

Unambiguity Makes Context Free Grammars Big

Stefan Mengel

Univ. Artois, CNRS, CRIL,
Lens

Harry Vinall-Smeeth

TU Ilmenau

FMT Workshop 2025,
Les Houches

Context Free Grammars (CFG)

G

$$S \rightarrow Aa \mid aB$$

Context Free Grammars (CFG)

G

$$\begin{array}{l} S \rightarrow Aa \mid aB \\ A \rightarrow a \mid c \end{array}$$

Context Free Grammars (CFG)

G

$$S \rightarrow Aa \mid aB$$

$$A \rightarrow a \mid c$$

$$B \rightarrow a \mid b$$

Context Free Grammars (CFG)

G

$$S \rightarrow Aa \mid aB$$

$$A \rightarrow a \mid c$$

$$B \rightarrow a \mid b$$

Derivations

Context Free Grammars (CFG)

G

$$S \rightarrow Aa \mid aB$$

$$A \rightarrow a \mid c$$

$$B \rightarrow a \mid b$$

Derivations

$$S \rightarrow Aa$$

Context Free Grammars (CFG)

G

$$S \rightarrow Aa \mid aB$$

$$A \rightarrow a \mid c$$

$$B \rightarrow a \mid b$$

Derivations

$$S \rightarrow Aa \rightarrow aa$$

Context Free Grammars (CFG)

G

$$S \rightarrow Aa \mid aB$$

$$A \rightarrow a \mid c$$

$$B \rightarrow a \mid b$$

Derivations

$$S \rightarrow Aa \rightarrow aa$$

$$S \rightarrow Aa \rightarrow ca$$

$$S \rightarrow aB \rightarrow aa$$

$$S \rightarrow aB \rightarrow ab$$

Context Free Grammars (CFG)

G

$$S \rightarrow Aa \mid aB$$

$$A \rightarrow a \mid c$$

$$B \rightarrow a \mid b$$

$$L(G) = \{aa, ab, ca\}$$

Derivations

$$S \rightarrow Aa \rightarrow aa$$

$$S \rightarrow Aa \rightarrow ca$$

$$S \rightarrow aB \rightarrow aa$$

$$S \rightarrow aB \rightarrow ab$$

Context Free Grammars (CFG)

G

$$S \rightarrow Aa \mid aB$$

$$A \rightarrow a \mid c$$

$$B \rightarrow a \mid b$$

$$L(G) = \{aa, ab, ca\}$$

Derivations

$$S \rightarrow Aa \rightarrow aa$$

$$S \rightarrow Aa \rightarrow ca$$

$$S \rightarrow aB \rightarrow aa$$

$$S \rightarrow aB \rightarrow ab$$

$$|G| = 8 = |Aa| + |aB| + |a| + |c| + |a| + |b|$$

CFG for Finite Languages

G

$$S \rightarrow Aa \mid aB$$

$$A \rightarrow a \mid c$$

$$B \rightarrow a \mid b$$

$$L(G) = \{aa, ab, ca\}$$

Derivations

$$S \rightarrow Aa \rightarrow aa$$

$$S \rightarrow Aa \rightarrow ca$$

$$S \rightarrow aB \rightarrow aa$$

$$S \rightarrow aB \rightarrow ab$$

$$|G| = 8 = |Aa| + |aB| + |a| + |c| + |a| + |b|$$

(Unambiguous) CFG for Finite Languages

G

$$S \rightarrow Aa \mid aB$$

$$A \rightarrow a \mid c$$

$$B \rightarrow a \mid b$$

$$L(G) = \{aa, ab, ca\}$$

Derivations

$$S \rightarrow Aa \rightarrow aa$$

$$S \rightarrow Aa \rightarrow ca$$

$$S \rightarrow aB \rightarrow aa$$

$$S \rightarrow aB \rightarrow ab$$

$$|G| = 8 = |Aa| + |aB| + |a| + |c| + |a| + |b|$$

(Unambiguous) CFG for Finite Languages

G

$$S \rightarrow Aa \mid aB$$

$$A \rightarrow a \mid c$$

$$B \rightarrow a \mid b$$

$$L(G) = \{aa, ab, ca\}$$

Derivations

$$S \rightarrow Aa \rightarrow \textcircled{aa}$$

$$S \rightarrow Aa \rightarrow ca$$

$$S \rightarrow aB \rightarrow aa$$

$$S \rightarrow aB \rightarrow ab$$

$$|G| = 8 = |Aa| + |aB| + |a| + |c| + |a| + |b|$$

(Unambiguous) CFG for Finite Languages

G

$$S \rightarrow Aa \mid aB$$

$$A \rightarrow a \mid c$$

$$B \rightarrow a \mid b$$

$$L(G) = \{aa, ab, ca\}$$

Derivations

$$S \rightarrow Aa \rightarrow aa$$

$$S \rightarrow Aa \rightarrow ca$$

$$S \rightarrow aB \rightarrow aa$$

$$S \rightarrow aB \rightarrow ab$$

$$|G| = 8 = |Aa| + |aB| + |a| + |c| + |a| + |b|$$

(Unambiguous) CFG for Finite Languages

G

$$S \rightarrow Aa \mid aB$$

$$A \rightarrow a \mid c$$

$$B \rightarrow a \mid b$$

$$L(G) = \{aa, ab, ca\}$$

Derivations

$$S \rightarrow Aa \rightarrow aa$$

$$S \rightarrow Aa \rightarrow ca$$

$$S \rightarrow aB \rightarrow aa$$

$$S \rightarrow aB \rightarrow ab$$

$$|G| = 8 = |Aa| + |aB| + |a| + |c| + |a| + |b|$$

G is ambiguous:
two derivations of the same word!

(Unambiguous) CFG for Finite Languages

G'

$$S \rightarrow Aa \mid aB$$

$$A \rightarrow a \mid c$$

$$B \rightarrow b$$

$$L(G') = \{aa, ab, ca\}$$

Derivations

$$S \rightarrow Aa \rightarrow aa$$

$$S \rightarrow Aa \rightarrow ca$$

$$S \rightarrow aB \rightarrow ab$$

$$|G'| = 7 = |Aa| + |aB| + |a| + |c| + |b|$$

G' is unambiguous:
every word has a unique derivation.

Motivation

CFG as Compression

Motivation

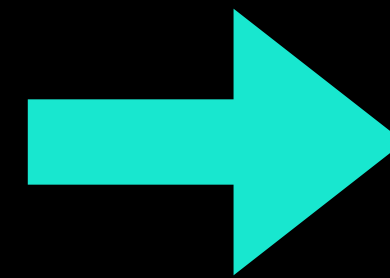
CFG as Compression



Large Object

Motivation

CFG as Compression



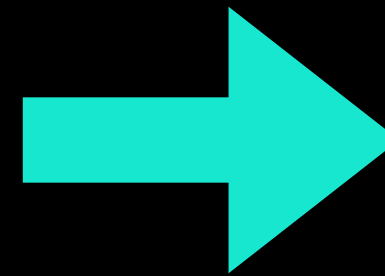
Large Object

Motivation

CFG as Compression



Large Object



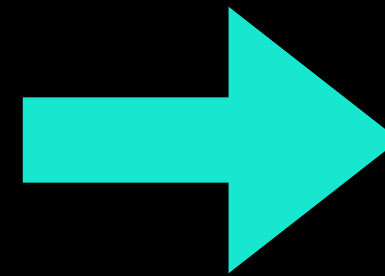
Smaller Object

Motivation

CFG as Compression



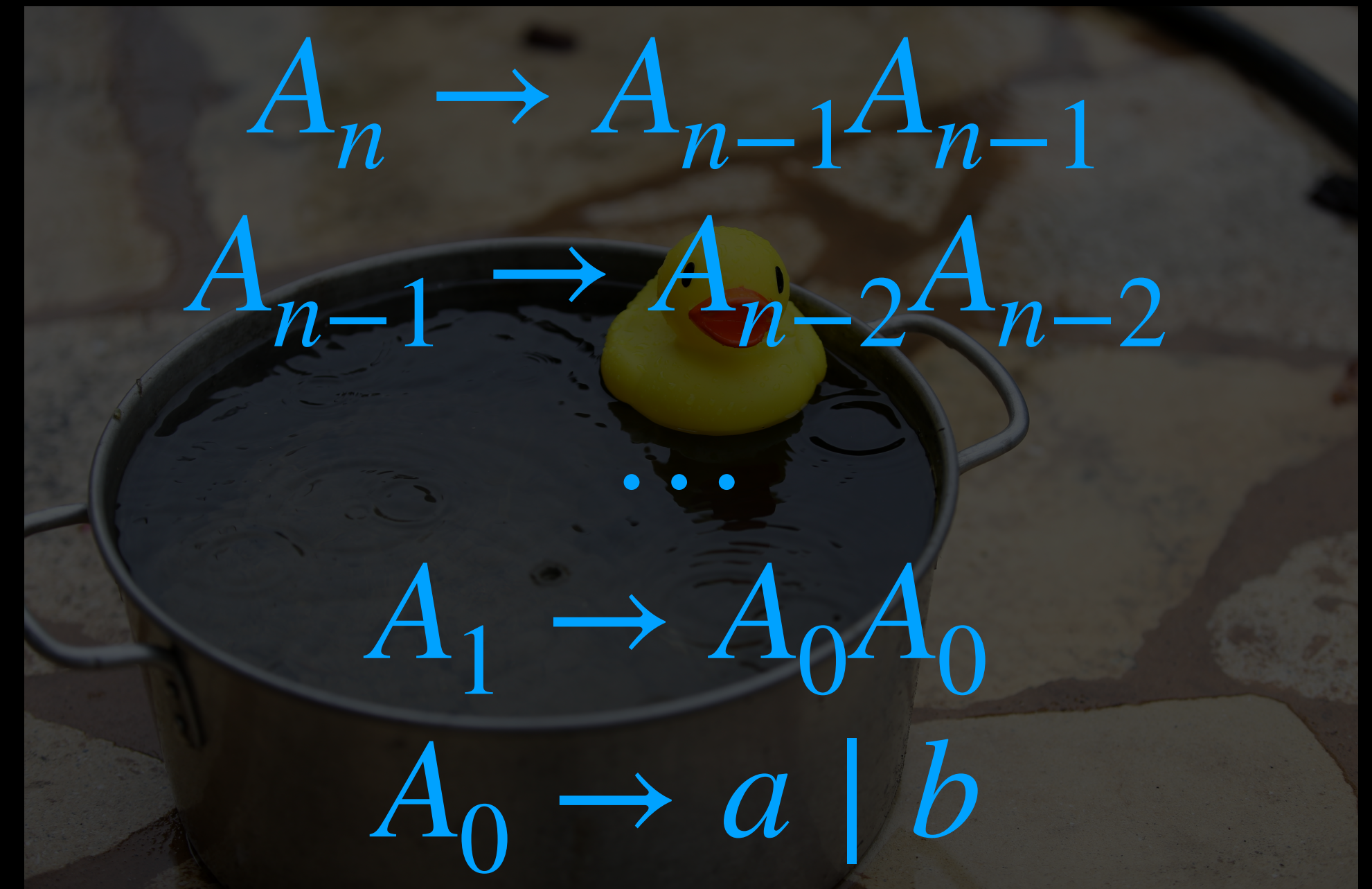
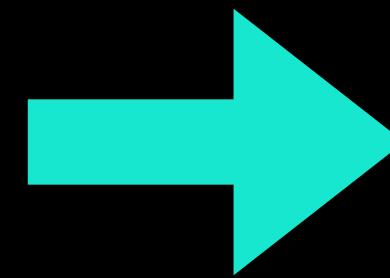
Large Object
(Encoded as a set of strings)



Smaller Object

Motivation

CFG as Compression



Large Object
(Encoded as a set of strings)

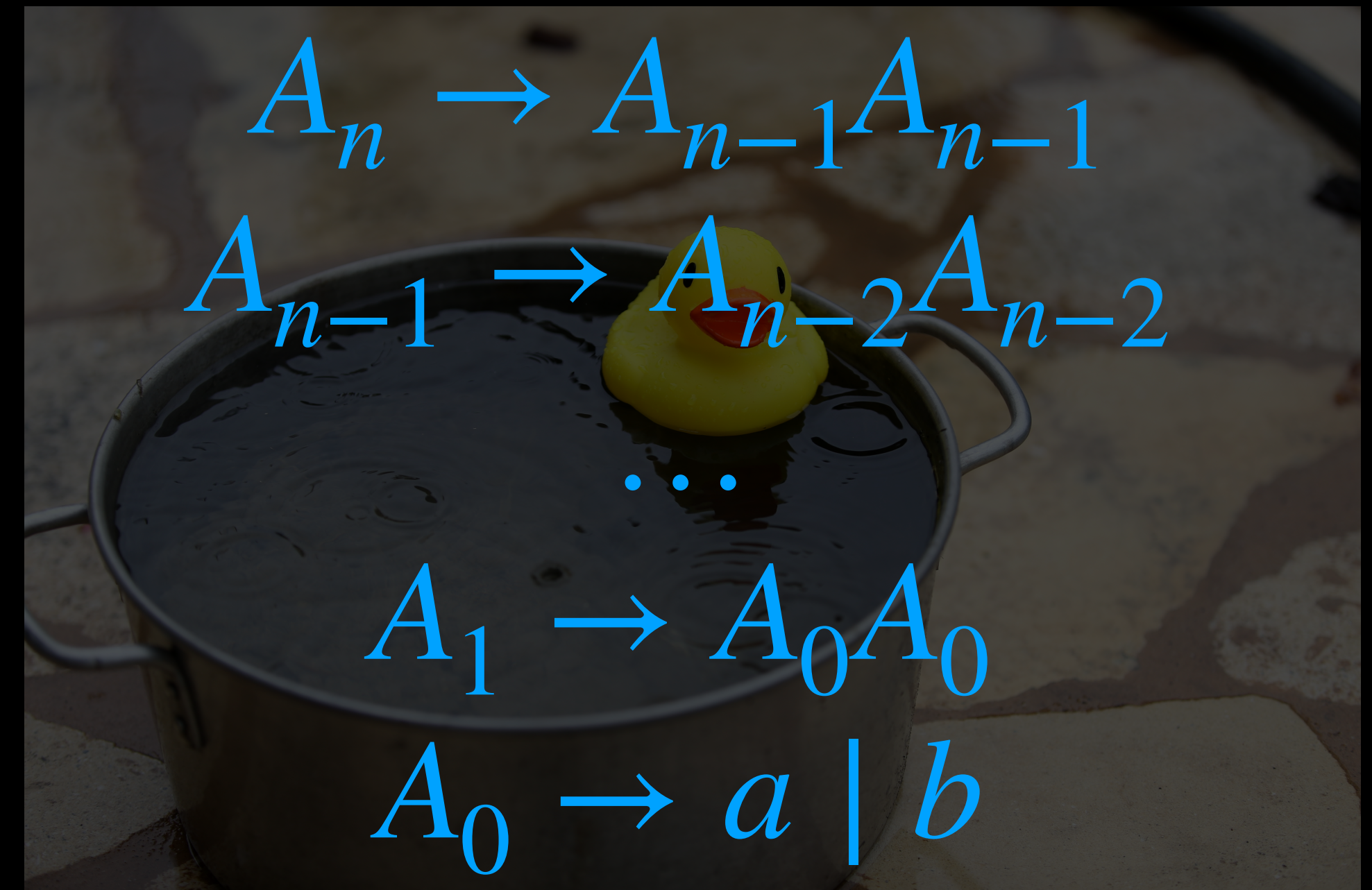
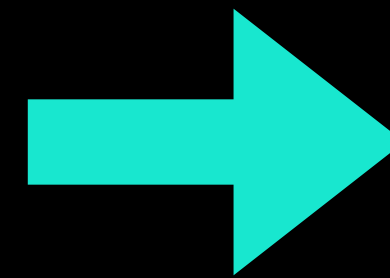
CFG

Motivation

CFG as Compression



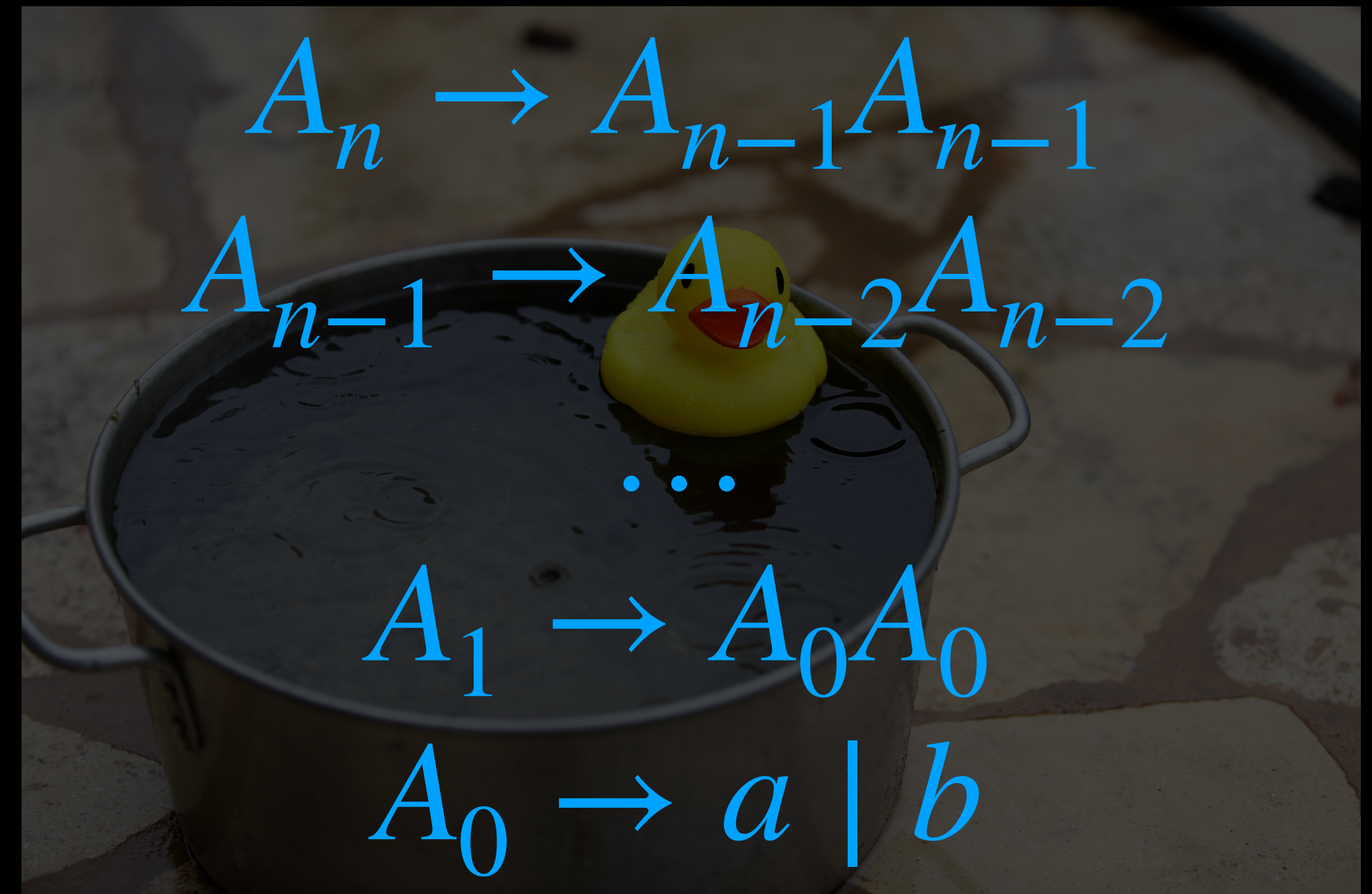
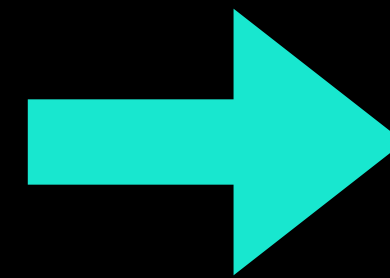
Large Object
(Encoded as a set of strings)



(unambiguous) CFG

Motivation

CFG as Compression



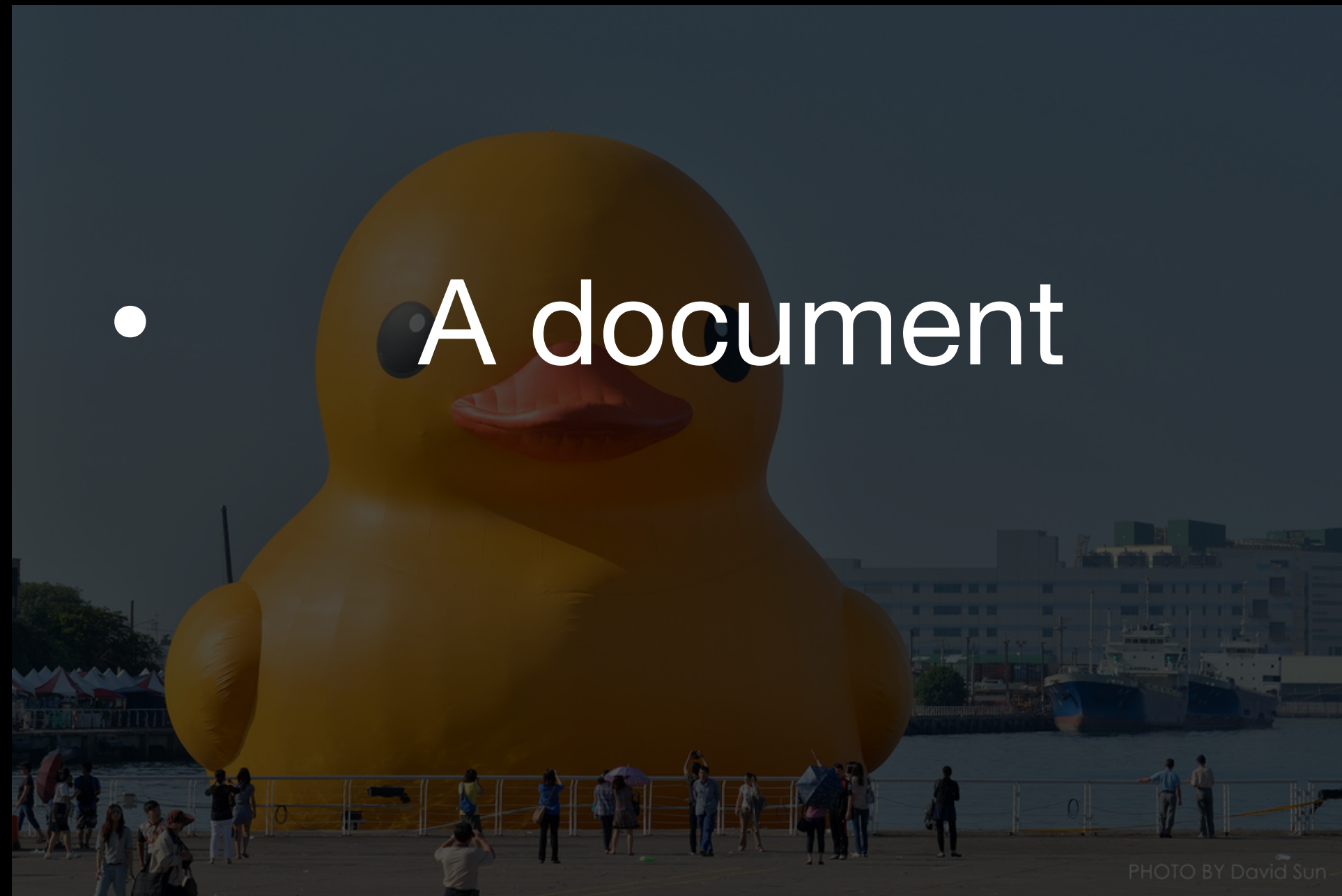
Large Object
(Encoded as a set of strings)

(unambiguous) CFG

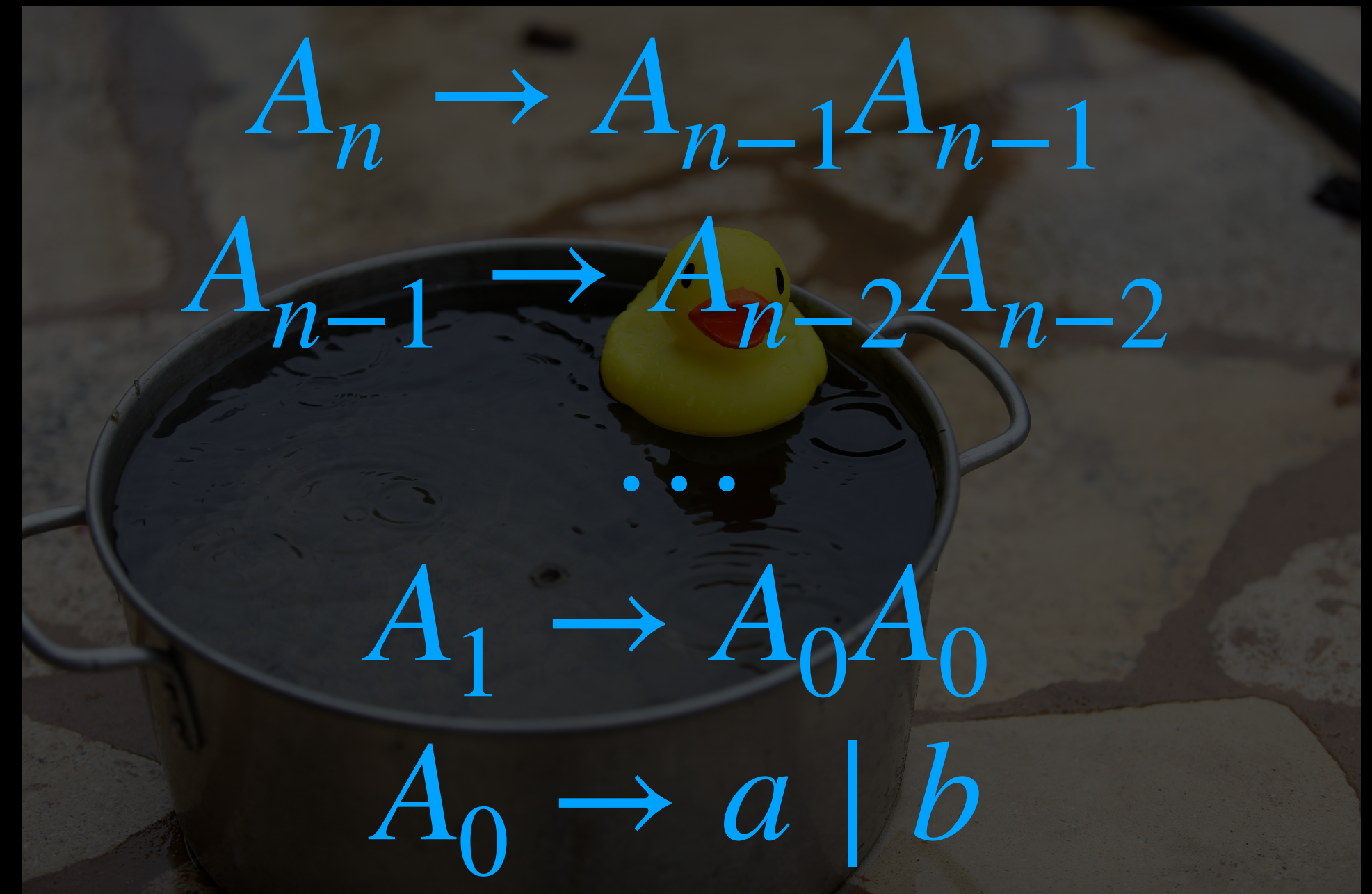
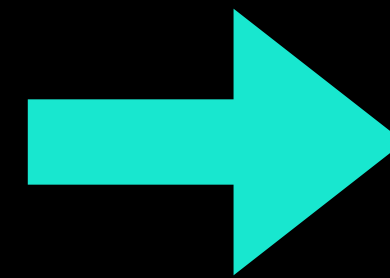
Motivation

CFG as Compression

- A document



Large Object
(Encoded as a set of strings)

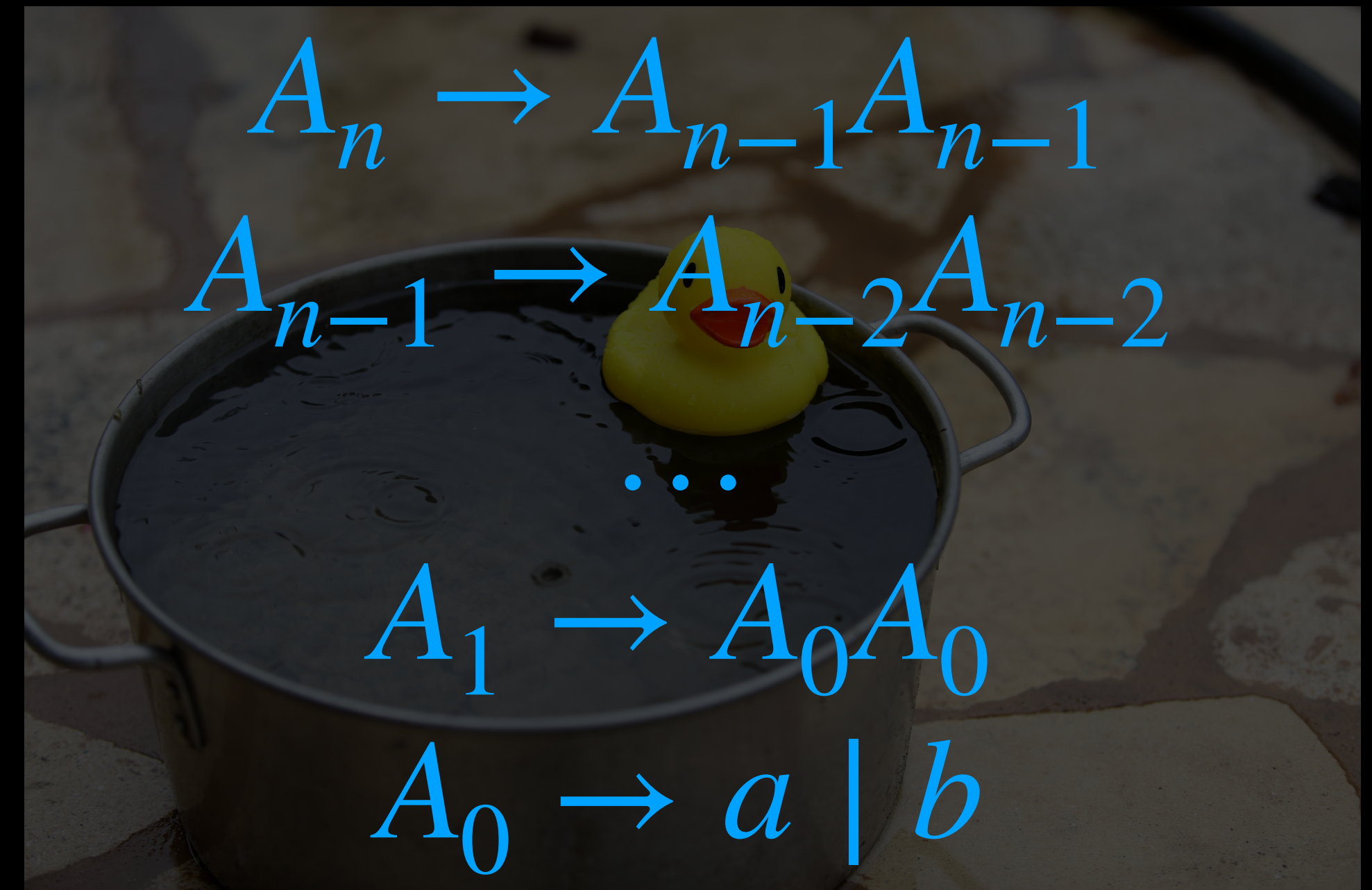
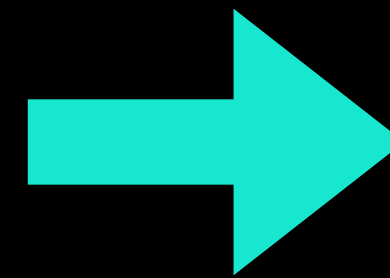


(unambiguous) CFG

Motivation

CFG as Compression

- A document
- A set of documents



Large Object
(Encoded as a set of strings)

(unambiguous) CFG

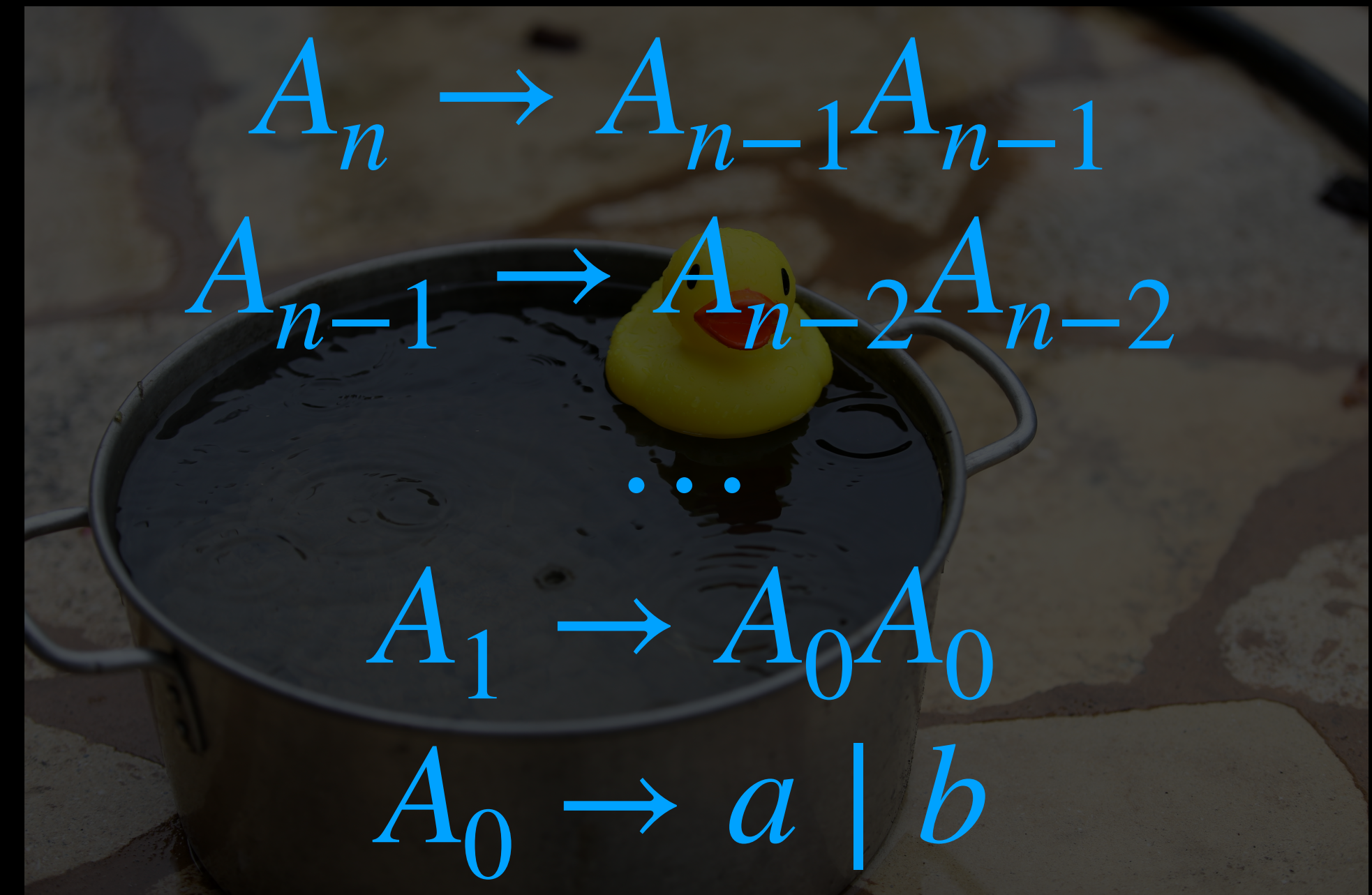
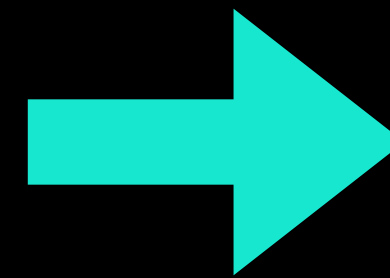
Motivation

CFG as Compression

- A document
- A set of documents
- Query answer

Large Object

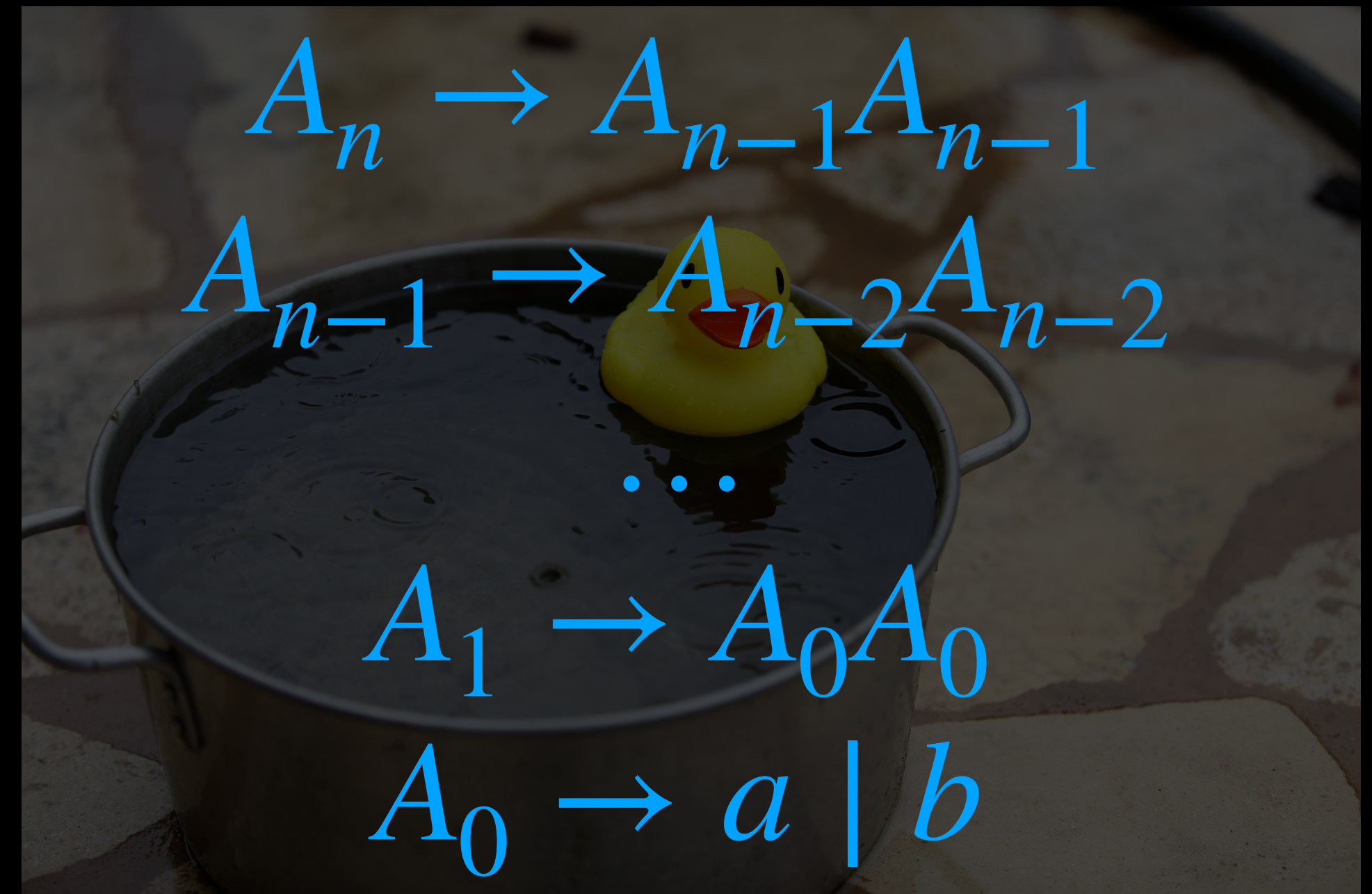
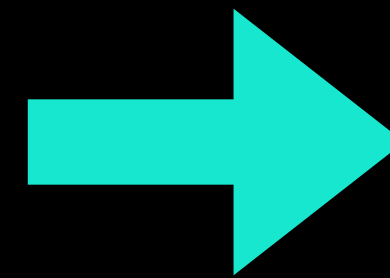
(Encoded as a set of strings)



(unambiguous) CFG

Motivation

CFGs to Represent Query Answers

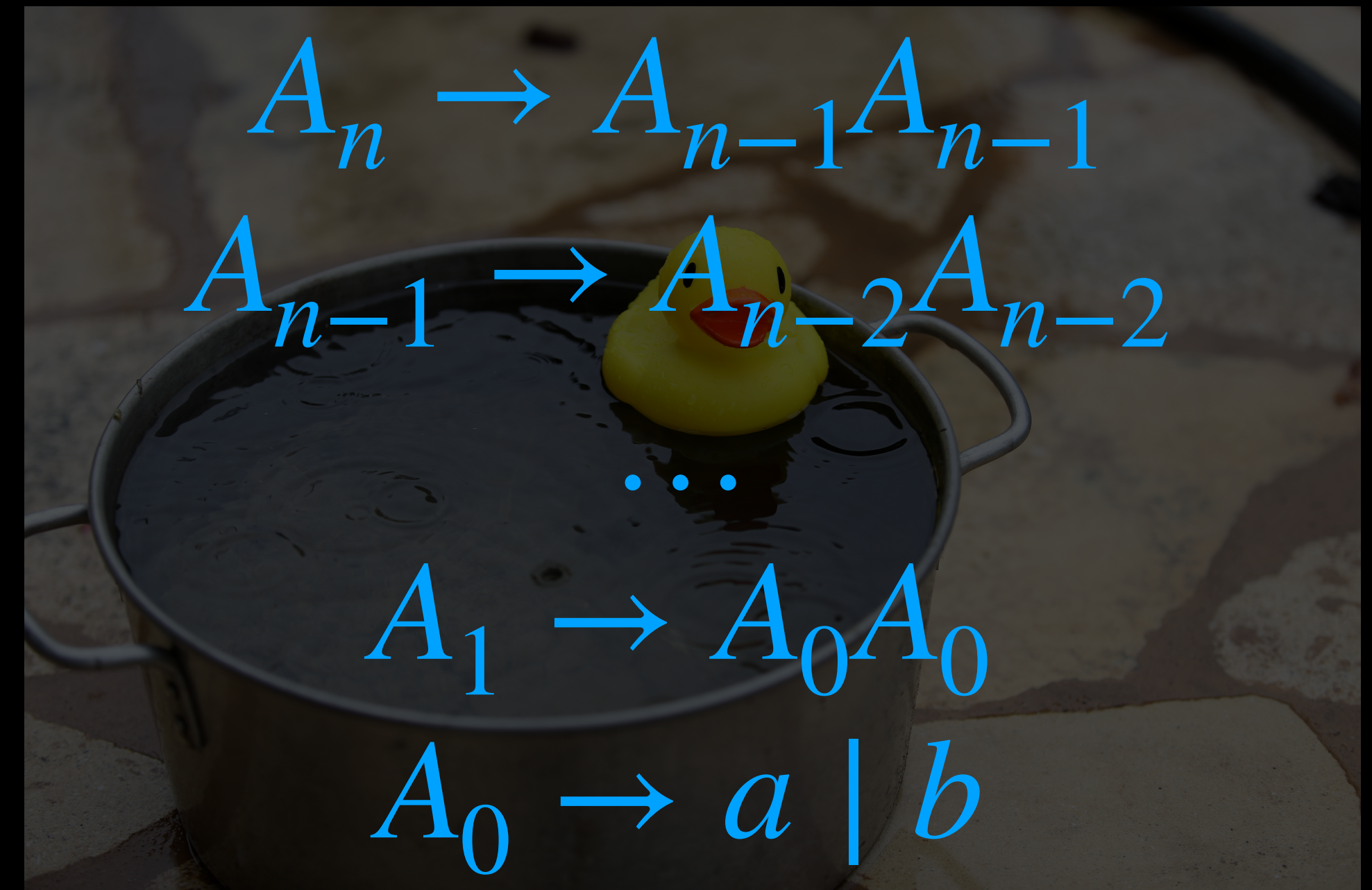
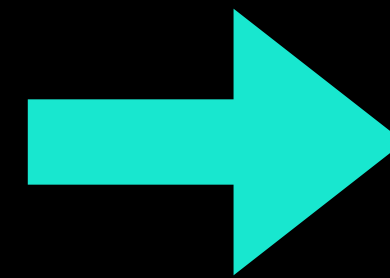
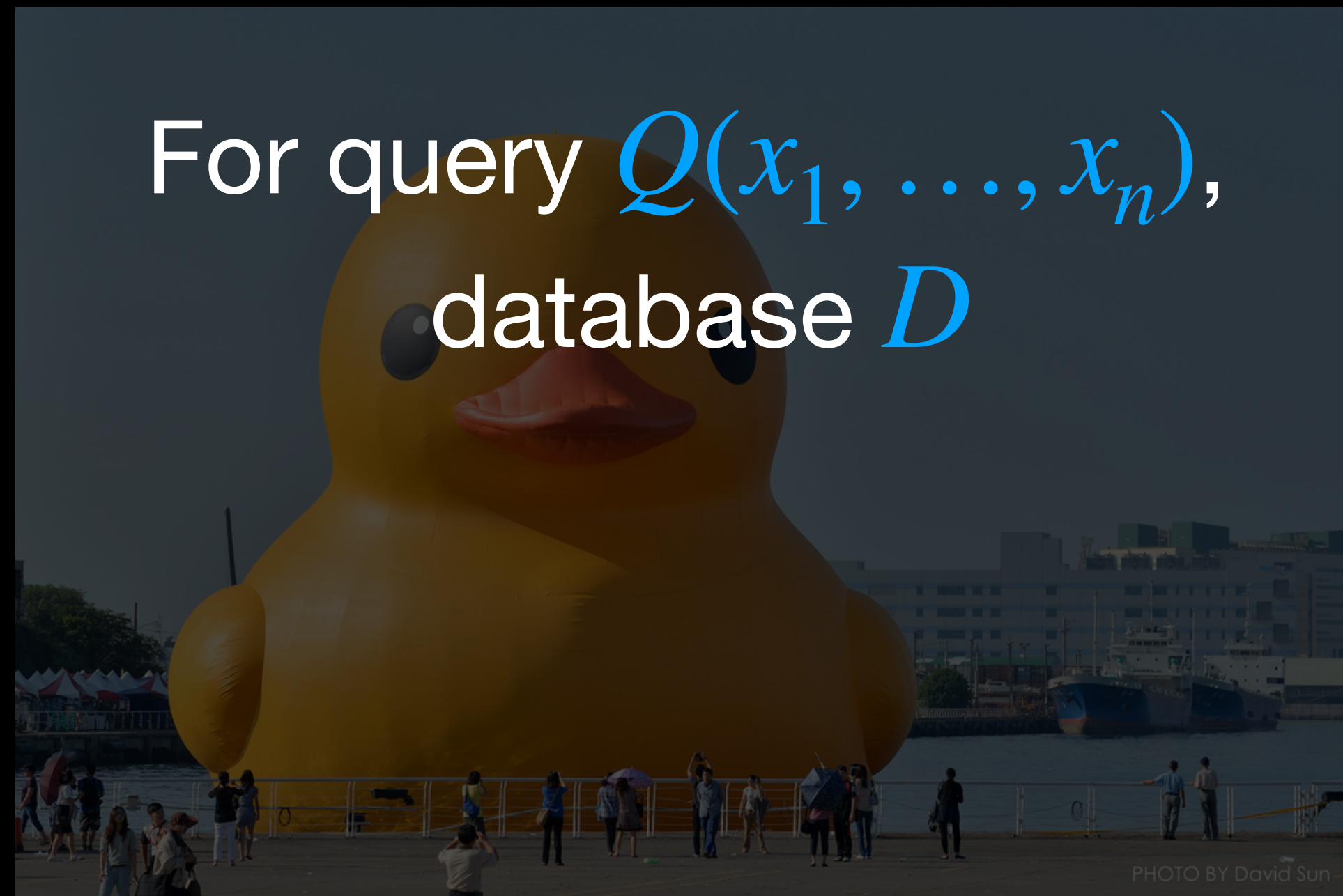


Large Object = $Q(D)$
(Encoded as a set of strings)

(unambiguous) CFG

Motivation

CFGs to Represent Query Answers



Large Object = $Q(D)$
(Encoded as a set of strings)

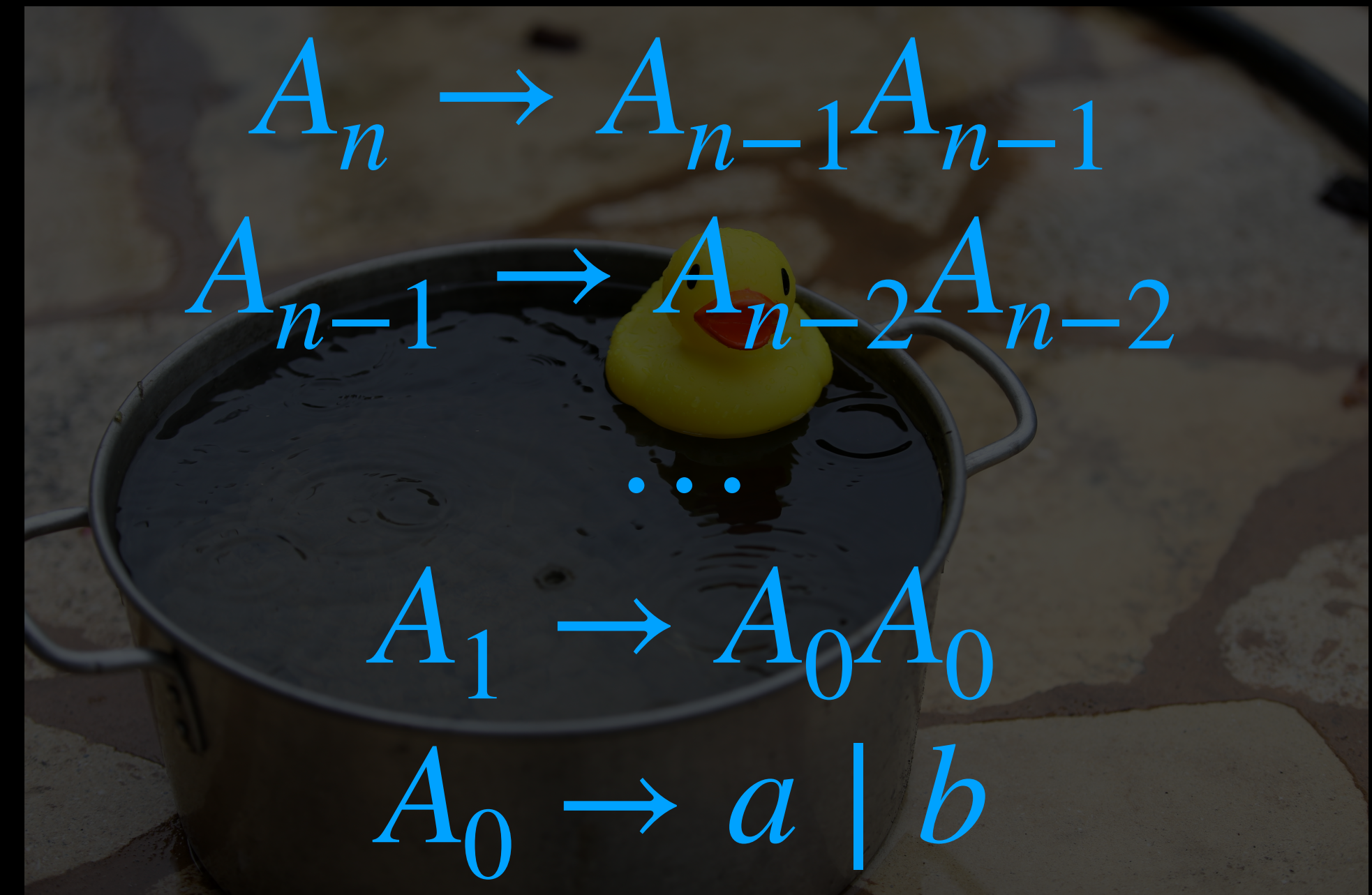
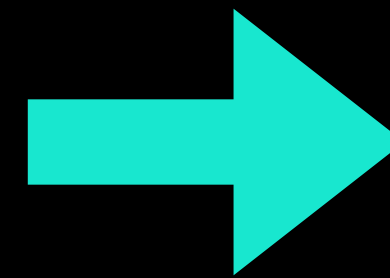
(unambiguous) CFG

Motivation

CFGs to Represent Query Answers

For query $Q(x_1, \dots, x_n)$,
database D
identify $(a_1, \dots, a_n) \in Q(D)$
with word $a_1 a_2 \dots a_n$

Large Object = $Q(D)$
(Encoded as a set of strings)



(unambiguous) CFG

Motivation

CFGs to Represent Query Answers

- This idea recently proposed by Kimelfeld, Martens and Niewerth.

Kimelfeld, Martens and Niewerth: *A Formal Language Perspective on Factorized Representations*. ICDT (2025)

Motivation

CFGs to Represent Query Answers

- This idea recently proposed by Kimelfeld, Martens and Niewerth.
- For bounded treewidth CQs can compute ‘small’ CFG representations directly from the query and the database.

Olteanu and Závodný: [Size Bounds for Factorised Representations of Query Results.](#)
ACM Trans. Database Syst. (2015)

Motivation

CFGs to Represent Query Answers

- This idea recently proposed by Kimelfeld, Martens and Niewerth.
- For bounded treewidth CQs can compute ‘small’ CFG representations directly from the query and the database.
- Then can perform algorithms directly on CFG, e.g. recent result shows you can do approximate counting!

Meel and de Colnet: [#CFG and #DNNF admit FPRAS](#). Preprint (2024)

Motivation

uCFGs to Represent Query Answers

- This idea recently proposed by Kimelfeld, Martens and Niewerth.
- For bounded treewidth CQs can compute ‘small’ uCFG representations directly from the query and the database.

Olteanu and Závodný: [Size Bounds for Factorised Representations of Query Results.](#)
ACM Trans. Database Syst. (2015)

Motivation

uCFGs to Represent Query Answers

- This idea recently proposed by Kimelfeld, Martens and Niewerth.
- For bounded treewidth CQs can compute ‘small’ uCFG representations directly from the query and the database.
- More algorithmically powerful! Now can do exact counting as well as more efficient enumeration.

Olteanu and Závodný: [Size Bounds for Factorised Representations of Query Results.](#)
ACM Trans. Database Syst. (2015)

Motivation

uCFGs to Represent Query Answers

- This idea recently proposed by Kimelfeld, Martens and Niewerth.
- For bounded treewidth, unambiguous representations directly from the query.
- More algorithmically powerful! Now can do exact counting as well as more efficient enumeration.

**But are there situations
where unambiguity
comes at a price?**

Olteanu and Závodný: *Size Bounds for Factorised Representations of Query Results.*
ACM Trans. Database Syst. (2015)

Our Contribution

Background

Our Contribution

Background

Let $L_n \subseteq \{a, b\}^{2n}$ contain all words containing two a symbols at distance n from each other.

Our Contribution

Background

Let $L_n \subseteq \{a, b\}^{2n}$ contain all words containing two a symbols at distance n from each other.

$$aaaa \in L_2$$

Our Contribution

Background

Let $L_n \subseteq \{a, b\}^{2n}$ contain all words containing two a symbols at distance n from each other.

$$aaaa \in L_2$$

$$abab \in L_2$$

Our Contribution

Background

Let $L_n \subseteq \{a, b\}^{2n}$ contain all words containing two a symbols at distance n from each other.

$$aaaa \in L_2$$

$$abab \in L_2$$

$$abba \notin L_2$$

Our Contribution

Background

Let $L_n \subseteq \{a, b\}^{2n}$ contain all words containing two a symbols at distance n from each other.

Then L_n admits a CFG of size $\Theta(\log(n))$.

Our Contribution

Background

Let $L_n \subseteq \{a, b\}^{2n}$ contain all words containing two a symbols at distance n from each other.

Then L_n admits a CFG of size $\Theta(\log(n))$.

Kimelfeld, Martens and Niewerth: *A Formal Language Perspective on Factorized Representations*. ICDT (2025)

Our Contribution

Background

Let $L_n \subseteq \{a, b\}^{2n}$ contain all words containing two a symbols at distance n from each other.

Then L_n admits a CFG of size $\Theta(\log(n))$.

Best possible compression.

Kimelfeld, Martens and Niewerth: *A Formal Language Perspective on Factorized Representations*. ICDT (2025)

Our Contribution

Background

Let $L_n \subseteq \{a, b\}^{2n}$ contain all words containing two a symbols at distance n from each other.

Conjecture: Every uCFG for L_n has size $2^{\Omega(n)}$.

Kimelfeld, Martens and Niewerth: *A Formal Language Perspective on Factorized Representations*. ICDT (2025)

Our Contribution

Let $L_n \subseteq \{a, b\}^{2n}$ contain all words containing two a symbols at distance n from each other.

Theorem: Every uCFG for L_n has size $2^{\Omega(n)}$.

Mengel, V-S: A Lower Bound on Unambiguous Context Free Grammars via Communication Complexity. PODS (2025)

Our Contribution

Theorem: Every uCFG for L_n has size $2^{\Omega(n)}$.

Mengel, V-S: *A Lower Bound on Unambiguous Context Free Grammars via Communication Complexity*. PODS (2025)

Our Contribution

Theorem: Every uCFG for L_n has size $2^{\Omega(n)}$.

\Rightarrow double exponential succinctness
separation CFG vs uCFG for finite languages.

Mengel, V-S: *A Lower Bound on Unambiguous Context Free Grammars via Communication Complexity*. PODS (2025)

Our Contribution

Theorem: Every uCFG for L_n has size $2^{\Omega(n)}$.

\Rightarrow double exponential succinctness separation CFG vs uCFG for finite languages.

\Rightarrow exponential succinctness separation NFA vs uCFG for finite languages.

Mengel, V-S: A Lower Bound on Unambiguous Context Free Grammars via Communication Complexity. PODS (2025)

Proof Sketch



Proof Sketch

From (u)CFG to Rectangle Covers

Let $L \subseteq \Sigma^n$, G be a CFG accepting L .

Proof Sketch

From (u)CFG to Rectangle Covers

Let $L \subseteq \Sigma^n$, G be a CFG accepting L .

Lemma:

$L = \bigcup_{i=1}^{\ell} R_i$, where each R_i is a
rectangle and $\ell \leq n |G|$.

Proof Sketch

From (u)CFG to Rectangle Covers

Let $L \subseteq \Sigma^n$, G be a CFG accepting L .

Lemma:

$L = \bigcup_{i=1}^{\ell} R_i$, where each R_i is a
rectangle and $\ell \leq n |G|$.

Rectangle:

Proof Sketch

From (u)CFG to Rectangle Covers

Let $L \subseteq \Sigma^n$, G be a CFG accepting L .

Lemma:

$L = \bigcup_{i=1}^{\ell} R_i$, where each R_i is a
rectangle and $\ell \leq n |G|$.

Rectangle:

$$R = \{xy \mid x \in L_1, y \in L_2\}$$

Proof Sketch

From (u)CFG to Rectangle Covers

Let $L \subseteq \Sigma^n$, G be a CFG accepting L .

Lemma:

$L = \bigcup_{i=1}^{\ell} R_i$, where each R_i is a
rectangle and $\ell \leq n |G|$.

Rectangle:

$$R = \{xy \mid x \in L_1, y \in L_2\}$$
$$L_1 \subseteq \Sigma^{n/2}$$

Proof Sketch

From (u)CFG to Rectangle Covers

Let $L \subseteq \Sigma^n$, G be a CFG accepting L .

Lemma:

$L = \bigcup_{i=1}^{\ell} R_i$, where each R_i is a
rectangle and $\ell \leq n |G|$.

Rectangle:

$R = \{xy \mid x \in L_1, y \in L_2\}$
 $L_1 \subseteq \Sigma^{n/2}$
 $L_2 \subseteq \Sigma^{n/2}$

Proof Sketch

From (u)CFG to Rectangle Covers

Let $L \subseteq \Sigma^n$, G be a CFG accepting L .

Lemma:

$L = \bigcup_{i=1}^{\ell} R_i$, where each R_i is a
rectangle and $\ell \leq n |G|$.

Rectangle:

$$R = \{xy \mid x \in L_1, y \in L_2\}$$
$$L_1 \subseteq \Sigma^{n/2}$$
$$L_2 \subseteq \Sigma^{n/2}$$

Proof Sketch

From (u)CFG to Rectangle Covers

Let $L \subseteq \Sigma^n$, G be a uCFG accepting L .

Lemma:

$L = \bigcup_{i=1}^{\ell} R_i$, where each R_i is a rectangle and $\ell \leq n |G|$.

Rectangle:

$$R = \{xy \mid x \in L_1, y \in L_2\}$$
$$L_1 \subseteq \Sigma^{n/2}$$
$$L_2 \subseteq \Sigma^{n/2}$$

Proof Sketch

From (u)CFG to Rectangle Covers

Let $L \subseteq \Sigma^n$, G be a uCFG accepting L .

Lemma:

$L = \bigcup_{i=1}^{\ell} R_i$, where each R_i is a rectangle and $\ell \leq n |G|$.

Idea:

Show every disjoint rectangle cover of L_n has size $2^{\Omega(n)}$.

Proof Sketch

From (u)CFG to P

Let $L \subseteq \Sigma^*$

Difficulty:
how to make use of
disjointness?

receiving L .

Lemma:

$L = \bigcup_{i=1}^{\ell} R_i$, where each R_i is a
rectangle and $\ell \leq n |G|$.

Idea:

Show every disjoint
rectangle cover of L_n
has size $2^{\Omega(n)}$.

Proof Sketch

Discrepancy Argument

All words over $\{a, b\}$
of length n

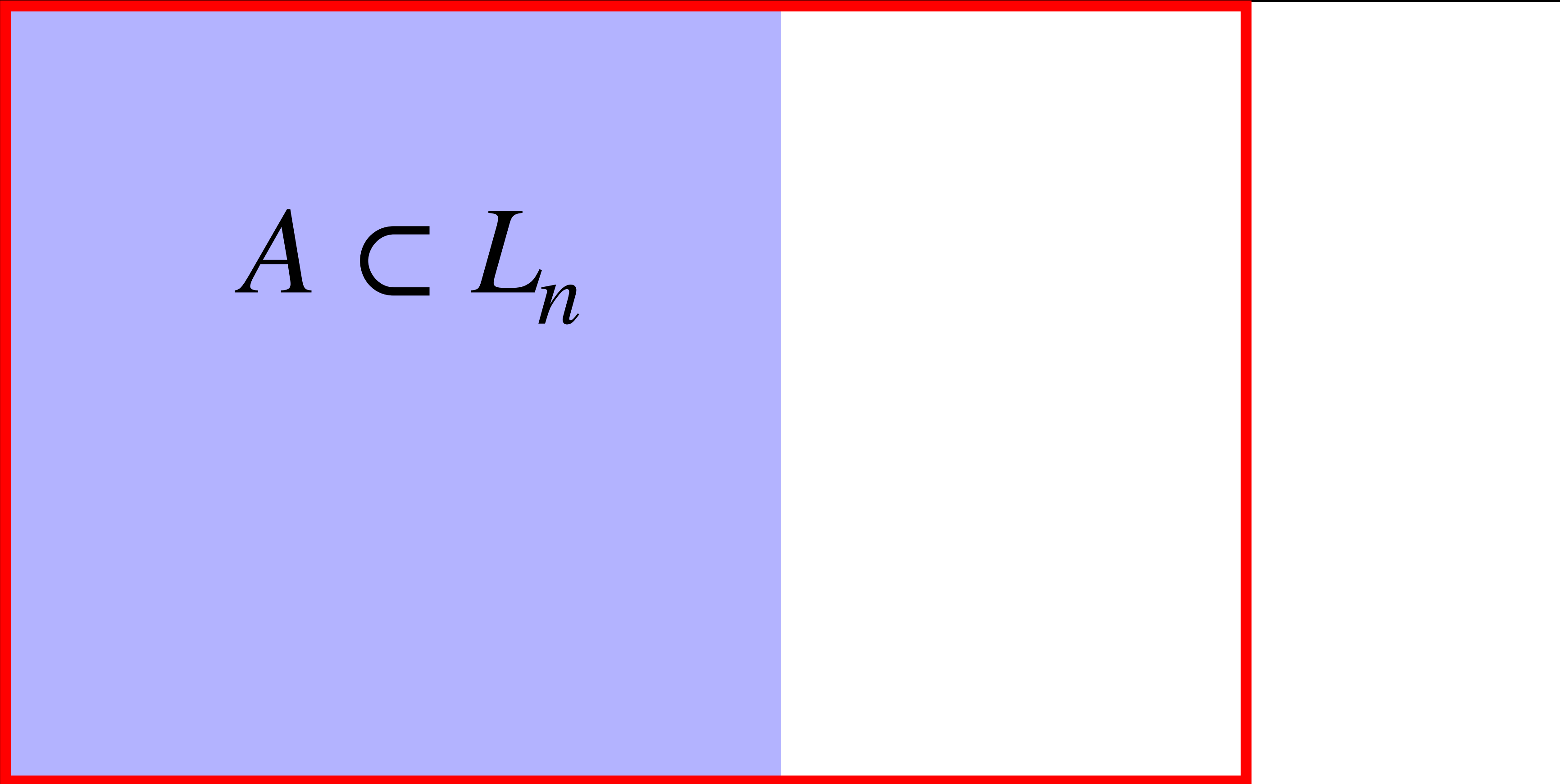
Proof Sketch

Discrepancy Argument

$$L_n$$

Proof Sketch

Discrepancy Argument


$$A \subset L_n$$

Proof Sketch

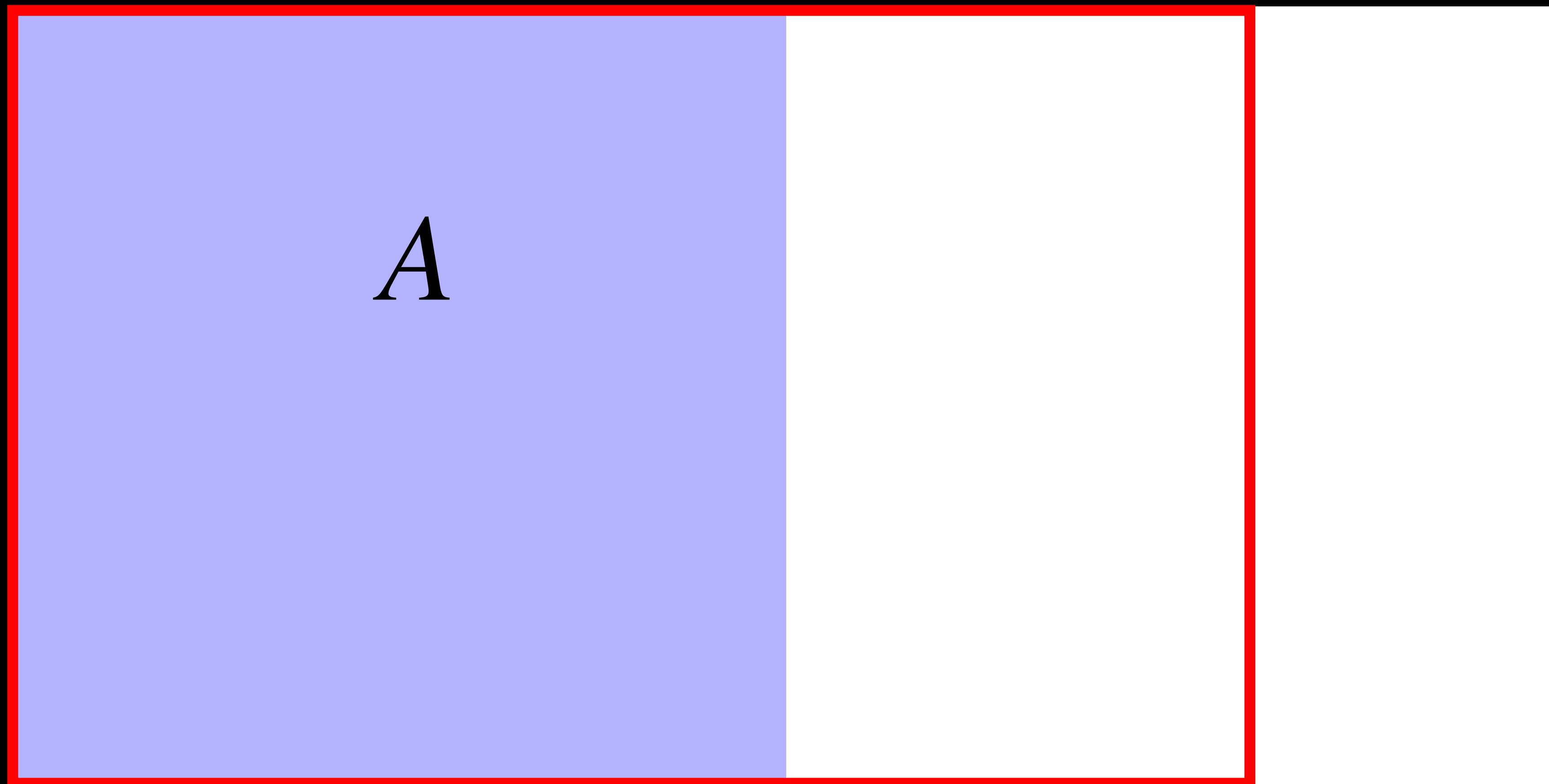
Discrepancy Argument

$$A \subset L_n$$

number of *as*
at distance *n*
is odd

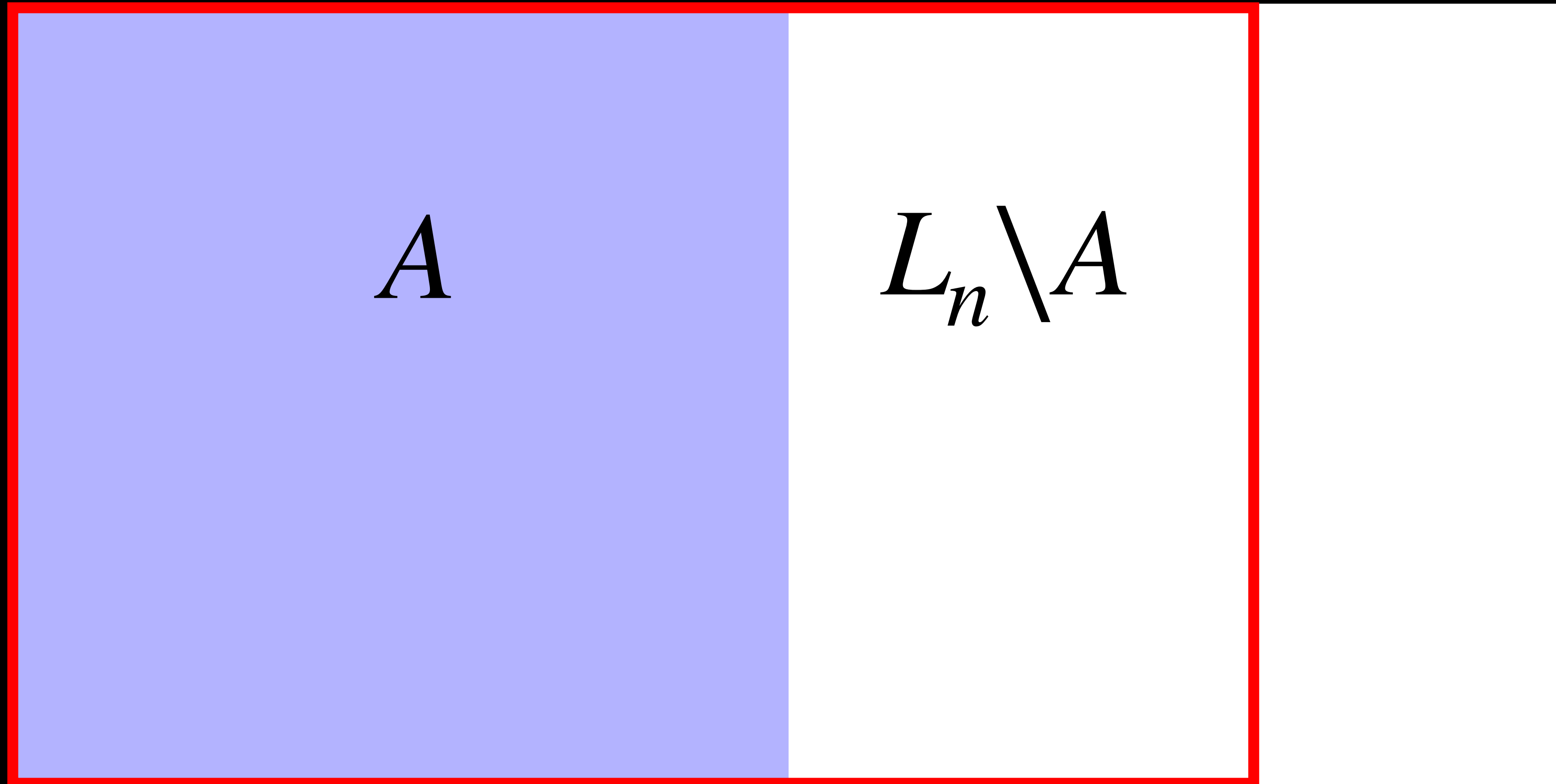
Proof Sketch

Discrepancy Argument



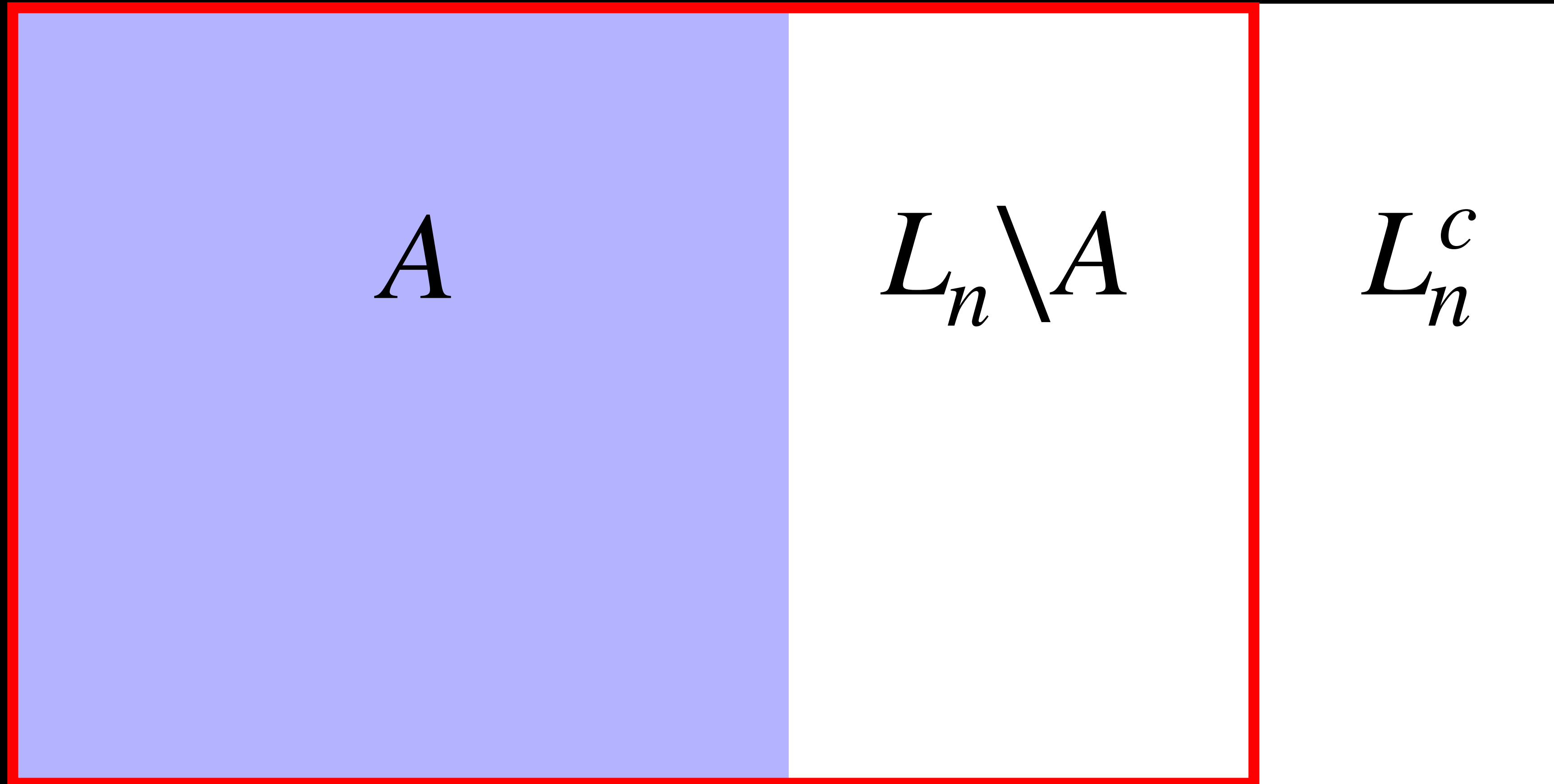
Proof Sketch

Discrepancy Argument



Proof Sketch

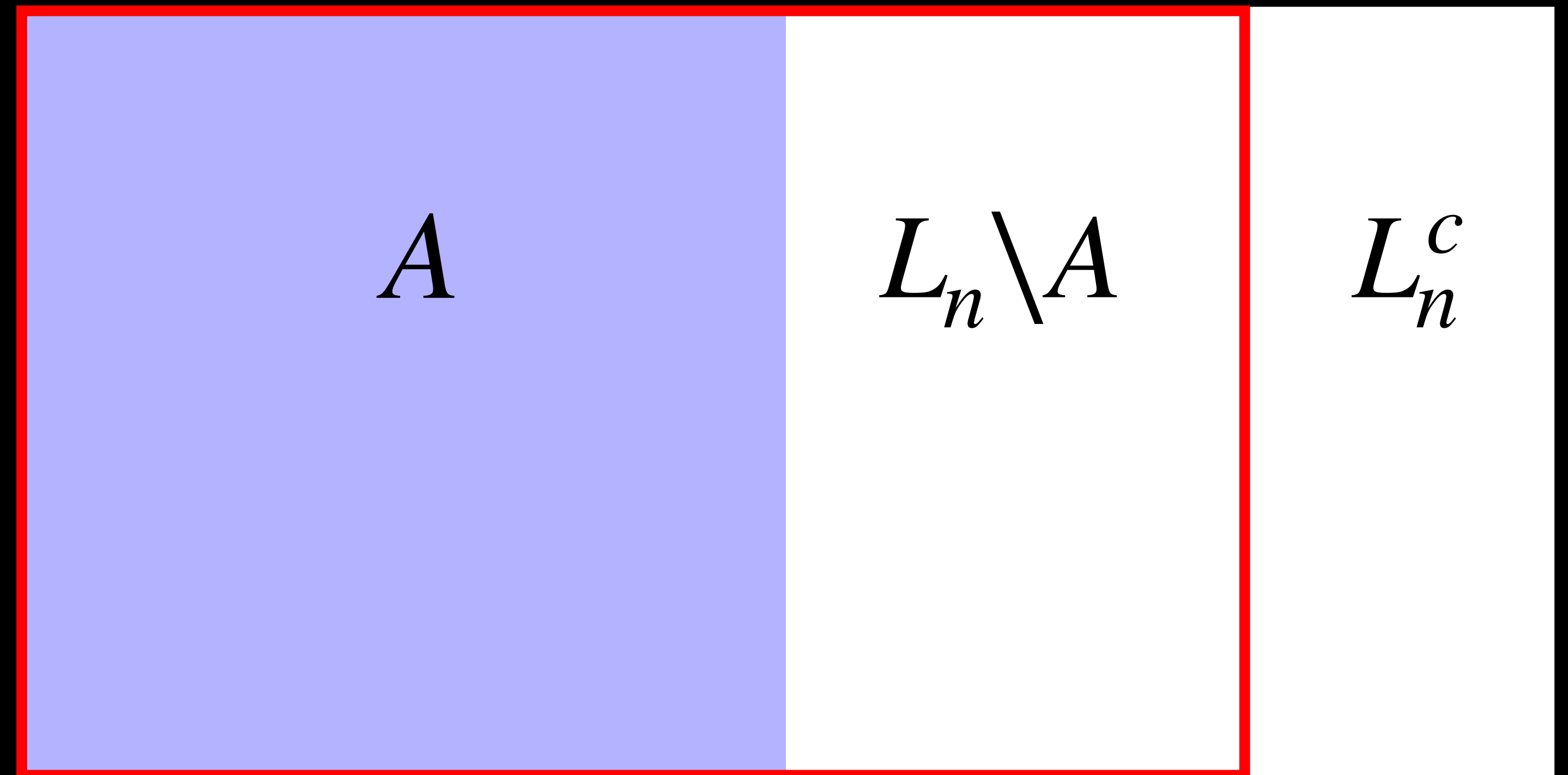
Discrepancy Argument



Proof Sketch

Discrepancy Argument

Facts:

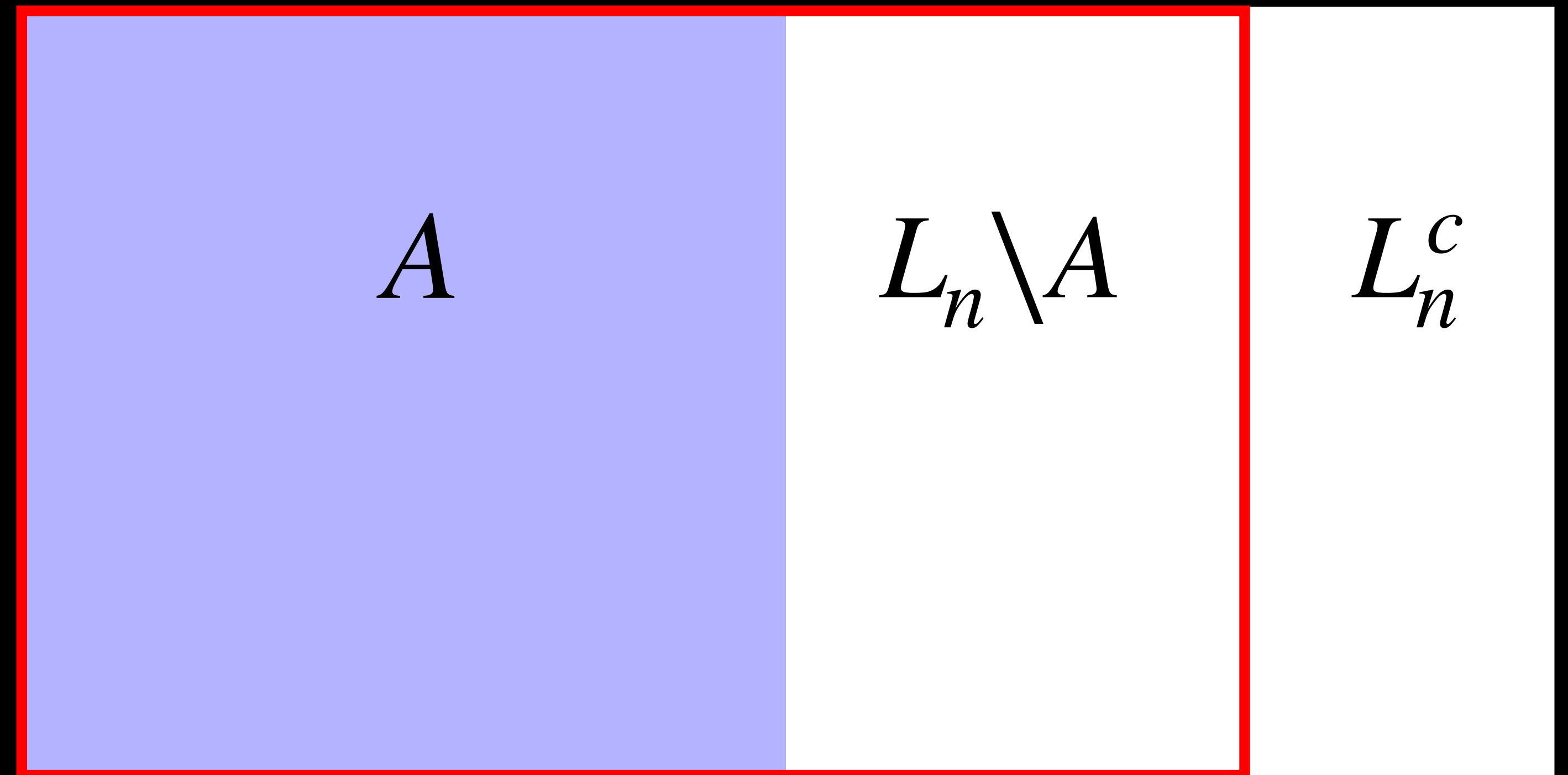


Proof Sketch

Discrepancy Argument

Facts:

(1) $|A| \gg |L_n \setminus A|$.



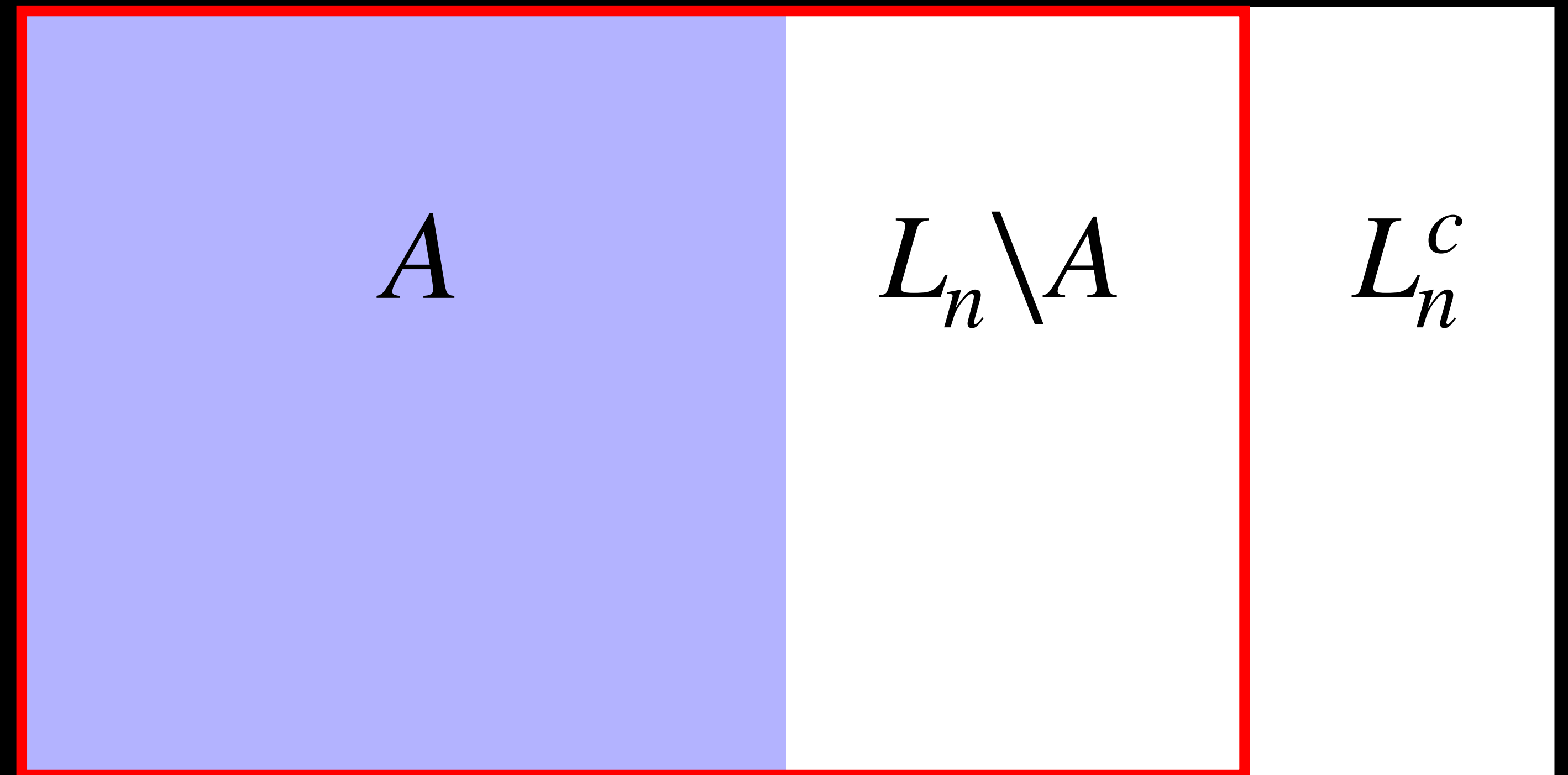
Proof Sketch

Discrepancy Argument

Facts:

$$(1) |A| \gg |L_n \setminus A|.$$

$$(2) |R \cap A| \approx |R \setminus A|$$



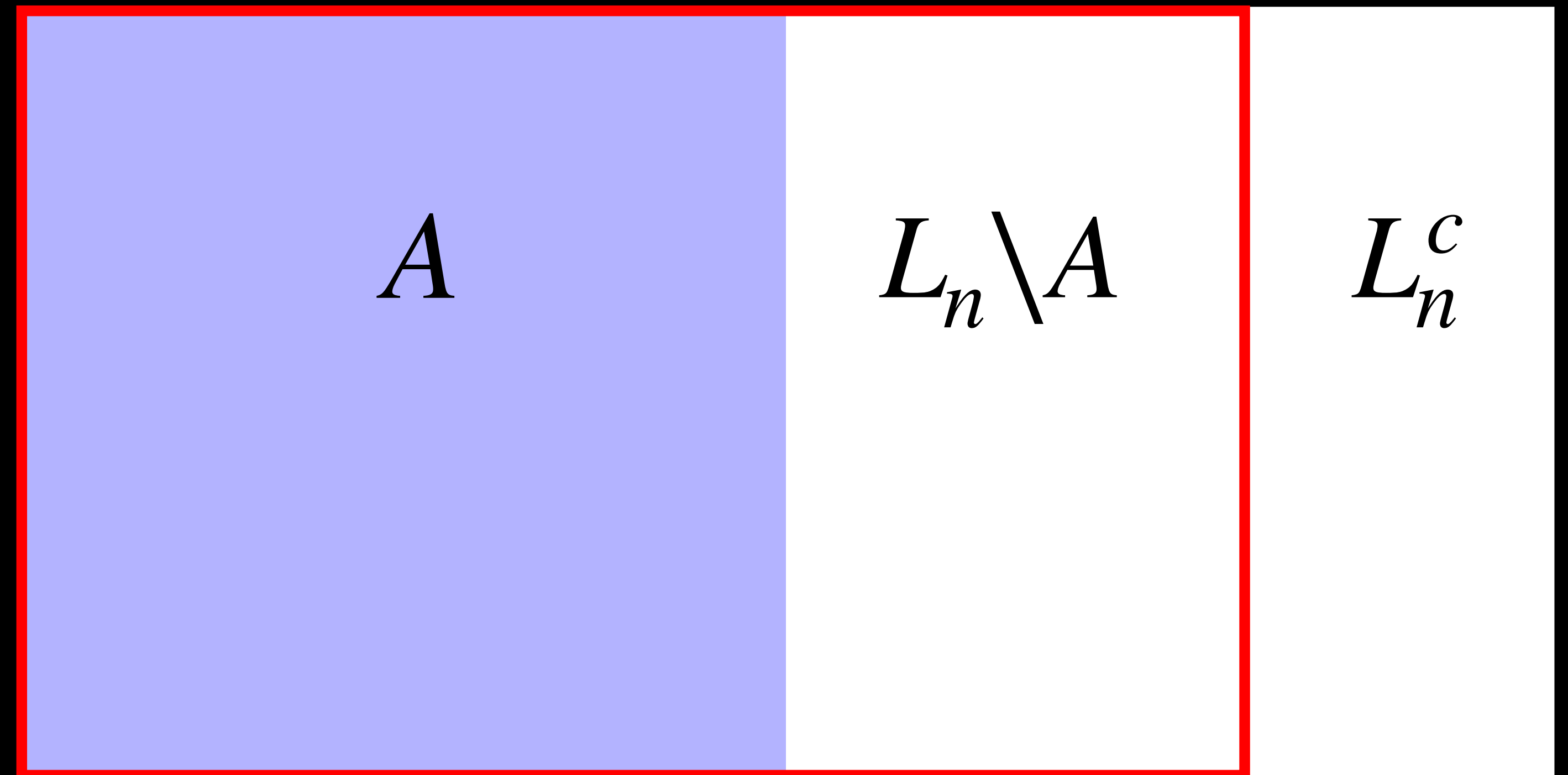
Proof Sketch

Discrepancy Argument

Facts:

(1) $|A| \gg |L_n \setminus A|$.

(2) $|R \cap A| \approx |R \setminus A|$
for all *big* rectangles R .



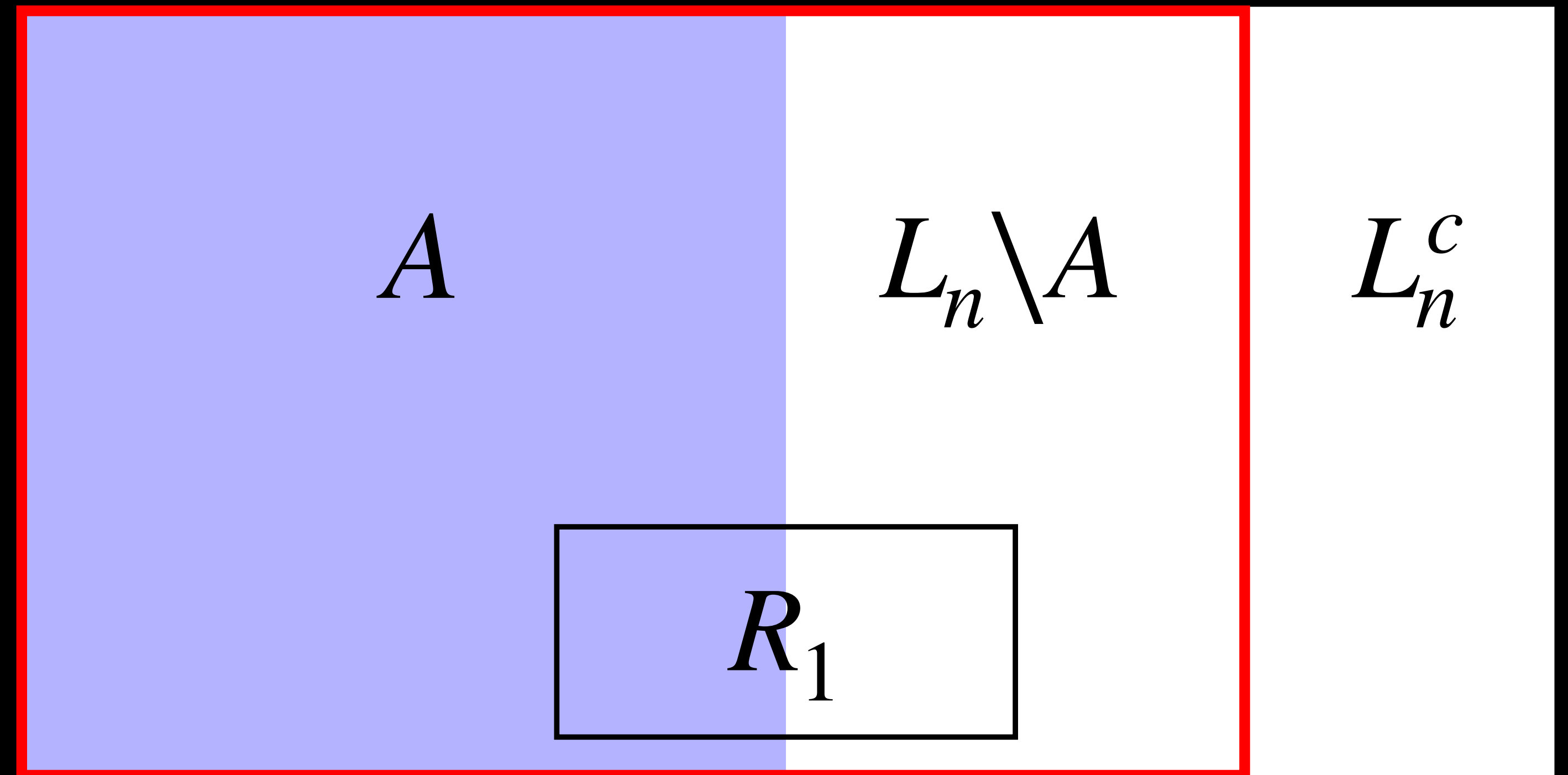
Proof Sketch

Discrepancy Argument

Facts:

(1) $|A| \gg |L_n \setminus A|$.

(2) $|R \cap A| \approx |R \setminus A|$
for all *big* rectangles R .



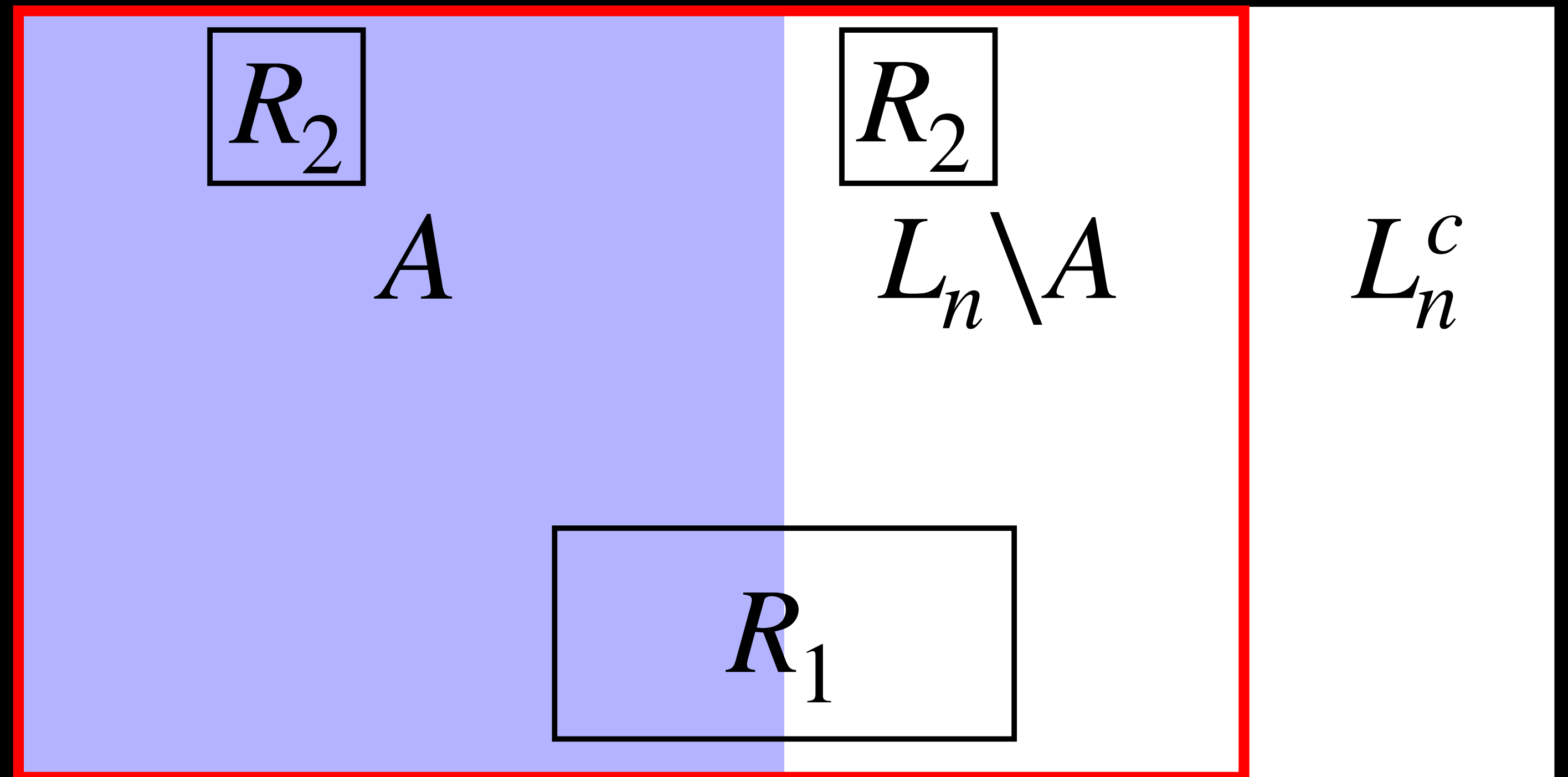
Proof Sketch

Discrepancy Argument

Facts:

(1) $|A| \gg |L_n \setminus A|$.

(2) $|R \cap A| \approx |R \setminus A|$
for all *big* rectangles R .



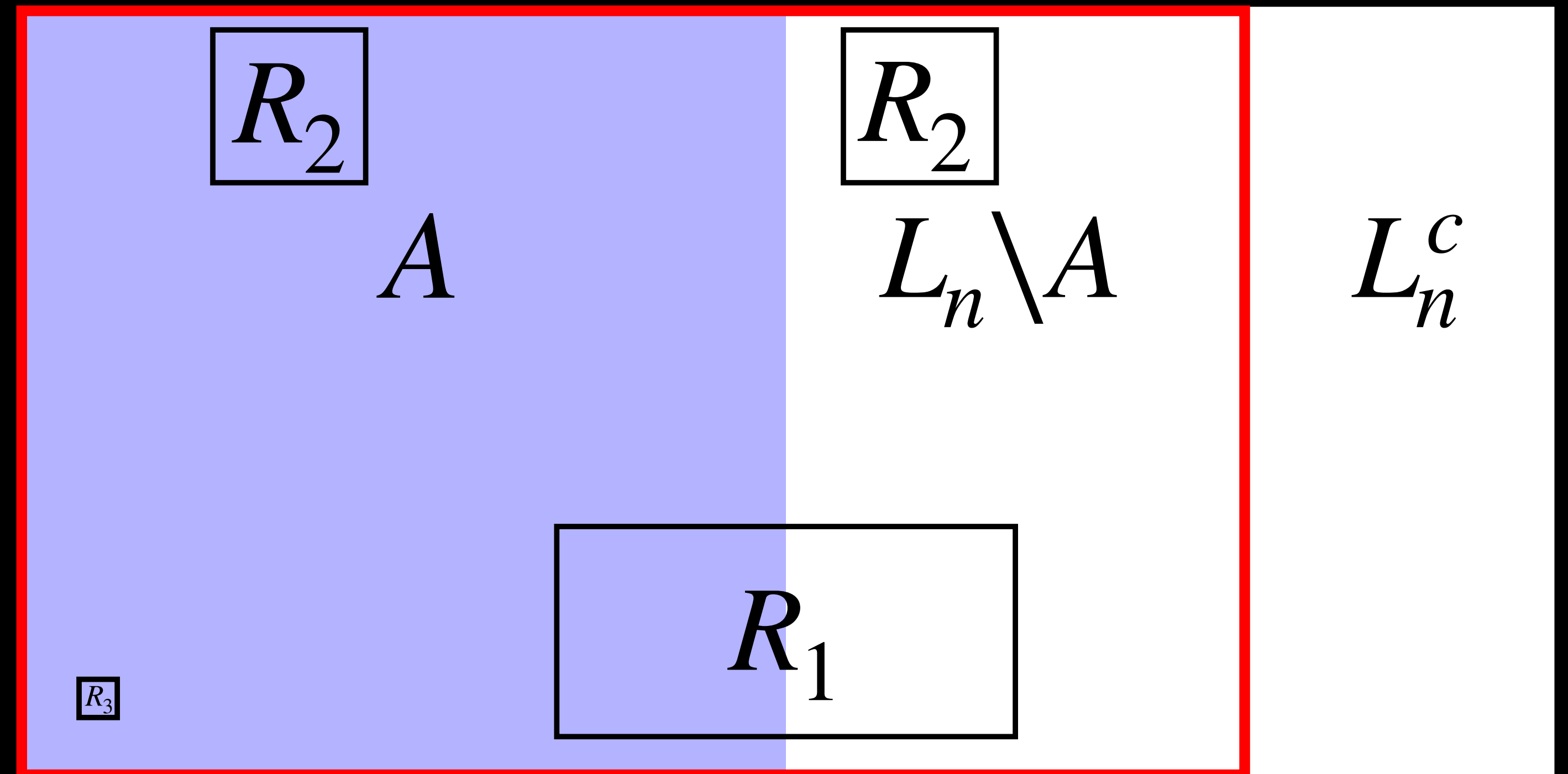
Proof Sketch

Discrepancy Argument

Facts:

(1) $|A| \gg |L_n \setminus A|$.

(2) $|R \cap A| \approx |R \setminus A|$
for all *big* rectangles R .



Proof Sketch

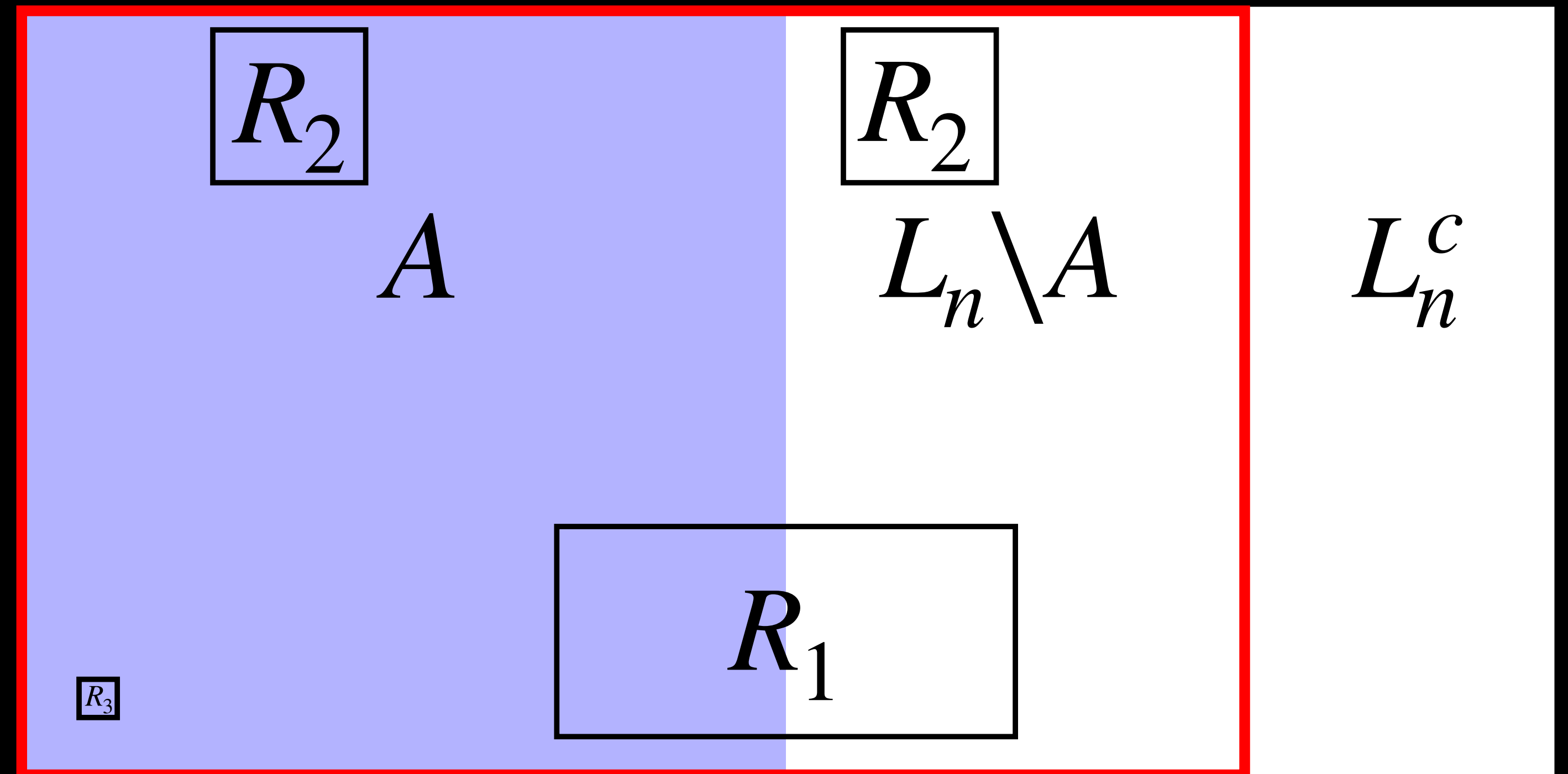
Discrepancy Argument

So a disjoint rectangle cover of L_n must use a lot of small rectangles to cover A .

Facts:

(1) $|A| \gg |L_n \setminus A|$.

(2) $|R \cap A| \approx |R \setminus A|$
for all *big* rectangles R .



Proof Sketch

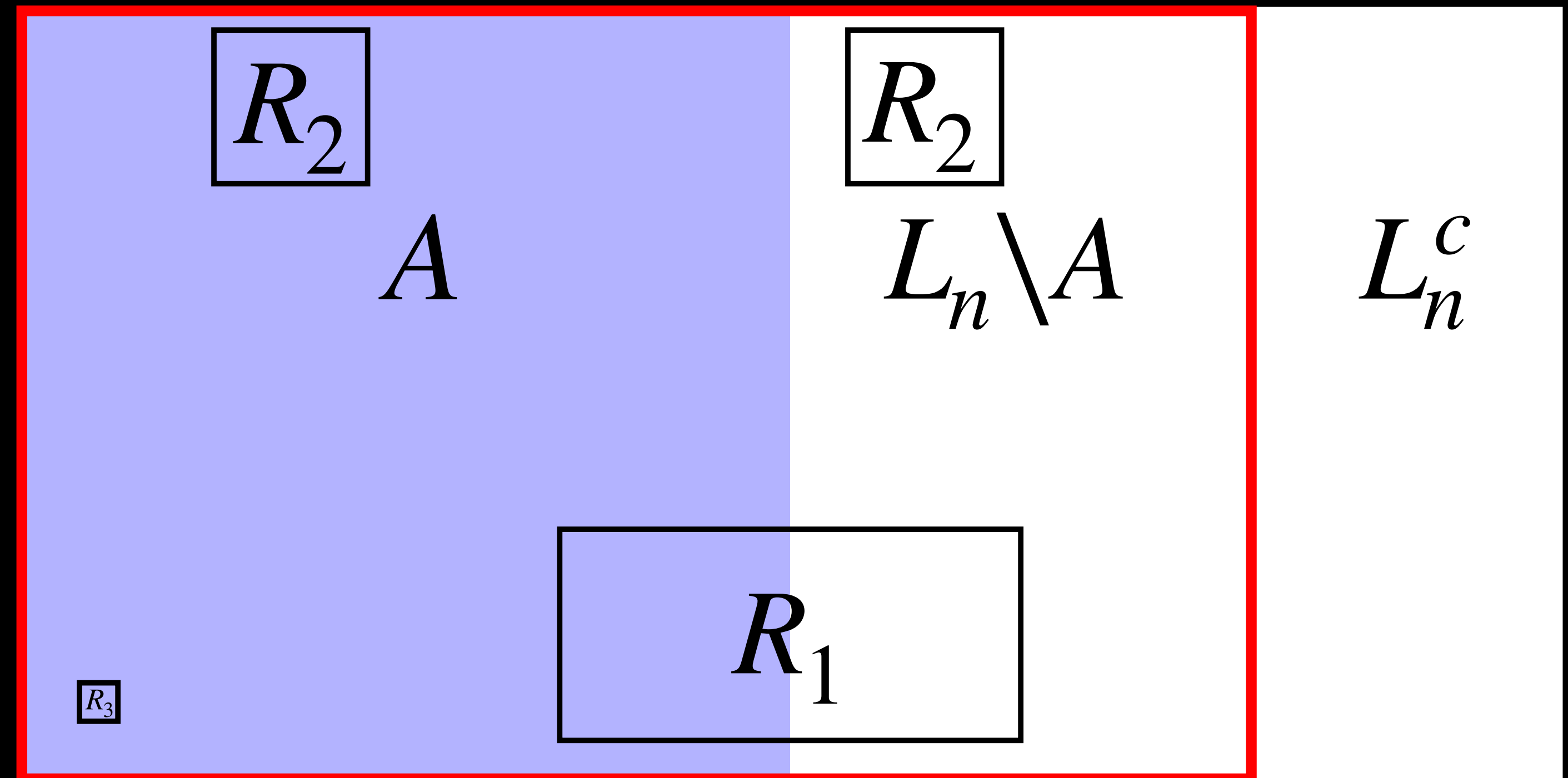
Discrepancy Argument

So every disjoint rectangle cover of L_n must be big.

Facts:

(1) $|A| \gg |L_n \setminus A|$.

(2) $|R \cap A| \approx |R \setminus A|$
for all *big* rectangles R .



Proof Sketch

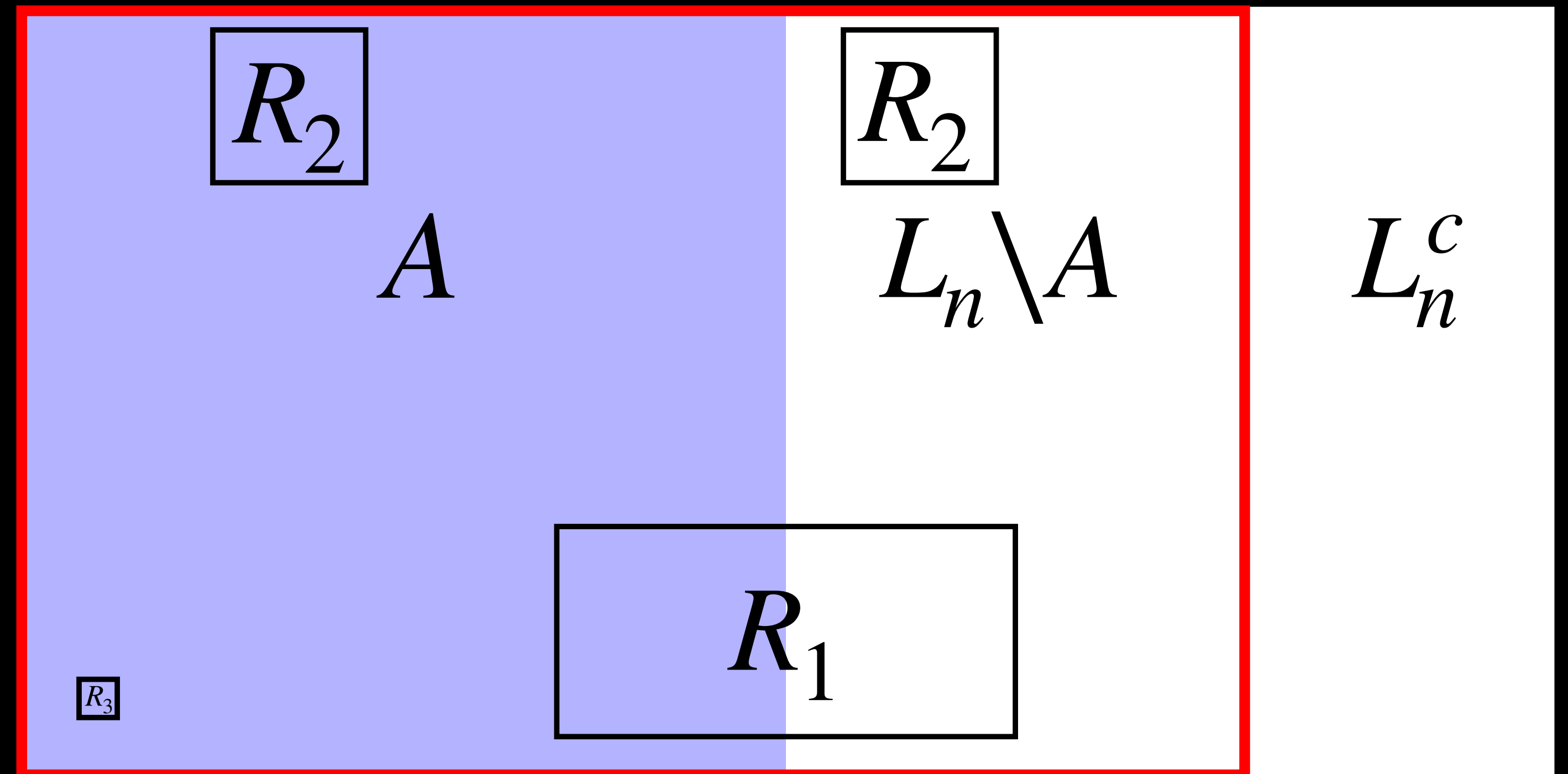
Discrepancy Argument

So every uCFG accepting L_n must be big.

Facts:

(1) $|A| \gg |L_n \setminus A|$.

(2) $|R \cap A| \approx |R \setminus A|$
for all *big* rectangles R .



Sherstov: The multiparty communication complexity of set disjointness. SIAM Journal on Computing (2016)

Outlook

Outlook

- Showed a optimal **double-exponential** succinctness separation of CFGs from uCFGs.

Outlook

- Showed a optimal **double-exponential** succinctness separation of CFGs from uCFGs.
- Lower bound on a natural language: could be basis for other lower bounds.

Outlook

- Showed an optimal **double-exponential** succinctness separation of CFGs from uCFGs.
- Lower bound on a natural language: could be basis for other lower bounds.
- Can we use similar techniques to answer other questions on ambiguity?

Outlook

- Showed a optimal **double-exponential** succinctness separation of CFGs from uCFGs.
- Lower bound on a natural language: could be basis for other lower bounds.
- Can we use similar techniques to answer other questions on ambiguity?
 - **Particularly intriguing**: understanding negation!

Outlook

- Showed an optimal **double-exponential** succinctness separation of **CFGs** from **uCFGs**.
- Lower bound on a natural language: could be basis for other lower bounds.
- Can we use similar techniques to answer other questions on ambiguity?
 - **Particularly intriguing**: understanding negation!
 - Recent quasi-poly lower bounds for **UFA** and **structured d-DNNF**.

Göös, Kiefer, and Yuan. “Lower Bounds for Unambiguous Automata via Communication Complexity.” ICALP (2022)

H.V.S.: “Structured d-DNNF is Not Closed Under Negation.” IJCAI (2024).

Outlook

- Showed an optimal **double-exponential** succinctness separation of **CFGs** from **uCFGs**.
- Lower bound on a natural language: could be basis for other lower bounds.
- Can we use similar techniques to answer other questions on ambiguity?
 - **Particularly intriguing**: understanding negation!
 - Recent quasi-poly lower bounds for **UFA** and **structured d-DNNF**.
 - What about e.g. **uCFG** or **d-DNNF**?

Göös, Kiefer, and Yuan. “Lower Bounds for Unambiguous Automata via Communication Complexity.” ICALP (2022)

H.V.S.: “Structured d-DNNF is Not Closed Under Negation.” IJCAI (2024).

END

Rectangle:

Rectangle:

$$R = \{x_1 y x_2 \mid x_1 x_2 \in L_1, y \in L_2, |x_1| = a_1, |x_2| = a_2\}$$

Rectangle:

$$R = \{x_1 y x_2 \mid x_1 x_2 \in L_1, y \in L_2, |x_1| = a_1, |x_2| = a_2\}$$

$$L_1 \subseteq \Sigma^{a_1+a_2}$$

Rectangle:

$$R = \{x_1 y x_2 \mid x_1 x_2 \in L_1, y \in L_2, |x_1| = a_1, |x_2| = a_2\}$$

$$L_1 \subseteq \Sigma^{a_1 + a_2}$$

$$L_2 \subseteq \Sigma^b$$

Rectangle:

$$R = \{x_1 y x_2 \mid x_1 x_2 \in L_1, y \in L_2, |x_1| = a_1, |x_2| = a_2\}$$

$$L_1 \subseteq \Sigma^{a_1 + a_2}$$

$$L_2 \subseteq \Sigma^b$$

$$\min(a_1 + a_2, b) \geq n/3$$

Rectangle:

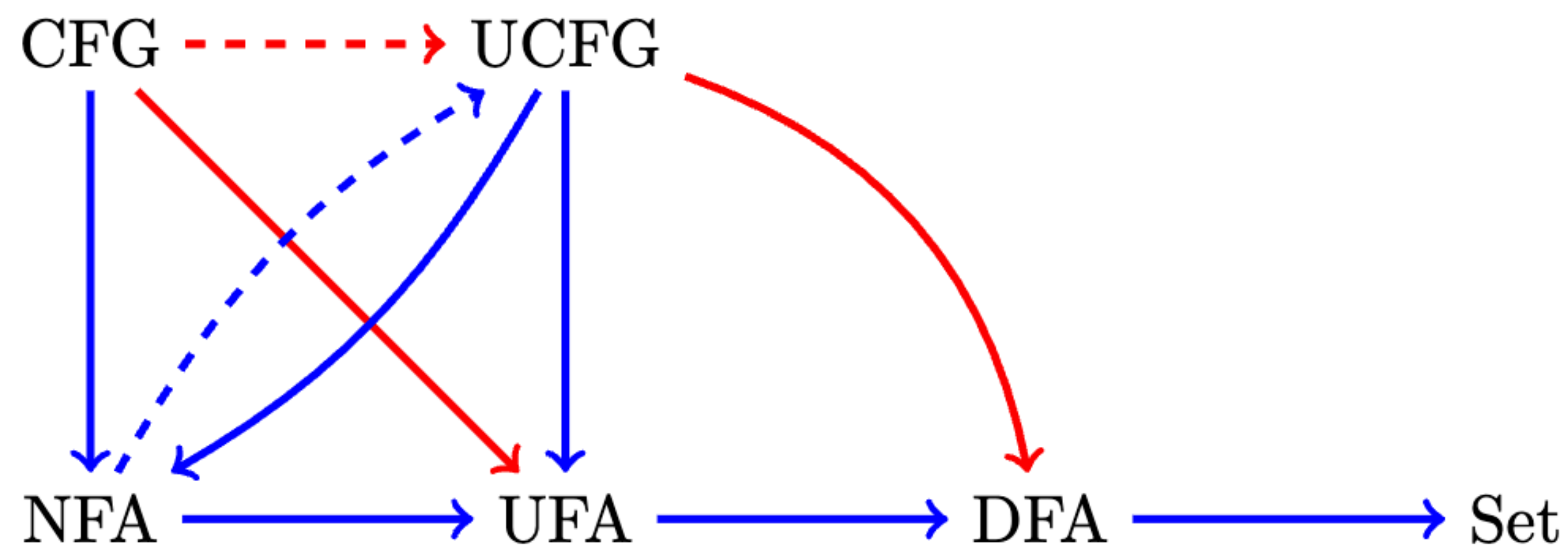
$$R = \{x_1 y x_2 \mid x_1 x_2 \in L_1, y \in L_2, |x_1| = a_1, |x_2| = a_2\}$$

$$L_1 \subseteq \Sigma^{a_1 + a_2}$$

$$L_2 \subseteq \Sigma^b$$

$$\min(a_1 + a_2, b) \geq n/3$$

$$a_1 + a_2 + b = n$$



■ **Figure 3** Worst-case unavoidable blow-ups for succinct representations of uniform length relations. Every path that consists of only blue edges represents an unavoidable exponential blow-up and every path that contains at least one red (solid) edge represents an unavoidable double exponential blow-up. If there is no path, then there exists a linear translation. For the dashed edges, we only prove an upper bound. The corresponding lower bounds are conditional on Conjecture 5.7.

Kimelfeld, Martens and Niewerth: [A Formal Language Perspective on Factorized Representations](#). ICDT (2025)

Image Credit

- **Big duck:** Jaykov <https://universe.roboflow.com/jaykov-l5qjv/ducks-eqnsf>
(<https://creativecommons.org/licenses/by/4.0/>)
- **Small Duck:** Steve Miller <https://www.flickr.com/photos/smiller999/29012477655> (<https://creativecommons.org/licenses/by-nc-sa/2.0/>)
- **Bad duck drawing:** Bella454578 <https://www.deviantart.com/bella454578/art/A-BAD-DRAWING-OF-A-BIRD-627093890> (<https://creativecommons.org/licenses/by-nd/3.0/>)