Deciding Emptiness of min-automata

Szymon Toruńczyk
joint work with
Mikołaj Bojańczyk

LSV Cachan / University of Warsaw

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$c{:=}c+1$

$c{:=}\min(d,e)$

acceptance condition is a boolean combination of:

$\liminf(c) = \infty$
||
"c tends to ∞"

Example. $L = \{a^{n_1}b\ a^{n_2}b\ a^{n_3}b\ldots : n_1, n_2\ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c{:=}c+1$
-when reading b, do $d{:=}\min(c,c);\ c{:=}\min(z,z)$

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$c{:=}c+1$

$c{:=}min(d,e)$

acceptance condition is a boolean combination of:

$\liminf(c) = \infty$
||
"c tends to ∞"

Example. $L = \{a^{n_1}b\ a^{n_2}b\ a^{n_3}b\ldots : n_1,n_2\ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c{:=}c+1$
-when reading b, do $d{:=}min(c,c);\ c{:=}min(z,z)$

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$c{:=}c+1$

$c{:=}min(d,e)$

acceptance condition is a boolean combination of:

$\liminf(c) = \infty$
||
"c tends to ∞"

Example. $L = \{a^{n_1} b\ a^{n_2} b\ a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c{:=}c+1$
-when reading b, do $d{:=}min(c,c)$; $c{:=}min(z,z)$

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$c{:=}c+1$

$c{:=}\min(d,e)$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$
||
"c tends to ∞"

Example. $L = \{a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c{:=}c+1$
-when reading b, do $d{:=}\min(c,c);\ c{:=}\min(z,z)$

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$$c := c+1$$

$$c := \min(d,e)$$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$
||
"c tends to ∞"

Example. $L = \{a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c := c+1$
-when reading b, do $d := \min(c,c);\ c := \min(z,z)$

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$$c{:=}c+1$$

$$c{:=}min(d,e)$$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$

||

"c tends to ∞"

Example. $L = \{a^{n_1} b\ a^{n_2} b\ a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c{:=}c+1$
-when reading b, do $d{:=}min(c,c);\ c{:=}min(z,z)$

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$$c := c + 1$$

$$c := \min(d, e)$$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$

||

"c tends to ∞"

Example. $L = \{a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c:=c+1$
-when reading b, do $d:=\min(c,c)$; $c:=\min(z,z)$

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$$c := c+1$$

$$c := min(d,e)$$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$

||

"c tends to ∞"

Example. $L = \{a^{n_1} b\ a^{n_2} b\ a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c:=c+1$
-when reading b, do $d:=min(c,c);\ c:=min(z,z)$

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$c:=c+1$

$c:=min(d,e)$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$

||

"c tends to ∞"

Example. $L = \{a^{n_1} b\ a^{n_2} b\ a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c:=c+1$
-when reading b, do $d:=min(c,c);\ c:=min(z,z)$

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

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$c{:=}c+1$

$c{:=}min(d,e)$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$
||
"c tends to ∞"

Example. $L = \{a^{n_1} b\ a^{n_2} b\ a^{n_3} b\ldots : n_1, n_2\ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c{:=}c+1$
-when reading b, do $d{:=}min(c,c);\ c{:=}min(z,z)$

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

c
d
z

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$$c := c+1$$

$$c := min(d,e)$$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$

||

"c tends to ∞"

Example. $L = \{a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c:=c+1$
-when reading b, do $d:=min(c,c);\ c:=min(z,z)$

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

	a a a b a b a a b...
c	0
d	0
z	0

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$c := c+1$

$c := min(d,e)$

acceptance condition is a boolean combination of:

$\liminf(c) = \infty$

||

"c tends to ∞"

Example. $L = \{a^{n_1} b\ a^{n_2} b\ a^{n_3} b\ldots : n_1, n_2\ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c := c+1$
-when reading b, do $d := min(c,c);\ c := min(z,z)$

		a	a	a	b	a	b	a	a	b...
c	0	1								
d	0	0								
z	0	0								

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$$c := c+1$$

$$c := \min(d,e)$$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$

||

"c tends to ∞"

Example. $L = \{a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c := c+1$
-when reading b, do $d := \min(c,c);\ c := \min(z,z)$

	a	a	a	b	a	b	a	a	$b\ldots$
c	0	1	2						
d	0	0	0						
z	0	0	0						

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$$c := c + 1$$

$$c := \min(d, e)$$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$

$$||$$

"c tends to ∞"

Example. $L = \{a^{n_1} b\ a^{n_2} b\ a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c:=c+1$
-when reading b, do $d:=\min(c,c);\ c:=\min(z,z)$

	a	a	a	b	a	b	a	a	$b \ldots$
c	0	1	2	3					
d	0	0	0	0					
z	0	0	0	0					

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$$c := c+1$$

$$c := \min(d,e)$$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$

||

"c tends to ∞"

Example. $L = \{a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c:=c+1$
-when reading b, do $d:=\min(c,c);\ c:=\min(z,z)$

	a	a	a	b	a	b	a	a	$b \ldots$
c	0	1	2	3	0				
d	0	0	0	0	3				
z	0	0	0	0	0				

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$$c := c+1$$

$$c := \min(d,e)$$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$

$$||$$

"c tends to ∞"

Example. $L = \{a^{n_1} b\ a^{n_2} b\ a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c:=c+1$
-when reading b, do $d:=\min(c,c);\ c:=\min(z,z)$

	a	a	a	b	a	b	a	a	$b\ldots$
c	0	1	2	3	0	1			
d	0	0	0	0	3	3			
z	0	0	0	0	0	0			

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$$c := c+1$$

$$c := \min(d,e)$$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$

||

"c tends to ∞"

Example. $L = \{a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$

Min-automaton has one state and three counters: c,d,z

-when reading a, do $c:=c+1$

-when reading b, do $d:=\min(c,c);\ c:=\min(z,z)$

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

c	0	1	2	3	0	1	0
d	0	0	0	0	3	3	1
z	0	0	0	0	0	0	0

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$$c{:=}c+1$$

$$c{:=}min(d,e)$$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$

||

"c tends to ∞"

Example. $L = \{a^{n_1} b\ a^{n_2} b\ a^{n_3} b \ldots : n_1, n_2 \ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c{:=}c+1$
-when reading b, do $d{:=}min(c,c);\ c{:=}min(z,z)$

	a	a	a	b	a	b	a	a	b…
c	0	1	2	3	0	1	0	1	
d	0	0	0	0	3	3	1	1	
z	0	0	0	0	0	0	0	0	

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$$c:=c+1$$

$$c:=min(d,e)$$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$
||
"c tends to ∞"

Example. $L = \{a^{n_1}b\ a^{n_2}b\ a^{n_3}b\ldots : n_1, n_2\ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: c,d,z
-when reading a, do $c:=c+1$
-when reading b, do $d:=min(c,c);\ c:=min(z,z)$

	a	a	a	b	a	b	a	a	b...
c	0	1	2	3	0	1	0	1	2
d	0	0	0	0	3	3	1	1	1
z	0	0	0	0	0	0	0	0	0

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Min-automata

deterministic automata with counters
transitions invoke counter operations:

$$c{:=}c+1$$

$$c{:=}\min(d,e)$$

acceptance condition is a boolean combination of:

$$\liminf(c) = \infty$$

||

"c tends to ∞"

Example. $L = \{a^{n_1} b\ a^{n_2} b\ a^{n_3} b\ldots : n_1, n_2\ldots$ does not converge to $\infty\}$
Min-automaton has one state and three counters: *c,d,z*
-when reading *a*, do $c{:=}c+1$
-when reading *b,* do $d{:=}\min(c,c);\ c{:=}\min(z,z)$

		a	*a*	*a*	*b*	*a*	*b*	*a*	*a*	*b*...
c	0	1	2	3	0	1	0	1	2	0
d	0	0	0	0	3	3	1	1	1	2
z	0	0	0	0	0	0	0	0	0	0

Acceptance condition: $\neg \liminf(c) = \infty \wedge \neg \liminf(d) = \infty$

Logic

Max-automata

Logic

Max-automata

Extension of WMSO by the quantifier

Logic

Max-automata

Extension of WMSO by the quantifier

$$\mathsf{U}X \quad \varphi(X)$$

which says

„*there exist arbitrarily large (finite) sets X, satisfying* $\varphi(X)$"

Logic

Max-automata

Extension of WMSO by the quantifier

$$\mathsf{U}X \;\; \varphi(X)$$

which says

„there exist arbitrarily large (finite) sets X, satisfying $\varphi(X)$"

Language: $\{a^{n_1} b\, a^{n_2} b\, a^{n_3} b\ldots : n_1\, n_2\, n_3\ldots$ is unbounded$\}$

Logic

Max-automata

Extension of WMSO by the quantifier

$$\mathsf{U}X \;\; \varphi(X)$$

which says

„there exist arbitrarily large (finite) sets X, satisfying $\varphi(X)$"

Language: $\{a^{n_1} b\ a^{n_2} b\ a^{n_3} b \ldots : n_1\, n_2\, n_3 \ldots$ is unbounded$\}$

$\mathsf{U}X$ *"X is a block of a's"*

Logic

Max-automata $\longleftrightarrow$ Min-automata

Extension of WMSO by the quantifier

$$\mathsf{U}X \;\; \varphi(X)$$

which says

„*there exist arbitrarily large (finite) sets X, satisfying* $\varphi(X)$"

Language: $\{a^{n_1} b\; a^{n_2} b\; a^{n_3} b \ldots : n_1\, n_2\, n_3 \ldots$ is unbounded$\}$

$\mathsf{U}X$ *"X is a block of a's"*

Logic

Max-automata $\longleftrightarrow$ Min-automata

Extension of WMSO by the quantifier

$\mathsf{U}X \ \varphi(X)$ $\longleftrightarrow$ $\mathsf{R}X \ \varphi(X)$

which says

„*there exist arbitrarily large (finite) sets X, satisfying* $\varphi(X)$"

Language: $\{a^{n_1} b\ a^{n_2} b\ a^{n_3} b \ldots : n_1\, n_2\, n_3 \ldots$ is unbounded$\}$

$\mathsf{U}X$ *"X is a block of a's"*

Logic

Max-automata $\longleftrightarrow$ Min-automata

Extension of WMSO by the quantifier

$\mathsf{U}X\ \ \varphi(X)$ $\longleftrightarrow$ $\mathsf{R}X\ \ \varphi(X)$

which says

„*there exist infinitely many sets X of same size, satisfying* $\varphi(X)$"

Language: $\{a^{n_1} b\ a^{n_2} b\ a^{n_3} b \ldots : n_1\, n_2\, n_3 \ldots$ is unbounded$\}$

$\mathsf{U}X$ *"X is a block of a's"*

Logic

Max-automata $\longleftrightarrow$ Min-automata

Extension of WMSO by the quantifier

$\mathsf{U}X\ \varphi(X)$ $\longleftrightarrow$ $\mathsf{R}X\ \varphi(X)$

which says

„*there exist infinitely many sets X of same size, satisfying* $\varphi(X)$"

Language: $\{a^{n_1} b\ a^{n_2} b\ a^{n_3} b \dots : n_1\, n_2\, n_3 \dots$ converges to $\infty\}$

$\mathsf{U}X$ *"X is a block of a's"*

Logic

Max-automata $\longleftrightarrow$ Min-automata

Extension of WMSO by the quantifier

$\mathsf{U}X \;\; \varphi(X)$ $\longleftrightarrow$ $\mathsf{R}X \;\; \varphi(X)$

which says

„*there exist infinitely many sets* X *of same size, satisfying* $\varphi(X)$"

Language: $\{a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots : n_1\, n_2\, n_3 \ldots$ converges to $\infty\}$

$\neg\mathsf{R}X$ *"X is a block of a's"*

WMSO + U	$\longleftrightarrow$	WMSO + R
max-automata	$\longleftrightarrow$	min-automata

$$
\begin{array}{ccc}
\text{WMSO} + \mathsf{U} & \longleftrightarrow & \text{WMSO} + \mathsf{R} \\
\| & & \\
\text{max-automata} & \longleftrightarrow & \text{min-automata}
\end{array}
$$

Theorem. WMSO+U has the same expressive power as deterministic max-automata.

$$
\begin{array}{ccc}
\text{WMSO} + \mathsf{U} & \longleftrightarrow & \text{WMSO} + \mathsf{R} \\
\| & & \| \\
\text{max-automata} & \longleftrightarrow & \text{min-automata}
\end{array}
$$

Theorem. WMSO+U has the same expressive power as deterministic max-automata.

Theorem. WMSO+R has the same expressive power as deterministic min-automata.

WMSO + U + R

||

max-automata min-automata

Theorem. WMSO+U has the same expressive power as deterministic max-automata.

Theorem. WMSO+R has the same expressive power as deterministic min-automata.

What if we allow both U and R?

WMSO + U + R

||

max-automata min-automata

Theorem. WMSO+U has the same expressive power as deterministic max-automata.

Theorem. WMSO+R has the same expressive power as deterministic min-automata.

Theorem. WMSO+U+R has the same expressive power as boolean combinations of min- and max-automata.

WMSO + U + R

||

Boolean combinations of min- & max-automata

Theorem. WMSO+U has the same expressive power as deterministic max-automata.

Theorem. WMSO+R has the same expressive power as deterministic min-automata.

Theorem. WMSO+U+R has the same expressive power as boolean combinations of min- and max-automata.

WMSO + U + R

||

Boolean combinations of min- & max-automata

Theorem. WMSO+U has the same expressive power as deterministic max-automata.

Theorem. WMSO+R has the same expressive power as deterministic min-automata.

Theorem. WMSO+U+R has the same expressive power as boolean combinations of min- and max-automata.

Equivalently: Nesting the quantifiers U and R does not contribute anything to the expressive power of WMSO.

Emptiness of min-automata

Emptiness of min-automata

Theorem. Emptiness of min-automata is decidable.

Emptiness of min-automata

Theorem. Emptiness of min-automata is decidable.

1st proof. min-automata are a special case of ωBS-automata (Bojańczyk, Colcombet [06]), so emptiness is decidable. This gives bad complexity, however.

Emptiness of min-automata

Theorem. Emptiness of min-automata is decidable.

1st proof. min-automata are a special case of ωBS-automata (Bojańczyk, Colcombet [06]), so emptiness is decidable. This gives bad complexity, however.

2nd proof. Reduction to the limitedness problem for distance-automata.
Gives PSPACE algorithm, which is optimal.

Emptiness of min-automata

Theorem. Emptiness of min-automata is decidable.

1st proof. min-automata are a special case of ωBS-automata (Bojańczyk, Colcombet [06]), so emptiness is decidable. This gives bad complexity, however.

2nd proof. Reduction to the limitedness problem for distance-automata. Gives PSPACE algorithm, which is optimal.

Theorem. Emptiness of max-automata is decidable.

Emptiness of min-automata

Theorem. Emptiness of min-automata is decidable.

1st proof. min-automata are a special case of ωBS-automata (Bojańczyk, Colcombet [06]), so emptiness is decidable. This gives bad complexity, however.

2nd proof. Reduction to the limitedness problem for distance-automata. Gives PSPACE algorithm, which is optimal.

Theorem. Emptiness of max-automata is decidable.

Theorem. Emptiness of a boolean combination of min- and max-automata is decidable.

Simplyfying the min-automata model

Simplyfying the min-automata model

- Instructions $c:=0$, $c:=d$ can be implemented into the model, as in the example

Simplyfying the min-automata model

- Instructions $c{:=}0$, $c{:=}d$ can be implemented into the model, as in the example
- Can introduce the undefined counter values $\top$
  this can be eliminated by storing in the states the info about which counters are defined

Simplyfying the min-automata model

- Instructions *c:=0, c:=d* can be implemented into the model, as in the example
- Can introduce the undefined counter values $\top$
  this can be eliminated by storing in the states the info about which counters are defined
- Can introduce the counter value ∞
  the difference between $\top$ and ∞ is that $lim\ \infty = \infty$ while $lim\ \top \neq \infty$

Simplyfying the min-automata model

- Instructions *c:=0, c:=d* can be implemented into the model, as in the example
- Can introduce the undefined counter values $\top$
  this can be eliminated by storing in the states the info about which counters are defined
- Can introduce the counter value ∞
  the difference between $\top$ and ∞ is that $\lim \infty = \infty$ while $\lim \top \neq \infty$
- Can introduce matrix operations on counters, which stems from the semiring structure on $\{0,1,2,\ldots, \infty, \top\}$, where *min* with respect to $0<1<2<\ldots< \infty<\top$ is addition and + is multiplication

Simplyfying the min-automata model

- Instructions *c:=0, c:=d* can be implemented into the model, as in the example
- Can introduce the undefined counter values $\top$
  this can be eliminated by storing in the states the info about which counters are defined
- Can introduce the counter value ∞
  the difference between $\top$ and ∞ is that $\lim \infty = \infty$ while $\lim \top \neq \infty$
- Can introduce matrix operations on counters, which stems from the semiring structure on $\{0,1,2,\ldots, \infty, \top\}$, where *min* with respect to $0<1<2<\ldots< \infty<\top$ is addition and + is multiplication

In Example 1, $c:=c+1$ can be written as:

Simplyfying the min-automata model

- Instructions *c:=0, c:=d* can be implemented into the model, as in the example
- Can introduce the undefined counter values $\top$
  this can be eliminated by storing in the states the info about which counters are defined
- Can introduce the counter value ∞
  the difference between $\top$ and ∞ is that $\quad lim\ \infty = \infty \quad$ while $\quad lim\ \top \neq \infty$
- Can introduce matrix operations on counters, which stems from the semiring structure on $\{0,1,2,\ldots, \infty, \top\}$, where *min* with respect to $0<1<2<\ldots<\infty<\top$ is addition and + is multiplication

In Example 1, *c:=c+1* can be written as:

$$\begin{pmatrix} c & d & z \end{pmatrix} \quad := \quad \begin{pmatrix} c & d & z \end{pmatrix} \cdot \begin{pmatrix} 1 & \top & \top \\ \top & 0 & \top \\ \top & \top & 0 \end{pmatrix}$$

Simplyfying the min-automata model

- Instructions *c:=0, c:=d* can be implemented into the model, as in the example
- Can introduce the undefined counter values $\top$
  this can be eliminated by storing in the states the info about which counters are defined
- Can introduce the counter value ∞
  the difference between $\top$ and ∞ is that $lim\ \infty = \infty$ while $lim\ \top \neq \infty$
- Can introduce matrix operations on counters, which stems from the semiring structure on $\{0,1,2,\ldots, \infty, \top\}$, where *min* with respect to $0<1<2<\ldots< \infty<\top$ is addition and + is multiplication

In Example 1, *c:=c+1* can be written as:

$$\begin{pmatrix} c & d & z \end{pmatrix} := \begin{pmatrix} c & d & z \end{pmatrix} \cdot \begin{pmatrix} 1 & \top & \top \\ \top & 0 & \top \\ \top & \top & 0 \end{pmatrix} = \begin{pmatrix} c+1 & d & z \end{pmatrix}$$

Simplyfying the min-automata model

- Instructions *c:=0, c:=d* can be implemented into the model, as in the example
- Can introduce the undefined counter values $\top$
  this can be eliminated by storing in the states the info about which counters are defined
- Can introduce the counter value ∞
  the difference between $\top$ and ∞ is that $lim\ \infty = \infty$ while $lim\ \top \neq \infty$
- Can introduce matrix operations on counters, which stems from the semiring structure on $\{0,1,2,\ldots, \infty, \top\}$, where *min* with respect to $0<1<2<\ldots< \infty<\top$ is addition and + is multiplication

In Example 1, *c:=c+1* can be written as:

$$\begin{pmatrix} c & d & z \end{pmatrix} := \begin{pmatrix} c & d & z \end{pmatrix} \cdot \begin{pmatrix} 1 & \top & \top \\ \top & 0 & \top \\ \top & \top & 0 \end{pmatrix} = \begin{pmatrix} c+1 & d & z \end{pmatrix}$$

d:=min(*c,c*)*; c:=min*(*z,z*) can be written as:

Simplyfying the min-automata model

- Instructions *c:=0, c:=d* can be implemented into the model, as in the example
- Can introduce the undefined counter values $\top$
  this can be eliminated by storing in the states the info about which counters are defined
- Can introduce the counter value ∞
  the difference between $\top$ and ∞ is that *lim* $\infty = \infty$ while *lim* $\top \neq \infty$
- Can introduce matrix operations on counters, which stems from the semiring structure on $\{0,1,2,\ldots,\infty,\top\}$, where *min* with respect to $0<1<2<\ldots<\infty<\top$ is addition and + is multiplication

In Example 1, *c:=c+1* can be written as:

$$\begin{pmatrix} c & d & z \end{pmatrix} := \begin{pmatrix} c & d & z \end{pmatrix} \cdot \begin{pmatrix} 1 & \top & \top \\ \top & 0 & \top \\ \top & \top & 0 \end{pmatrix} = \begin{pmatrix} c+1 & d & z \end{pmatrix}$$

d:=min(*c,c*)*; c:=min*(*z,z*) can be written as:

$$\begin{pmatrix} c & d & z \end{pmatrix} := \begin{pmatrix} c & d & z \end{pmatrix} \cdot \begin{pmatrix} \top & 0 & \top \\ \top & \top & \top \\ 0 & \top & 0 \end{pmatrix}$$

Simplyfying the min-automata model

- Instructions *c:=0, c:=d* can be implemented into the model, as in the example
- Can introduce the undefined counter values $\top$
  this can be eliminated by storing in the states the info about which counters are defined
- Can introduce the counter value ∞
  the difference between $\top$ and ∞ is that *lim* $\infty = \infty$ while *lim* $\top \neq \infty$
- Can introduce matrix operations on counters, which stems from the semiring structure on $\{0,1,2,\ldots, \infty, \top\}$, where *min* with respect to $0<1<2<\ldots< \infty<\top$ is addition and + is multiplication

In Example 1, *c:=c+1* can be written as:

$$\begin{pmatrix} c & d & z \end{pmatrix} := \begin{pmatrix} c & d & z \end{pmatrix} \cdot \begin{pmatrix} 1 & \top & \top \\ \top & 0 & \top \\ \top & \top & 0 \end{pmatrix} = \begin{pmatrix} c+1 & d & z \end{pmatrix}$$

d:=min(c,c); c:=min(z,z) can be written as:

$$\begin{pmatrix} c & d & z \end{pmatrix} := \begin{pmatrix} c & d & z \end{pmatrix} \cdot \begin{pmatrix} \top & 0 & \top \\ \top & \top & \top \\ 0 & \top & 0 \end{pmatrix} = \begin{pmatrix} z & c & z \end{pmatrix}$$

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts a's on odd positions.

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts a's on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c:=c+1$

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts a's on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c:=c+1$
- saw a in state q_1 – go to q_0

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c{:=}c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c:=c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c := c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c:=c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1)=(0, \top)$.

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c := c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1) = (0, \top)$.

$$a : \qquad \begin{pmatrix} c_0 & c_1 \end{pmatrix} \quad := \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 1 & \top \end{pmatrix}.$$

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c{:=}c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1)=(0, \top)$.

$$
\begin{aligned}
a: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} &:= \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 1 & \top \end{pmatrix}. \\
b: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} &:= \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 0 & \top \end{pmatrix}.
\end{aligned}
$$

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- -saw a in state q_0 – go to q_1; $c{:=}c+1$
- -saw a in state q_1 – go to q_0
- -saw b in state q_0 – go to q_1
- -saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1)=(0, \top)$.

$$a: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 1 & \top \end{pmatrix}.$$

$$b: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 0 & \top \end{pmatrix}.$$

a a a b b b a a b...

c_0

c_1

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c{:=}c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1)=(0, \top)$.

$$a: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 1 & \top \end{pmatrix}.$$

$$b: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 0 & \top \end{pmatrix}.$$

a a a b b b a a b...

c_0	0
c_1	$\top$

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c:=c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1)=(0, \top)$.

$$a: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 1 & \top \end{pmatrix}.$$

$$b: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 0 & \top \end{pmatrix}.$$

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

$$\begin{array}{lll} c_0 & 0 & \top \\ c_1 & \top & 1 \end{array}$$

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts a's on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c:=c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1)=(0, \top)$.

$$a: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 1 & \top \end{pmatrix}.$$

$$b: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 0 & \top \end{pmatrix}.$$

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

c_0 0 $\top$ 1

c_1 $\top$ 1 $\top$

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c:=c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1)=(0, \top)$.

$$a: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 1 & \top \end{pmatrix}.$$

$$b: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 0 & \top \end{pmatrix}.$$

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

c_0 0 $\top$ 1 $\top$

c_1 $\top$ 1 $\top$ 2

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c:=c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1)=(0, \top)$.

$$a: \qquad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 1 & \top \end{pmatrix}.$$

$$b: \qquad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 0 & \top \end{pmatrix}.$$

a a a b b b a a b...

c_0	0	$\top$	1	$\top$	2
c_1	$\top$	1	$\top$	2	$\top$

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c:=c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1)=(0, \top)$.

$$
\begin{aligned}
a: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} &:= \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 1 & \top \end{pmatrix}. \\
b: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} &:= \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 0 & \top \end{pmatrix}.
\end{aligned}
$$

$$
\begin{array}{cccccccccc}
 & a & a & a & b & b & b & a & a & b\ldots \\
c_0 & 0 & \top & 1 & \top & 2 & \top & & & \\
c_1 & \top & 1 & \top & 2 & \top & 2 & & &
\end{array}
$$

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c:=c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1)=(0, \top)$.

$$a: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 1 & \top \end{pmatrix}.$$

$$b: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 0 & \top \end{pmatrix}.$$

a a a b b b a a b...

c_0	0	$\top$	1	$\top$	2	$\top$	2		
c_1	$\top$	1	$\top$	2	$\top$	2	$\top$		

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c := c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1) = (0, \top)$.

$$a: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 1 & \top \end{pmatrix}.$$

$$b: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 0 & \top \end{pmatrix}.$$

	a	a	a	b	b	b	a	a	b...
c_0	0	$\top$	1	$\top$	2	$\top$	2	$\top$	
c_1	$\top$	1	$\top$	2	$\top$	2	$\top$	3	

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c{:}=c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1)=(0, \top)$.

$$a: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 1 & \top \end{pmatrix}.$$

$$b: \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} := \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 0 & \top \end{pmatrix}.$$

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

$$\begin{array}{llllllllll} c_0 & 0 & \top & 1 & \top & 2 & \top & 2 & \top & 3 \\ c_1 & \top & 1 & \top & 2 & \top & 2 & \top & 3 & \top \end{array}$$

Theorem. Min-automata are equivalent to min-automata in matrix form, with one state.

Proof. We eliminate states as in the following example.

Example. Min-automaton which counts *a*'s on odd positions.

States: q_0, q_1, one counter c.

Transitions:

- saw a in state q_0 – go to q_1; $c:=c+1$
- saw a in state q_1 – go to q_0
- saw b in state q_0 – go to q_1
- saw b in state q_1 – go to q_0

Min-automaton in matrix form with one state and two counters: c_0, c_1.

The initial counter valuation is $(c_0, c_1)=(0, \top)$.

$$a: \qquad \begin{pmatrix} c_0 & c_1 \end{pmatrix} \quad := \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 1 & \top \end{pmatrix}.$$

$$b: \qquad \begin{pmatrix} c_0 & c_1 \end{pmatrix} \quad := \quad \begin{pmatrix} c_0 & c_1 \end{pmatrix} \cdot \begin{pmatrix} \top & 0 \\ 0 & \top \end{pmatrix}.$$

$$\begin{array}{lccccccccc} & & a & a & a & b & b & b & a & a\ b\ldots \\ c_0 & 0 & \top & 1 & \top & 2 & \top & 2 & \top & 3\ \top \\ c_1 & \top & 1 & \top & 2 & \top & 2 & \top & 3 & \top\ 3 \end{array}$$

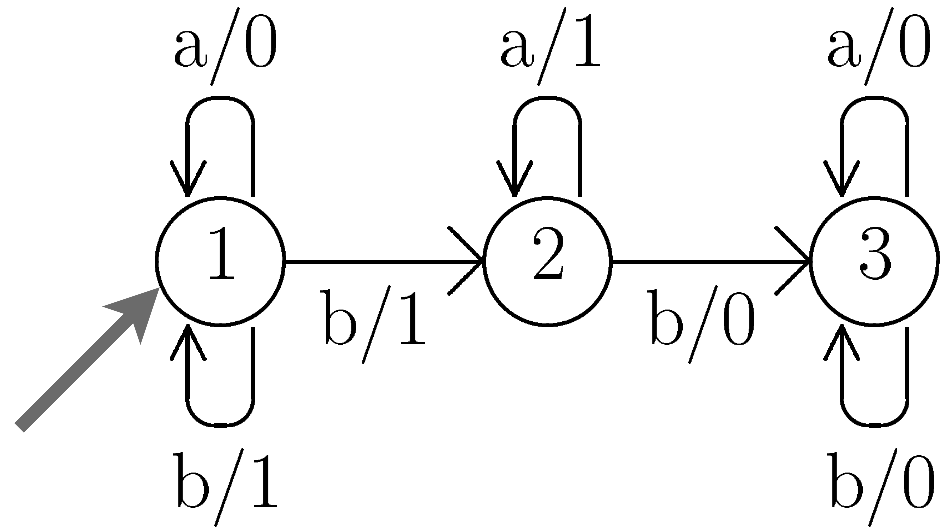


Figure 1: Example of a distance automaton A.

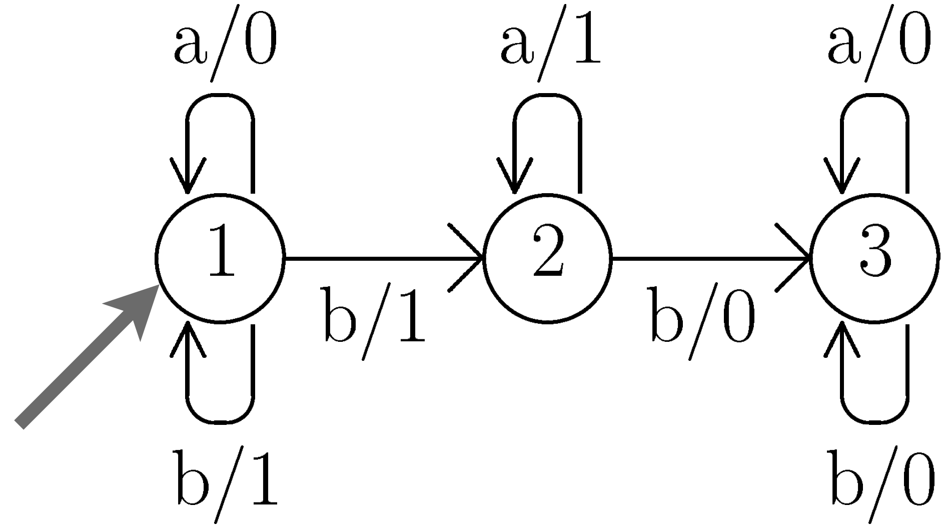


Figure 1: Example of a distance automaton A.

Example of a min-automaton *A.*

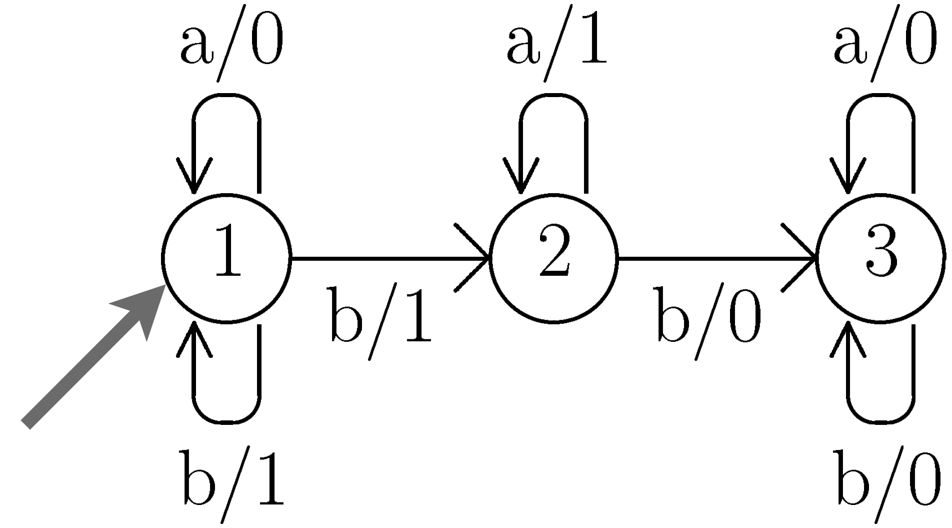


Figure 1: Example of a distance automaton A.

Example of a min-automaton *A.*

$$a: \begin{bmatrix} 0 & \top & \top \\ \top & 1 & \top \\ \top & \top & 0 \end{bmatrix} \qquad b: \begin{bmatrix} 1 & 1 & \top \\ \top & \top & 0 \\ \top & \top & 0 \end{bmatrix}$$

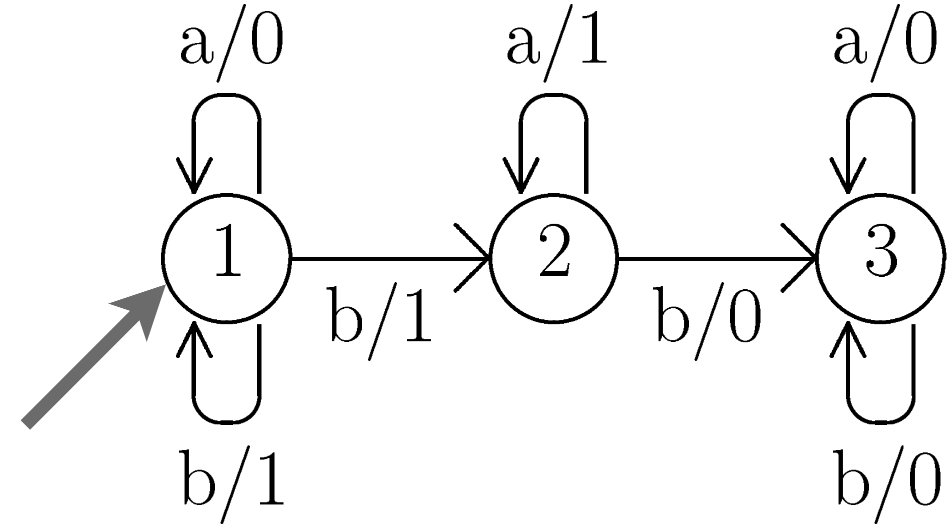


Figure 1: Example of a distance automaton A.

Example of a min-automaton *A.*

$$a: \begin{array}{|ccc|} 0 & \top & \top \\ \top & 1 & \top \\ \top & \top & 0 \end{array} \qquad b: \begin{array}{|ccc|} 1 & 1 & \top \\ \top & \top & 0 \\ \top & \top & 0 \end{array}$$

$$abb: \begin{array}{|ccc|} 2 & 2 & 1 \\ \top & \top & 0 \\ \top & \top & 0 \end{array}$$

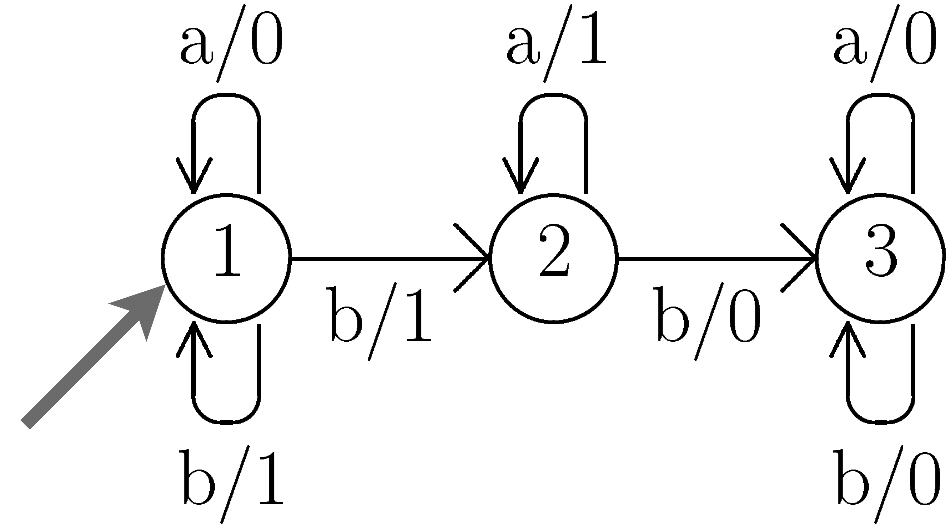


Figure 1: Example of a distance automaton A.

Example of a min-automaton *A*.

$$a: \quad \begin{bmatrix} 0 & \top & \top \\ \top & 1 & \top \\ \top & \top & 0 \end{bmatrix} \qquad b: \quad \begin{bmatrix} 1 & 1 & \top \\ \top & \top & 0 \\ \top & \top & 0 \end{bmatrix}$$

$$abb: \quad \begin{bmatrix} 2 & 2 & 1 \\ \top & \top & 0 \\ \top & \top & 0 \end{bmatrix}$$

Initial valuation: $\begin{bmatrix} 0 & \top & \top \end{bmatrix}$

Acceptance condition: $\neg \ \mathrm{liminf}(c_3) = \infty$

The tropical semiring

The tropical semiring

$$T = \{0, 1, 2, \ldots, \infty, \top\}$$

The tropical semiring

$$T = \{0,1,2,\ldots, \infty, \top\}$$

with operations +, min

ordered by $0<1<2<\ldots<\infty<\top$

where $\top + x = x + \top = \top$

The tropical semiring

$$T = \{0,1,2,\ldots,\ \infty,\ \top\}$$

with operations +, min

ordered by $0<1<2<\ldots<\infty<\top$

where $\top + x = x + \top = \top$

$$T_n = \{0,1,2,\ldots,n,\ \infty,\ \top\}$$

where $n+1=\infty$

The tropical semiring

$$T = \{0,1,2,\ldots,\ \infty,\ \top\}$$

with operations +, min

ordered by $0<1<2<\ldots<\infty<\top$

where $\top + x = x + \top = \top$

$$T_n = \{0,1,2,\ldots,n,\ \infty,\ \top\}$$

where $n+1=\infty$

$$\pi_n : T \to T_n$$

maps $n+1$, $n+2$, ... to ∞

is a homomorphism of semirings

The tropical semiring

$$T = \{0,1,2,\ldots, \infty, \top\}$$

with operations +, min

ordered by $0<1<2<\ldots<\infty<\top$

where $\top + x = x + \top = \top$

$$T_n = \{0,1,2,\ldots,n, \infty, \top\}$$

where $n+1=\infty$

$$\pi_n : T \to T_n$$

$$\pi_n : T_m \to T_n \text{ for } m>n$$

maps $n+1$, $n+2$, ... to ∞

is a homomorphism of semirings

The tropical semiring

$$T = \{0,1,2,\ldots,\ \infty,\ \top\}$$

with operations +, min

ordered by $0<1<2<\ldots<\infty<\top$

where $\top + x = x + \top = \top$

$$T_n = \{0,1,2,\ldots,n,\ \infty,\ \top\}$$

where $n+1=\infty$

$$\pi_n : T \to T_n$$

$$\pi_n : T_m \to T_n \text{ for } m>n$$

maps $n+1$, $n+2$, ... to ∞

is a homomorphism of semirings

The tropical semiring

$$T = \{0, 1, 2, \ldots, \infty, \top\}$$

with operations +, min

ordered by $0 < 1 < 2 < \ldots < \infty < \top$

where $\top + x = x + \top = \top$

$M_k T$ – *k by k* matrices over T

with matrix multiplication

$$T_n = \{0, 1, 2, \ldots, n, \infty, \top\}$$

where $n+1=\infty$

$M_k T_n$ – *k by k* matrices over T_n

with matrix multiplication

$$\pi_n : T \to T_n$$

$$\pi_n : T_m \to T_n \text{ for } m > n$$

maps $n+1$, $n+2$, ... to ∞

is a homomorphism of semirings

$$\pi_n : M_k T \to M_k T_n$$

$$\pi_n : T_m \to T_n \text{ for } m > n$$

maps $n+1$, $n+2$, ... to ∞

is a homomorphism of semirings

The tropical semiring

$T = \{0,1,2,\ldots, \infty, \top\}$
with operations +, min
ordered by $0<1<2<\ldots<\infty<\top$
where $\top + x = x + \top = \top$

$M_k T$ – *k by k* matrices over T
with matrix multiplication

$T_n = \{0,1,2,\ldots,n, \infty, \top\}$
where $n+1=\infty$

$M_k T_n$ – *k by k* matrices over T_n
with matrix multiplication

$\pi_n : T \to T_n$

$\pi_n : T_m \to T_n$ for $m>n$

maps $n+1$, $n+2$, ... to ∞

is a homomorphism of semirings

$\pi_n : M_k T \to M_k T_n$

$\pi_n : T_m \to T_n$ for $m>n$

maps $n+1$, $n+2$, ... to ∞

is a homomorphism of semirings

$$\begin{array}{|ccc|} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{array} \xrightarrow{\pi_6} \begin{array}{|ccc|} 3 & \infty & 1 \\ 0 & \infty & 1 \\ 2 & \infty & \infty \end{array} \xrightarrow{\pi_2} \begin{array}{|ccc|} \infty & \infty & 1 \\ 0 & \infty & 1 \\ \infty & \infty & \infty \end{array}$$

Profinite monoid

Profinite monoid

$$S_0 \longleftarrow S_1 \longleftarrow S_2 \longleftarrow S_3 \;\cdots\; S_7 \;\cdots\; S_{32} \;\cdots\; S_{1000} \;\cdots$$

Profinite monoid

$$S_0 \leftarrow S_1 \leftarrow S_2 \leftarrow S_3 \dashleftarrow S_7 \dashleftarrow S_{32} \dashleftarrow S_{1000} \dashleftarrow \cdots$$

preserve multiplication

Profinite monoid

$$M_3 T_0 \leftarrow M_3 T_1 \leftarrow M_3 T_2 \leftarrow M_3 T_3 \leftarrow \cdots$$

$$S_0 \leftarrow S_1 \leftarrow S_2 \leftarrow S_3 \leftarrow \cdots\ S_7 \leftarrow \cdots\ S_{32} \leftarrow \cdots\ S_{1000} \leftarrow \cdots$$

preserve multiplication

Profinite monoid

$$S_0 \longleftarrow S_1 \longleftarrow S_2 \longleftarrow S_3 \leftarrow\cdots\; S_7 \leftarrow\cdots\; S_{32} \leftarrow\cdots\; S_{1000} \leftarrow\cdots\cdots$$

preserve multiplication

Profinite monoid

3	∞	1
0	∞	1
2	∞	∞

$$S_0 \leftarrow S_1 \leftarrow S_2 \leftarrow S_3 \leftarrow \cdots S_7 \leftarrow \cdots S_{32} \leftarrow \cdots S_{1000} \leftarrow \cdots\cdots$$

preserve multiplication

Profinite monoid

$$\begin{bmatrix} \infty & \infty & 1 \\ 0 & \infty & 1 \\ 2 & \infty & \infty \end{bmatrix} \quad \begin{bmatrix} 3 & \infty & 1 \\ 0 & \infty & 1 \\ 2 & \infty & \infty \end{bmatrix}$$

$$S_0 \leftarrow S_1 \leftarrow S_2 \leftarrow S_3 \leftarrow \cdots S_7 \leftarrow \cdots S_{32} \leftarrow \cdots S_{1000} \leftarrow \cdots$$

preserve multiplication

Profinite monoid

$$\begin{matrix} \infty & \infty & 1 \\ 0 & \infty & 1 \\ \infty & \infty & \infty \end{matrix} \qquad \begin{matrix} \infty & \infty & 1 \\ 0 & \infty & 1 \\ 2 & \infty & \infty \end{matrix} \qquad \begin{matrix} 3 & \infty & 1 \\ 0 & \infty & 1 \\ 2 & \infty & \infty \end{matrix}$$

$$S_0 \leftarrow S_1 \leftarrow S_2 \leftarrow S_3 \dashleftarrow S_7 \dashleftarrow S_{32} \dashleftarrow S_{1000} \dashleftarrow \cdots$$

preserve multiplication

Profinite monoid

$$\begin{bmatrix} \infty & \infty & \infty \\ 0 & \infty & \infty \\ \infty & \infty & \infty \end{bmatrix} \quad \begin{bmatrix} \infty & \infty & 1 \\ 0 & \infty & 1 \\ \infty & \infty & \infty \end{bmatrix} \quad \begin{bmatrix} \infty & \infty & 1 \\ 0 & \infty & 1 \\ 2 & \infty & \infty \end{bmatrix} \quad \begin{bmatrix} 3 & \infty & 1 \\ 0 & \infty & 1 \\ 2 & \infty & \infty \end{bmatrix}$$

$$S_0 \leftarrow S_1 \leftarrow S_2 \leftarrow S_3 \leftarrow \cdots\; S_7 \leftarrow \cdots\; S_{32} \leftarrow \cdots\; S_{1000} \leftarrow \cdots$$

preserve multiplication

Profinite monoid

$$S_0 \leftarrow S_1 \leftarrow S_2 \leftarrow S_3 \leftarrow \cdots S_7 \leftarrow \cdots S_{32} \leftarrow \cdots S_{1000} \leftarrow \cdots\cdots$$

preserve multiplication

Profinite monoid

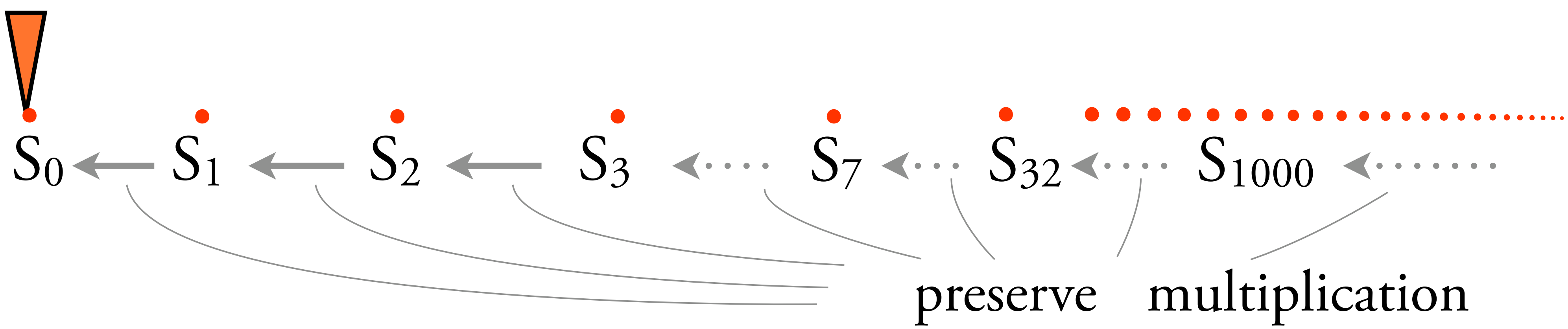
$$S_0 \leftarrow S_1 \leftarrow S_2 \leftarrow S_3 \leftarrow \cdots S_7 \leftarrow \cdots S_{32} \leftarrow \cdots S_{1000} \leftarrow \cdots$$

preserve multiplication

Profinite monoid

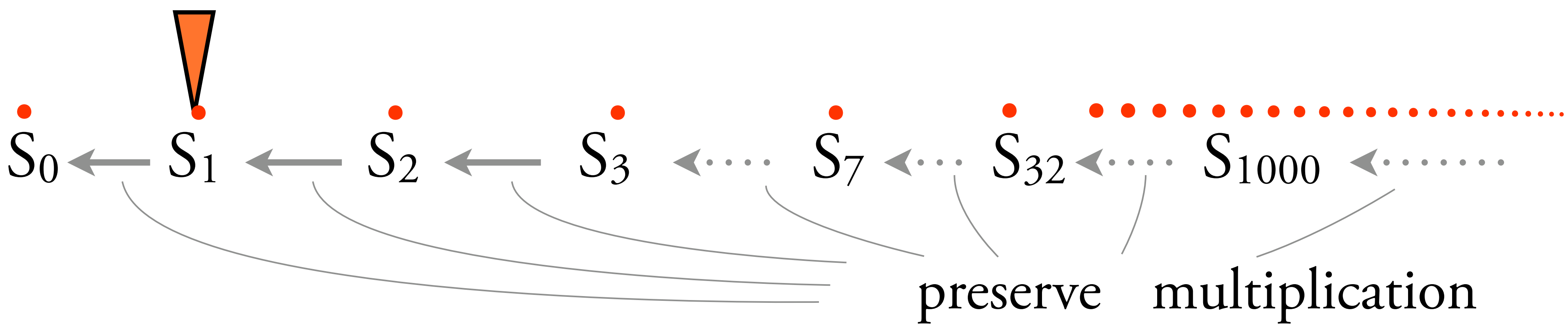
0 -approximation:

$$\begin{matrix} \infty & \infty & \infty \\ 0 & \infty & \infty \\ \infty & \infty & \infty \end{matrix}$$



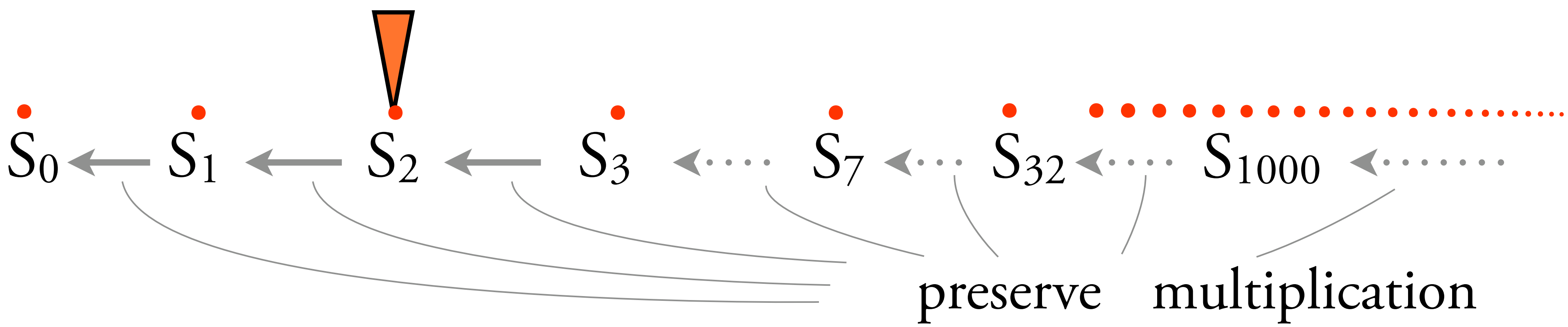
Profinite monoid

1 -approximation: $\begin{array}{ccc} \infty & \infty & 1 \\ 0 & \infty & 1 \\ \infty & \infty & \infty \end{array}$



Profinite monoid

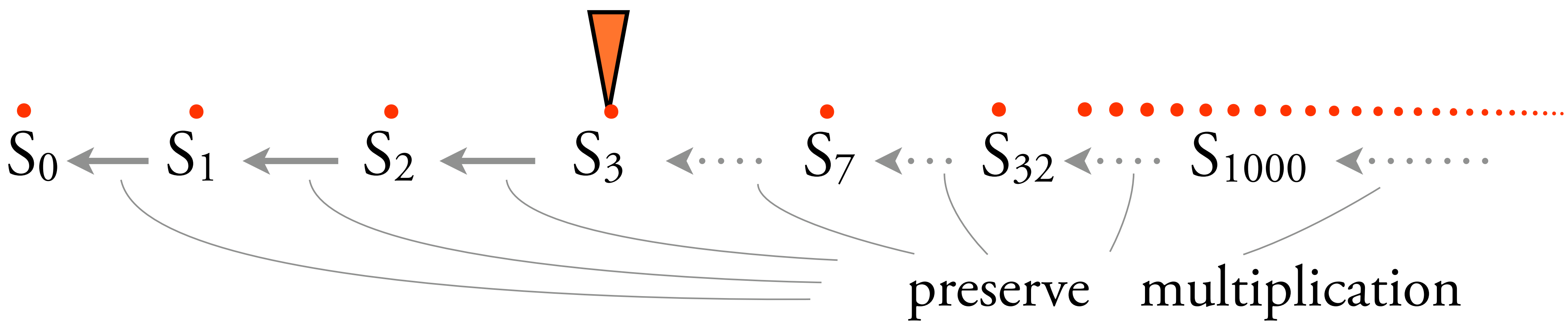
2 -approximation: $\begin{matrix} \infty & \infty & 1 \\ 0 & \infty & 1 \\ 2 & \infty & \infty \end{matrix}$



Profinite monoid

3 -approximation:

3	∞	1
0	∞	1
2	∞	∞



Profinite monoid

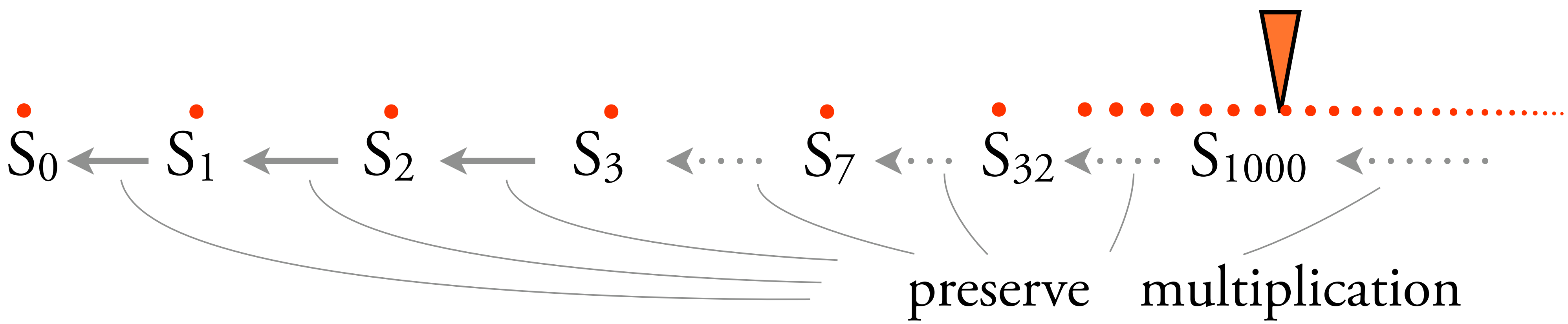
7 -approximation: $\begin{array}{ccc} 3 & \infty & 1 \\ 0 & \infty & 1 \\ 2 & 7 & \infty \end{array}$

$$S_0 \leftarrow S_1 \leftarrow S_2 \leftarrow S_3 \leftarrow \cdots S_7 \leftarrow \cdots S_{32} \leftarrow \cdots S_{1000} \leftarrow \cdots$$

preserve multiplication

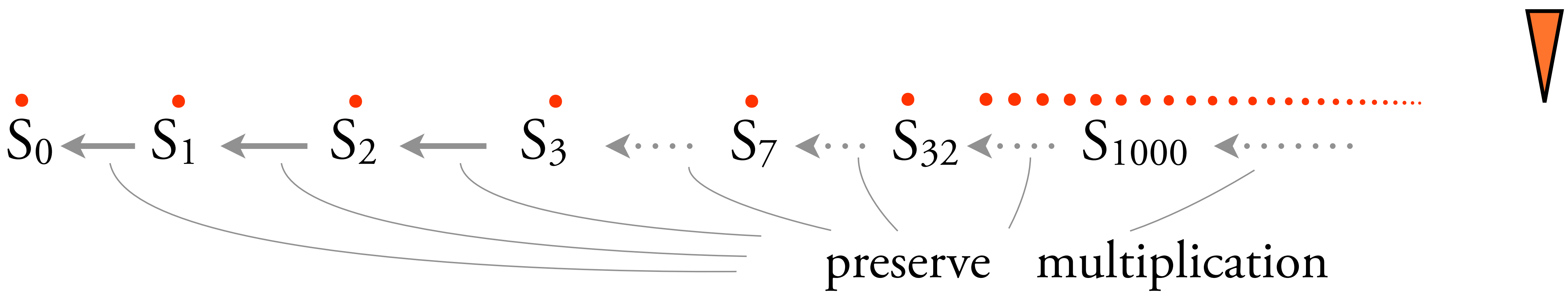
Profinite monoid

1000 -approximation: $\begin{array}{ccc} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{array}$



Profinite monoid

3	32	1
0	11	1
2	7	∞



Profinite monoid

3	32	1
0	11	1
2	7	∞

$$S_0 \leftarrow S_1 \leftarrow S_2 \leftarrow S_3 \leftarrow \cdots S_7 \leftarrow \cdots S_{32} \leftarrow \cdots S_{1000} \leftarrow \cdots$$

preserve multiplication

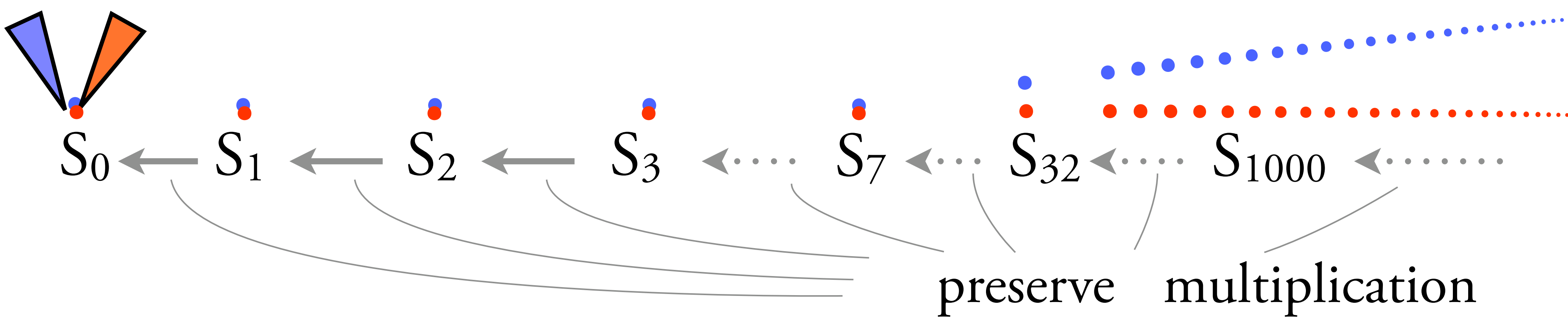
Metric

Two elements are close if only an approx. with high threshold can distinguish them

Profinite monoid

3	50	1
0	11	1
2	7	∞

3	32	1
0	11	1
2	7	∞

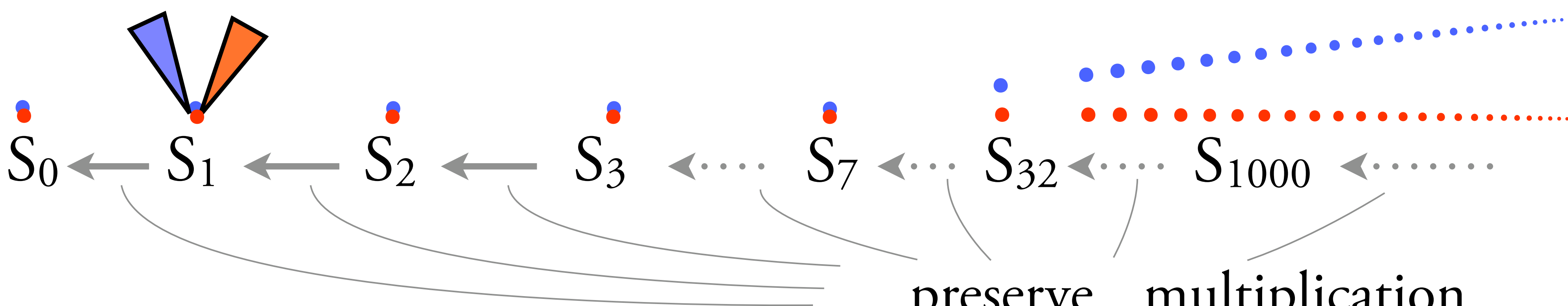


Metric

Two elements are close if only an approx. with high threshold can distinguish them

Profinite monoid

$$\begin{array}{ccc} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{array} \qquad \begin{array}{ccc} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{array}$$



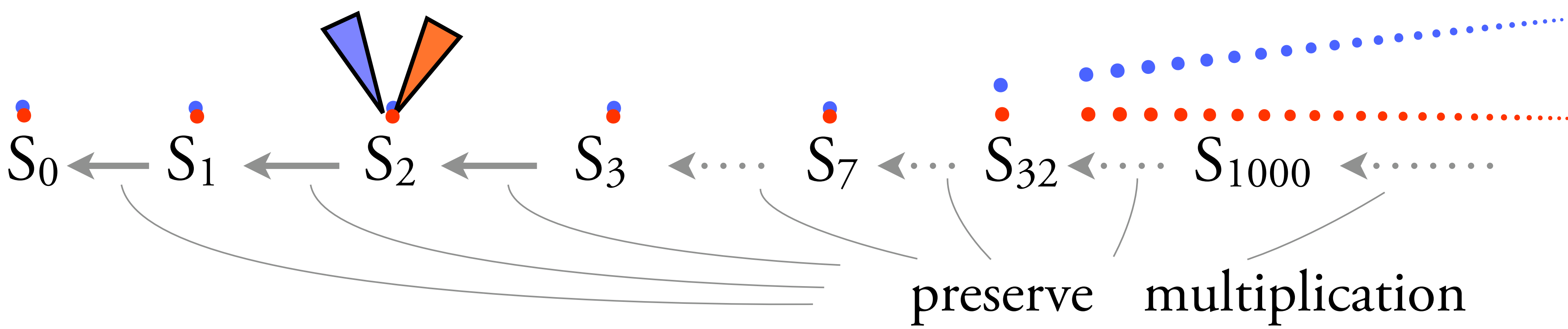
Metric

Two elements are close if only an approx. with high threshold can distinguish them

Profinite monoid

3	50	1
0	11	1
2	7	∞

3	32	1
0	11	1
2	7	∞

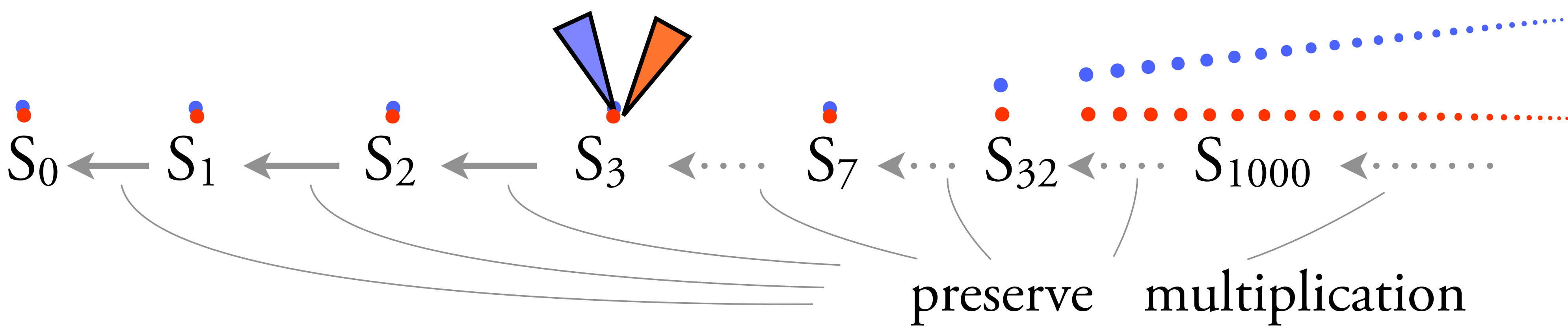


Metric

Two elements are close if only an approx. with high threshold can distinguish them

Profinite monoid

$$\begin{matrix} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{matrix} \qquad \begin{matrix} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{matrix}$$



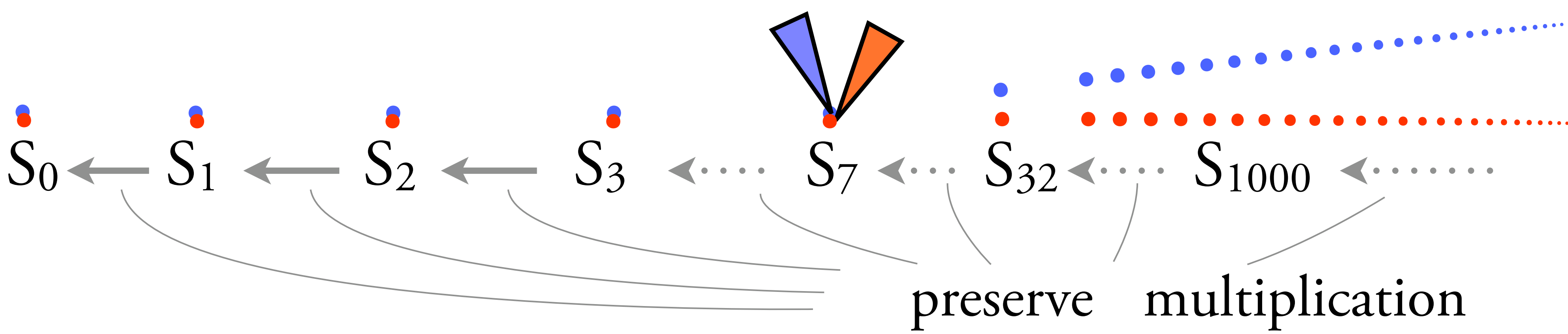
Metric

Two elements are close if only an approx. with high threshold can distinguish them

Profinite monoid

3	50	1
0	11	1
2	7	∞

3	32	1
0	11	1
2	7	∞



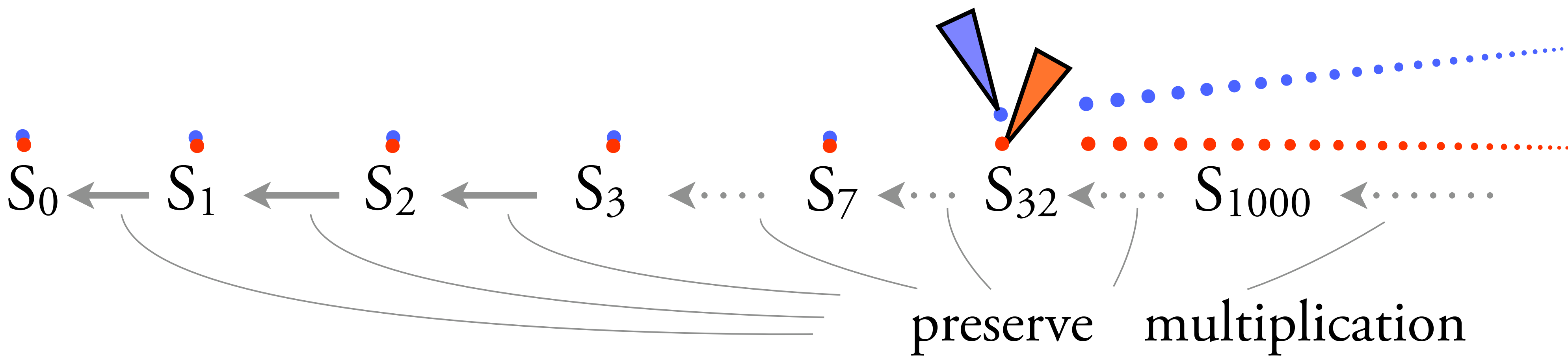
Metric

Two elements are close if only an approx. with high threshold can distinguish them

Profinite monoid

3	50	1
0	11	1
2	7	∞

3	32	1
0	11	1
2	7	∞

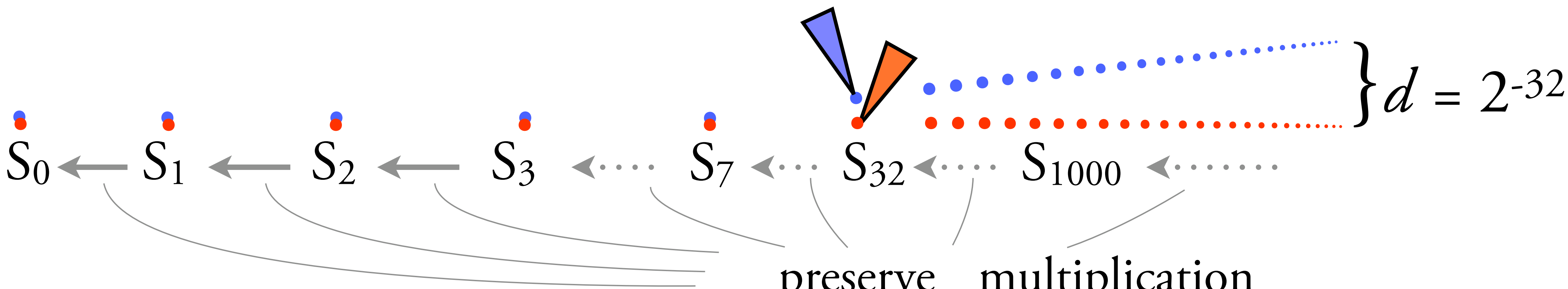


Metric

Two elements are close if only an approx. with high threshold can distinguish them

Profinite monoid

$$\begin{matrix} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{matrix} \qquad \begin{matrix} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{matrix}$$

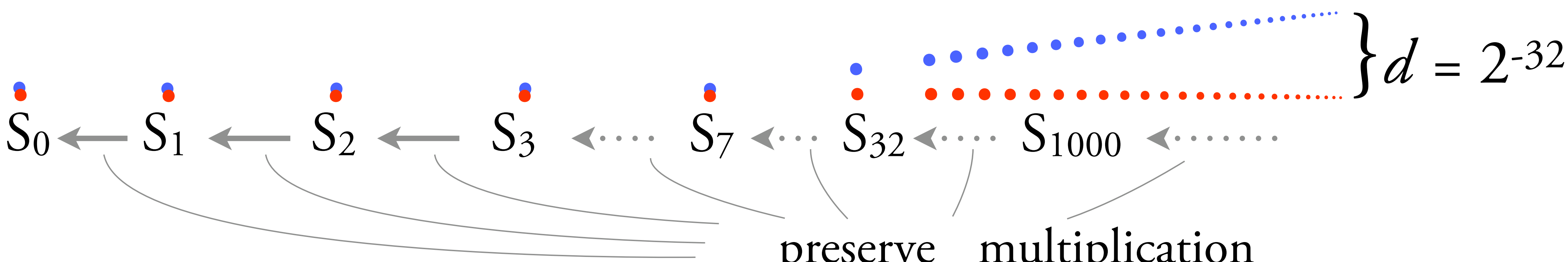


Metric

Two elements are close if only an approx. with high threshold can distinguish them

Profinite monoid

$$\begin{matrix} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{matrix} \qquad\qquad \begin{matrix} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{matrix}$$

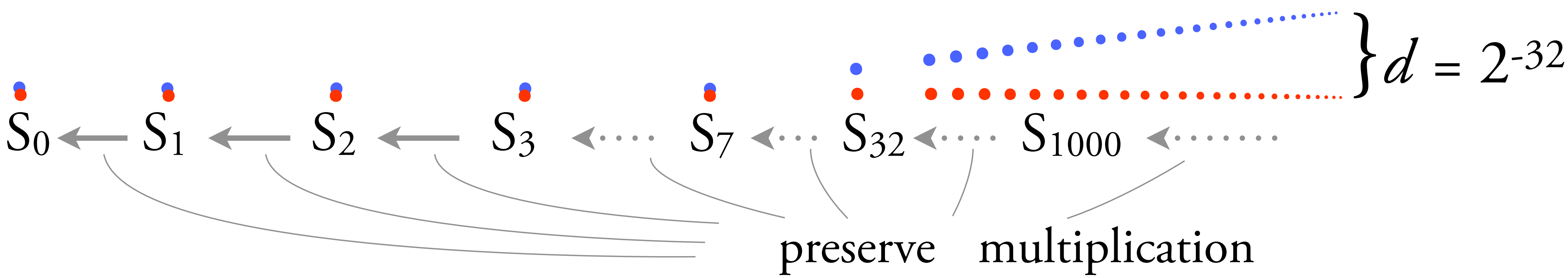


Metric

Two elements are close if only an approx. with high threshold can distinguish them

Profinite monoid

$$\begin{array}{ccc} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{array} \qquad \begin{array}{ccc} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{array}$$



Metric

Two elements are close if only an approx. with high threshold can distinguish them

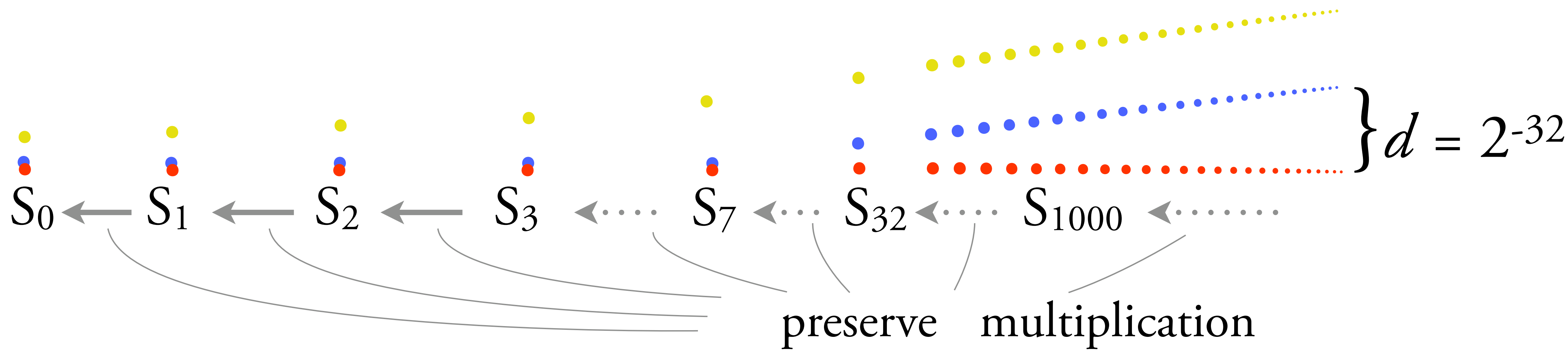
Multiplication

The n-approximation of $x \cdot y$ is the product of their n-approximations.

we again obtain a sequence consistent with the mappings

Profinite monoid

$$\begin{bmatrix} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{bmatrix} \cdot \begin{bmatrix} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{bmatrix} = \begin{bmatrix} 3 & 8 & 4 \\ 3 & 8 & 4 \\ 5 & 18 & 3 \end{bmatrix}$$



Metric

Two elements are close if only an approx. with high threshold can distinguish them

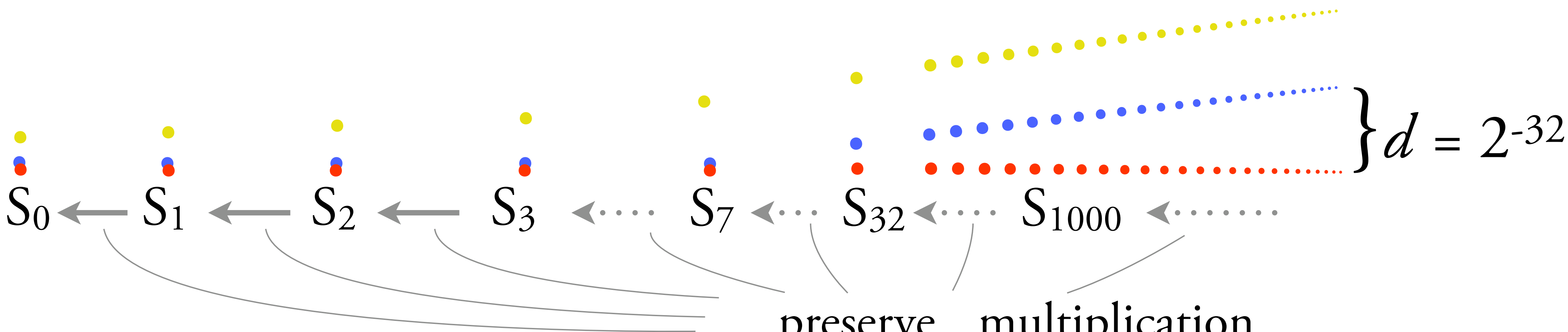
Multiplication

The n-approximation of $x{\cdot}y$ is the product of their n-approximations.

we again obtain a sequence consistent with the mappings

Profinite monoid

$$\begin{bmatrix} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{bmatrix} \cdot \begin{bmatrix} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{bmatrix} = \begin{bmatrix} 3 & 8 & 4 \\ 3 & 8 & 4 \\ 5 & 18 & 3 \end{bmatrix}$$



Metric

Two elements are close if only an approx. with high threshold can distinguish them

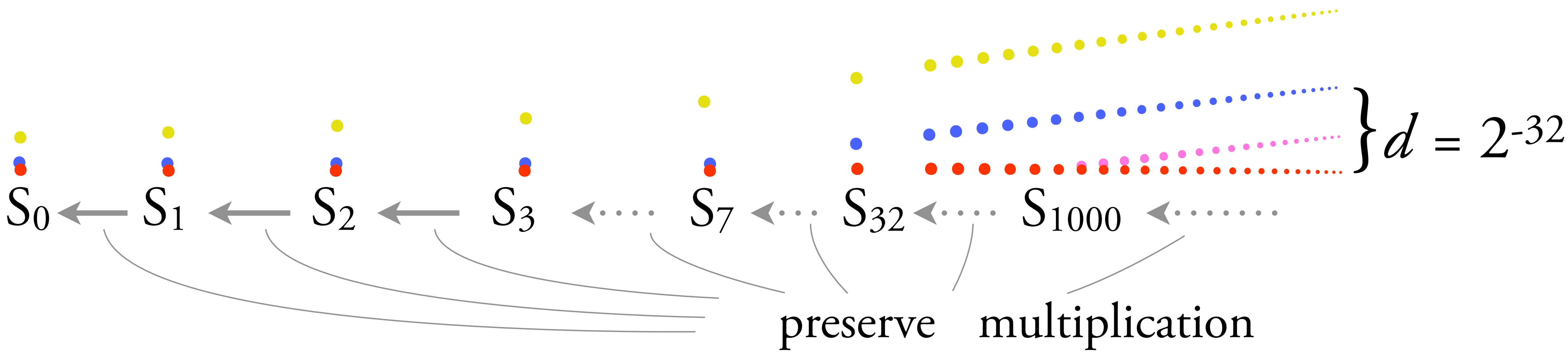
Multiplication *is continuous*

The n-approximation of $x{\cdot}y$ is the product of their n-approximations.

we again obtain a sequence consistent with the mappings

Profinite monoid

$$\begin{pmatrix} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} \cdot \begin{pmatrix} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} = \begin{pmatrix} 3 & 8 & 4 \\ 3 & 8 & 4 \\ 5 & 18 & 3 \end{pmatrix}$$



Metric

Two elements are close if only an approx. with high threshold can distinguish them

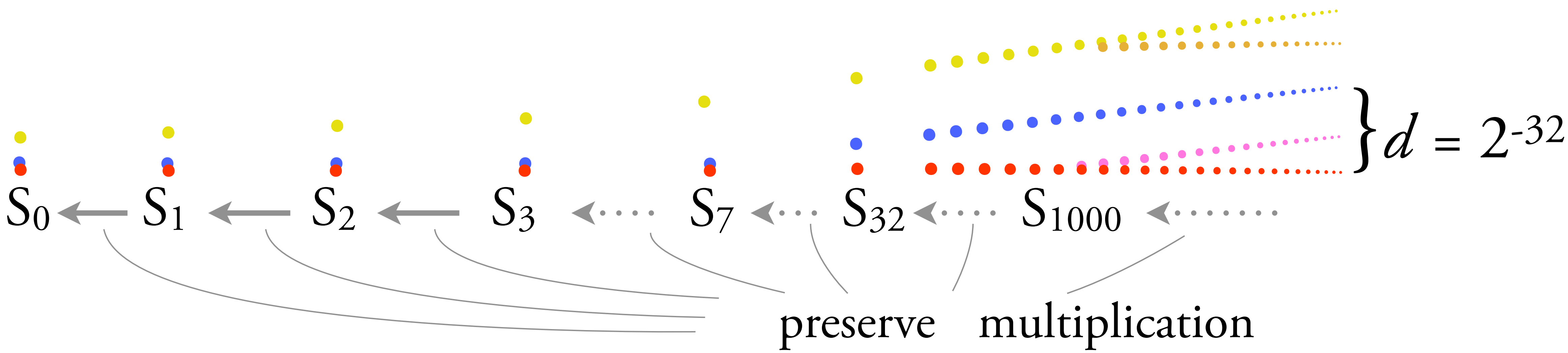
Multiplication *is continuous*

The n-approximation of $x{\cdot}y$ is the product of their n-approximations.

we again obtain a sequence consistent with the mappings

Profinite monoid

$$\begin{pmatrix} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} \cdot \begin{pmatrix} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} = \begin{pmatrix} 3 & 8 & 4 \\ 3 & 8 & 4 \\ 5 & 18 & 3 \end{pmatrix}$$



Metric

Two elements are close if only an approx. with high threshold can distinguish them

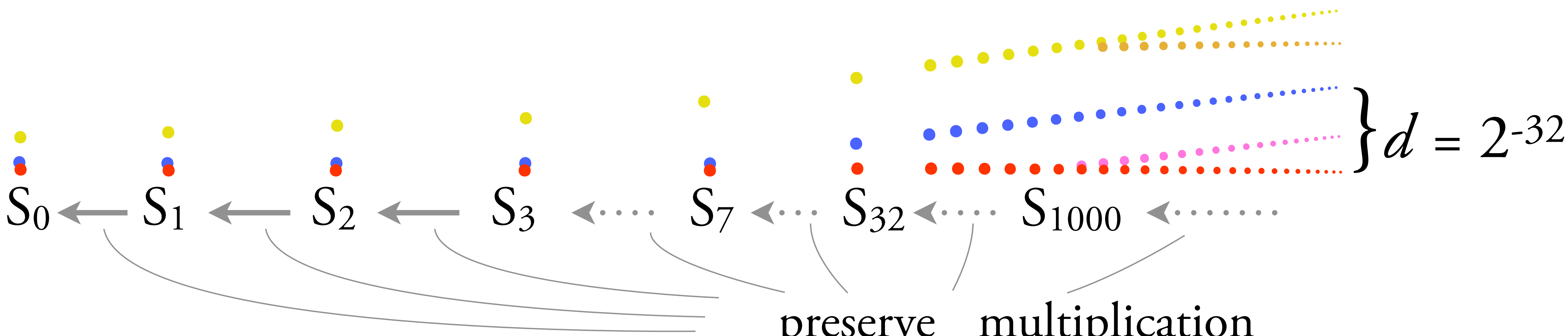
Multiplication *is continuous*

The n-approximation of $x{\cdot}y$ is the product of their n-approximations.

we again obtain a sequence consistent with the mappings

Profinite monoid

$$\begin{pmatrix} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} \cdot \begin{pmatrix} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} = \begin{pmatrix} 3 & 8 & 4 \\ 3 & 8 & 4 \\ 5 & 18 & 3 \end{pmatrix}$$



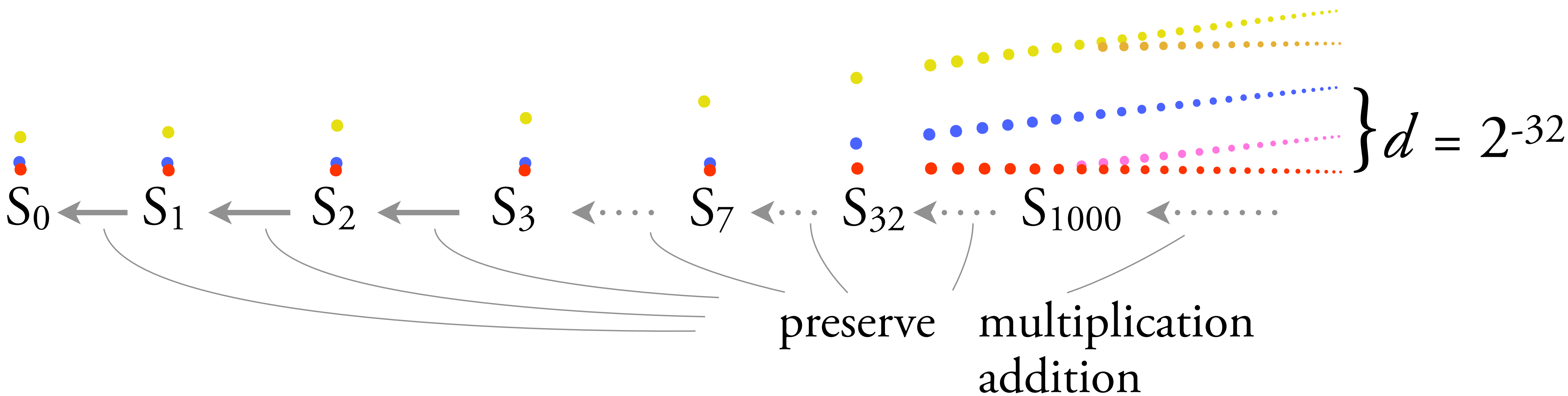
Metric

Two elements are close if only an approx. with high threshold can distinguish them

Multiplication *is continuous*

Profinite monoid

$$\begin{pmatrix} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} \cdot \begin{pmatrix} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} = \begin{pmatrix} 3 & 8 & 4 \\ 3 & 8 & 4 \\ 5 & 18 & 3 \end{pmatrix}$$

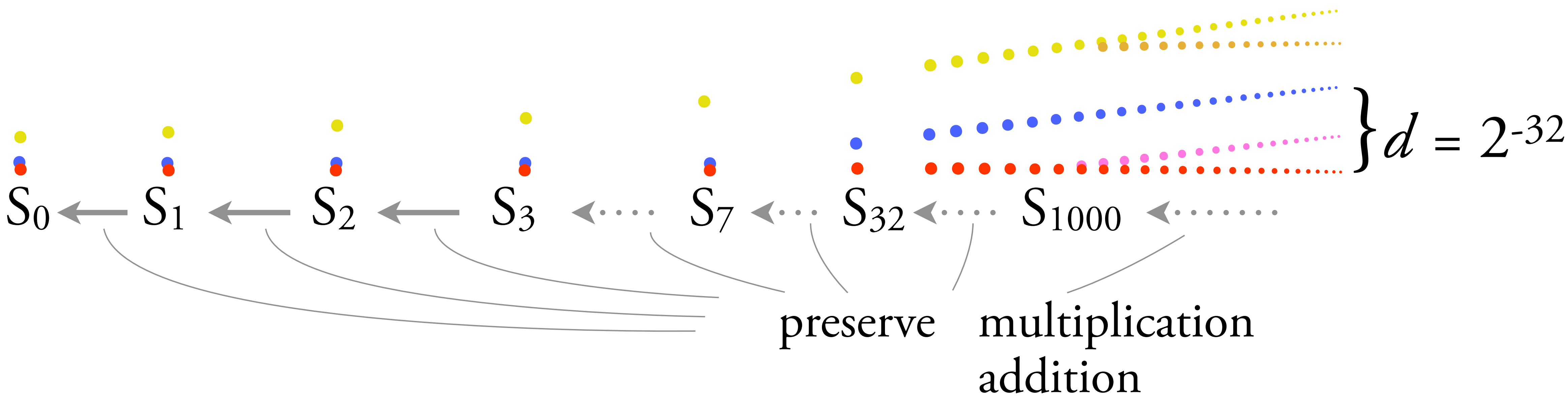


Metric
  Two elements are close if only an approx. with high threshold can distinguish them

Multiplication *is continuous*

Profinite monoid

$$\begin{matrix} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{matrix} \cdot \begin{matrix} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{matrix} = \begin{matrix} 3 & 8 & 4 \\ 3 & 8 & 4 \\ 5 & 18 & 3 \end{matrix}$$



Metric

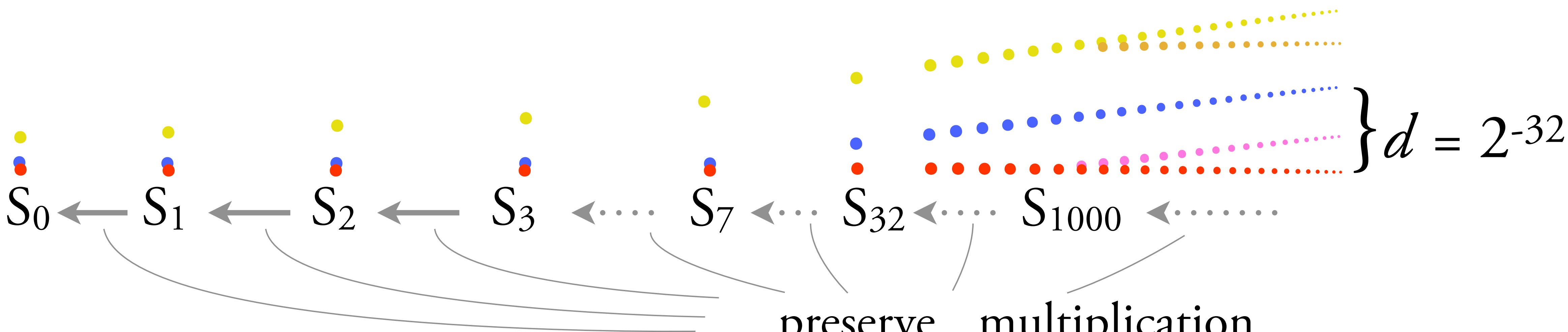
Two elements are close if only an approx. with high threshold can distinguish them

Multiplication *is continuous*

addition

Profinite monoid

$$\begin{pmatrix} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} \cdot \begin{pmatrix} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} = \begin{pmatrix} 3 & 8 & 4 \\ 3 & 8 & 4 \\ 5 & 18 & 3 \end{pmatrix}$$



preserve multiplication
addition
idempotent power (ω)

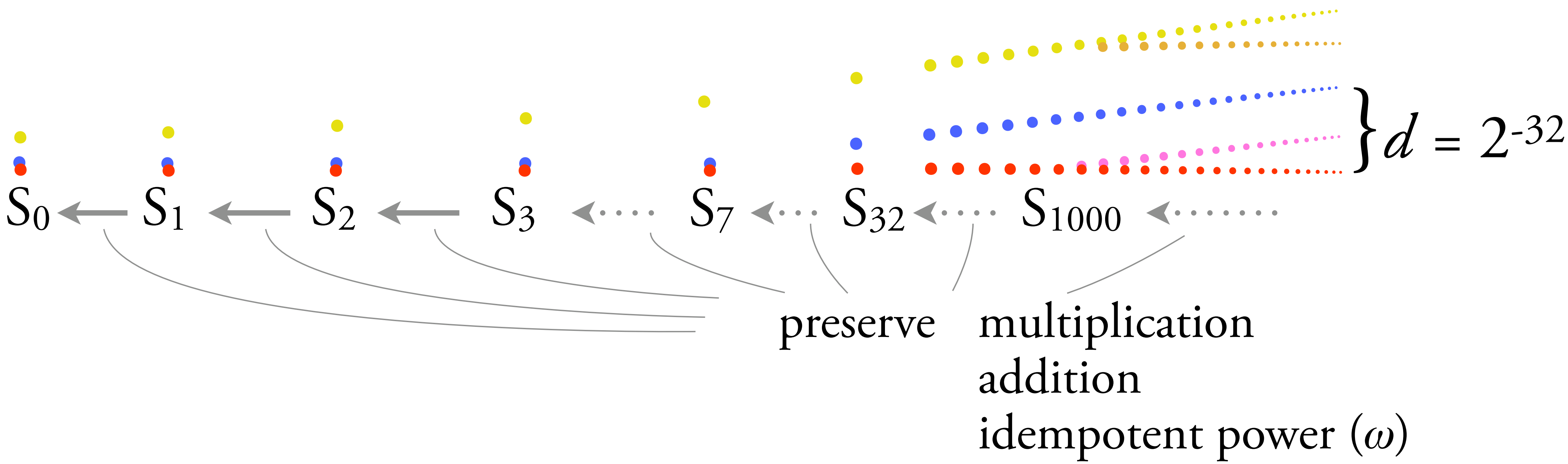
Metric

Two elements are close if only an approx. with high threshold can distinguish them

Multiplication *is continuous*
addition

Profinite monoid

$$\begin{pmatrix} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} \cdot \begin{pmatrix} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} = \begin{pmatrix} 3 & 8 & 4 \\ 3 & 8 & 4 \\ 5 & 18 & 3 \end{pmatrix}$$



Metric

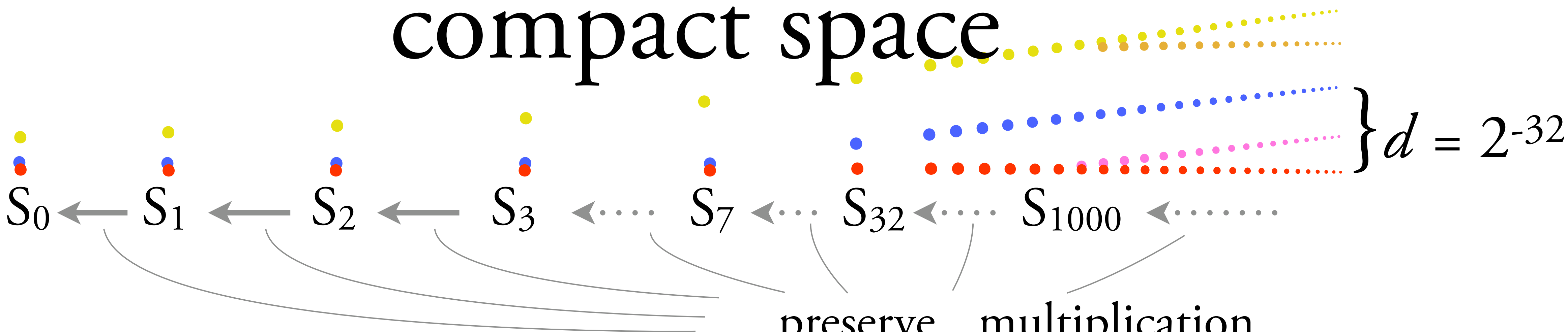
Two elements are close if only an approx. with high threshold can distinguish them

Multiplication *is continuous*
addition
ω-power

Profinite monoid

$$\begin{pmatrix} 3 & 50 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} \cdot \begin{pmatrix} 3 & 32 & 1 \\ 0 & 11 & 1 \\ 2 & 7 & \infty \end{pmatrix} = \begin{pmatrix} 3 & 8 & 4 \\ 3 & 8 & 4 \\ 5 & 18 & 3 \end{pmatrix}$$

compact space



Metric

Two elements are close if only an approx. with high threshold can distinguish them

Multiplication *is continuous*
addition
ω-power

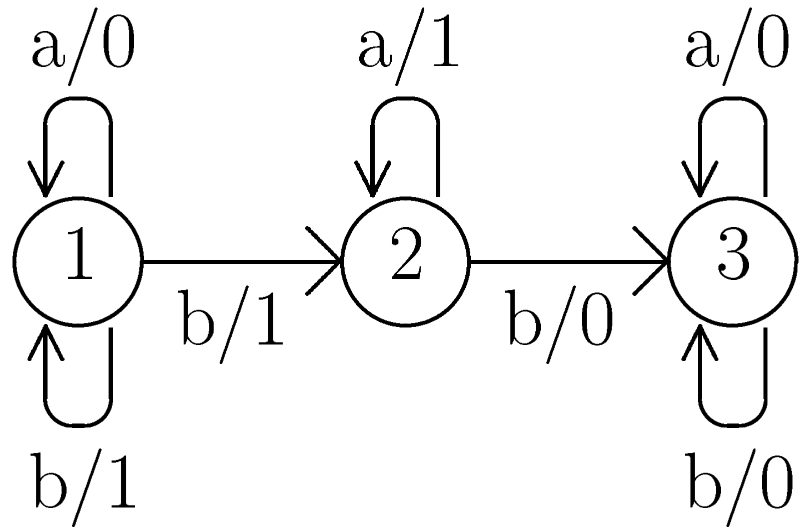


Figure 1: Example of a distance automaton A.

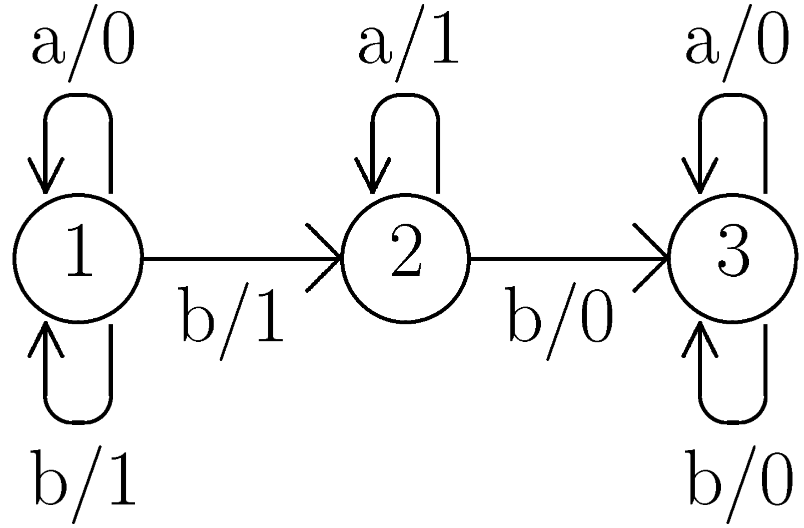


Figure 1: Example of a distance automaton A.

Example of a min-automaton *A.*

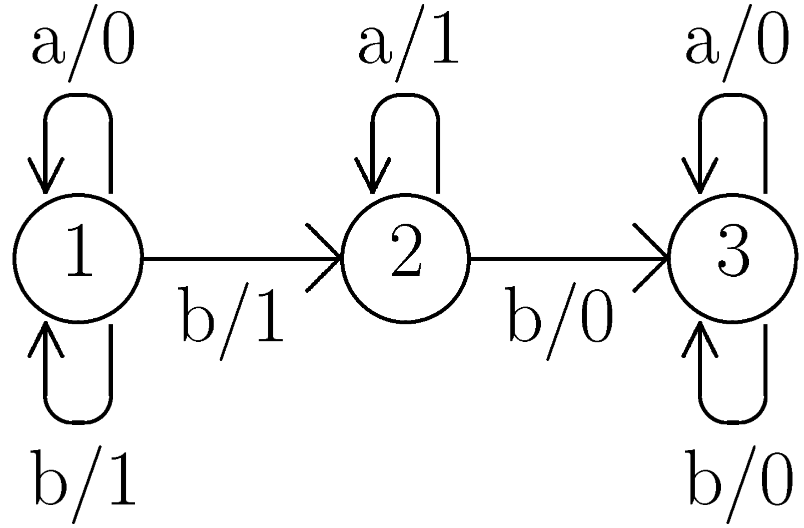


Figure 1: Example of a distance automaton A.

Example of a min-automaton *A*.

$$a: \quad \begin{bmatrix} 0 & \top & \top \\ \top & 1 & \top \\ \top & \top & 0 \end{bmatrix} \qquad b: \quad \begin{bmatrix} 1 & 1 & \top \\ \top & \top & 0 \\ \top & \top & 0 \end{bmatrix}$$

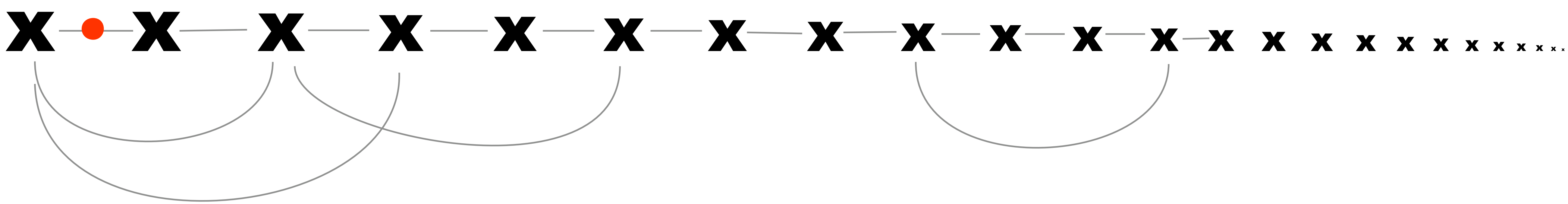
$$b^{\omega}: \quad \begin{bmatrix} \infty & \infty & 1 \\ \top & \top & 0 \\ \top & \top & 0 \end{bmatrix}$$

Ramsey Theorem

for compact spaces

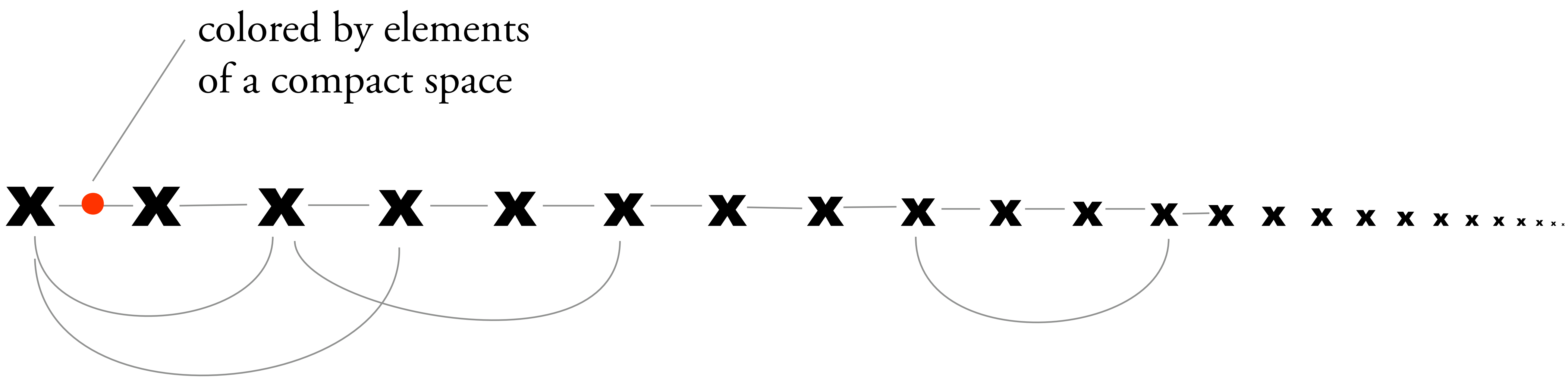
Ramsey Theorem

for compact spaces



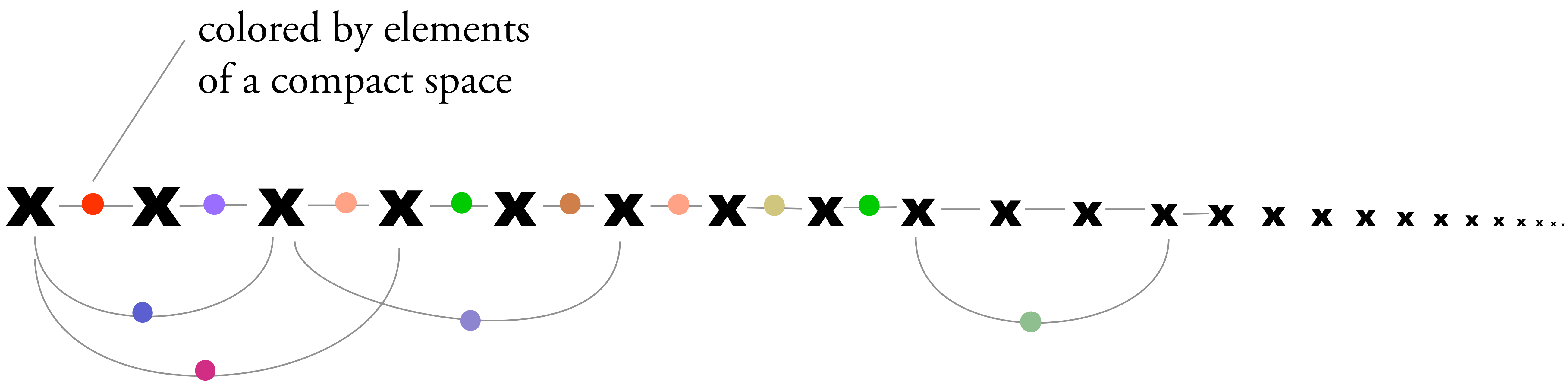
Ramsey Theorem

for compact spaces



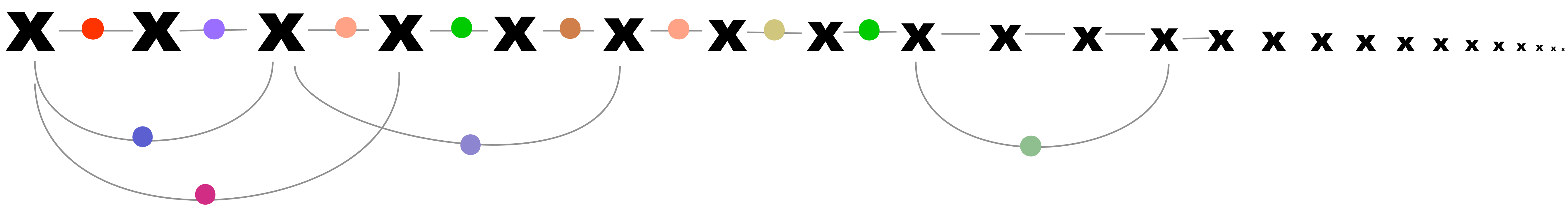
Ramsey Theorem

for compact spaces



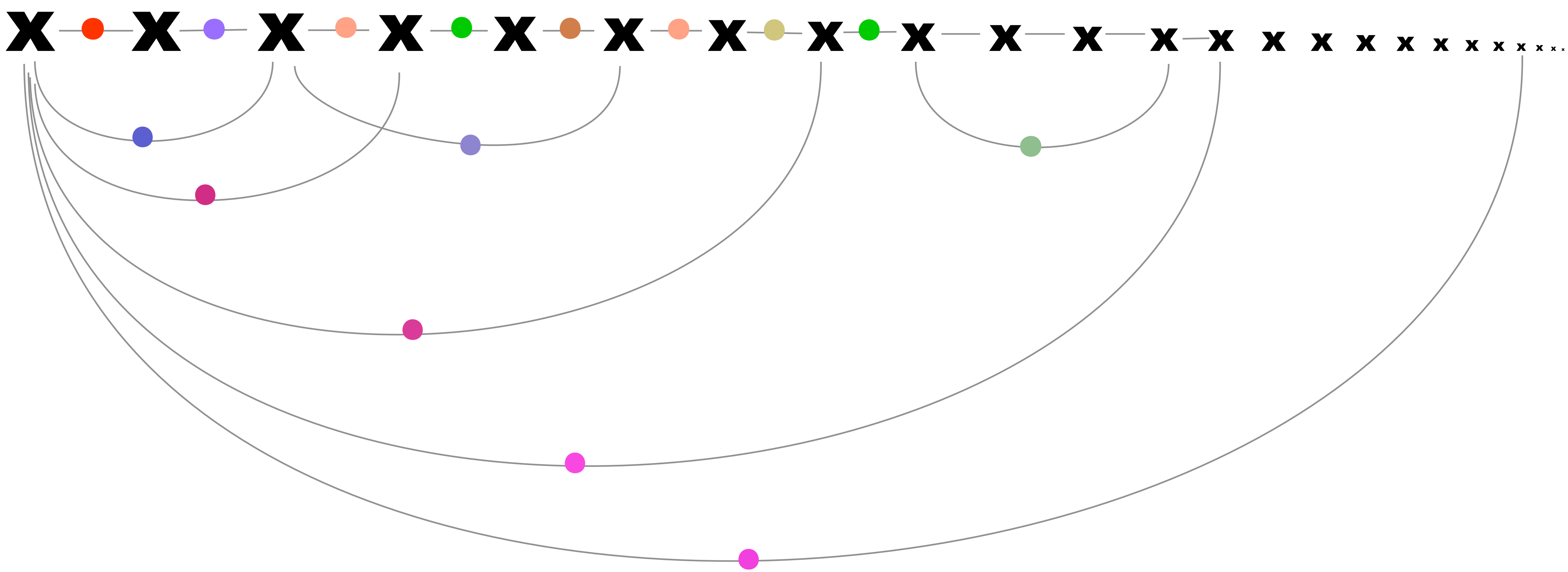
Ramsey Theorem

for compact spaces



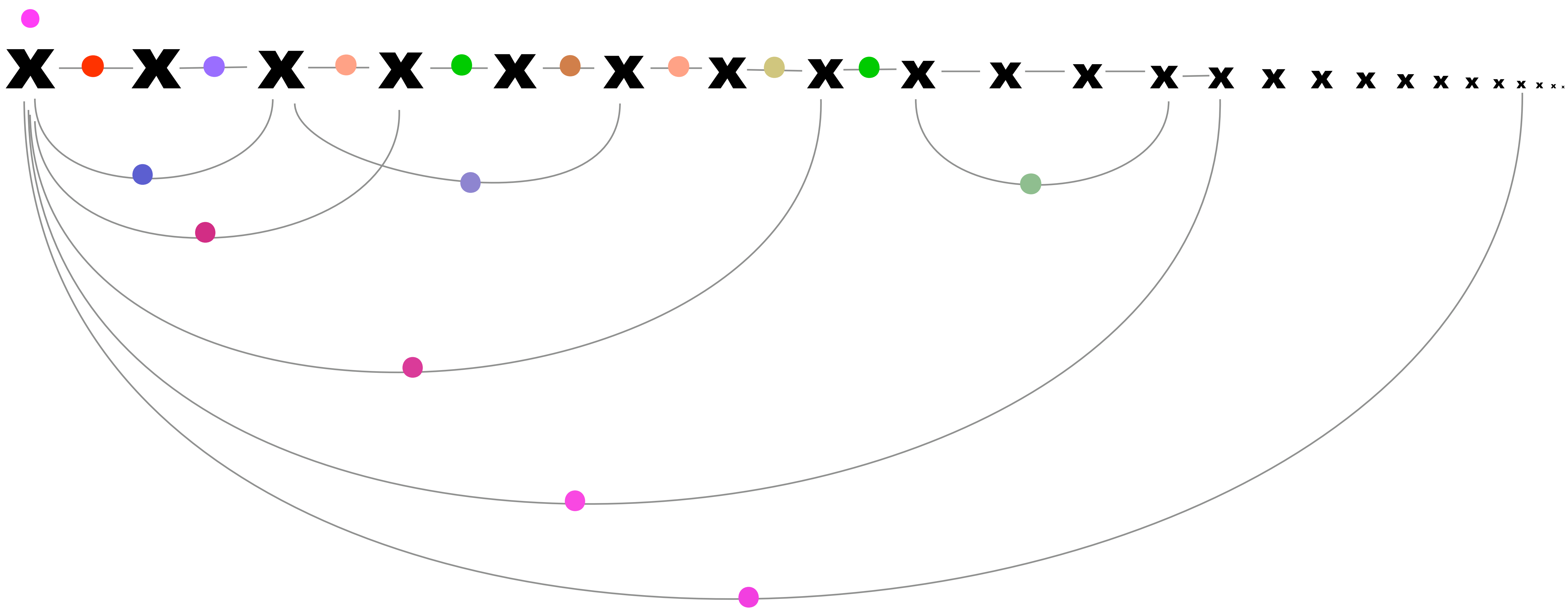
Ramsey Theorem

for compact spaces



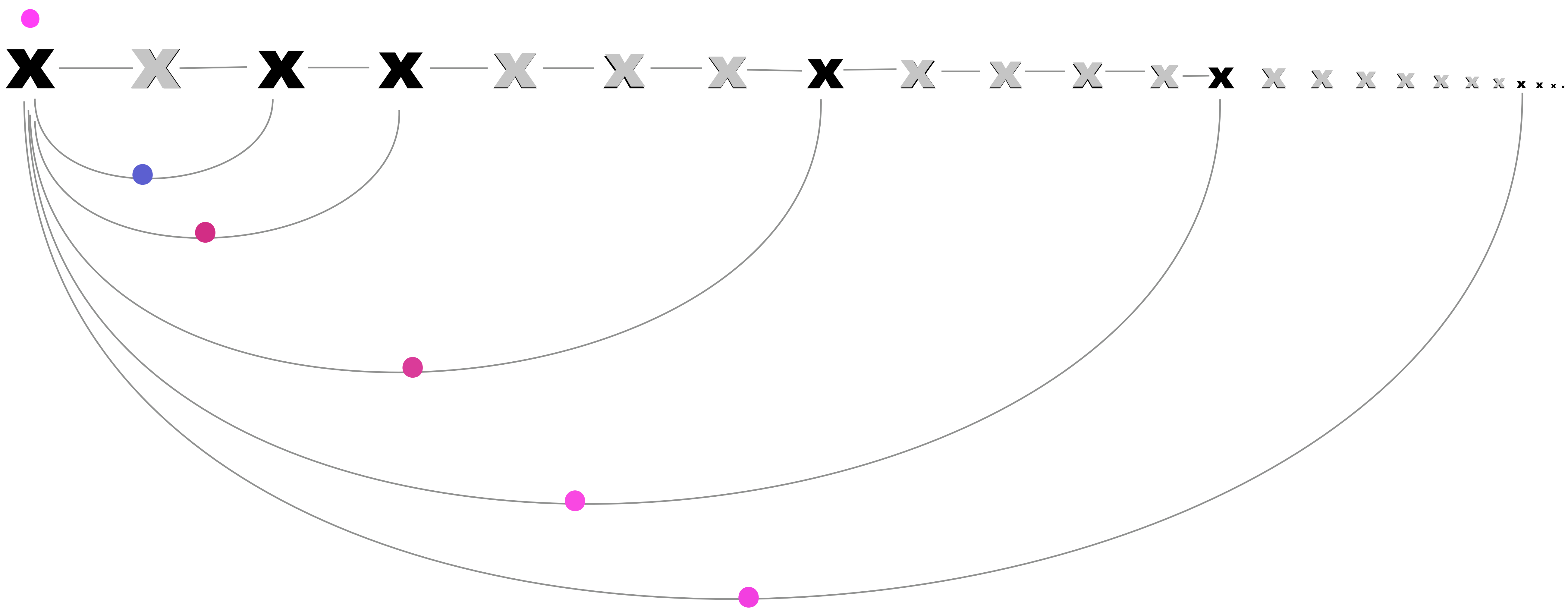
Ramsey Theorem

for compact spaces



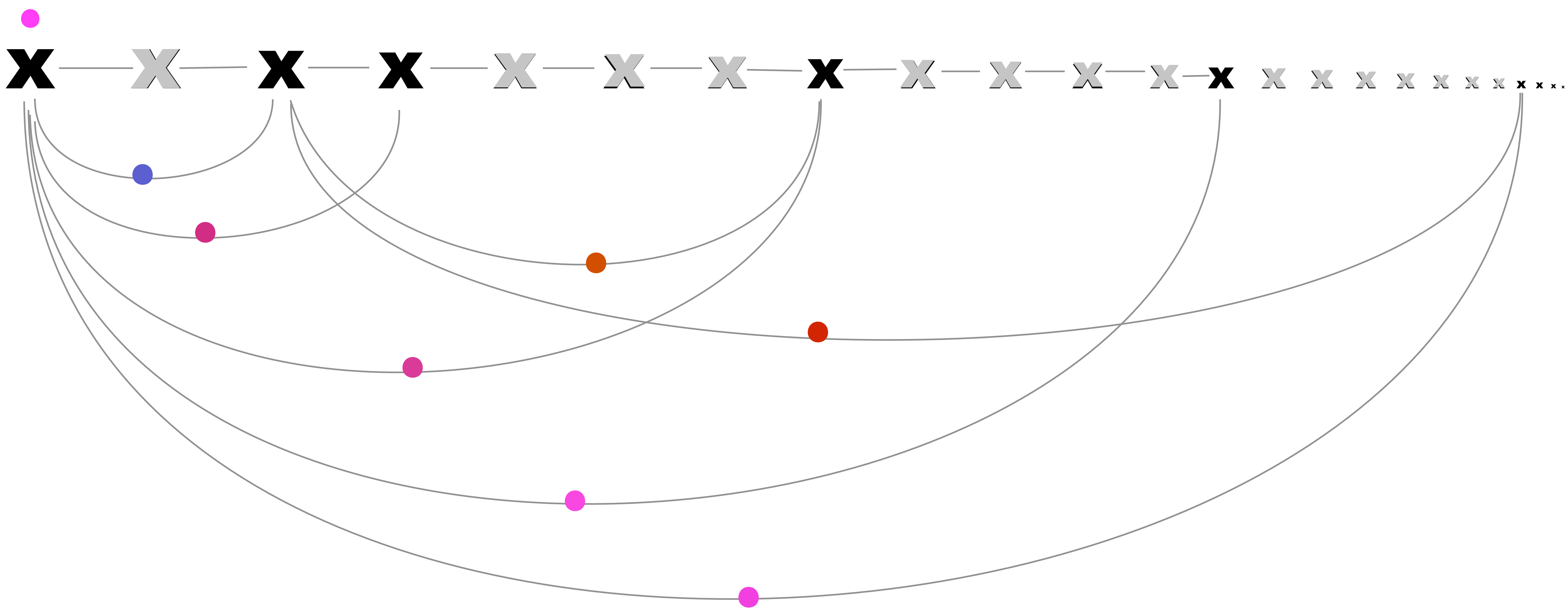
Ramsey Theorem

for compact spaces



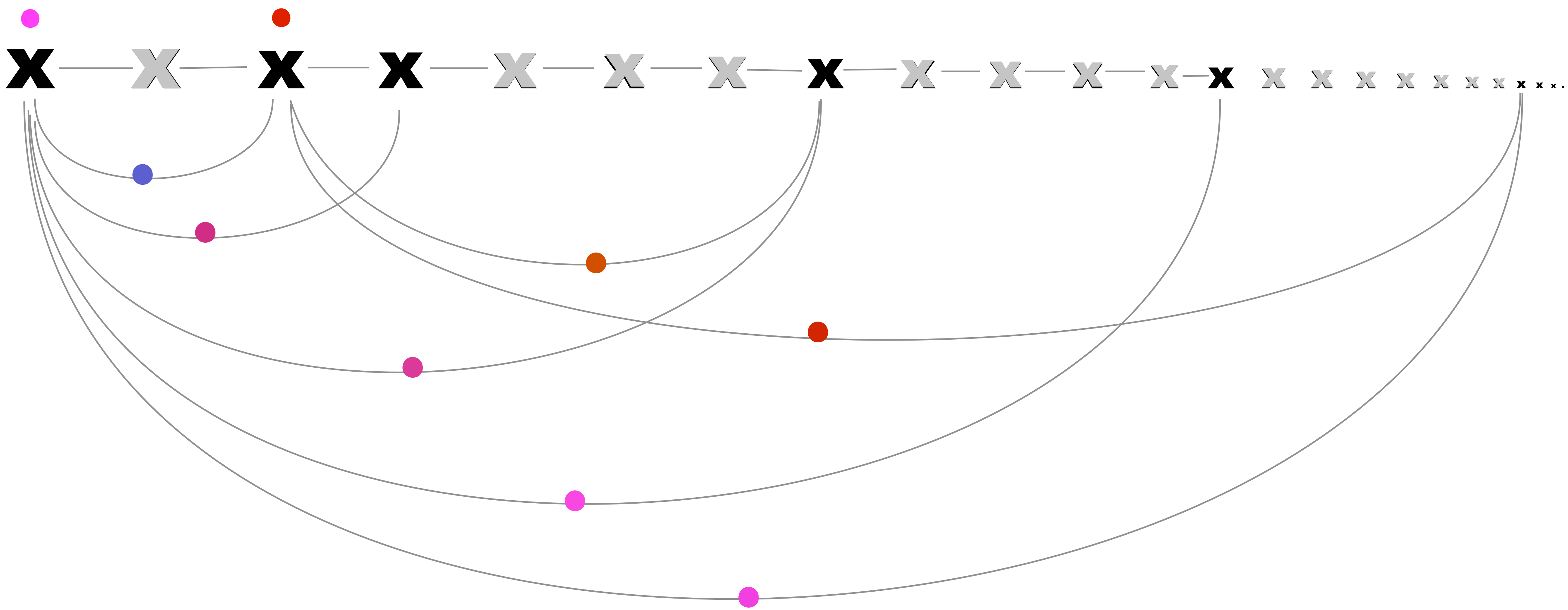
Ramsey Theorem

for compact spaces



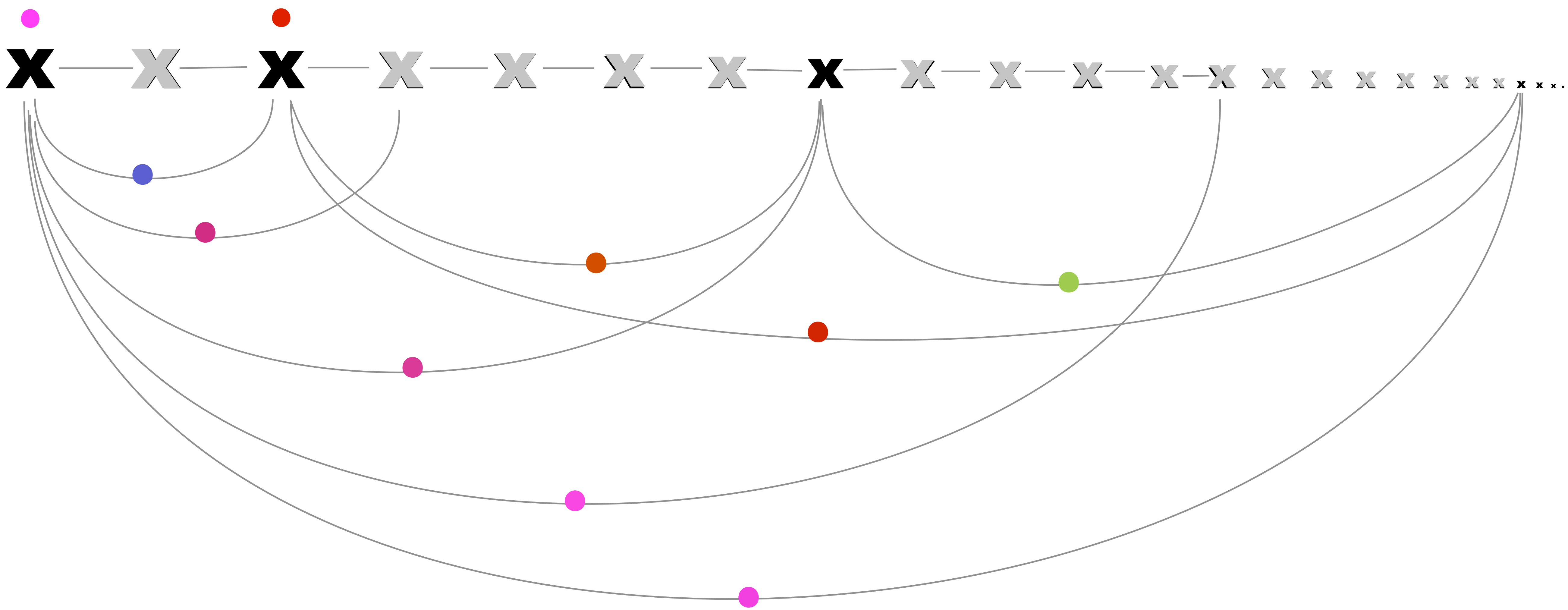
Ramsey Theorem

for compact spaces



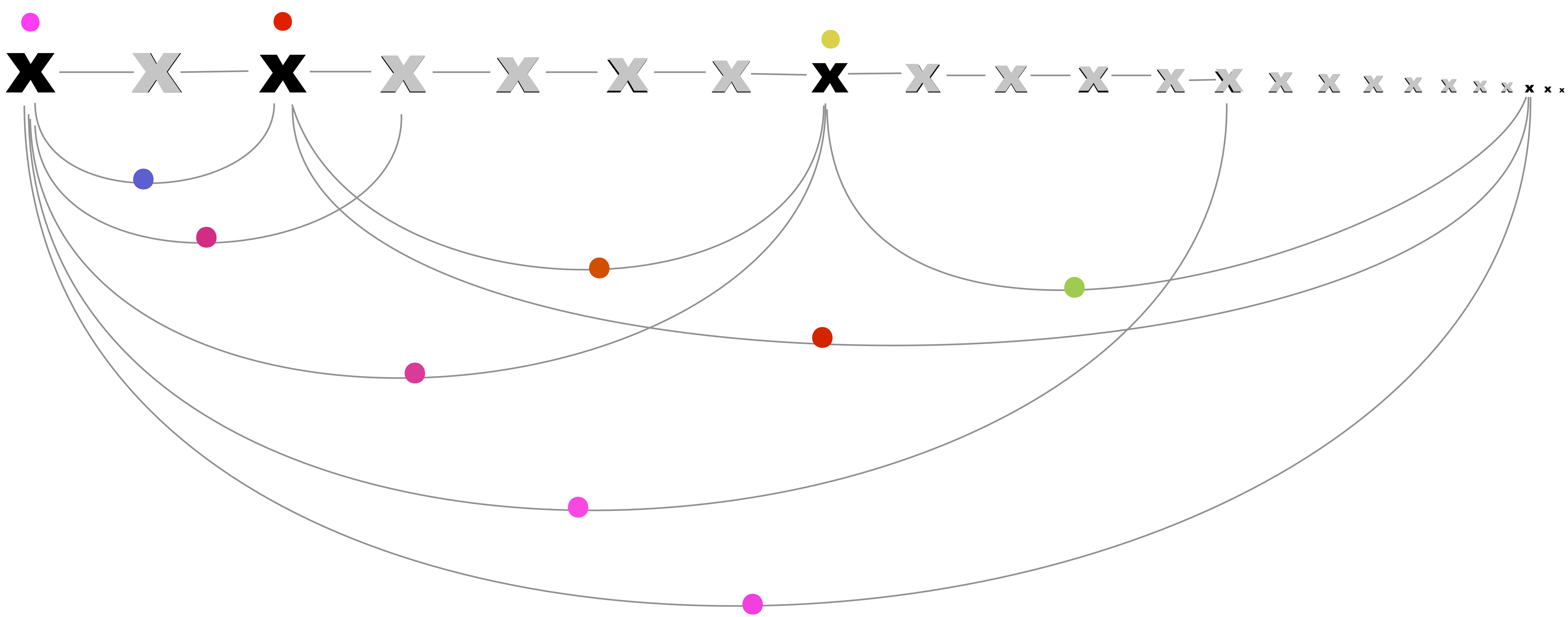
Ramsey Theorem

for compact spaces



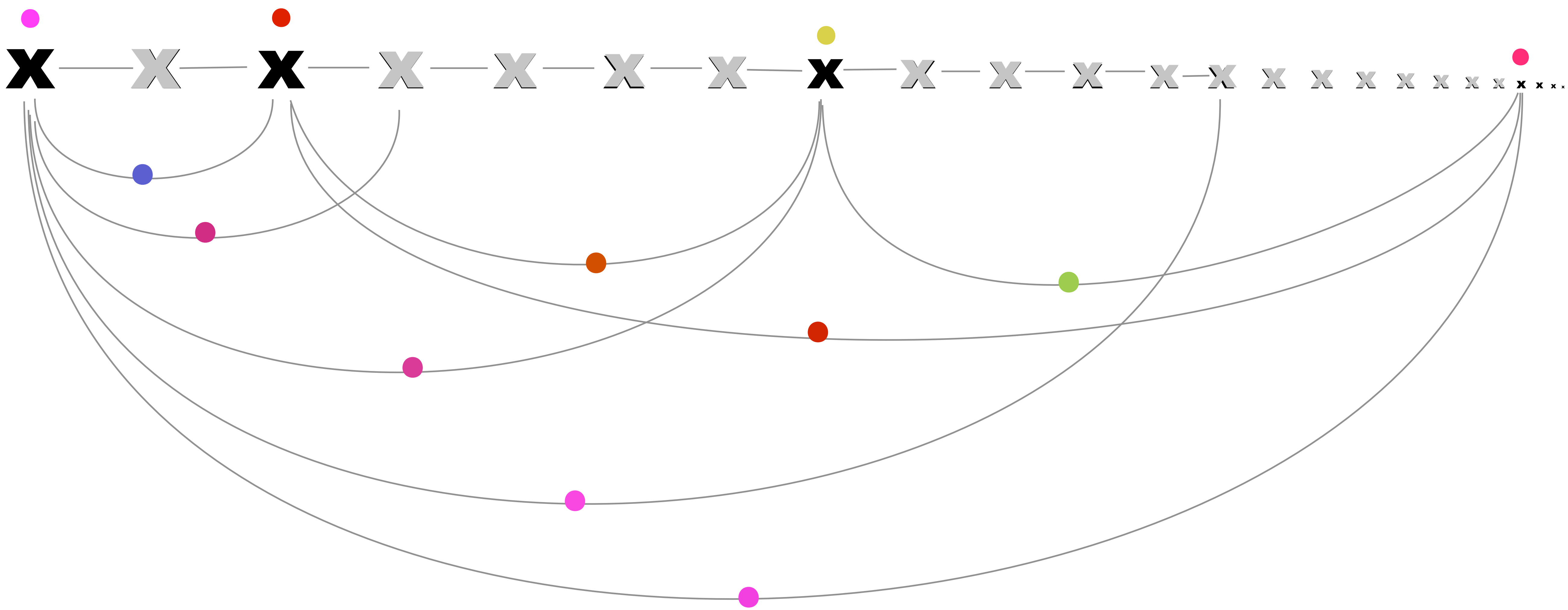
Ramsey Theorem

for compact spaces



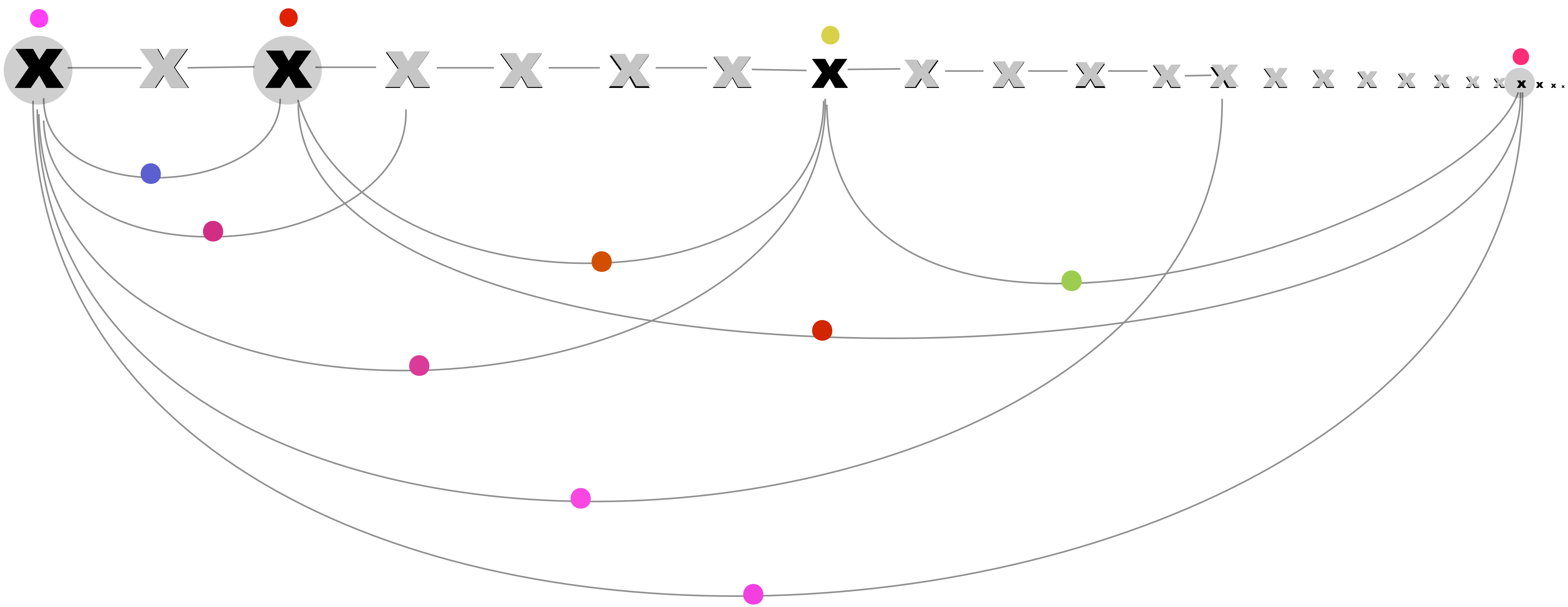
Ramsey Theorem

for compact spaces



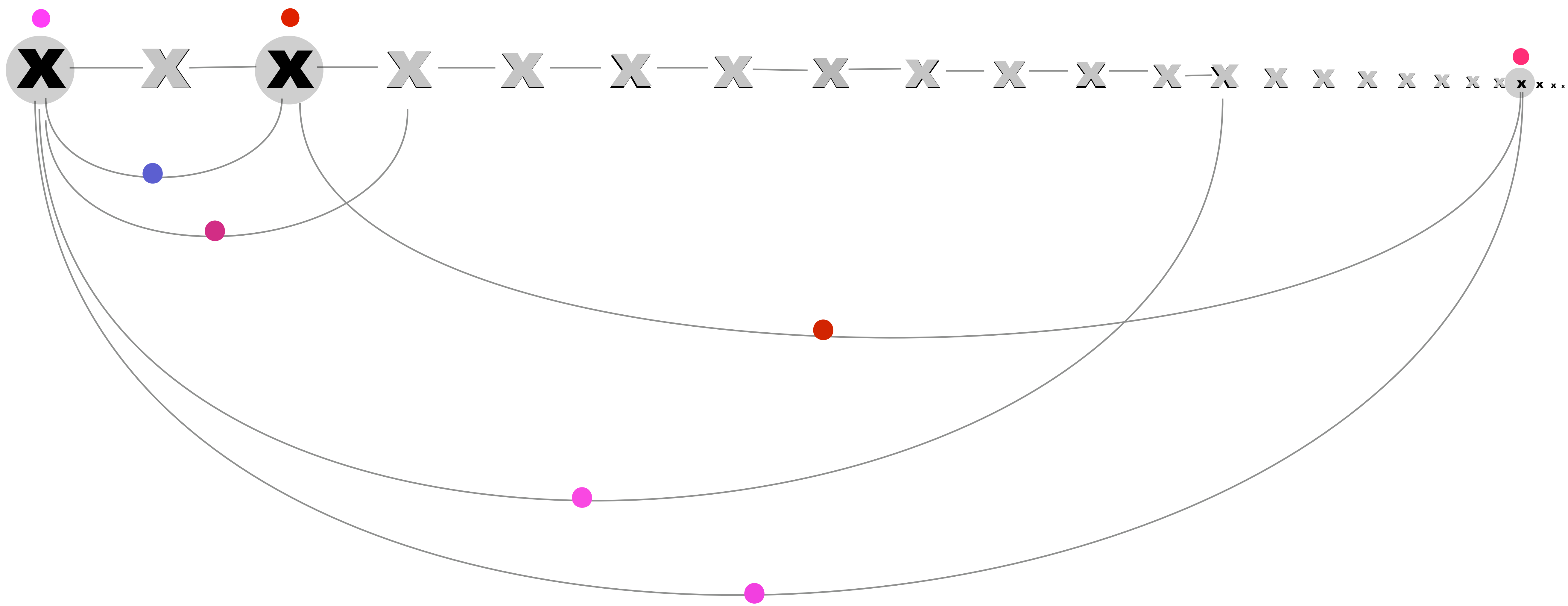
Ramsey Theorem

for compact spaces



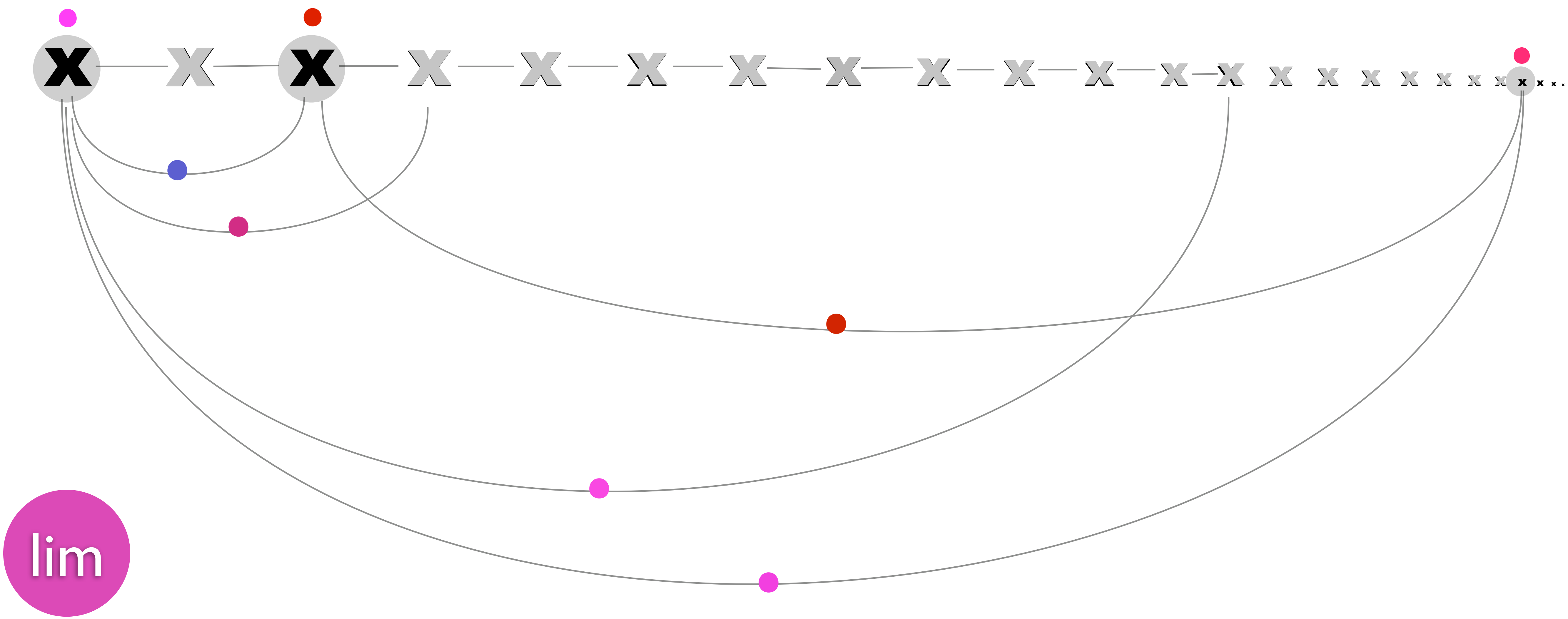
Ramsey Theorem

for compact spaces



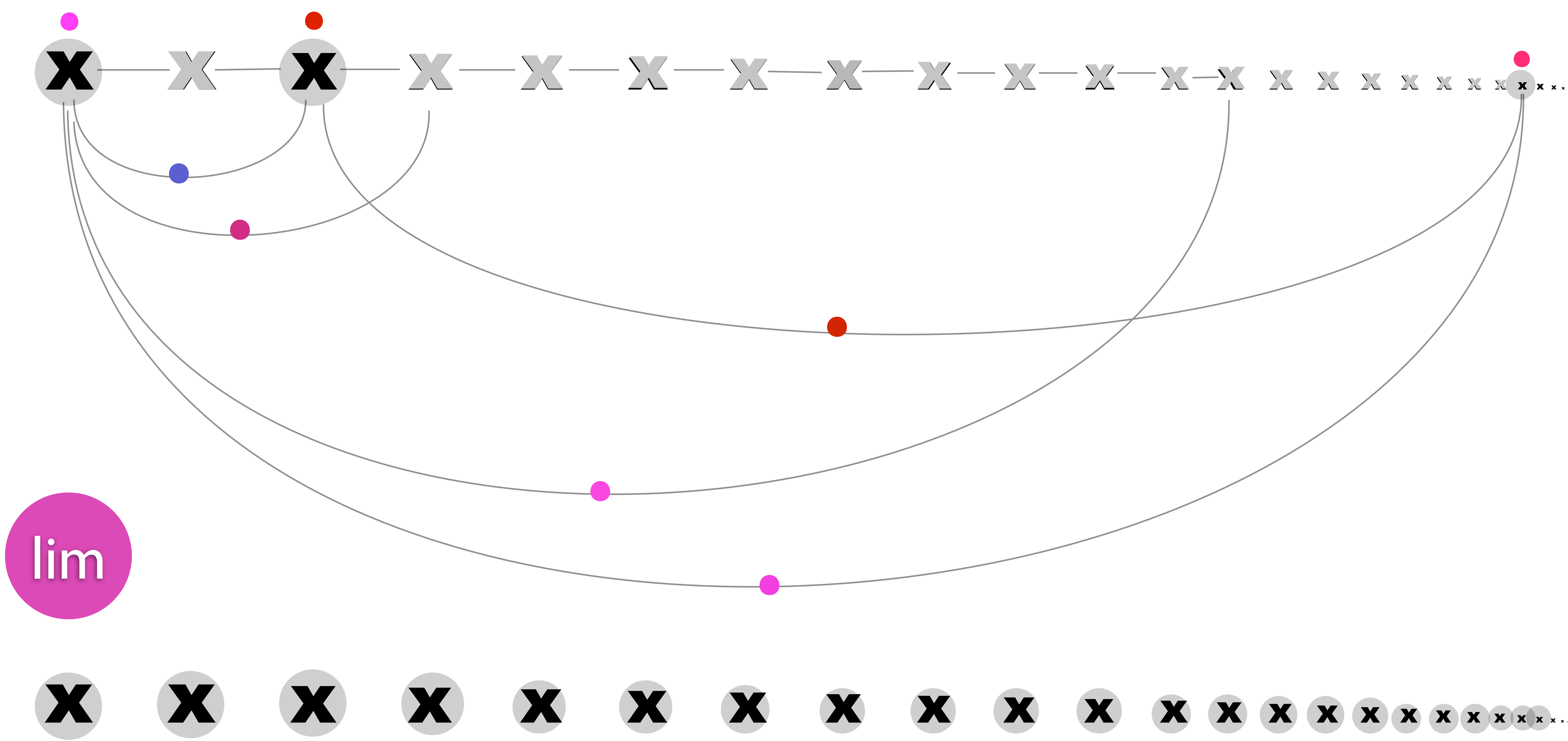
Ramsey Theorem

for compact spaces



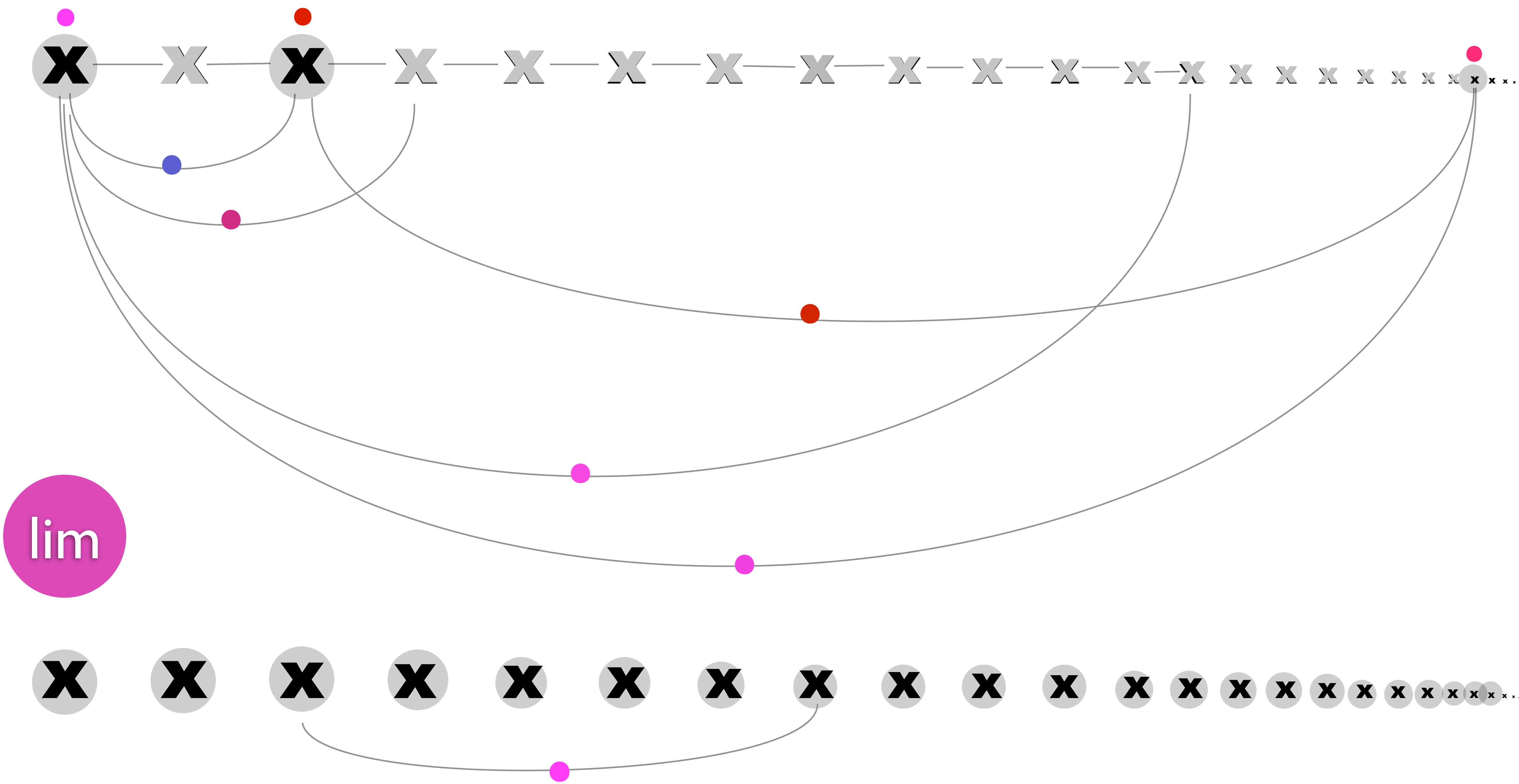
Ramsey Theorem

for compact spaces



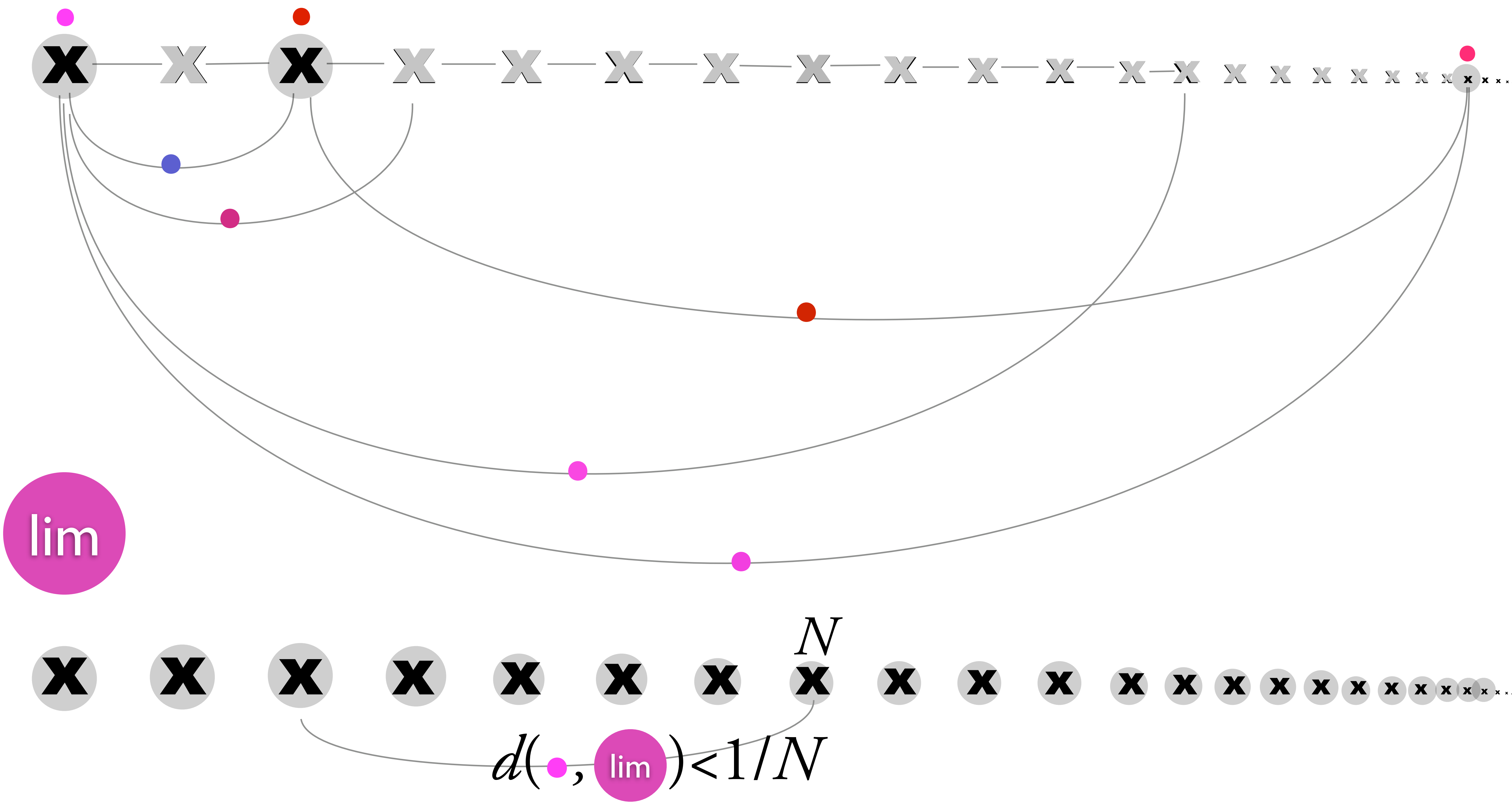
Ramsey Theorem

for compact spaces



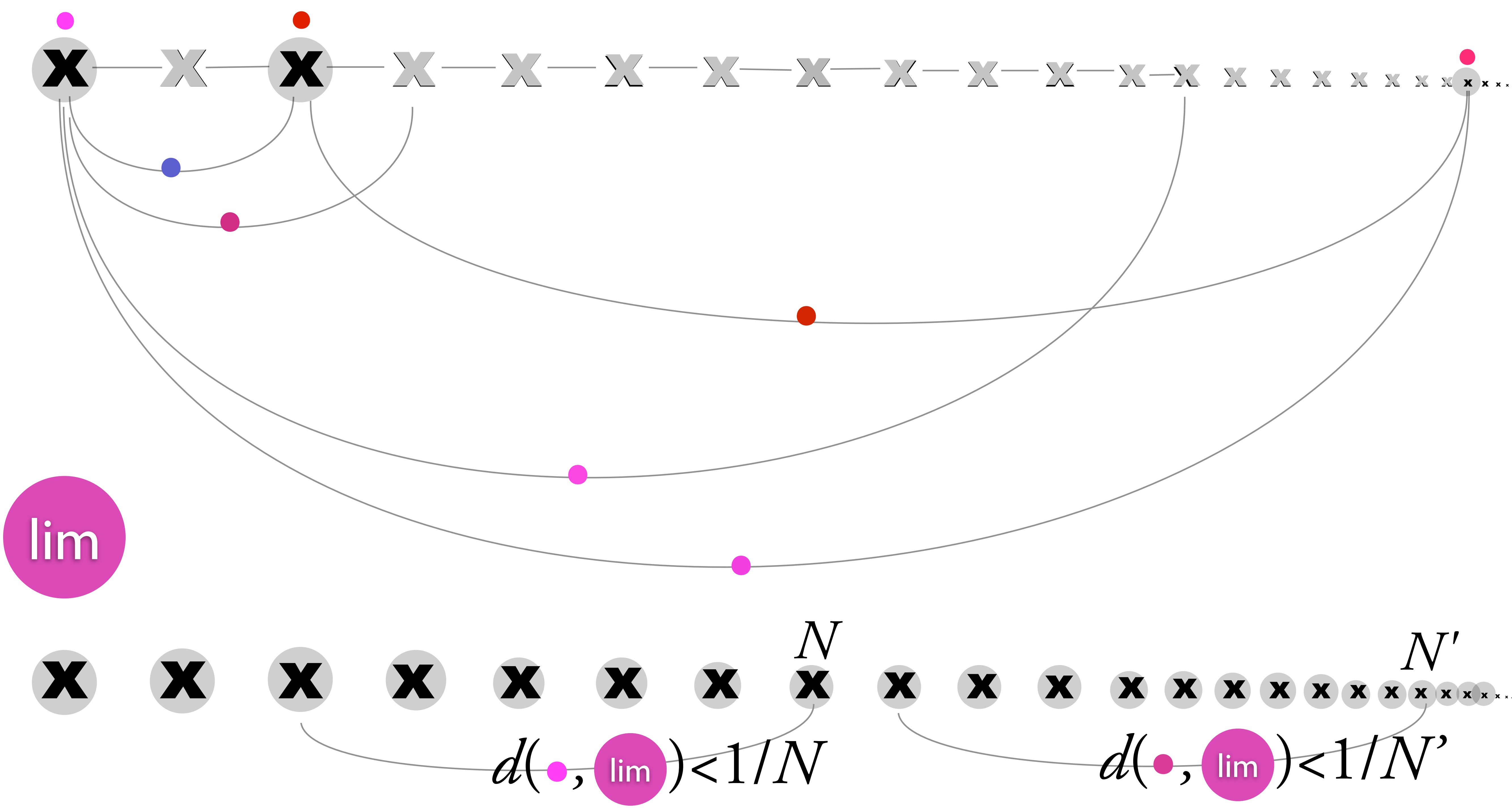
Ramsey Theorem

for compact spaces



Ramsey Theorem

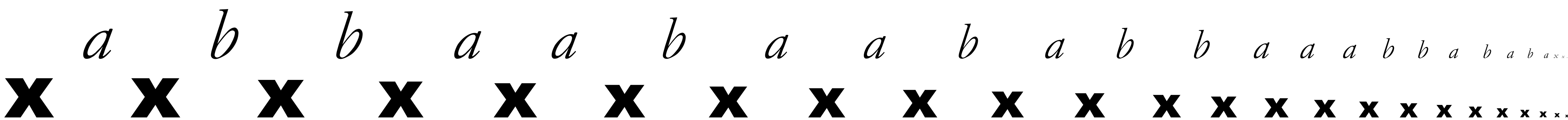
for compact spaces



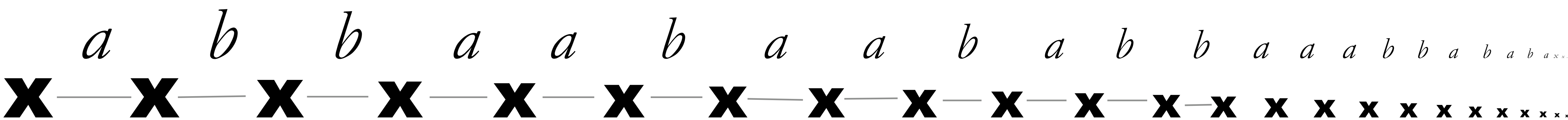
Emptiness of min-automata

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

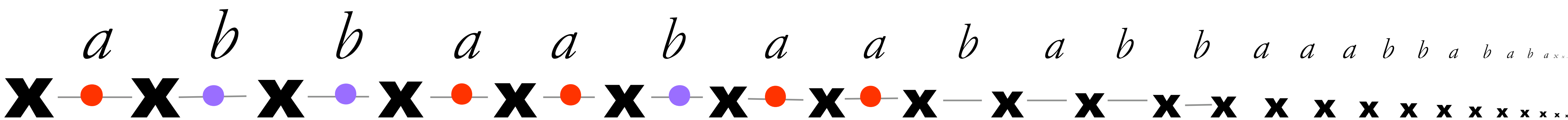
Emptiness of min-automata



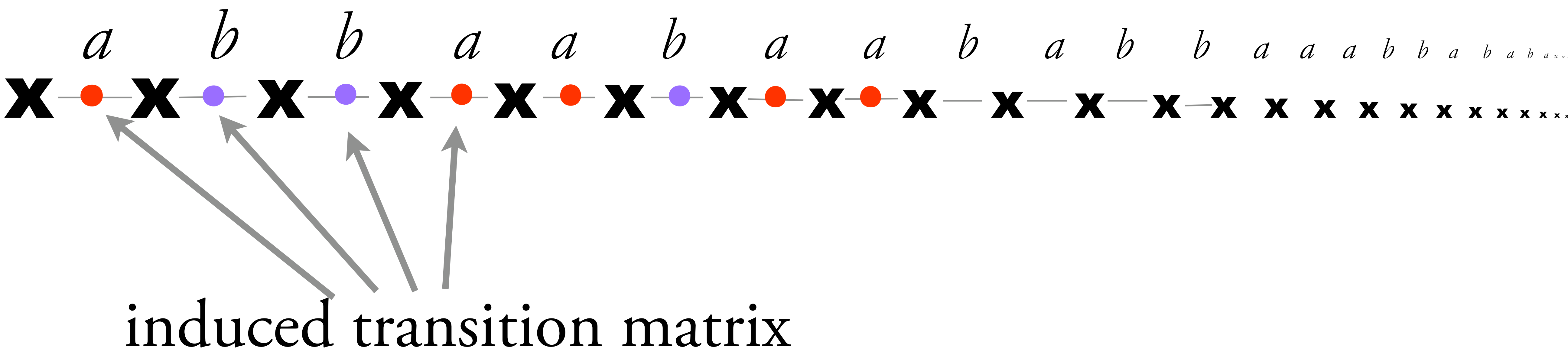
Emptiness of min-automata



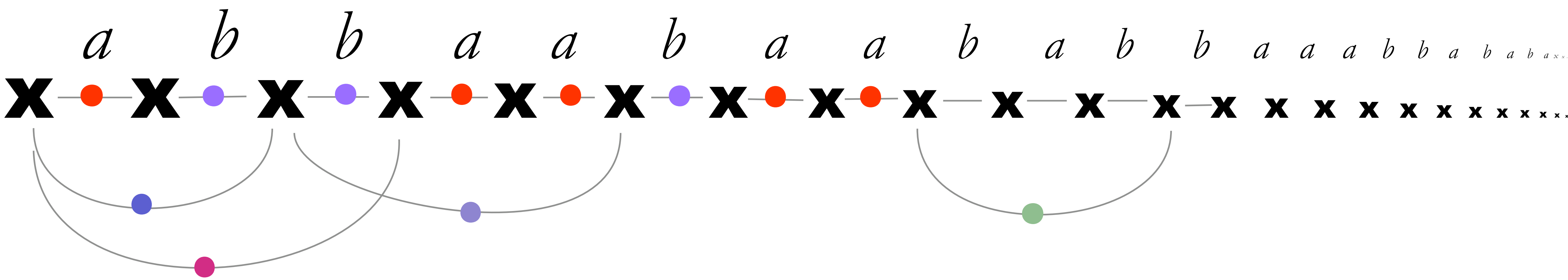
Emptiness of min-automata



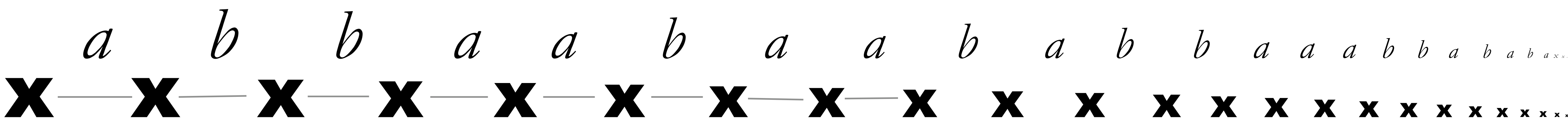
Emptiness of min-automata



Emptiness of min-automata

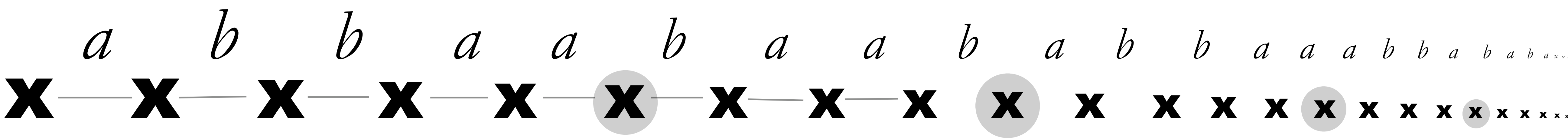


Emptiness of min-automata



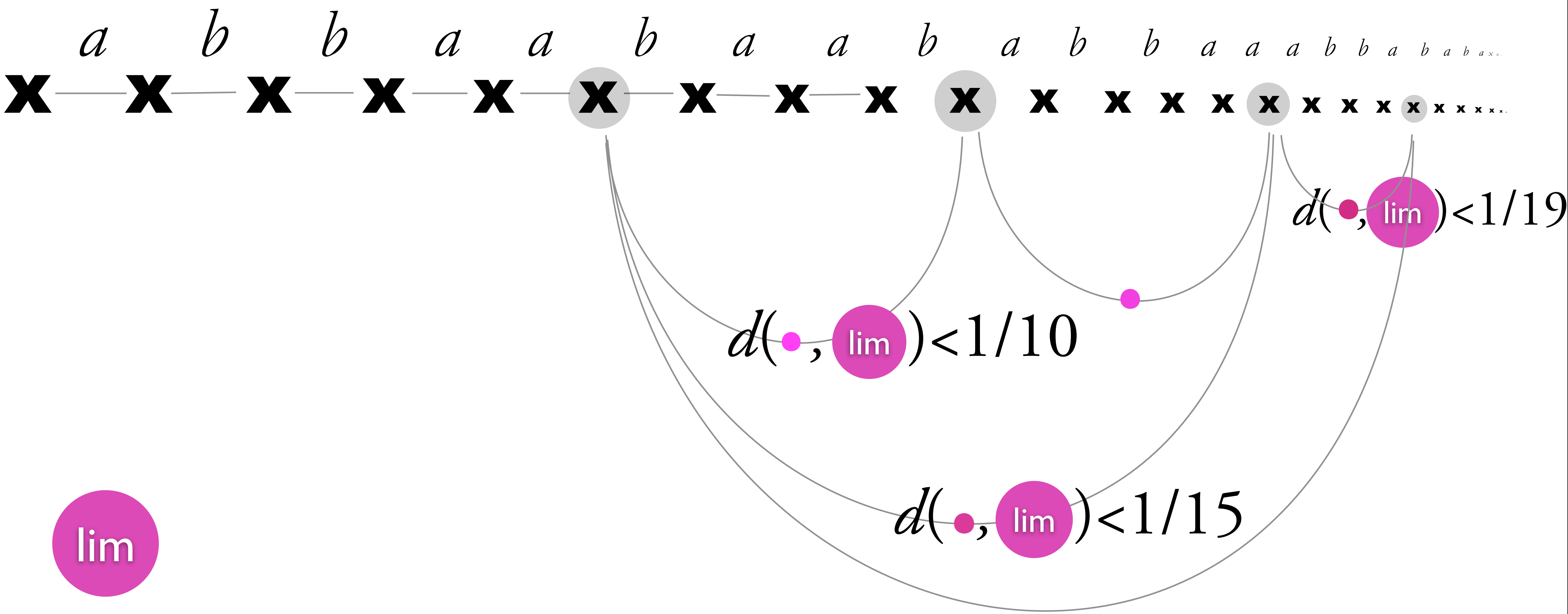
lim

Emptiness of min-automata

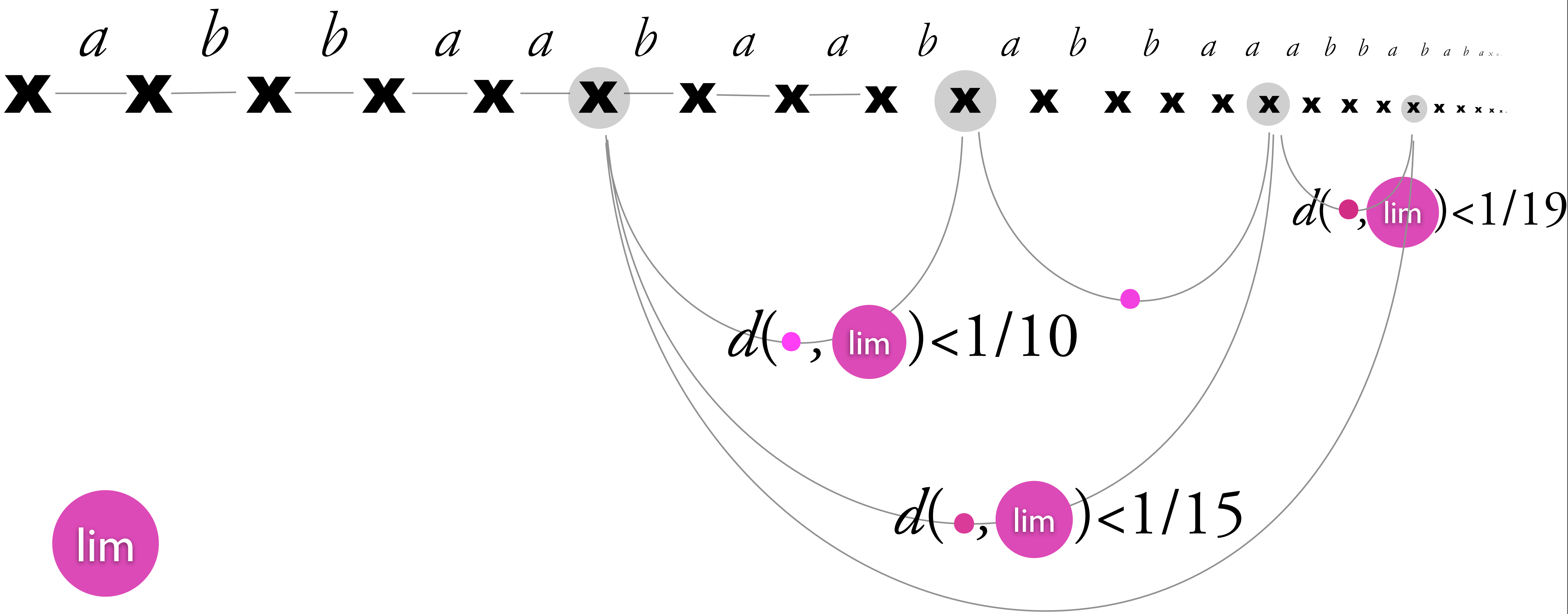


lim

Emptiness of min-automata



Emptiness of min-automata

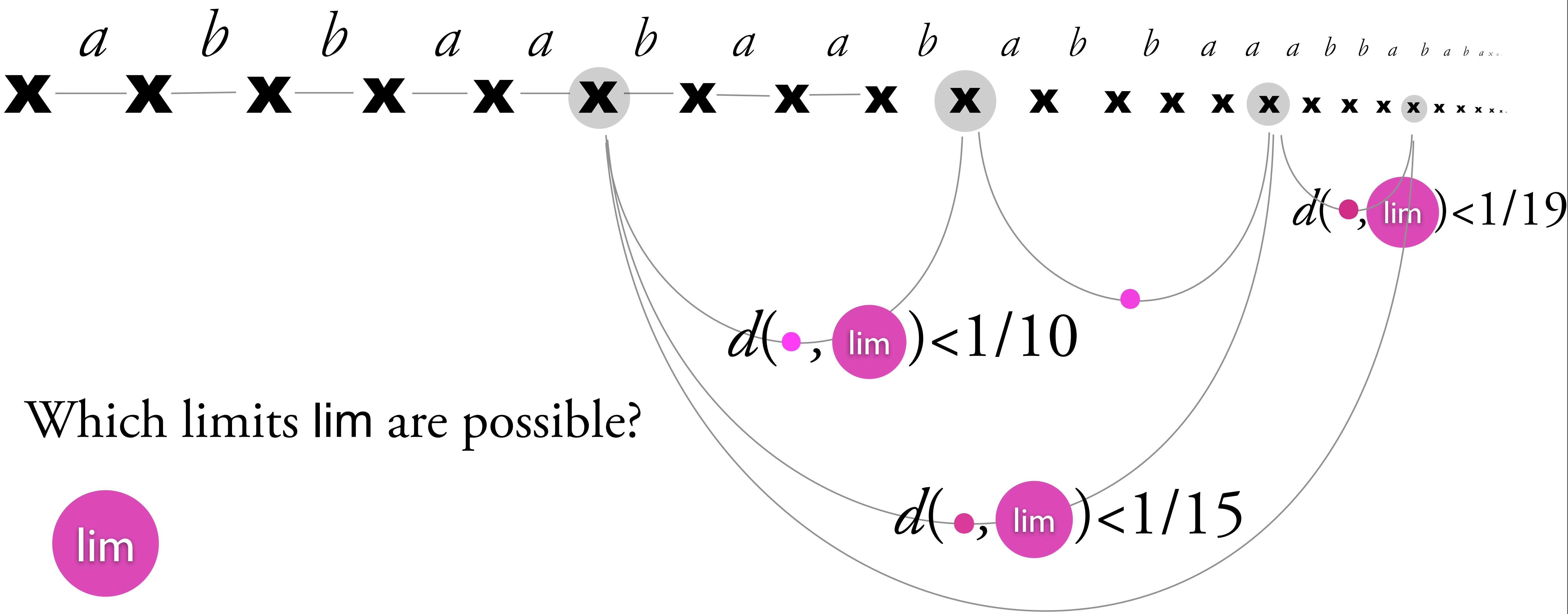


Counter c does *not* converge to ∞

iff exists a counter d such that

$\lim[d,d]=0$ *and* $\lim[d,c]<\infty$.

Emptiness of min-automata



Which limits lim are possible?

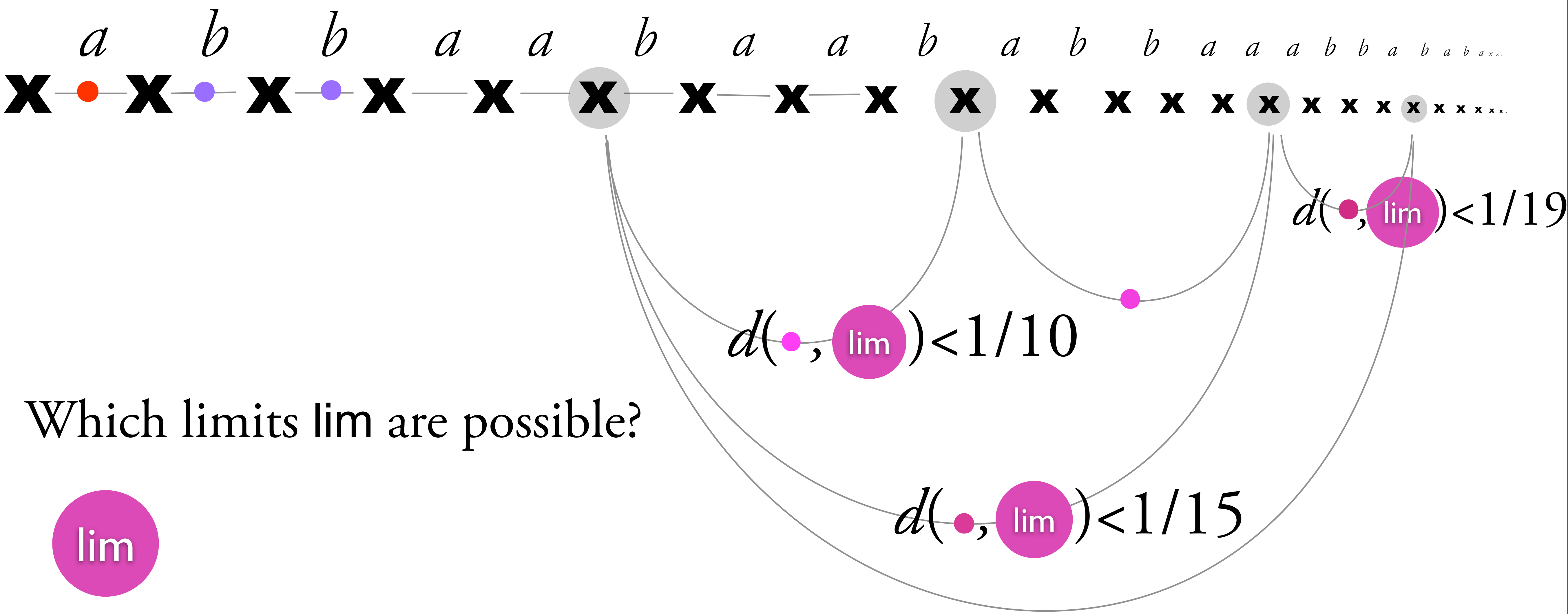
lim

Counter c does *not* converge to ∞

iff exists a counter d such that

$\text{lim}[d,d] = 0$ *and* $\text{lim}[d,c] < \infty$.

Emptiness of min-automata



Which limits lim are possible?

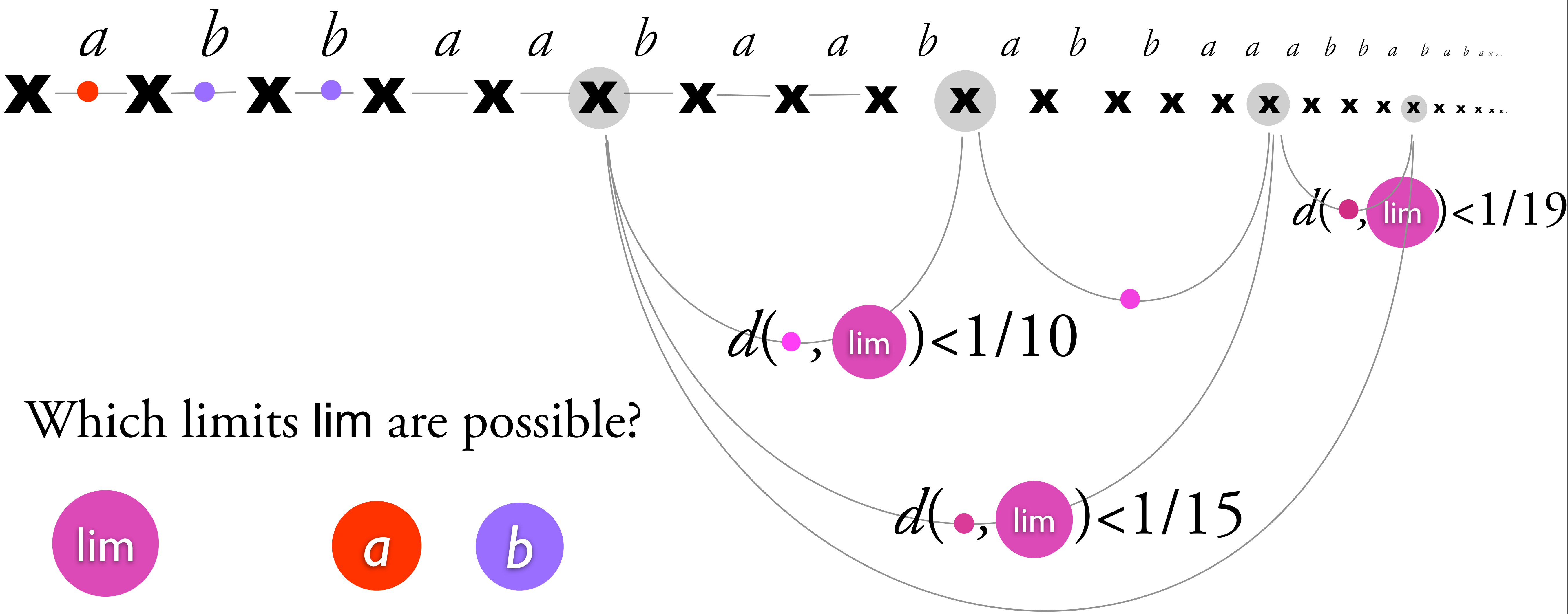
lim

Counter c does *not* converge to ∞

iff exists a counter d such that

lim$[d,d]$ *= 0 and* lim$[d,c] < \infty$.

Emptiness of min-automata



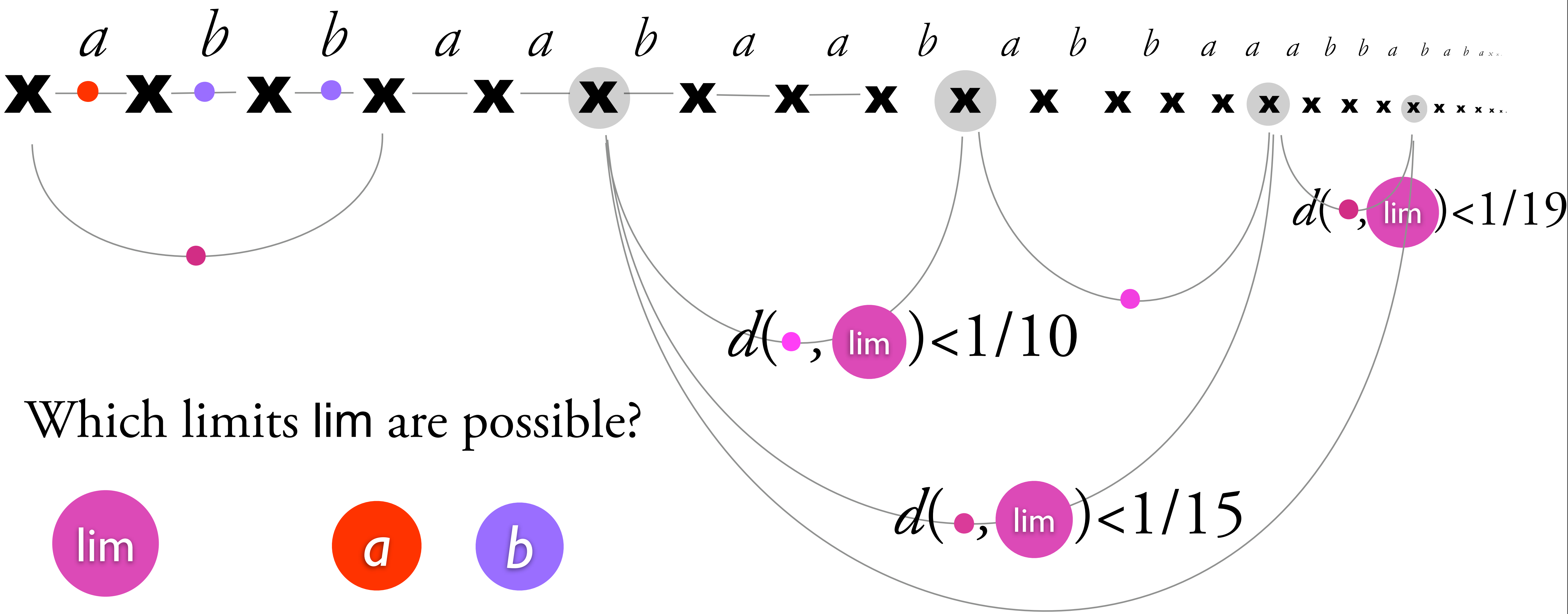
Which limits lim are possible?

Counter c does *not* converge to ∞

iff exists a counter d such that

$\mathrm{lim}[d,d] = 0$ *and* $\mathrm{lim}[d,c] < \infty$.

Emptiness of min-automata



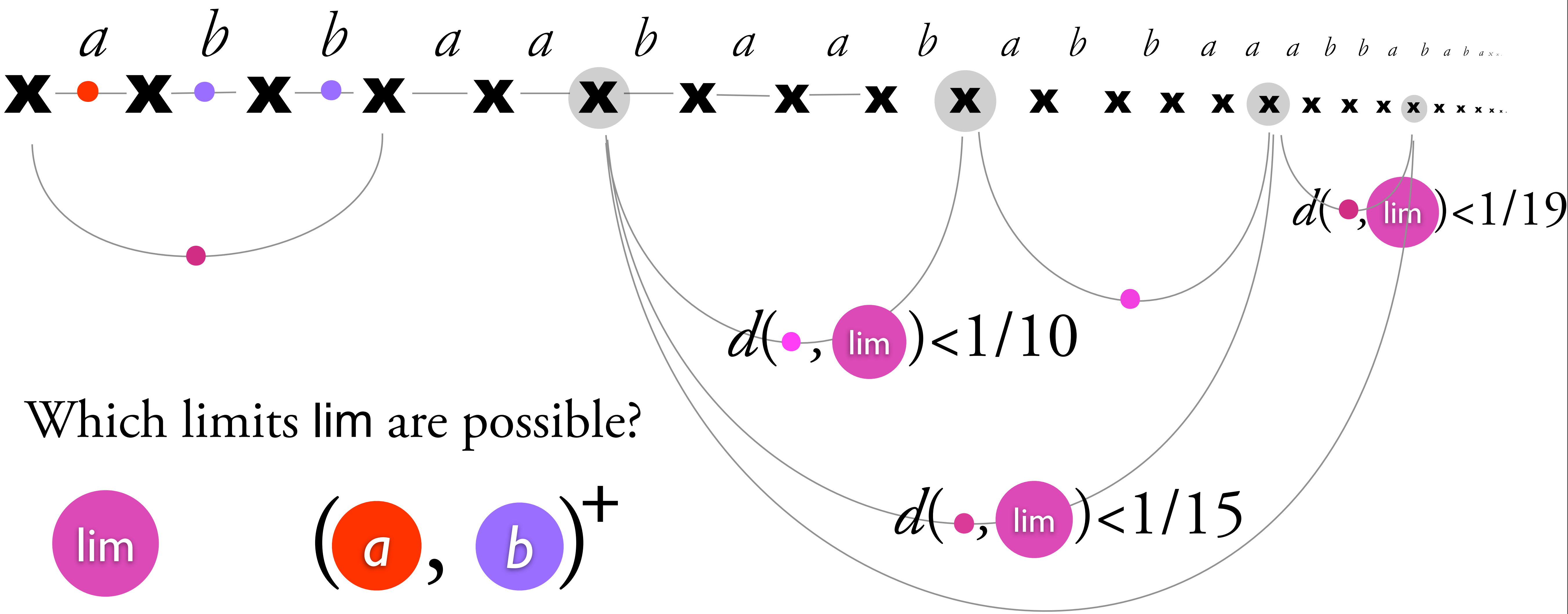
Which limits lim are possible?

Counter c does *not* converge to ∞

iff exists a counter d such that

lim$[d,d]=0$ *and* lim$[d,c]<\infty$.

Emptiness of min-automata



Which limits lim are possible?

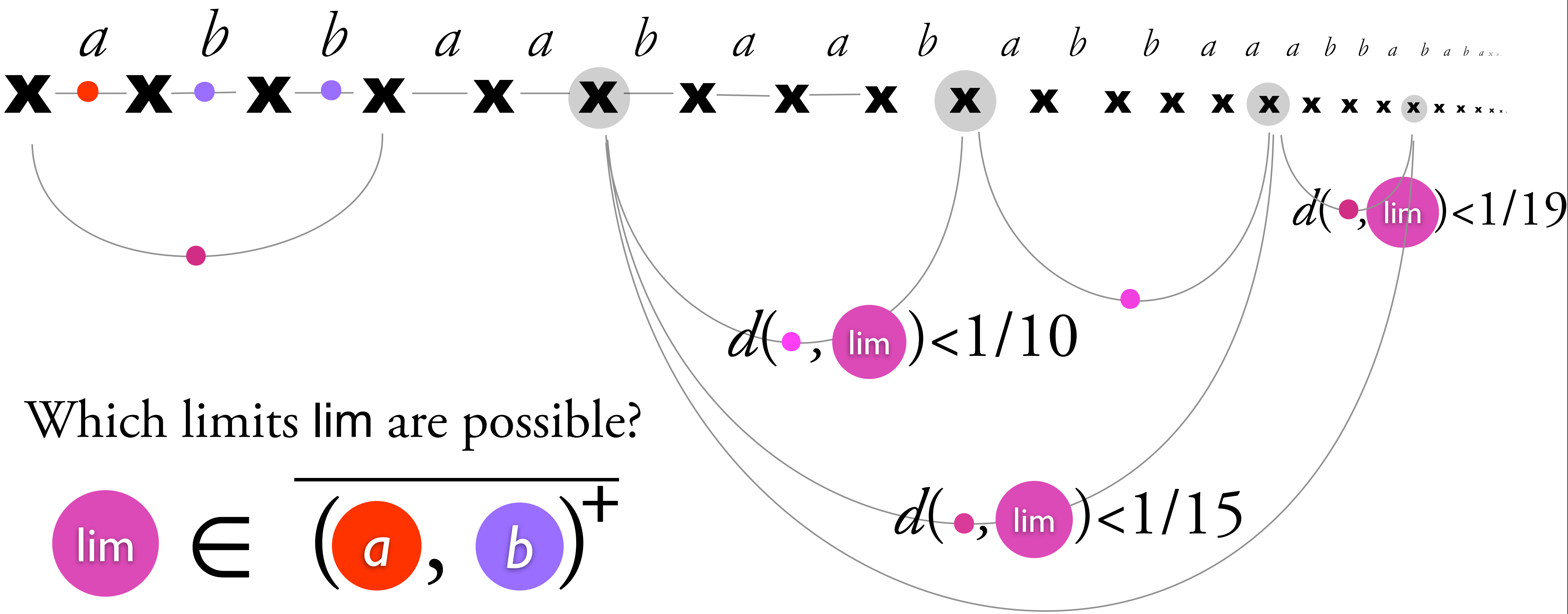
lim $\quad (a, b)^+$

Counter c does *not* converge to ∞

iff exists a counter d such that

lim$[d,d] = 0$ *and* lim$[d,c] < \infty$.

Emptiness of min-automata



Which limits lim are possible?

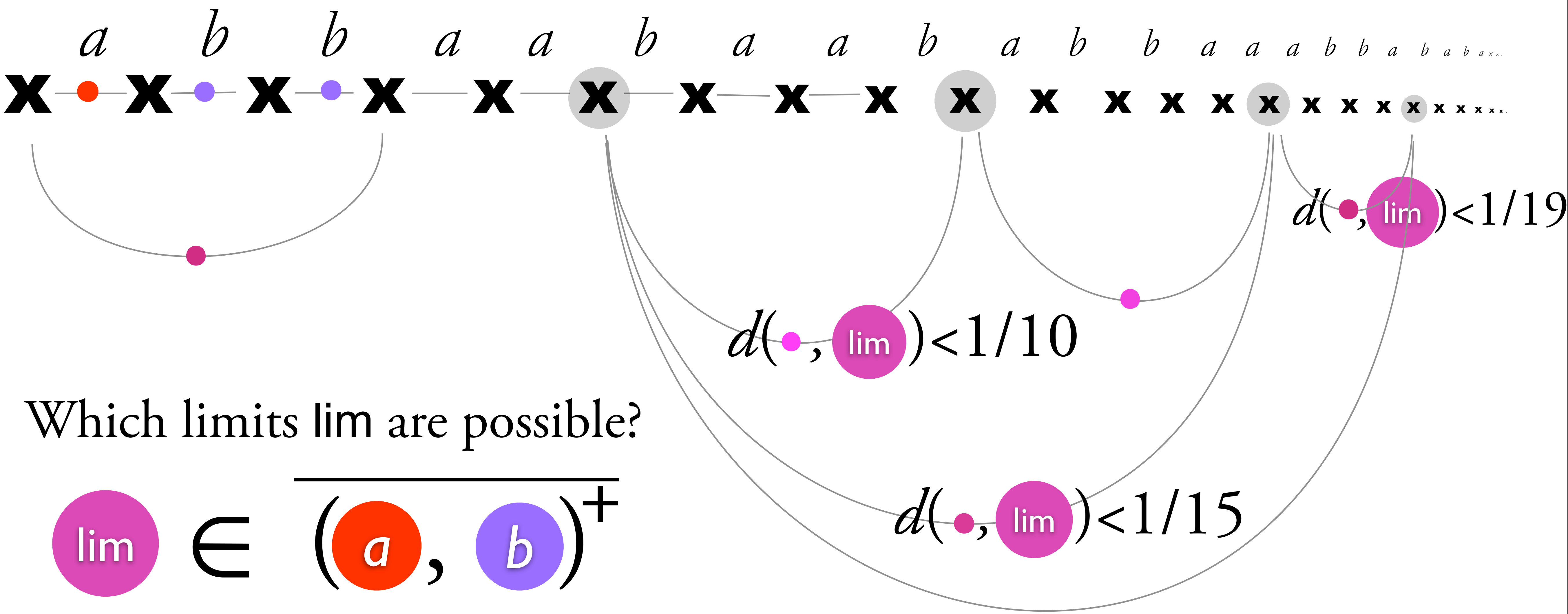
$$\mathrm{lim} \in \overline{(a, b)^{+}}$$

Counter c does *not* converge to ∞

iff exists a counter d such that

$\mathrm{lim}[d,d] = 0$ *and* $\mathrm{lim}[d,c] < \infty$.

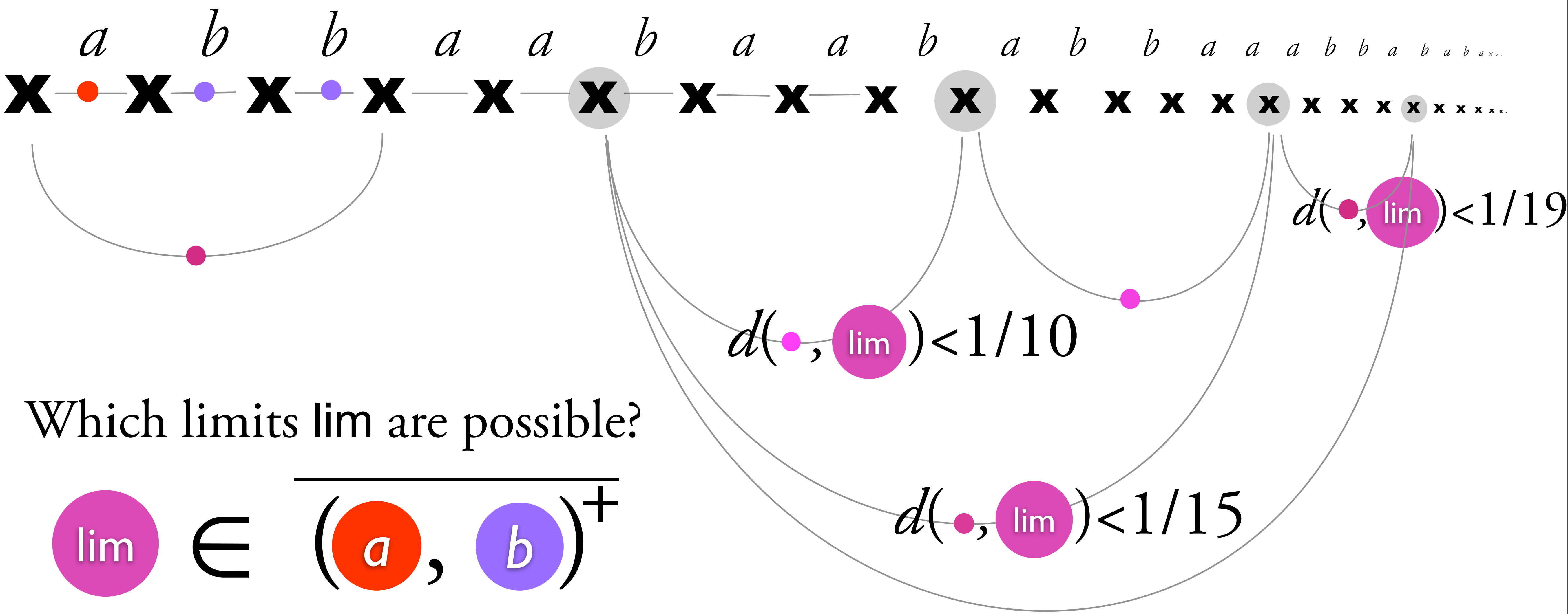
Emptiness of min-automata



Which limits lim are possible?

$$\mathrm{lim} \in \overline{(a, b)^+}$$

Emptiness of min-automata

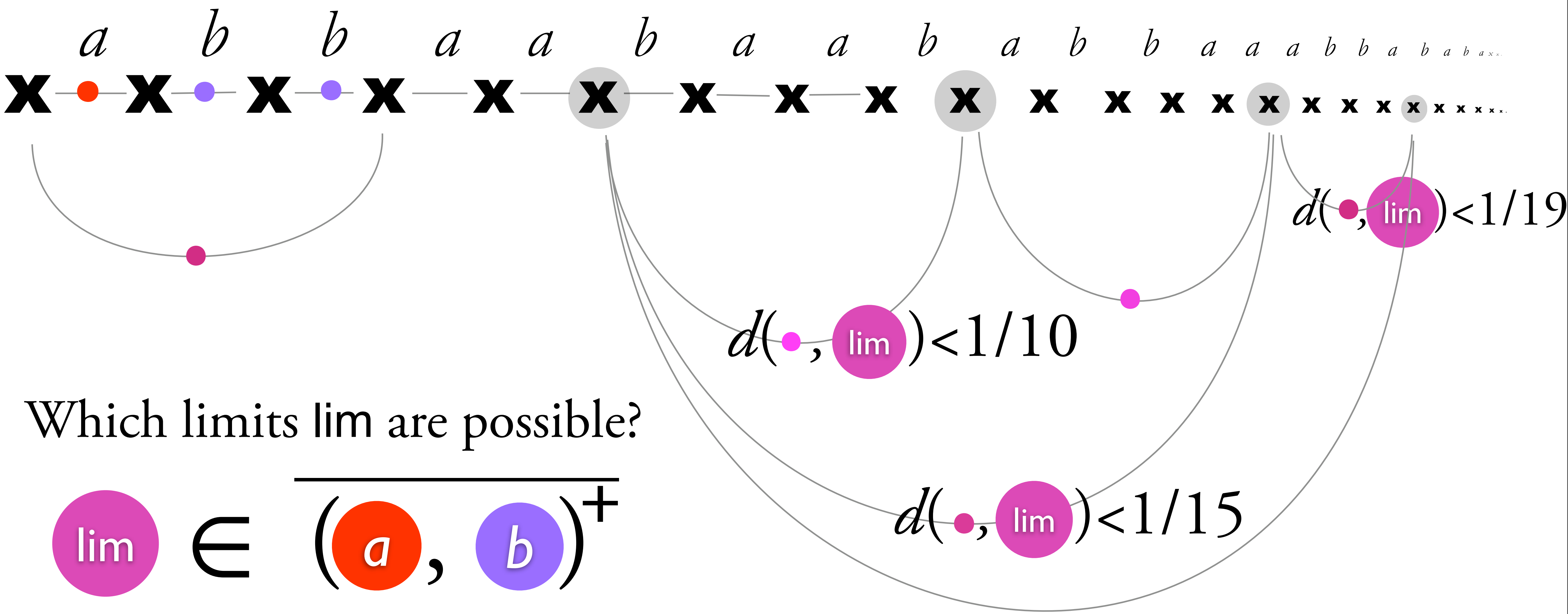


Which limits lim are possible?

$$\mathrm{lim} \in \overline{(a, b)^+}$$

Theorem.

Emptiness of min-automata

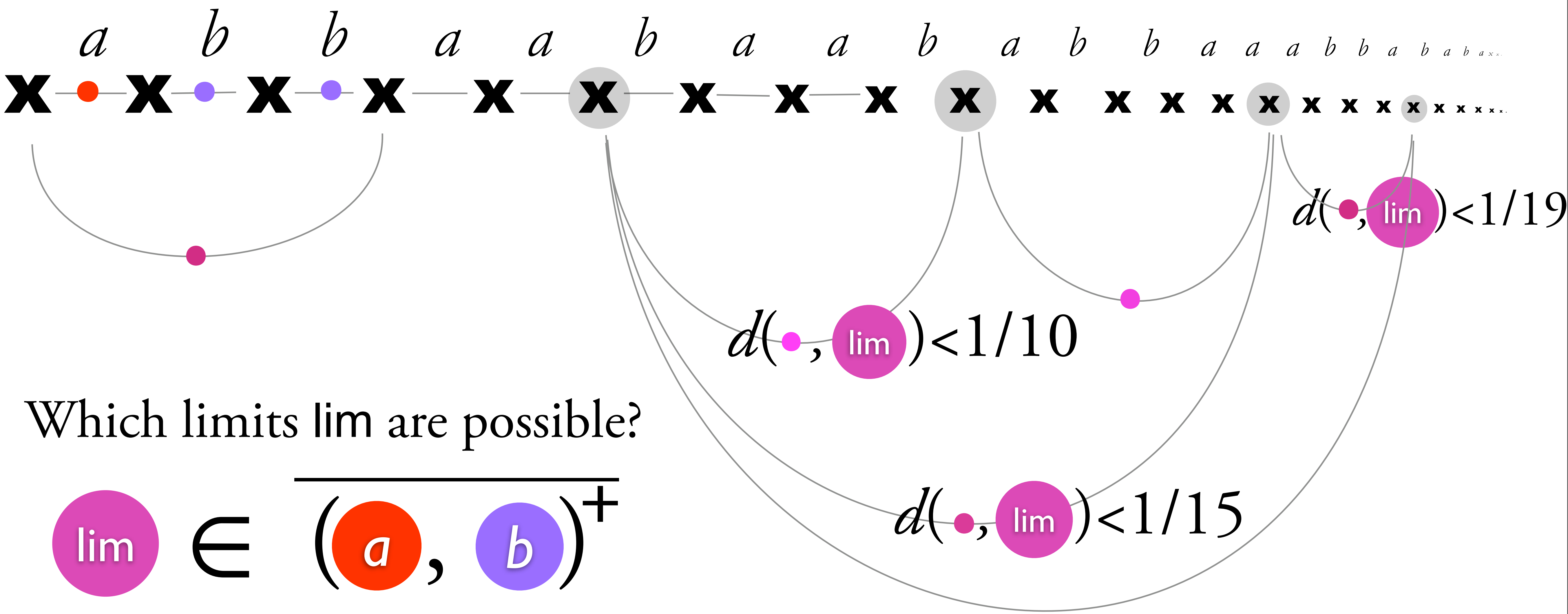


Which limits lim are possible?

$$\mathrm{lim} \in \overline{(a, b)^+}$$

Theorem. $\overline{(a, b)^+}$

Emptiness of min-automata

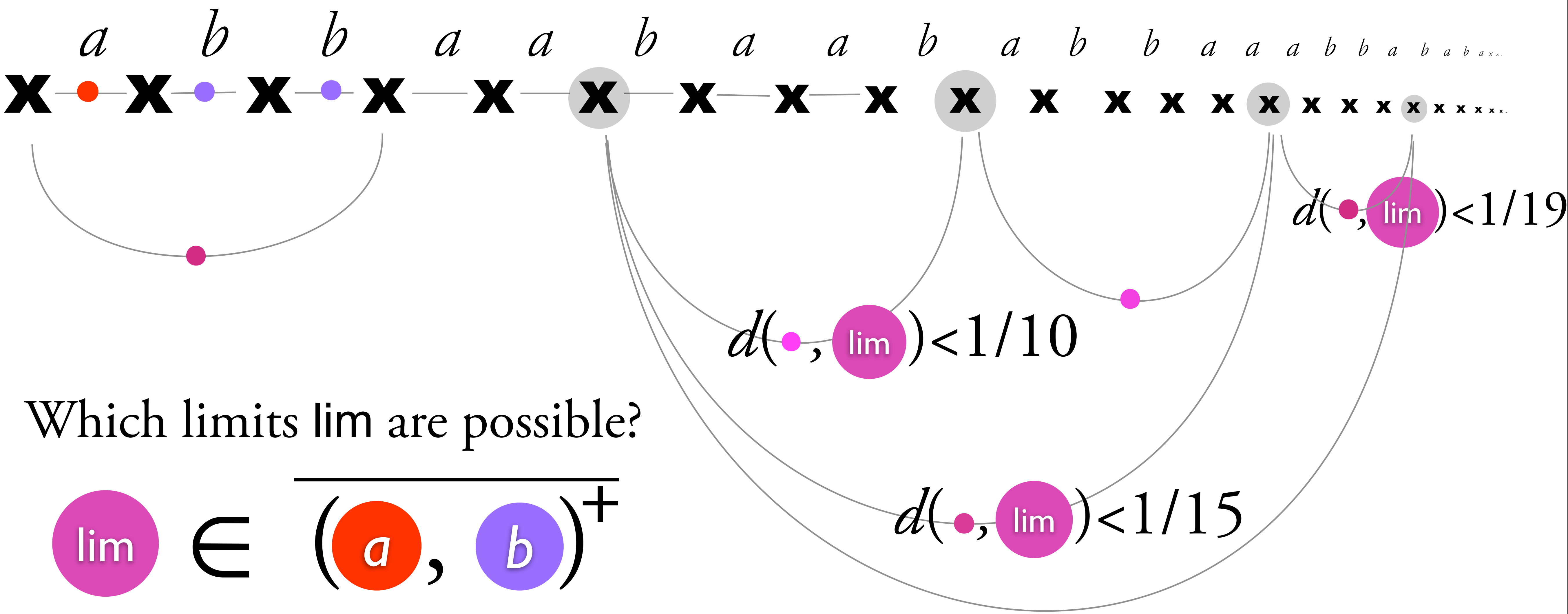


Which limits lim are possible?

$$\text{lim} \in \overline{(a, b)^{+}}$$

Theorem. $\overline{(a, b)^{+}} =$

Emptiness of min-automata

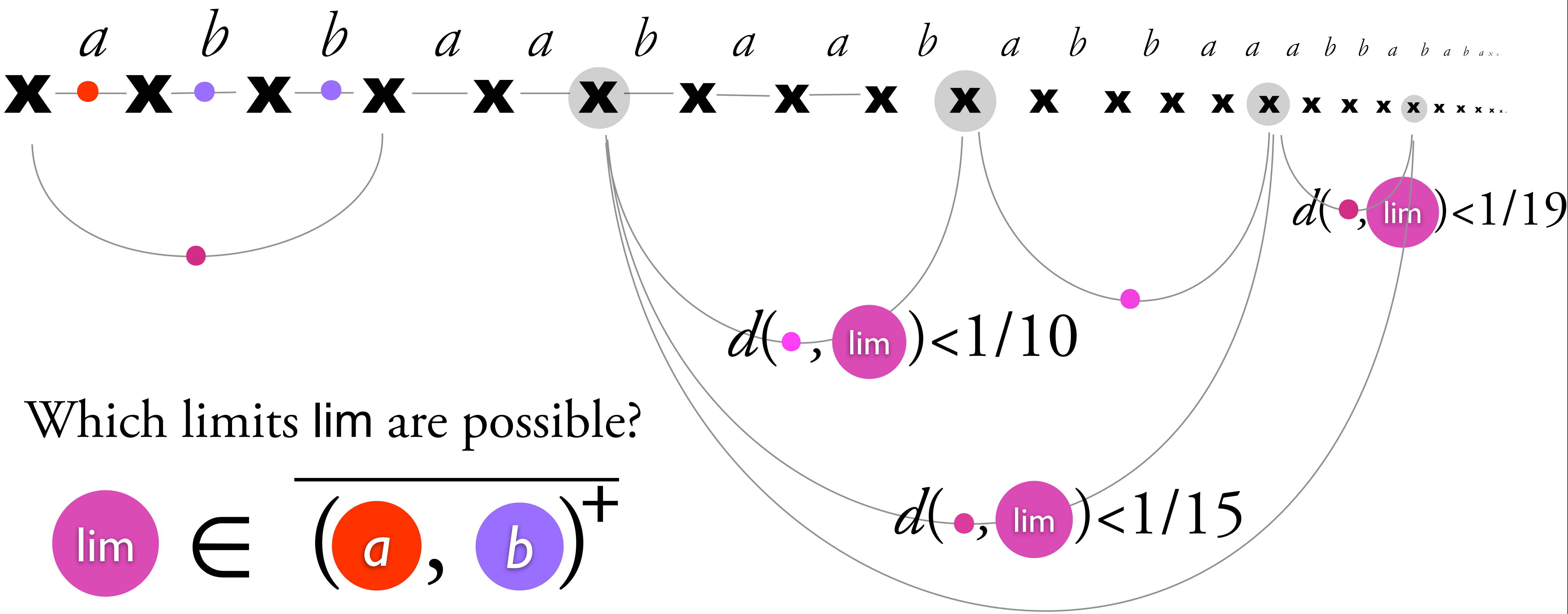


Which limits lim are possible?

$$\mathrm{lim} \in \overline{(a, b)^{+}}$$

Theorem. $\overline{(a, b)^{+}} = a \quad b$

Emptiness of min-automata

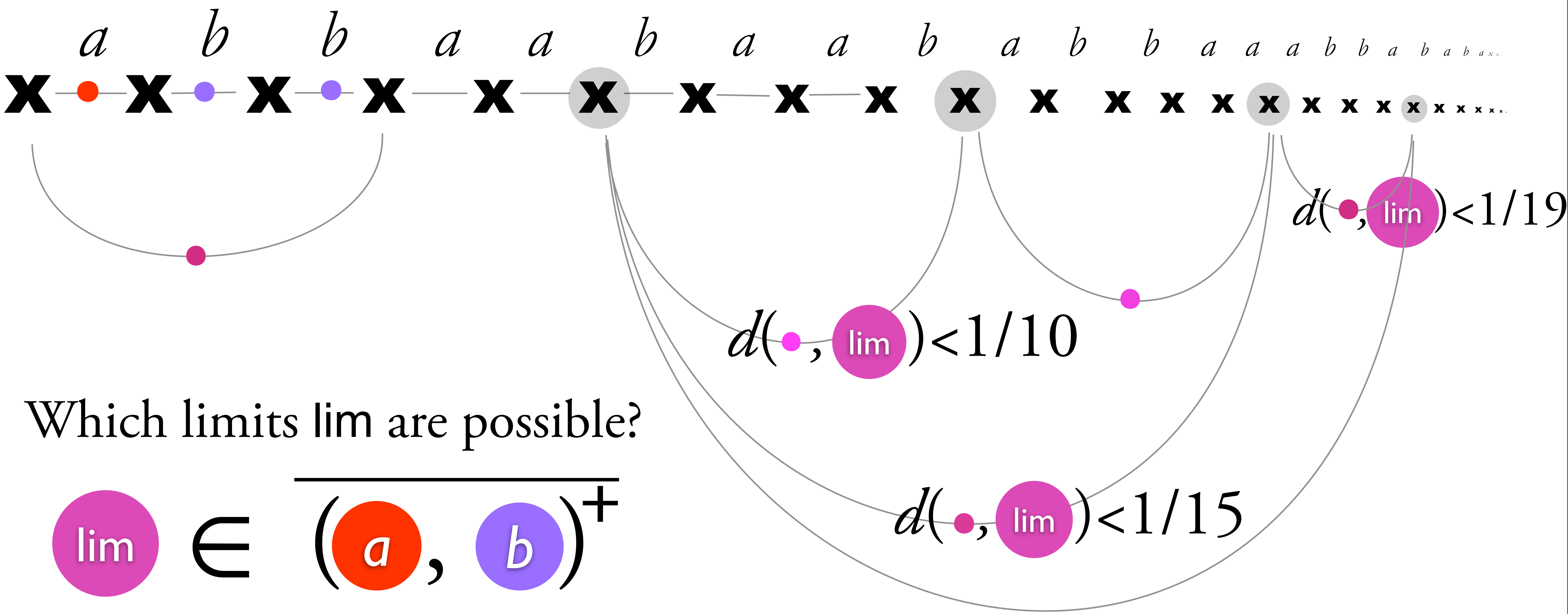


Which limits lim are possible?

$$\text{lim} \in \overline{(a, b)^+}$$

Theorem. $$\overline{(a, b)^+} = (a, b)^{+,\omega}$$

Emptiness of min-automata



Which limits lim are possible?

$$\text{lim} \in \overline{(a, b)^{+}}$$

Theorem. $$\overline{(a, b)^{+}} = (a, b)^{+,\omega}$$

if a, b are matrices over the (min,+)-semiring.

Simon's Factorization Theorem

for semigroups with stabilization

Simon's Factorization Theorem

for semigroups with stabilization

semigroup with stabilization

Simon's Factorization Theorem

for semigroups with stabilization

$$(S , \cdot , {}^{\#})$$

semigroup with stabilization

Simon's Factorization Theorem

for semigroups with stabilization

$$(\underbrace{S, \cdot}, {}^{\#})$$

semigroup with stabilization

Simon's Factorization Theorem

for semigroups with stabilization

$$(\underbrace{S, \cdot}, \ ^{\#})$$

semigroup with stabilization

Simon's Factorization Theorem

for semigroups with stabilization

$$(\underbrace{S\,,\,\cdot}\,,\,{}^{\#})$$

semigroup with stabilization

- $s^{\#} = (s^{n})^{\#}$ *for* n=1,2,3,...

Simon's Factorization Theorem

for semigroups with stabilization

$$(\underbrace{S\,,\,\cdot}_{\text{semigroup with stabilization}}\,,\ {}^{\#})$$

semigroup with stabilization

- $s^{\#} = (s^{n})^{\#}$ *for* n=1,2,3,...
- $(s\ t)^{\#}\, s = s\,(t\ s)^{\#}$

Simon's Factorization Theorem

for semigroups with stabilization

$$(\underbrace{S\ ,\ \cdot}_{}\ ,\ {}^{\#})$$

semigroup with stabilization

- $s^{\#} = (s^{n})^{\#}$ *for* n=1,2,3,...
- $(s\ t)^{\#}\, s = s\ (t\ s)^{\#}$
- $s^{\#} s^{\#} = s^{\#}$

Simon's Factorization Theorem

for semigroups with stabilization

$$(\underbrace{S\, , \,\cdot}_{} \, , \; ^{\#})$$

semigroup with stabilization

- $s^{\#} = (s^{n})^{\#}$ *for* n=1,2,3,...
- $(s\,t)^{\#}\, s = s\,(t\,s)^{\#}$
- $s^{\#}\, s^{\#} = s^{\#}$
- $e^{\#}\, e = e^{\#}$ *if e is idempotent*

Simon's Factorization Theorem

for semigroups with stabilization

$$(\underbrace{S\,,\,\cdot}\,,\ ^{\#})$$

semigroup with stabilization

- $s^{\#} = (s^{n})^{\#}$ *for* n=1,2,3,...
- $(s\,t)^{\#}\,s = s\,(t\,s)^{\#}$
- $s^{\#}\,s^{\#} = s^{\#}$
- $e^{\#}\,e = e^{\#}$ *if e is idempotent*

$e = e^{\#}$ *if e is idempotent*

Simon's Factorization Theorem

for semigroups with stabilization

$$(\underbrace{S , \cdot}_{\text{semigroup with stabilization}} , \ {}^{\#})$$

semigroup with stabilization

- $s^{\#} = (s^{n})^{\#}$ *for n*=1,2,3,...
- $(s\,t)^{\#}\, s = s\,(t\,s)^{\#}$
- $s^{\#}\, s^{\#} = s^{\#}$
- $e^{\#}\, e = e^{\#}$ *if e is idempotent*

~~$e = e^{\#}$ *if e is idempotent*~~

Simon's Factorization Theorem

for semigroups with stabilization

$$(\underbrace{S , \cdot}_{} , {}^{\#})$$

semigroup with stabilization

- $s^{\#} = (s^{n})^{\#}$ *for* $n = 1,2,3,\ldots$
- $(s\,t)^{\#}\,s = s\,(t\,s)^{\#}$
- $s^{\#}\,s^{\#} = s^{\#}$
- $e^{\#}\,e = e^{\#}$ *if e is idempotent*

$\cancel{e = e^{\#}}$ ~~*if e is idempotent*~~

Example 1 (infinite)

$(\{0,1,2,\ldots,\infty\}, +, {}^{\omega}),$

$0^{\omega} = 0, \quad 1^{\omega} = 2^{\omega} = \ldots = \infty$

Simon's Factorization Theorem

for semigroups with stabilization

$$(\underbrace{S , \cdot}_{} , {}^{\#})$$

semigroup with stabilization

- $s^{\#} = (s^{n})^{\#}$ *for* $n = 1,2,3,\ldots$
- $(s\,t)^{\#} s = s\,(t\,s)^{\#}$
- $s^{\#} s^{\#} = s^{\#}$
- $e^{\#} e = e^{\#}$ *if e is idempotent*

$\cancel{e = e^{\#}}$ ~~*if e is idempotent*~~

Example 1 (infinite)
$(\{0,1,2,\ldots,\infty\}, +, {}^{\omega})$,
$0^{\omega} = 0, \quad 1^{\omega} = 2^{\omega} = \ldots = \infty$

Example 2 (finite)
$(\{0,1,\infty\}, +, {}^{\#})$,
$1+1=1, \quad 0^{\#} = 0, \quad 1^{\#} = \infty^{\#} = \infty$

Simon's Factorization Theorem

for semigroups with stabilization

$$(\underbrace{S \,,\, \cdot}_{} \,,\, ^{\#})$$

semigroup with stabilization

- $s^{\#} = (s^{n})^{\#}$ *for* $n=1,2,3,\ldots$
- $(s\,t)^{\#}\,s = s\,(t\,s)^{\#}$
- $s^{\#}\,s^{\#} = s^{\#}$
- $e^{\#}\,e = e^{\#}$ *if e is idempotent*

~~$e = e^{\#}$ *if e is idempotent*~~

Example 1 (infinite)
$(\{0,1,2,\ldots,\infty\}, +, {}^{\omega})$,
$0^{\omega} = 0, \quad 1^{\omega} = 2^{\omega} = \ldots = \infty$

$$\xrightarrow{1,2,3\ldots \to 1}$$

Example 2 (finite)
$(\{0,1,\infty\}, +, {}^{\#})$,
$1+1=1, \quad 0^{\#} = 0, \quad 1^{\#} = \infty^{\#} = \infty$

Simon's Factorization Theorem

for semigroups with stabilization

Example 2 (finite)

$(\{0,1,\infty\}, +, ^{\#})$,

$1+1=1, \quad 0^{\#}=0, \quad 1^{\#}=\infty^{\#}=\infty$

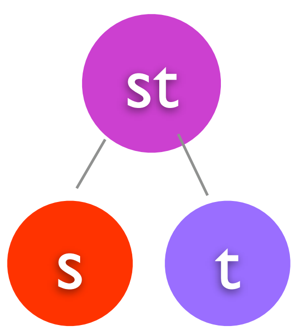
Simon's Factorization Theorem

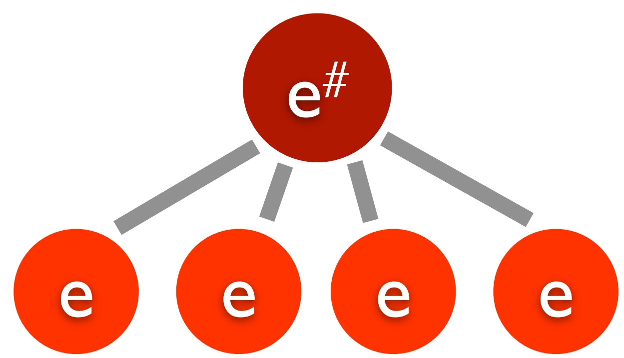
for semigroups with stabilization

Factorization tree of word $w \in S^+$

Use the two rules to construct tree:

binary rule idempotent rule





Example 2 (finite)

$(\{0,1,\infty\}, +, {}^{\#})$,

$1+1=1, \quad 0^{\#}=0, \quad 1^{\#}=\infty^{\#}=\infty$

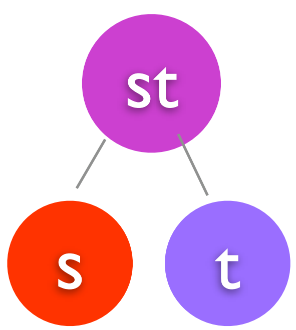
Simon's Factorization Theorem

for semigroups with stabilization

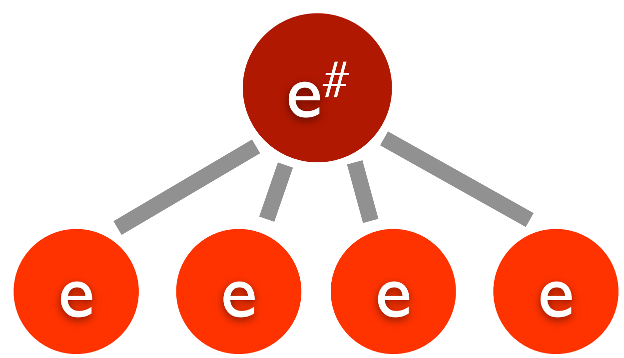
Factorization tree of word $w \in S^+$

Use the two rules to construct tree:

binary rule



idempotent rule



Example 2 (finite)

$(\{0, 1, \infty\}, +, {}^{\#})$,

$1+1=1, \quad 0^{\#} = 0, \quad 1^{\#} = \infty^{\#} = \infty$

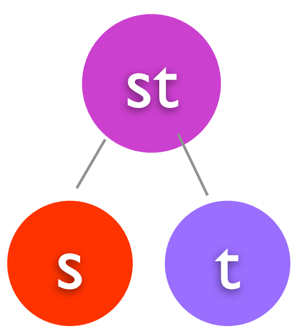
Simon's Factorization Theorem

for semigroups with stabilization

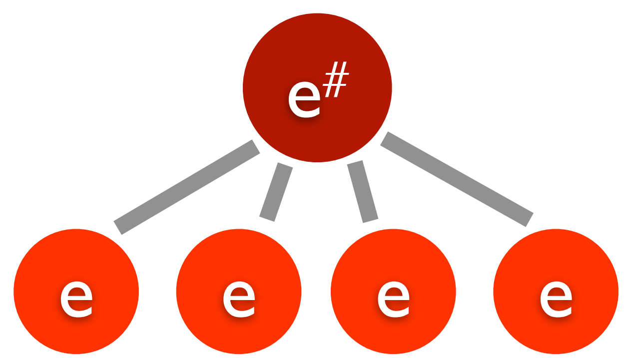
Factorization tree of word $w \in S^+$

Use the two rules to construct tree:

binary rule



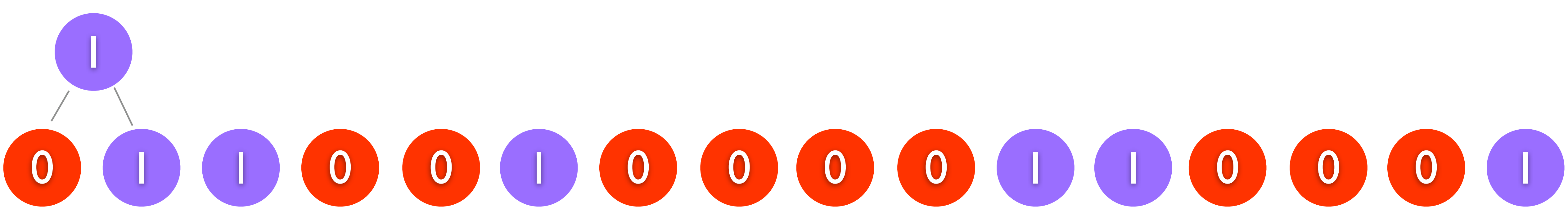
idempotent rule



Example 2 (finite)

$(\{0,1,\infty\}, +, ^{\#})$,

$1+1=1$, $0^{\#} = 0$, $1^{\#} = \infty^{\#} = \infty$



Simon's Factorization Theorem

for semigroups with stabilization

Factorization tree of word $w \in S^+$

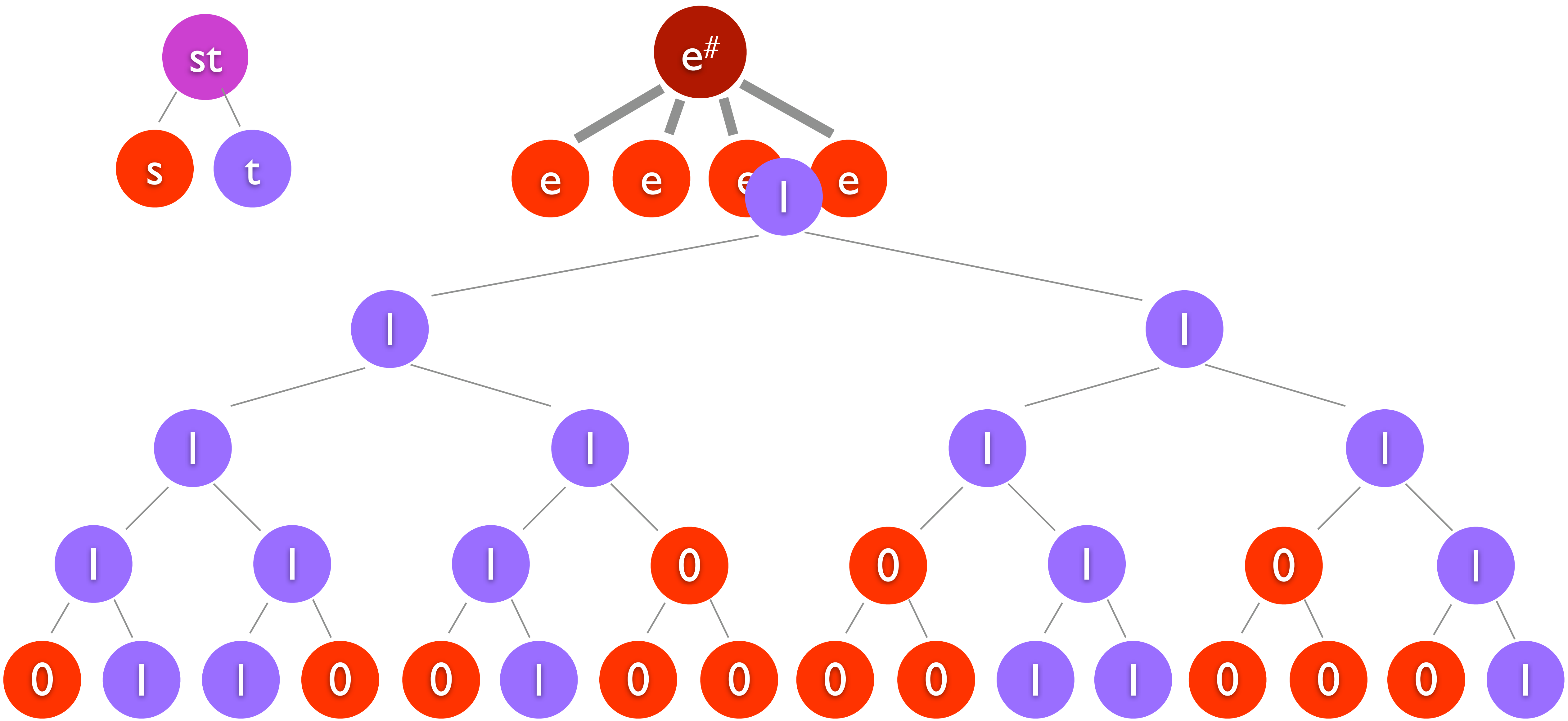
Use the two rules to construct tree:

binary rule idempotent rule

Example 2 (finite)

$(\{0,1,\infty\}, +, {}^{\#})$,

$1+1=1, \quad 0^{\#}=0, \quad 1^{\#}=\infty^{\#}=\infty$



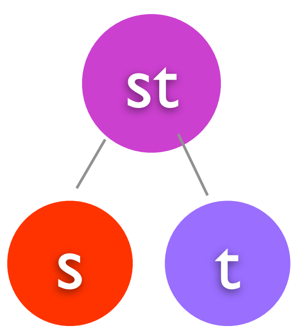
Simon's Factorization Theorem

for semigroups with stabilization

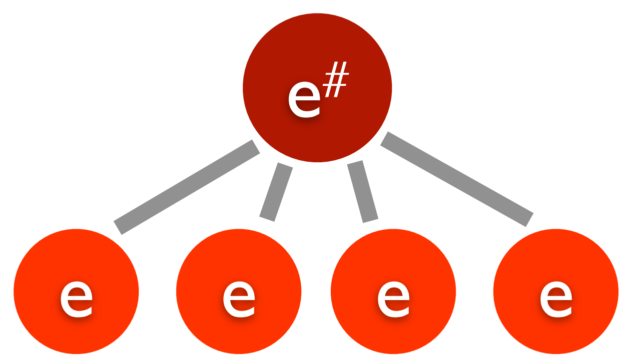
Factorization tree of word $w \in S^+$

Use the two rules to construct tree:

binary rule



idempotent rule



Example 2 (finite)

$(\{0,1,\infty\}, +, {}^{\#})$,

$1+1=1$, $0^{\#}=0$, $1^{\#}=\infty^{\#}=\infty$

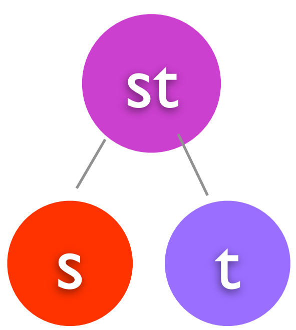
Simon's Factorization Theorem

for semigroups with stabilization

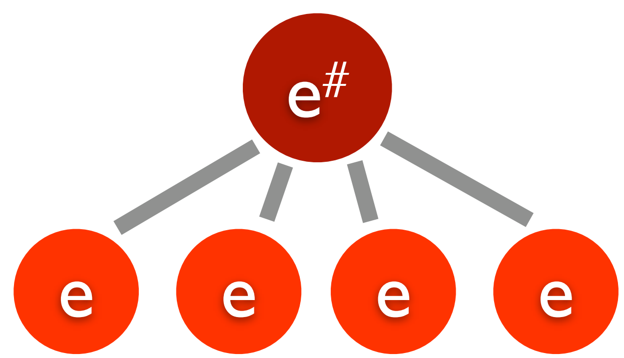
Factorization tree of word $w \in S^+$

Use the two rules to construct tree:

binary rule



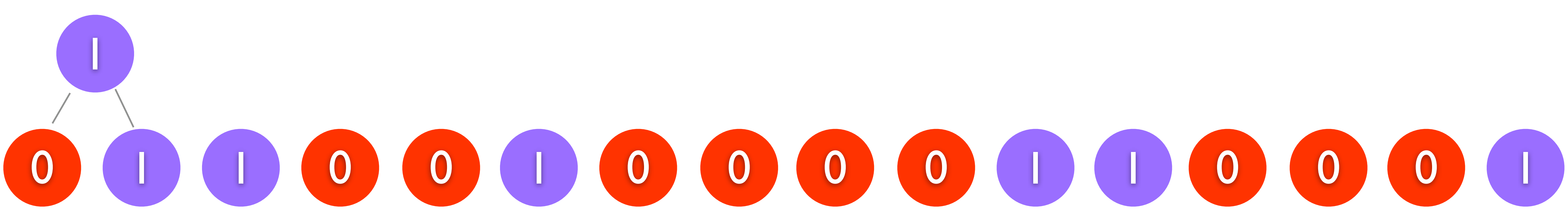
idempotent rule



Example 2 (finite)

$(\{0, 1, \infty\}, +, {}^{\#})$,

$1+1=1$, $0^{\#} = 0$, $1^{\#} = \infty^{\#} = \infty$



Simon's Factorization Theorem

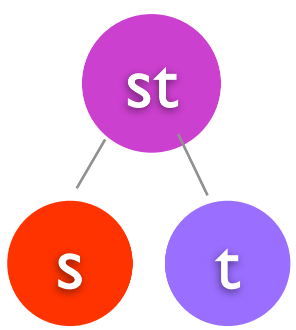
for semigroups with stabilization

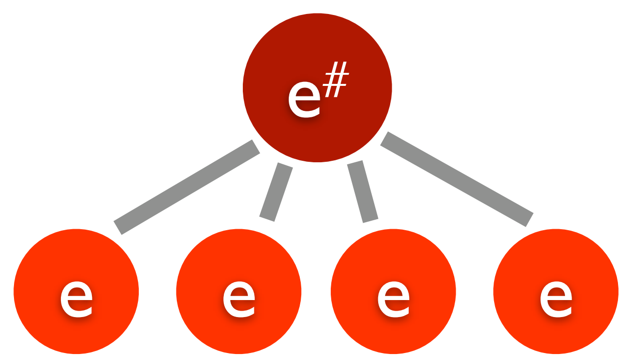
Factorization tree of word $w \in S^+$

Use the two rules to construct tree:

binary rule

idempotent rule

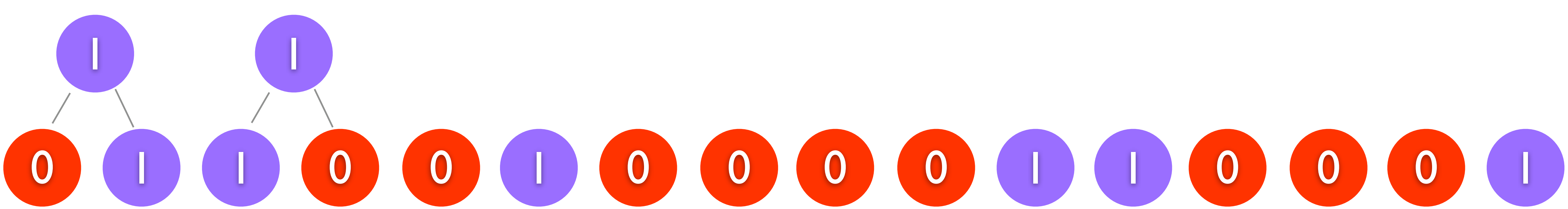




Example 2 (finite)

$(\{0,1,\infty\}, +, {}^{\#})$,

$1+1=1$, $0^{\#}=0$, $1^{\#}=\infty^{\#}=\infty$



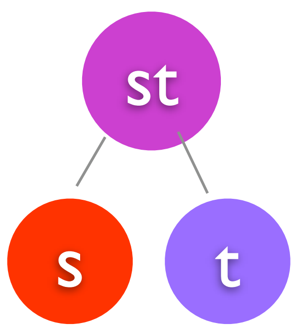
Simon's Factorization Theorem

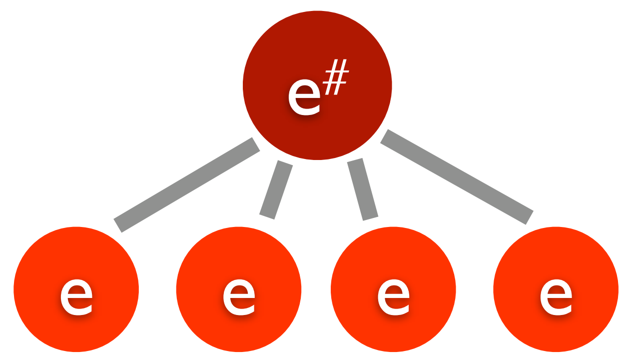
for semigroups with stabilization

Factorization tree of word $w \in S^+$

Use the two rules to construct tree:

binary rule idempotent rule

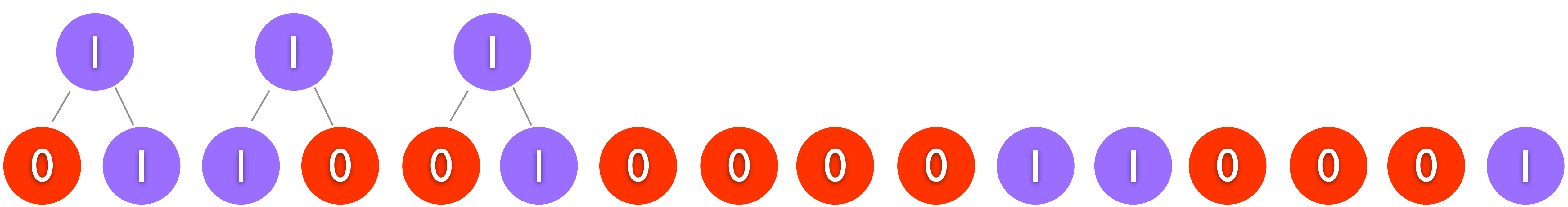




Example 2 (finite)

$(\{0, 1, \infty\}, +, ^{\#})$,

$1+1=1$, $0^{\#}=0$, $1^{\#}=\infty^{\#}=\infty$



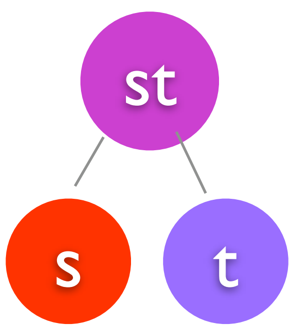
Simon's Factorization Theorem

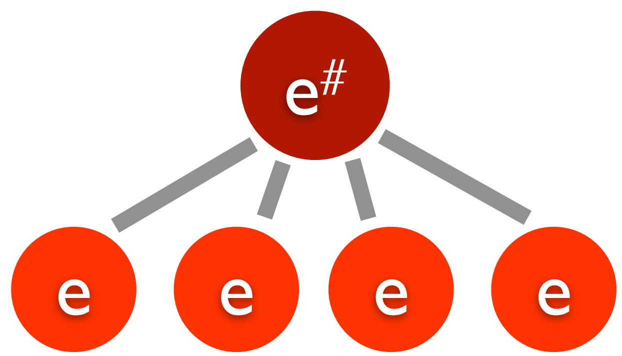
for semigroups with stabilization

Factorization tree of word $w \in S^+$

Use the two rules to construct tree:

binary rule idempotent rule

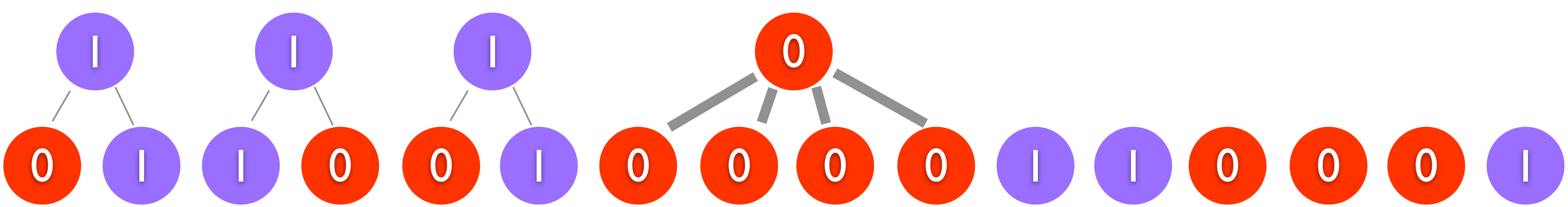




Example 2 (finite)

$(\{0, 1, \infty\}, +, {}^\#)$,

$1+1=1, \quad 0^\# = 0, \quad 1^\# = \infty^\# = \infty$



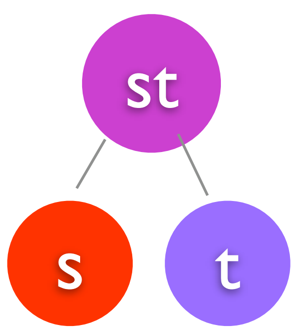
Simon’s Factorization Theorem

for semigroups with stabilization

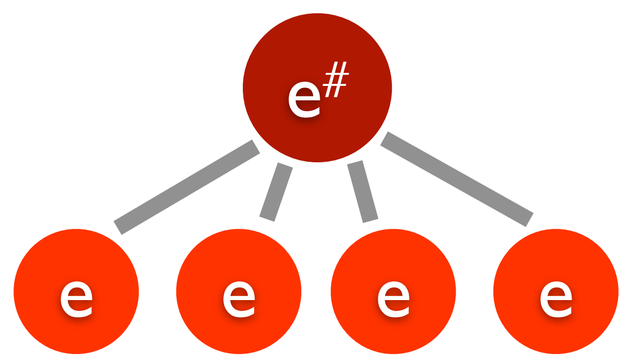
Factorization tree of word $w \in S^+$

Use the two rules to construct tree:

binary rule



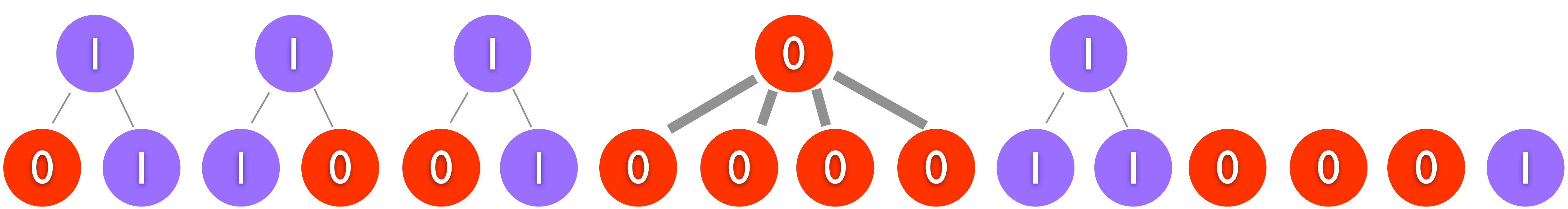
idempotent rule



Example 2 (finite)

$(\{0,1,\infty\}, +, {}^{\#})$,

$1+1=1, \quad 0^{\#}=0, \quad 1^{\#}=\infty^{\#}=\infty$



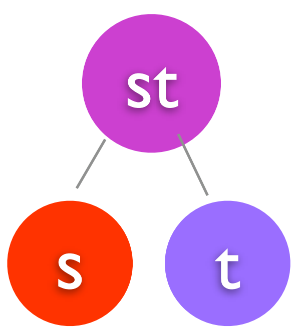
Simon's Factorization Theorem

for semigroups with stabilization

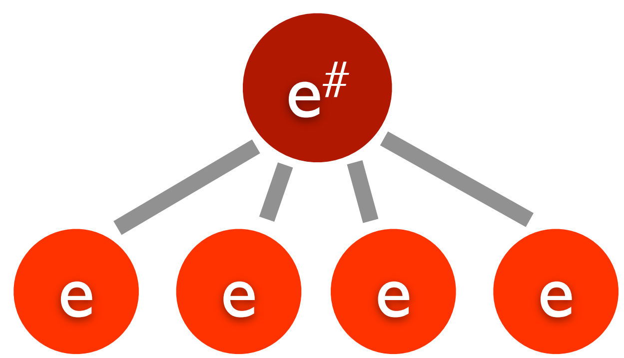
Factorization tree of word $w \in S^+$

Use the two rules to construct tree:

binary rule



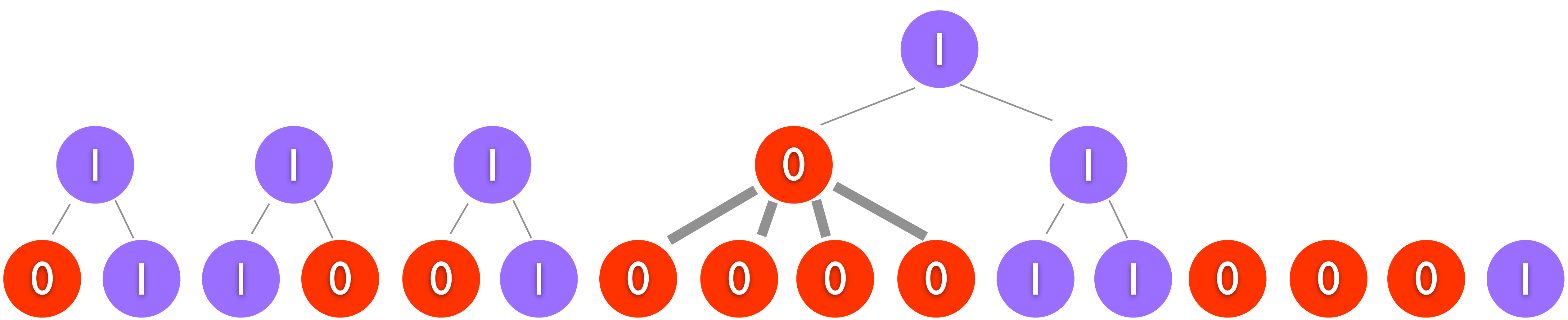
idempotent rule



Example 2 (finite)

$(\{0,1,\infty\}, +, {}^{\#})$,

$1+1=1, \quad 0^{\#}=0, \quad 1^{\#}=\infty^{\#}=\infty$



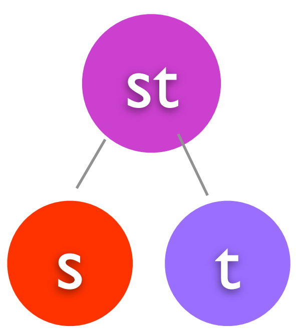
Simon's Factorization Theorem

for semigroups with stabilization

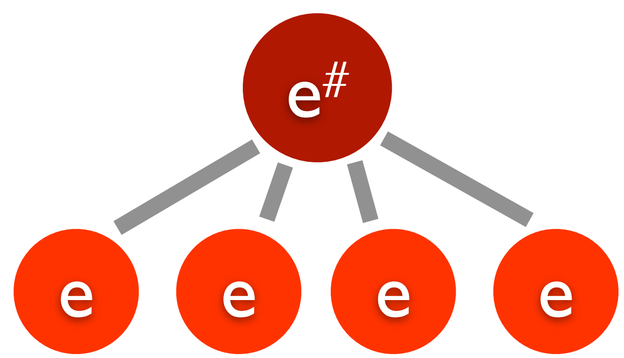
Factorization tree of word $w \in S^+$

Use the two rules to construct tree:

binary rule



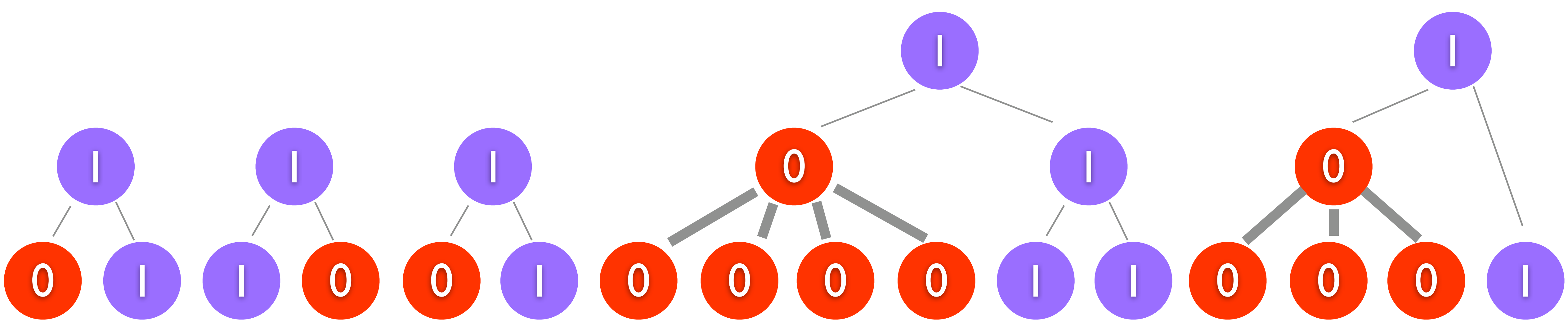
idempotent rule



Example 2 (finite)

$(\{0,1,\infty\}, +, {}^{\#})$,

$1+1=1, \quad 0^{\#}=0, \quad 1^{\#}=\infty^{\#}=\infty$



Simon's Factorization Theorem

for semigroups with stabilization

Factorization tree of word $w \in S^+$

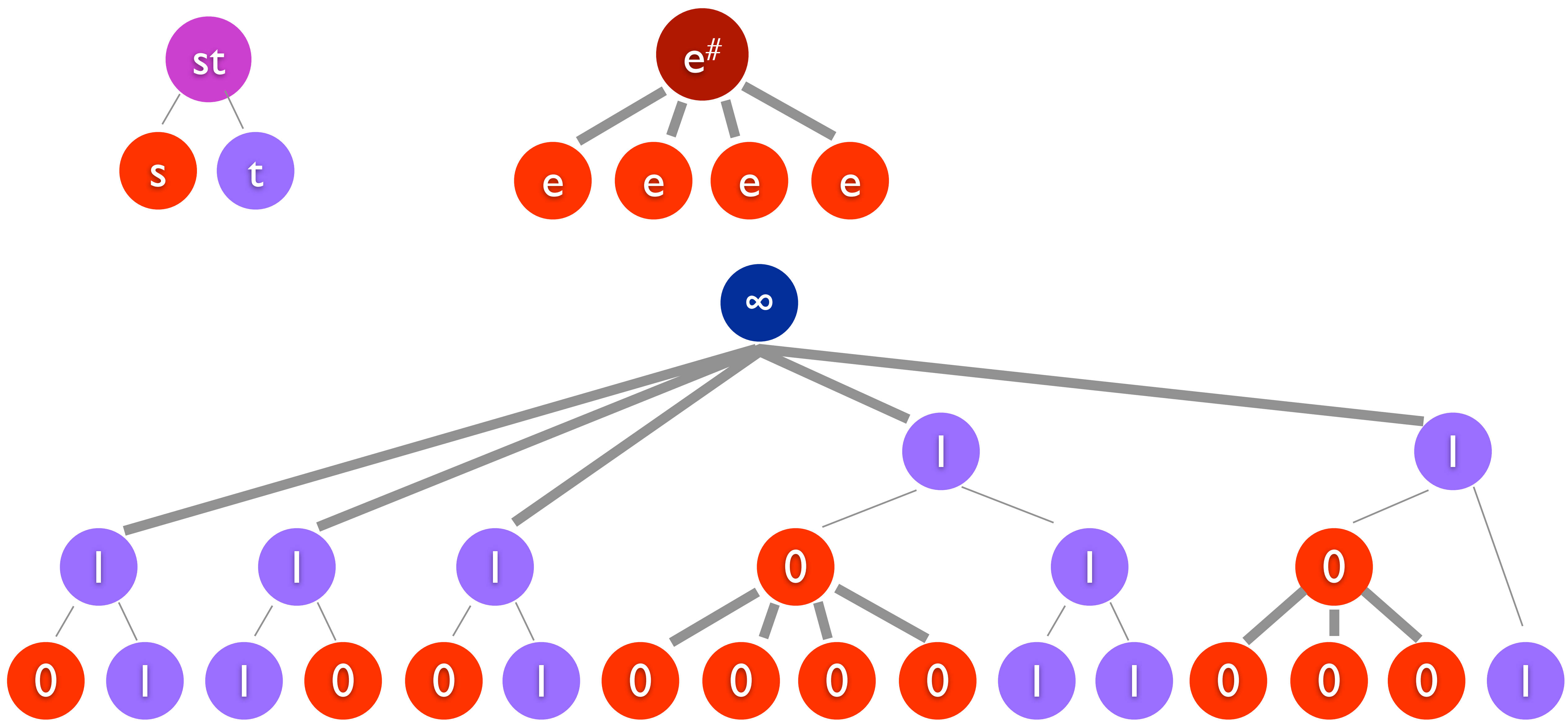
Use the two rules to construct tree:

binary rule idempotent rule

Example 2 (finite)

$(\{0,1,\infty\}, +, {}^{\#})$,

$1+1=1$, $0^{\#}=0$, $1^{\#}=\infty^{\#}=\infty$



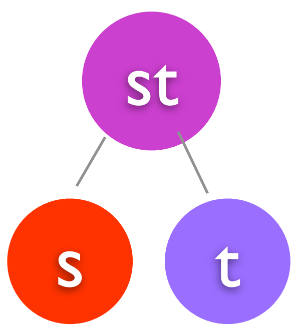
Simon's Factorization Theorem

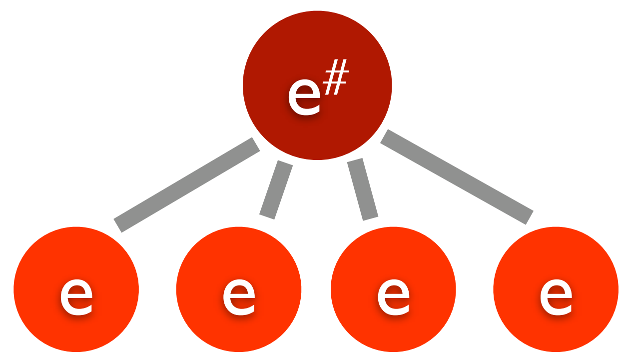
for semigroups with stabilization

Factorization tree of word $w \in S^+$

Use the two rules to construct tree:

binary rule idempotent rule





Example 2 (finite)

$(\{0,1,\infty\}, +, {}^{\#})$,

$1+1=1$, $0^{\#}=0$, $1^{\#} = \infty^{\#} = \infty$

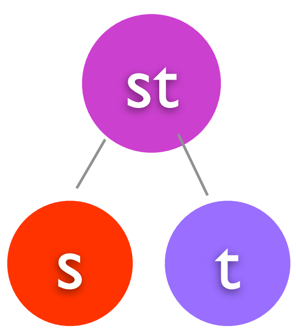
Simon's Factorization Theorem

for semigroups with stabilization

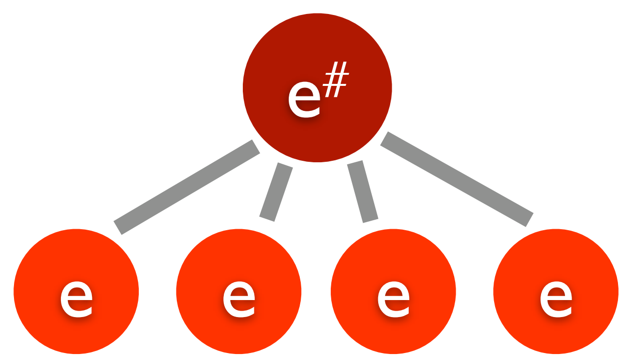
Factorization tree of word $w \in S^+$

Use the two rules to construct tree:

binary rule



idempotent rule



Example 2 (finite)

$(\{0,1,\infty\}, +, {}^{\#})$,

$1+1=1$, $0^{\#}=0$, $1^{\#}=\infty^{\#}=\infty$

Theorem. For any finite stabilization semigroup S and word $w \in S^+$ there exists a factorization tree over w of height $\leq 9|S|^2$.

Thank you for your attention!