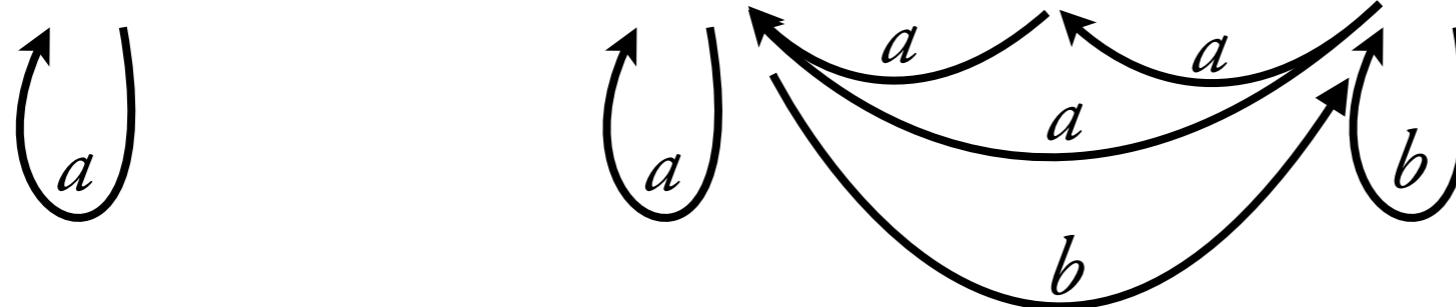
$$\tau_I: (x>0) \wedge (y=0)$$

$$\delta_a: (x'=x) \wedge (y < y' < x) \wedge R(x,y)$$

$$\delta_b: (y'=0) \wedge \neg R(x,y)$$

$$\tau_F: (y=1)$$

$0 < y < 1 < x$, $0 < x < 1 = y$, $0 = 1 < y < x$, $0 < 1 = y < x$, $0 = y < 1 < x$, $0 = y < 1 = x$, ...



can be guaranteed
to be new

	a	a	a	ℓ	a	a	a	ℓ
x	5	5	5	5	6	6	6	100
y	0	1	2	4	0	1	2	4
\cap	R	R	R	R	$\not\in$	$\not\in$	R	

is forced to be zero

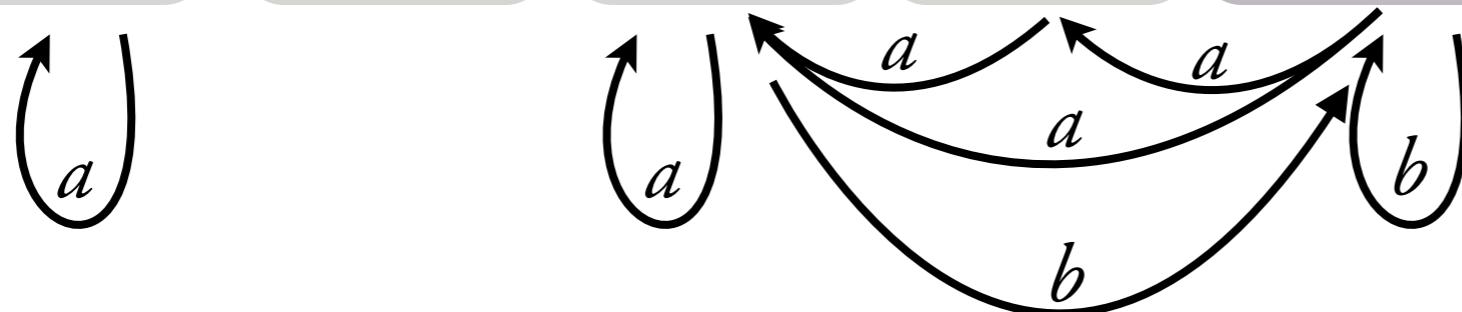
$$\tau_I: (x>0) \wedge (y=0)$$

$$\delta_a: (x'=x) \wedge (y < y' < x) \wedge R(x,y)$$

$$\delta_b: (y'=0) \wedge \neg R(x,y)$$

$$\tau_F: (y=1)$$

$0 < y < 1 < x$, $0 < x < 1 = y$, $0 = 1 < y < x$, $0 < 1 = y < x$, $0 = y < 1 < x$, $0 = y < 1 = x$, ...



can be guaranteed
to be new

	a	a	a	ℓ	a	a	a	ℓ
x	5	5	5	5	6	6	6	100
y	0	1	2	4	0	1	2	4
\cap	R	R	R	R	$\not\in$	R	$\not\in$	R

is forced to be zero

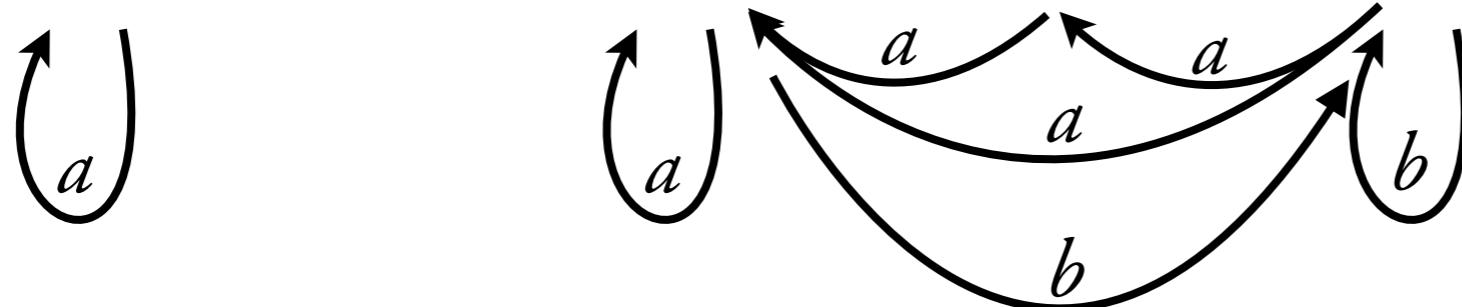
$$\tau_I: (x>0) \wedge (y=0)$$

$$\delta_a: (x'=x) \wedge (y < y' < x) \wedge R(x,y)$$

$$\delta_b: (y'=0) \wedge \neg R(x,y)$$

$$\tau_F: (y=1)$$

$0 < y < 1 < x$, $0 < x < 1 = y$, $0 = 1 < y < x$, $0 < 1 = y < x$, $0 = y < 1 < x$, $0 = y < 1 = x$, ...



can be guaranteed
to be new

	a	a	a	ℓ	a	a	a	ℓ
x	5	5	5	5	6	6	6	100
y	0	1	2	4	0	1	2	4
\cap	R	R	R	R	$\not R$	R	R	$\not R$

is forced to be zero

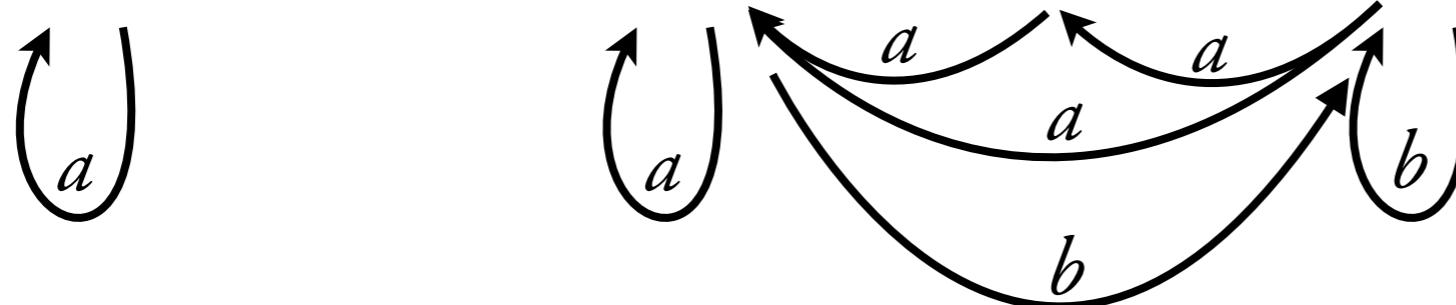
$$\tau_I: (x>0) \wedge (y=0)$$

$$\delta_a: (x'=x) \wedge (y < y' < x) \wedge R(x,y)$$

$$\delta_b: (y'=0) \wedge \neg R(x,y)$$

$$\tau_F: (y=1)$$

$0 < y < 1 < x$, $0 < x < 1 = y$, $0 = 1 < y < x$, $0 < 1 = y < x$, $0 = y < 1 < x$, $0 = y < 1 = x$, ...



can be guaranteed
to be new

	a	a	a	ℓ	a	a	a	ℓ
x	5	5	5	5	6	6	6	100
y	0	1	2	4	0	1	2	4
\cap	R	R	R	R	R	R	R	R

is forced to be zero

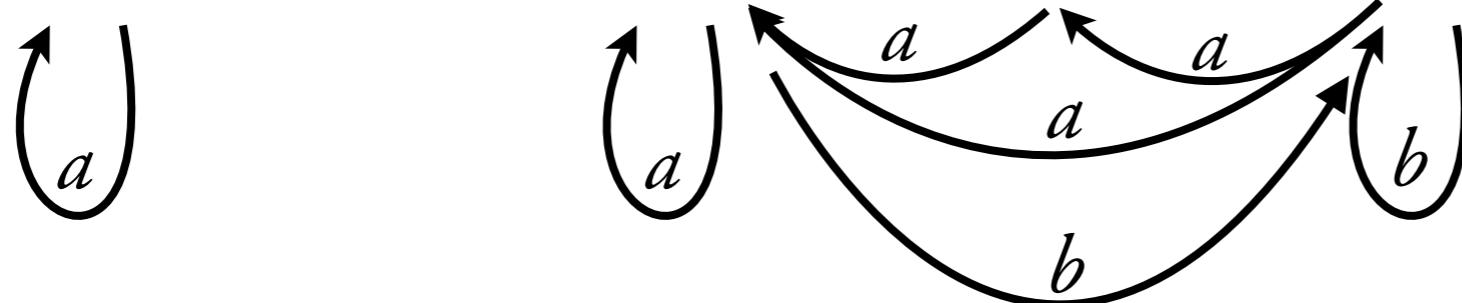
$$\tau_I: (x>0) \wedge (y=0)$$

$$\delta_a: (x'=x) \wedge (y < y' < x) \wedge R(x,y)$$

$$\delta_b: (y'=0) \wedge \neg R(x,y)$$

$$\tau_F: (y=1)$$

$0 < y < 1 < x$, $0 < x < 1 = y$, $0 = 1 < y < x$, $0 < 1 = y < x$, $0 = y < 1 < x$, $0 = y < 1 = x$, ...



can be guaranteed
to be new

	a	a	a	ℓ	a	a	a	ℓ
x	5	5	5	5	6	6	6	100
y	0	1	2	4	0	1	2	4
\cap	R	R	R	R	R	R	R	R

is forced to be zero

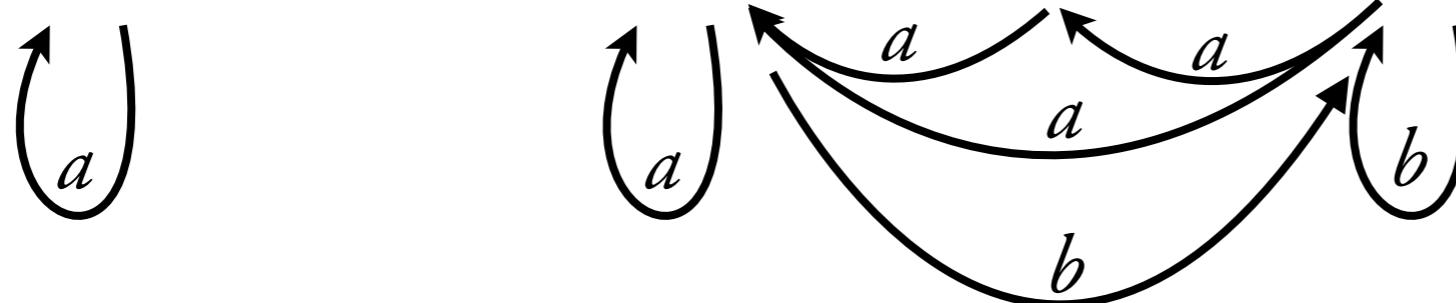
$$\tau_I: (x>0) \wedge (y=0)$$

$$\delta_a: (x'=x) \wedge (y < y' < x) \wedge R(x,y)$$

$$\delta_b: (y'=0) \wedge \neg R(x,y)$$

$$\tau_F: (y=1)$$

$0 < y < 1 < x$, $0 < x < 1 = y$, $0 = 1 < y < x$, $0 < 1 = y < x$, $0 = y < 1 < x$, $0 = y < 1 = x$, ...



can be guaranteed
to be new

	a	a	a	ℓ	a	a	a	ℓ
x	5	5	5	5	6	6	6	100
y	0	1	2	4	0	1	2	4
\cap	R	R	R	R	\cancel{R}	R	R	\cancel{R}

is forced to be zero

En elaboration of these ideas solves the problem for automata with database, over infinite words:

En elaboration of these ideas solves the problem for automata with database, over infinite words:

Theorem. For all \mathcal{D} , \mathcal{D} -automata over infinite words accept ω B-regular languages.

En elaboration of these ideas solves the problem for automata with database, over infinite words:

Theorem. For all \mathcal{D} , \mathcal{D} -automata over infinite words accept ω B-regular languages.

Theorem. Emptiness of \mathcal{D} -automata is decidable* in PSPACE

*For any reasonable linearly ordered structure \mathcal{D}

En elaboration of these ideas solves the problem for automata with database, over infinite words:

Theorem. For all \mathcal{D} , \mathcal{D} -automata over infinite words accept ω B-regular languages.

Theorem. Emptiness of \mathcal{D} -automata is decidable* in PSPACE

*For any reasonable linearly ordered structure \mathcal{D}

Corollary. Deciding LTL₊(data tests) properties of \mathcal{D} -automata is PSPACE-complete.

Thank you!