Regular Separability of One Counter Automata

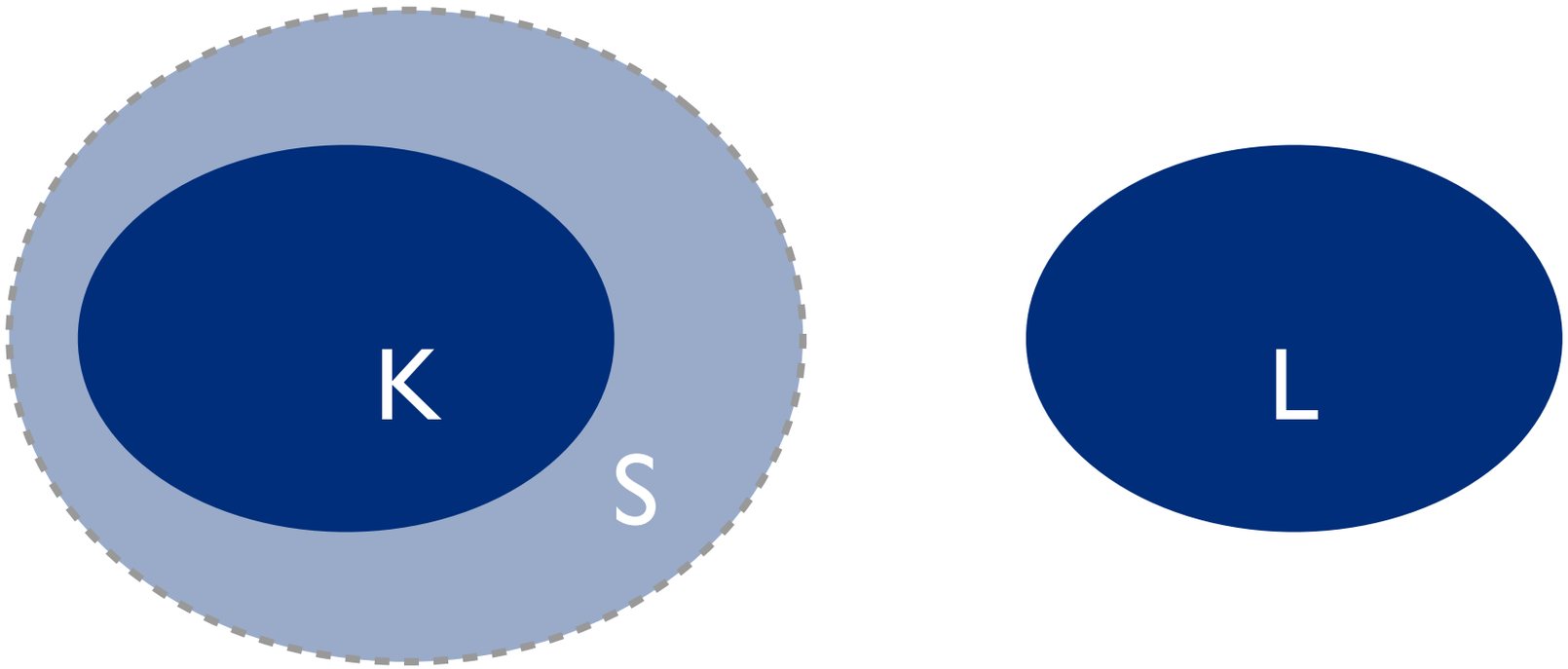
Wojciech Czerwiński

Sławomir Lasota

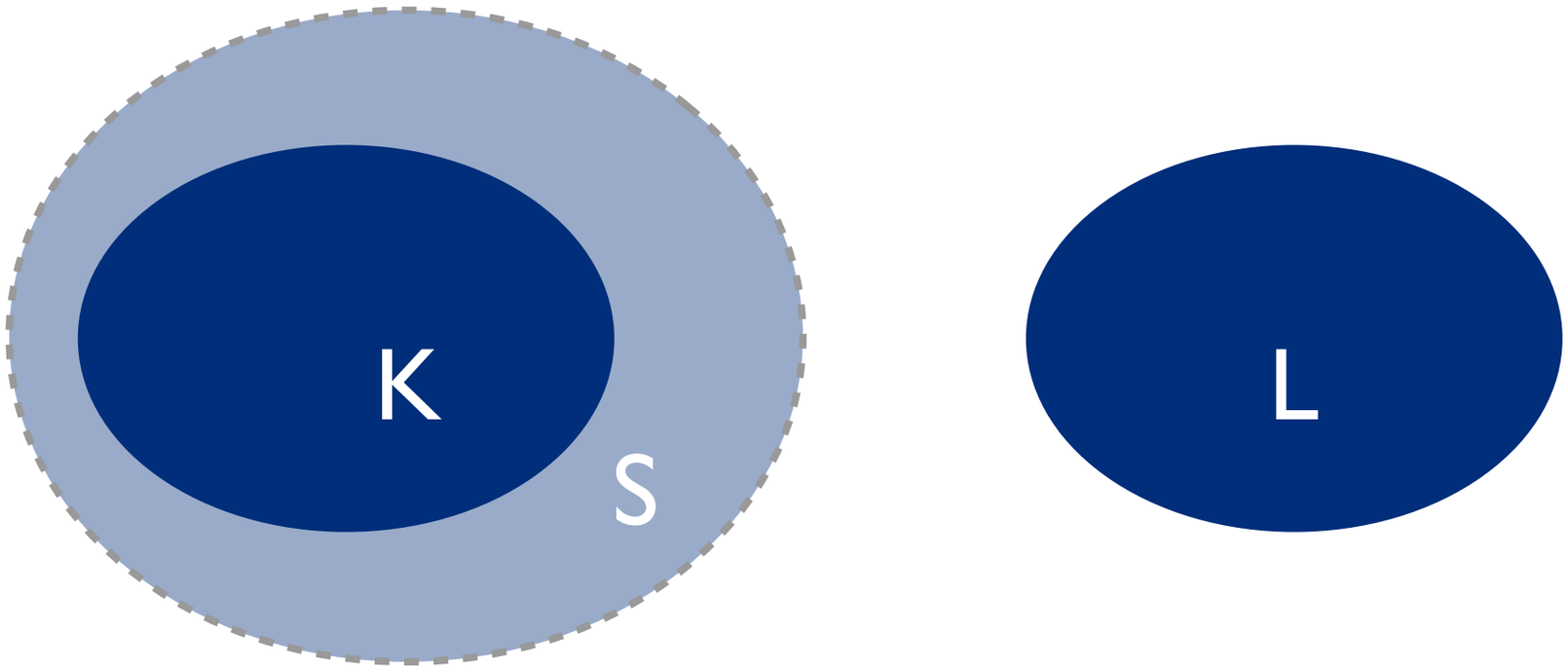
Separability

Separability

Separability

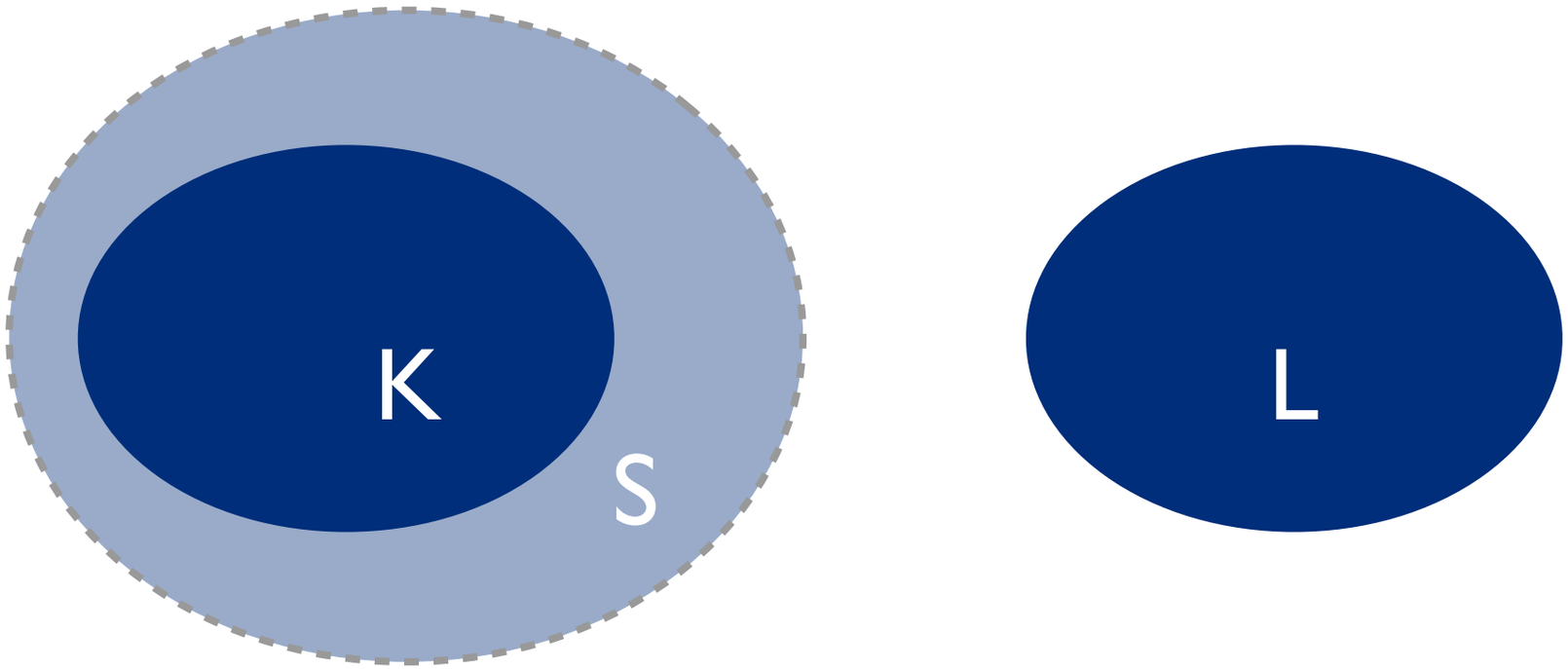


Separability



S *separates* K and L

Separability



S *separates* K and L

K and L are *separable by* family F
if some S from F separates them

General problem

F separability of G

General problem

F separability of G

Given: two languages K and L from family G

General problem

F separability of G

Given: two languages K and L from family G

Question: are K and L separable by some language from family F

History of separability

History of separability

- till recently not a lot of interest

History of separability

- till recently not a lot of interest
- recently many decidability results for F sep. of regular languages for F = languages of FO, Σ_i, Π_i

History of separability

- till recently not a lot of interest
- recently many decidability results for F sep. of regular languages for F = languages of FO, Σ_i, Π_i
- mostly obtained by algebraic methods

History of separability

- till recently not a lot of interest
- recently many decidability results for F sep. of regular languages for F = languages of FO, Σ_i, Π_i
- mostly obtained by algebraic methods
- regular separability of CFL is undecidable (Szymański, Williams '76)

History of separability

History of separability

- F sep. of CFL is undecidable for any F closed under boolean combination and containing $w\Sigma^*$ (Hunt '82)

History of separability

- F sep. of CFL is undecidable for any F closed under boolean combination and containing $w\Sigma^*$ (Hunt '82)
- PTL separability of CFL is decidable (Cz. et al. '15)

History of separability

- F sep. of CFL is undecidable for any F closed under boolean combination and containing $w\Sigma^*$ (Hunt '82)
- PTL separability of CFL is decidable (Cz. et al. '15)
- regular separability of visibly pushdown languages is undecidable (Kopczyński '15)

Motivation

Motivation

- understand for which classes regular separability is decidable

Motivation

- understand for which classes regular separability is decidable
- extending towards stack is hopeless (Kopczyński ’15)

Motivation

- understand for which classes regular separability is decidable
- extending towards stack is hopeless (Kopczyński ’15)
- maybe counters?

Motivation

- understand for which classes regular separability is decidable
- extending towards stack is hopeless (Kopczyński ’15)
- maybe counters?
- conjecture: decidable for VAS-languages

Motivation

- understand for which classes regular separability is decidable
- extending towards stack is hopeless (Kopczyński '15)
- maybe counters?
- conjecture: decidable for VAS-languages
- this talk: one counter languages

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

One counter automata

configuration: state + one nonnegative counter

transition: based on state chooses new state and counter change

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

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

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 integer n define automaton A_n such that:

Proof idea: approximants

Idea: canonical separators

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

1) $L(A_n)$ regular

Proof idea: approximants

Idea: canonical separators

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

1) L(A_n) regular

2) L(A_n) includes L(A)

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

Proof idea: approximants

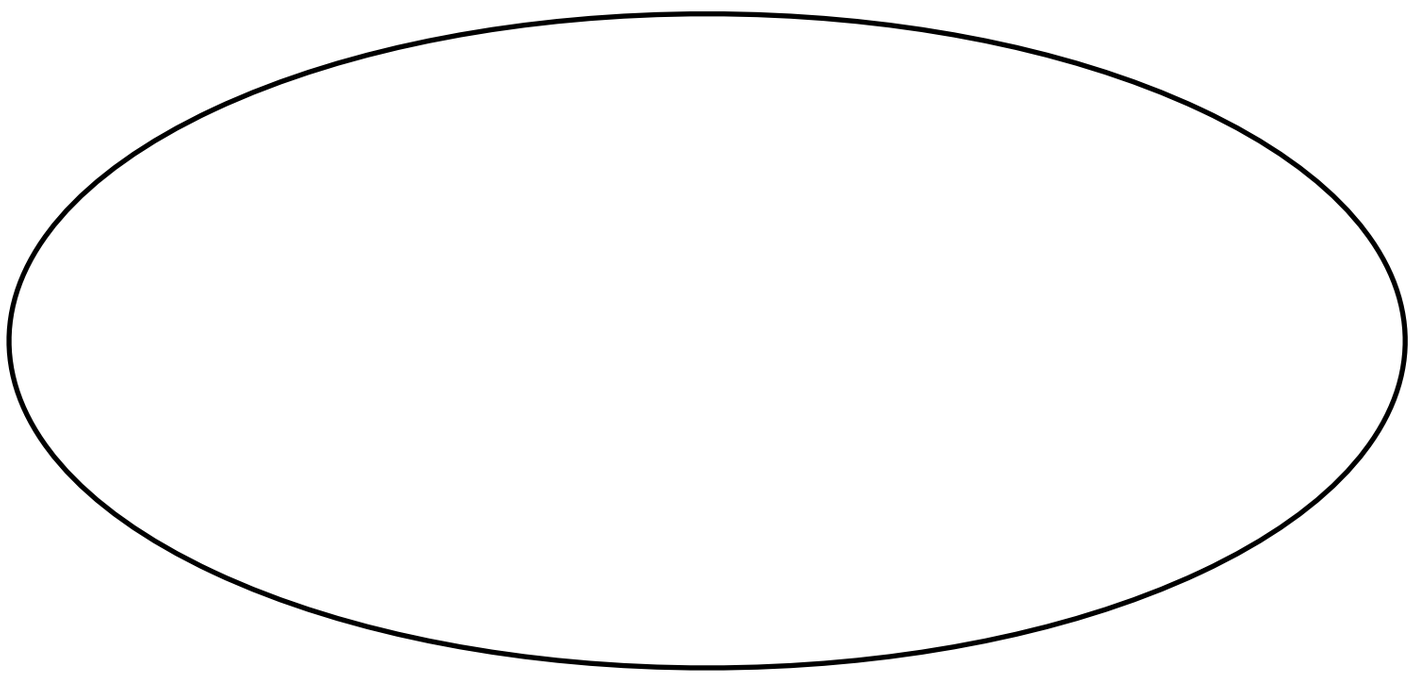
Idea: canonical separators

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

1) L(A_n) regular

2) L(A_n) includes L(A)

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



Proof idea: approximants

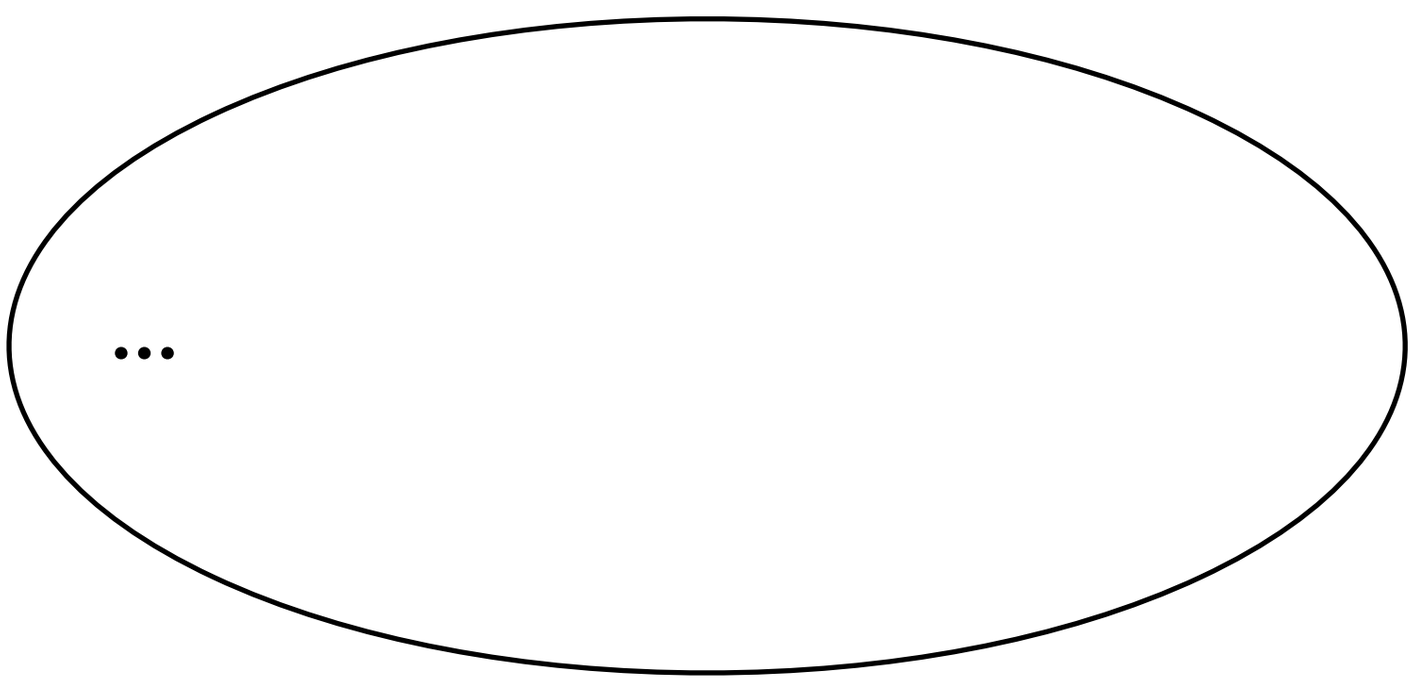
Idea: canonical separators

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

1) L(A_n) regular

2) L(A_n) includes L(A)

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



Proof idea: approximants

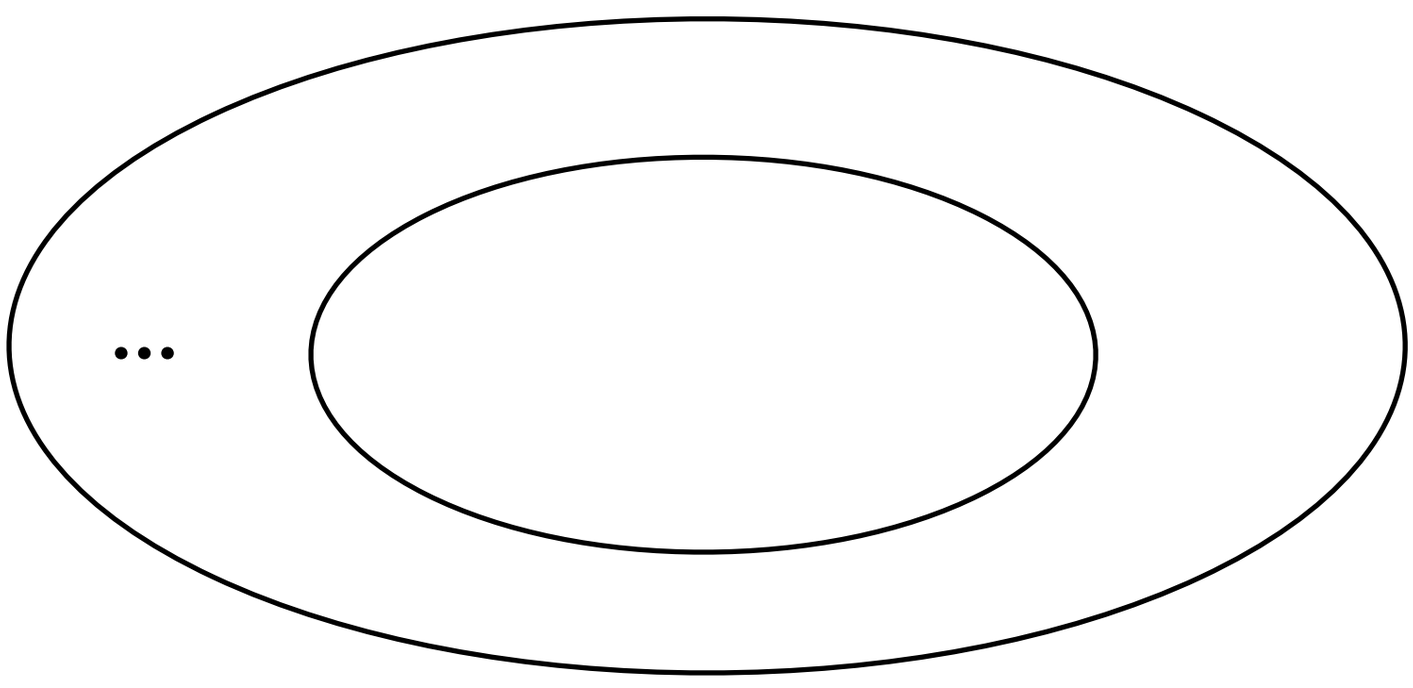
Idea: canonical separators

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

1) L(A_n) regular

2) L(A_n) includes L(A)

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



Proof idea: approximants

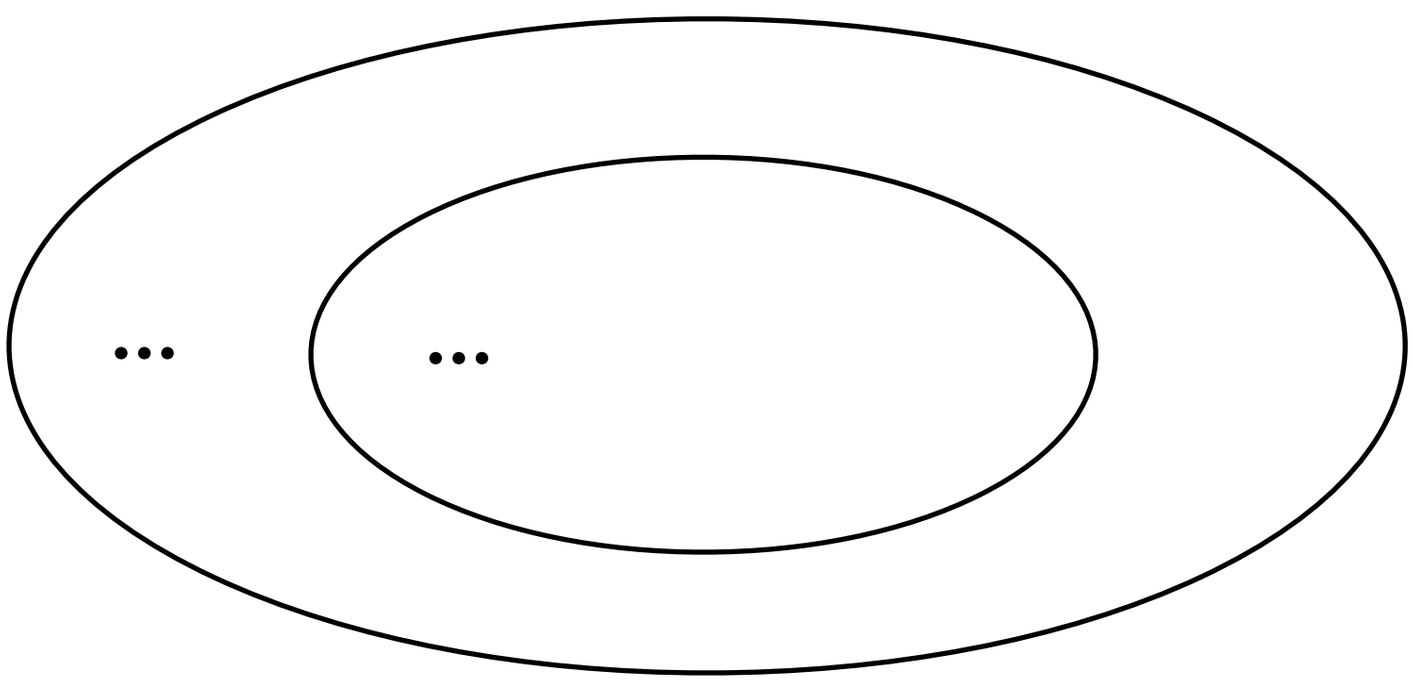
Idea: canonical separators

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

1) $L(A_n)$ regular

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

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



Proof idea: approximants

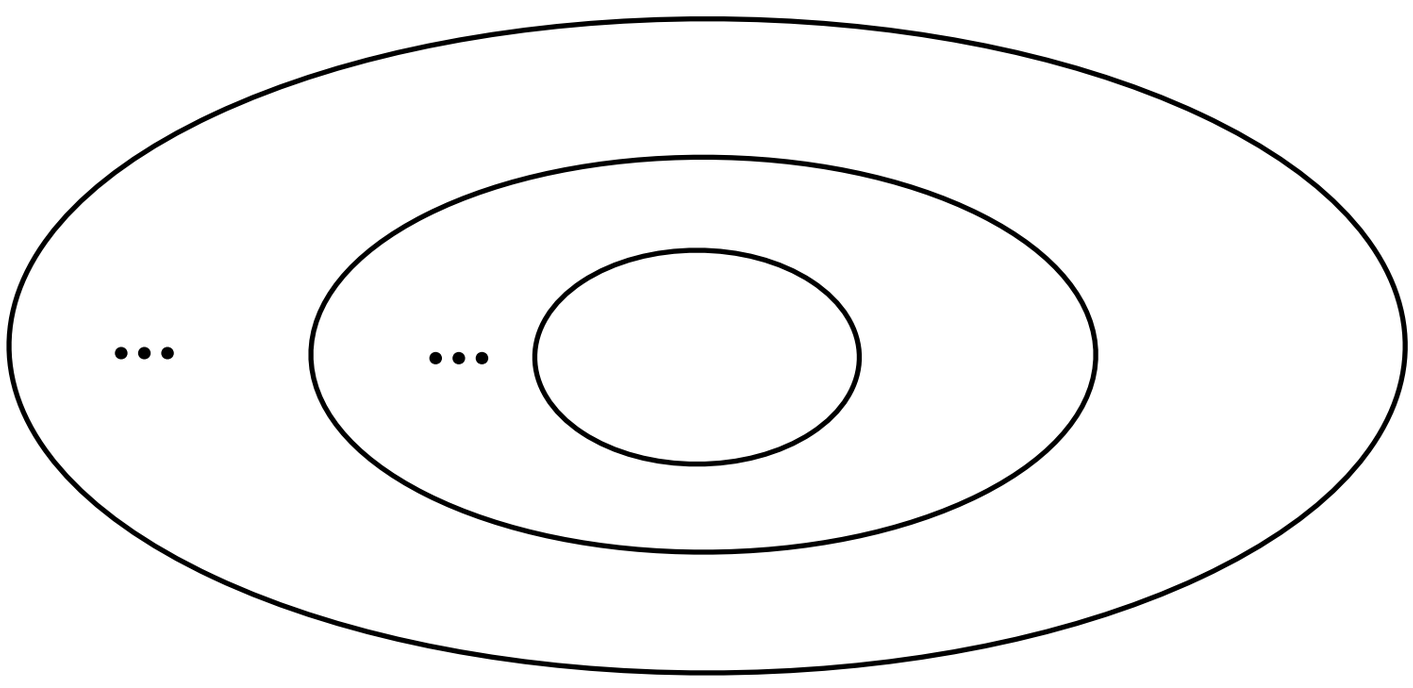
Idea: canonical separators

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

1) L(A_n) regular

2) L(A_n) includes L(A)

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



Proof idea: approximants

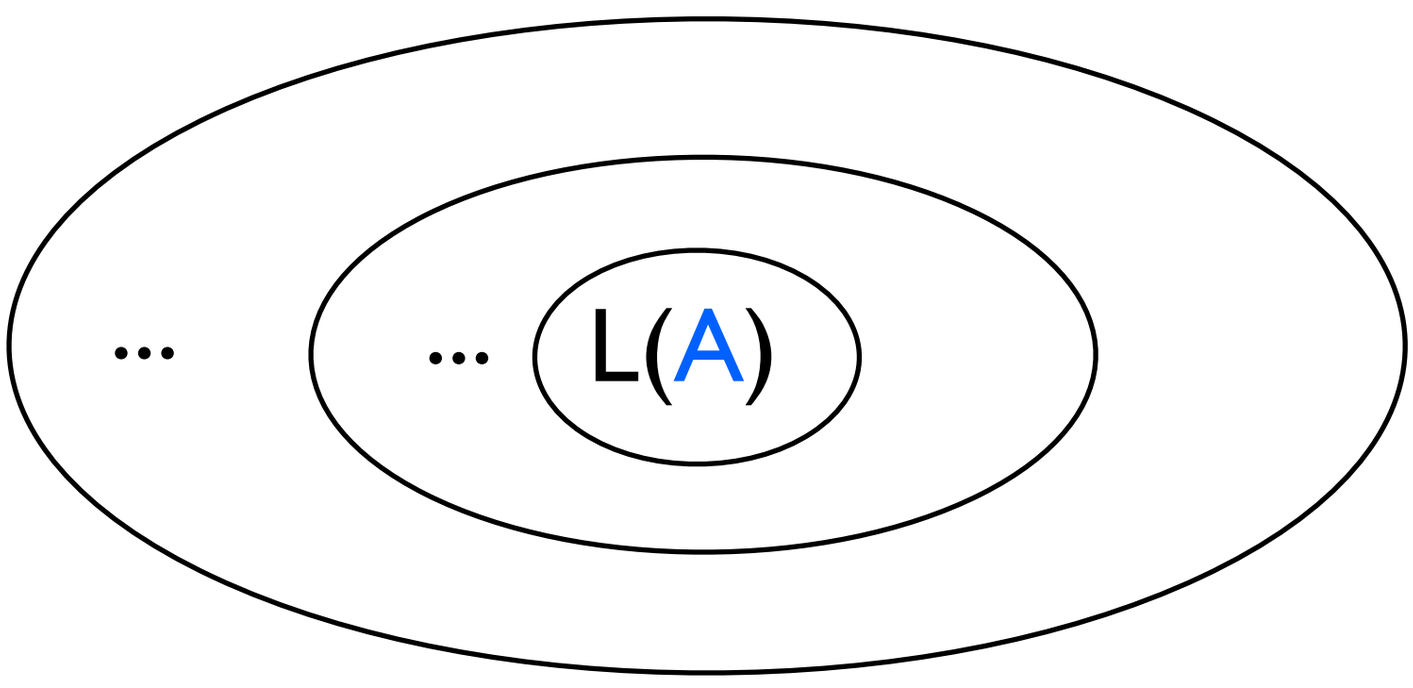
Idea: canonical separators

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

1) L(A_n) regular

2) L(A_n) includes L(A)

3) for m | 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!

Definition

Definition

A - set of states: Q

Definition

A - set of states: Q

A_n - set of states: $Q \times \{0, ..., n-1\} \times \{small, big\}$

Definition

A - set of states: Q

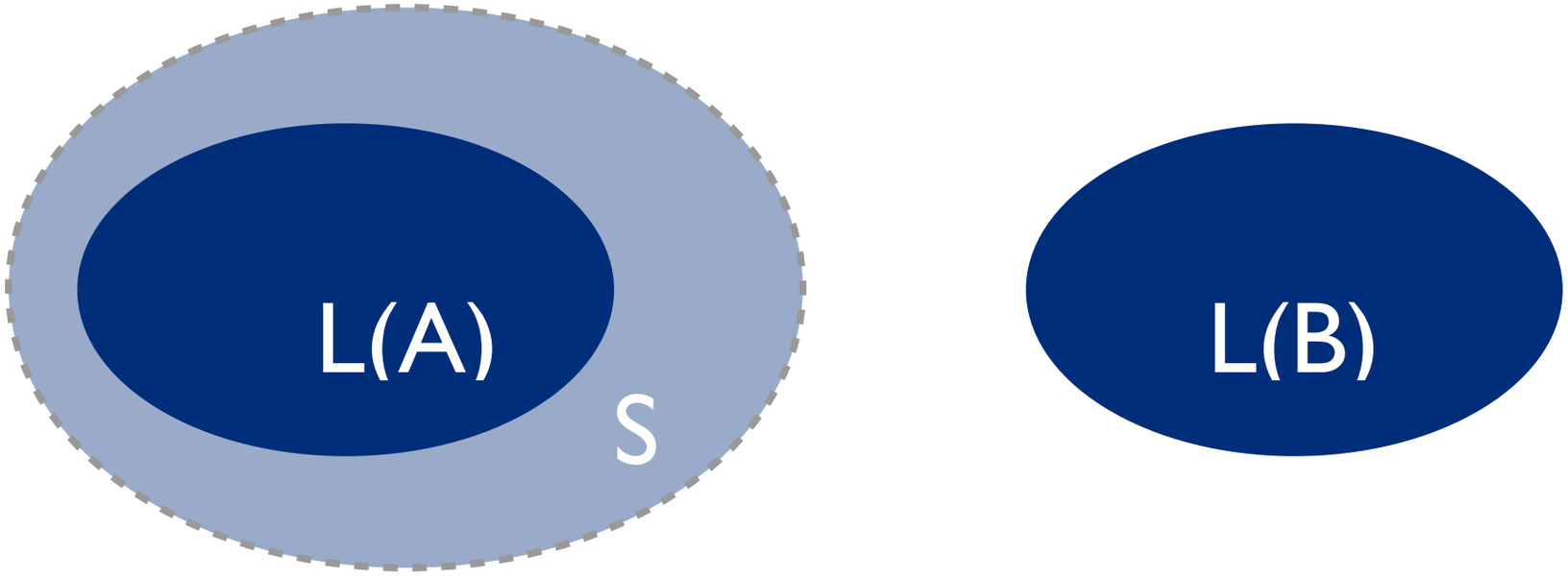
A_n - set of states: Q x {0, ..., n-1} x {small, big}

A_n remembers:

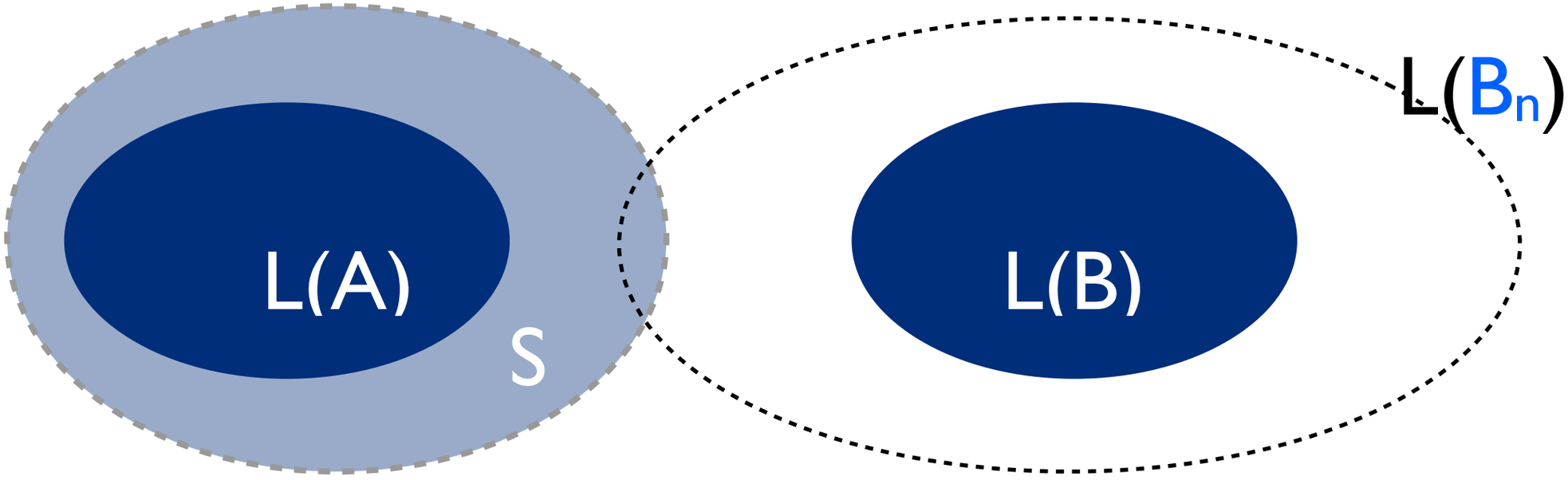
- state of A
- counter value modulo n
- is counter smaller than n? (maybe wrong)

Proof idea

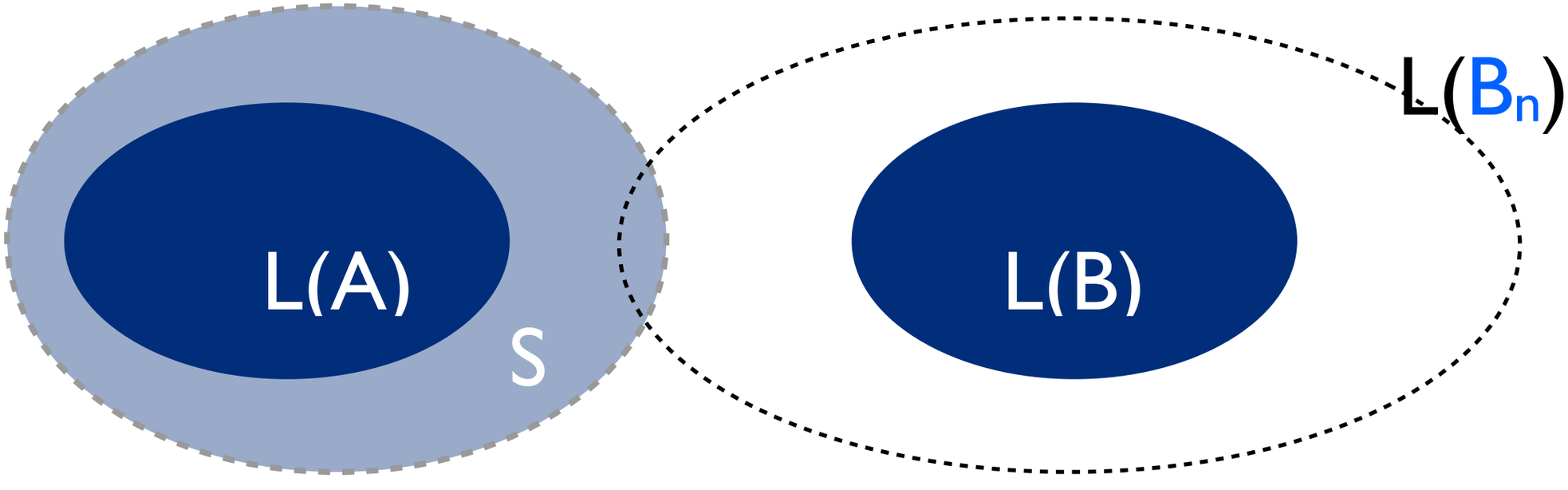
Proof idea



Proof idea

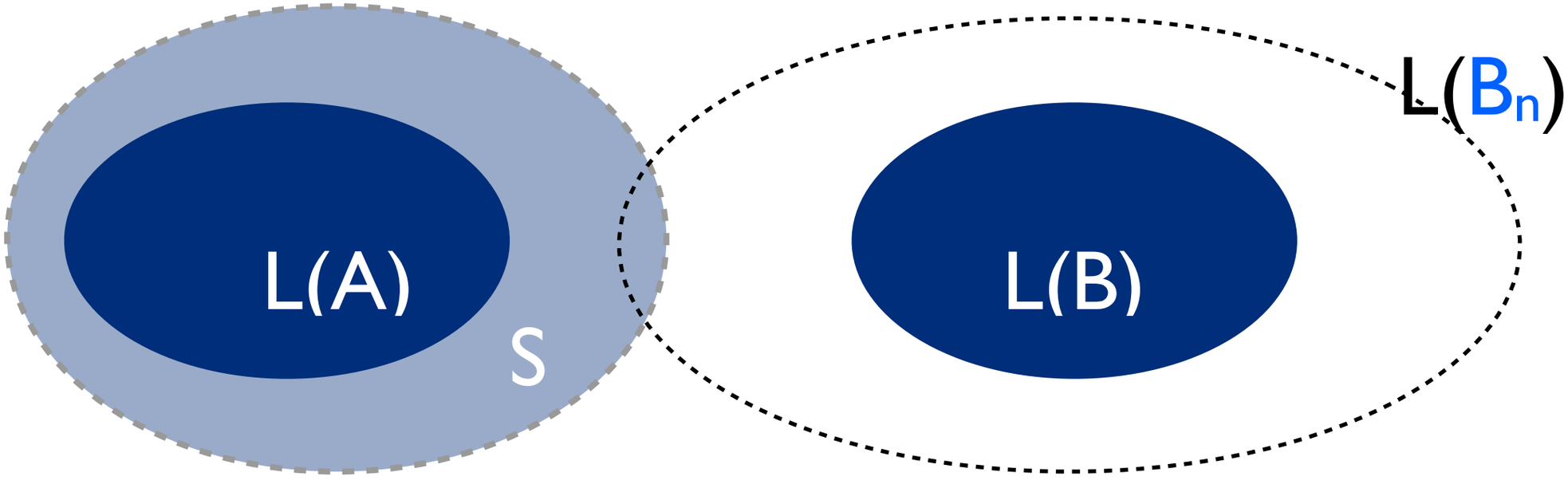


Proof idea



For big n every run of B_n labelled by word in S can be modified to a run of B still labelled by word in S

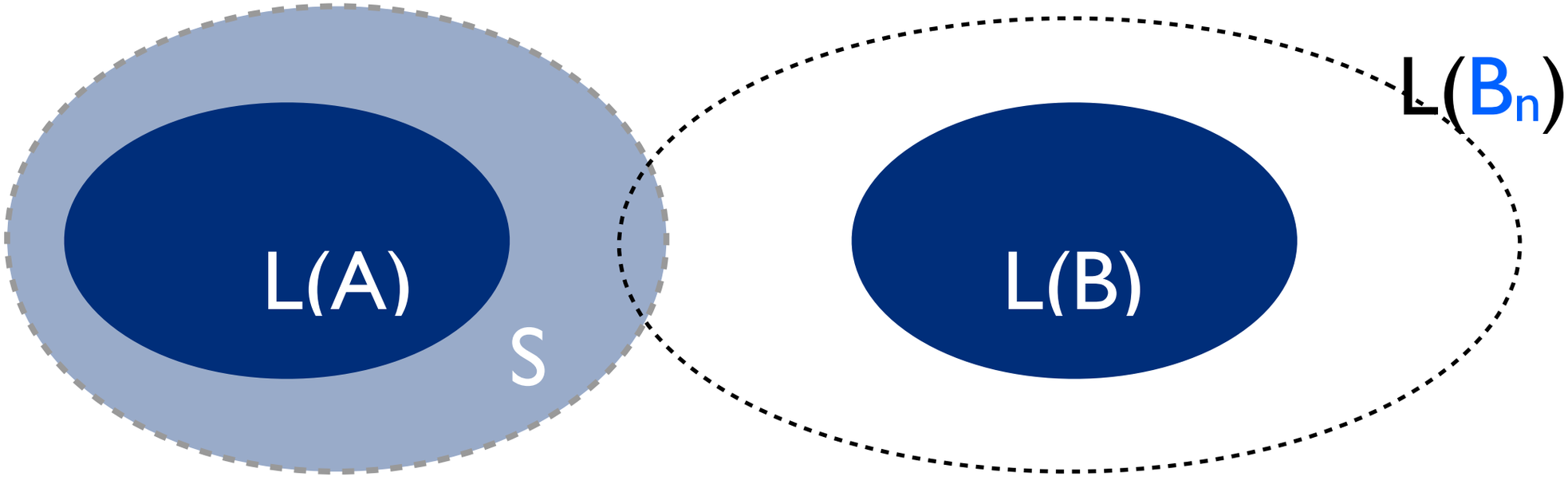
Proof idea



For big n every run of B_n labelled by word in S can be modified to a run of B still labelled by word in S

L(B) is disjoint from S, so $L(B_n)$ as well

Proof idea



For big n every run of B_n labelled by word in S can be modified to a run of B still labelled by word in S

L(B) is disjoint from S, so $L(B_n)$ as well

Similarly $L(A_n)$ is disjoint from compl(S)

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

Thank you!