Circuits and Branching Programs in Meta-Complexity

Ludmila Glinskih

Thesis committee:
Mark Bun
Sofya Raskhodnikova
Steven Homer
Marco Carmosino

Department of Computer Science
Boston University
April 5, 2024

Outline

- Importance of studying computational complexity
- Meta-complexity
- Connections between circuit and structural complexity
- Complexity of representing MCSP via 1-NBP
- Complexity of branching program minimization

Computational complexity

Studies how much computational resources are required for solving a computational problem

Possible resources:

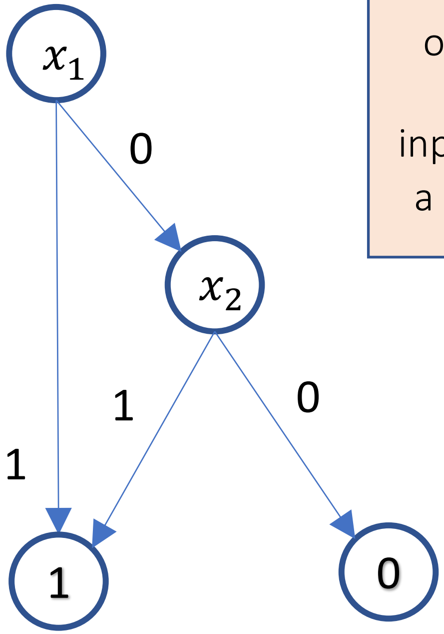
- Time
- Space
- Randomness
- Bits of communication
- ...

focus of my work

Computational models

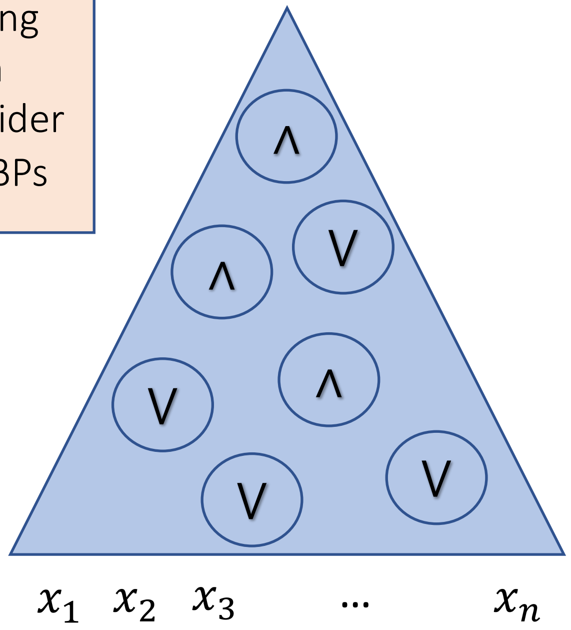
State: s1
Transitions: …

Turing machine



Branching program (BP)

one circuit/branching program for each input length => consider a family of circuits/BPs



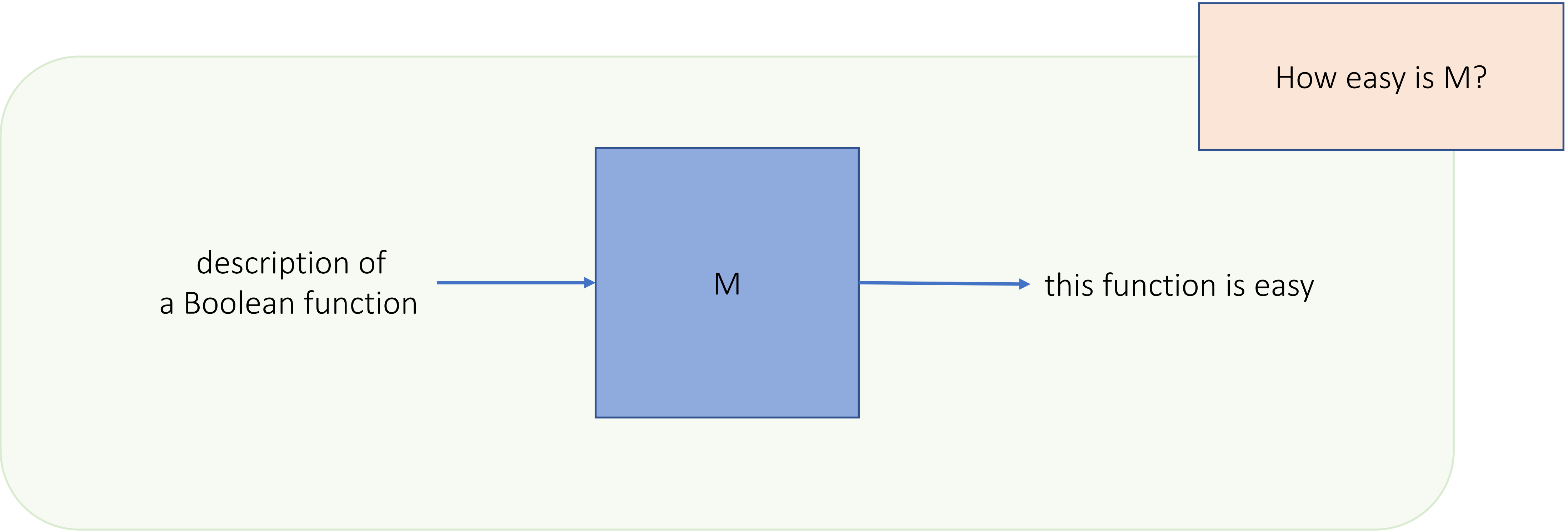
Boolean circuit

Known hardness results

- Almost all boolean functions require large circuits [Shannon’49]
  - Method: counting argument
  - The same result holds for branching programs and for time-complexity in the Turing machine model
- The best lower bound for an explicit function
  - $\Omega(n)$ for Boolean circuits [Find, Golovnev, Hirsch, Kulikov’15]
  - $\widetilde{\Omega}(n^2)$ for branching programs [Nechiporuk’66]
  - $\widetilde{\Omega}(n^{1.5})$ for Turing machines [Kalyanasundaram, Schnitger’92]

Meta-complexity

Studies complexity of functions which compute complexity of an input function



Minimum Circuit Size Problem

Input:

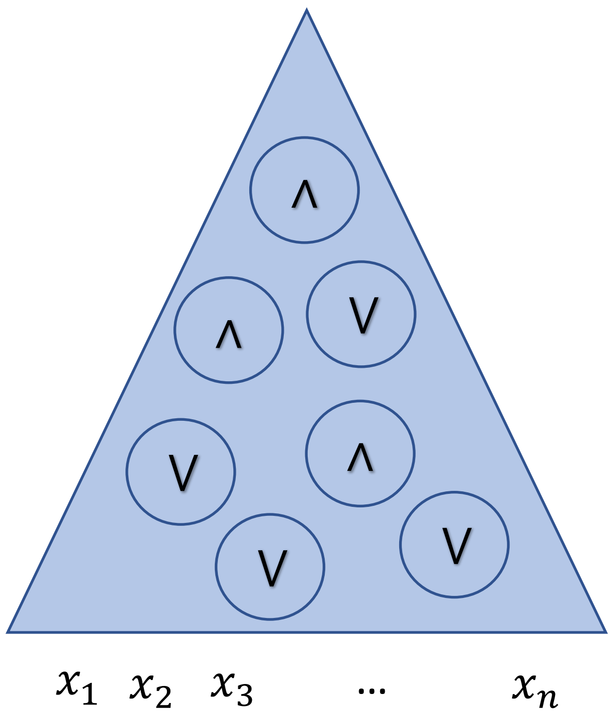
- truth table of a Boolean function $f: \{0,1\}^n \to \{0,1\}$
- size parameter s

1	0	0	1	0	1	1	0	…	1

Truth table of f of length $N = 2^n$

Output:

yes, if f can be computed by a circuit of size at most s



Hardness of MCSP

- MCSP is in NP
  Guess a circuit and check, whether it computes f or not

- MCSP $\in P \Rightarrow$ no strong PRGs [Razborov, Rudich, 1994]

- MCSP is NP-complete $\Rightarrow EXP \neq ZPP$ [Murray, Williams, 2015]

- Complexity of MCSP in restricted classes is important too:
  If MCSP cannot be computed by
    - branching program of size $N^{2.01}$
    - or circuit of size $N^{1.01}$

  Then NP $\not\subset$ BP-SIZE[n^k] (or SIZE$[n^k]$) for all k [Chen, Jin, Williams, 2019]

Goal of my dissertation

Understand connections between circuit and branching program complexity of meta-complexity problems

Results I present:

I. New Karp-Lipton Theorems from RP circuit lower bounds — [Bun, Carmosino, G]

II. MCSP is hard for read-once nondeterministic branching programs — [G, Riazanov'22]

III. Partial minimum branching program size problem is ETH-hard — [G, Riazanov'24]

Plan for the remainder of the talk

I. New Karp-Lipton Theorems from RP Circuit Lower Bounds

II. MCSP is Hard for Read-Once Nondeterministic Branching Programs

III. Partial Minimum Branching Program Size Problem is ETH hard

Original Karp-Lipton Theorem

Theorem [Karp–Lipton'80]:

Every language in NP can be computed by a poly-size circuit

$$\underbrace{\mathrm{NP} \subset \mathrm{P}/poly}_{\text{non-uniform}} \Rightarrow \underbrace{\mathrm{PH} = \Sigma_2}_{\text{uniform}}$$

Polynomial hierarchy collapses to the second level

Recent Karp-Lipton style result

Theorem [Impagliazzo, Kabanets, Volkovich'18]:

PSPACE is bigger than PH
$\mathrm{ZPP}^{\mathrm{MCSP}}$ is smaller than Σ_2

$$\mathrm{PSPACE} \subset \mathrm{P}/poly \Rightarrow \mathrm{PSPACE} \subseteq \mathrm{ZPP}^{\mathrm{MCSP}}$$

Can we get stronger conditional collapses with a stronger complexity assumptions?

Stronger Karp-Lipton theorems from additional assumptions

Theorem [Chen, McKay, Murray, Williams'19]:

Suppose $\mathrm{NP} \not\subset \mathrm{SIZE}[n^k]$ for all k. Then for all $\epsilon > 0$

$$\mathrm{PSPACE} \subset \mathrm{P}/poly \Rightarrow \mathrm{PSPACE} \subset_{i.o} \mathrm{NP}/n^{\epsilon}$$

It is not known whether NP/n^{ϵ} is smaller than $\mathrm{ZPP}^{\mathrm{MCSP}}$

Can we get a deeper collapse if we assume hardness of a complexity class smaller than NP? For instance, RP?

Our result

Theorem 1: Suppose $\mathrm{RP} \not\subset \mathrm{SIZE}[n^k]$ for all k. Then there exists $c > 0$ such that either

- $\mathrm{PSPACE} \subset \mathrm{P}/poly \Rightarrow \mathrm{LINSPACE} \subset_{i.o} \mathrm{ZPP}/n^c$
- $\mathrm{PSPACE} \subset \mathrm{P}/poly \Rightarrow \mathrm{PSPACE} \subset_{i.o} \mathrm{SUBEXP}^{\mathrm{MCSP}}$

Proof ideas: what hardness of RP gives us

$\mathrm{PSPACE} \subset \mathrm{P}/poly \Rightarrow \mathrm{PSPACE} \subseteq \mathrm{ZPP}^{\mathrm{MCSP}}$ [IKV'18]

Theorem 1: Suppose $\mathrm{RP} \not\subset \mathrm{SIZE}[n^k]$ for all k. Then there exists $c > 0$ so either

- $\boxed{\mathrm{PSPACE} \subset \mathrm{P}/poly \quad \Rightarrow \mathrm{LINSPACE} \subset_{i.o} \mathrm{ZPP}/n^c}$
- $\mathrm{PSPACE} \subset \mathrm{P}/poly \quad \Rightarrow \mathrm{PSPACE} \subset_{i.o} \mathrm{SUBEXP}^{\mathrm{MCSP}}$

Proof ideas:

Based on the [IKV'18] result we either:

- Get rid of the MCSP oracle
- Or derandomize ZPP into SUBEXP

MCSP oracle plays a role of a *natural property* =>
we need to extract a natural property from
the assumption $\mathrm{RP} \not\subset \mathrm{SIZE}[n^k]$

Natural property

An algorithm is called a natural property if

- It runs in polynomial time
- It outputs *no* on all easy functions
- It outputs *yes* on a significant fraction of functions

constructivity

usefulness

largeness

An efficient algorithm for MCSP can be converted to a natural property

RP-verifiers

$\boxed{M(x, r)}$ is an RP-verifier for $L \in \mathrm{RP}$

- Runs in polynomial time
- Rejects **all** random seeds r if x is not in the language
- Accepts at least **half** of random seeds r if x is in the language

Algorithm $A(T)$ is a natural property

- Runs in polynomial time
- Rejects **all** input truth tables T with a small circuit complexity
- Accepts a **significant fraction** of all truth tables

What if we fix an input $x \in L$ such that every r that $M(x, r)$ accepts, has large circuit complexity?

Natural property from RP seeds hardness

Assume that exists a language L in RP that does not have small seeds:

For every RP-verifier M holds for infinitely many $x \in L$:
$M(x,r) = 1$ implies that r has a large circuit complexity

We get a natural property algorithm A as follows:

- Fix an input x such that $M(x,r)$ accepts the seed r only if it is hard
- Set $A(\cdot) = M(x,\cdot)$, then if $A(T) = 1$ => T is hard

Non-uniform choice of ***x*** is the reason why we need an advice

Proof ideas: what hardness of RP gives us

$\mathrm{PSPACE} \subset \mathrm{P}/poly \Rightarrow \mathrm{PSPACE} \subseteq \mathrm{ZPP}^{\mathrm{MCSP}}$ [IKV'18]

Theorem 1: Suppose $\mathrm{RP} \not\subset \mathrm{SIZE}[n^k]$ for all k. Then there exists $c > 0$ so either

- $\boxed{\mathrm{PSPACE} \subset \mathrm{P}/poly \quad \Rightarrow \mathrm{LINSPACE} \subset_{i.o} \mathrm{ZPP}/n^c}$
- $\mathrm{PSPACE} \subset \mathrm{P}/poly \quad \Rightarrow \mathrm{PSPACE} \subset_{i.o} \mathrm{SUBEXP}^{\mathrm{MCSP}}$

Proof ideas:

Based on the [IKV'18] result we either:

- Get rid of the MCSP oracle
- Or derandomize ZPP into SUBEXP

MCSP oracle plays a role of a *natural property* => we need to extract a natural property from the assumption $\mathrm{RP} \not\subset \mathrm{SIZE}[n^k]$

But our current assumption: RP-seeds $\not\subset \mathrm{SIZE}[n^k]$

Easy-seeds and Kolmogorov’s conjecture

We want to relate the circuit complexity of RP-seeds with the circuit complexity of RP

Conjecture: Suppose $\mathrm{PSPACE} \subset \mathrm{P}/poly$ and there exists k such that all RP-seeds $\subset \mathrm{SIZE}[n^k]$. Then $\mathrm{RP} \subset \mathrm{SIZE}[n^{poly(k)}]$

We do not know a *nice* RP-complete language, and we do not know whether this conjecture is true

Kolmogorov’s Conjecture: $\mathrm{P} \subset \mathrm{SIZE}[n^c]$ for some c

Easy-witness lemma for RP

$PSPACE \subset P/poly$

Kolmogorov's conjecture

$NP \subset P/poly$, RP has easy seeds, $P \subset SIZE[n^k]$

We can construct a small circuit for every language L in RP

To construct a fixed poly-size circuit for L we use fixed poly-size circuits for:

- Circuit-SAT problem
- Seed for *yes*-instances
- Circuit for an RP-verifier

Without Kolmogorov's conjecture we cannot have a fixed bound on the circuit size for all RP verifiers

Proof ideas: what hardness of RP gives us

$\mathrm{PSPACE} \subset \mathrm{P}/poly \Rightarrow \mathrm{PSPACE} \subseteq \mathrm{ZPP}^{\mathrm{MCSP}}$ [IKV'18]

Theorem 1: Suppose $\mathrm{RP} \not\subset \mathrm{SIZE}[n^k]$ for all k. Then there exists $c > 0$ so either

- $\boxed{\mathrm{PSPACE} \subset \mathrm{P}/poly \quad \Rightarrow \mathrm{LINSPACE} \subset_{i.o} \mathrm{ZPP}/n^c}$
- $\mathrm{PSPACE} \subset \mathrm{P}/poly \quad \Rightarrow \mathrm{PSPACE} \subset_{i.o} \mathrm{SUBEXP}^{\mathrm{MCSP}}$

assuming Kolmogorov's conj is **true**

Proof ideas:

Based on the [IKV'18] result we either:

- Get rid of the MCSP oracle
- Or derandomize ZPP into SUBEXP

Assuming hardness of RP and Kolmogorov's conjecture we get a natural property, which we use instead of MCSP in [IKV'18] result

Proof ideas: what hardness of RP gives us

$\mathrm{PSPACE} \subset \mathrm{P}/poly \Rightarrow \mathrm{PSPACE} \subseteq \mathrm{ZPP}^{\mathrm{MCSP}}$ [IKV'18]

Theorem 1: Suppose $\mathrm{RP} \not\subset \mathrm{SIZE}[n^k]$ for all k. Then there exists $c > 0$ so either

- $\mathrm{PSPACE} \subset \mathrm{P}/poly \quad \Rightarrow \mathrm{LINSPACE} \subset_{i.o} \mathrm{ZPP}/n^c$
- $\boxed{\mathrm{PSPACE} \subset \mathrm{P}/poly \quad \Rightarrow \mathrm{PSPACE} \subset_{i.o} \mathrm{SUBEXP}^{\mathrm{MCSP}}}$

assuming Kolmogorov's conj is **false**

Proof ideas:

Based on the [IKV'18] result we either:

- Get rid of the MCSP oracle
- Or derandomize ZPP into SUBEXP

Assuming the Kolmogorov's conjecture is false
we get a hard function in P, which we use
for derandomizing ZPP in the [IKV'18] result

Umans' pseudorandom generator

If Kolmogorov's conjecture **does not** hold

$\Downarrow$

for every k exists a language $L \in \mathrm{P}$ such that $L \notin \mathrm{SIZE}[\mathrm{n}^{\mathrm{k}}]$

Using the hard language L we derandomize $\mathrm{ZPP}^{\mathrm{MCSP}}$ into $\mathrm{SUBEXP}^{\mathrm{MCSP}}$

using Umans' generator [Umans'02]

Putting everything together

Theorem 1: Suppose $\mathrm{RP} \not\subset \mathrm{SIZE}[n^k]$ for all k. Then there exists $c > 0$ so either

- $\mathrm{PSPACE} \subset \mathrm{P}/poly \quad \Rightarrow \mathrm{LINSPACE} \subset_{i.o} \mathrm{ZPP}/n^c$
- $\mathrm{PSPACE} \subset \mathrm{P}/poly \quad \Rightarrow \mathrm{PSPACE} \subset_{i.o} \mathrm{SUBEXP}^{\mathrm{MCSP}}$

Consider Kolmogorov Conjecture $\mathrm{P} \subset SIZE[n^c]$ for some c

- If it is true ⇒ combining with $\mathrm{RP} \not\subset \mathrm{SIZE}[n^k]$ assumption, we get a natural property ⇒ we use it instead of MCSP in $\mathrm{ZPP}^{\mathrm{MCSP}}$

- If it is false ⇒ exists a hard function in P, which we use to derandomize $\mathrm{ZPP}^{\mathrm{MCSP}}$ into $\mathrm{SUBEXP}^{\mathrm{MCSP}}$ in [IKV'18]

Next steps in strengthening our KL theorem

- Understand, whether an NP-intermediate version of MCSP is sufficient to get a similar Karp-Lipton theorem as [IKV'18] got
  - Then we would get that $ZPP^{\widetilde{MCSP}}$ is a smaller class than ZPP^{SAT}

- Currently, in one of the branches of our proof we assume that Kolmogorov's conjecture holds ($\mathrm{P} \subset \mathrm{SIZE}[n^k]$)
  - We need this assumption to extract a natural property from the hardness assumption on RP
  - Can we extract natural property without this assumption, or show that existence of extractable natural property from hardness of RP implies that Kolmogorov conjecture holds?

Plan for the remainder of the talk

I. New Karp-Lipton Theorems from RP Circuit Lower Bounds

II. MCSP is Hard for Read-Once Nondeterministic Branching Programs

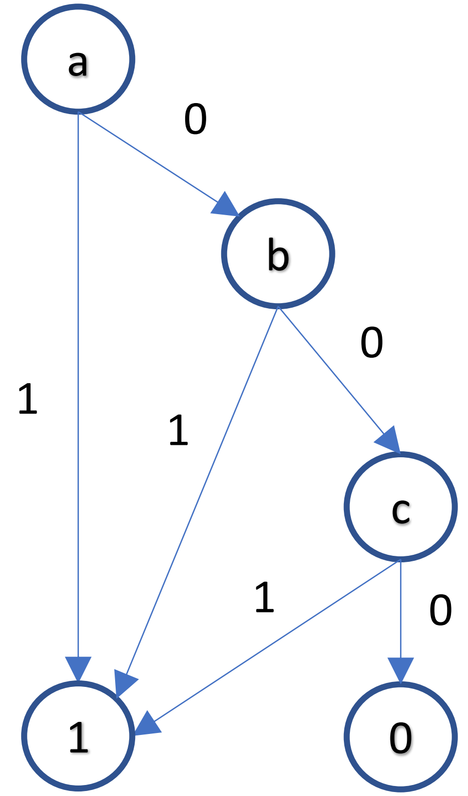
III. Partial Minimum Branching Program is ETH hard

MCSP vs 1-NBP

Theorem 2: size of every read-once nondeterministic branching program computing MCSP is $N^{\boldsymbol{\Omega}(\mathbf{loglog}\, N)}$

Branching program

- BP is a way to represent Boolean function:
  - directed graph without cycles
  - one source
  - two sinks: labeled with 0 and 1
  - all other vertices labeled with variables
  - values of variables on edges
- Size of a BP is a number of vertices

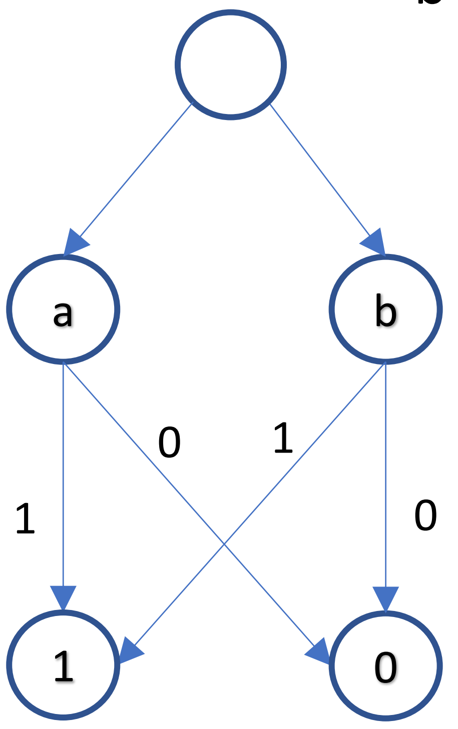


a = 0
b = 1
c = 1

Non-deterministic branching program

- NBP additionally has non-deterministic nodes:
  - non-deterministic nodes are unlabeled
  - the value equals 1 $\Leftrightarrow$ exists a path to 1-sink

a = 1
b = 0

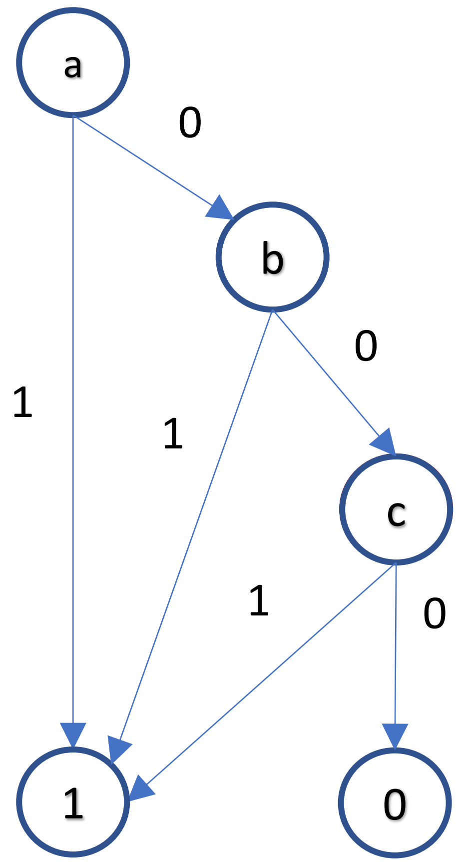


Best lower bounds for branching programs

- At least a $1 - \frac{1}{2^n}$ fraction of functions require BP size $\frac{2^n}{4n}$
- The best lower bound: BP(ED)=$\Omega\left(\frac{n^2}{\log^2 n}\right)$ [Nechiporuk, 1966]
- Recent results:
  - BP(MCSP)=$\widetilde{\Omega}(N^2)$ [Cheraghchi, Kabanets, Lu, Myrisiotis, 2019]
  - Barrier on proving better than $\widetilde{\Omega}(N^2)$ for MCSP [Chen, Jin, Williams, 2019]

Read-Once Branching Programs

1-BP (1-NBP) if for every path every variable occurs no more than 1 time



Known lower bounds for 1-NBPs

- 1-NBP(CLIQUE_ONLY) = $2^{\Omega(\sqrt{n})}$ [Borodin, Razborov, Smolensky, 1993]
- 1-NBP($\oplus_\Delta$)= $2^{\Omega(n)}$ [Duris, Hromkovic, Jukna, Sauerhoff, Schnitger, 2004]
  - $\oplus_\Delta$ parity of triangles in a graph
- 1-NBP(!MCSP) = $2^{\Omega(n)}$ [Cheraghchi, Kabanets, Lu, Myrisiotis, 2019]

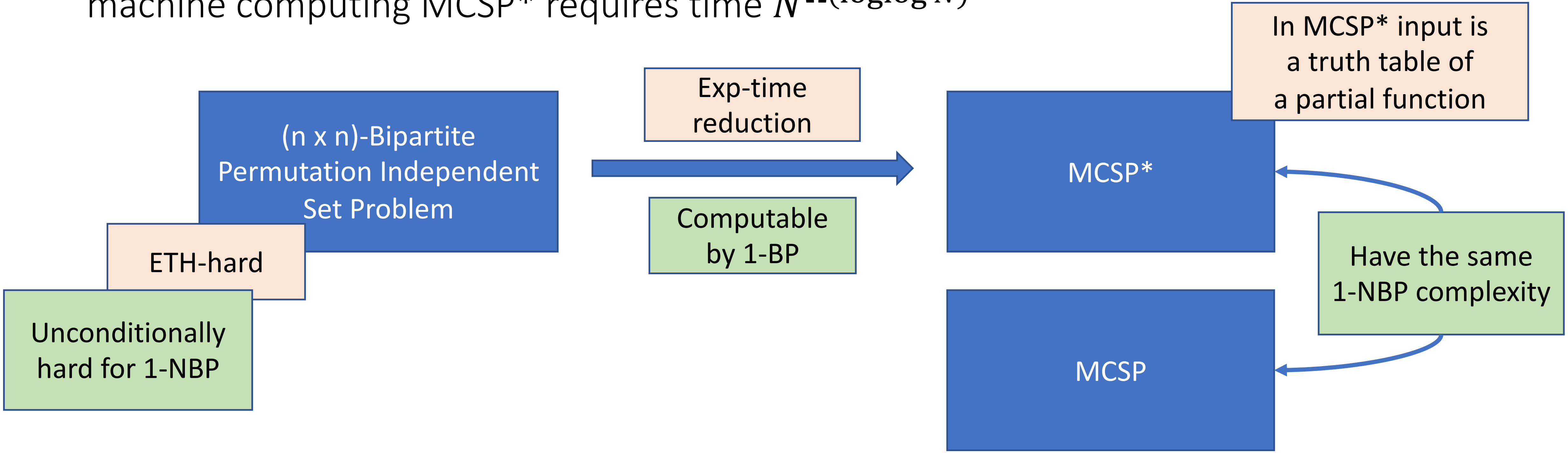
MCSP naturally a nondeterministic problem, so it is harder to prove a lower bound against NBP

Main result

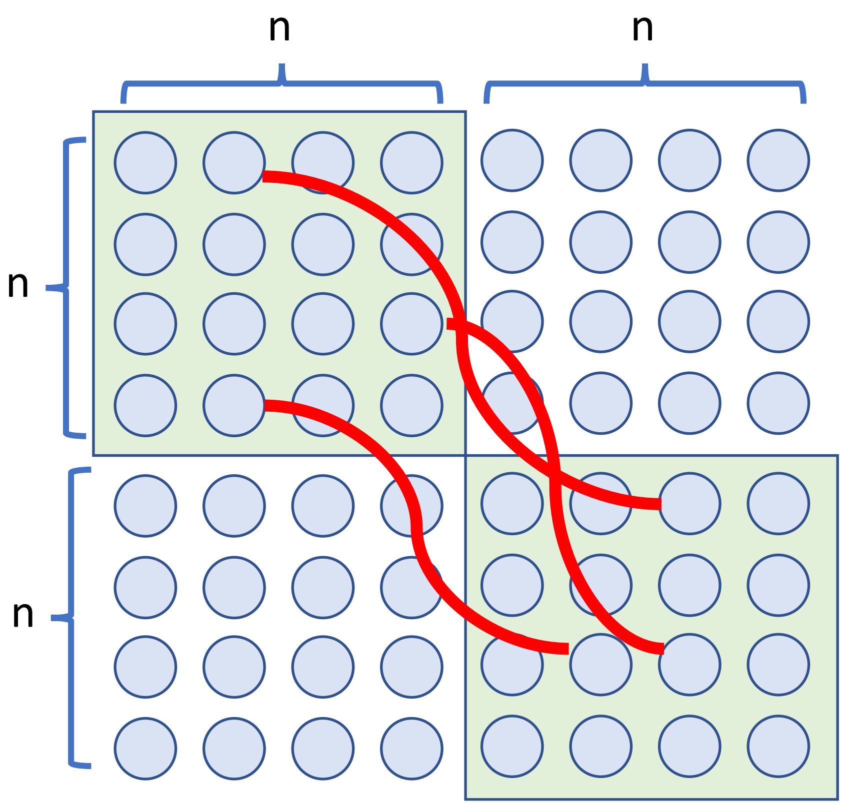
This result is tight for MCSP with linear size parameter

Theorem: size of 1-NBP computing MCSP is $N^{\Omega(\log\log N)}$

Theorem [Ilango'20]: assuming Exponential Time Hypothesis every Turing machine computing MCSP* requires time $N^{\Omega(\log\log N)}$



(n x n)-Bipartite Permutation Independent Set



- Graph with 2n x 2n vertices,
- Edges exist only between vertices from two quadrants
- Need to find exactly one vertex from every row, and exactly one vertex from every column, such that
  - These vertices are from the two quadrants
  - These vertices form independent set

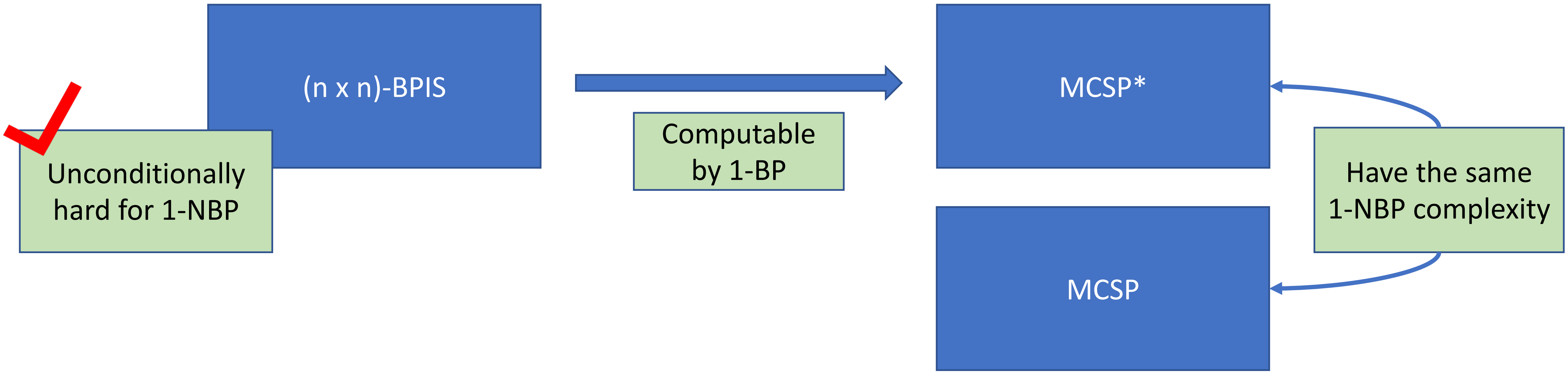
(n x n)-BPIS is hard for 1-NBP

Lemma: size of 1-NBP computing an (n x n)-BPIS is $\Omega(n!)$

Idea of the proof:

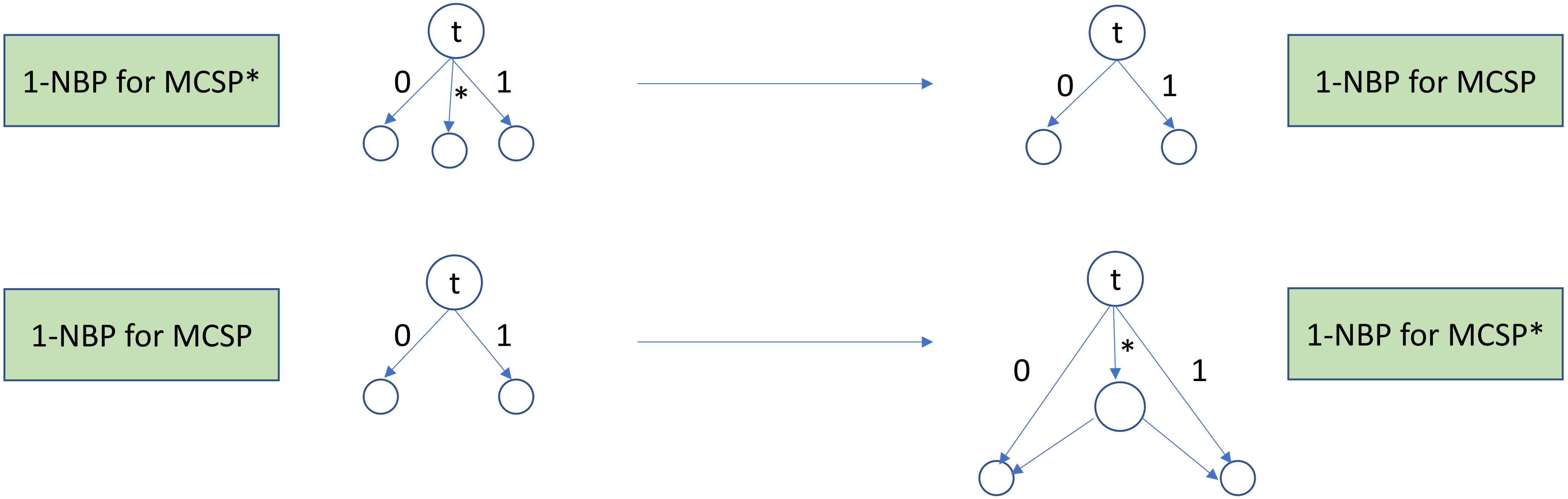
- Show that the minimum 1-NBP for the Bipartite Permutation Independent Set has the same size as the minimum 1-NBP for the Bipartite Permutation Clique
- Adapt the proof of the lower bound on 1-NBP for CLIQUE_ONLY to get a lower bound on the Bipartite Permutation Clique problem

Progress so far

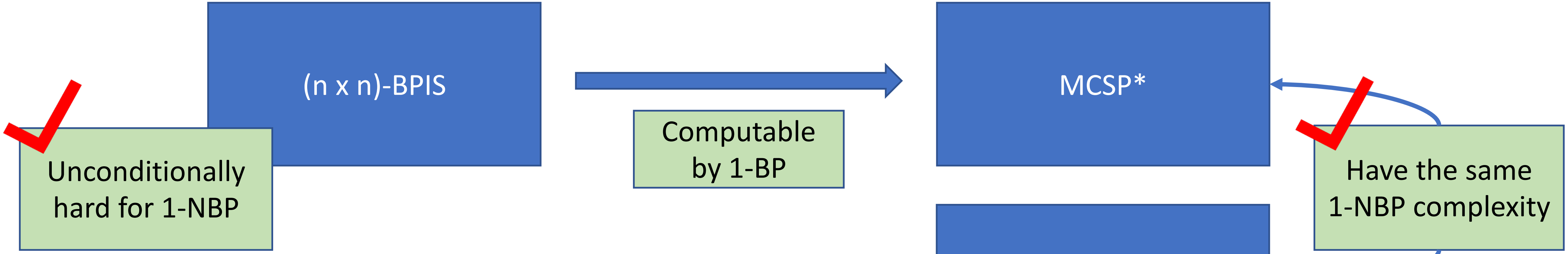


MCSP* and MCSP have the same 1-NBP complexity

Lemma: the size of the minimal 1-NBP computing MCSP* equals the size of the minimal 1–NBP computing MCSP



Putting all together



If MCSP could be computed by a small 1-NBP, then (n x n)-BPIS could be computed by a 1-NBP of even smaller size.
That leads to a contradiction.

Upper bound

Simple guess
and check strategy

Lemma: MCSP on an input of length 2^n with a size parameter s can be computed by a 1-NBP of size $O(2^n 2^{s \log s})$

Corollary: our lower bound is tight for inputs with a linear size parameter

Open questions

- Show tight lower bound for MCSP with higher size parameters
  - The same technique cannot work, as we cannot construct a truth table of a function with higher than linear circuit complexity
- Extend this result to other models of computations
  - For any model in which (n x n)-BPIS is hard and the reduction to the truth table is efficiently computable the same size lower bound will hold

Plan for the remainder of the talk

I. New Karp-Lipton Theorems from RP Circuit Lower Bounds

II. MCSP is Hard for Read-Once Nondeterministic Branching Programs

III. Partial Minimum Branching Program Size Problem is ETH hard

Hardness of branching program minimization

Theorem 3: assuming Exponential Time Hypothesis every Turing machine computing Partial Minimum Branching Program Size Problem requires time $N^{\Omega(\log\log N)}$

holds also for minimizing
1-BP, k-BP, OBDD

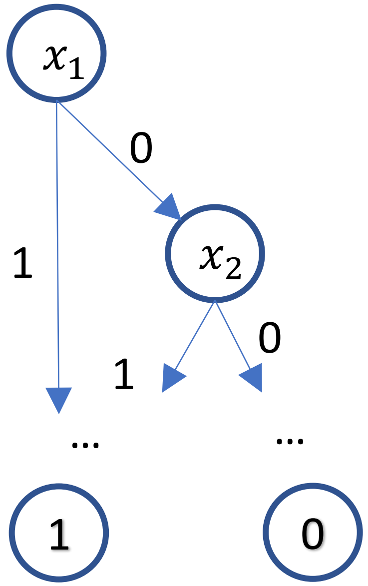
Partial Minimum Branching Program Size Problem

Input:

- truth table of a partial Boolean function $f: \{0,1\}^n \to \{0,1,*\}$
- size parameter s

1	*	*	1	*	1	1	0	…	1

Truth table of f of length $N = 2^n$



Output:

yes, if exists a total function g that is consistent with f and can be computed by a branching program of size at most s

Branching program minimization

Previous results:

Minimization of OBDD is **NP**-hard

- Given an OBDD [Bollig, Wegener'96]
- Given a set of pairs $(x_1, f(x_1)), \ldots, (x_t, f(x_t))$ [Takenaga, Yajima'93][Sieling'02]

Our result:

Minimization of OBDD, k-BP, and BPs is **ETH**-hard given a truth-table of a partial function

Other related minimization problems

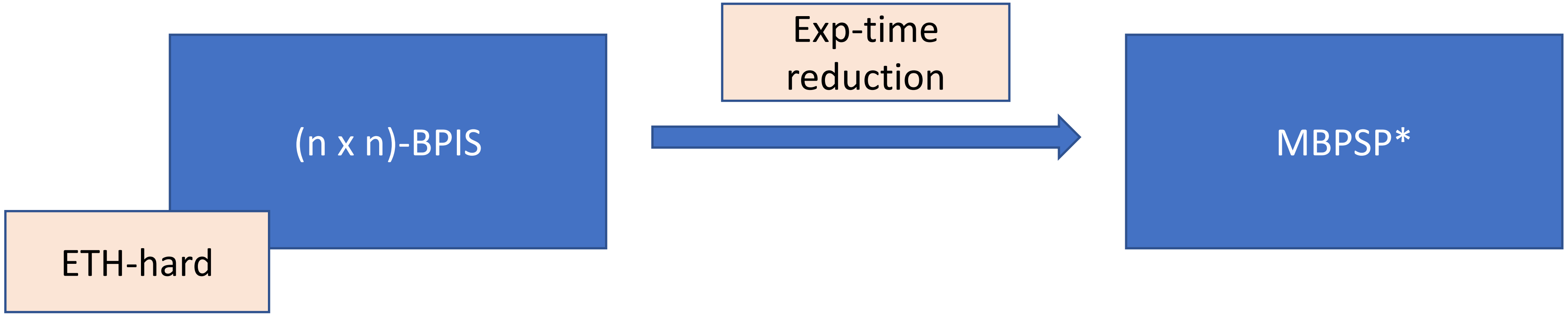
Minimizing the size of

- DNF is NP-hard [Macek’79]
- DeMorgan Formula is ETH-hard [Ilango’21]
  - First shown for a partial version in [Ilango’20]
- Partial MBPSP is ETH-hard [this work]
- Partial MCSP is ETH-hard [Ilango’20]
- Partial MCSP is NP-hard under randomized reductions [Hirahara’22]

Proof idea of hardness MBPSP*

Theorem: assuming Exponential Time Hypothesis every Turing machine computing MBPSP* requires time $N^{\Omega(\log\log N)}$

We use the same proof structure introduce by Ilango for showing ETH-hardness of MCSP*



The hardness reduction

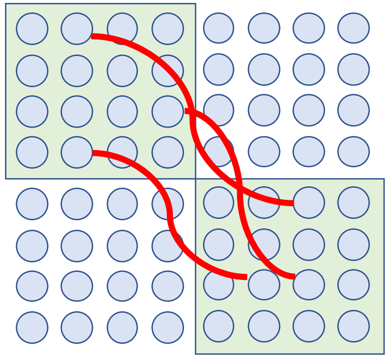
(n x n)-BPIS

ETH-hard

Exp-time reduction

MBPSP*

Graph G



1	*	0	*	*	1	1	0	…	1

Truth table of a partial function γ_G that depends on 6n variables

Key lemma: any total Boolean function consistent with γ_G can be computed by a branching program of size $6n$ ⇔ G is a yes-instance of (n x n)-BPIS

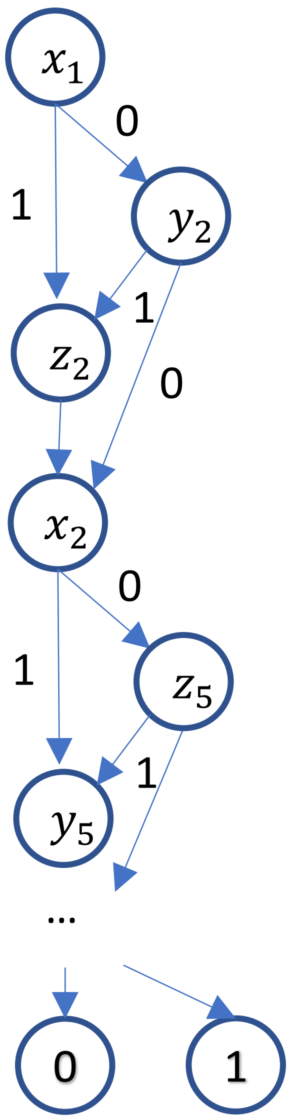
The hardness reduction

Key lemma: any total Boolean function consistent with γ_G can be computed by a branching program of size 6n iff ***G*** is a yes-instance of (n x n)-BPIS

Proof idea:

γ_G depends on $6n$ variables $x_1, \ldots, x_{2n}, y_1, \ldots, y_{2n}, z_1, \ldots, z_{2n}$,

There exists a BP computing γ_G that queries every variable at most once => we can extract a permutation on $[2n]$ corresponding to an independent set in ***G*** from such BP.

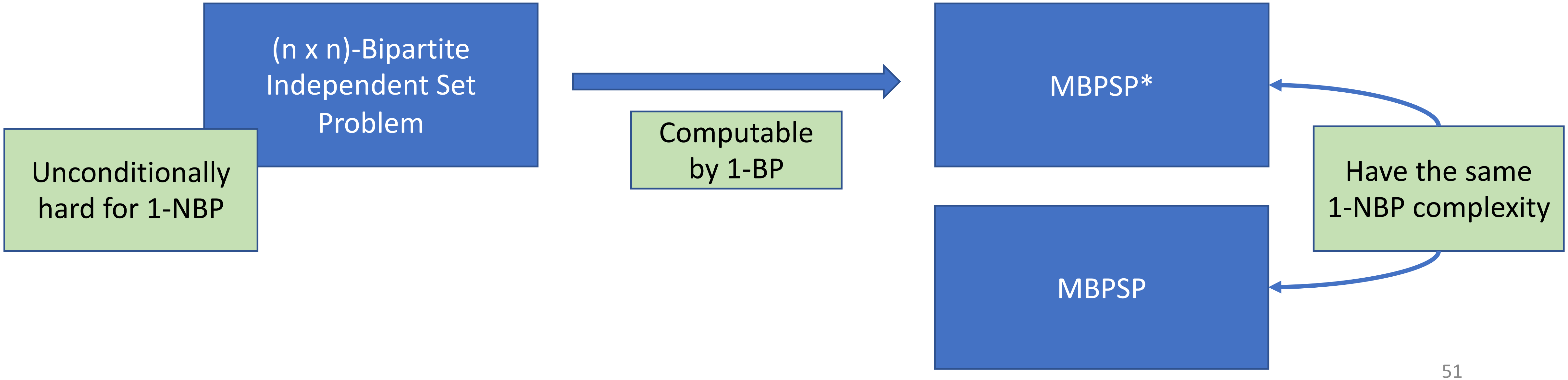


Corollaries

Corollary 1: assuming Exponential Time Hypothesis for every k every Turing machine computing Partial Minimum k-BP Size Problem requires time $N^{\Omega(\log\log N)}$

As the hardness is in distinguishing whether a BP queries every variable exactly once or not

Corollary 2: size of 1-NBP computing MBPSP is $N^{\Omega(\log\log N)}$



Next steps in studying MBPSP

- Extend this result to total MBPSP
  - Already known for DeMorgan Formulas [Ilango’21], DNFs [Masek’79]

- Show NP-hardness of MBPSP*
  - Possibly, using techniques from the work of Hirahara [Hirahara’22]

Recap

- Results covered today:
  - New Karp-Lipton style theorems from hardness assumption on RP [in progress]
  - Unconditional 1-NBP hardness of MCSP [published, LATIN 2022]
  - ETH hardness of Partial MBPSP [in submission, CCC 2024]
    - With an unconditional 1-NBP hardness of branching program minimization for various restricted versions of BPs