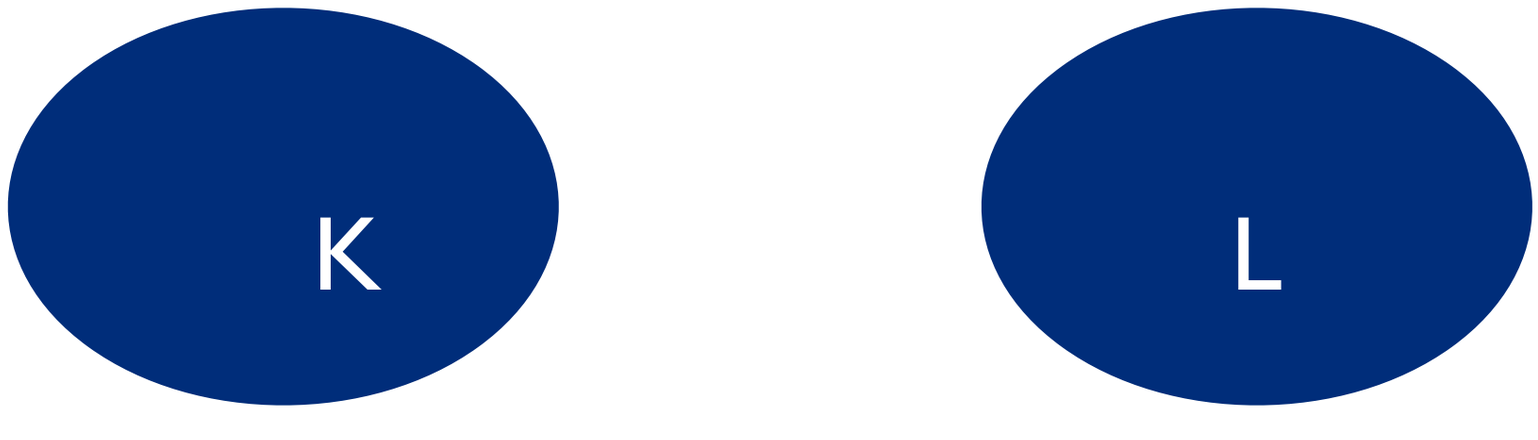
Regular Separability of One Counter Automata

Wojciech Czerwiński

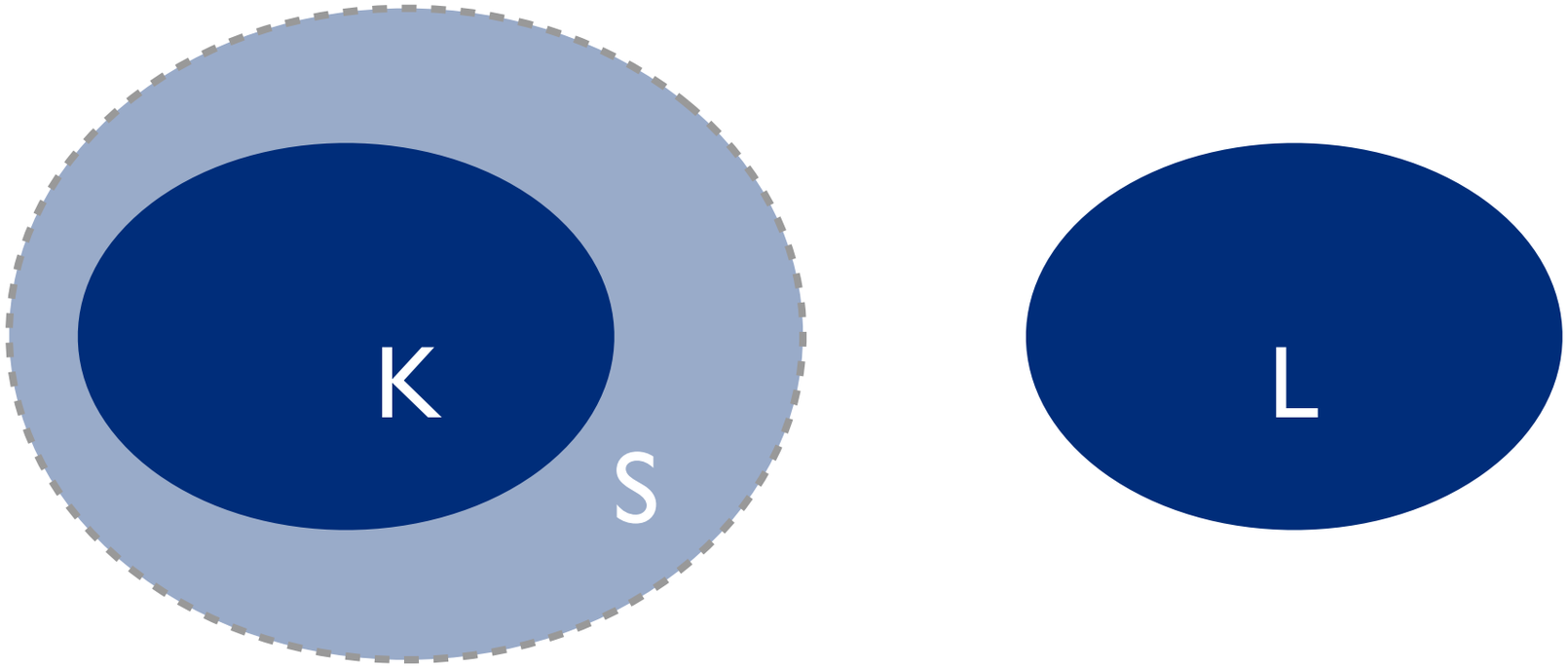
Sławomir Lasota

Separability

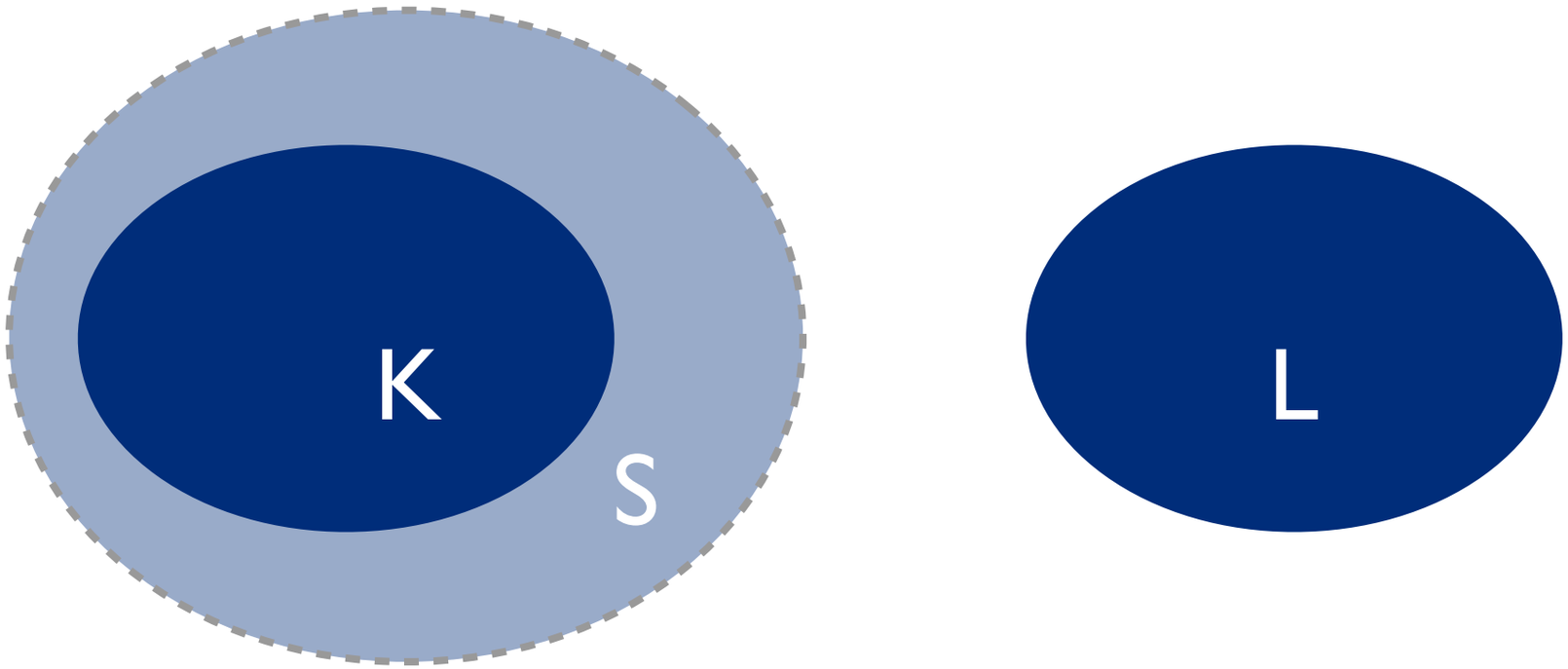
Separability



Separability

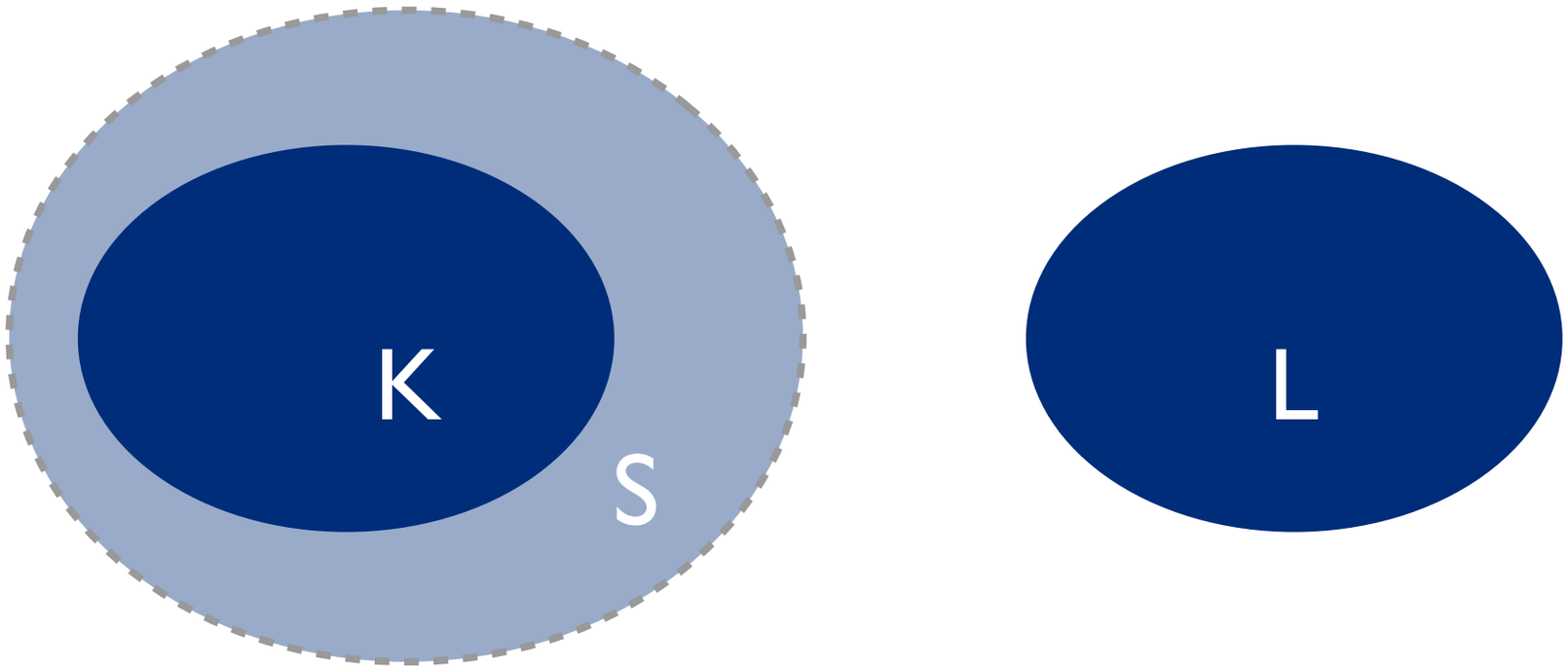


Separability



S *separates* K and L

Separability



S *separates* K and L

K and L are *regular separable*
if some regular S separates them

One counter automata

One counter automata

configuration: state + one nonnegative counter

One counter automata

configuration: state + one nonnegative counter

transition: based on state chooses new state and counter change (in binary)

One counter automata

configuration: state + one nonnegative counter

transition: based on state chooses new state
and counter change (in binary)

zero test: transition fireable only when counter is zero

One counter automata

configuration: state + one nonnegative counter

transition: based on state chooses new state
and counter change (in binary)

zero test: transition fireable only when counter is zero

zero tests allowed: one counter automata

disallowed: one counter nets

One counter automata

configuration: state + one nonnegative counter

transition: based on state chooses new state
and counter change (in binary)

zero test: transition fireable only when counter is zero

zero tests allowed: one counter automata
disallowed: one counter nets

language: single source and target configuration
(assume counter 0), every transition labelled

Problem(s)

Problem(s)

Given: two one counter nets (automata) A and B

Problem(s)

Given: two one counter nets (automata) A and B

Question: are L(A) and L(B) regular separable?

Main result

Main result

Theorem:

Regular separability of
languages of one counter nets
is decidable

Main result

Theorem:

Regular separability of
languages of one counter nets
is decidable

Remark: even PSPACE-complete

Another result

Another result

Theorem:

Regular separability of
languages of one counter automata
is undecidable

Another result

Theorem:

Regular separability of
languages of one counter automata
is undecidable

Remark: even by any F closed
under boolean combination and containing $w\Sigma^*$

Proof idea: approximants

Proof idea: approximants

Idea: canonical separators

Proof idea: approximants

Idea: canonical separators

For OCN A and number n define automaton A_n such that:

Proof idea: approximants

Idea: canonical separators

For OCN A and number n define automaton A_n such that:

1) $L(A_n)$ regular

Proof idea: approximants

Idea: canonical separators

For OCN A and number n define automaton A_n such that:

1) L(A_n) regular

2) L(A_n) includes L(A)

Proof idea: approximants

Idea: canonical separators

For OCN A and number n define automaton A_n such that:

1) $L(A_n)$ regular

2) $L(A_n)$ includes $L(A)$

3) for m dividing n holds $L(A_m)$ includes $L(A_n)$

Proof idea: approximants

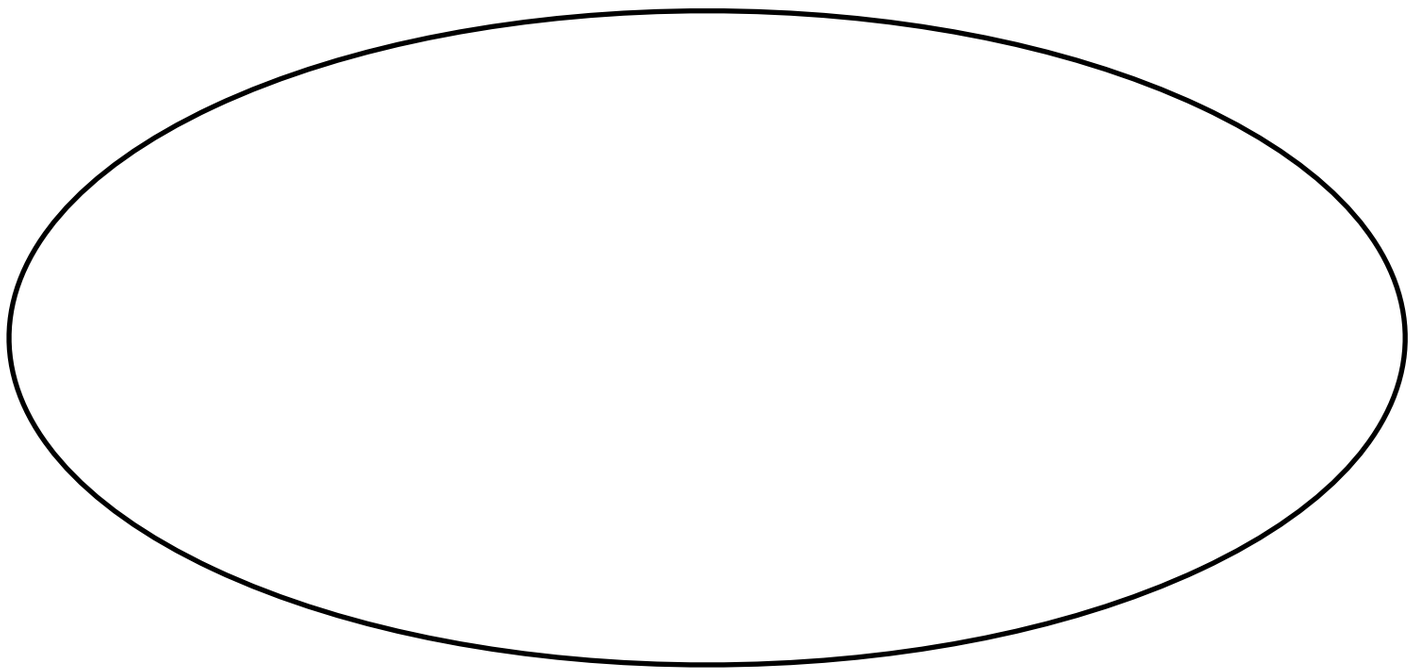
Idea: canonical separators

For OCN A and number n define automaton A_n such that:

1) L(A_n) regular

2) L(A_n) includes L(A)

3) for m dividing n holds L(A_m) includes L(A_n)



Proof idea: approximants

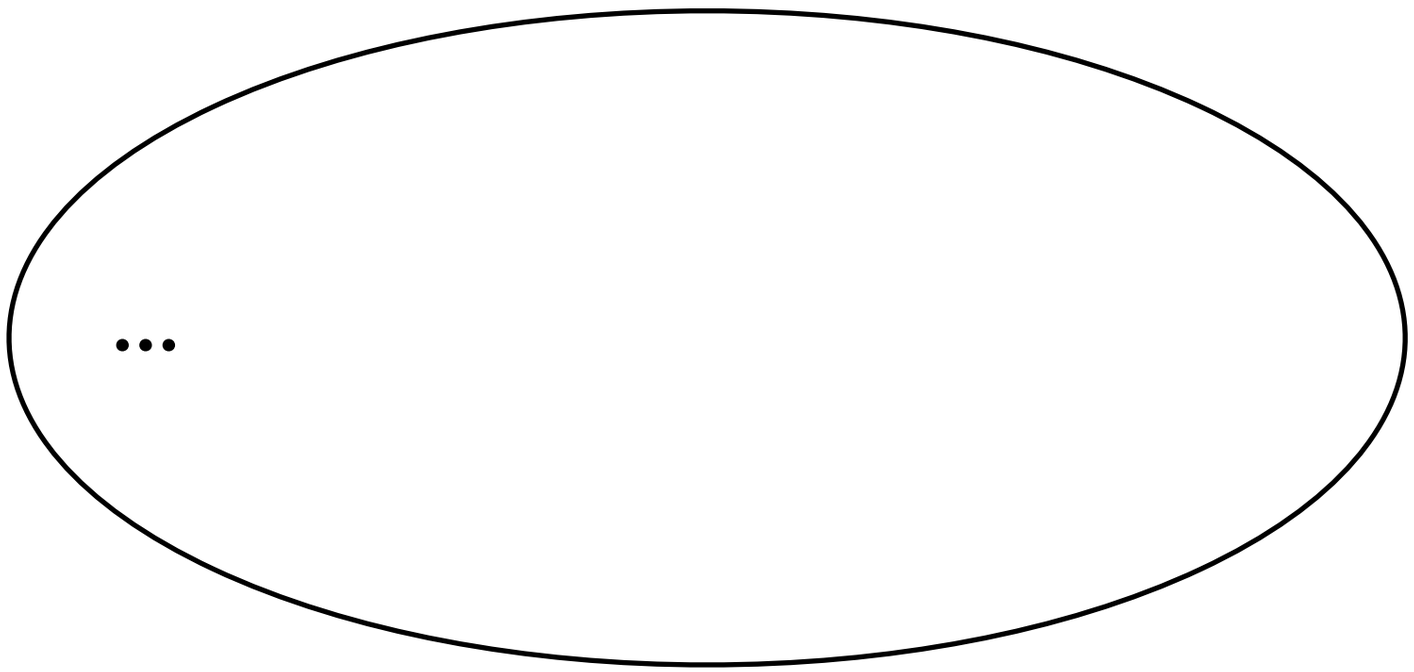
Idea: canonical separators

For OCN A and number n define automaton A_n such that:

1) L(A_n) regular

2) L(A_n) includes L(A)

3) for m dividing n holds L(A_m) includes L(A_n)



Proof idea: approximants

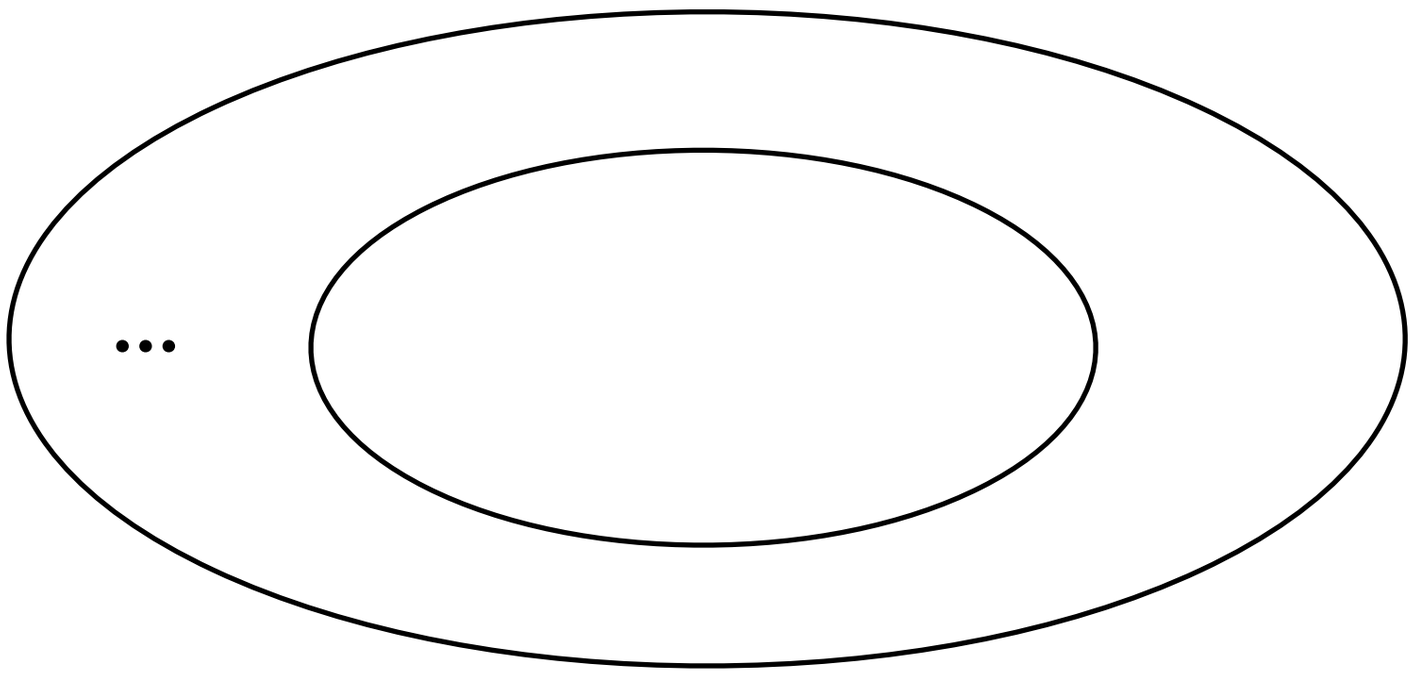
Idea: canonical separators

For OCN A and number n define automaton A_n such that:

1) L(A_n) regular

2) L(A_n) includes L(A)

3) for m dividing n holds L(A_m) includes L(A_n)



Proof idea: approximants

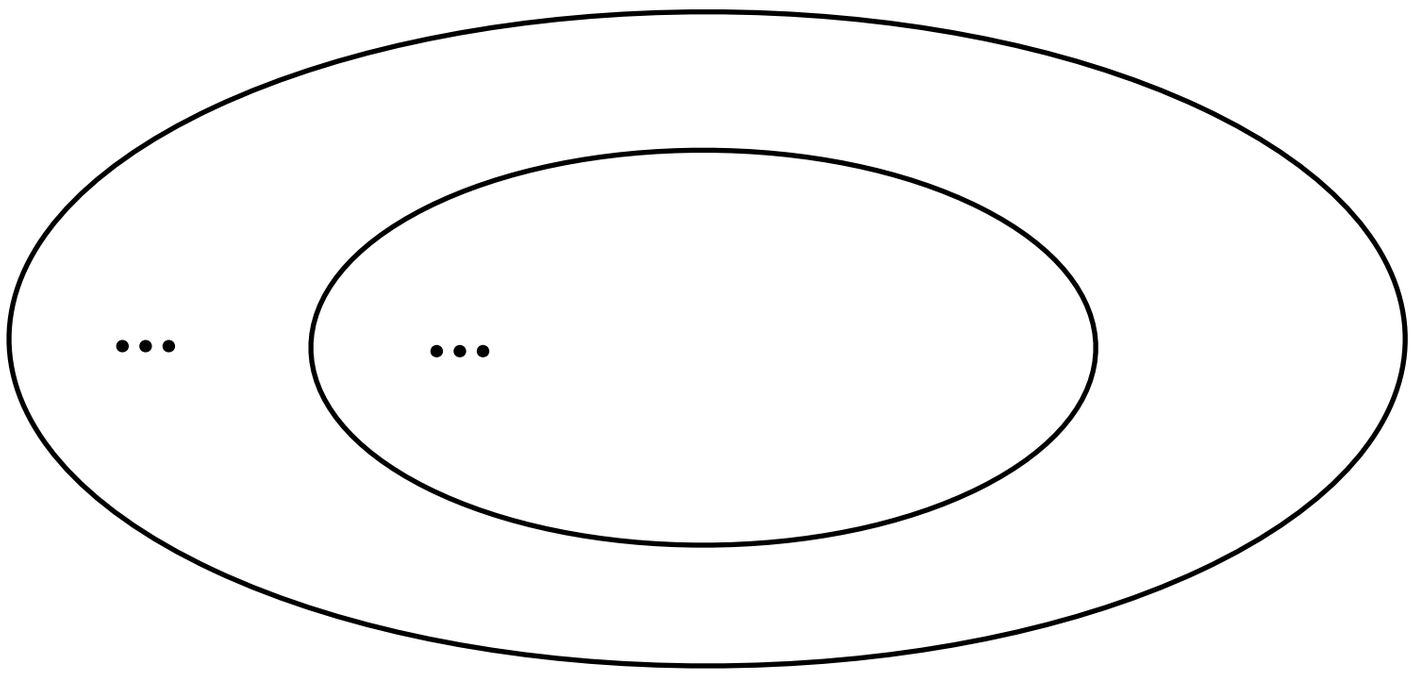
Idea: canonical separators

For OCN A and number n define automaton A_n such that:

1) L(A_n) regular

2) L(A_n) includes L(A)

3) for m dividing n holds L(A_m) includes L(A_n)



Proof idea: approximants

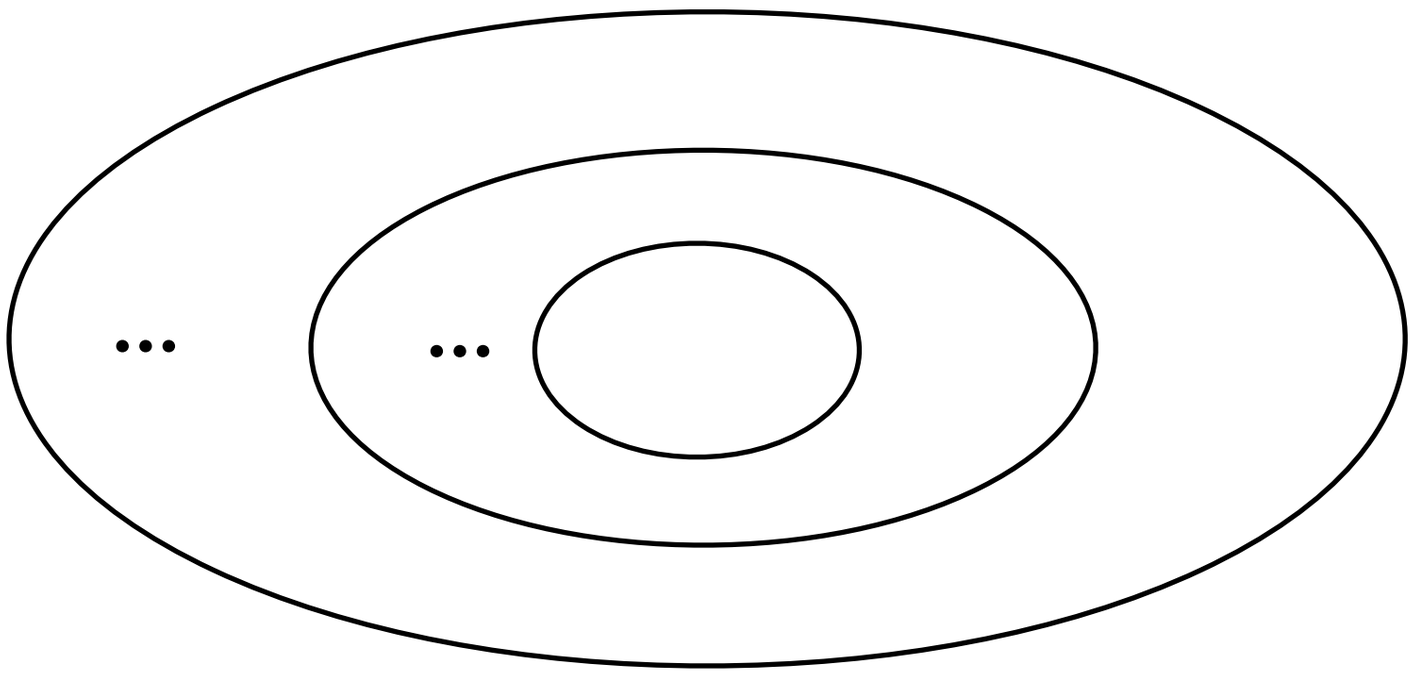
Idea: canonical separators

For OCN A and number n define automaton A_n such that:

1) $L(A_n)$ regular

2) $L(A_n)$ includes $L(A)$

3) for m dividing n holds $L(A_m)$ includes $L(A_n)$



Proof idea: approximants

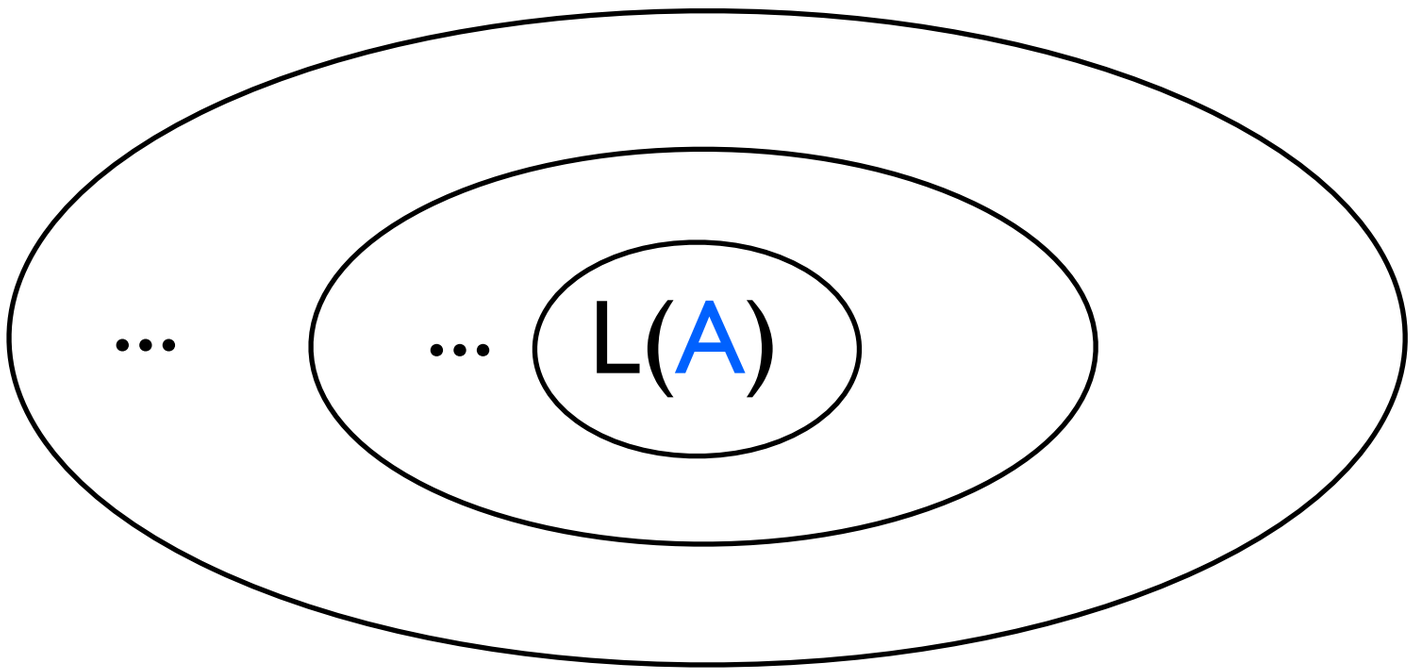
Idea: canonical separators

For OCN A and number n define automaton A_n such that:

1) L(A_n) regular

2) L(A_n) includes L(A)

3) for m dividing n holds L(A_m) includes L(A_n)



Approximation lemma

Approximation lemma

Lemma

Approximation lemma

Lemma

For two OCNs A and B tfae:

Approximation lemma

Lemma

For two OCNs A and B tfae:

1) L(A) and L(B) are regular separable

Approximation lemma

Lemma

For two OCNs A and B tfae:

1) L(A) and L(B) are regular separable

2) there exists n s.t. $L(A_n)$ and $L(B_n)$ disjoint

Approximation lemma

Lemma

For two OCNs A and B tfae:

1) L(A) and L(B) are regular separable

2) there exists n s.t. $L(A_n)$ and $L(B_n)$ disjoint

One direction clear!

Algorithm

Algorithm

Checks whether there is some n such that L(A_n) and L(B_n) disjoint

Algorithm

Checks whether there is some n such that L(A_n) and L(B_n) disjoint

Quite standard

Algorithm

Checks whether there is some n such that L(A_n) and L(B_n) disjoint

Quite standard

Semininear set techniques - PSPACE

Hardness

Hardness

Undecidability and PSPACE-hardness - similar techniques

Hardness

Undecidability and PSPACE-hardness - similar techniques

Old technique by Hunt

Hardness

Undecidability and PSPACE-hardness - similar techniques

Old technique by Hunt

Reduction from every decidable problem
(of bounded blowup)

Thank you!