A robust extension of ω-regular word languages.

Mikołaj Bojańczyk
Warsaw University

What is a regular word language?

– regular expressions
– automata
– monadic second-order logic
– closure properties
– semigroups
– Myhill-Nerode equivalence

What is a regular word language?

- regular expressions
- automata
- monadic second-order logic
- closure properties
- semigroups
- Myhill-Nerode equivalence

What about infinite words?

What is a regular word language?

- regular expressions
- automata
- monadic second-order logic
- closure properties
- semigroups
- Myhill-Nerode equivalence

What about infinite words?

ω-regular languages

What is a regular word language?

- regular expressions
- automata
- monadic second-order logic
- closure properties
- semigroups
- Myhill-Nerode equivalence

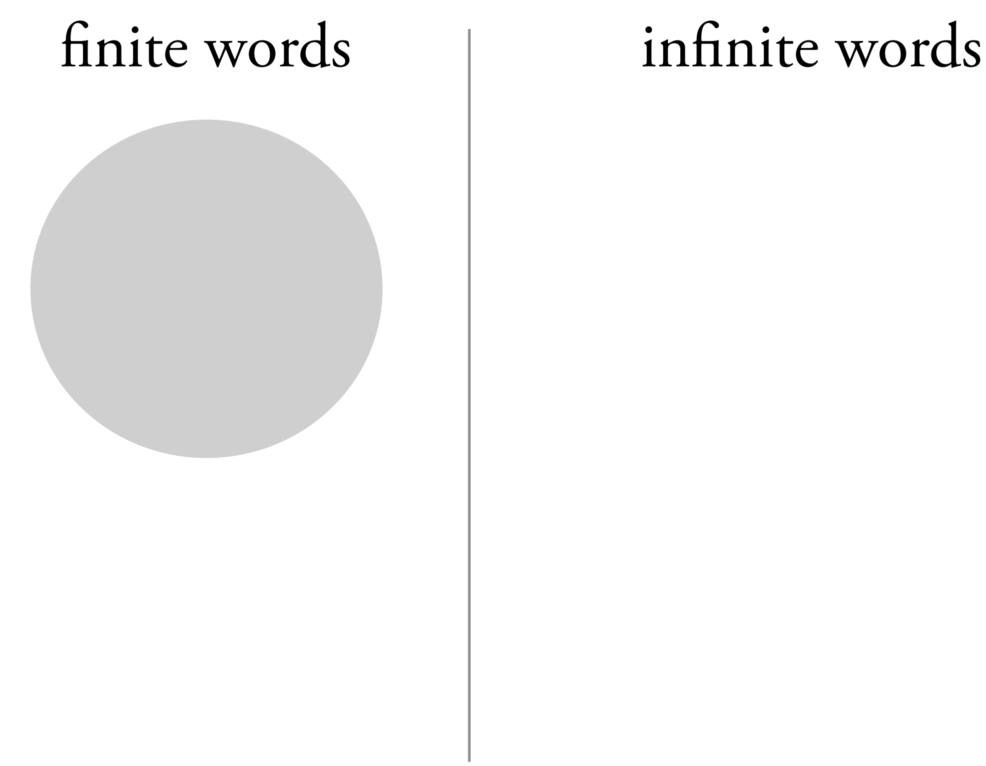
What about infinite words?

ω-regular languages

Claim: there are robust extensions of ω-regular languages

What is a regular word language?

- regular expressions
- automata
- monadic second-order logic
- closure properties
- semigroups
- Myhill-Nerode equivalence



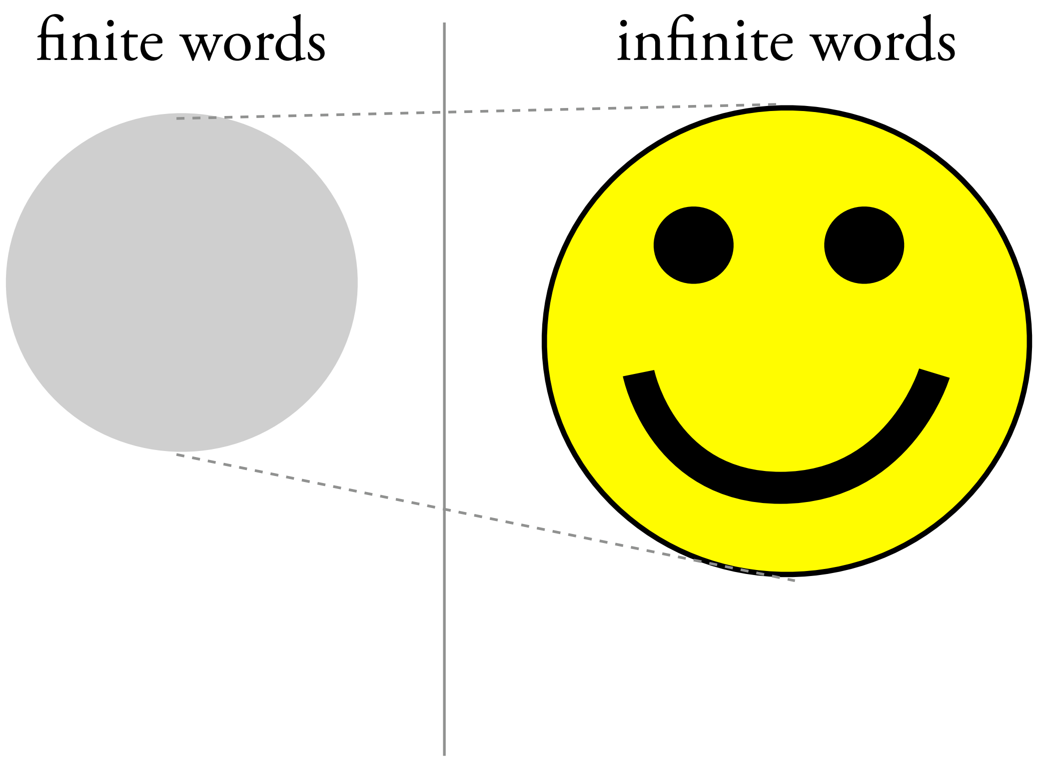
What about infinite words?

ω-regular languages

Claim: there are robust extensions of ω-regular languages

What is a regular word language?

- regular expressions
- automata
- monadic second-order logic
- closure properties
- semigroups
- Myhill-Nerode equivalence



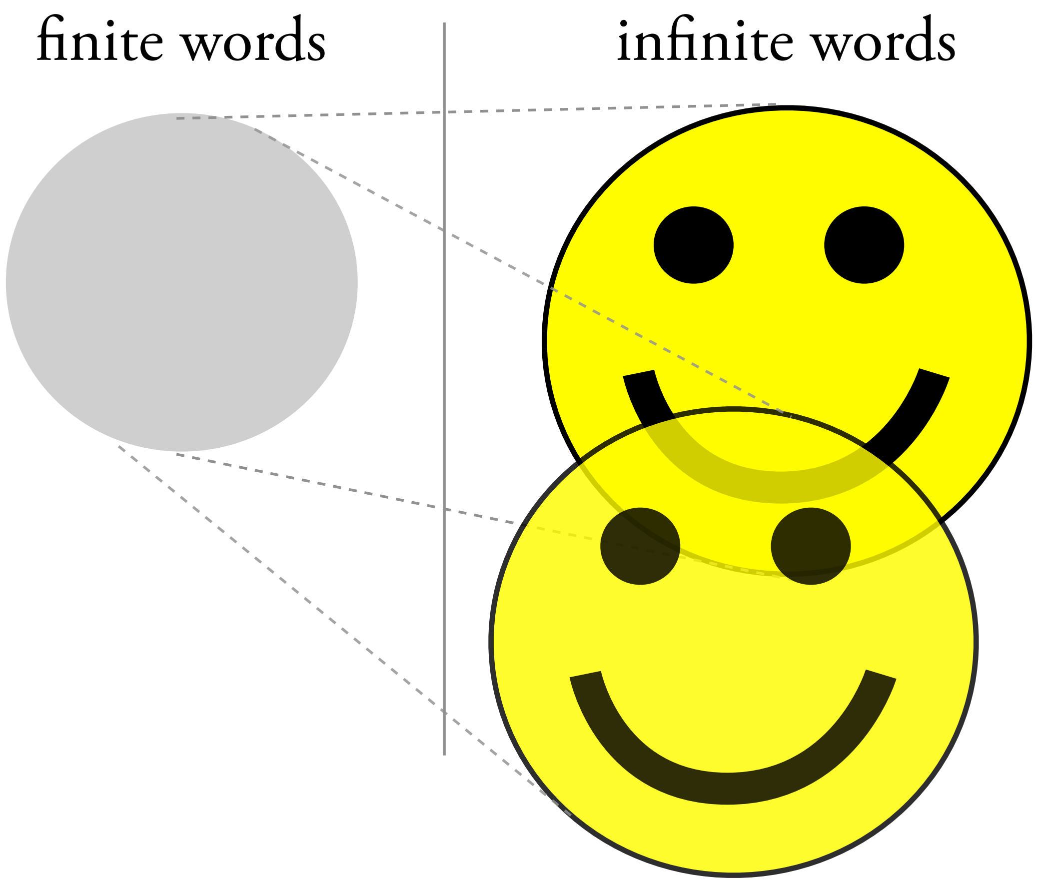
What about infinite words?

ω-regular languages

Claim: there are robust extensions of ω-regular languages

What is a regular word language?

- regular expressions
- automata
- monadic second-order logic
- closure properties
- semigroups
- Myhill-Nerode equivalence



What about infinite words?

ω-regular languages

Claim: there are robust extensions of ω-regular languages

A word $a^{n_1} b\, a^{n_2} b\, a^{n_3} b$... describes a number sequence $n_1\, n_2\, n_3$...

A word $a^{n_1} b\ a^{n_2} b\ a^{n_3} b\ldots$ describes a number sequence $n_1\ n_2\ n_3\ldots$

Which properties of number sequences are regular?

A word $a^{n_1} b\, a^{n_2} b\, a^{n_3} b\ldots$ describes a number sequence $n_1\, n_2\, n_3\ldots$

Which properties of number sequences are regular?

– odd numbers on even positions

A word $a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots$ describes a number sequence $n_1 n_2 n_3 \ldots$

Which properties of number sequences are regular?

– odd numbers on even positions
– infinitely many odd numbers

A word $a^{n_1} b\, a^{n_2} b\, a^{n_3} b\ldots$ describes a number sequence $n_1\, n_2\, n_3\ldots$

Which properties of number sequences are regular?

– odd numbers on even positions
– infinitely many odd numbers
– the first number n_1 is prime

A word $a^{n_1} b\, a^{n_2} b\, a^{n_3} b\ldots$ describes a number sequence $n_1\, n_2\, n_3\ldots$

Which properties of number sequences are regular?

– odd numbers on even positions
– infinitely many odd numbers
– the first number n_1 is prime
– infinitely many prime numbers

A word $a^{n_1} b\, a^{n_2} b\, a^{n_3} b\ldots$ describes a number sequence $n_1\, n_2\, n_3\ldots$

Which properties of number sequences are regular?

– odd numbers on even positions
– infinitely many odd numbers
– the first number n_1 is prime
– infinitely many prime numbers
– sequence is ultimately constant

A word $a^{n_1} b\, a^{n_2} b\, a^{n_3} b\ldots$ describes a number sequence $n_1\, n_2\, n_3\ldots$

Which properties of number sequences are regular?

- odd numbers on even positions
- infinitely many odd numbers
- the first number n_1 is prime
- infinitely many prime numbers
- sequence is ultimately constant
- sequence is bounded

A word $a^{n_1} b\, a^{n_2} b\, a^{n_3} b\ldots$ describes a number sequence $n_1 n_2 n_3\ldots$

Which properties of number sequences are regular?

– odd numbers on even positions
– infinitely many odd numbers
– the first number n_1 is prime
– infinitely many prime numbers
– sequence is ultimately constant
– sequence is bounded
– sequence tends to ∞

A word $a^{n_1} b\, a^{n_2} b\, a^{n_3} b\ldots$ describes a number sequence $n_1\, n_2\, n_3\ldots$

Which properties of number sequences are regular?

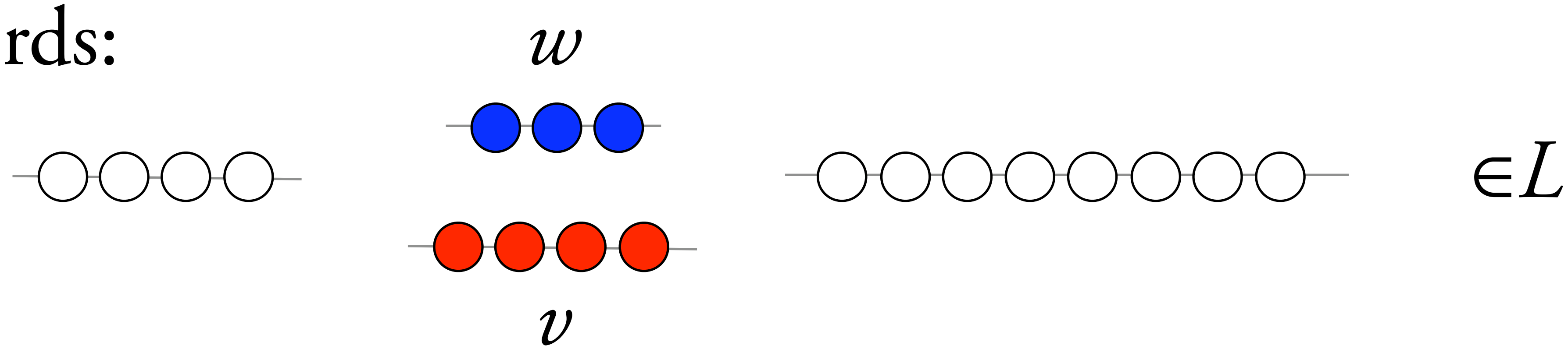
– odd numbers on even positions
– infinitely many odd numbers
– the first number n_1 is prime
– infinitely many prime numbers
– sequence is ultimately constant
– sequence is bounded
– sequence tends to ∞
– exists a bounded subsequence

Myhill-Nerode equivalence.
Two finite words v, w are equivalent for a language L if they can be swapped in any environment, and L will not notice.

Myhill-Nerode equivalence.

Two finite words v, w are equivalent for a language L if they can be swapped in any environment, and L will not notice.

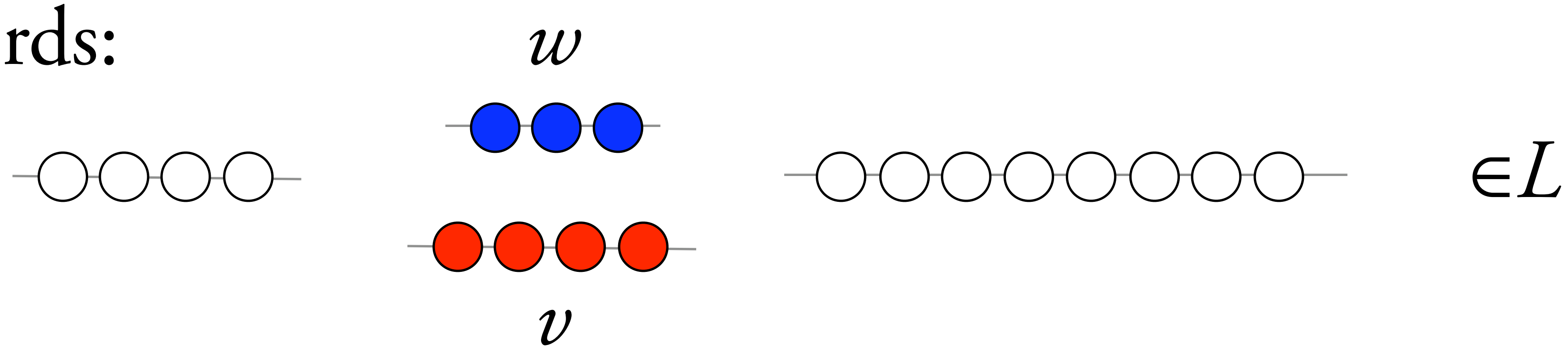
for finite words:



Myhill-Nerode equivalence.

Two finite words v, w are equivalent for a language L if they can be swapped in any environment, and L will not notice.

for finite words:

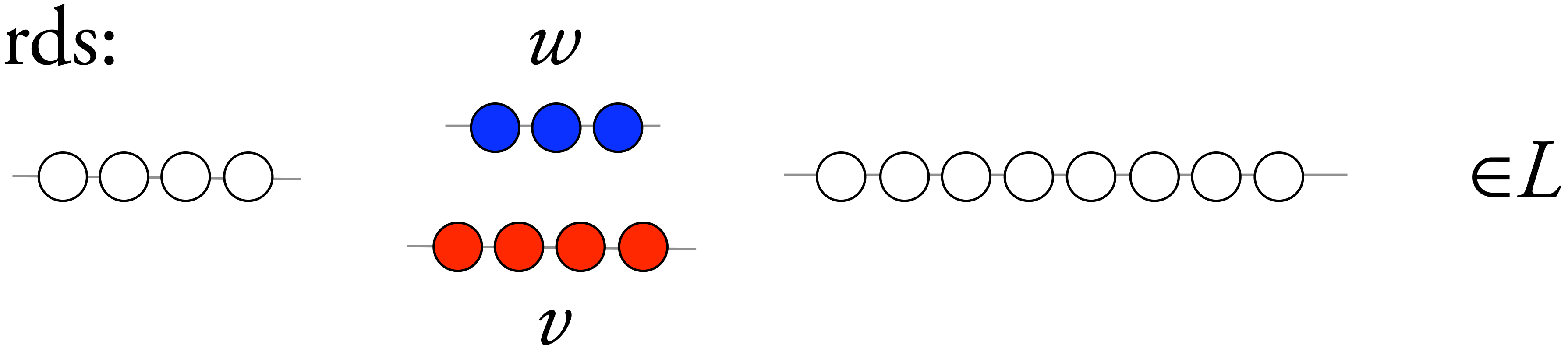


for infinite words:

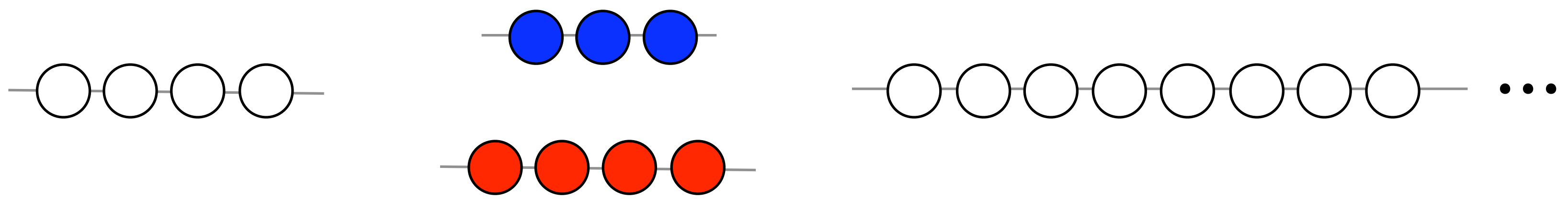
Myhill-Nerode equivalence.

Two finite words v, w are equivalent for a language L if they can be swapped in any environment, and L will not notice.

for finite words:



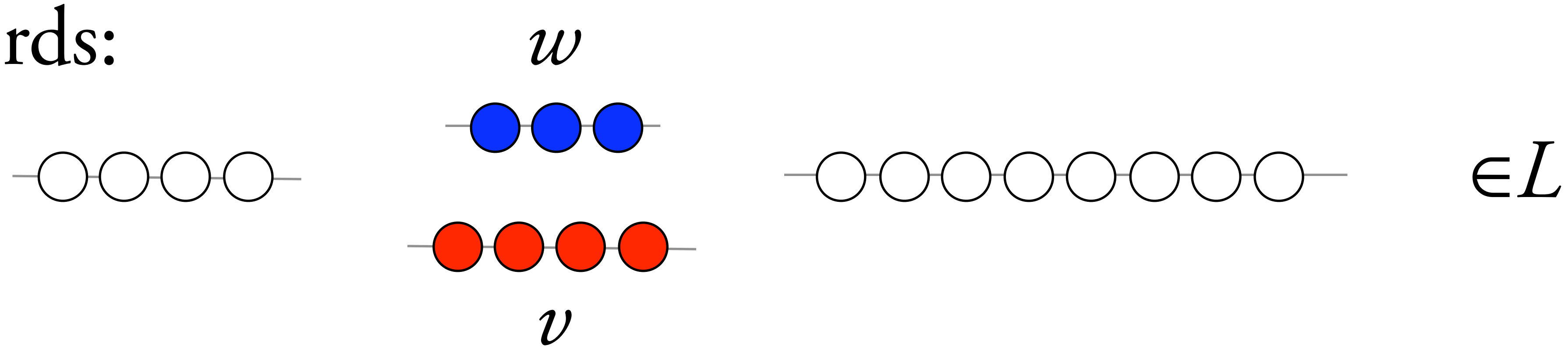
for infinite words:



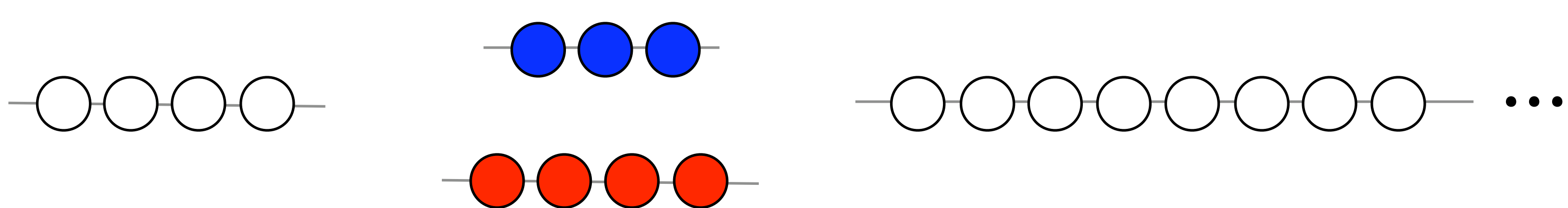
Myhill-Nerode equivalence.

Two finite words v, w are equivalent for a language L if they can be swapped in any environment, and L will not notice.

for finite words:



for infinite words:

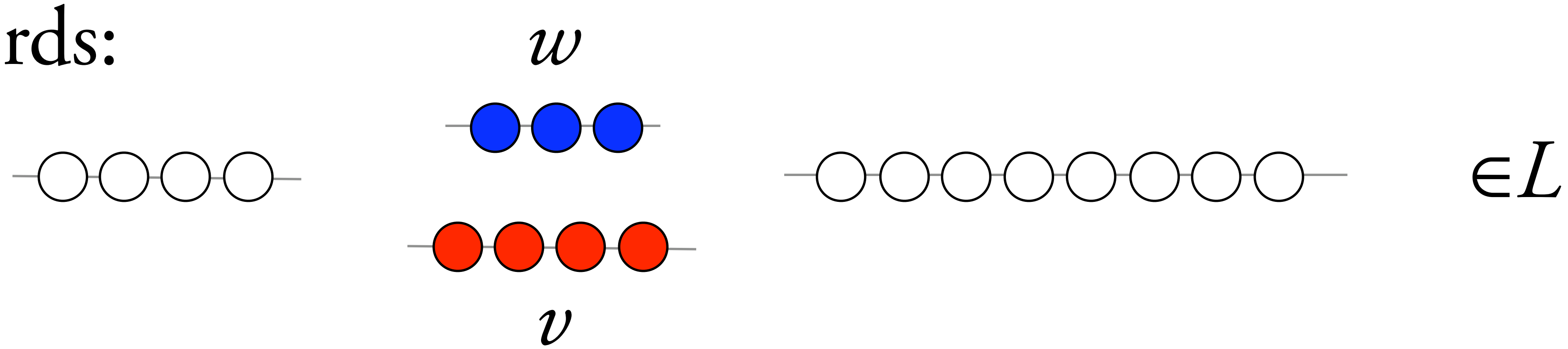


“infinitely many primes” has 1 equivalence class

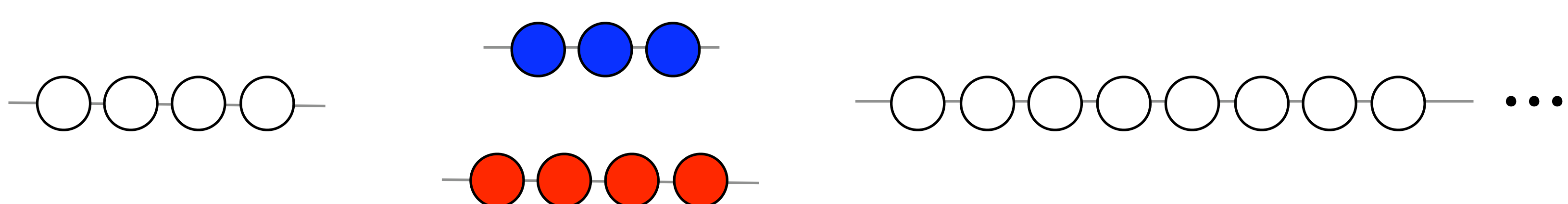
Myhill-Nerode equivalence.

Two finite words *v*, *w* are equivalent for a language *L* if they can be swapped in any environment, and *L* will not notice.

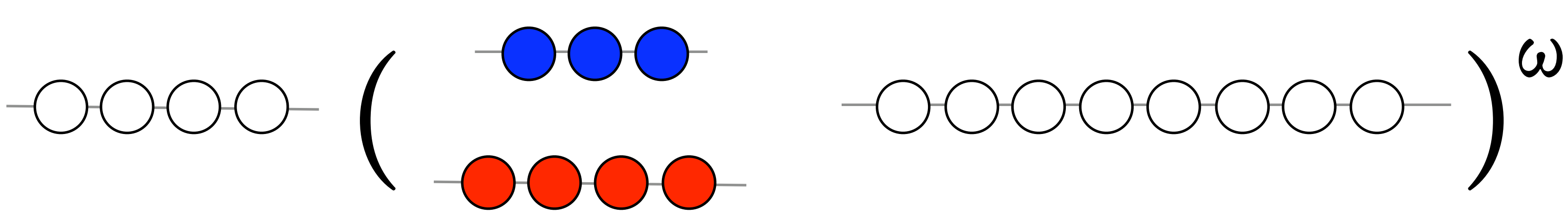
for finite words:



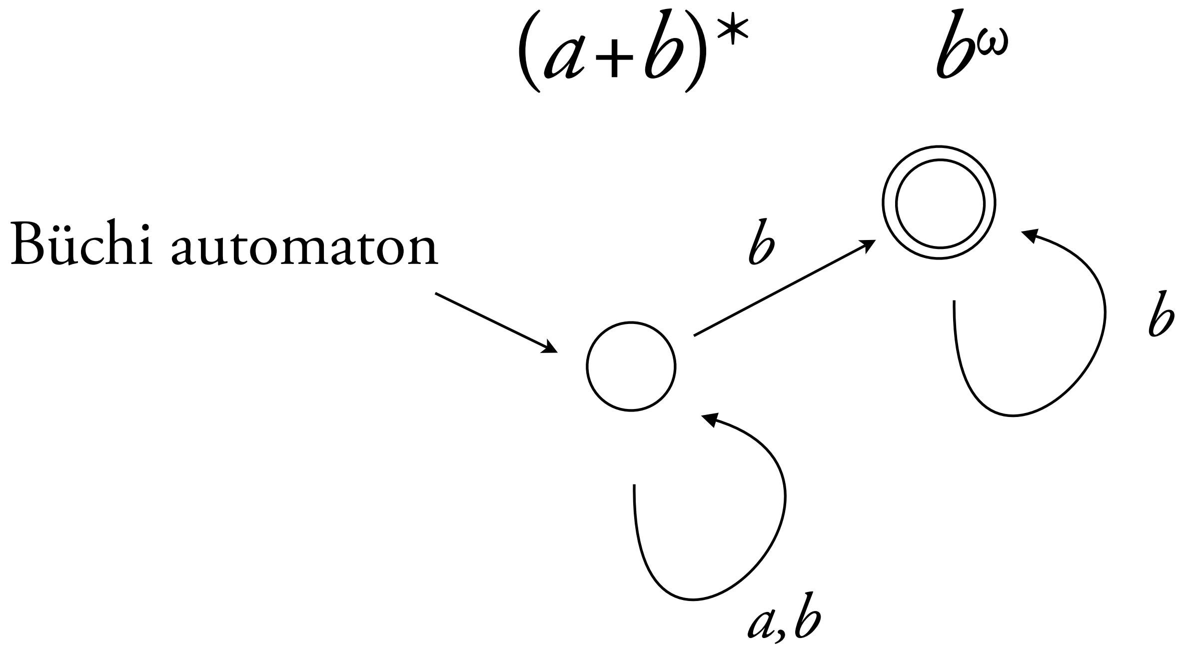
for infinite words:



"infinitely many primes" has 1 equivalence class



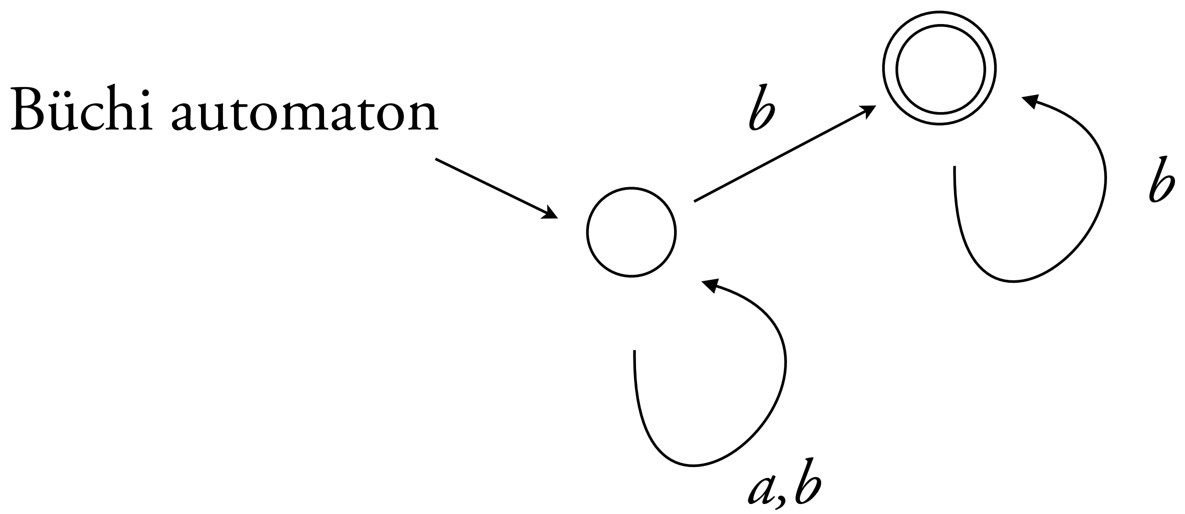
$(a+b)^*$ $\quad$ b^ω



acceptance condition:

(double circle) appears infinitely often,

$(a+b)^*$ $\quad$ b^ω

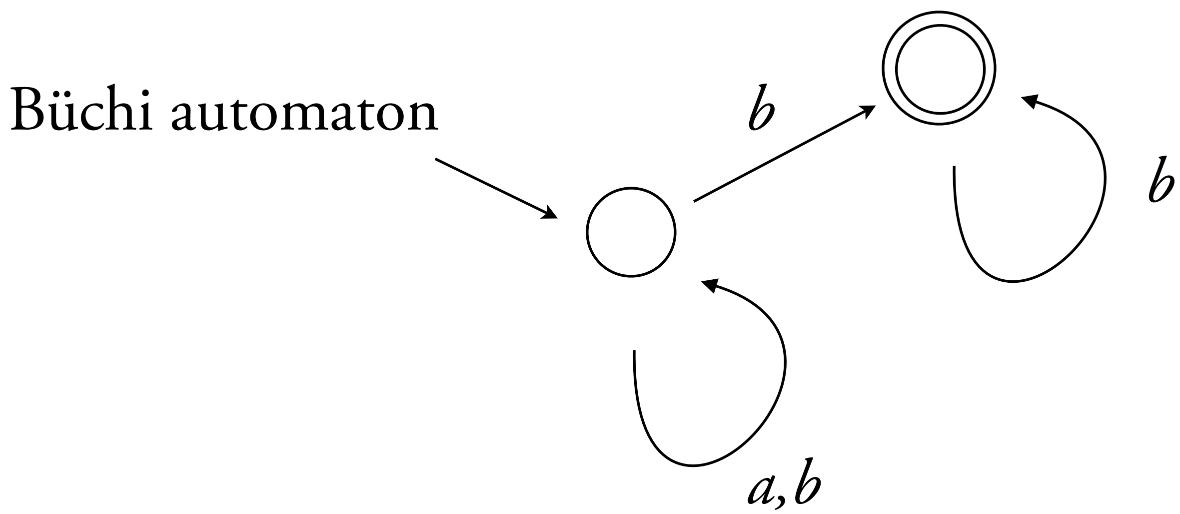


acceptance condition:

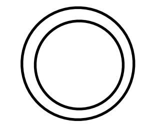
◎ appears infinitely often,

Thm. (McNaughton `66)
For every nondeterministic Büchi automaton, there is an equivalent deterministic Muller automaton.

$(a+b)^* \quad b^\omega$

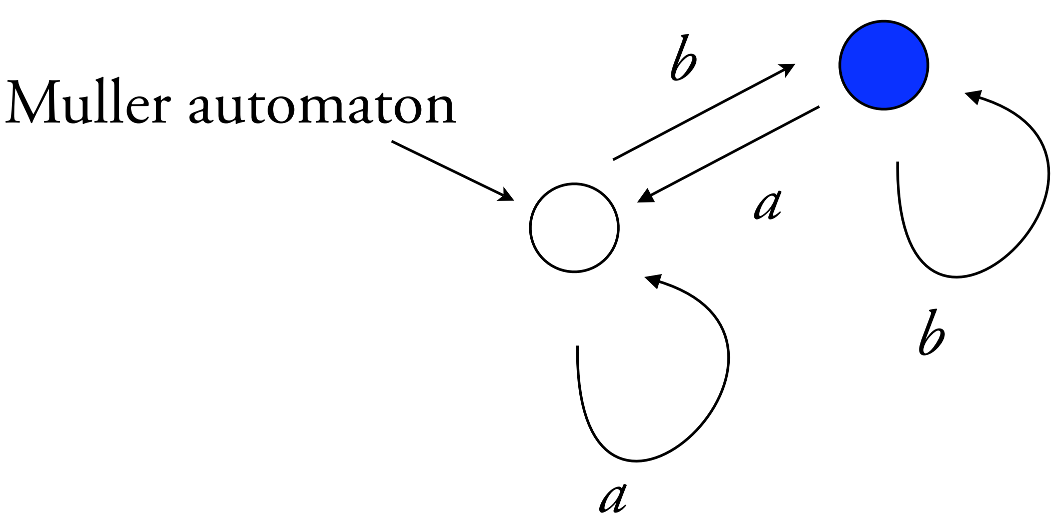


acceptance condition:

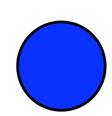
 appears infinitely often,

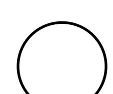
Thm. (McNaughton `66)

For every nondeterministic Büchi automaton, there is an equivalent deterministic Muller automaton.



acceptance condition:

 appears infinitely often,

and  appears finitely often

Monadic second-order logic (MSO)

Monadic second-order logic (MSO)

"infinitely many *a*'s on even positions"

Monadic second-order logic (MSO)

"infinitely many *a*'s on even positions"

$$\exists X \begin{cases} \forall x\ \exists y \leq x \quad y \in X \\ \forall x\ \forall y \quad \mathrm{suc}(x,y) \Rightarrow (x \in X \Leftrightarrow y \notin X) \\ \forall x\ \forall y \geq x \quad a(y) \wedge y \in X \end{cases}$$

Monadic second-order logic (MSO)

"infinitely many *a*'s on even positions"

There is a set X of positions

contains the first position

contains every second position

$$\exists X \left\{ \begin{array}{ll} \forall x\, \exists y{\leq}x & y{\in}X \\ \forall x\, \forall y & \mathrm{suc}(x,y) \Rightarrow (x{\in}X \Leftrightarrow y{\notin}X) \\ \forall x\, \forall y{\geq}x & a(y) \wedge y{\in}X \end{array} \right.$$

contains infinitely many *a*'s

Monadic second-order logic (MSO)

"infinitely many *a*'s on even positions"

There is a set X of positions

contains the first position

contains every second position

$$\exists X \begin{cases} \forall x\ \exists y \leq x \quad y \in X \\ \forall x\ \forall y \quad \mathrm{suc}(x,y) \Rightarrow (x \in X \Leftrightarrow y \notin X) \\ \forall x\ \forall y \geq x \quad a(y) \wedge y \in X \end{cases}$$

contains infinitely many *a*'s

Thm. (Büchi `60)

Büchi automata and MSO have the same expressive power.

Monadic second-order logic (MSO)

"infinitely many *a*'s on even positions"

There is a set X of positions — contains the first position — contains every second position — contains infinitely many *a*'s

$$\exists X \begin{cases} \forall x\ \exists y \leq x \quad y \in X \\ \forall x\ \forall y \quad \mathrm{suc}(x,y) \Rightarrow (x \in X \Leftrightarrow y \notin X) \\ \forall x\ \forall y \geq x \quad a(y) \wedge y \in X \end{cases}$$

Thm. (Büchi `60)

Büchi automata and MSO have the same expressive power.

Corollary of determinization.

For infinite words, MSO = Weak MSO

(Weak MSO: set quantification only over finite sets.)

Monadic second-order logic (MSO)

"infinitely many *a*'s on even positions"

There is a set X of positions

contains the first position

contains every second position

contains infinitely many *a*'s

$$\exists X \begin{cases} \forall x\ \exists y \leq x \quad y \in X \\ \forall x\ \forall y \quad \mathrm{suc}(x,y) \Rightarrow (x \in X \Leftrightarrow y \notin X) \\ \forall x\ \forall y \geq x \quad a(y) \wedge y \in X \end{cases}$$

$\forall x\ \exists y > x\ \exists X \ldots$

Thm. (Büchi `60)

Büchi automata and MSO have the same expressive power.

Corollary of determinization.

For infinite words, MSO = Weak MSO

(Weak MSO: set quantification only over finite sets.)

logic		automata
MSO = WMSO	⟷	Büchi, Muller

logic		automata
MSO = WMSO	⟷	Büchi, Muller
WMSO+U	⟷	deterministic max-automata

Max-automaton

Max-automaton

Has finite state space Q and a finite set of counters C.

The counters are only read by the acceptance condition, and not during the run.

Max-automaton

Has finite state space Q and a finite set of counters C.

The counters are only read by the acceptance condition, and not during the run.

Transitions can do counter operations:

c:=c+1

c:=0

c:=max(c,d)

Max-automaton

Has finite state space *Q* and a finite set of counters *C*.

The counters are only read by the acceptance condition, and not during the run.

Transitions can do counter operations:

c:=c+1

c:=0

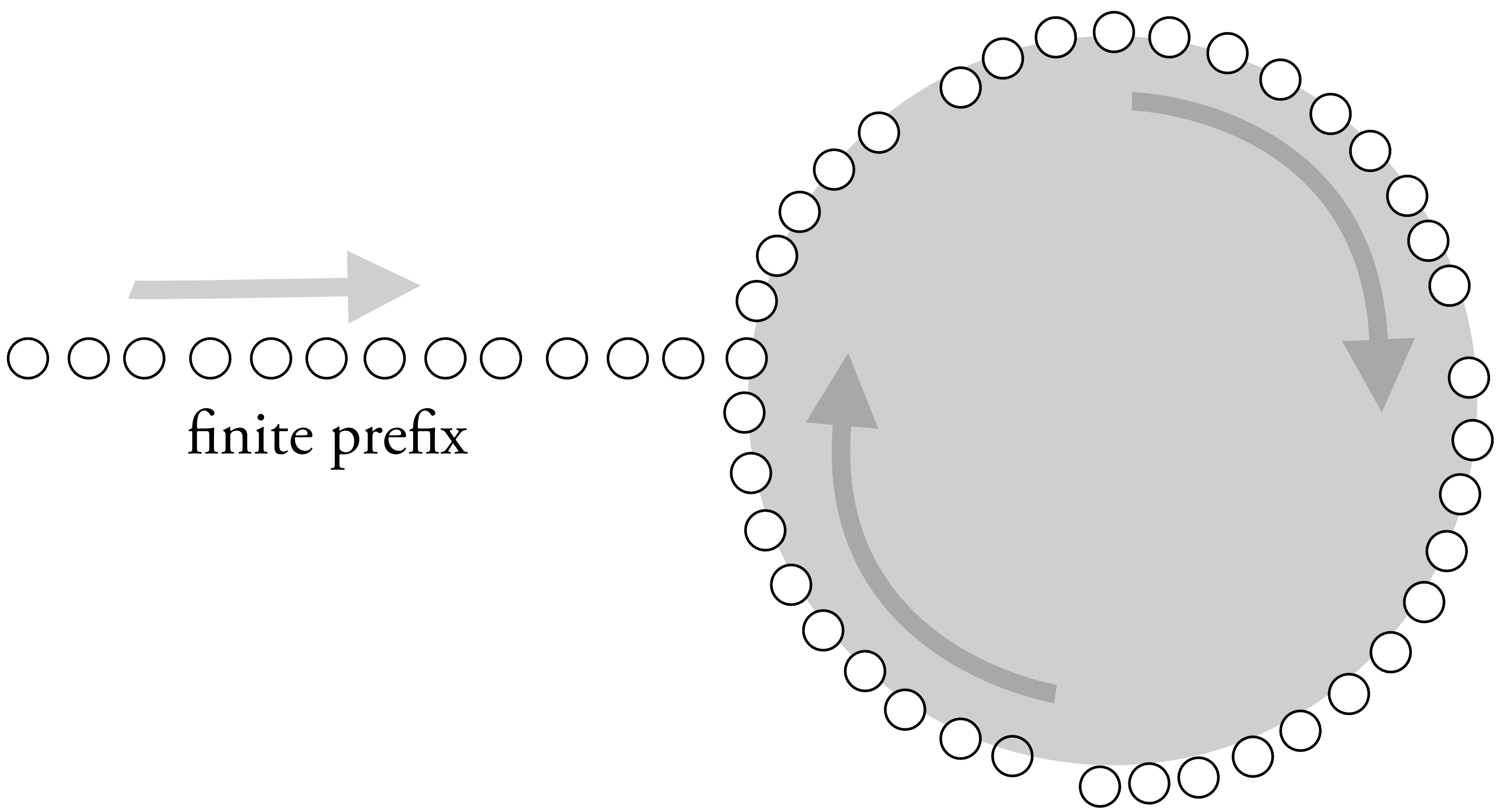
c:=max(c,d)

Acceptance condition: boolean combination of clauses

"counter *c* is bounded"

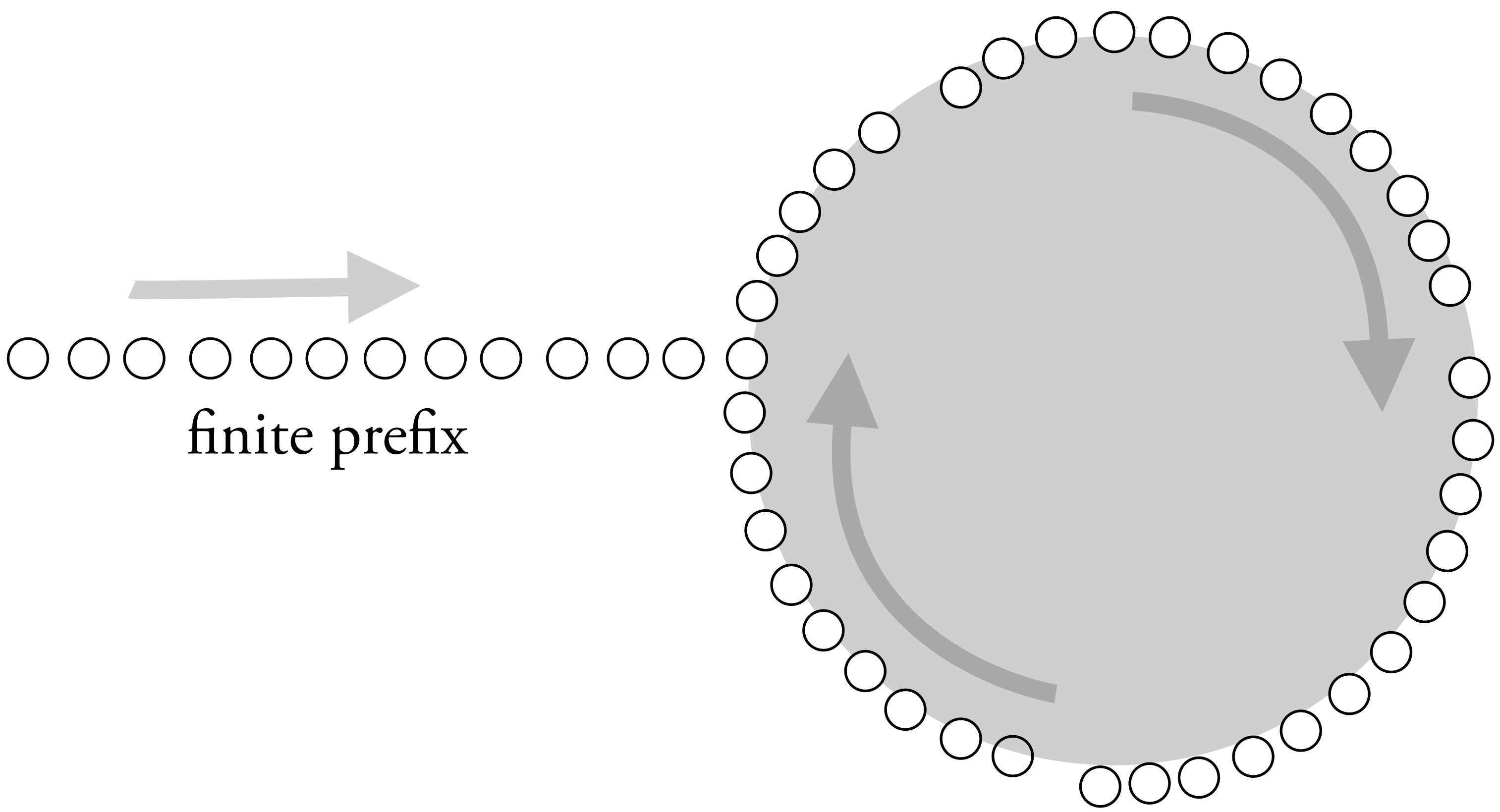
Thm. Emptiness decidable for max-automata.

Thm. Emptiness decidable for max-automata.



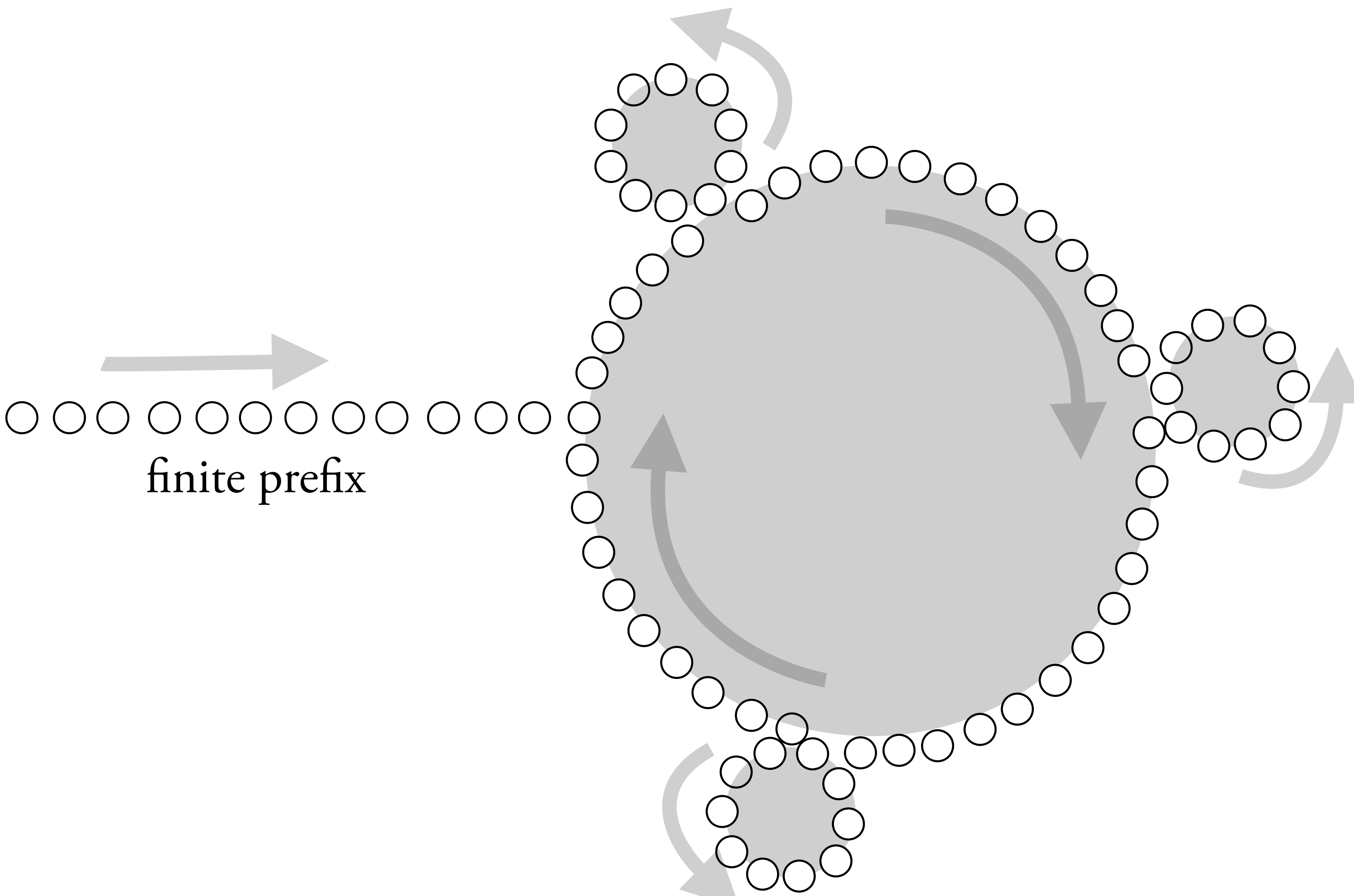
Thm. Emptiness decidable for max-automata.

For a max-automaton, the accepting condition says some counters are bounded, and some are not.



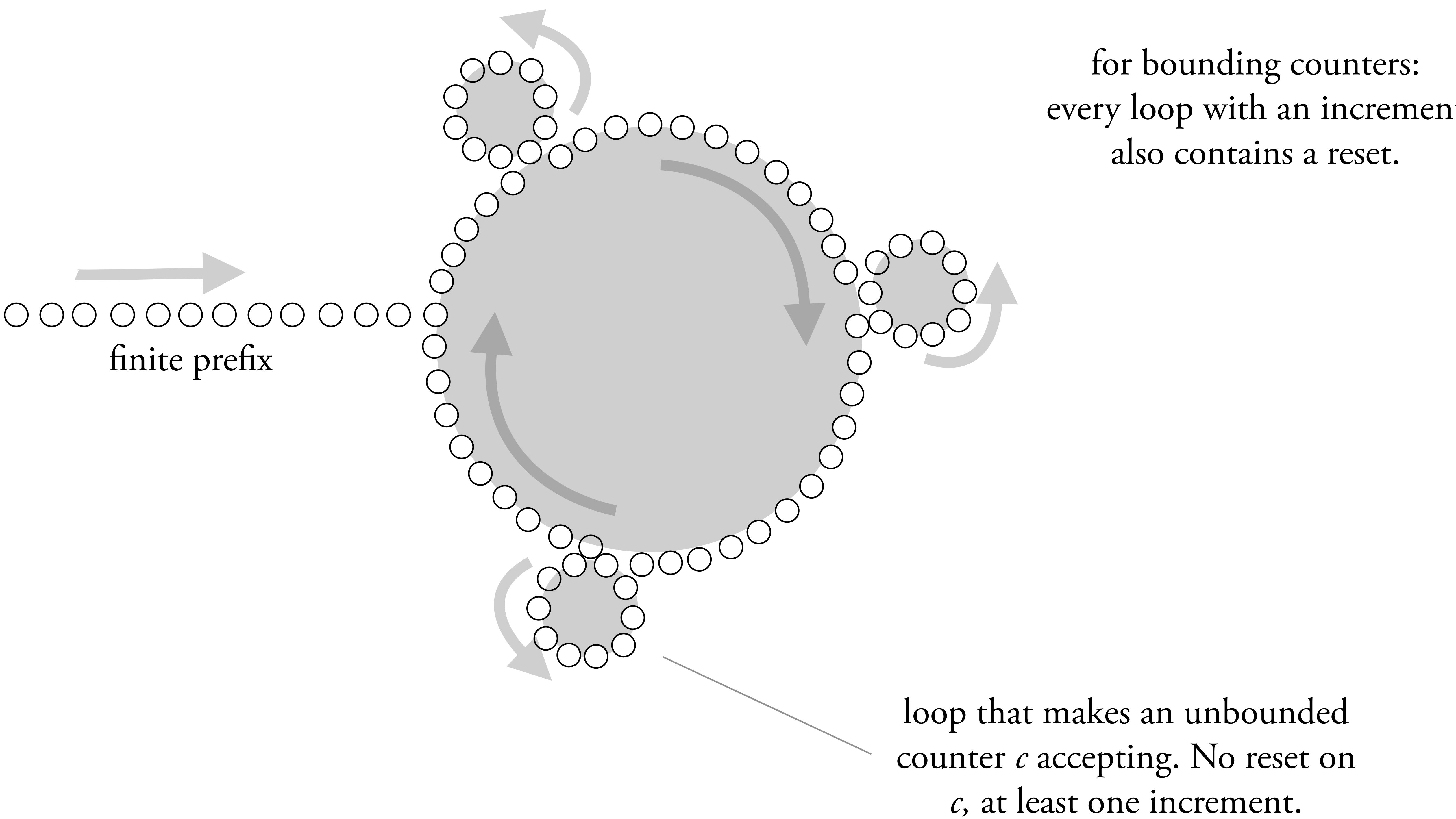
Thm. Emptiness decidable for max-automata.

For a max-automaton, the accepting condition says some counters are bounded, and some are not.



Thm. Emptiness decidable for max-automata.

For a max-automaton, the accepting condition says some counters are bounded, and some are not.



What is the logic for max-automata?

Extend weak MSO with the following quantifier:

Extend weak MSO with the following quantifier:

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

which is the same as

" $\varphi(X)$ holds for finite sets X of arbitrarily large size"

which is the same as

$$\bigwedge_n \varphi(X) \wedge \; n < |X| < \infty$$

Extend weak MSO with the following quantifier:

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

which is the same as

" $\varphi(X)$ holds for finite sets X of arbitrarily large size"

which is the same as

$$\bigwedge_n \varphi(X) \wedge n < |X| < \infty$$

Example: $\{a^{n_1}\, b\; a^{n_2}\, b\; a^{n_3}\, b \ldots : n_1\, n_2\, n_3 \ldots$ is not bounded$\}$

Extend weak MSO with the following quantifier:

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

which is the same as

" $\varphi(X)$ holds for finite sets X of arbitrarily large size"

which is the same as

$$\bigwedge_n \varphi(X) \wedge \; n<|X|<\infty$$

Example: $\{a^{n_1}\, b\; a^{n_2}\, b\; a^{n_3}\, b\ldots : n_1\, n_2\, n_3\ldots$ is not bounded$\}$

$\mathrm{U}X$ "X is a set of consecutive a's"

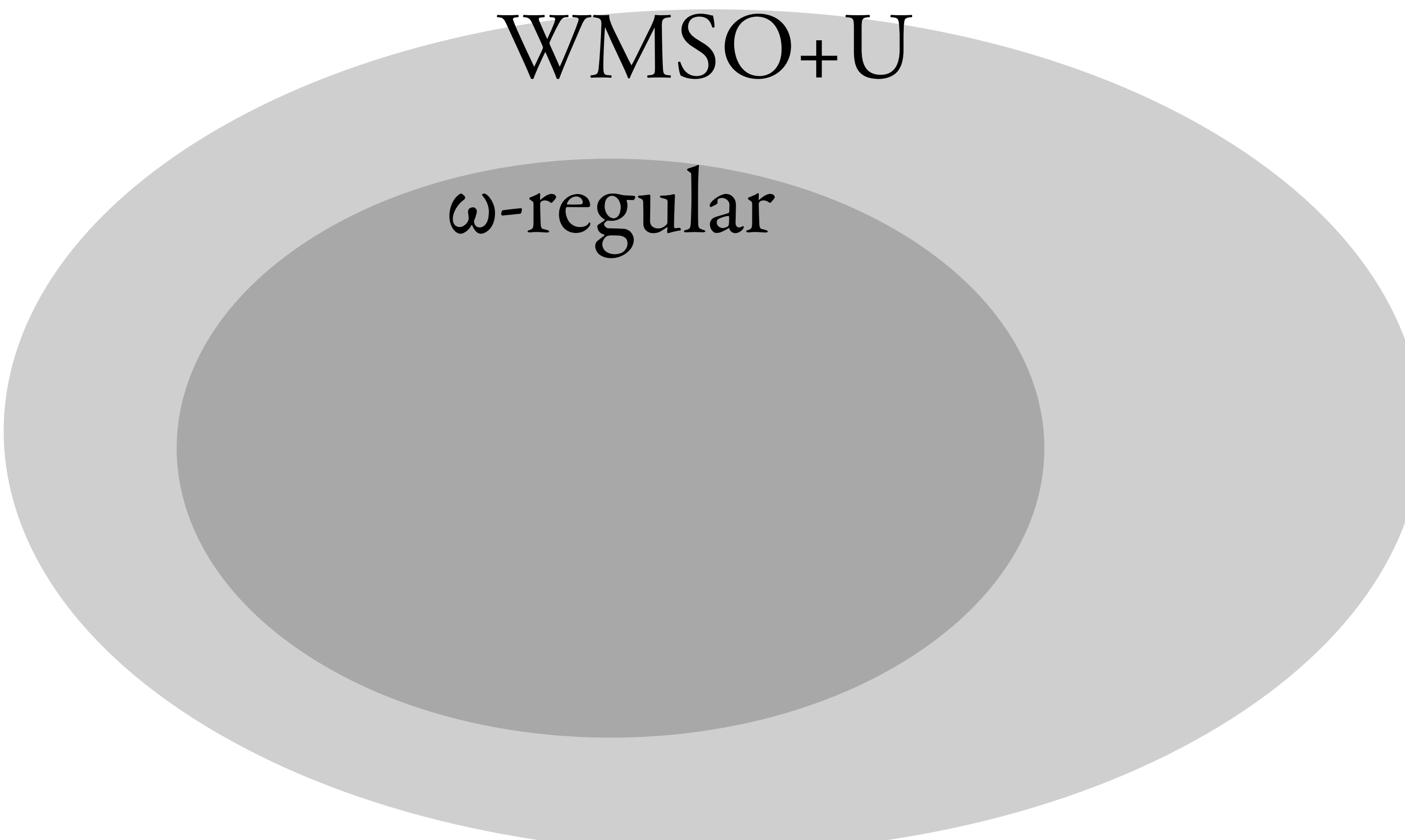
Thm. Deterministic max-automata recognize the same langauges as weak MSO with the unbounding quantifier.

Thm. Deterministic max-automata recognize the same langauges as weak MSO with the unbounding quantifier.

Proof. Effective translations both ways.

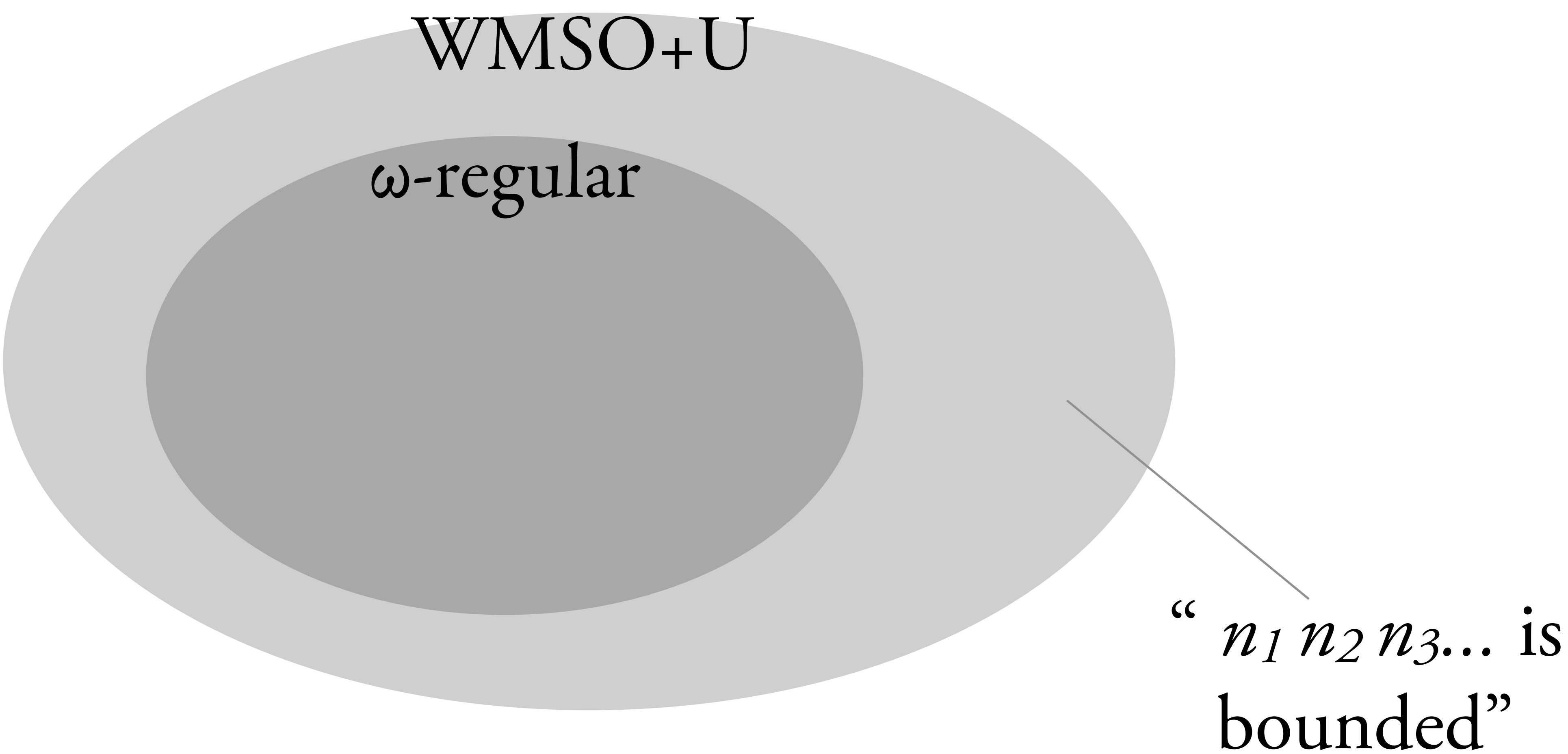
Thm. Deterministic max-automata recognize the same langauges as weak MSO with the unbounding quantifier.

Proof. Effective translations both ways.



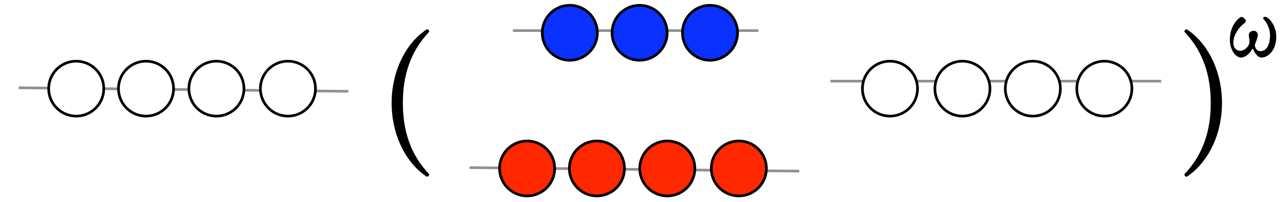
Thm. Deterministic max-automata recognize the same langauges as weak MSO with the unbounding quantifier.

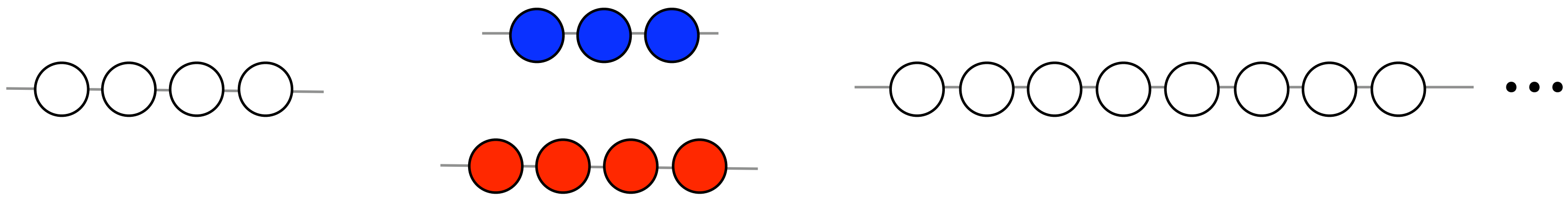
Proof. Effective translations both ways.



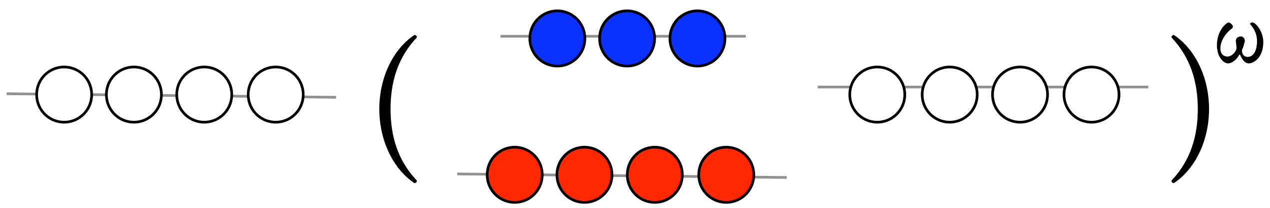
–logic
–automata
–decidability
–?

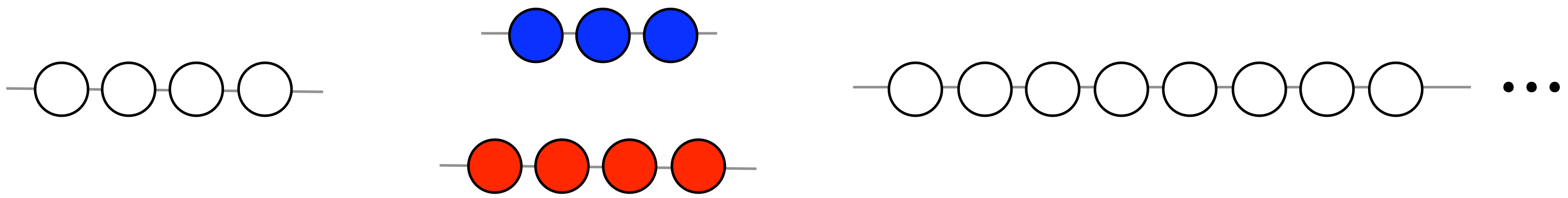
Myhill-Nerode equivalence.





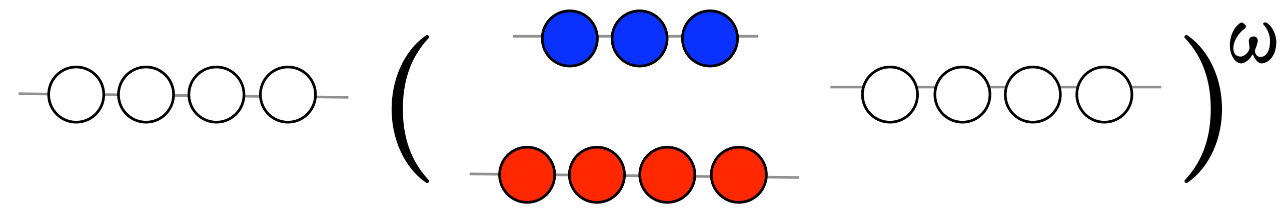
Myhill-Nerode equivalence.

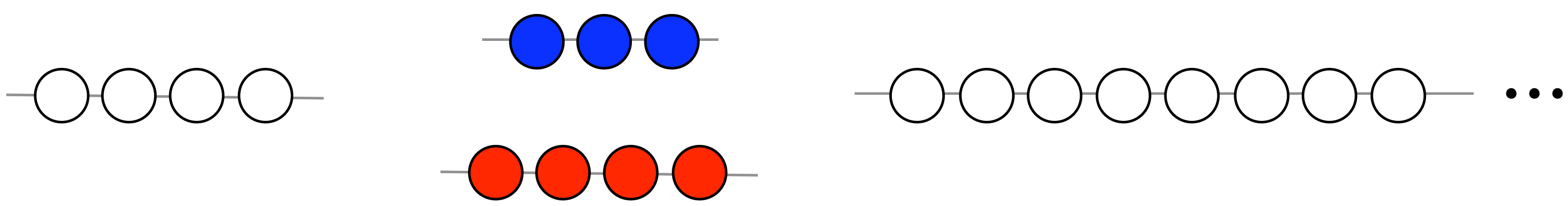




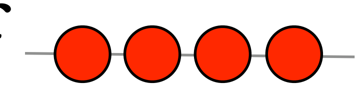
Prop. Languages recognized by max-automata have finitely many equivalence classes. Each class is a regular language of finite words.

Myhill-Nerode equivalence.

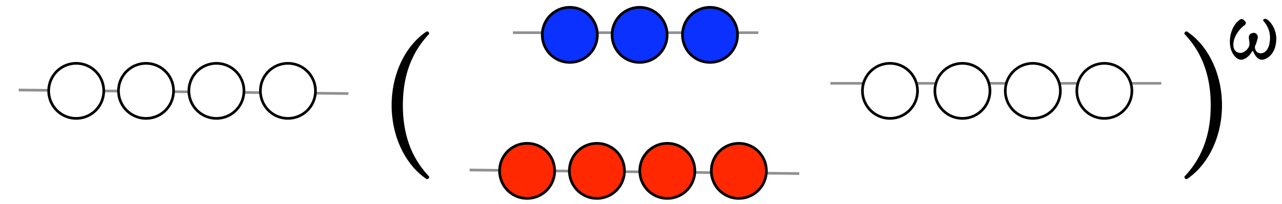


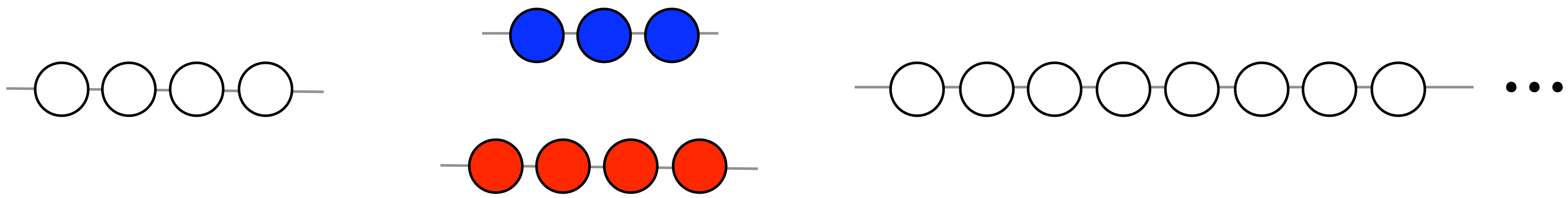


Prop. Languages recognized by max-automata have finitely many equivalence classes. Each class is a regular language of finite words.

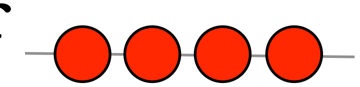
Proof sketch. Equivalence class of  depends on state transformations, which counters are incremented (but not how much), and which counters are reset.

Myhill-Nerode equivalence.

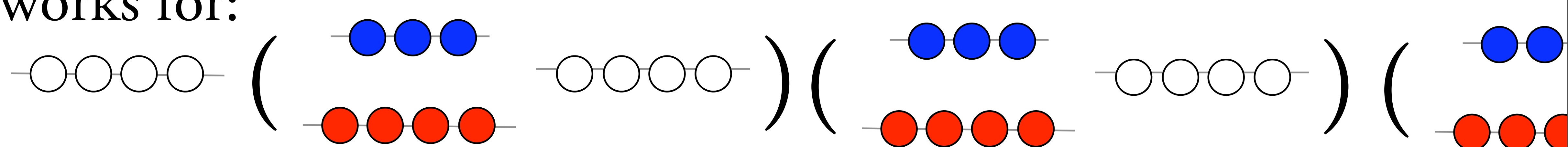




Prop. Languages recognized by max-automata have finitely many equivalence classes. Each class is a regular language of finite words.

Proof sketch. Equivalence class of  depends on state transformations, which counters are incremented (but not how much), and which counters are reset.

also works for:



What about full MSO with the unbounding quantifier?

What about full MSO with the unbounding quantifier?

Thm. MSO+U is strictly more expressive than WMSO+U

What about full MSO with the unbounding quantifier?

Thm. MSO+U is strictly more expressive than WMSO+U

separating language $L=\{a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots : n_1\, n_2\, n_3 \ldots$ tends to $\infty\}$

What about full MSO with the unbounding quantifier?

Thm. MSO+U is strictly more expressive than WMSO+U

separating language $L=\{a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots : n_1\, n_2\, n_3 \ldots$ tends to $\infty\}$

$L \in$ MSO+U

What about full MSO with the unbounding quantifier?

Thm. MSO+U is strictly more expressive than WMSO+U

separating language $L=\{a^{n_1} b\, a^{n_2} b\, a^{n_3} b\ldots : n_1\, n_2\, n_3\ldots \text{ tends to } \infty\}$

$L \in$ MSO+U

complement of L : exists a bounded subsequence.

What about full MSO with the unbounding quantifier?

Thm. MSO+U is strictly more expressive than WMSO+U

separating language $L = \{a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots : n_1\, n_2\, n_3 \ldots$ tends to $\infty\}$

$L \in$ MSO+U

complement of L : exists a bounded subsequence.

$L \notin$ WMSO+U

What about full MSO with the unbounding quantifier?

Thm. MSO+U is strictly more expressive than WMSO+U

separating language $L=\{a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots : n_1\, n_2\, n_3 \ldots \text{ tends to } \infty\}$

$L \in$ MSO+U

complement of L : exists a bounded subsequence.

$L \notin$ WMSO+U   topological argument.

What about full MSO with the unbounding quantifier?

Thm. MSO+U is strictly more expressive than WMSO+U

separating language $L = \{a^{n_1} b\, a^{n_2} b\, a^{n_3} b \ldots : n_1\, n_2\, n_3 \ldots$ tends to $\infty\}$

$L \in$ MSO+U

complement of L : exists a bounded subsequence.

$L \notin$ WMSO+U   topological argument.

acceptance condition "$n_1\, n_2\, n_3 \ldots$ is bounded"

What about full MSO with the unbounding quantifier?

Thm. MSO+U is strictly more expressive than WMSO+U

separating language $L=\{a^{n_1} b\, a^{n_2} b\, a^{n_3} b\ldots : n_1\, n_2\, n_3\ldots$ tends to $\infty\}$

$L \in$ MSO+U

complement of L : exists a bounded subsequence.

$L \notin$ WMSO+U   topological argument.

acceptance condition "$n_1\, n_2\, n_3\ldots$ is bounded"

is a countable union of closed sets (Σ_2)

What about full MSO with the unbounding quantifier?

Thm. MSO+U is strictly more expressive than WMSO+U

separating language $L=\{a^{n_1} b\, a^{n_2} b\, a^{n_3} b\ldots : n_1 n_2 n_3\ldots$ tends to $\infty\}$

$L \in$ MSO+U

complement of L : exists a bounded subsequence.

$L \notin$ WMSO+U   topological argument.

acceptance condition "$n_1 n_2 n_3\ldots$ is bounded"

is a countable union of closed sets (Σ_2)

"sequence bounded by N" is a closed set

What about full MSO with the unbounding quantifier?

Thm. MSO+U is strictly more expressive than WMSO+U

separating language $L=\{a^{n_1} b\, a^{n_2} b\, a^{n_3} b\ldots : n_1\, n_2\, n_3\ldots$ tends to $\infty\}$

$L \in$ MSO+U

complement of L : exists a bounded subsequence.

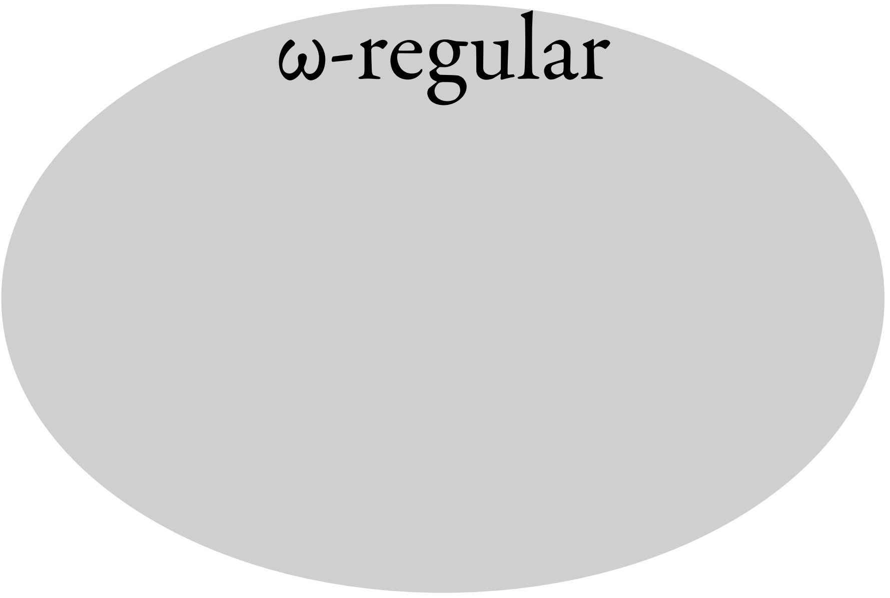
$L \notin$ WMSO+U   topological argument.

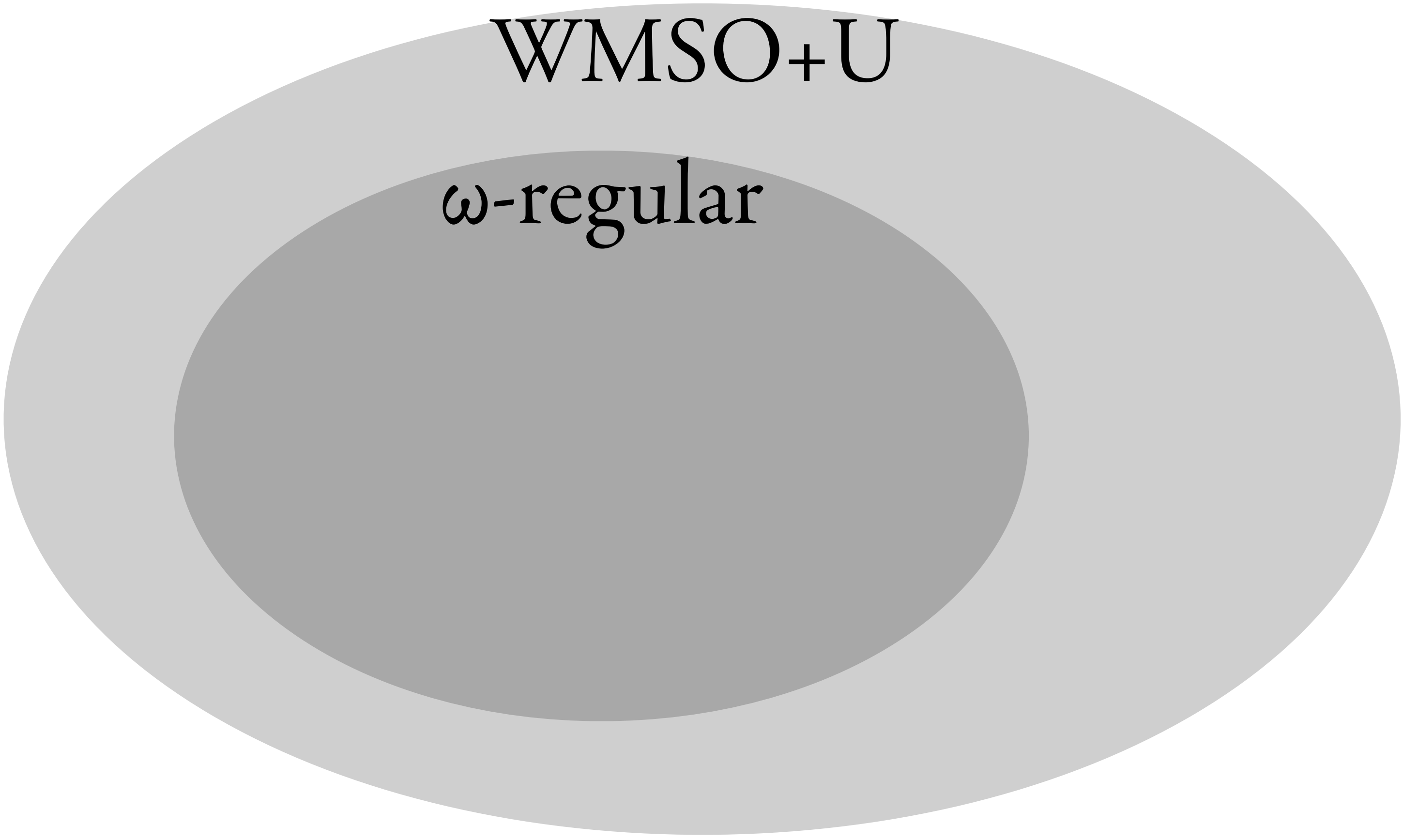
acceptance condition "$n_1\, n_2\, n_3\ldots$ is bounded"

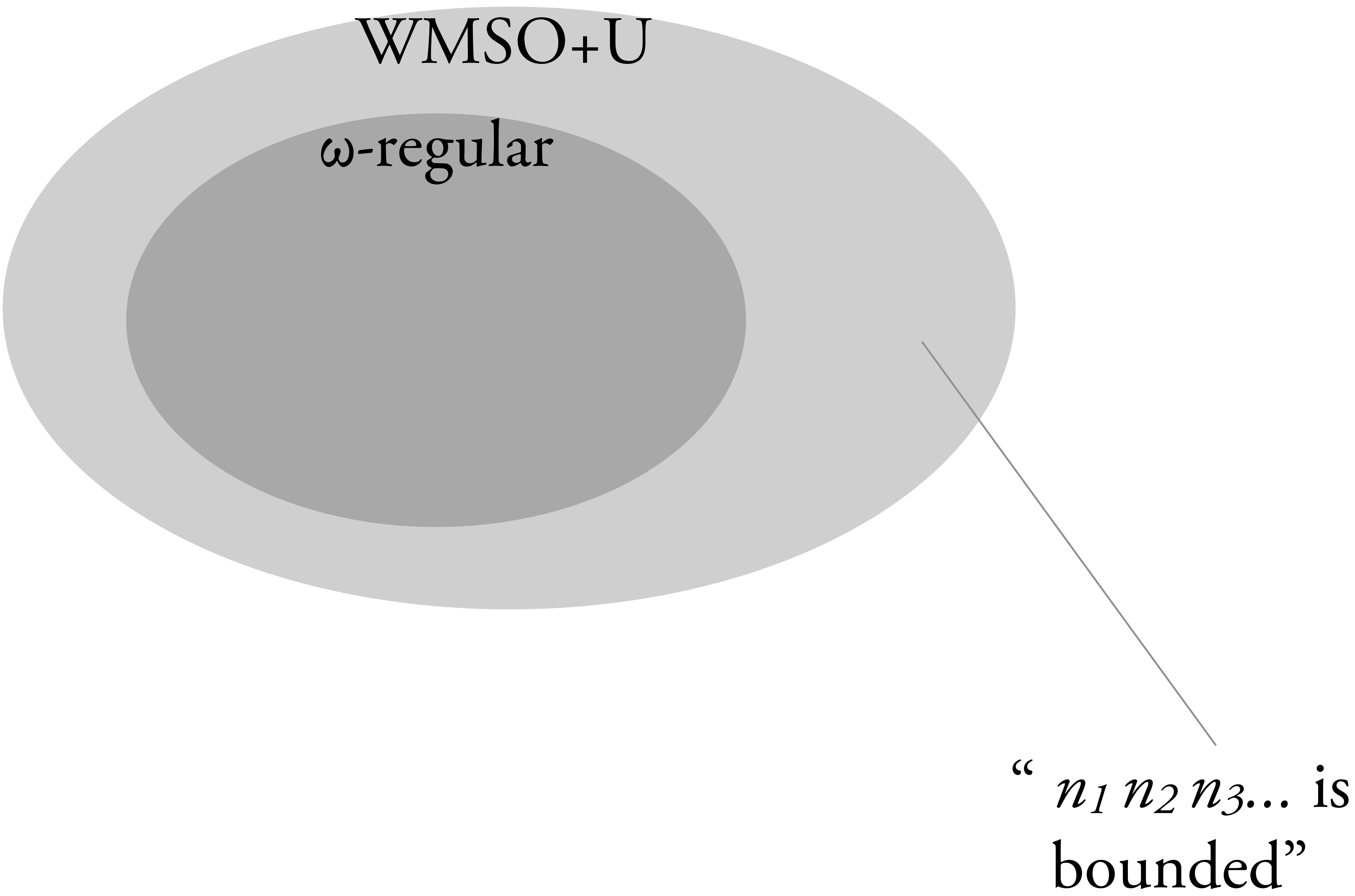
is a countable union of closed sets (Σ_2)

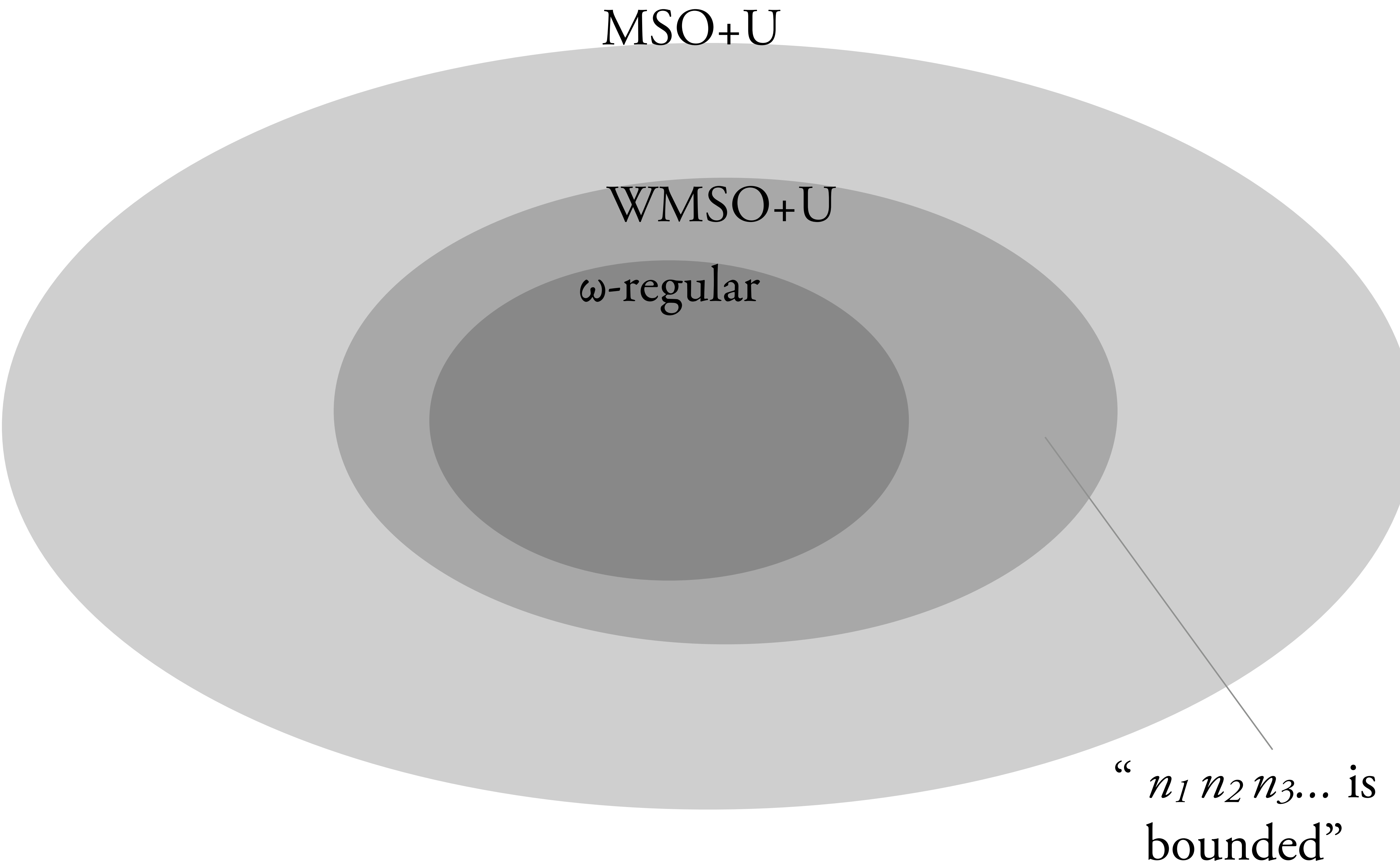
"sequence bounded by N" is a closed set

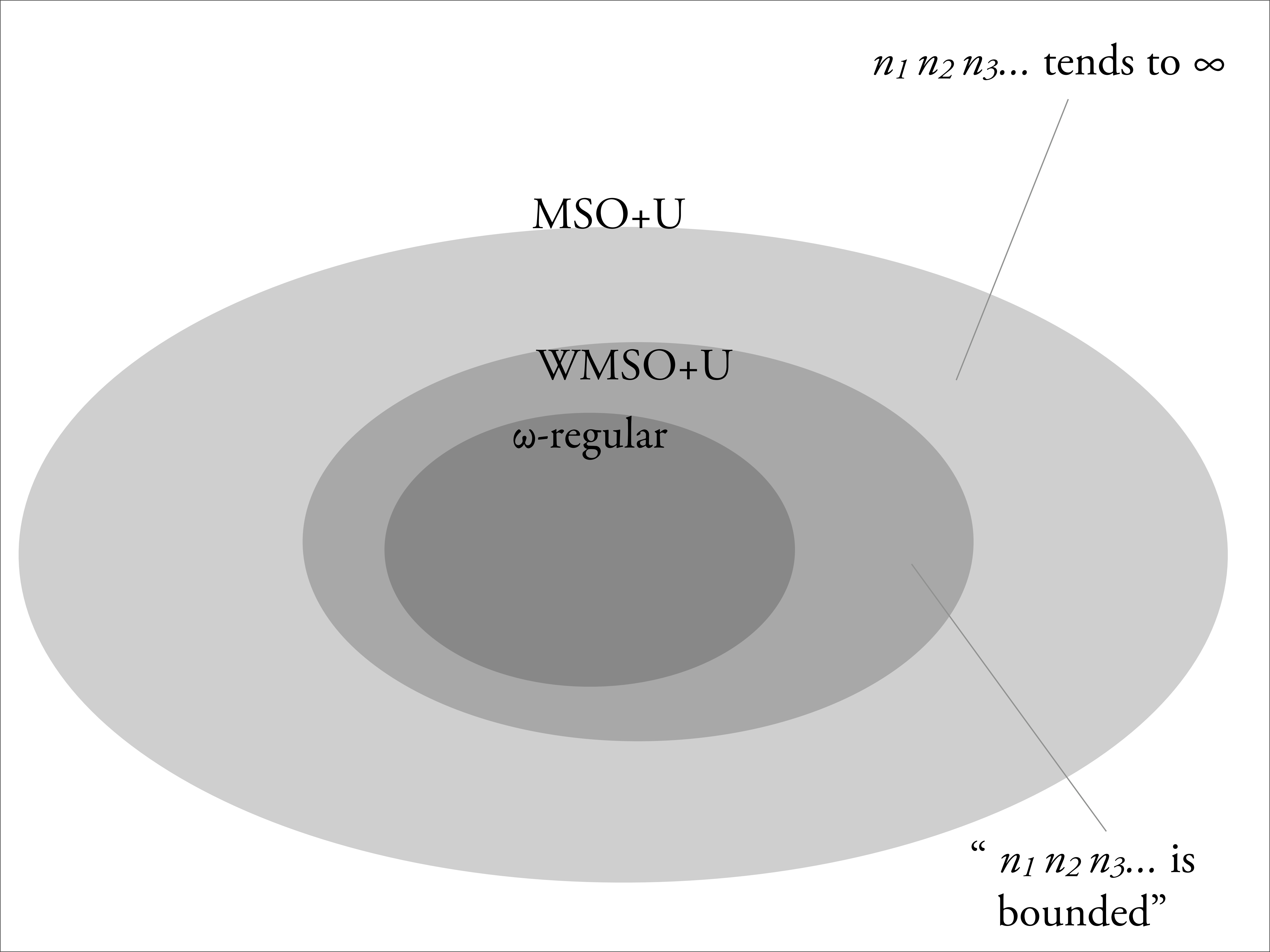
Prop. A language recognized by a max automaton is a boolean combination of Σ_2 sets, while L is not.


ω-regular

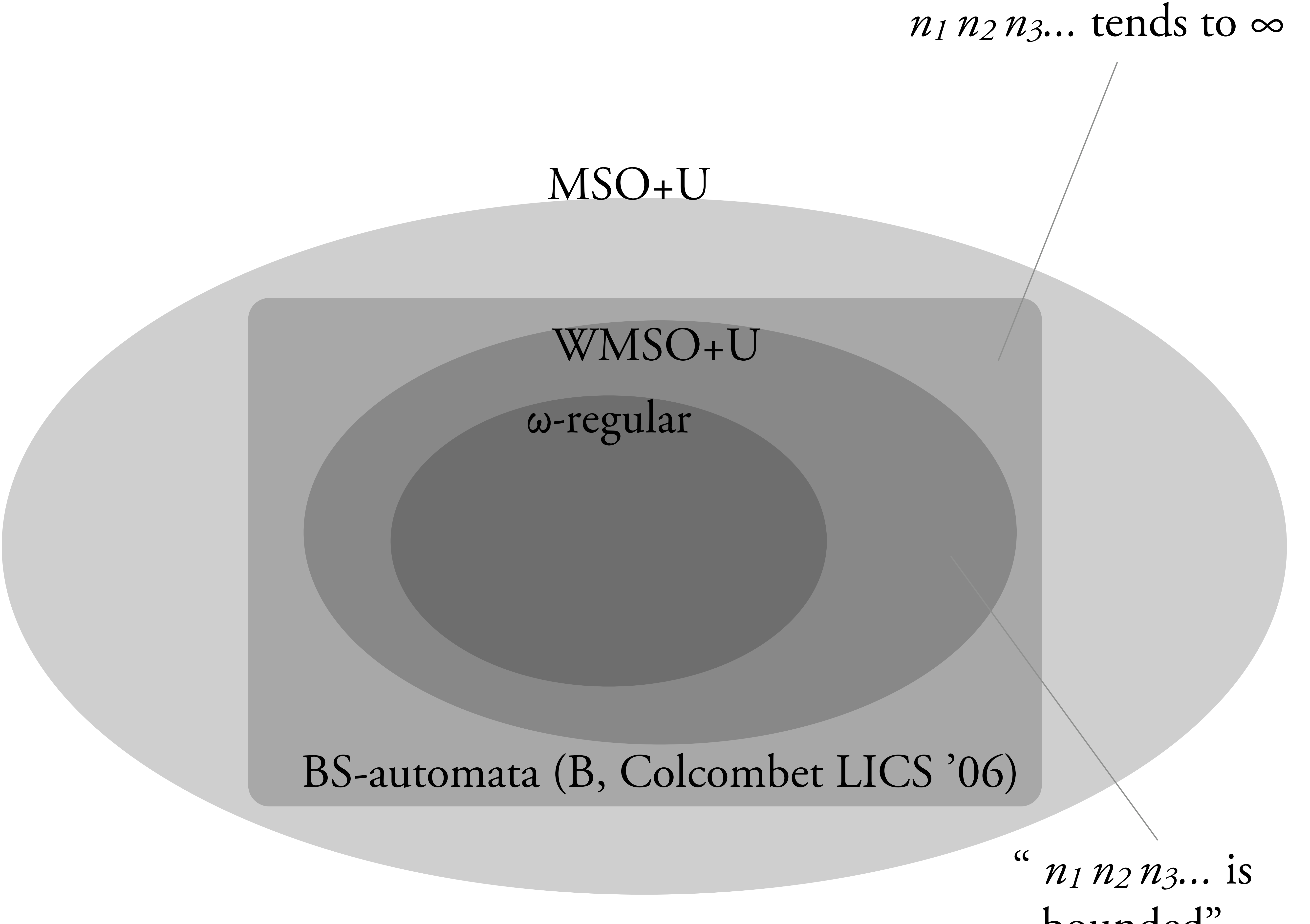

WMSO+U
ω-regular


WMSO+U
ω-regular
“ $n_1 n_2 n_3$… is bounded”

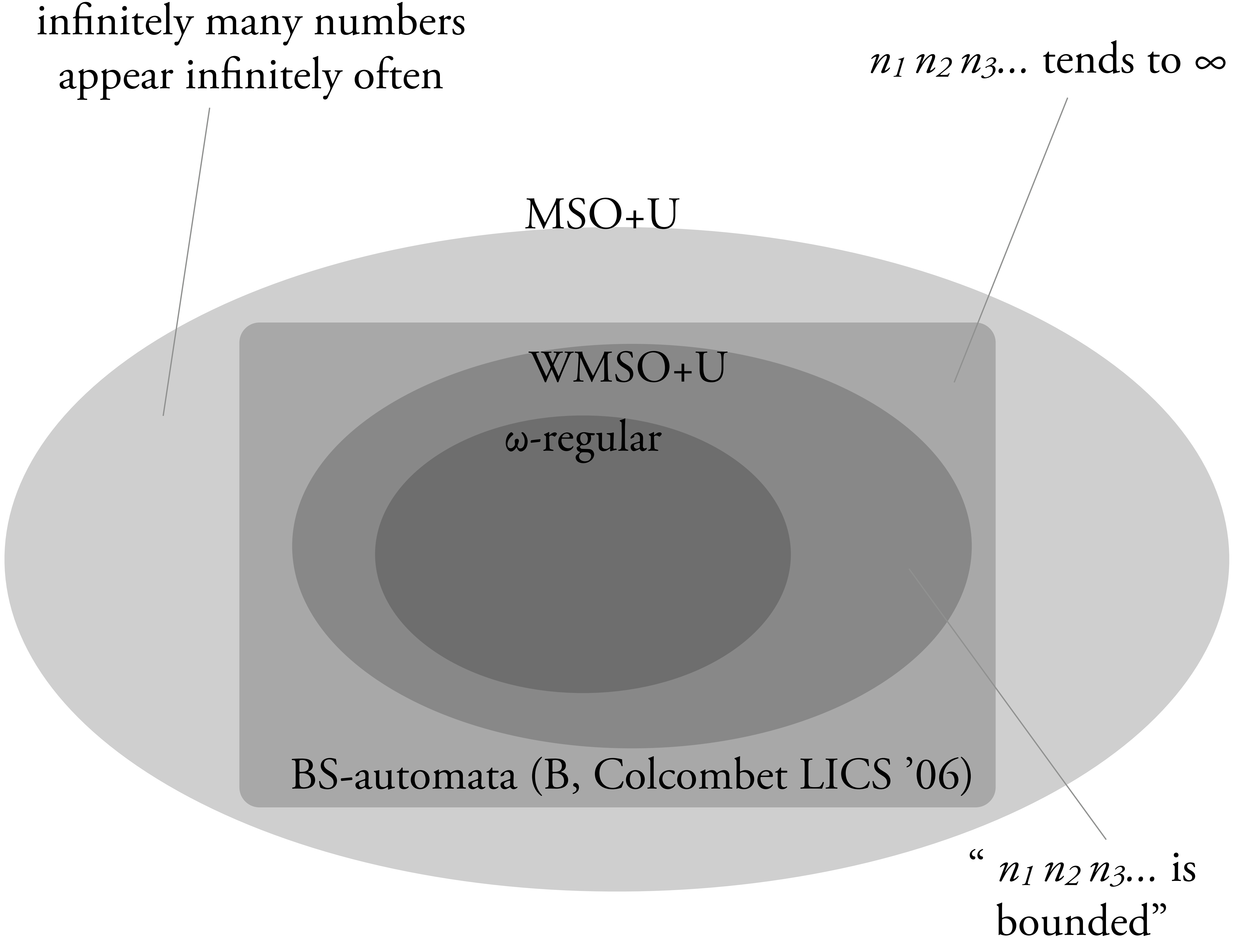

MSO+U
WMSO+U
ω-regular
“ n1 n2 n3… is bounded”



$n_1 n_2 n_3$… tends to ∞
MSO+U
WMSO+U
ω-regular
“ $n_1 n_2 n_3$… is bounded”



$n_1\, n_2\, n_3 \ldots$ tends to ∞
MSO+U
WMSO+U
ω-regular
BS-automata (B, Colcombet LICS '06)
" $n_1\, n_2\, n_3 \ldots$ is bounded"


infinitely many numbers appear infinitely often
$n_1\, n_2\, n_3 \ldots$ tends to ∞
MSO+U
WMSO+U
ω-regular
BS-automata (B, Colcombet LICS ’06)
“ $n_1\, n_2\, n_3 \ldots$ is bounded”

Conclusion

New robust class of languages extending ω-regular languages.
(automata, logic, decidability)

Conclusion

New robust class of languages extending ω-regular languages.
(automata, logic, decidability)

Future work

Conclusion

New robust class of languages extending ω-regular languages.
(automata, logic, decidability)

Future work

– Full MSO+U

Conclusion

New robust class of languages extending ω-regular languages.
(automata, logic, decidability)

Future work

– Full MSO+U
– Tree extensions

Conclusion

New robust class of languages extending ω-regular languages.
(automata, logic, decidability)

Future work

– Full MSO+U
– Tree extensions
– Algebra

Conclusion

New robust class of languages extending ω-regular languages.
(automata, logic, decidability)

Future work

– Full MSO+U
– Tree extensions
– Algebra
– Regular expressions

a bit about the proofs

WMSO+U → deterministic max-automata

WMSO+U ⟶ deterministic max-automata

Proof strategy: Automata are closed under all operations in the logic.

WMSO+U → deterministic max-automata

Proof strategy: Automata are closed under all operations in the logic.

Boolean operations: free for a deterministic automaton.

WMSO+U → deterministic max-automata

Proof strategy: Automata are closed under all operations in the logic.

Boolean operations: free for a deterministic automaton.

Weak existential quantification

Unbounding quantification

WMSO+U $\longrightarrow$ deterministic max-automata

Proof strategy: Automata are closed under all operations in the logic.

Boolean operations: free for a deterministic automaton.

Let w be a word over alphabet Σ, and X a set of positions.
$w[X]$: word over alphabet $\Sigma\times\{0,1\}$

Weak existential quantification

Unbounding quantification

WMSO+U → deterministic max-automata

Proof strategy: Automata are closed under all operations in the logic.

Boolean operations: free for a deterministic automaton.

Let w be a word over alphabet Σ, and X a set of positions.
$w[X]$: word over alphabet $\Sigma\times\{0,1\}$

Weak existential quantification

Prop. If $L\subseteq (\Sigma\times\{0,1\})^\omega$ is recognized by a deterministic max-automaton, then so is $\{w : w[X]\in L$ for some finite set $X\} \subseteq \Sigma^\omega$

Unbounding quantification

WMSO+U $\longrightarrow$ deterministic max-automata

Proof strategy: Automata are closed under all operations in the logic.

Boolean operations: free for a deterministic automaton.

Let w be a word over alphabet Σ, and X a set of positions.
$w[X]$: word over alphabet $\Sigma\times\{0,1\}$

Weak existential quantification

Prop. If $L\subseteq (\Sigma\times\{0,1\})^\omega$ is recognized by a deterministic max-automaton, then so is $\{w : w[X]\in L$ for some finite set $X\} \subseteq \Sigma^\omega$

Unbounding quantification

Prop. If $L\subseteq (\Sigma\times\{0,1\})^\omega$ is recognized by a deterministic max-automaton, then so is $\{w : w[X]\in L$ for arbitrarily large $X\} \subseteq \Sigma^\omega$

WMSO+U $\longrightarrow$ deterministic max-automata

Proof strategy: Automata are closed under all operations in the logic.

Boolean operations: free for a deterministic automaton.

Let w be a word over alphabet Σ, and X a set of positions.
$w[X]$: word over alphabet $\Sigma\times\{0,1\}$

Weak existential quantification

Prop. If $L\subseteq (\Sigma\times\{0,1\})^\omega$ is recognized by a deterministic max-automaton, then so is $\{w : w[X]\in L$ for some finite set $X\} \subseteq \Sigma^\omega$

Unbounding quantification

Prop. If $L\subseteq (\Sigma\times\{0,1\})^\omega$ is recognized by a deterministic max-automaton, then so is $\{w : w[X]\in L$ for arbitrarily large $X\} \subseteq \Sigma^\omega$

The proof uses a combinatoric theorem of I. Simon.

Let A be an automaton with state space Q

Two rules for splitting words.

Let A be an automaton with state space Q

Two rules for splitting words.

Simon Theorem. For fixed A, there is a splitting depth K, such that every word can be split in depth K down to single letters.

Let A be an automaton with state space Q

Two rules for splitting words.

Rule 1.

split into two parts

abaabbbababbbabba | *bbabbbabbbabbaba*

Simon Theorem. For fixed A, there is a splitting depth K, such that every word can be split in depth K down to single letters.

Let A be an automaton with state space Q

Two rules for splitting words.

Rule 1. split into two parts

abaabbbababbbabba	*bbabbbabbbabbaba*

Rule 2. split into many parts, each with the same transformation $\delta : Q \to Q$

abaab	*bbababb*	*babba*	*bba*	*bbbabb*	*babba*	*ba*
δ	δ	δ	δ	δ	δ	δ

Simon Theorem. For fixed A, there is a splitting depth K, such that every word can be split in depth K down to single letters.

Let A be an automaton with state space Q

Two rules for splitting words.

Rule 1. split into two parts

abaabbbababbbabba	*bbabbbabbbabbaba*

Rule 2. split into many parts, each with the same transformation $\delta : Q \to Q$

abaab	*bbababb*	*babba*	*bba*	*bbbabb*	*babba*	*ba*
δ	δ	δ	δ	δ	δ	δ

$$\delta \circ \delta = \delta$$

Simon Theorem. For fixed A, there is a splitting depth K, such that every word can be split in depth K down to single letters.

“even number of *a*’s” has a decomposition of depth 5

two transition functions: even (0) and odd (1)

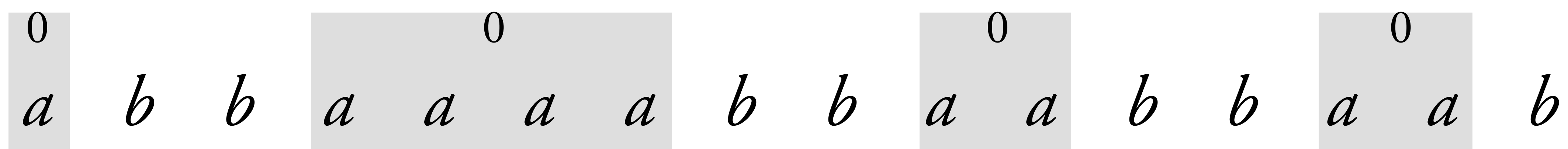
“even number of *a*’s” has a decomposition of depth 5

two transition functions: even (0) and odd (1)

a *b* *b* *a* *a* *a* *a* *b* *b* *a* *a* *b* *b* *a* *a* *b*

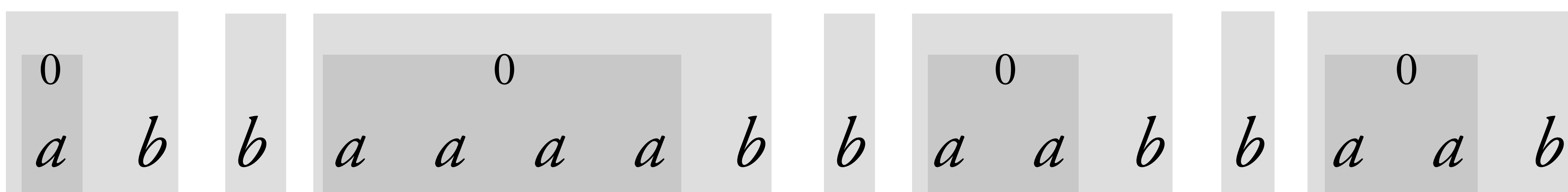
“even number of *a*’s” has a decomposition of depth 5

two transition functions: even (0) and odd (1)



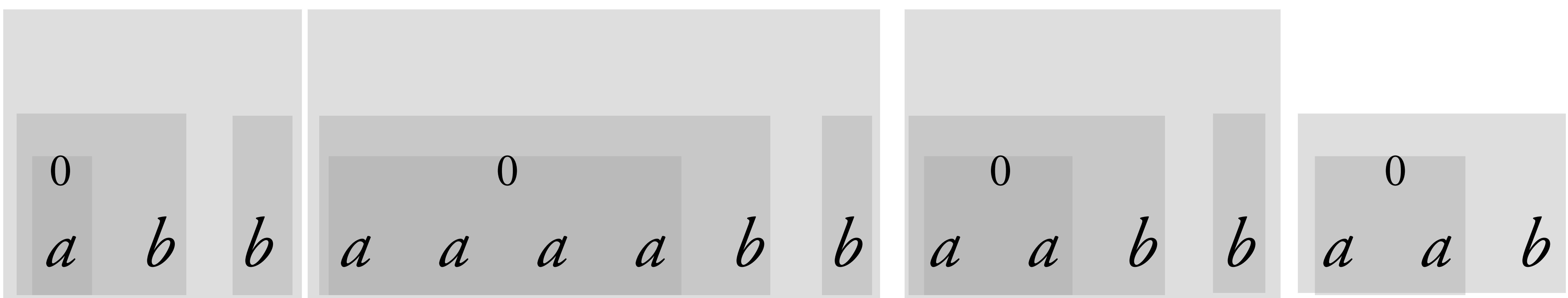
“even number of *a*’s” has a decomposition of depth 5

two transition functions: even (0) and odd (1)



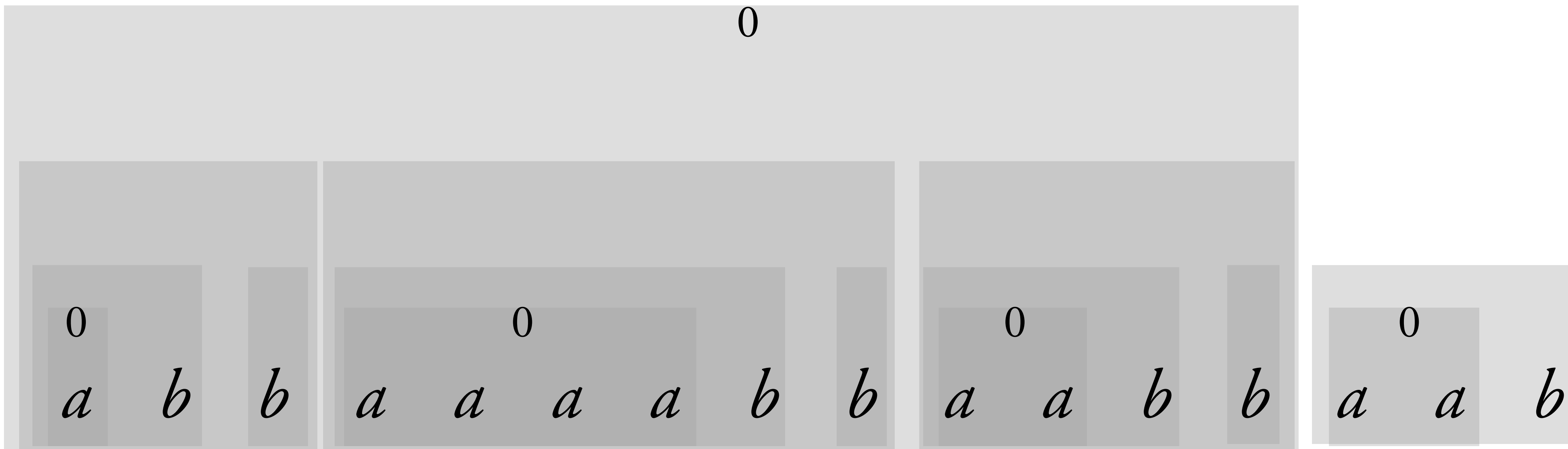
“even number of *a*’s” has a decomposition of depth 5

two transition functions: even (0) and odd (1)



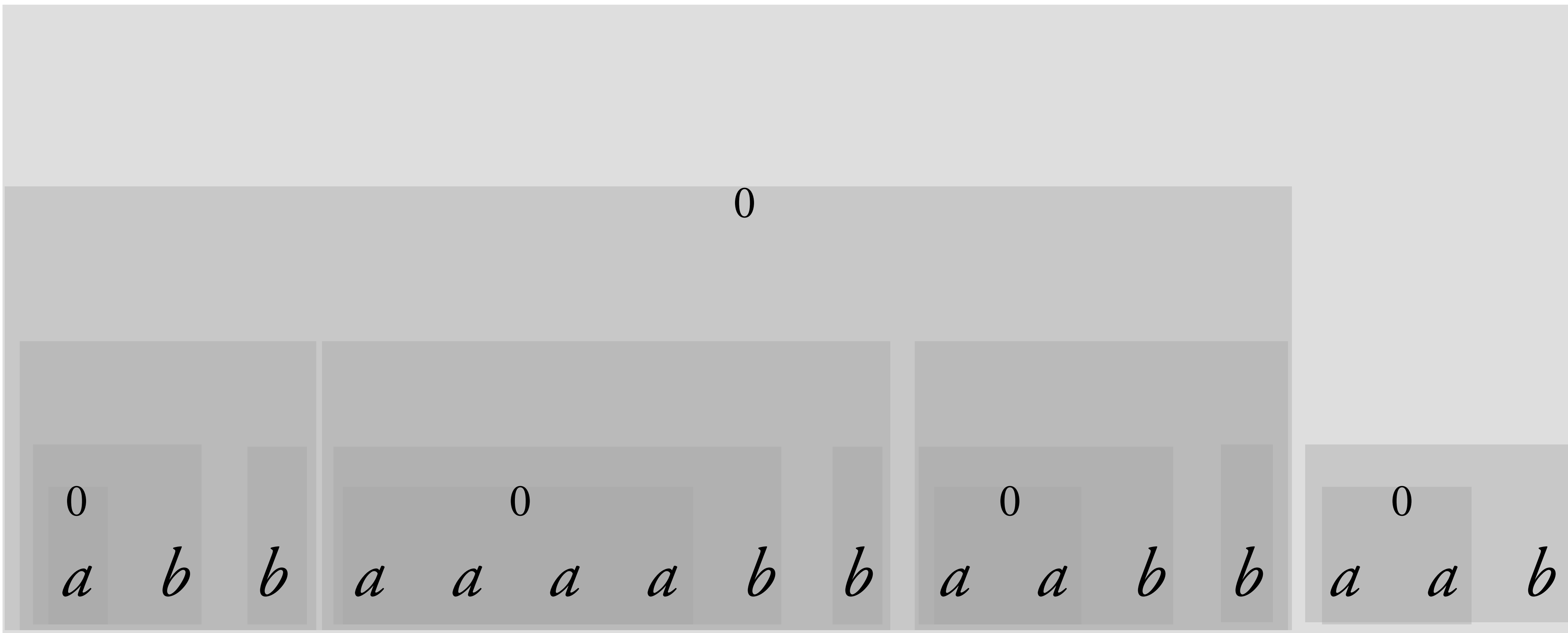
“even number of *a*’s” has a decomposition of depth 5

two transition functions: even (0) and odd (1)



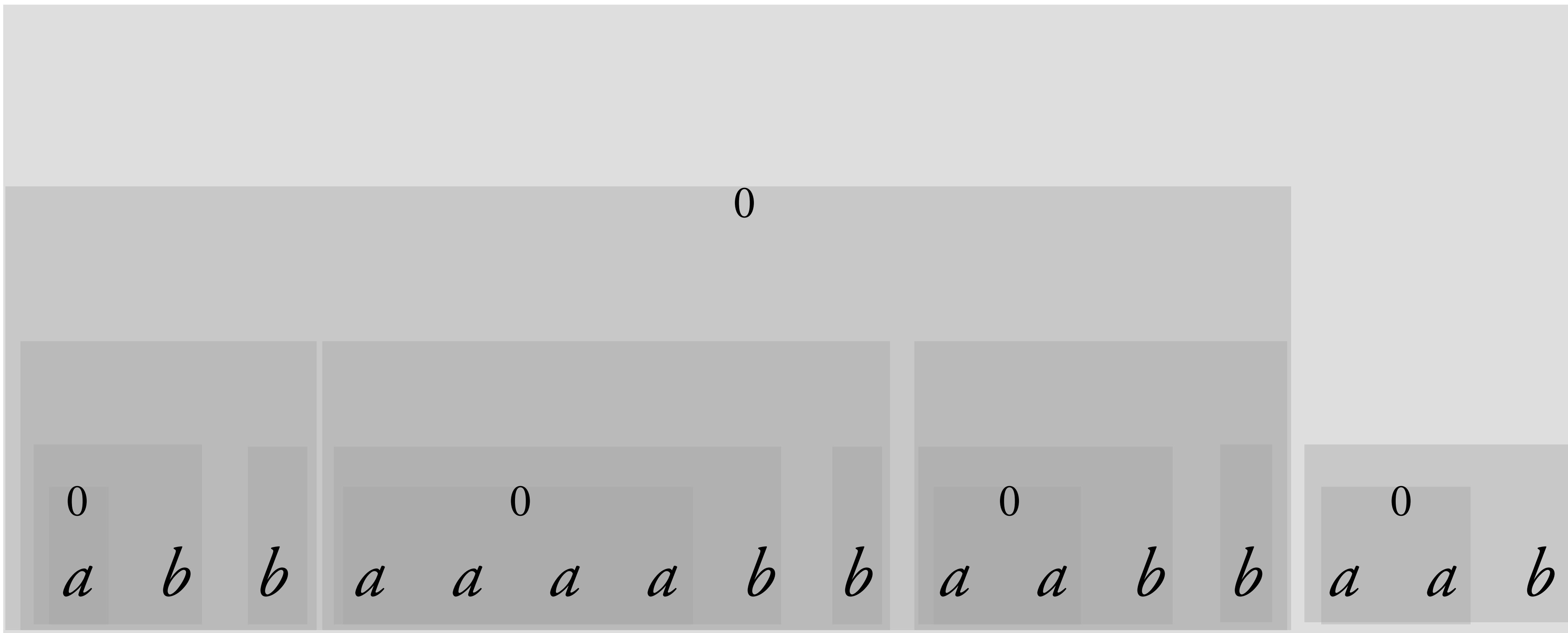
"even number of *a*'s" has a decomposition of depth 5

two transition functions: even (0) and odd (1)



"even number of *a*'s" has a decomposition of depth 5

two transition functions: even (0) and odd (1)



Thm. (Colcombet '07)
The decomposition can be output by a deterministic finite state transducer.

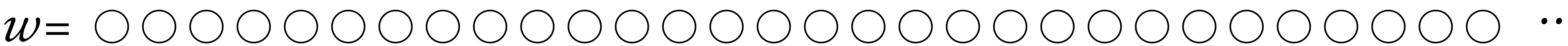
Prop. If $L \subseteq (\Sigma \times \{0,1\})^\omega$ is recognized by a deterministic max-automaton, then so is $\{w : w[X] \in L \text{ for arbitrarily large } X\} \subseteq \Sigma^\omega$

Prop. If $L \subseteq (\Sigma\times\{0,1\})^\omega$ is recognized by a deterministic max-automaton, then so is $\{w : w[X]\in L \text{ for arbitrarily large } X\} \subseteq \Sigma^\omega$

given a word $w\in\Sigma^\omega$, how can a deterministic automaton tell if $w[X]\in L$ holds for arbitrarily large X?

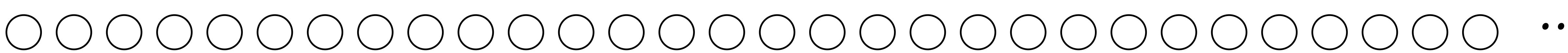
Prop. If $L \subseteq (\Sigma \times \{0,1\})^\omega$ is recognized by a deterministic max-automaton, then so is $\{w : w[X] \in L$ for arbitrarily large $X\} \subseteq \Sigma^\omega$

given a word $w \in \Sigma^\omega$, how can a deterministic automaton tell if $w[X] \in L$ holds for arbitrarily large X?



Prop. If $L \subseteq (\Sigma \times \{0,1\})^\omega$ is recognized by a deterministic max-automaton, then so is $\{w : w[X] \in L$ for arbitrarily large $X\} \subseteq \Sigma^\omega$

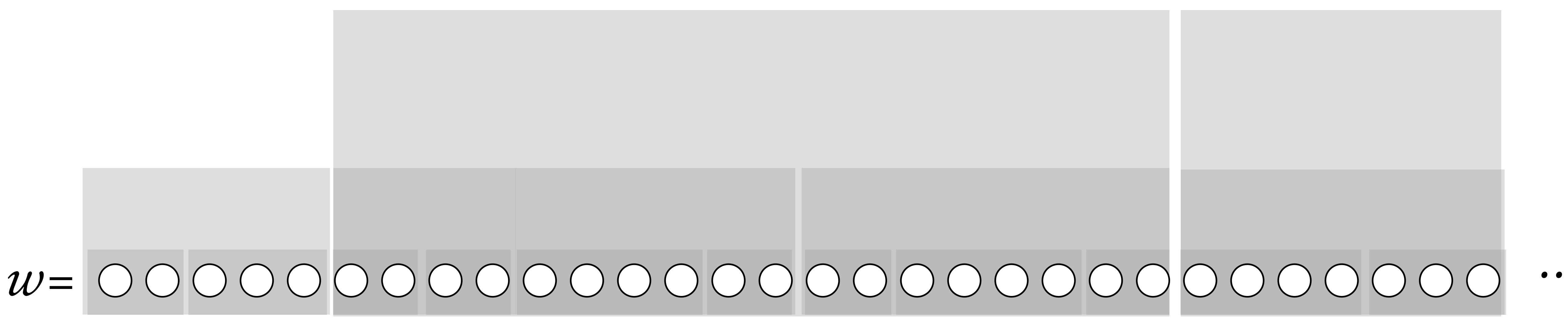
given a word $w \in \Sigma^\omega$, how can a deterministic automaton tell if $w[X] \in L$ holds for arbitrarily large X?

$w =$ 

1. compute Simon decomposition for well chosen automaton (a modification of the automaton for L)

Prop. If $L \subseteq (\Sigma\times\{0,1\})^\omega$ is recognized by a deterministic max-automaton, then so is $\{w : w[X]\in L \text{ for arbitrarily large } X\} \subseteq \Sigma^\omega$

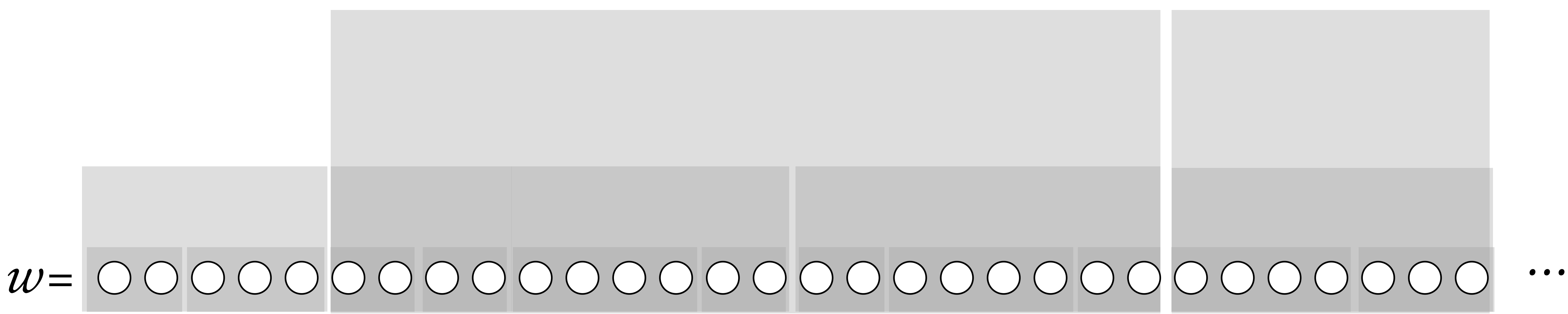
given a word $w\in\Sigma^\omega$, how can a deterministic automaton tell if $w[X]\in L$ holds for arbitrarily large X?



1. compute Simon decomposition for well chosen automaton (a modification of the automaton for L)

Prop. If $L \subseteq (\Sigma \times \{0,1\})^\omega$ is recognized by a deterministic max-automaton, then so is $\{w : w[X] \in L$ for arbitrarily large $X\} \subseteq \Sigma^\omega$

given a word $w \in \Sigma^\omega$, how can a deterministic automaton tell if $w[X] \in L$ holds for arbitrarily large X?



1. compute Simon decomposition for well chosen automaton (a modification of the automaton for L)

2. find large boxes