Automata based verification over linearly ordered data domains

Luc Segoufin
INRIA and ENS Cachan

Szymon Toruńczyk
University of Warsaw

Automata based verification over linearly ordered data domains

Luc Segoufin
INRIA and ENS Cachan

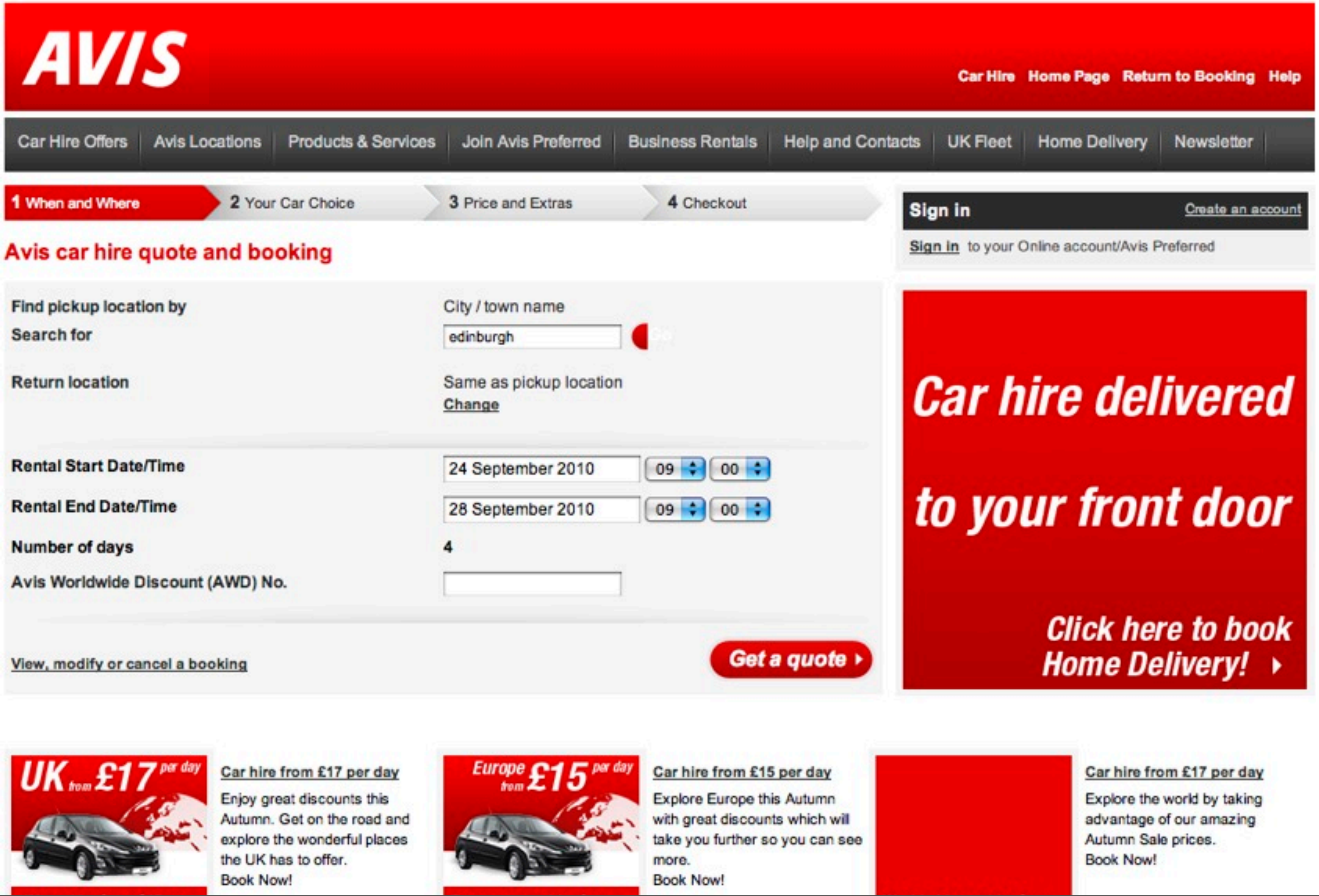
Szymon Toruńczyk
University of Warsaw

Motivation

Motivation

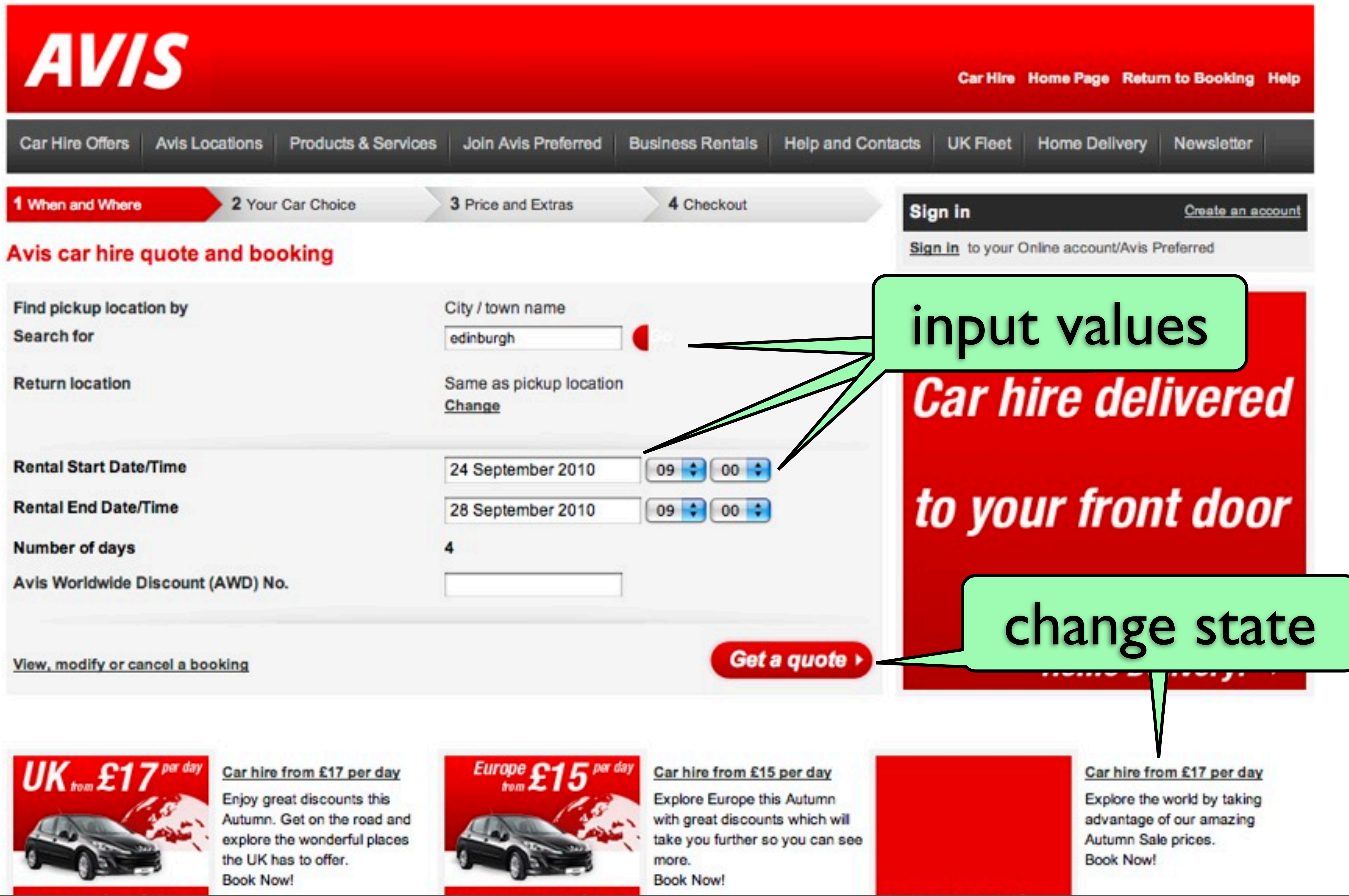
Motivation

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



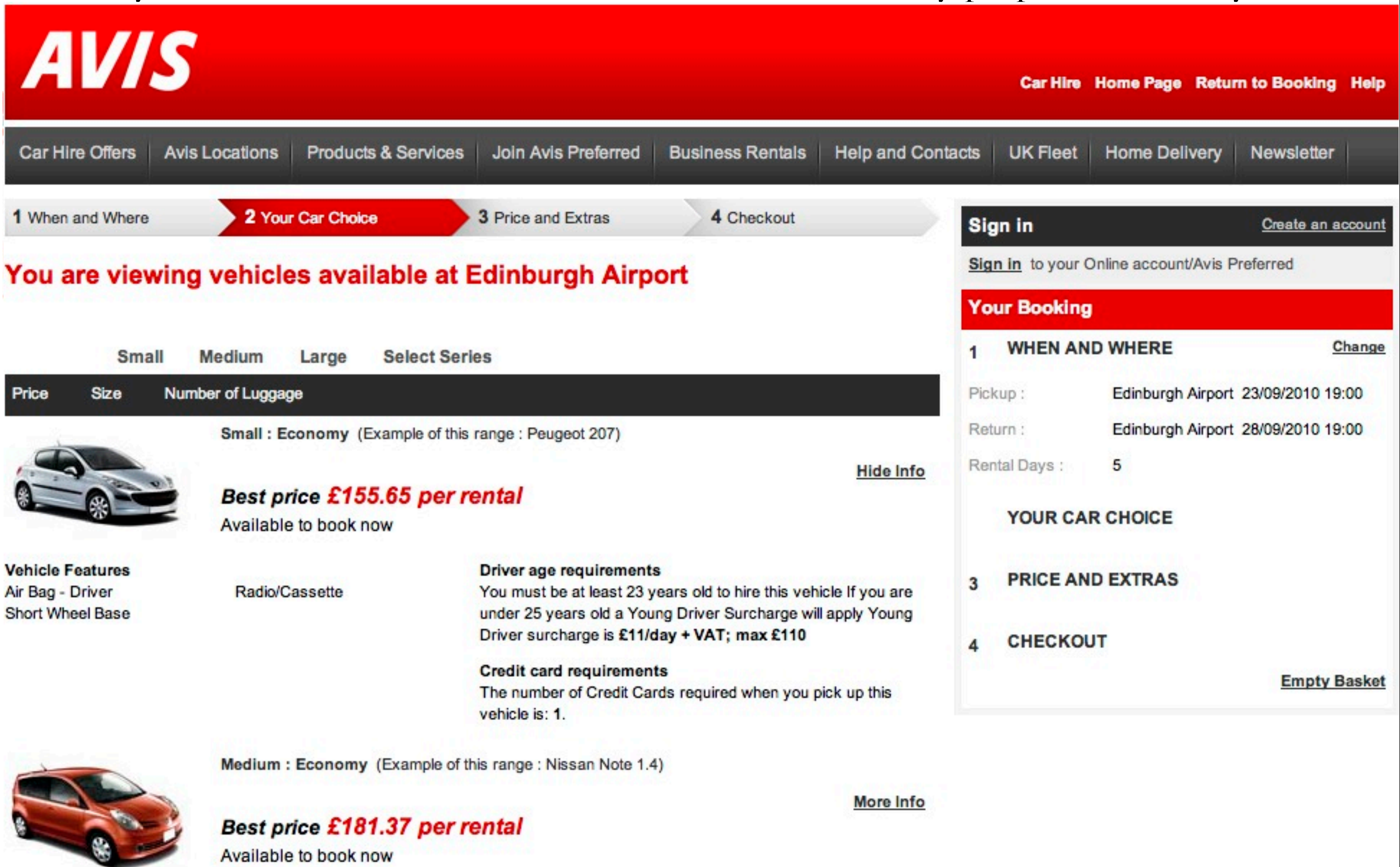
Motivation

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



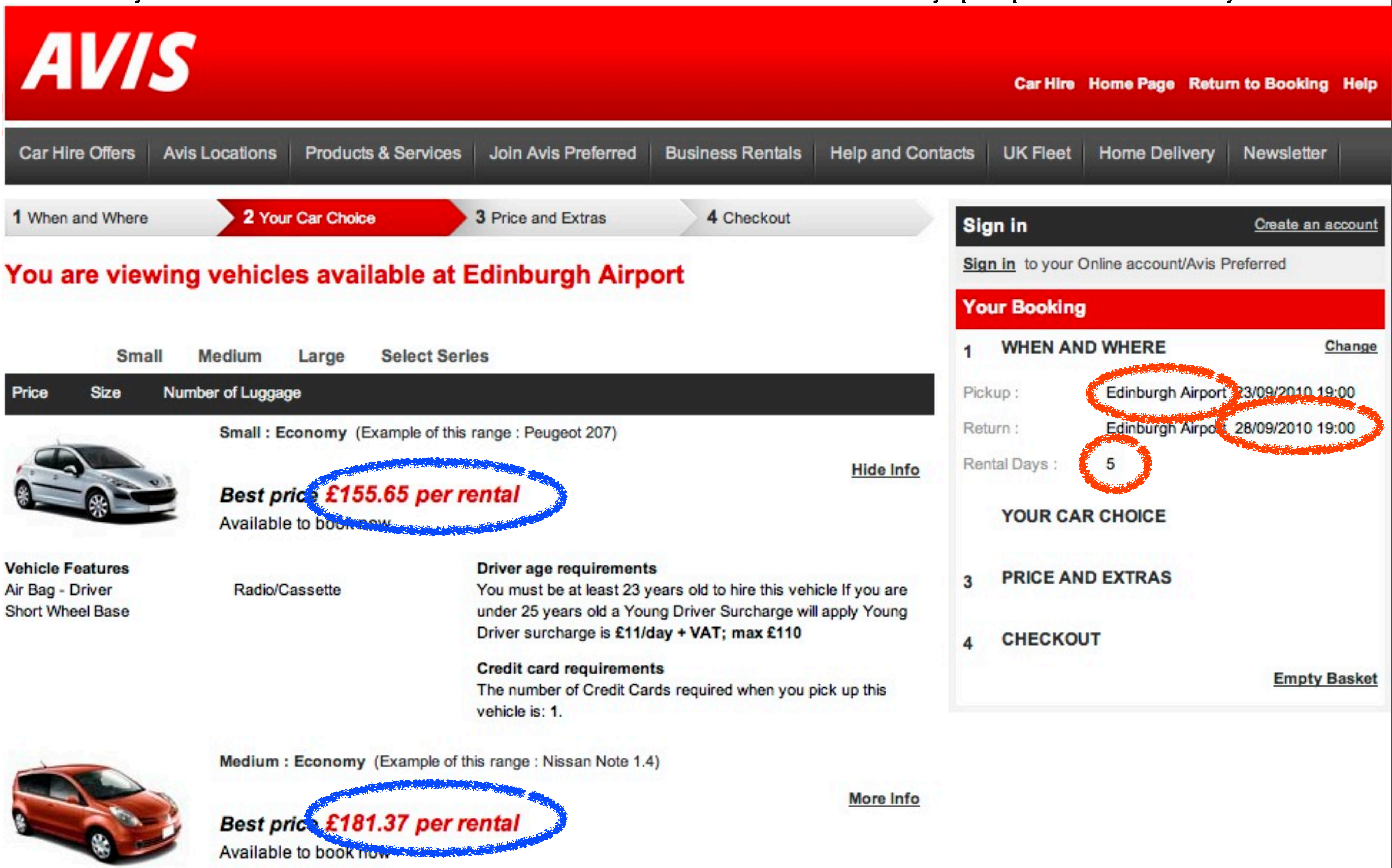
Motivation

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



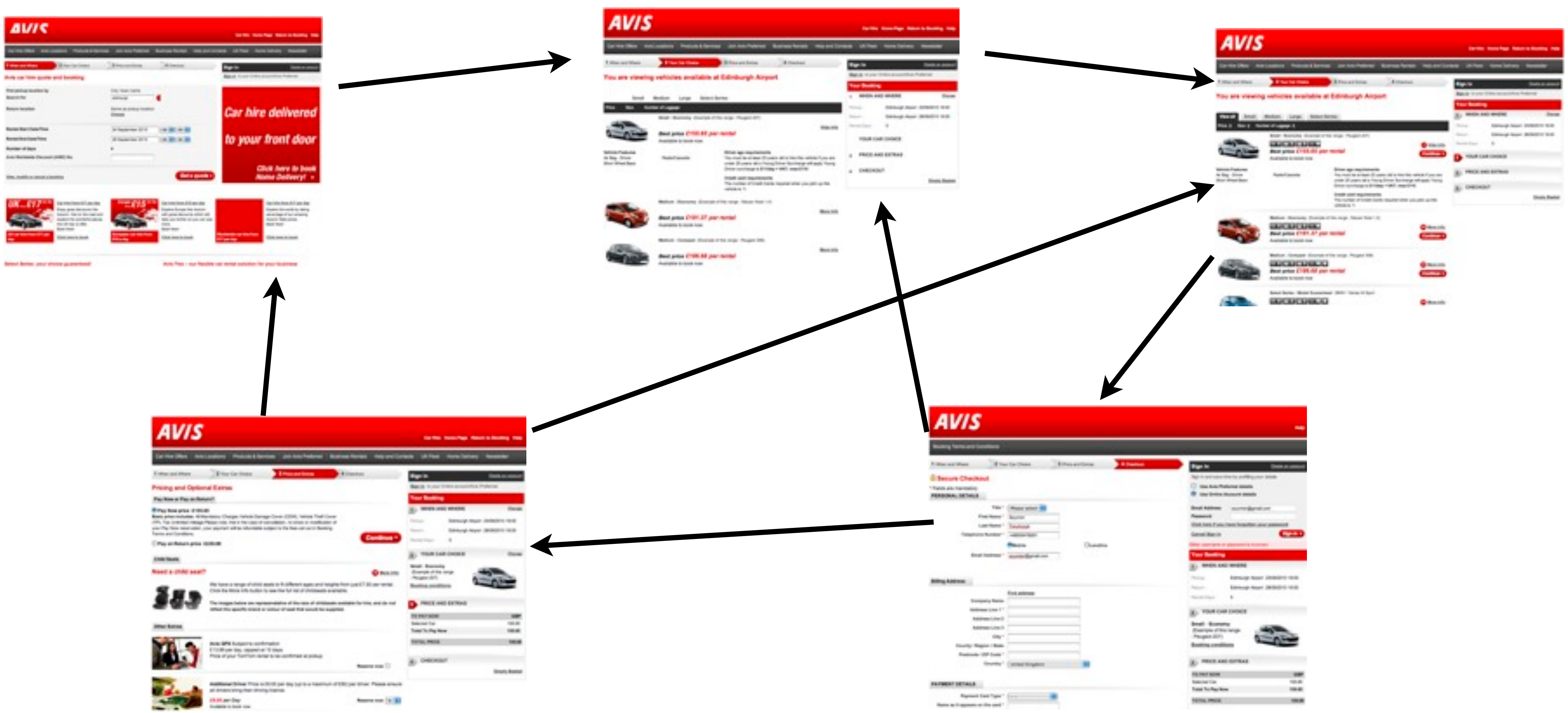
Motivation

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



Motivation

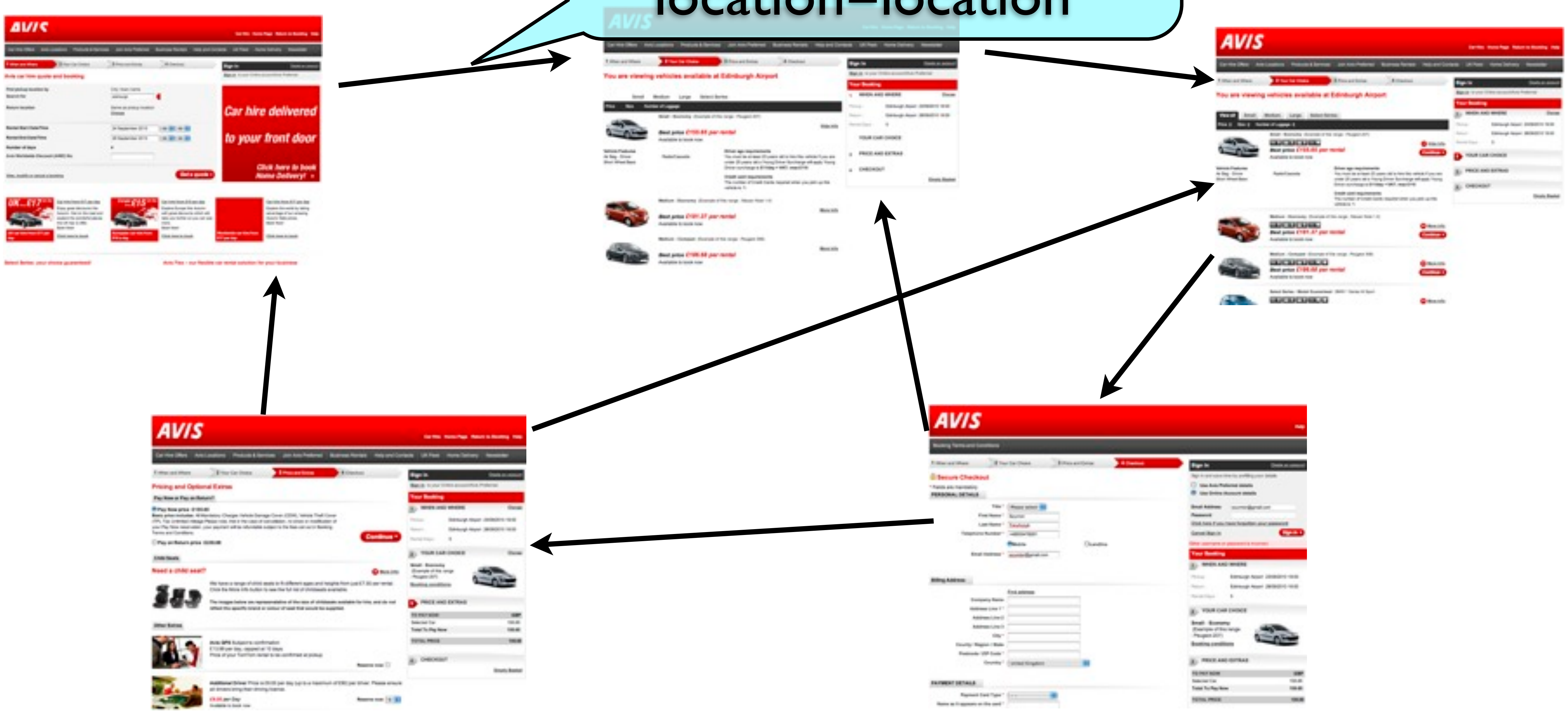
Given a system which refers to a database and data values, verify properties of the system



Motivation

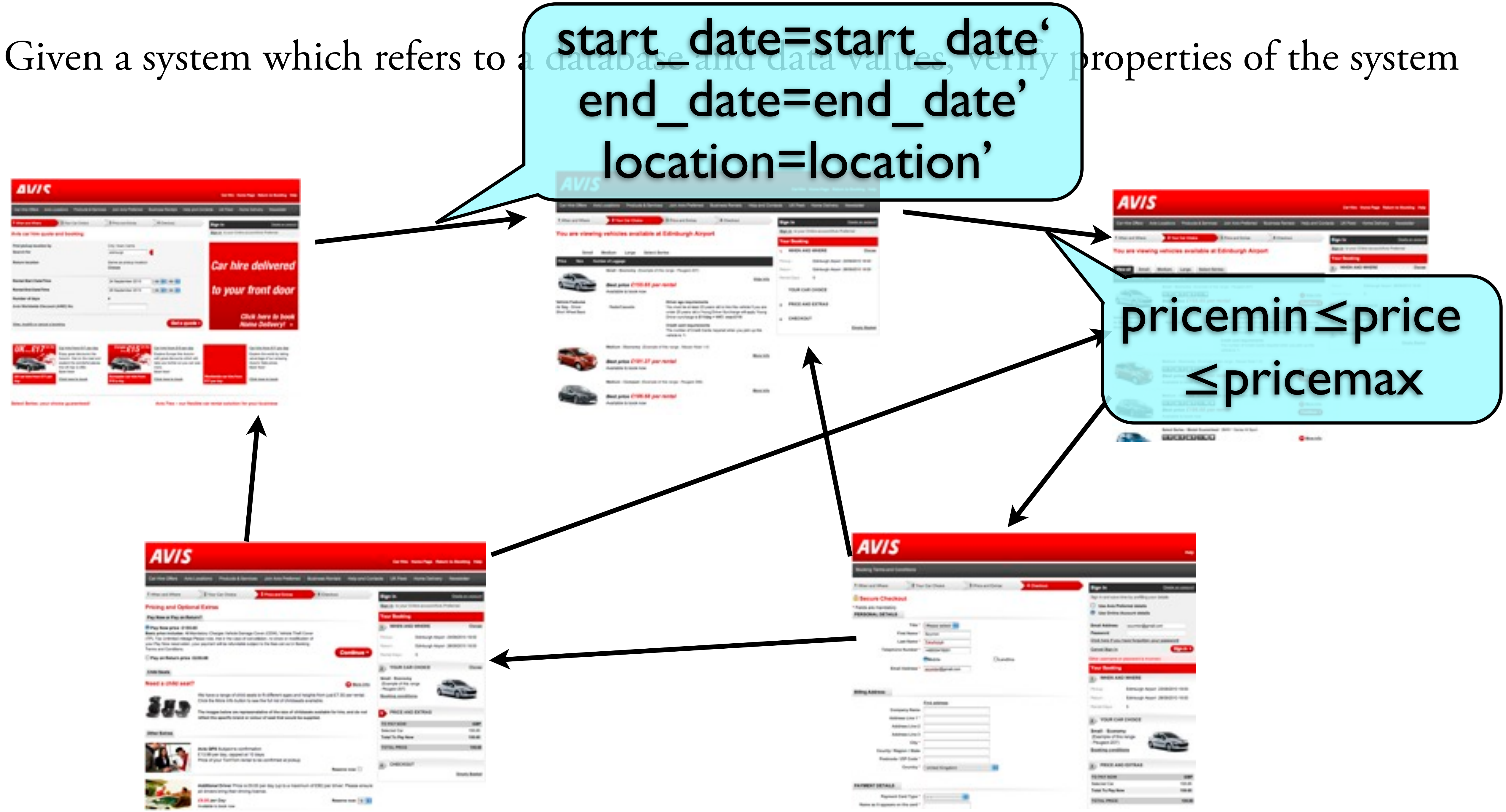
Given a system which refers to a database and data values, verify properties of the system

start_date=start_date‘
end_date=end_date’
location=location’



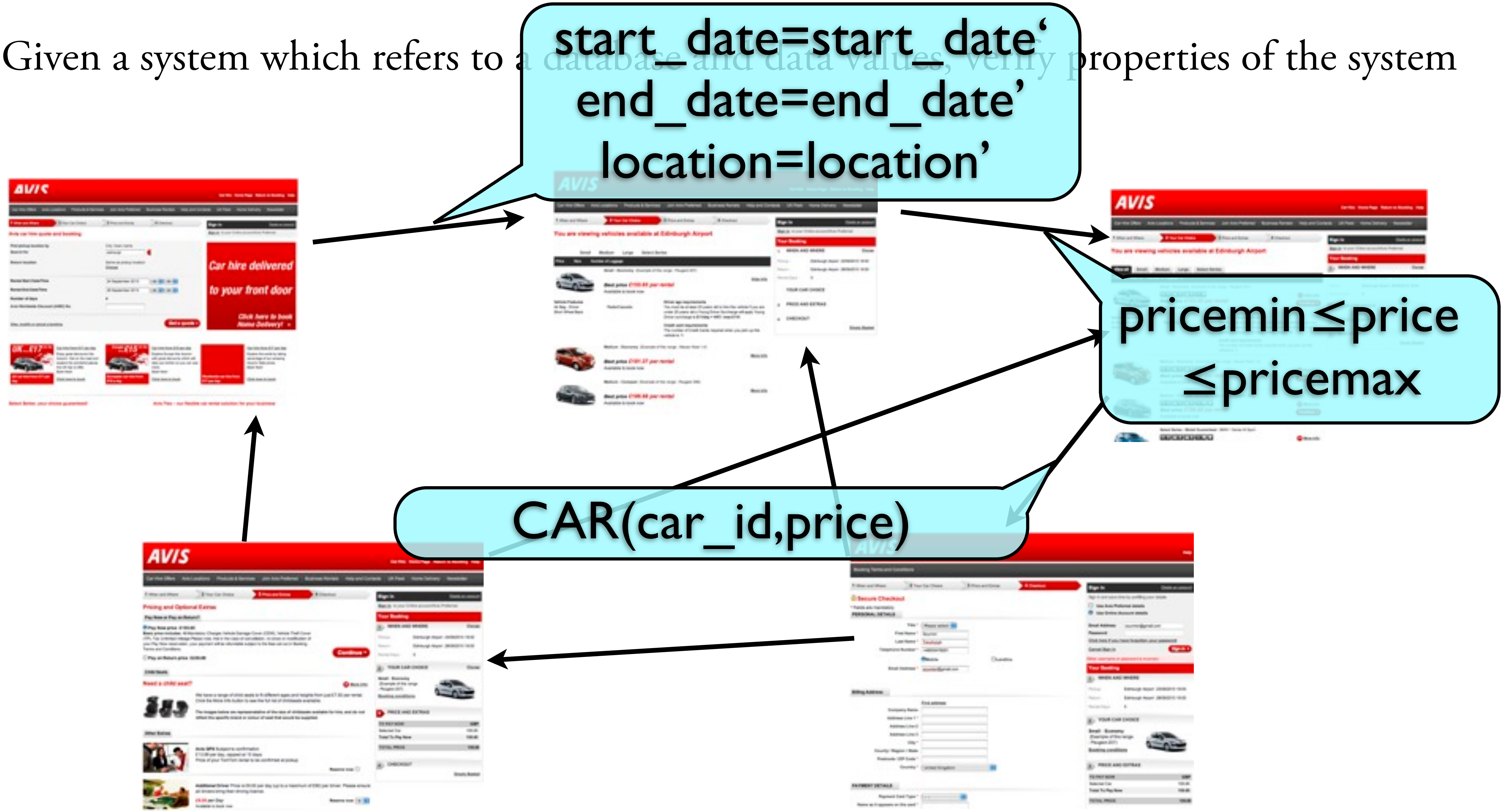
Motivation

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



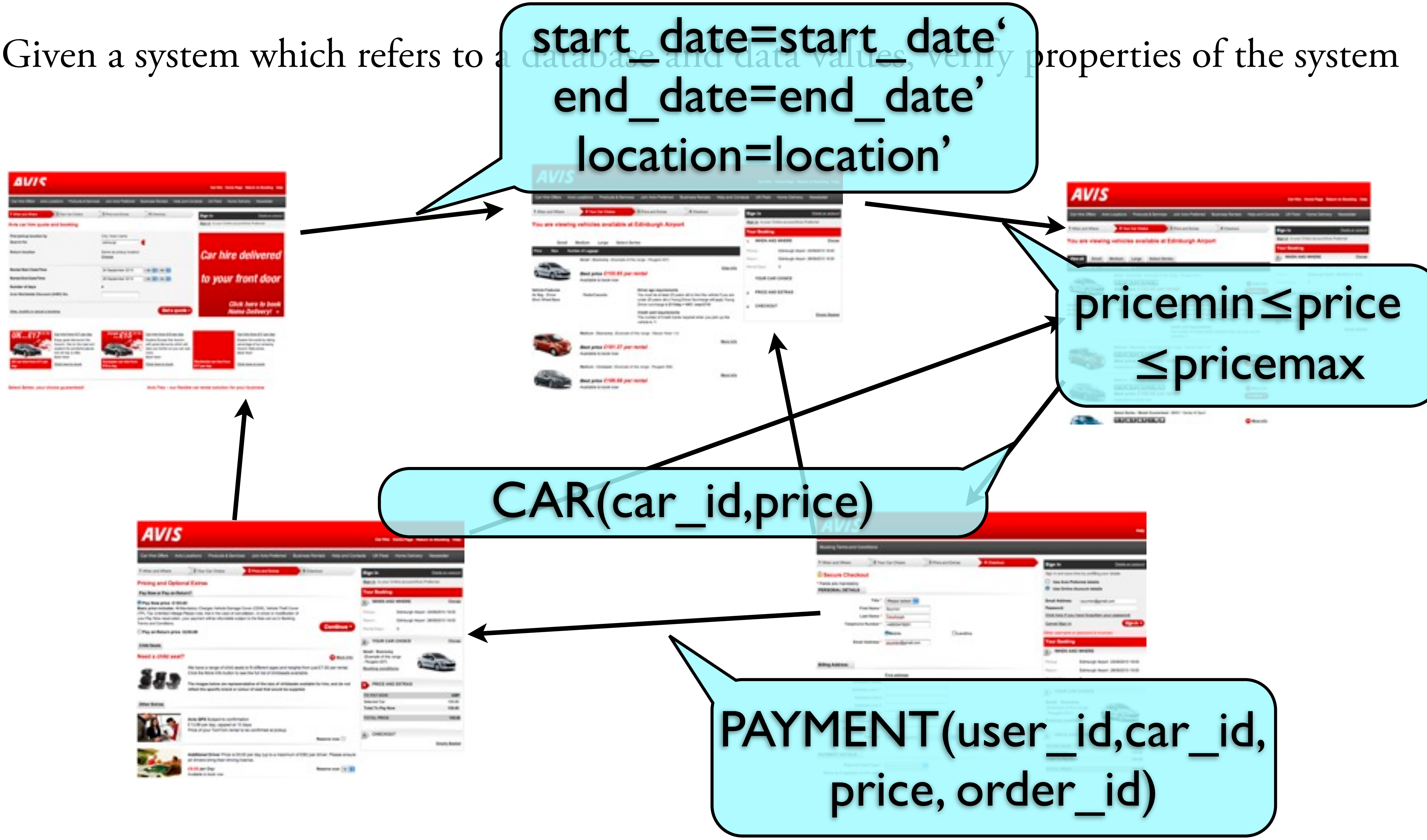
Motivation

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



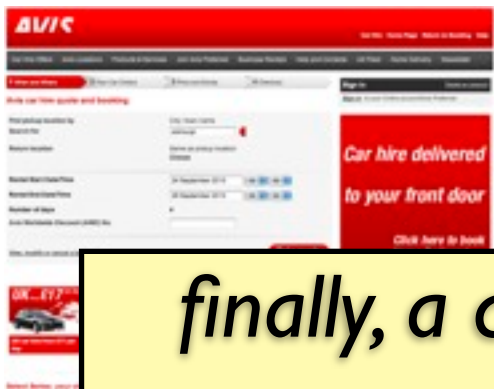
Motivation

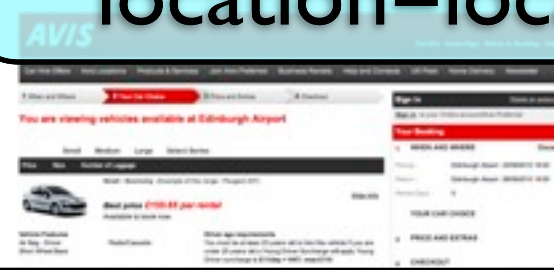
Given a system which refers to a database and data values, verify properties of the system

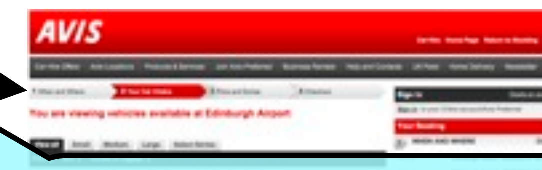


Motivation

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



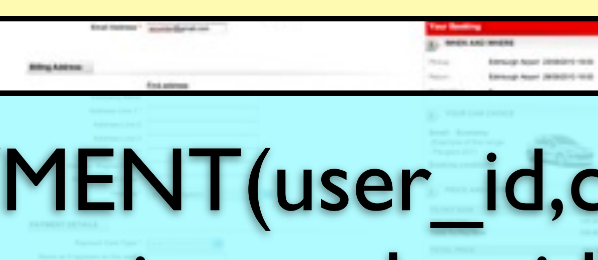




start_date=start_date'
end_date=end_date'
location=location'

pricemin≤price
x

finally, a car is rented and

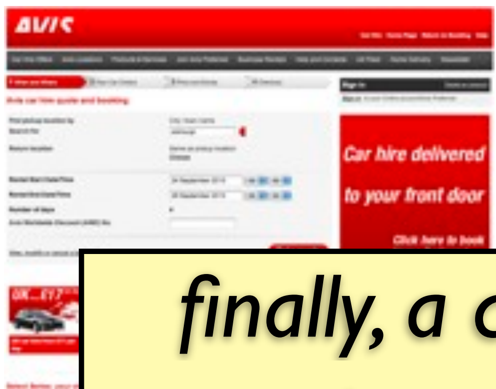


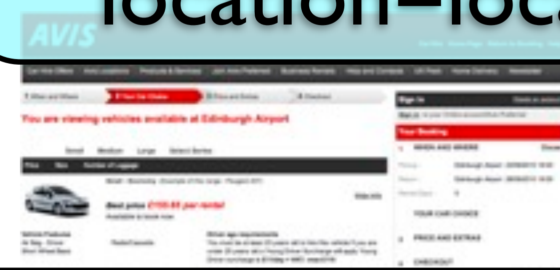
PAYMENT(user_id,car_id, price, order_id)

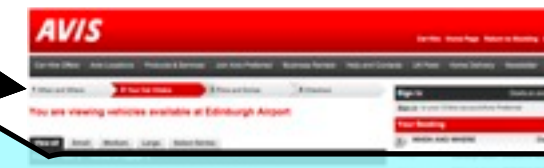
Motivation

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

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



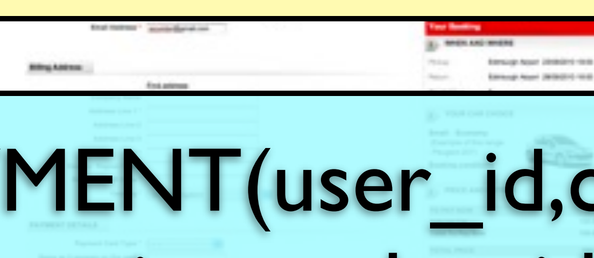




pricemin≤price

finally, a car is rented and

- *no payment was recieved*

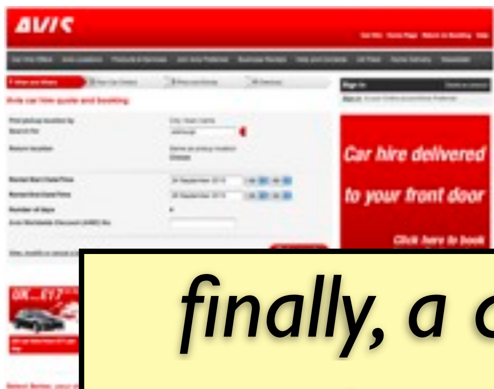


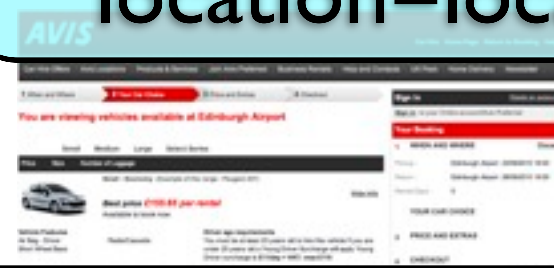
PAYMENT(user_id,car_id,
price, order_id)

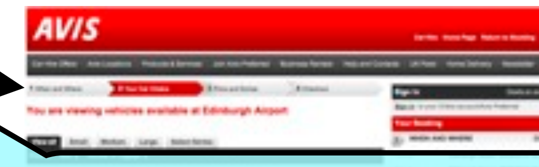
Motivation

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

start_date=start_date‘
end_date=end_date’
location=location’





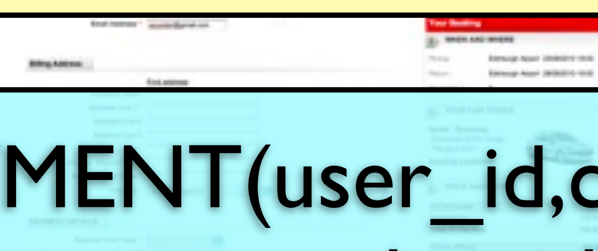


pricemin≤price

x

finally, a car is rented and

- *no payment was recieved*
- *a payment was recieved but* payed=0

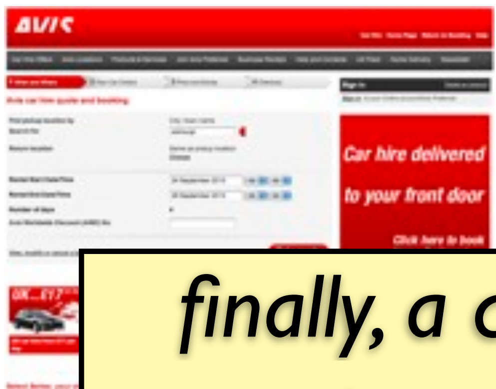


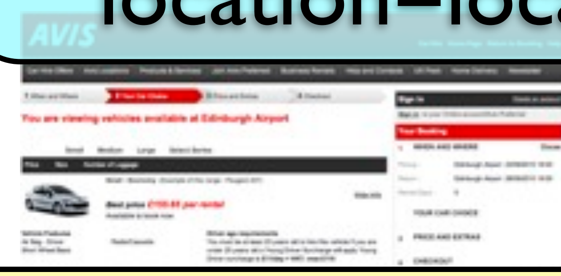
PAYMENT(user_id,car_id,
price, order_id)

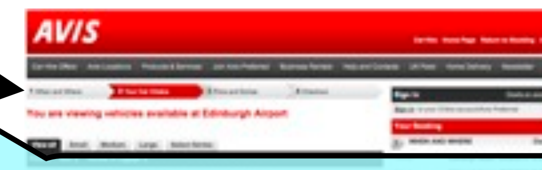
Motivation

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

start_date=start_date‘
end_date=end_date’
location=location’





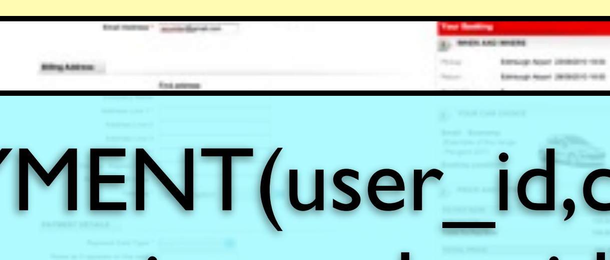


pricemin≤price

x

finally, a car is rented and

- *no payment was recieved*
- *a payment was recieved but* payed=0
- *there was a payment, but* payed<100 & end_date>2050

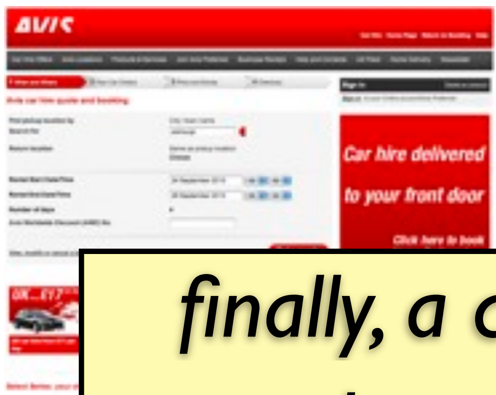


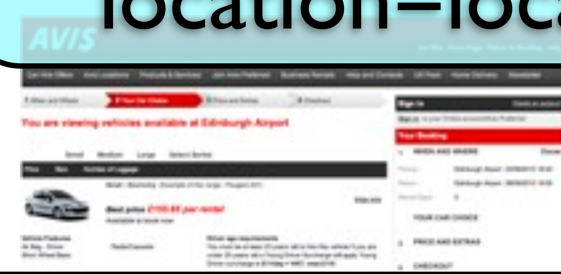
PAYMENT(user_id,car_id, price, order_id)

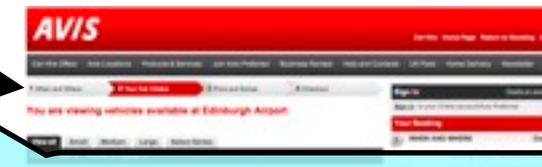
Motivation

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

start_date=start_date‘
end_date=end_date’
location=location’



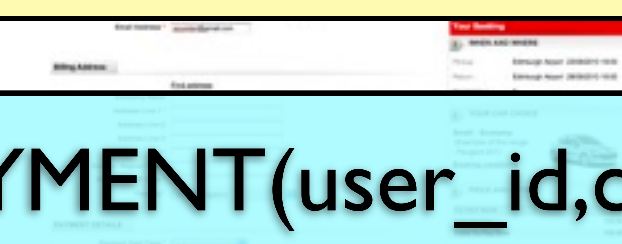




pricemin≤price

finally, a car is rented and

- *no payment was recieved*
- *a payment was recieved but* payed=0
- *there was a payment, but* payed<100 & end_date>2050
- *there was a payment, but* payed<car_price



PAYMENT(user_id,car_id,
price, order_id)

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

$D^X = \{ v: \; X \to D \}$ – 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*: $D = \mathbb{R}$

$$(q, v) \xrightarrow[c_2' = 0]{c_1 < 2} (q', v')$$

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

$$(q, v) \xrightarrow[c_1' = c_1 \cdot b]{first_a\,(c_1)} (q', v')$$

Our setting
(without the database)

The domain:

$$\mathscr{D} = \langle\ D, < ,\ P_1,\ P_2,\ P_3, \ldots, P_l\ \rangle$$

linearly ordered set

unary predicates
(subsets of D)

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:

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

are specified by quantifier free formulas over $\mathscr{D}$

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

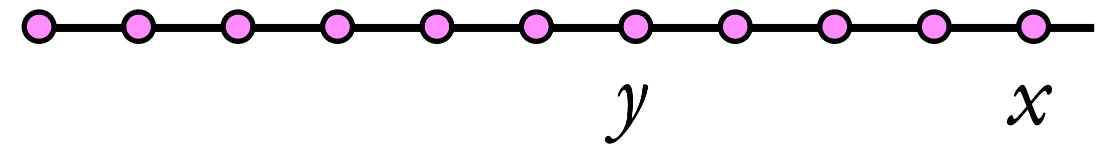
no states

$$\mathcal{A} \left\langle \begin{array}{ll} \tau_{\mathrm{I}}: & (x>0) \ \wedge \ (y=0) \\ \delta_a: & (x'=x) \ \wedge \ (y<y'<x) \\ \delta_b: & (x'=x) \ \wedge \ (y'=0) \\ \tau_{\mathrm{F}}: & (y=1) \end{array} \right.$$

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_{I}: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

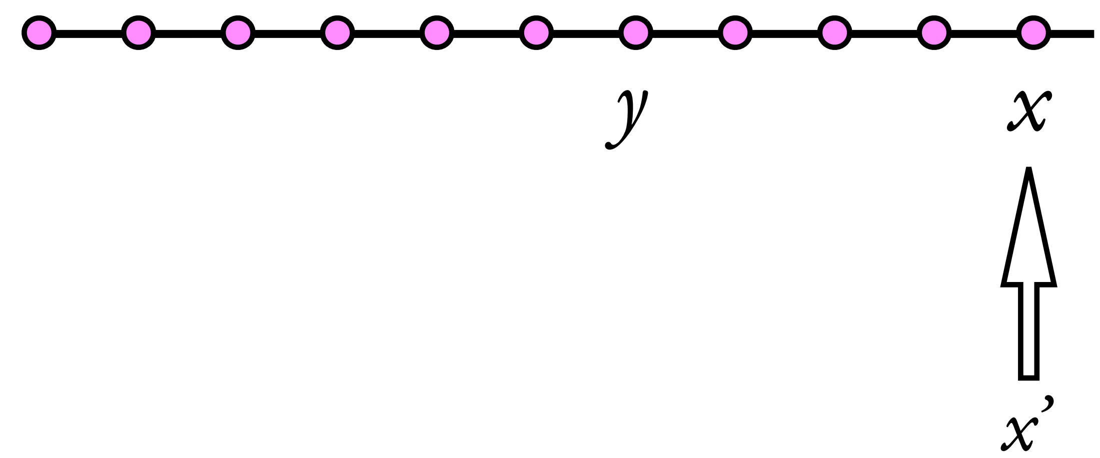
δ_b: $(x'=x) \wedge (y'=0)$

τ_{F}: $(y=1)$

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_{I}: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

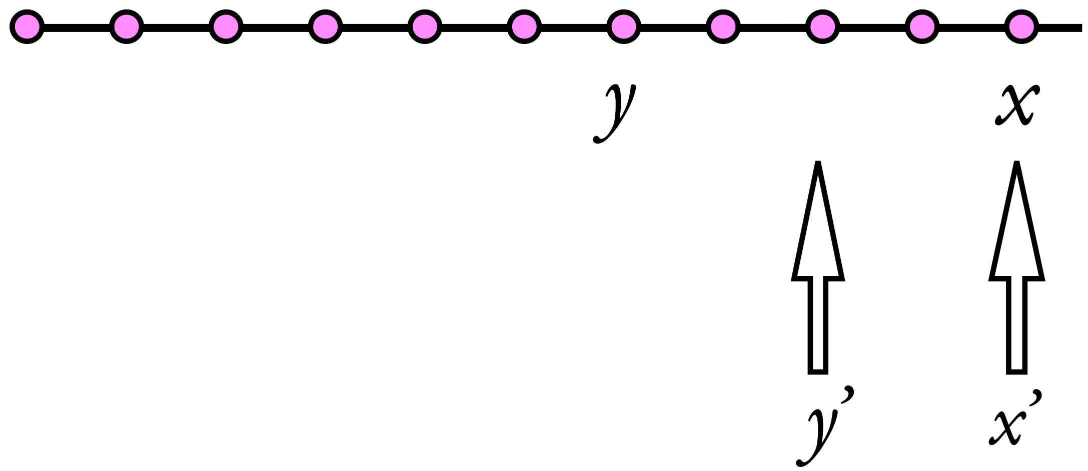
δ_b: $(x'=x) \wedge (y'=0)$

τ_{F}: $(y=1)$

Example D-automaton $\mathscr{A}$

x,y – variables of $\mathscr{A}$

no states



$\mathscr{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

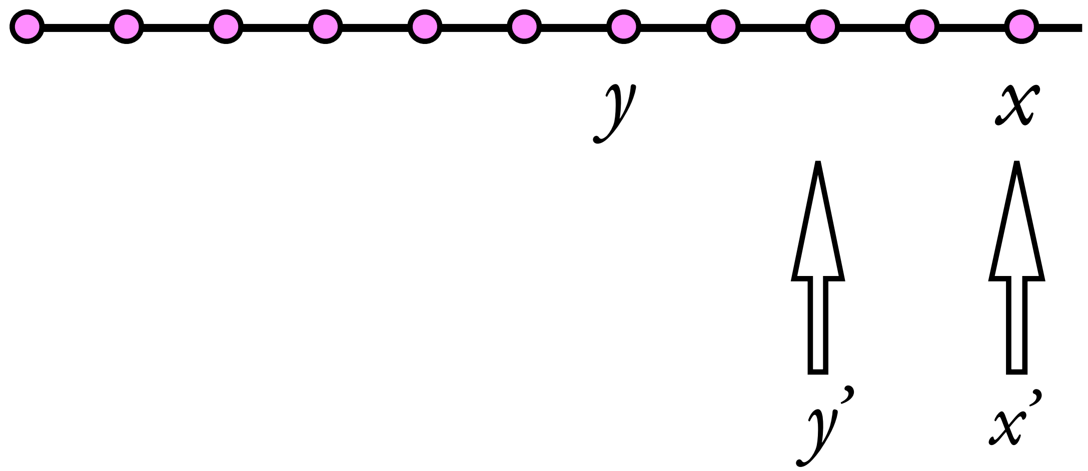
δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

Example D-automaton $\mathcal{A}$

x, y – variables of $\mathcal{A}$

no states



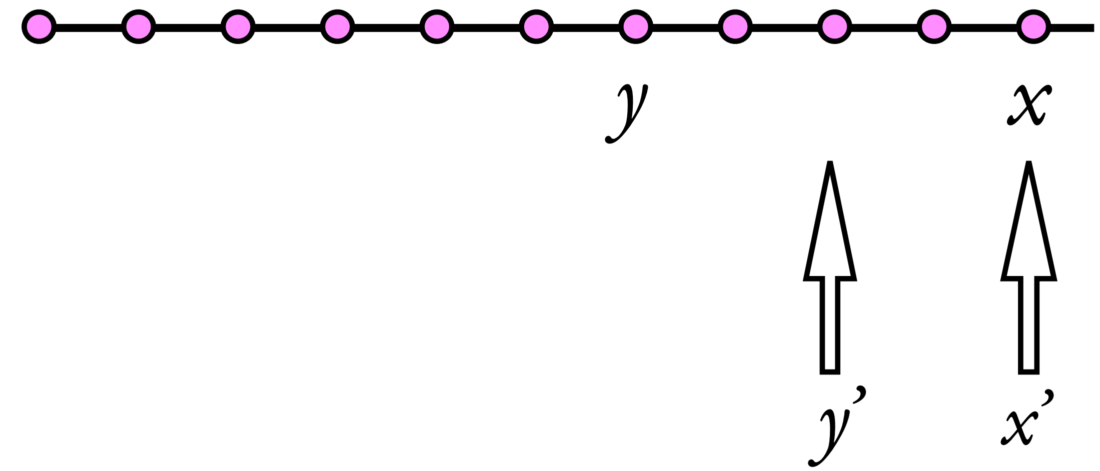
$$\mathcal{A} \begin{cases} \tau_I: & (x>0) \wedge (y=0) \\ \delta_a: & (x'=x) \wedge (y<y'<x) \\ \delta_b: & (x'=x) \wedge (y'=0) \\ \tau_F: & (y=1) \end{cases}$$

a finite run over $\mathbb{Q}$:

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

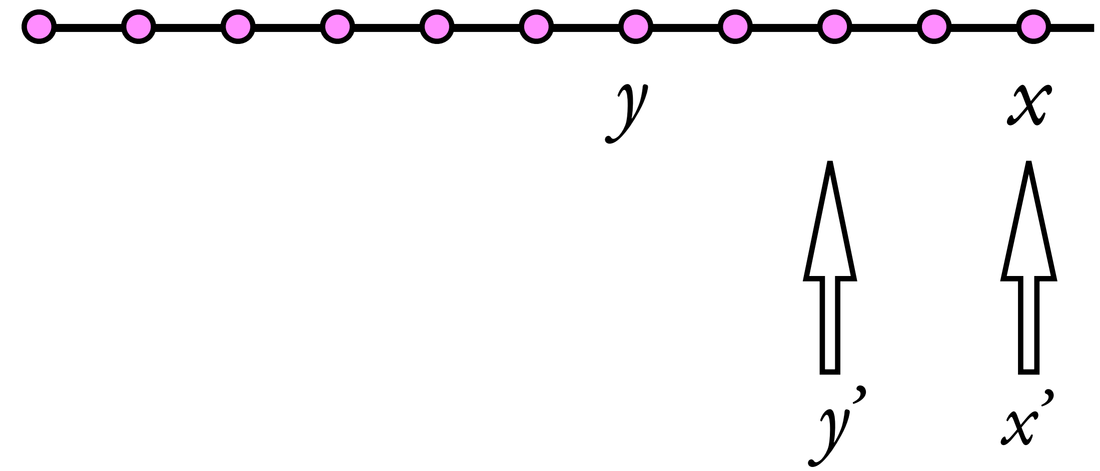
a finite run over $\mathbb{Q}$:

$a\ a\ a\ b\ a\ b\ a\ a\ a\ a\ a$

Example D-automaton $\mathcal{A}$

x, y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

a finite run over $\mathbb{Q}$:

a a a b a b a a a a a

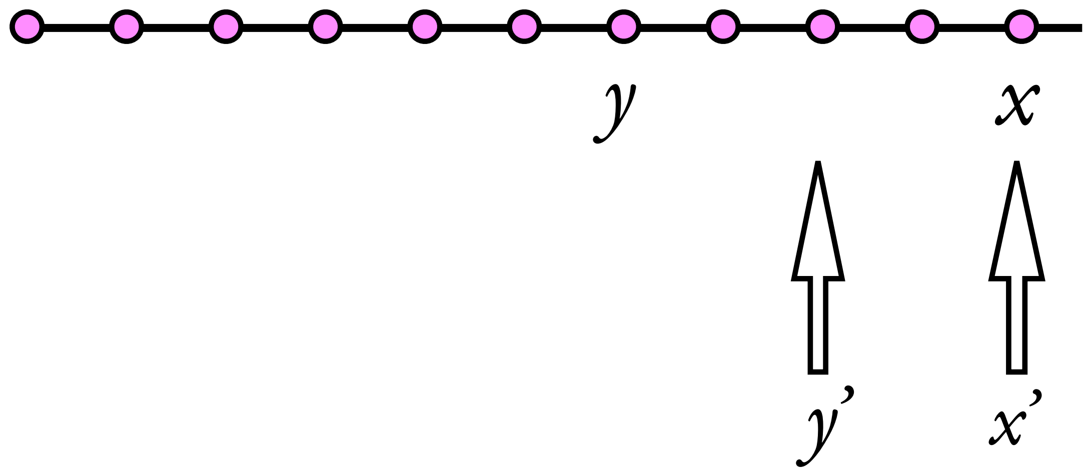
x

y

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

a finite run over $\mathbb{Q}$:

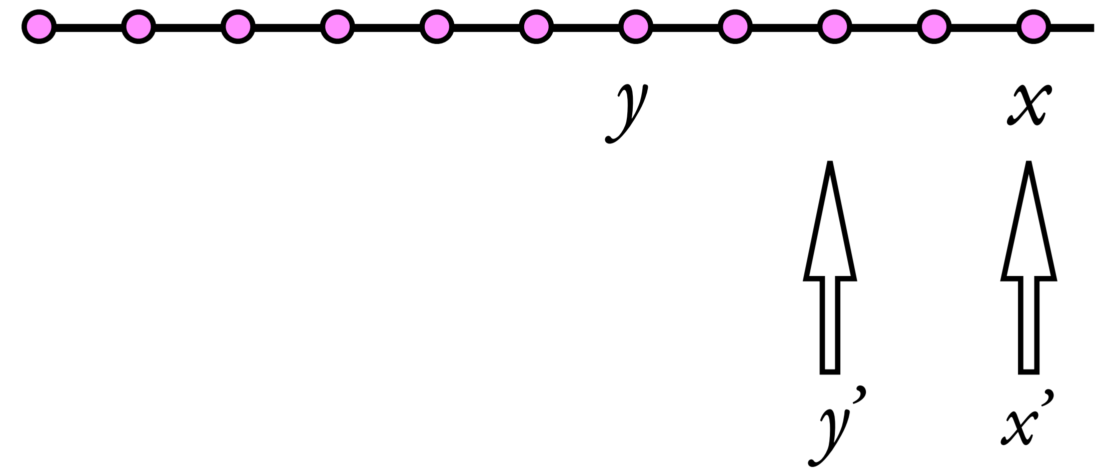
a a a b a b a a a a a

x	5
y	0

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_{I}: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_{F}: $(y=1)$

a finite run over $\mathbb{Q}$:

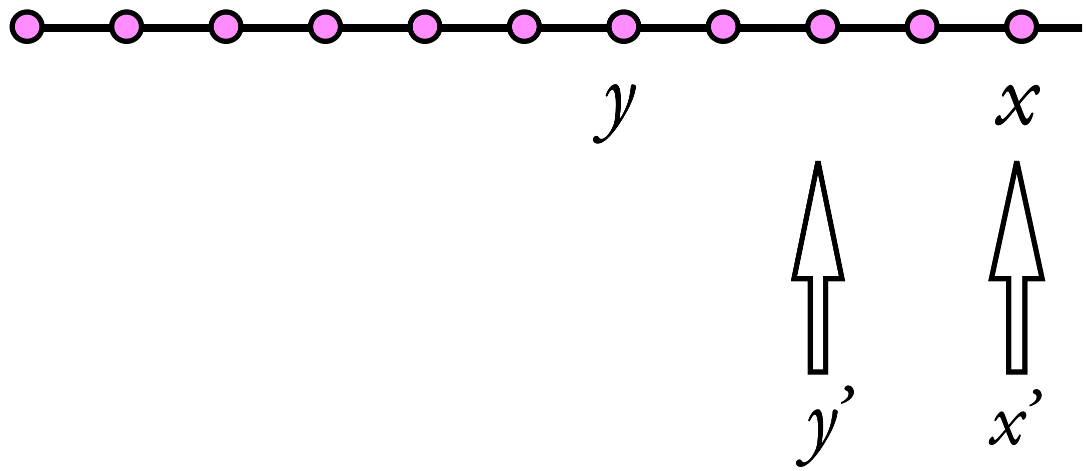
a a a b a b a a a a a

x	5	5
y	0	1

Example D-automaton $\mathscr{A}$

x, y – variables of $\mathscr{A}$

no states



$\mathscr{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

a finite run over $\mathbb{Q}$:

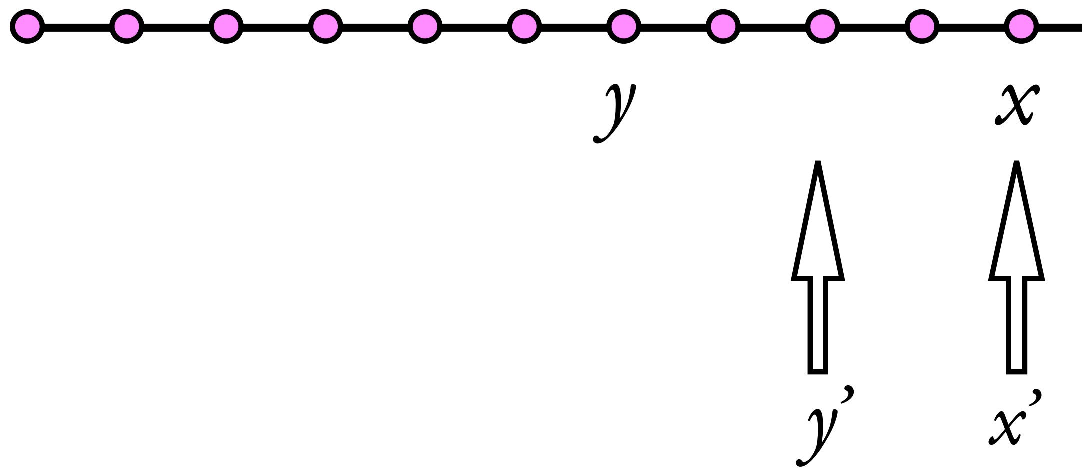
a a a b a b a a a a a

x	5	5	5
y	0	1	2

Example D-automaton $\mathcal{A}$

x, y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

a finite run over $\mathbb{Q}$:

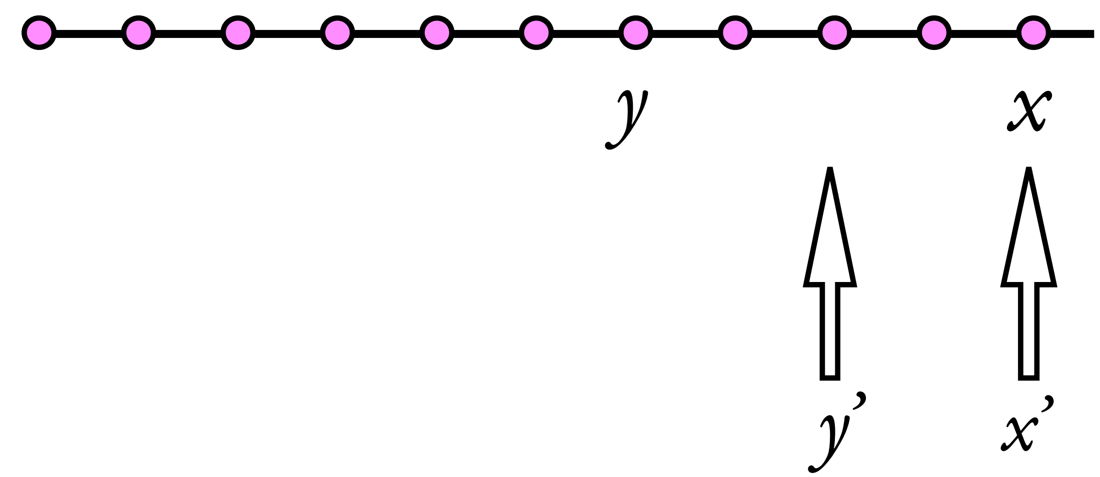
a a a b a b a a a a a

x	5	5	5	5
y	0	1	2	4

Example D-automaton $\mathcal{A}$

x, y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_I: $(x>0) \;\wedge\; (y=0)$

δ_a: $(x'=x) \;\wedge\; (y<y'<x)$

δ_b: $(x'=x) \;\wedge\; (y'=0)$

τ_F: $(y=1)$

a finite run over $\mathbb{Q}$:

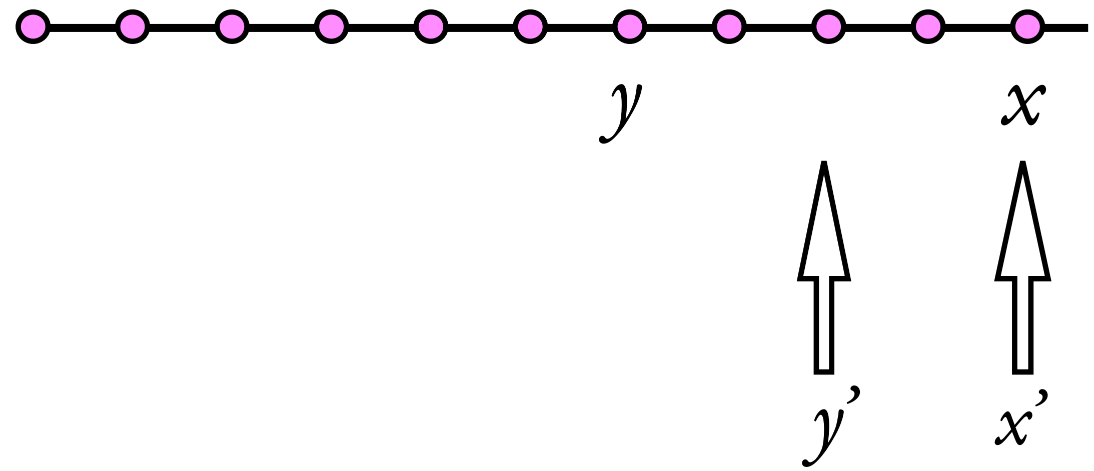
a a a b a b a a a a a

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

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

a finite run over $\mathbb{Q}$:

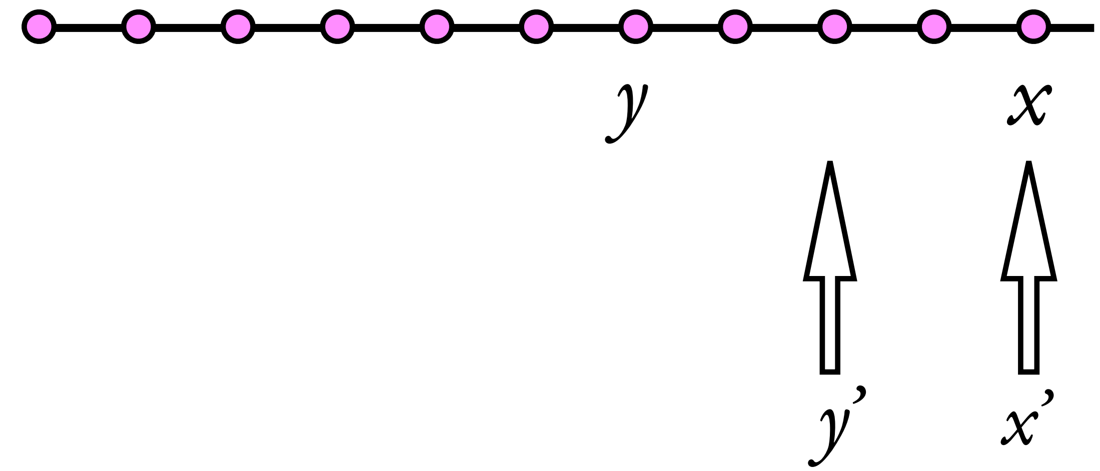
a a a b a b a a a a a

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

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

a finite run over $\mathbb{Q}$:

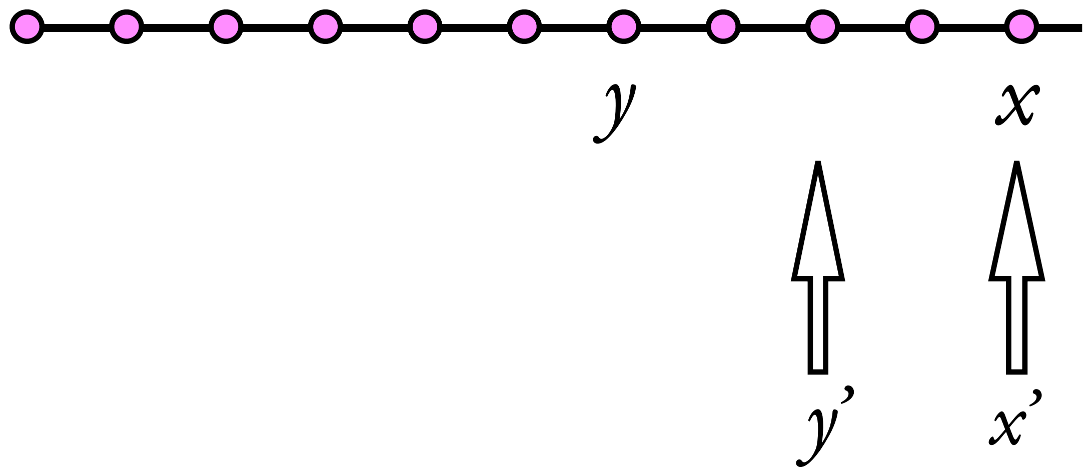
a a a b a b a a a a a

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

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$$
\mathcal{A} \left\langle
\begin{array}{ll}
\tau_I: & (x>0) \ \wedge \ (y=0) \\
\delta_a: & (x'=x) \ \wedge \ (y<y'<x) \\
\delta_b: & (x'=x) \ \wedge \ (y'=0) \\
\tau_F: & (y=1)
\end{array}
\right.
$$

a finite run over $\mathbb{Q}$:

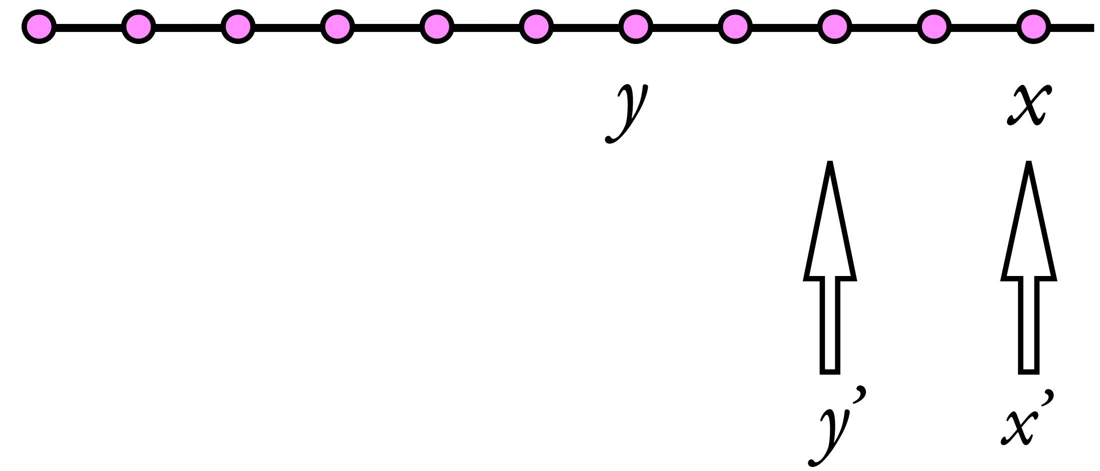
a a a b a b a a a a a

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

Example D-automaton $\mathscr{A}$

x,y – variables of $\mathscr{A}$

no states



$\mathscr{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

a finite run over $\mathbb{Q}$:

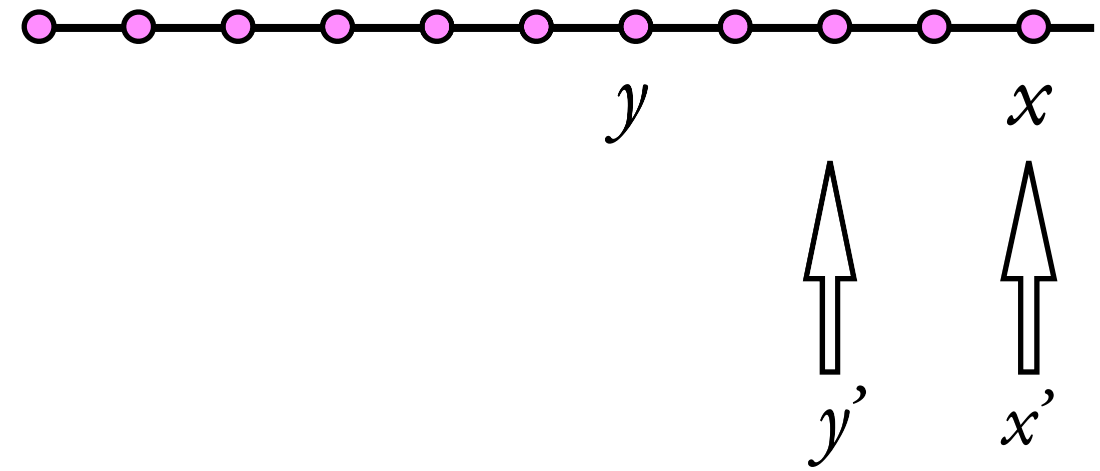
a a a b a b a a a a a

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

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

a finite run over $\mathbb{Q}$:

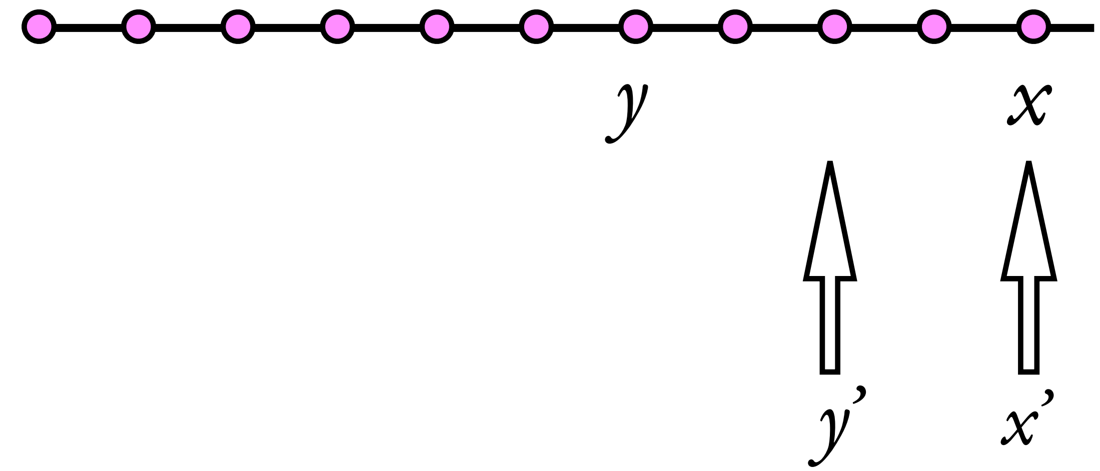
a a a b a b a a a a a

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 $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

a finite run over $\mathbb{Q}$:

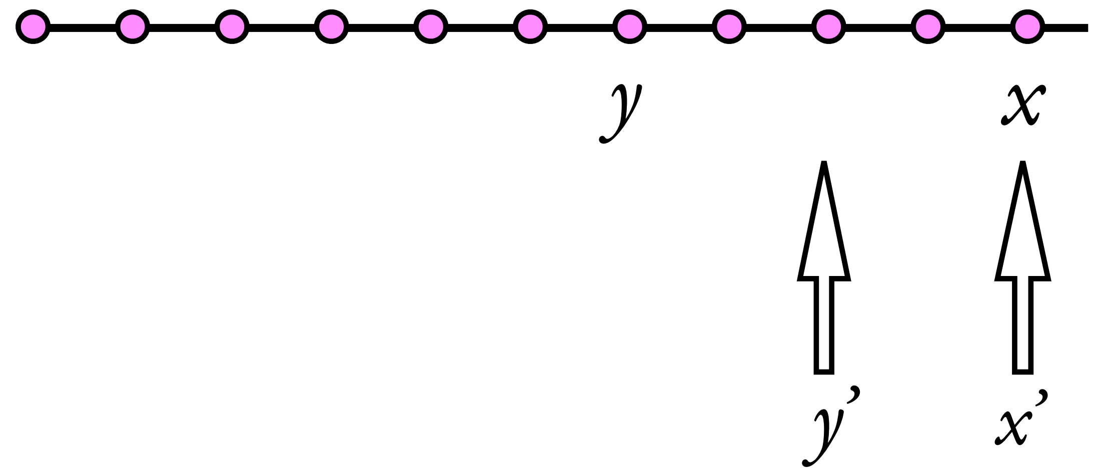
a a a b a b a a a a a

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

Example D-automaton $\mathscr{A}$

x,y – variables of $\mathscr{A}$

no states



$$\mathscr{A} \quad \begin{cases} \tau_I: & (x>0) \ \wedge \ (y=0) \\ \delta_a: & (x'=x) \ \wedge \ (y<y'<x) \\ \delta_b: & (x'=x) \ \wedge \ (y'=0) \\ \tau_F: & (y=1) \end{cases}$$

a finite run over $\mathbb{Q}$:

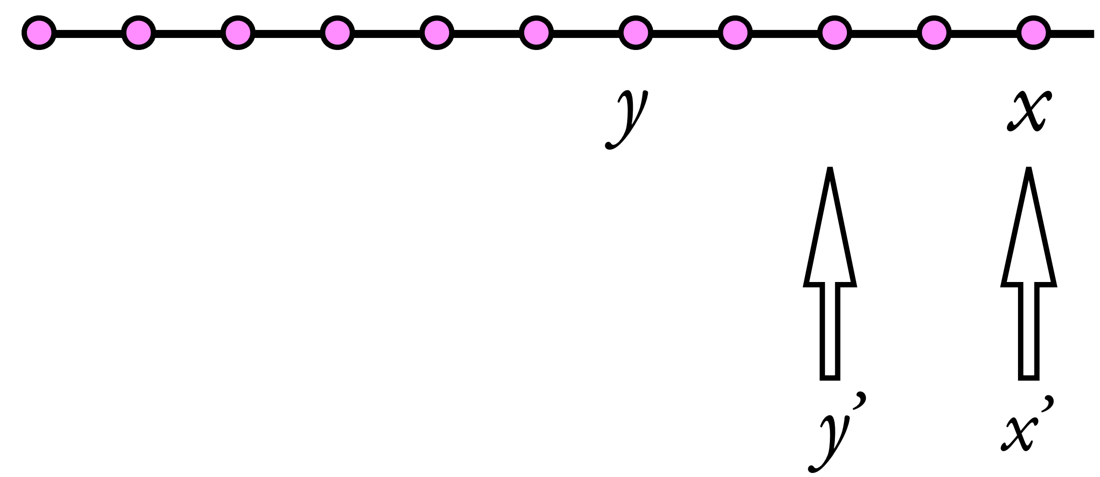
a a a b a b a a a a a

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

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

$\boxed{\tau_F\text{: } (y=1)}$

a finite run over $\mathbb{Q}$:

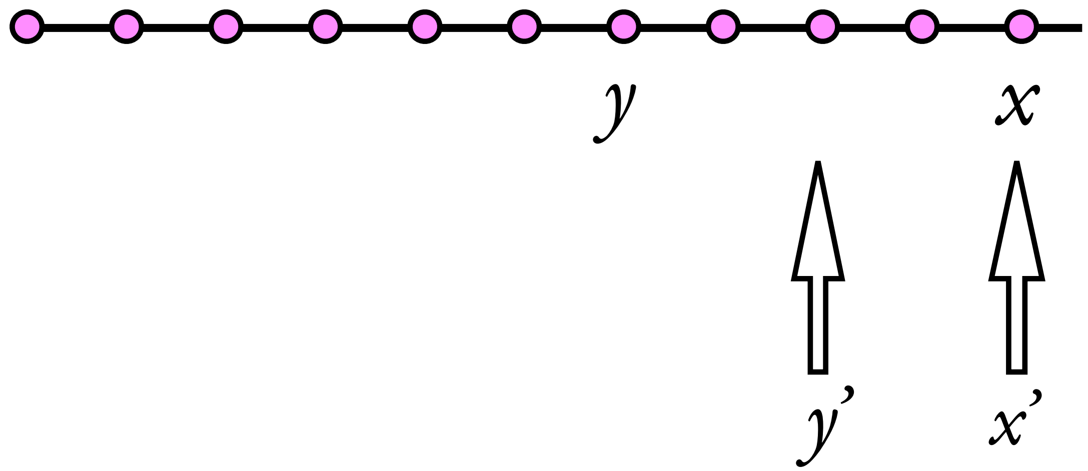
a a a b a b a a a a a

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

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

a finite run over $\mathbb{Q}$:

a a a b a b a a a a a

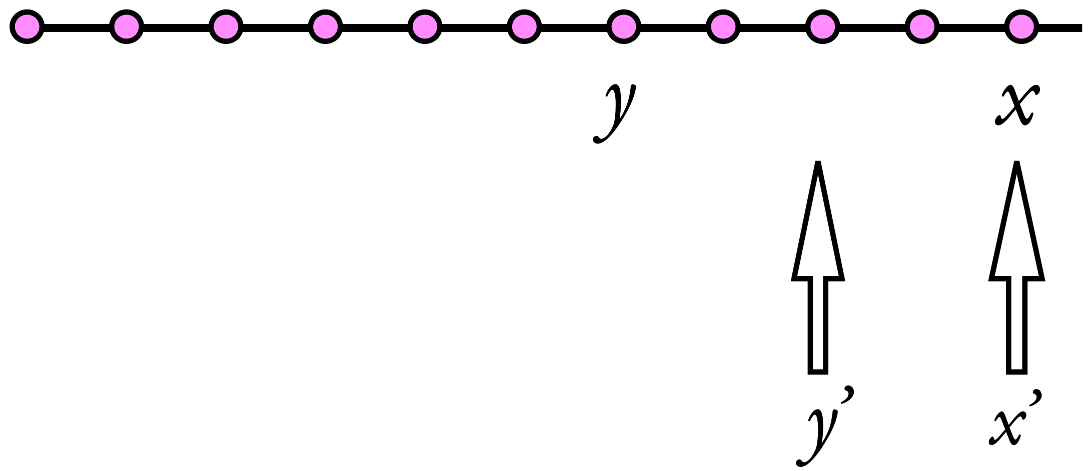
x	5	5	5	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{4}{5}$	1

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

Example D-automaton $\mathcal{A}$

x, y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

$\boxed{\tau_F: (y=1)}$

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

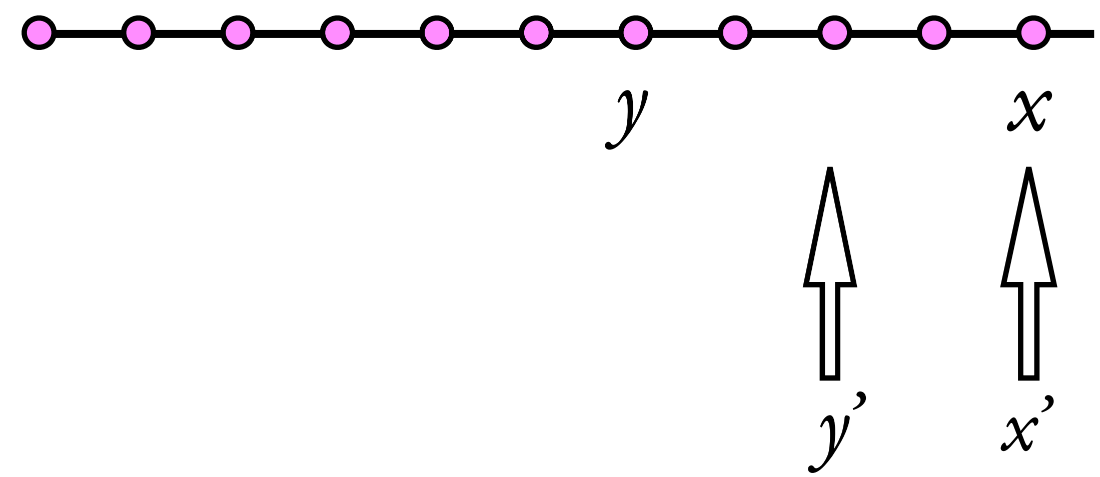
accepted language: $(a+b)^*a$

a finite run over $\mathbb{N}$:

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$$\mathcal{A} \begin{cases} \tau_I: & (x>0) \wedge (y=0) \\ \delta_a: & (x'=x) \wedge (y<y'<x) \\ \delta_b: & (x'=x) \wedge (y'=0) \\ \tau_F: & \boxed{(y=1)} \end{cases}$$

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

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

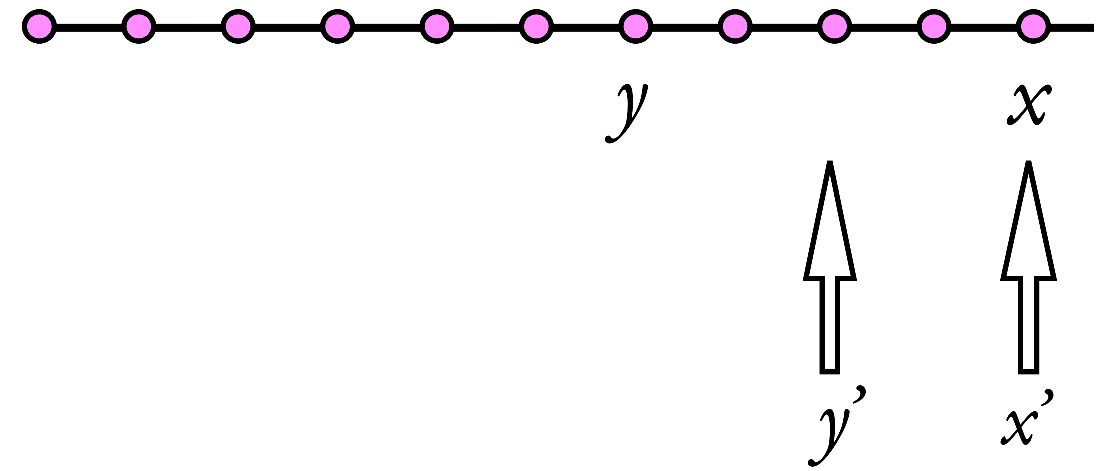
a finite run over $\mathbb{N}$:

$a\ a\ a\ b\ a\ b\ a\ a\ a\ a$

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$$\mathcal{A} \left\langle \begin{array}{ll} \tau_I: & (x>0) \wedge (y=0) \\ \delta_a: & (x'=x) \wedge (y<y'<x) \\ \delta_b: & (x'=x) \wedge (y'=0) \\ \tau_F: & (y=1) \end{array} \right.$$

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

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

a finite run over $\mathbb{N}$:

$a\ a\ a\ b\ a\ b\ a\ a\ a\ a$

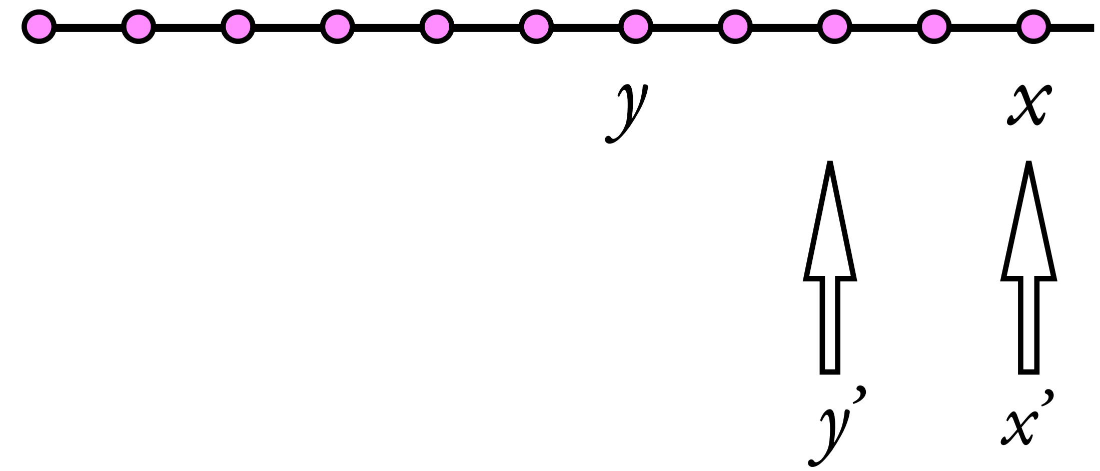
x

y

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$$\mathcal{A} \begin{cases} \tau_I: & (x>0) \wedge (y=0) \\ \delta_a: & (x'=x) \wedge (y<y'<x) \\ \delta_b: & (x'=x) \wedge (y'=0) \\ \boxed{\tau_F: \quad (y=1)} \end{cases}$$

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

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

a finite run over $\mathbb{N}$:

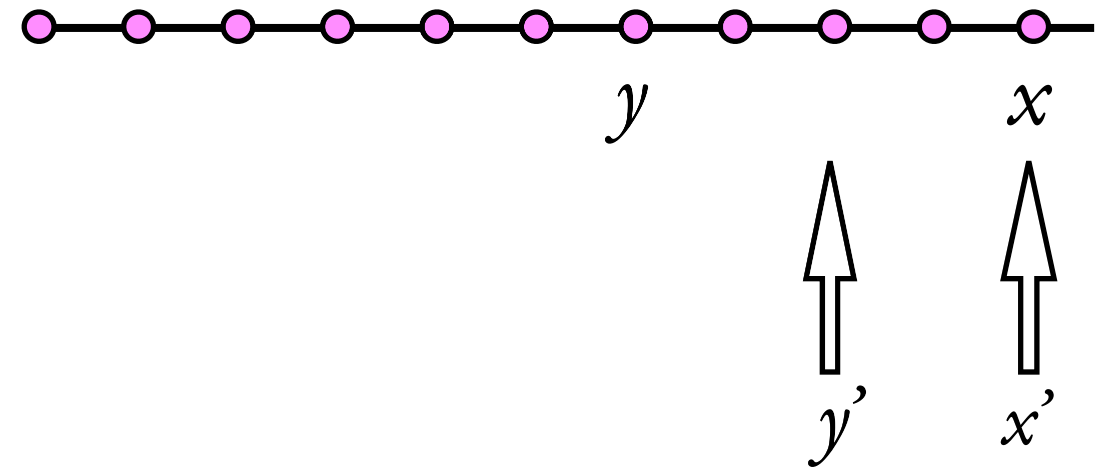
a a a b a b a a a a

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

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$\mathcal{A}$

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

$\boxed{\tau_F\text{: } (y=1)}$

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

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

a finite run over $\mathbb{N}$:

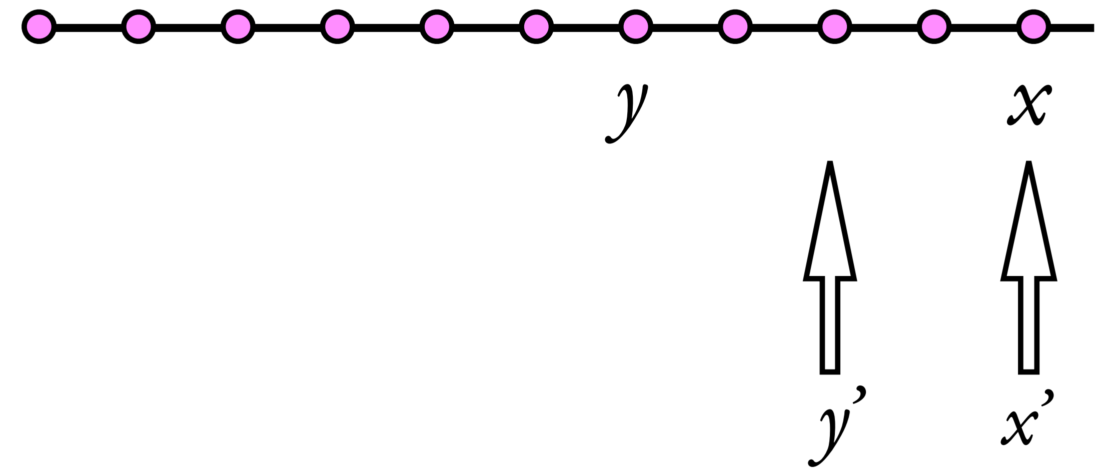
		a	a	a	b	a	b	a	a	a	a
x	5	5	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	1	2	3	4

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

Example D-automaton $\mathcal{A}$

x, y – variables of $\mathcal{A}$

no states



$$\mathcal{A} \left\langle \begin{array}{ll} \tau_I: & (x>0) \wedge (y=0) \\ \delta_a: & (x'=x) \wedge (y<y'<x) \\ \delta_b: & (x'=x) \wedge (y'=0) \\ \tau_F: & (y=1) \end{array} \right.$$

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

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

a finite run over $\mathbb{N}$:

$a\ a\ a\ b\ a\ b\ a\ a\ a\ a$

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

an infinite run over $\mathbb{N}$:

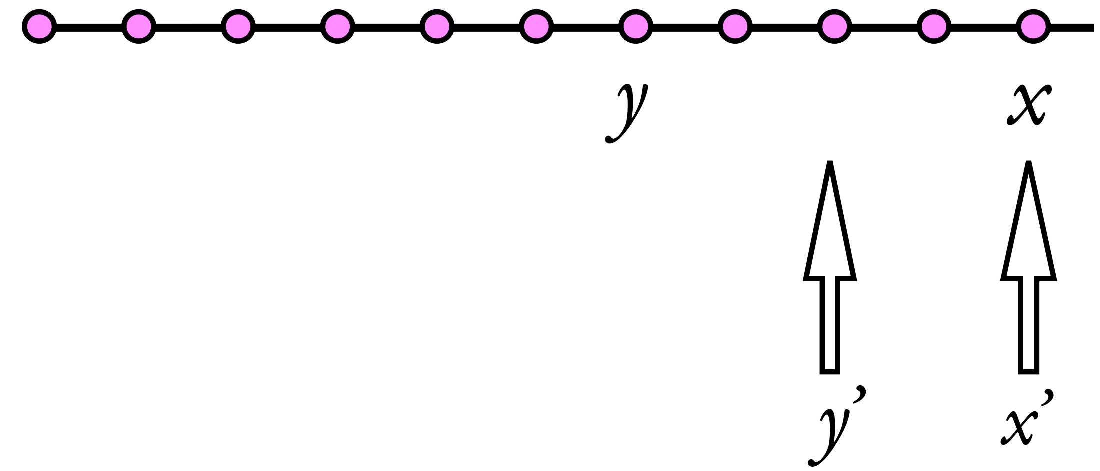
$a\ a\ \ a\ b\ a\ b\ a\ \ a\ a\ a$

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

Example D-automaton $\mathcal{A}$

x,y – variables of $\mathcal{A}$

no states



$$\mathcal{A} \begin{cases} \tau_I: & (x>0) \wedge (y=0) \\ \delta_a: & (x'=x) \wedge (y<y'<x) \\ \delta_b: & (x'=x) \wedge (y'=0) \\ \tau_F: & \boxed{(y=1)} \end{cases}$$

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

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

a finite run over $\mathbb{N}$:

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

an infinite run over $\mathbb{N}$:

a a a b a b 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: $(a^{\mathrm{B}}b)^\omega$

Adding database constraints

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

τ_{I}: $(x=0)$

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

δ_b: $(x'=0)$

τ_{F}: $(x=1)$

Adding database constraints

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

τ_I: $(x=0)$

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

δ_b: $(x'=0)$

τ_F: $(x=1)$

Adding database constraints

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

τ_{I}: $(x=0)$

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

δ_b: $(x'=0)$

τ_{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

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

τ_{I}: $(x=0)$

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

δ_b: $(x'=0)$

τ_{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

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

τ_{I}: $(x=0)$

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

δ_b: $(x'=0)$

τ_{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

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

τ_I: $(x=0)$

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

δ_b: $(x'=0)$

τ_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 $\mathbb{N}$:

$a\ a\ a\ b\ a\ b\ a$

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

Adding database constraints

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

τ_I: $(x=0)$

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

δ_b: $(x'=0)$

$\boxed{\tau_F: (x=1)}$

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={}

Adding database constraints

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

τ_I: $(x=0)$

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

δ_b: $(x'=0)$

τ_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={}

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

Adding database constraints

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

τ_{I}: $(x=0)$

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

δ_b: $(x'=0)$

$\boxed{\tau_{\mathrm{F}}: \ (x=1)}$

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={}

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

an infinite run over $\mathbb{N}$:

Adding database constraints

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

τ_{I}: $(x=0)$

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

δ_b: $(x'=0)$

$\boxed{\tau_{\mathrm{F}}: \ (x=1)}$

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={}

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

an infinite run over $\mathbb{N}$:

$a\ a\ a\ b\ a\ b\ a\ a\ b\ a\ a\ldots$

Adding database constraints

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

τ_I: $(x=0)$

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

δ_b: $(x'=0)$

τ_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={}

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

an infinite run over $\mathbb{N}$:

	a	a	a	b	a	b	a	a	b	a	$a\ldots$	
x	0	1	3	4	0	3	0	1	4	0	1	3

Adding database constraints

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

τ_I: $(x=0)$

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

δ_b: $(x'=0)$

$\boxed{\tau_F: \ (x=1)}$

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

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

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

an infinite run over $\mathbb{N}$:

$a\ a\ a\ b\ a\ b\ a\ a\ b\ a\ a\ldots$

x	0	1	3	4	0	3	0	1	4	0	1	3

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

Adding database constraints

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

τ_I: $(x=0)$

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

δ_b: $(x'=0)$

τ_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={}

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

an infinite run over $\mathbb{N}$:

$a\ a\ a\ b\ a\ b\ a\ a\ b\ a\ a\ldots$

x 0 1 3 4 0 3 0 1 4 0 1 3

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

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

Adding database constraints

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

τ_I: $(x=0)$

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

δ_b: $(x'=0)$

τ_F: $(x=1)$

Decide emptiness of D-automata over ω-words, with databases

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={}

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

an infinite run over $\mathbb{N}$:

$a\ a\ a\ b\ a\ b\ a\ a\ b\ a\ a\ldots$

x 0 1 3 4 0 3 0 1 4 0 1 3

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

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

Known results
deciding emptiness

	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

	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	?
$\mathscr{D}$,<	PSPACE	PSPACE

Known results

deciding emptiness

	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
$\mathscr{D}$,<	PSPACE	PSPACE

Known results
deciding emptiness

	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
$\mathscr{D},<$	PSPACE	PSPACE

Dense case

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

Dense case

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

τ_I: $\boxed{(x>0) \wedge (y=0)}$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $\boxed{(y=1)}$

Dense case

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

τ_{I}: $\boxed{(x>0) \wedge (y=0)}$

δ_{a}: $(x'=x) \wedge (y<y'<x)$

δ_{b}: $(x'=x) \wedge (y'=0)$

τ_{F}: $\boxed{(y=1)}$

The region construction

$\boxed{0<y<1<x}$, $\boxed{0<x<1=y}$, $\boxed{0<1<y<x}$, $\boxed{0<1=y<x}$, $\boxed{0=y<1<x}$, $\boxed{0=y<1=x}$, …

Dense case

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

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction

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

a

Dense case

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

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction

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

a

⋓

$\exists(x,y)$

Dense case

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

τ_{I}: $\boxed{(x>0) \wedge (y=0)}$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_{F}: $\boxed{(y=1)}$

The region construction

$0<y<1<x$, $\boxed{0<x<1=y}$, $0=1<y<x$, $\boxed{0<1=y<x}$, $\boxed{0=y<1<x}$, $\boxed{0=y<1=x}$, …

$\boxed{0<1=y<x} \xleftarrow{a} \boxed{0=y<1<x}$

$\exists(x',y') \qquad \exists(x,y)$

Dense case

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

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction

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

$$\overset{\rotatebox{90}{$\in$}}{\exists(x',y')} \quad \overset{\rotatebox{90}{$\in$}}{\exists(x,y)}: \quad \delta_a(x, y, x', y')$$

Dense case

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

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction

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

a

$$\overset{\Cup}{\exists(x',y')} \quad \overset{\Cup}{\exists(x,y)}: \quad \delta_a(x, y, x', y')$$

$(5,0)$

Dense case

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

τ_{I}: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_{F}: $(y=1)$

The region construction

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

$$\underset{(5,1)}{\exists\overset{\Cup}{(x',y')}} \quad \underset{(5,0)}{\exists\overset{\Cup}{(x,y)}}: \quad \delta_a(x,\ y,\ x',\ y')$$

Dense case

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

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction

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

Dense case

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

τ_I: $(x>0) \wedge (y=0)$

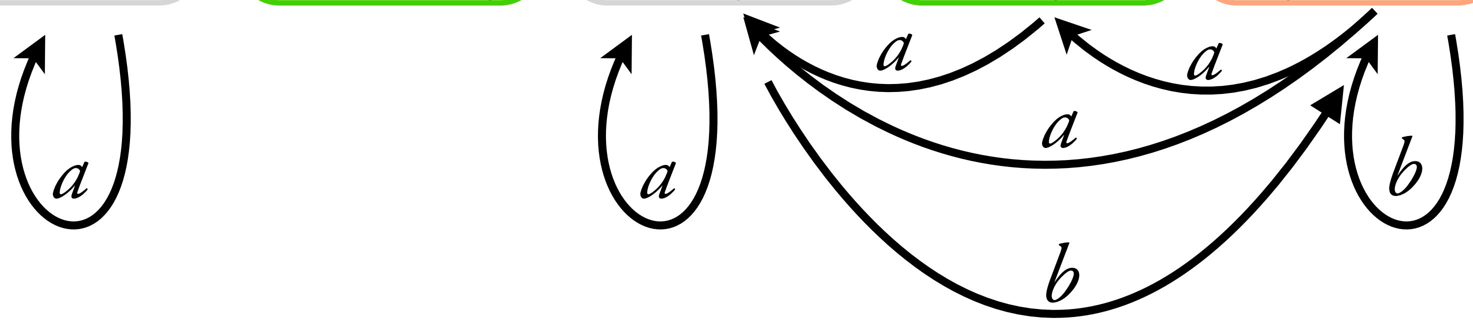
δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction

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



Dense case

The region construction

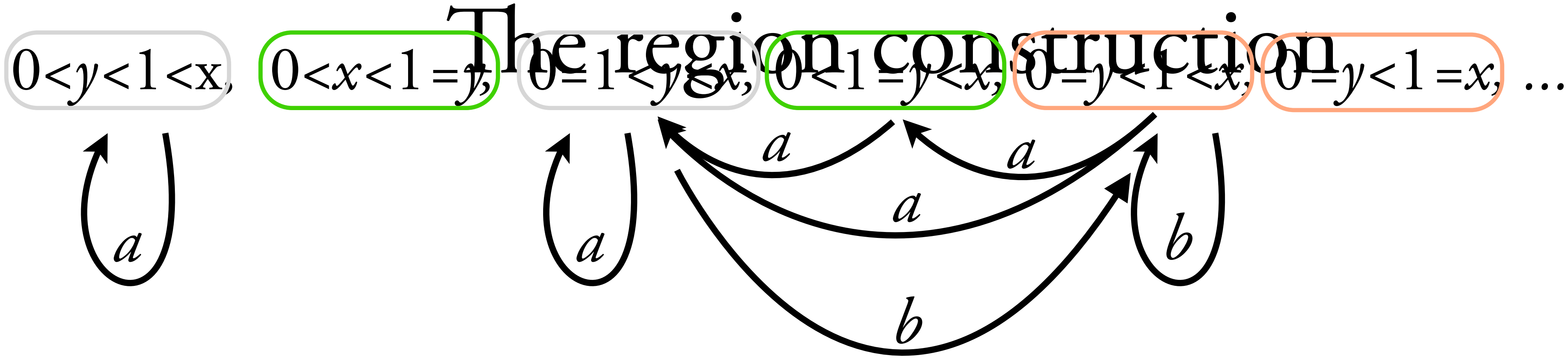
- $D = \mathbb{Q}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



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

Dense case

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

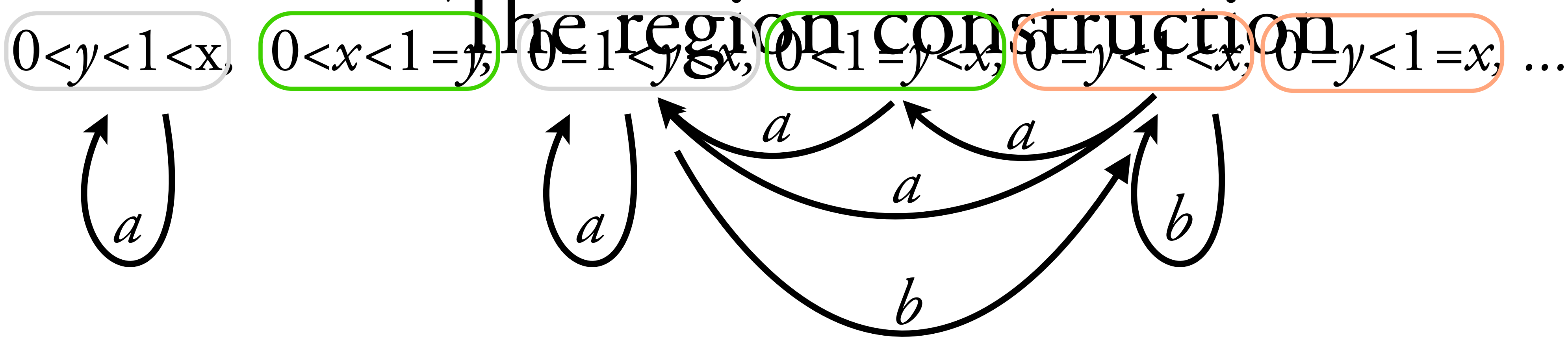
τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction



a a a b a b a a a a a

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

Dense case

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

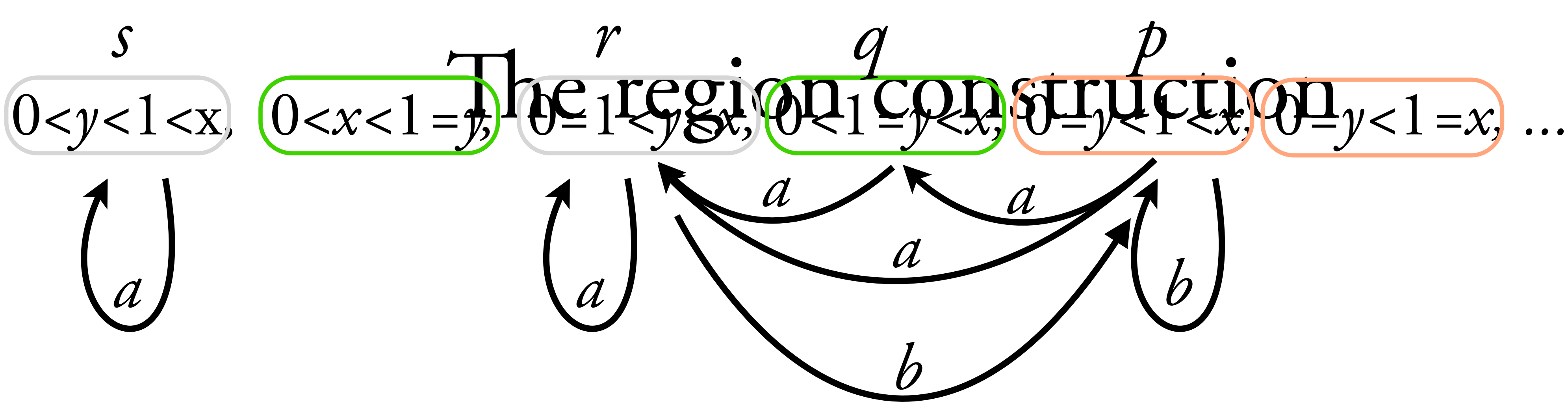
τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction



a a a b a b a a a a a

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

Dense case

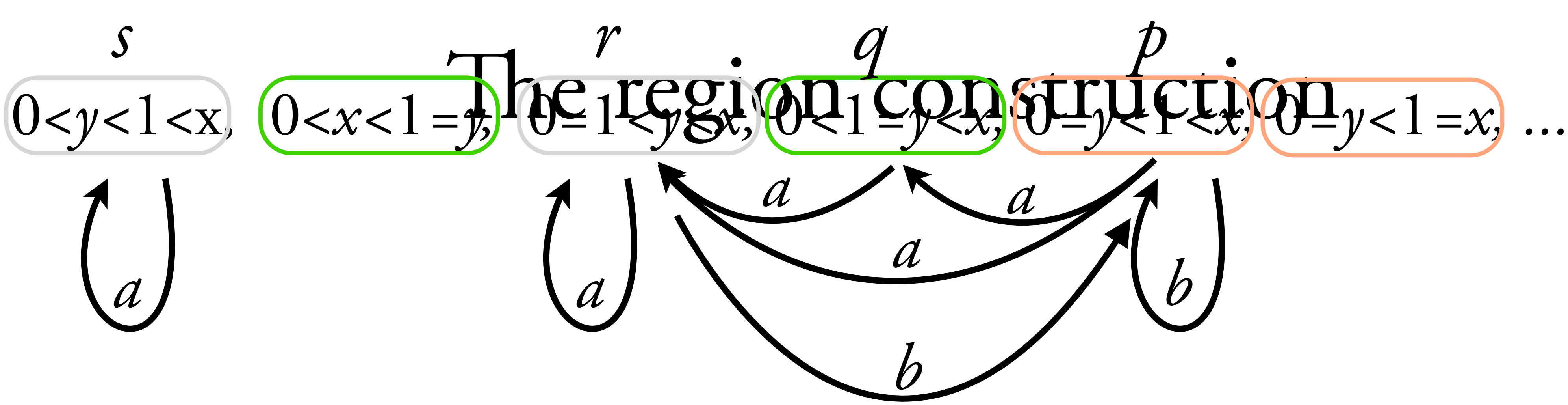
- $D = \mathbb{Q}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



a a a b a b a a a a a

p q r r q r q s s s s q

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

Dense case

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

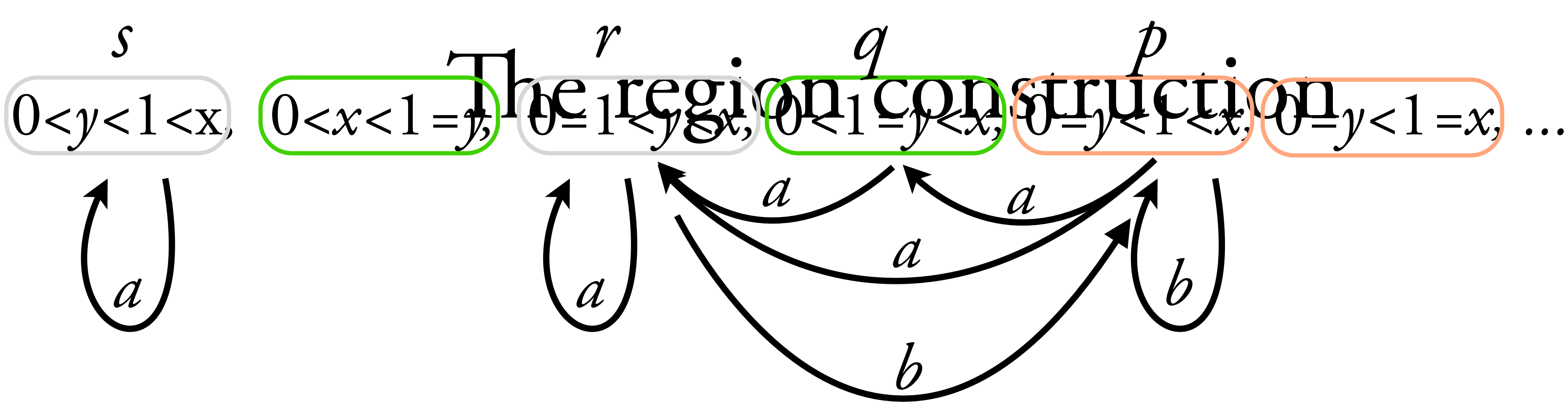
τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction



$a\ a\ a\ b\ a\ b\ a\ a\ a\ a\ a$

$p\ q\ r\ r\ q\ r\ q\ s\ s\ s\ s\ q$

x

y

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

Dense case

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

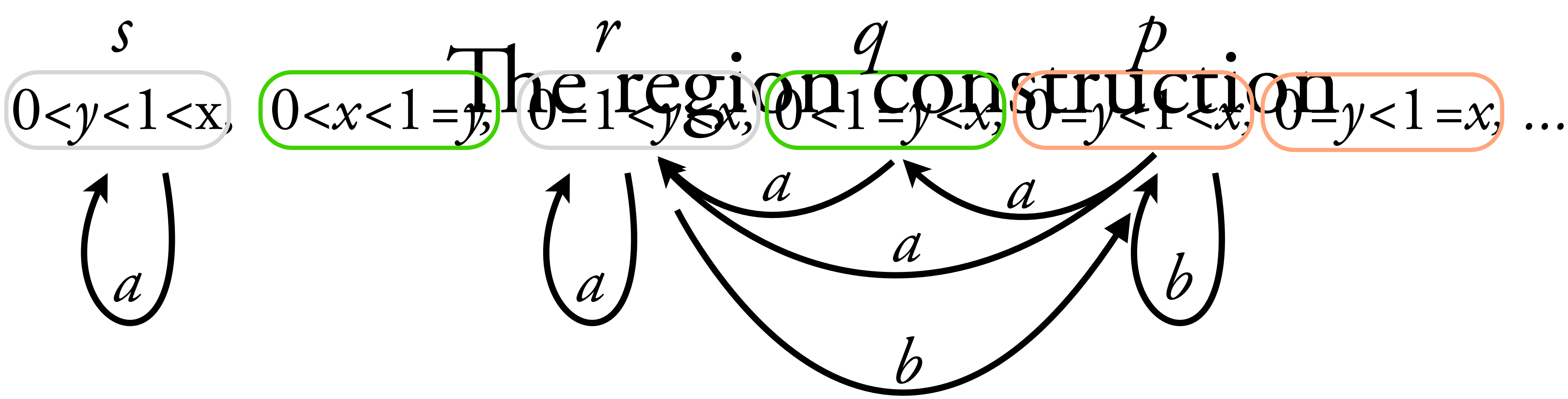
τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction



$a\ a\ a\ b\ a\ b\ a\ a\ a\ a\ a$

$p\ q\ r\ r\ q\ r\ q\ s\ s\ s\ s\ q$

x 5

y 0

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

Dense case

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

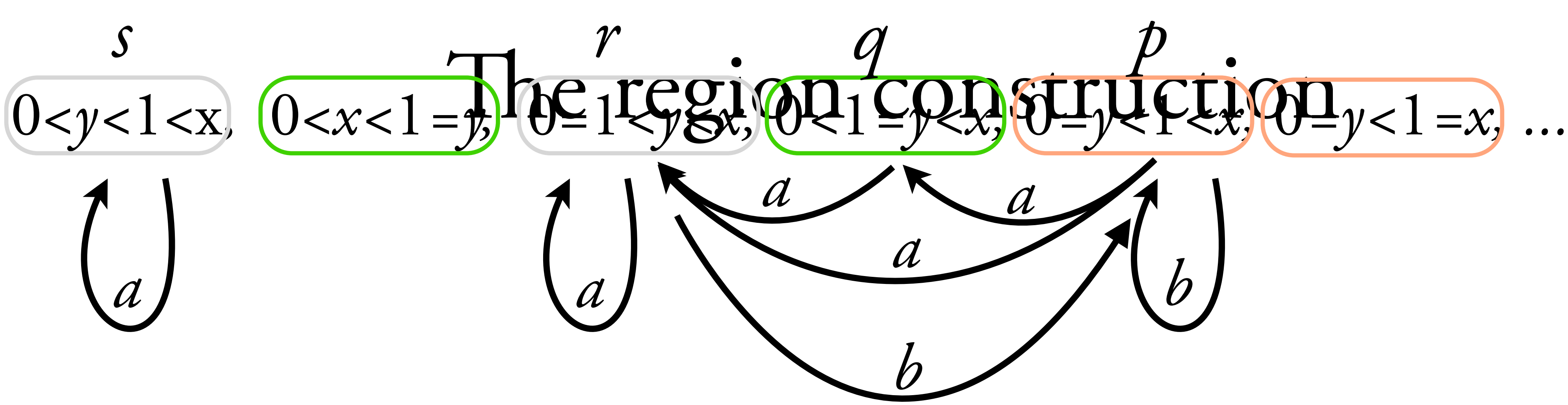
τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction



		a	*a*	*a*	*b*	*a*	*b*	*a*	*a*	*a*	*a*	*a*
	p	*q*	*r*	*r*	*q*	*r*	*q*	*s*	*s*	*s*	*s*	*q*
x	5	5										
y	0	1										

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

Dense case

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

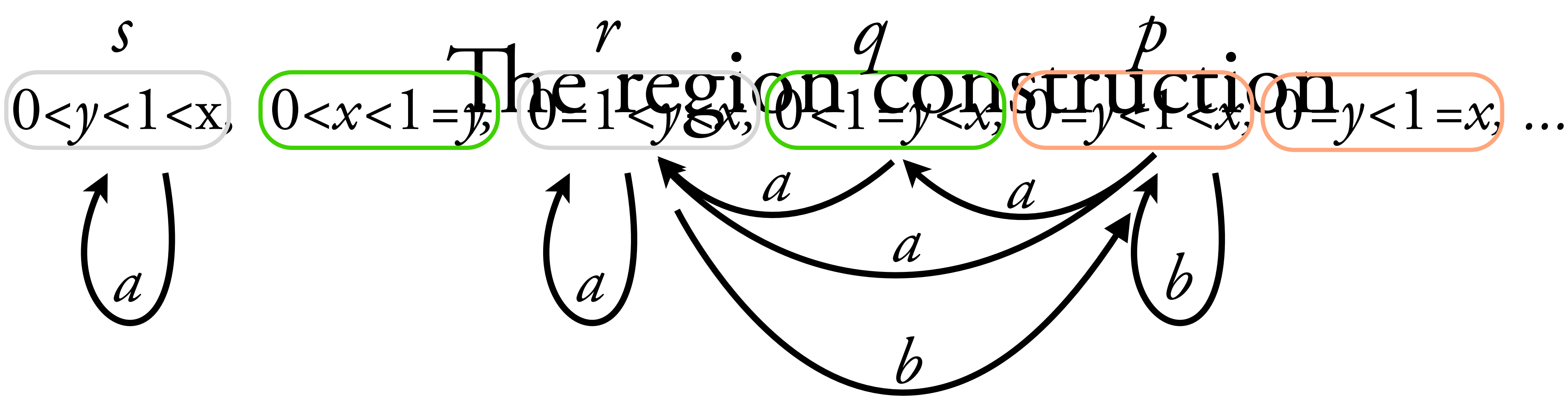
τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction



a a a b a b a a a a a

p q r r q r q s s s s q

x	5	5	5
y	0	1	2

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

Dense case

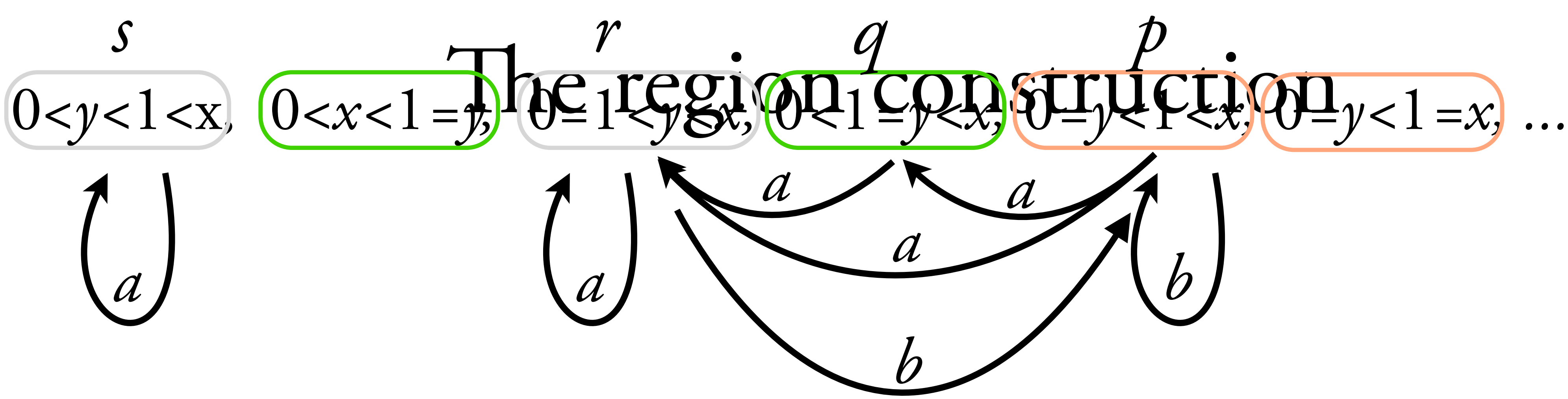
- $D = \mathbb{Q}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



	a	a	a	b	a	b	a	a	a	a	a
	p	q	r	r	q	r	q	s	s	s	s	q
x	5	5	5	5							
y	0	1	2	4							

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

Dense case

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

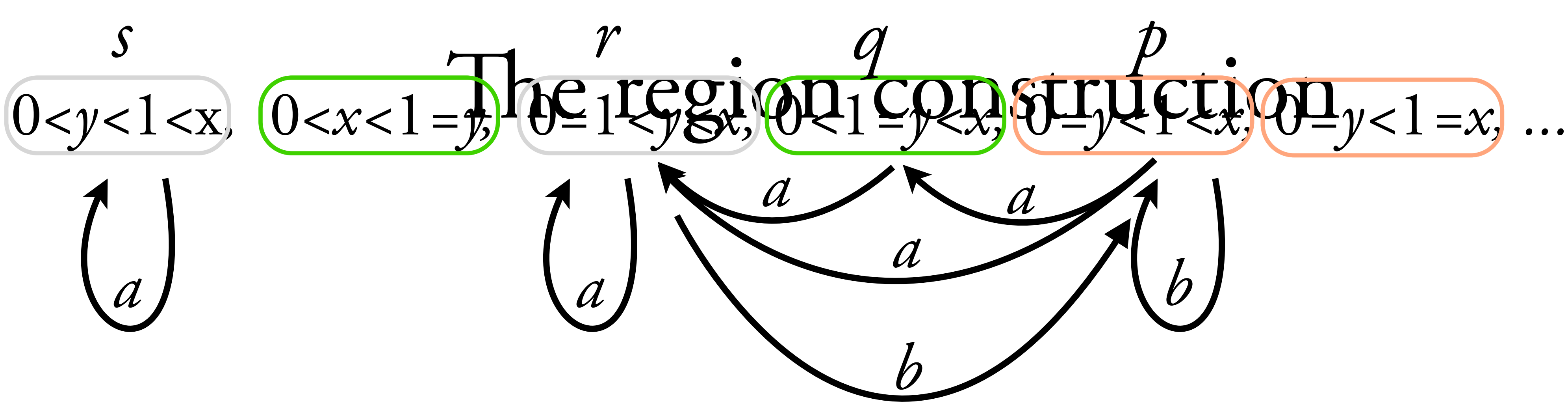
τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction



$a\ a\ a\ b\ a\ b\ a\ a\ a\ a\ a$

$p\ q\ r\ r\ q\ r\ q\ s\ s\ s\ s\ q$

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

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

Dense case

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

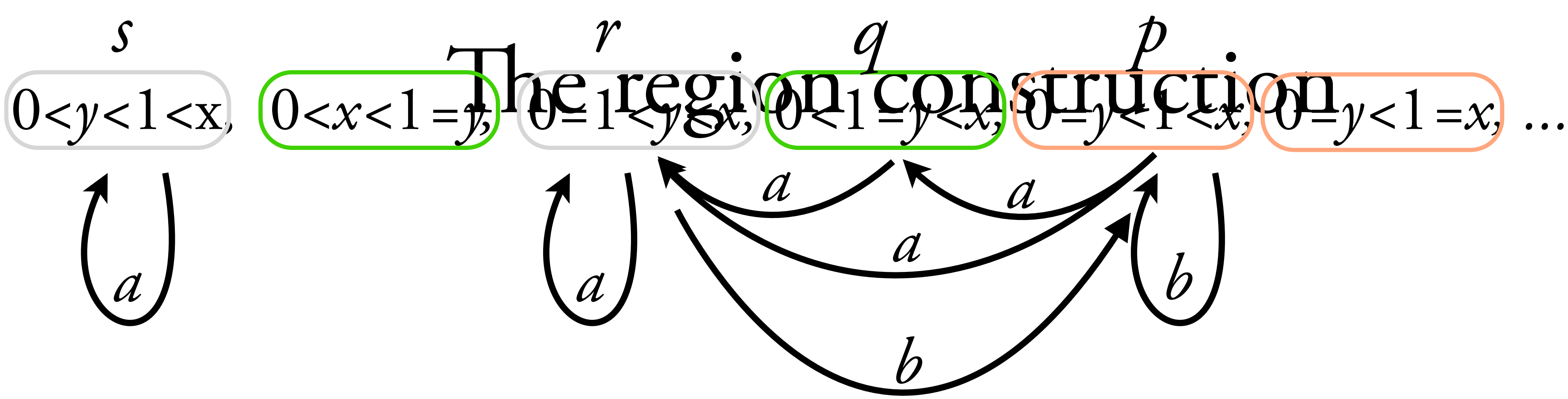
τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction



		a	a	a	b	a	b	a	a	a	a	a
	p	q	r	r	q	r	q	s	s	s	s	q
x	5	5	5	5	5	5						
y	0	1	2	4	0	3						

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

Dense case

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

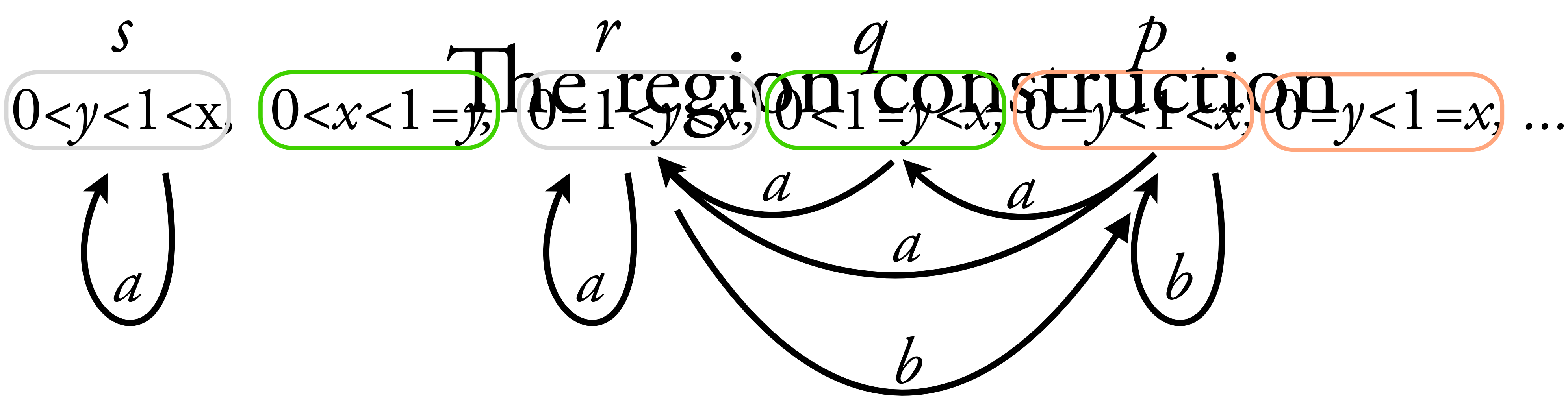
τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction



		a	*a*	*a*	*b*	*a*	*b*	*a*	*a*	*a*	*a* *a*
	p	*q*	*r*	*r*	*q*	*r*	*q*	*s*	*s*	*s*	*s* *q*
x	5	5	5	5	5	5	5				
y	0	1	2	4	0	3	0				

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

Dense case

The region construction

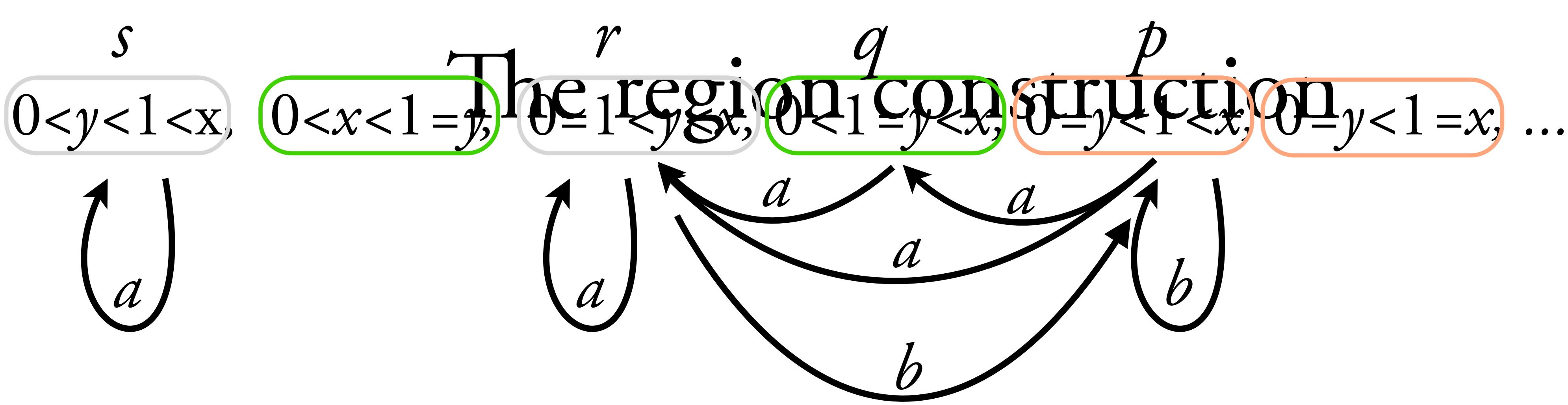
- $D = \mathbb{Q}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



$$\begin{array}{cccccccccccc} & a & a & a & b & a & b & a & a & a & a & a \\ & p & q & r & r & q & r & q & s & s & s & s\ \ q \\ x & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & & & \\ y & 0 & 1 & 2 & 4 & 0 & 3 & 0 & .1 & & & \end{array}$$

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

Dense case

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

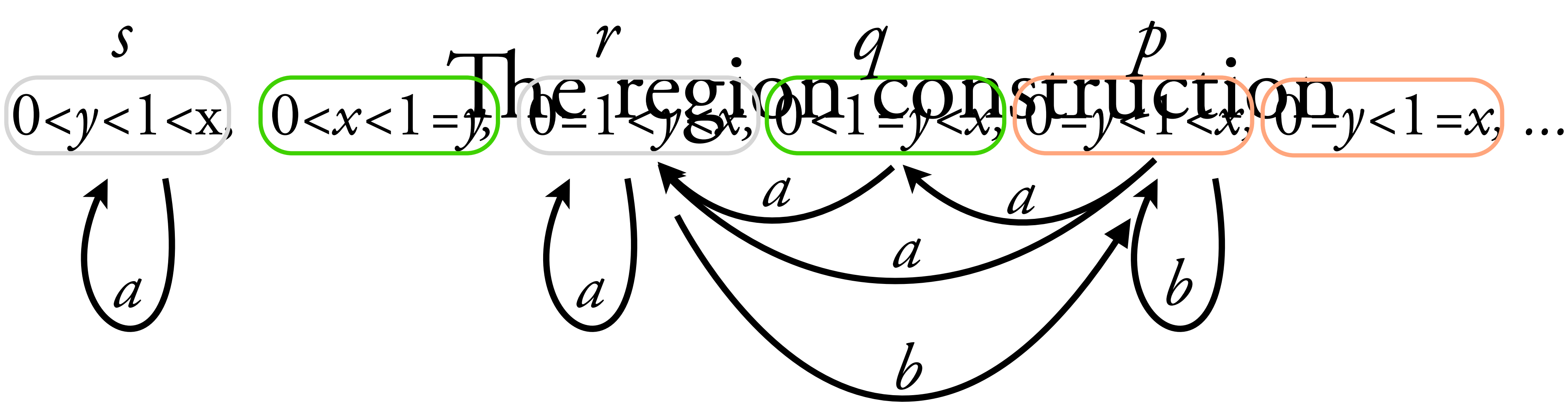
τ_I: $(x{>}0) \wedge (y{=}0)$

δ_a: $(x'{=}x) \wedge (y{<}y'{<}x)$

δ_b: $(x'{=}x) \wedge (y'{=}0)$

τ_F: $(y{=}1)$

The region construction



	a	a	a	b	a	b	a	a	a	a	a
	p	q	r	r	q	r	q	s	s	s	s	q
x	5	5	5	5	5	5	5	5	5		
y	0	1	2	4	0	3	0	.1	.2		

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

Dense case

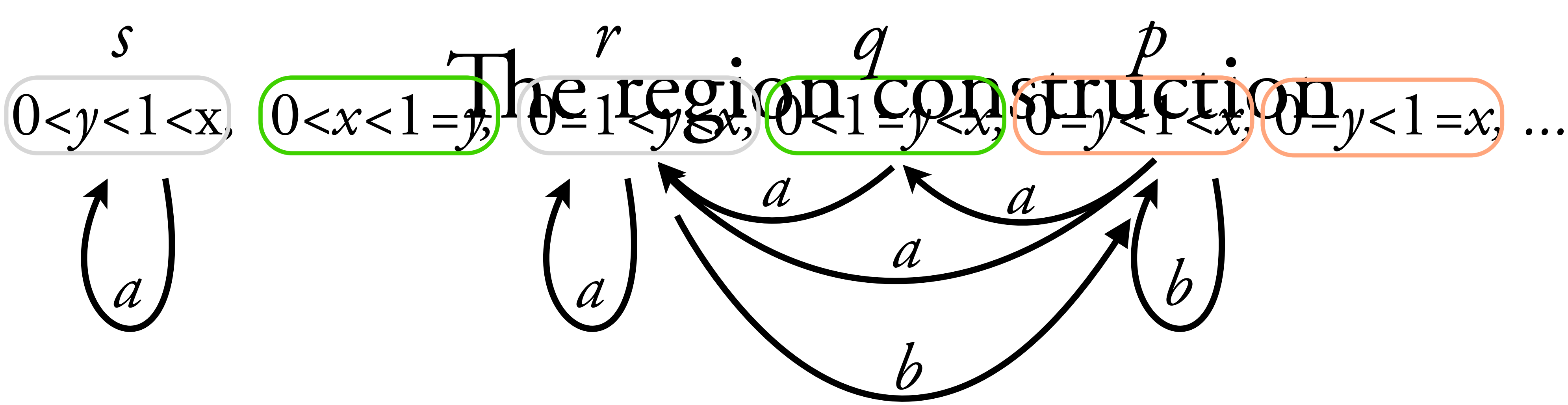
- $D = \mathbb{Q}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



		a	*a*	*a*	*b*	*a*	*b*	*a*	*a*	*a*	*a*
	p	*q*	*r*	*r*	*q*	*r*	*q*	*s*	*s*	*s*	*s*

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

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

Dense case

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

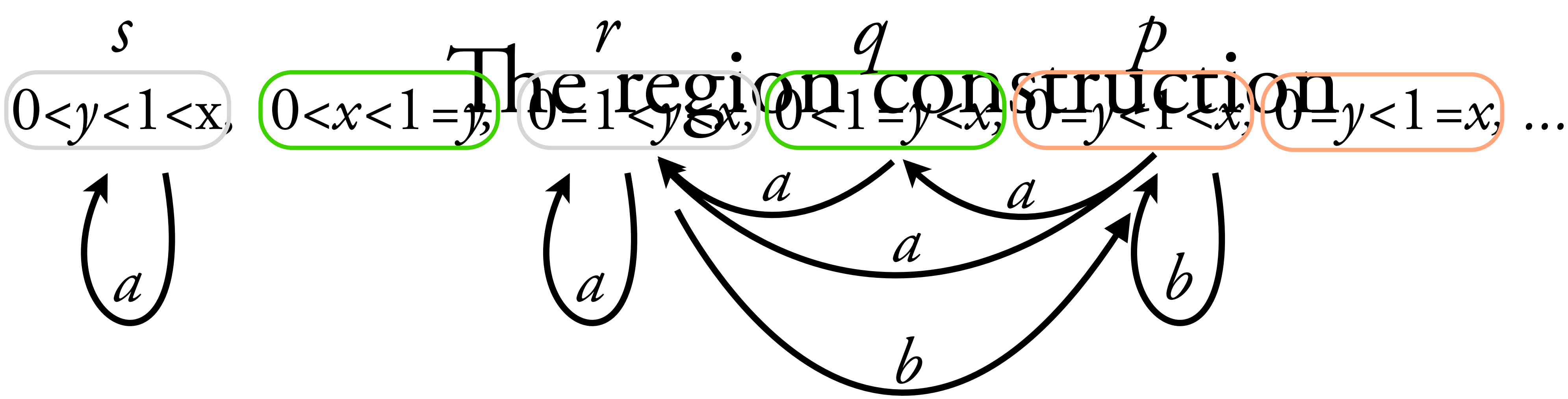
τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction



	a	*a*	*a*	*b*	*a*	*b*	*a*	*a*	*a*	*a*	*a*
p	*q*	*r*	*r*	*q*	*r*	*q*	*s*	*s*	*s*	*s*	*q*
x	5	5	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	.1	.2	.4	.9

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

Dense case

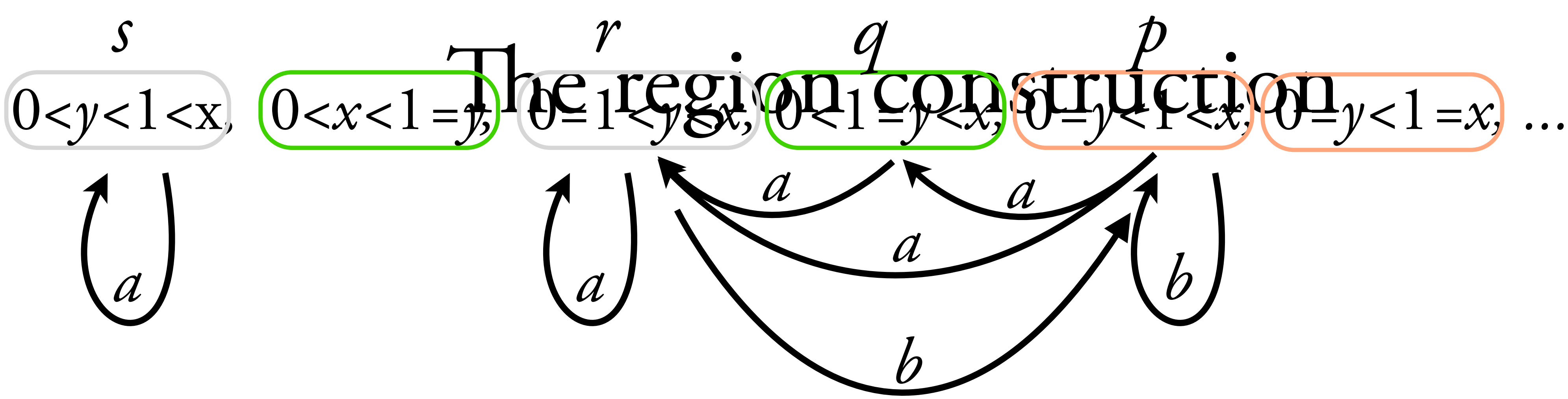
- $D = \mathbb{Q}$
- finite words
- no database

τ_{I}: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_{F}: $(y=1)$



The region construction

		a	*a*	*a*	*b*	*a*	*b*	*a*	*a*	*a*	*a*	*a*
	p	*q*	*r*	*r*	*q*	*r*	*q*	*s*	*s*	*s*	*s*	*q*
x	5	5	5	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	.1	.2	.4	.9	1

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

Dense case

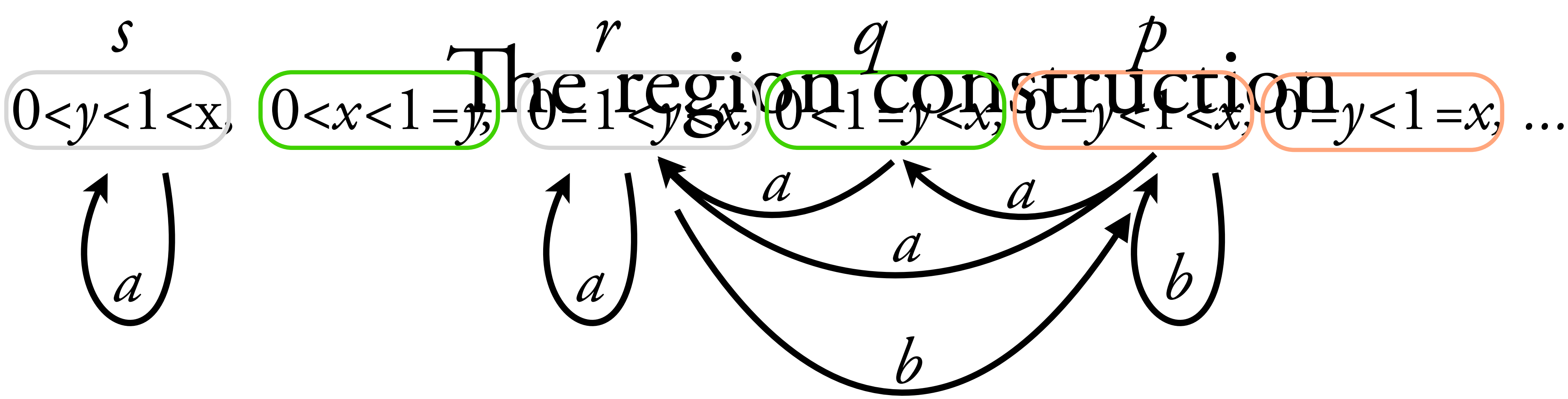
- $D = \mathbb{Q}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



		a	*a*	*a*	*b*	*a*	*b*	*a*	*a*	*a*	*a*	*a*
	p	*q*	*r*	*r*	*q*	*r*	*q*	*s*	*s*	*s*	*s*	*q*
x	5	5	5	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	.1	.2	.4	.9	1

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

Dense case

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

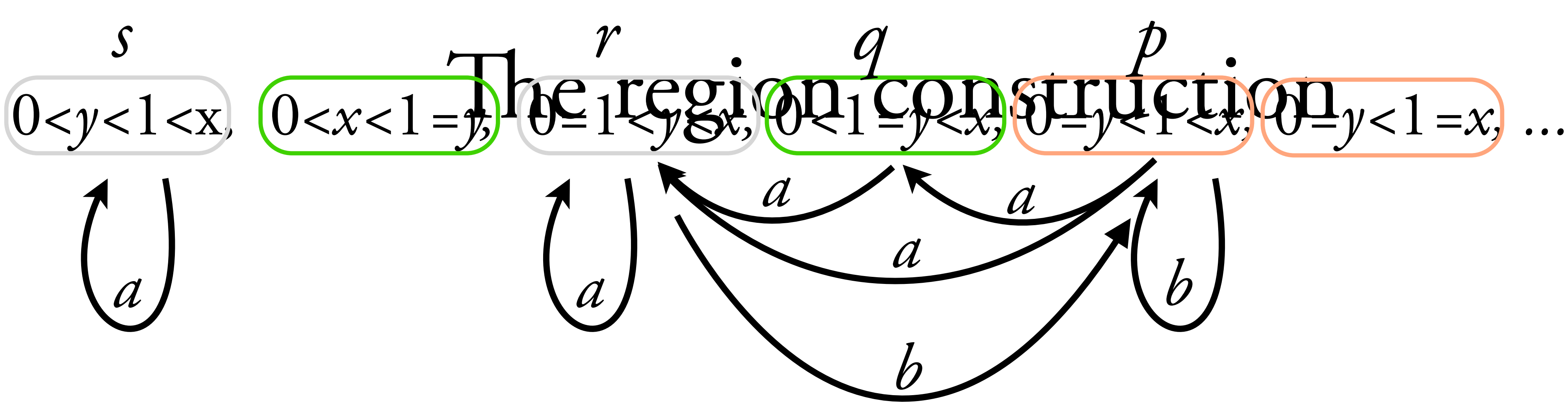
τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$

The region construction



		a	a	a	b	a	b	a	a	a	a	a
	p	q	r	r	q	r	q	s	s	s	s	q
x	5	5	5	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	.1	.2	.4	.9	1

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

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

The region construction

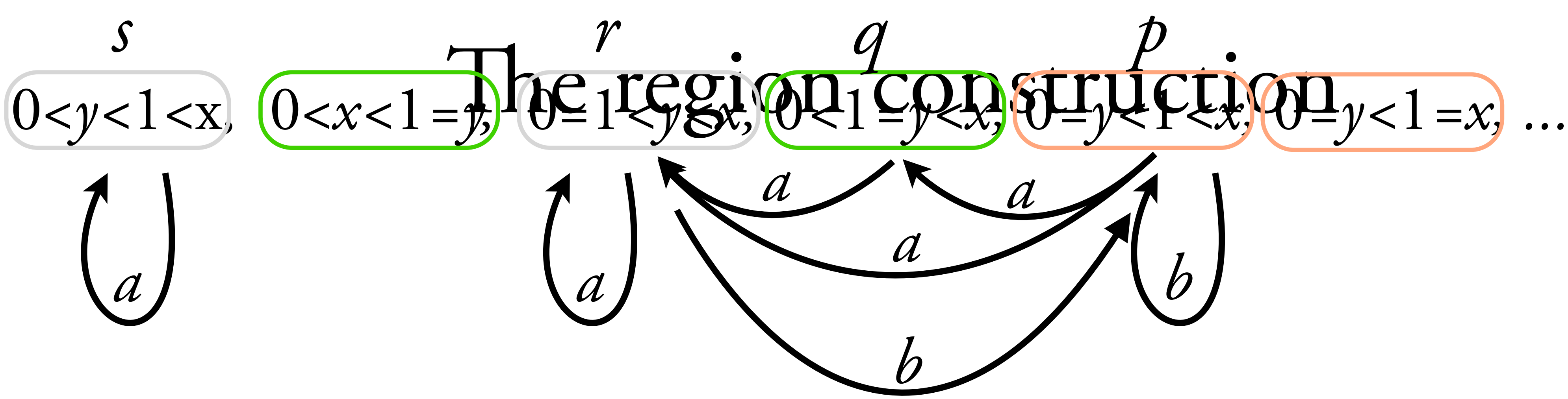
- $D = \mathbb{Q}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



		a	*a*	*a*	*b*	*a*	*b*	*a*	*a*	*a*	*a*	*a*
	p	*q*	*r*	*r*	*q*	*r*	*q*	*s*	*s*	*s*	*s*	*q*
x	5	5	5	5	5	5	5	5	5	5	5	5
y	0	1	2	4	0	3	0	.1	.2	.4	.9	1

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

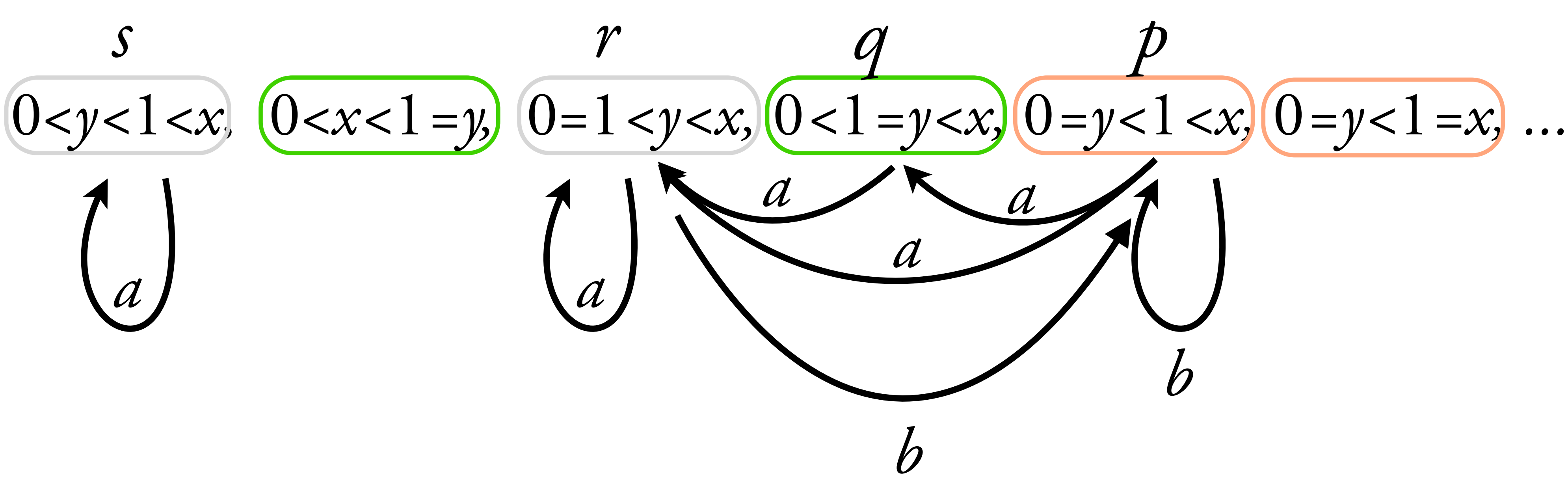
- $D = \mathbb{N}$
- finite words
- no database

τ_I: $\boxed{(x>0) \wedge (y=0)}$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $\boxed{(y=1)}$



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

Discrete case

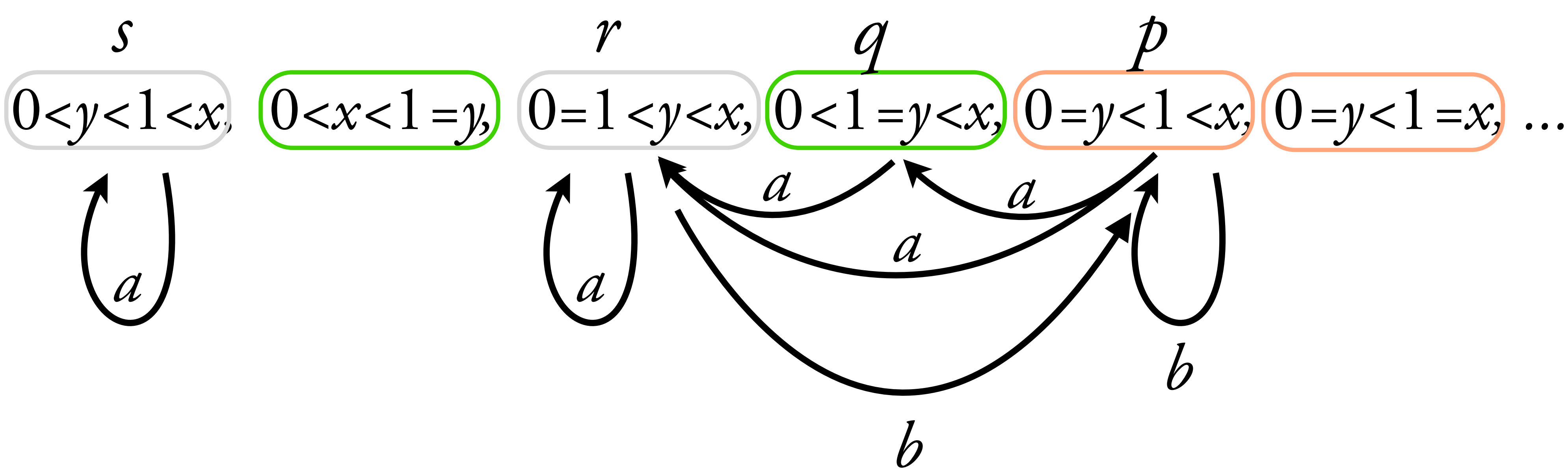
- $D = \mathbb{N}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



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

Discrete case

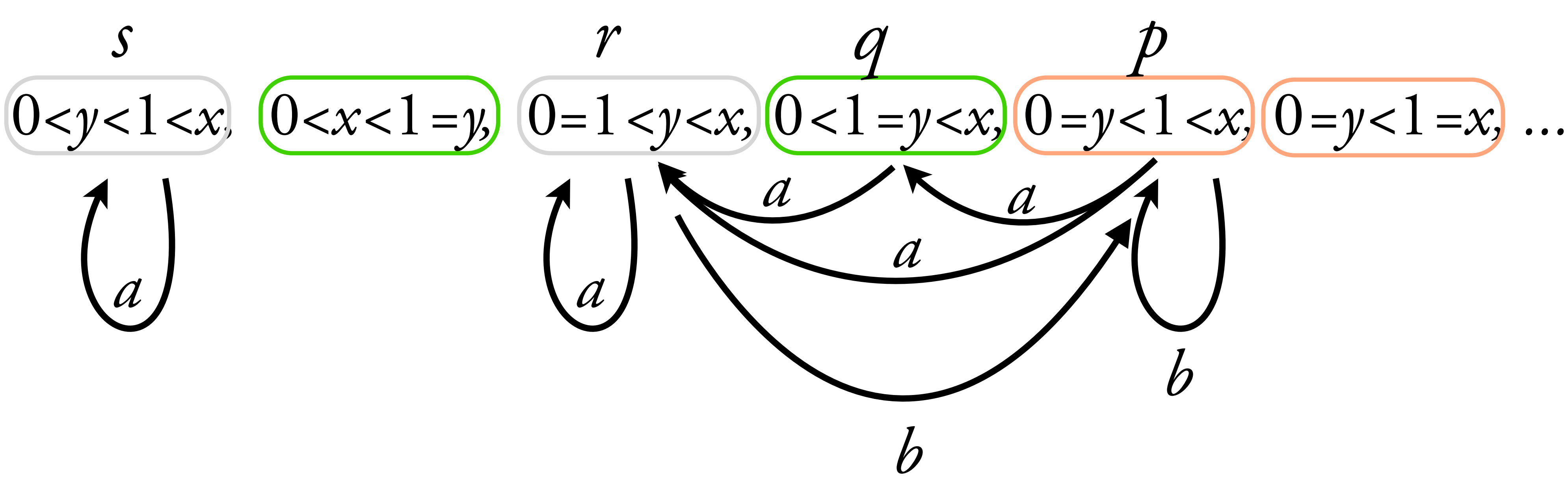
- $D = \mathbb{N}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



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

Discrete case

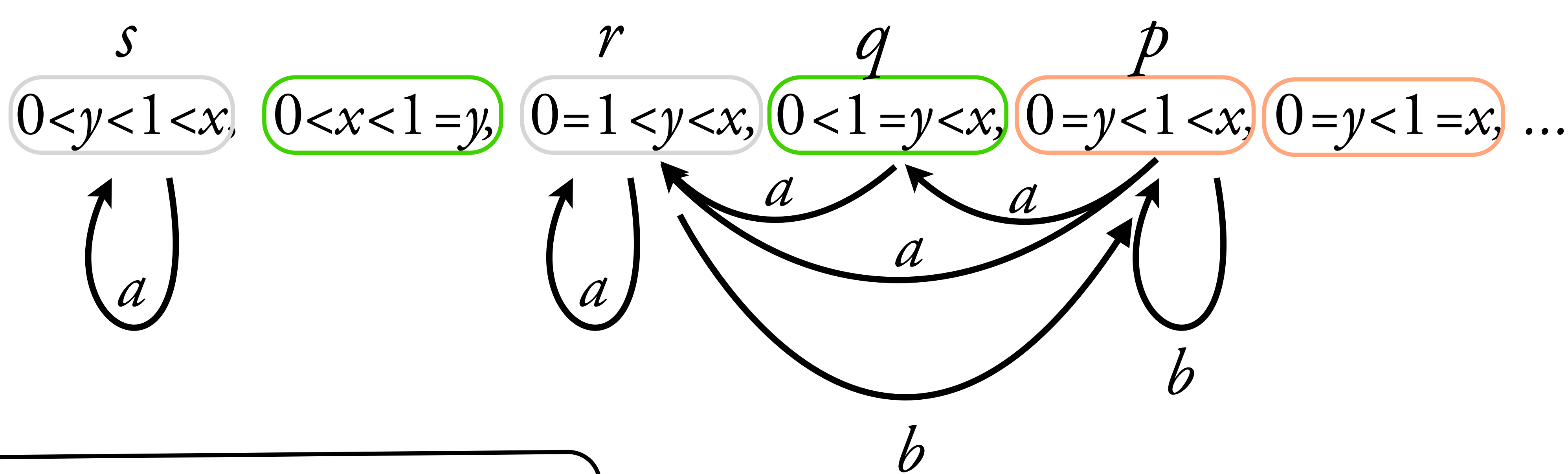
- $D = \mathbb{N}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



$a\ a\ a\ b\ a\ b\ a\ a\ a\ a\ a$
$p\ s\ r\ r\ q\ r\ p\ s\ r\ r\ r\ q$

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

Discrete case

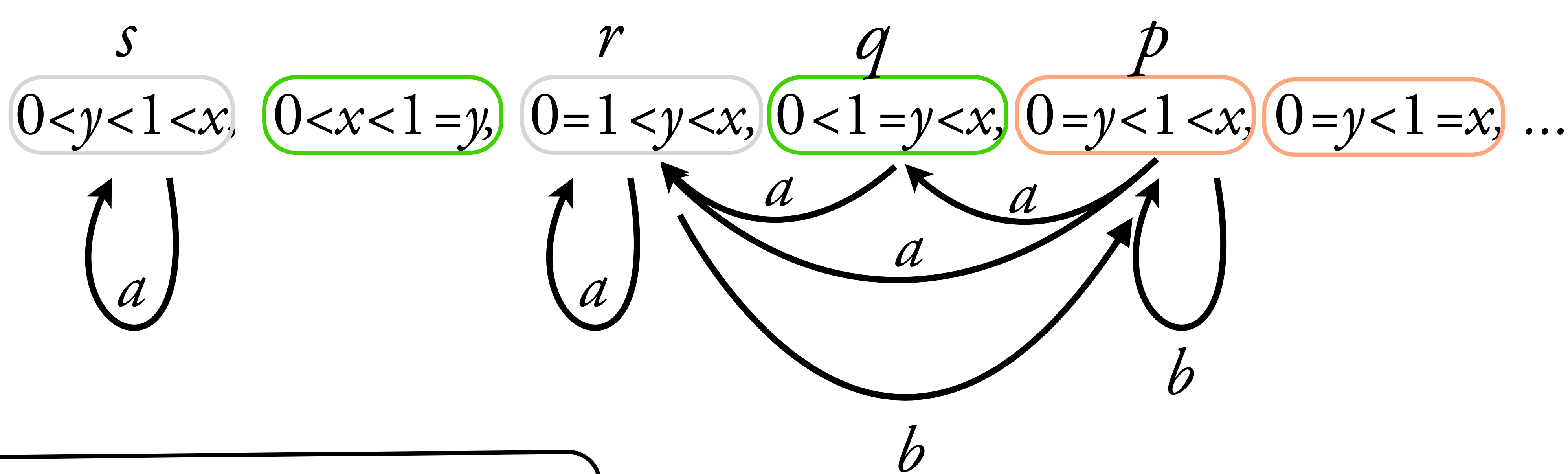
- $D = \mathbb{N}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



$a\ a\ a\ b\ a\ b\ a\ a\ a\ a\ a$

$p\ s\ r\ r\ q\ r\ p\ s\ r\ r\ r\ q$

x

y

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

Discrete case

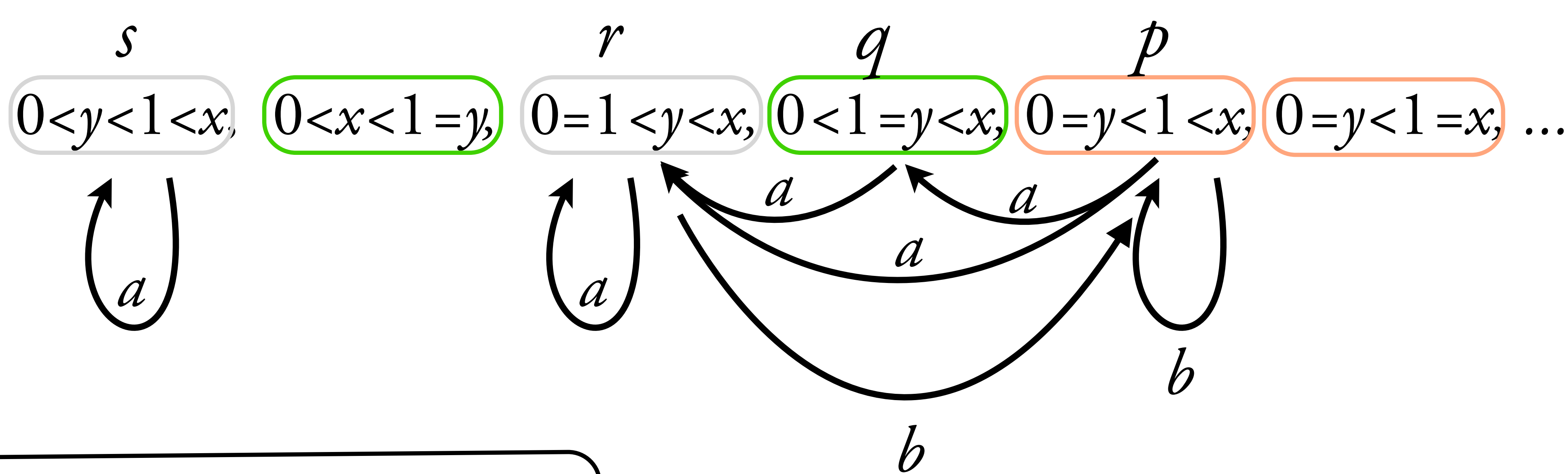
- $D = \mathbb{N}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



	a	*a*	*a*	*b*	*a*	*b*	*a*	*a*	*a*	*a*	*a*	
	p	*s*	*r*	*r*	*q*	*r*	*p*	*s*	*r*	*r*	*r*	*q*
x	5	5	5	5	5	5	5	5				
y	0	1	2	4	0	4	0	1				

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

Discrete case

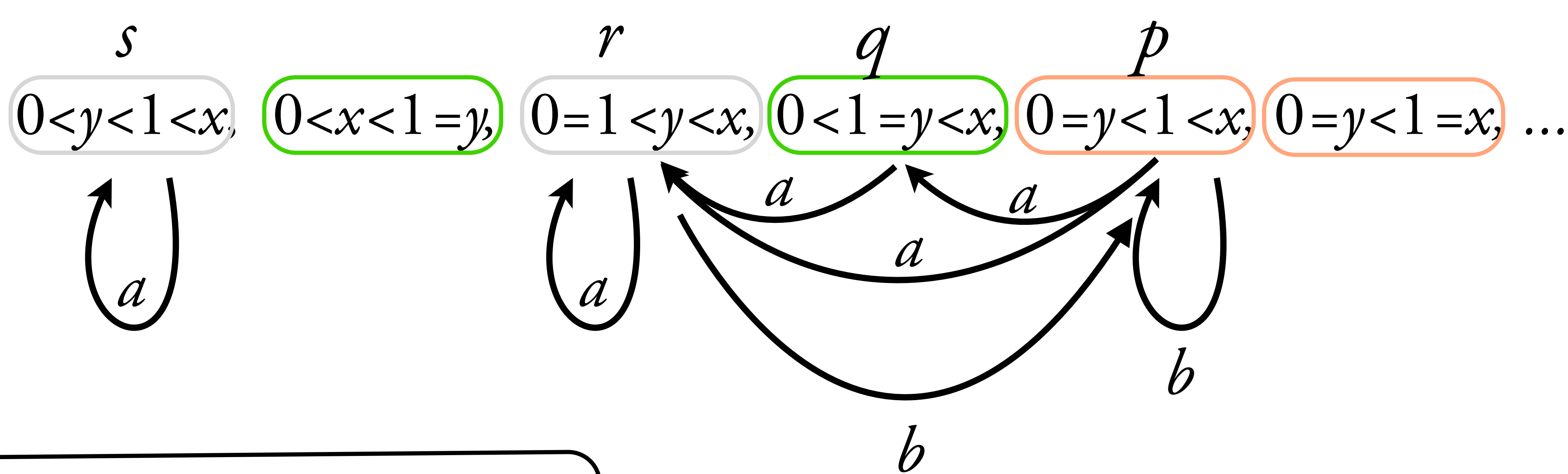
- $D = \mathbb{N}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



a a a b a b a a a a a

p s r r q r p s r r r q

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

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

Discrete case

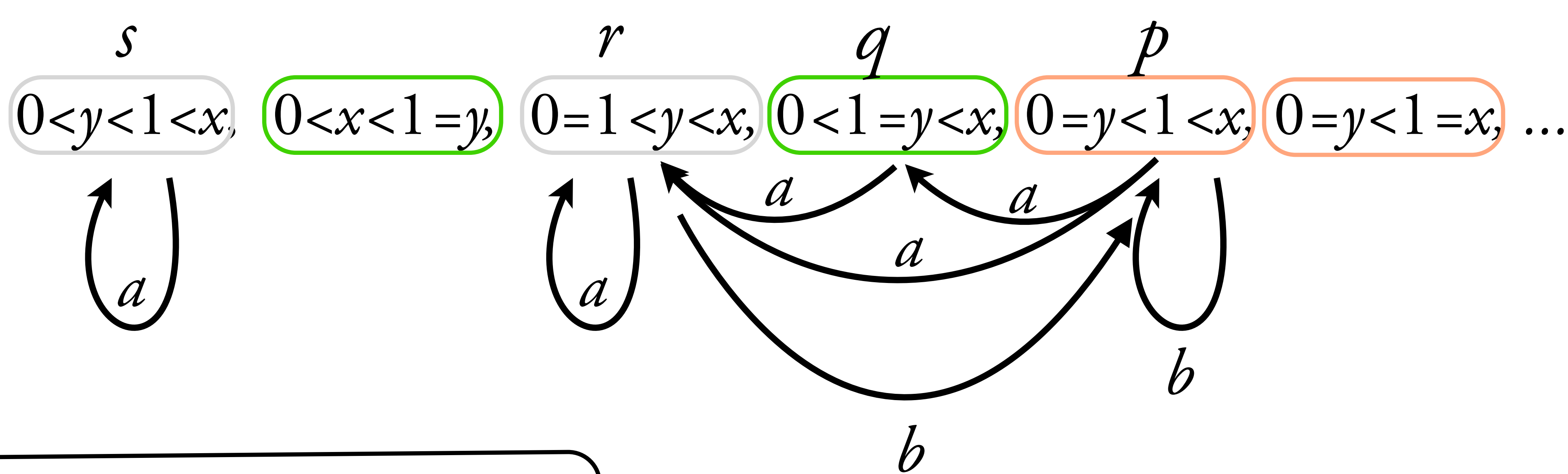
- $D = \mathbb{N}$
- finite words
- no database

τ_I: $\boxed{(x>0) \wedge (y=0)}$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $\boxed{(y=1)}$



$a\ a\ a\ b\ a\ b\ a\ a\ a\ a\ a$

$p\ s\ r\ r\ q\ r\ p\ s\ r\ r\ r\ q$

x 5 5 5 5 5 5 5 5 5 ?

y 0 1 2 4 0 4 0 1 4 ?

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

Discrete case

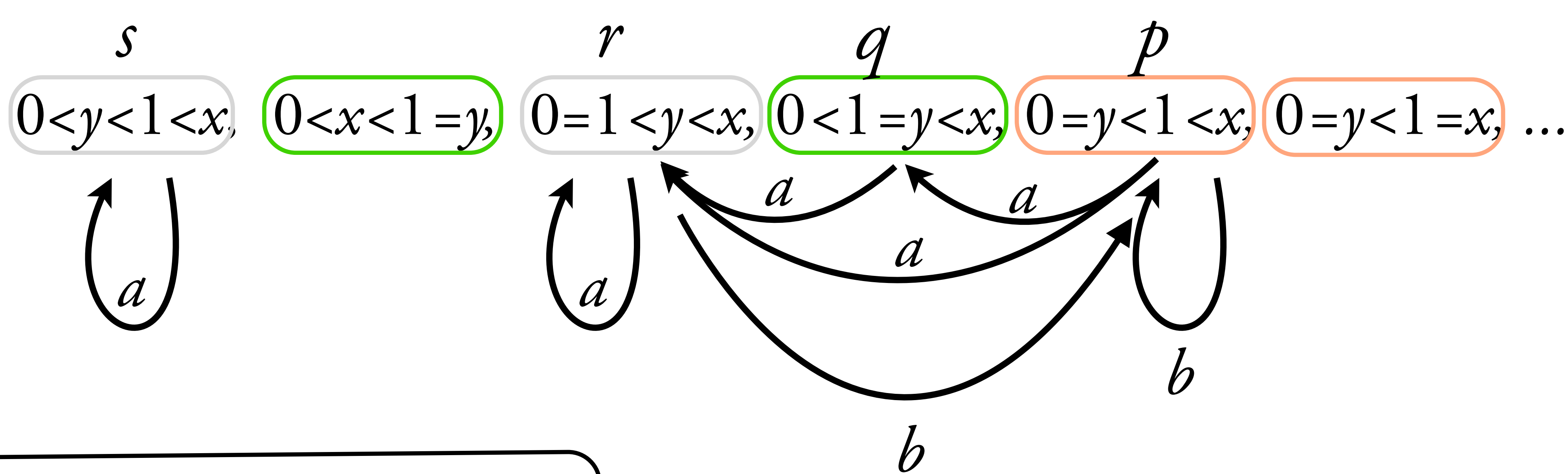
- $D = \mathbb{N}$
- finite words
- no database

τ_I: $(x>0) \wedge (y=0)$

δ_a: $(x'=x) \wedge (y<y'<x)$

δ_b: $(x'=x) \wedge (y'=0)$

τ_F: $(y=1)$



a a a b a b a a a a a
p s r r q r p s r r r q

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

but: in any cell and any n we can find configurations which are *n-bisimilar* → *finite* runs of the region automaton correspond to runs of $\mathcal{A}$

An elaboration of these ideas yields the results for infinite runs and database constraints and unary predicates in $\mathscr{D}$.

An elaboration of these ideas yields the results for infinite runs and database constraints and unary predicates in $\mathscr{D}$.

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

An elaboration of these ideas yields the results for infinite runs and database constraints and unary predicates in $\mathscr{D}$.

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

Theorem. $\mathbb{N}$-automata over infinite words accept B-regular languages.

An elaboration of these ideas yields the results for infinite runs and database constraints and unary predicates in $\mathscr{D}$.

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

Theorem. $\mathbb{N}$-automata over infinite words accept B-regular languages.

Theorem. For any reasonable structure $\mathscr{D}$, emptiness of $\mathscr{D}$-automata is decidable and in PSPACE.

An elaboration of these ideas yields the results for infinite runs and database constraints and unary predicates in $\mathscr{D}$.

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

Theorem. $\mathbb{N}$-automata over infinite words accept B-regular languages.

Theorem. For any reasonable structure $\mathscr{D}$, emptiness of $\mathscr{D}$-automata is decidable and in PSPACE.

Corollary. Deciding LTL+data tests properties of $\mathscr{D}$-automata is PSPACE-complete.

Thank you!

Thank you!

Thank you!