Baby PIH: Parameterized Inapproximability of Min CSP

Venkatesan Guruswami
UC Berkeley

Xuandi Ren
UC Berkeley

Sai Sandeep
UC Berkeley

Outline

- Background
  - Parameterized Complexity
  - Constraint Satisfaction Problem (CSP)
  - Parameterized Inapproximability Hypothesis (PIH)
- Our Result
  - Baby PIH
- Proof Overview

Outline

- Background
  - Parameterized Complexity
  - Constraint Satisfaction Problem (CSP)
  - Parameterized Inapproximability Hypothesis (PIH)
- Our Result
  - Baby PIH
- Proof Overview

Parameterized Complexity

- Each input instance x is associated with a parameter $k \in \mathbb{N}$
- Complexity is measured as a function of both $n = |x|$ and k.

- FPT (Fixed-Parameter Tractable):
  - problems that admit $f(k) \cdot n^{O(1)}$ time algorithms, f can be any computable function

Parameterized Complexity

- Each input instance x is associated with a parameter $k \in \mathbb{N}$
- Complexity is measured as a function of both $n = |x|$ and k.

- FPT (Fixed-Parameter Tractable):
  - problems that admit $f(k) \cdot n^{O(1)}$ time algorithms, f can be any computable function

k-Vertex Cover

- Input:
  - $G = (V, E)$
- Output:
  - $\exists v_1, \ldots, v_k \in V$ covering all the edges?

(Multi-colored) k-Clique

- Input:
  - $G = (V = V_1 \cup \cdots \cup V_k, E)$
- Output:
  - $\exists v_1 \in V_1, \ldots, v_k \in V_k$ which form a clique?

Parameterized Complexity

- Each input instance x is associated with a parameter $k \in \mathbb{N}$
- Complexity is measured as a function of both $n = |x|$ and k.

- FPT (Fixed-Parameter Tractable):
  - problems that admit $f(k) \cdot n^{O(1)}$ time algorithms, f can be any computable function

k-Vertex Cover

- Input:
  - $G = (V, E)$
- Output:
  - $\exists v_1, \ldots, v_k \in V$ covering all the edges?

(Multi-colored) k-Clique

- Input:
  - $G = (V = V_1 \cup \cdots \cup V_k, E)$
- Output:
  - $\exists v_1 \in V_1, \ldots, v_k \in V_k$ which form a clique?

FPT $\subseteq$ W[1]

Constraint Satisfaction Problem

2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
  - X: a set of variables
  - Σ: the domain of each variable
  - Φ: a set of 2-ary constraints
- Output:
  - $\exists \sigma: X \to \Sigma$ satisfying all constraints?

CSP Value

max. fraction of constraints satisfiable by some $\sigma: X \to \Sigma$

Constraint Satisfaction Problem

2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
  - X: a set of variables
  - Σ: the domain of each variable
  - Φ: a set of 2-ary constraints
- Output:
  - $\exists \sigma: X \to \Sigma$ satisfying all constraints?

CSP Value

max. fraction of constraints satisfiable by some $\sigma: X \to \Sigma$

- Let $n = |\Pi|$,
  - 2-CSP is NP-Complete (e.g. from 3-Coloring)
    - no $n^{O(1)}$ time algorithm assuming NP≠P

Constraint Satisfaction Problem

2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
  - X: a set of variables
  - Σ: the domain of each variable
  - Φ: a set of 2-ary constraints
- Output:
  - $\exists \sigma: X \to \Sigma$ satisfying all constraints?

CSP Value

max. fraction of constraints satisfiable by some $\sigma: X \to \Sigma$

- Let $n = |\Pi|$,
  - 2-CSP is NP-Complete (e.g. from 3-Coloring)
    - no $n^{O(1)}$ time algorithm assuming NP≠P
  - PCP Theorem:
    - no $n^{O(1)}$ time algorithm for (1 vs 0.9) gap 2-CSP assuming NP≠P

Parameterized 2-CSP

2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
  - X: a set of variables
  - Σ: the domain of each variable
  - Φ: a set of 2-ary constraints
- Output:
  - $\exists \sigma: X \to \Sigma$ satisfying all constraints?

CSP Value
max. fraction of constraints satisfiable by some $\sigma: X \to \Sigma$

- Let $k = |X|, n = |\Sigma|$,
  - Parameterized 2-CSP is W[1]-Complete (e.g. from Multi-colored k-Clique)
    - no $f(k) \cdot n^{O(1)}$ time algorithm assuming W[1]≠FPT

Parameterized 2-CSP

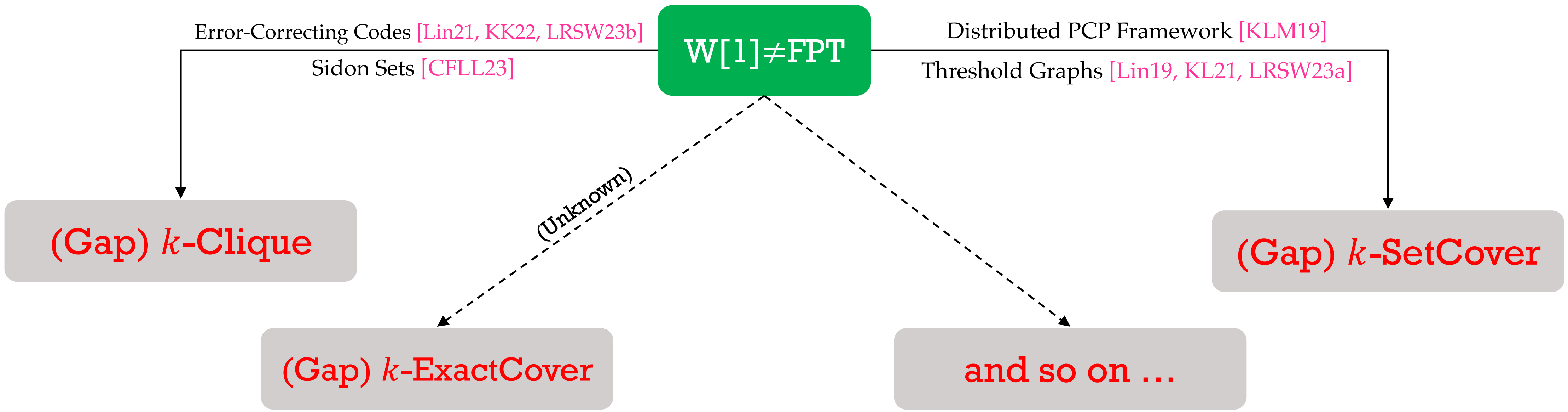
2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
  - X: a set of variables
  - Σ: the domain of each variable
  - Φ: a set of 2-ary constraints
- Output:
  - $\exists \sigma: X \to \Sigma$ satisfying all constraints?

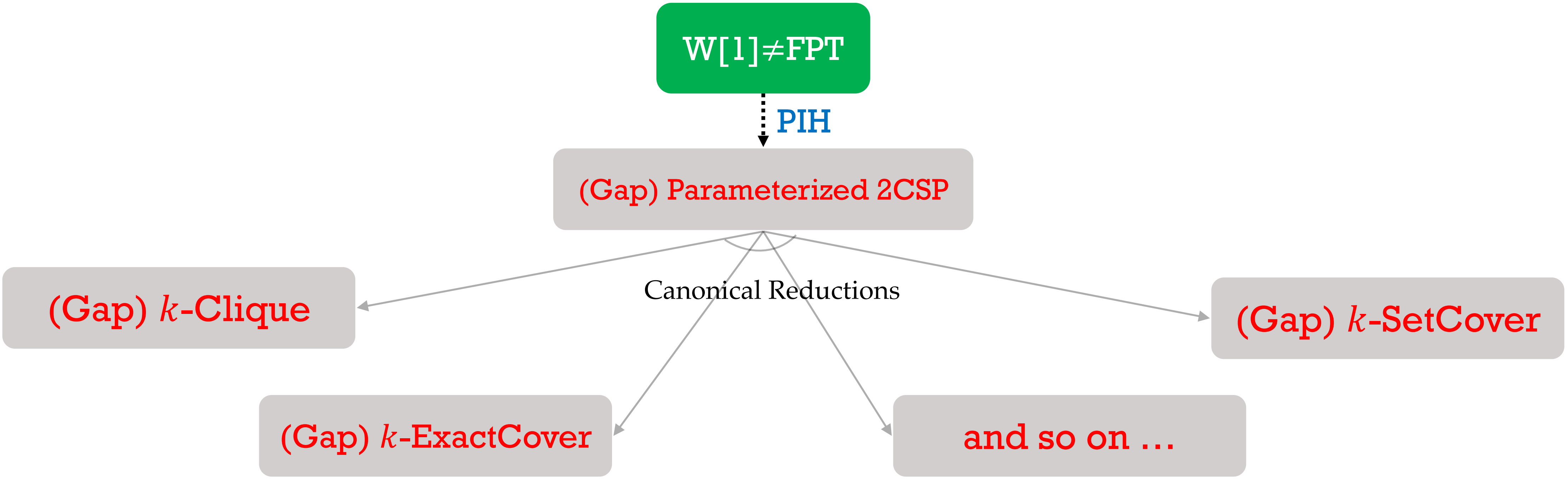
CSP Value
max. fraction of constraints satisfiable by some $\sigma: X \to \Sigma$

- Let $k = |X|, n = |\Sigma|$,
  - Parameterized 2-CSP is W[1]-Complete (e.g. from Multi-colored k-Clique)
    - no $f(k) \cdot n^{O(1)}$ time algorithm assuming W[1]≠FPT
  - PIH (Parameterized Inapproximability Hypothesis) [LRSZ20]:
    - no $f(k) \cdot n^{O(1)}$ time algorithm for (1 vs 0.9) gap version assuming W[1]≠FPT
    - Parameterized analog of the PCP theorem!

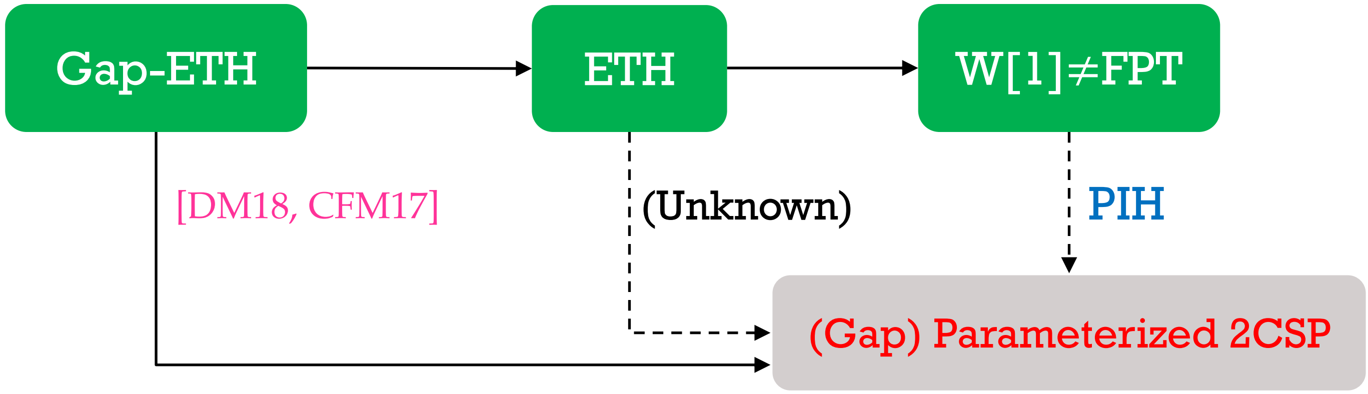
Parameterized Inapproximability Hypothesis



Parameterized Inapproximability Hypothesis



Parameterized Inapproximability Hypothesis



Outline

- Background
  - Parameterized Complexity
  - Constraint Satisfaction Problem (CSP)
  - Parameterized Inapproximability Hypothesis (PIH)
- Our Result
  - Baby PIH
- Proof Overview

List Satisfiability of CSP

2-CSP
- Input: $\Pi = (X, \Sigma, \Phi)$
- Output:
  - $\exists$ multi-assignment $\sigma: X \to 2^{\Sigma}$ list-satisfying all constraints?

List Value
max. list size of a list-satisfying multi-assignment $\sigma: X \to 2^{\Sigma}$

- A constraint on (x, y) is list-satisfied iff $\exists u \in \sigma(x), v \in \sigma(y)$, s.t. (u, v) satisfies this constraint.

List Satisfiability of CSP

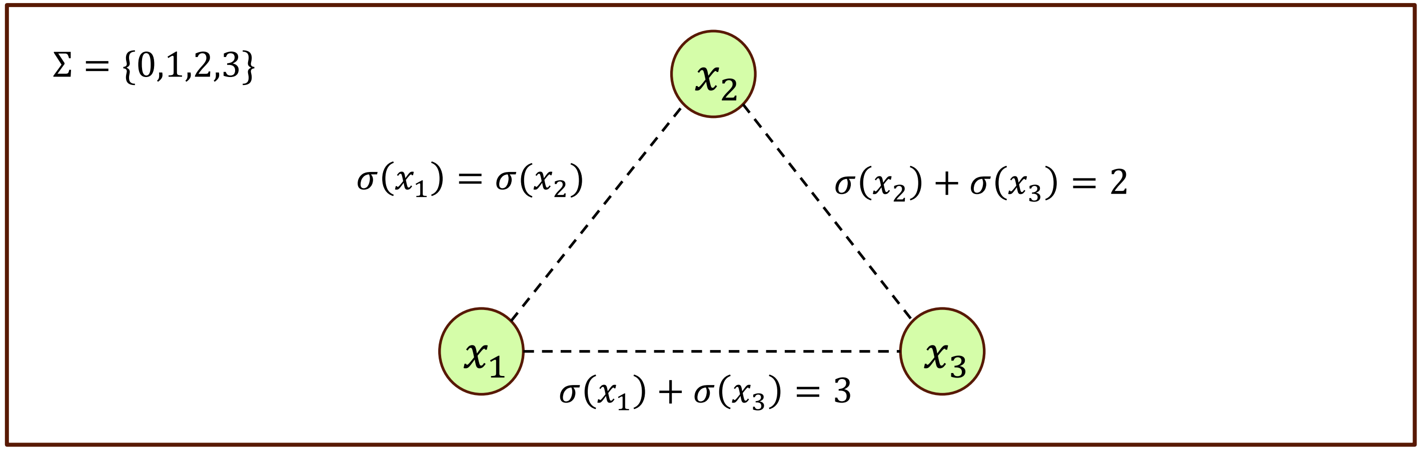
2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
- Output:
  - $\exists$ multi-assignment $\sigma: X \to 2^{\Sigma}$ list-satisfying all constraints?

List Value

max. list size of a list-satisfying multi-assignment $\sigma: X \to 2^{\Sigma}$

- A constraint on (x, y) is list-satisfied iff $\exists u \in \sigma(x), v \in \sigma(y)$, s.t. (u, v) satisfies this constraint.



List Satisfiability of CSP

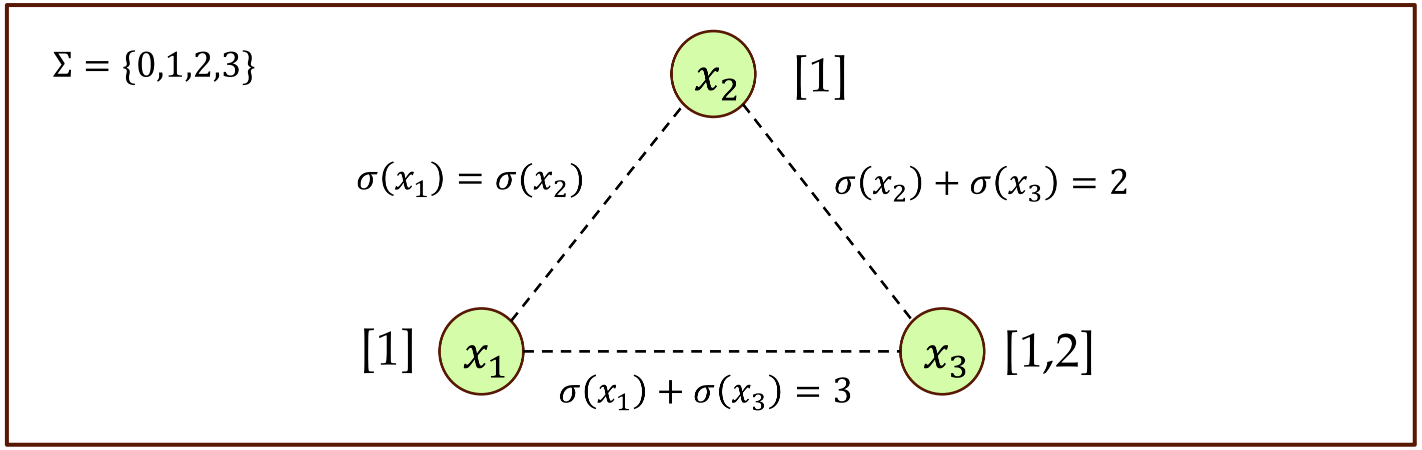
2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
- Output:
  - $\exists$ multi-assignment $\sigma: X \to 2^{\Sigma}$ list-satisfying all constraints?

List Value

max. list size of a list-satisfying multi-assignment $\sigma: X \to 2^{\Sigma}$

- A constraint on (x, y) is list-satisfied iff $\exists u \in \sigma(x), v \in \sigma(y)$, s.t. (u, v) satisfies this constraint.



List Satisfiability of CSP

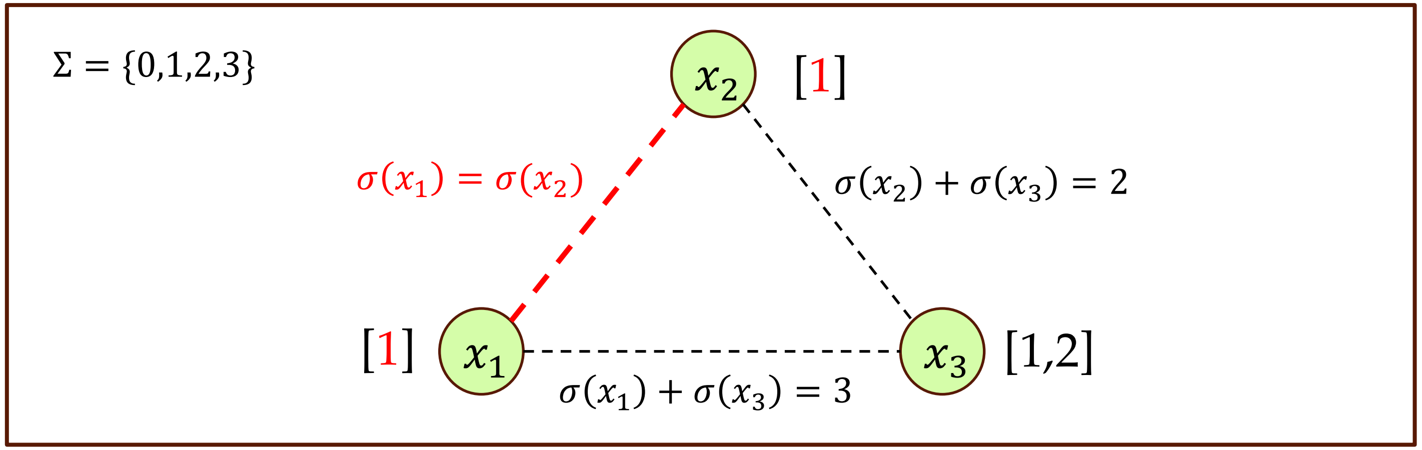
2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
- Output:
  - $\exists$ multi-assignment $\sigma: X \to 2^{\Sigma}$ list-satisfying all constraints?

List Value

max. list size of a list-satisfying multi-assignment $\sigma: X \to 2^{\Sigma}$

- A constraint on (x, y) is list-satisfied iff $\exists u \in \sigma(x), v \in \sigma(y)$, s.t. (u, v) satisfies this constraint.



List Satisfiability of CSP

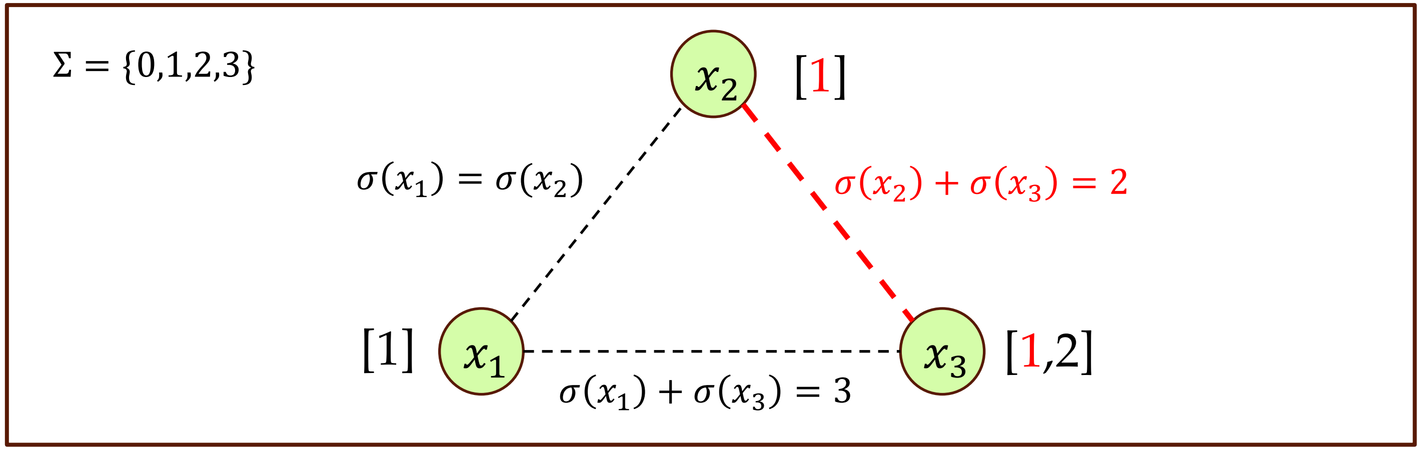
2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
- Output:
  - $\exists$ multi-assignment $\sigma: X \to 2^{\Sigma}$ list-satisfying all constraints?

List Value

max. list size of a list-satisfying multi-assignment $\sigma: X \to 2^{\Sigma}$

- A constraint on (x, y) is list-satisfied iff $\exists u \in \sigma(x), v \in \sigma(y)$, s.t. (u, v) satisfies this constraint.



List Satisfiability of CSP

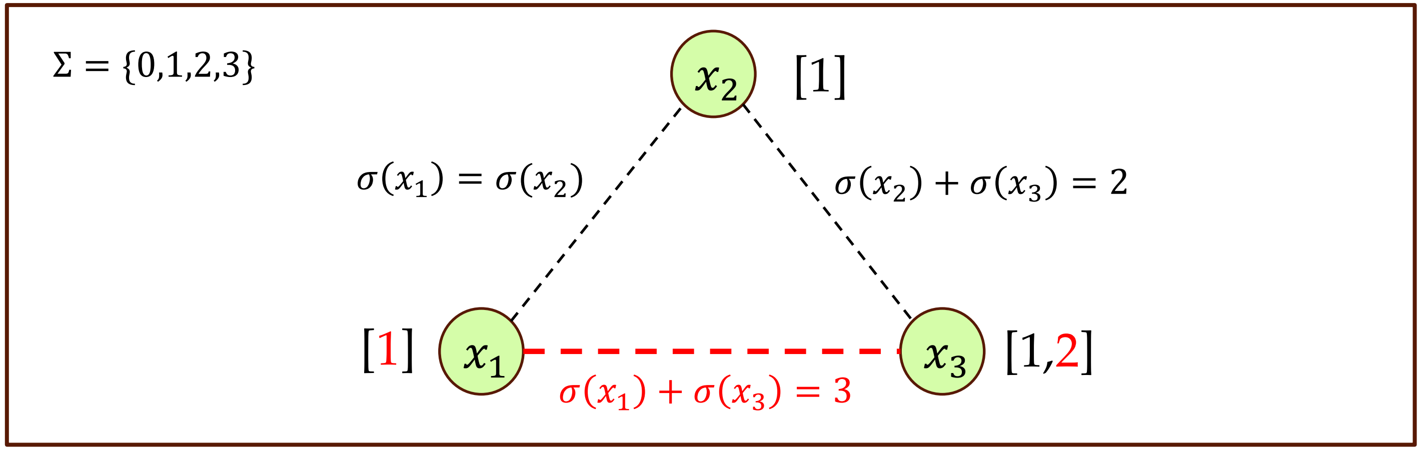
2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
- Output:
  - $\exists$ multi-assignment $\sigma: X \to 2^{\Sigma}$ list-satisfying all constraints?

List Value

max. list size of a list-satisfying multi-assignment $\sigma: X \to 2^{\Sigma}$

- A constraint on (x, y) is list-satisfied iff $\exists u \in \sigma(x), v \in \sigma(y)$, s.t. (u, v) satisfies this constraint.



List Satisfiability of CSP

2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
- Output:
  - $\exists$ multi-assignment $\sigma: X \to 2^{\Sigma}$ list-satisfying all constraints?

List Value

max. list size of a list-satisfying multi-assignment $\sigma: X \to 2^{\Sigma}$

- We say a 2-CSP is r-list satisfiable iff $\exists \sigma$ with $\max_{x \in X} |\sigma(x)| \le r$ list-satisfying all constraints.

List Satisfiability of CSP

2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
- Output:
  - $\exists$ multi-assignment $\sigma: X \to 2^{\Sigma}$ list-satisfying all constraints?

List Value

max. list size of a list-satisfying multi-assignment $\sigma: X \to 2^{\Sigma}$

- We say a 2-CSP is r-list satisfiable iff $\exists \sigma$ with $\max_{x \in X} |\sigma(x)| \leq r$ list-satisfying all constraints.
  - CSP Value=1 $\Leftrightarrow$ 1-list satisfiable $\Rightarrow$ r-list satisfiable for $r \geq 2$
  - r-list satisfiable $\Rightarrow$ CSP Value $\geq 1/r^2$

Baby PCP

2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
- Output:
  - $\exists$ multi-assignment $\sigma: X \to 2^{\Sigma}$ list-satisfying all constraints?

List Value

max. list size of a list-satisfying multi-assignment $\sigma: X \to 2^{\Sigma}$

- We say a 2-CSP is r-list satisfiable iff $\exists \sigma$ with $\max_{x \in X} |\sigma(x)| \leq r$ list-satisfying all constraints.
  - CSP Value=1 $\Leftrightarrow$ 1-list satisfiable $\Rightarrow$ r-list satisfiable for $r \geq 2$
  - r-list satisfiable $\Rightarrow$ CSP Value $\geq 1/r^2$

- Baby PCP [Barto-Kozik'22]
  - For any $r > 1$, It's NP-hard to distinguish between [1-list satisfiable] and [not even r-list satisfiable].

Baby PCP

2-CSP

- Input: $\Pi = (X, \Sigma, \Phi)$
- Output:
  - $\exists$ multi-assignment $\sigma: X \to 2^{\Sigma}$ list-satisfying all constraints?

List Value

max. list size of a list-satisfying multi-assignment $\sigma: X \to 2^{\Sigma}$

- We say a 2-CSP is r-list satisfiable iff $\exists \sigma$ with $\max_{x \in X} |\sigma(x)| \leq r$ list-satisfying all constraints.
  - CSP Value=1 $\Leftrightarrow$ 1-list satisfiable $\Rightarrow$ r-list satisfiable for $r \geq 2$
  - r-list satisfiable $\Rightarrow$ CSP Value $\geq 1/r^2$

- Baby PCP [Barto-Kozik'22]
  - For any $r > 1$, It's NP-hard to distinguish between [1-list satisfiable] and [not even r-list satisfiable].
- $\Leftarrow$ PCP $\quad\Uparrow\quad\quad \Uparrow$ (when $\varepsilon < 1/r^2$)
  - For any $\varepsilon > 0$, It's NP-hard to distinguish between [CSP Value $=1$] and [CSP Value $<\varepsilon$].

Baby PCP

- Baby PCP [Barto-Kozik'22]
  - Assuming NP≠P, for any $r > 1$, distinguishing between [1-list satisfiable] and [not even r-list satisfiable] cannot be done in $|\Pi|^{O(1)}$ time.
  - (A combinatorial proof)
  - (Enough to prove the NP-hardness of some PCSPs (e.g., $(2+\varepsilon)$-SAT))

Baby PIH

- Baby PCP [Barto-Kozik'22]
  - Assuming NP≠P, for any $r > 1$, distinguishing between [1-list satisfiable] and [not even r-list satisfiable] cannot be done in $|\Pi|^{O(1)}$ time.
  - (A combinatorial proof)
  - (Enough to prove the NP-hardness of some PCSPs (e.g., $(2+\varepsilon)$-SAT))

- Baby PIH [This work]
  - Assuming W[1]≠FPT, … cannot be done in $f(|X|) \cdot |\Sigma|^{O(1)}$ time.
  - (An itself interesting inapproximability result for list-satisfiability of CSP)
  - (A step towards PIH)
  - (Enough to get some applications of PIH?)

Baby PIH

- Baby PCP [Barto-Kozik'22]
  - Assuming NP≠P, for any $r > 1$, distinguishing between [1-list satisfiable] and [not even r-list satisfiable] cannot be done in $|\Pi|^{O(1)}$ time.
  - (A combinatorial proof)
  - (Enough to prove the NP-hardness of some PCSPs (e.g., $(2+\varepsilon)$-SAT))

- Baby PIH [This work]
  - Assuming W[1]≠FPT, … cannot be done in $f(|X|) \cdot |\Sigma|^{O(1)}$ time.
  - (An itself interesting inapproximability result for list-satisfiability of CSP)
  - (A step towards PIH)
  - (Enough to get some applications of PIH?)
    - not sure…, but something stronger is enough!
    - PIH ⇒ Average Baby PIH ⇒ Baby PIH

Outline

- Background
  - Parameterized Complexity
  - Constraint Satisfaction Problem (CSP)
  - Parameterized Inapproximability Hypothesis (PIH)
- Our Result
  - Baby PIH
- Proof Overview

Proof Overview

- Follows from and extends [Barto-Kozik'22]'s proof of Baby PCP Theorem
- Direct Product Construction

2-CSP

$\Pi = (X, \Sigma, \Phi)$

→

***t*-wise Direct Product 2-CSP**

$$\Pi^{\odot t} = \left(\binom{X}{t}, \Sigma^t, \Phi'\right)$$

partial satisfying assignments
for the set of variables

Proof Overview

- Follows from and extends [Barto-Kozik'22]'s proof of Baby PCP Theorem
- Direct Product Construction

2-CSP

$\Pi = (X, \Sigma, \Phi)$

$\longrightarrow$

t-wise Direct Product 2-CSP

$$\Pi^{\odot t} = \left(\binom{X}{t}, \Sigma^t, \Phi'\right)$$

consistency checks

Proof Overview

- Follows from and extends [Barto-Kozik'22]'s proof of Baby PCP Theorem
- Direct Product Construction

2-CSP
$\Pi = (X, \Sigma, \Phi)$

$\longrightarrow$

***t*-wise Direct Product 2-CSP**
$\Pi^{\odot t} = \left(\binom{X}{t}, \Sigma^t, \Phi'\right)$

partial satisfying assignments for the set of variables

consistency checks

- (Want to show):
- For any $r > 1$, there exists t depending on r, such that for every Π,
  - (Completeness) If Π is satisfiable, then so is $\Pi^{\odot t}$.
  - (Soundness) If Π is not satisfiable, then $\Pi^{\odot t}$ is not r-list satisfiable.

Proof Overview

- Follows from and extends [Barto-Kozik'22]'s proof of Baby PCP Theorem
- Direct Product Construction

2-CSP
$\Pi = (X, \Sigma, \Phi)$

→

t-wise Direct Product 2-CSP
$\Pi^{\odot t} = \left(\binom{X}{t}, \Sigma^t, \Phi'\right)$

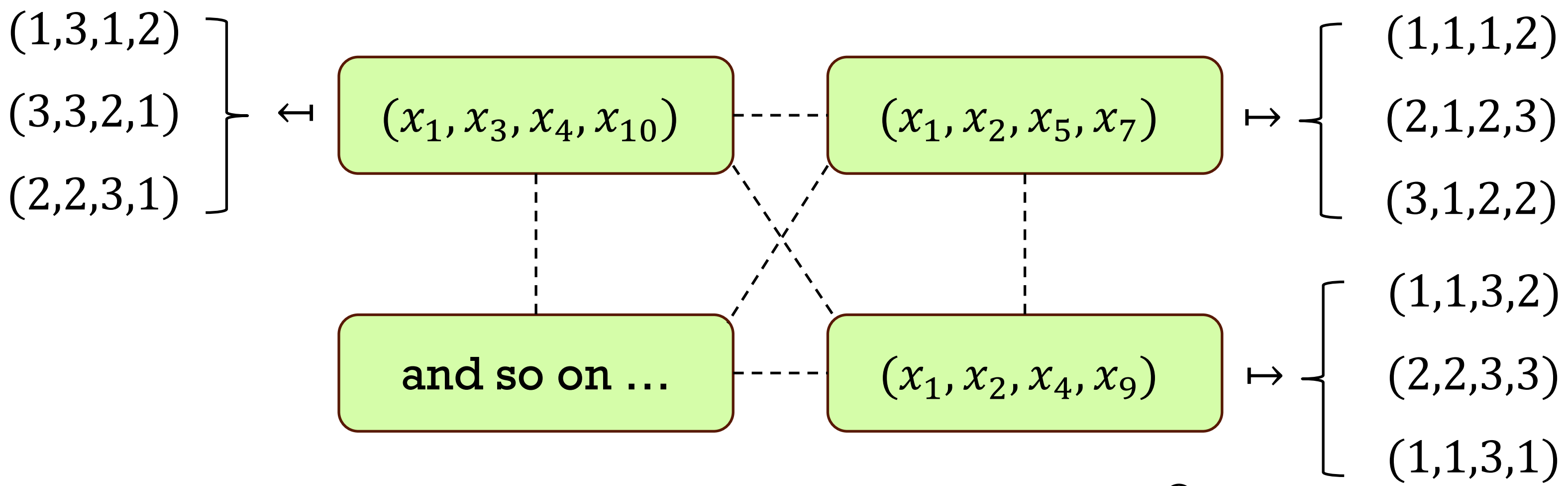
partial satisfying assignments for the set of variables

consistency checks

- (Want to show):
- For any $r > 1$, there exists t depending on r, such that for every Π,
  - (Completeness) If Π is satisfiable, then so is $\Pi^{\odot t}$.
  - (Soundness) If Π is not satisfiable, then $\Pi^{\odot t}$ is not r-list satisfiable.

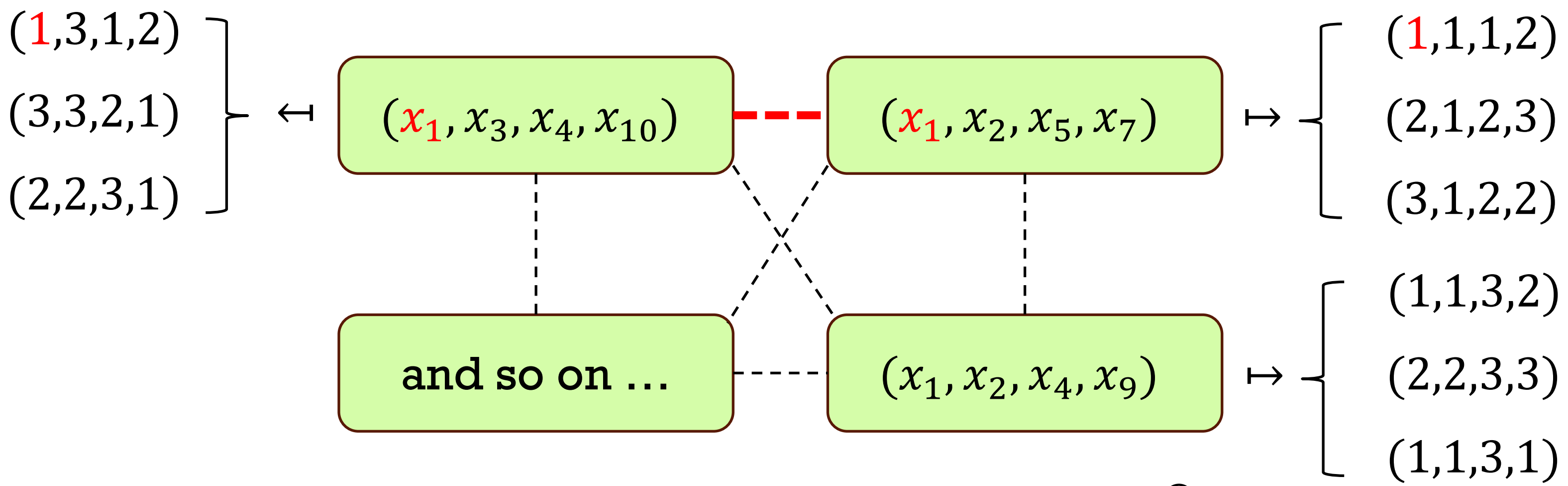
- Reduction time: $n^{O_r(1)}$ where $n = |\Pi|$
  - a unified proof for both Baby PCP and Baby PIH!

Proof Overview



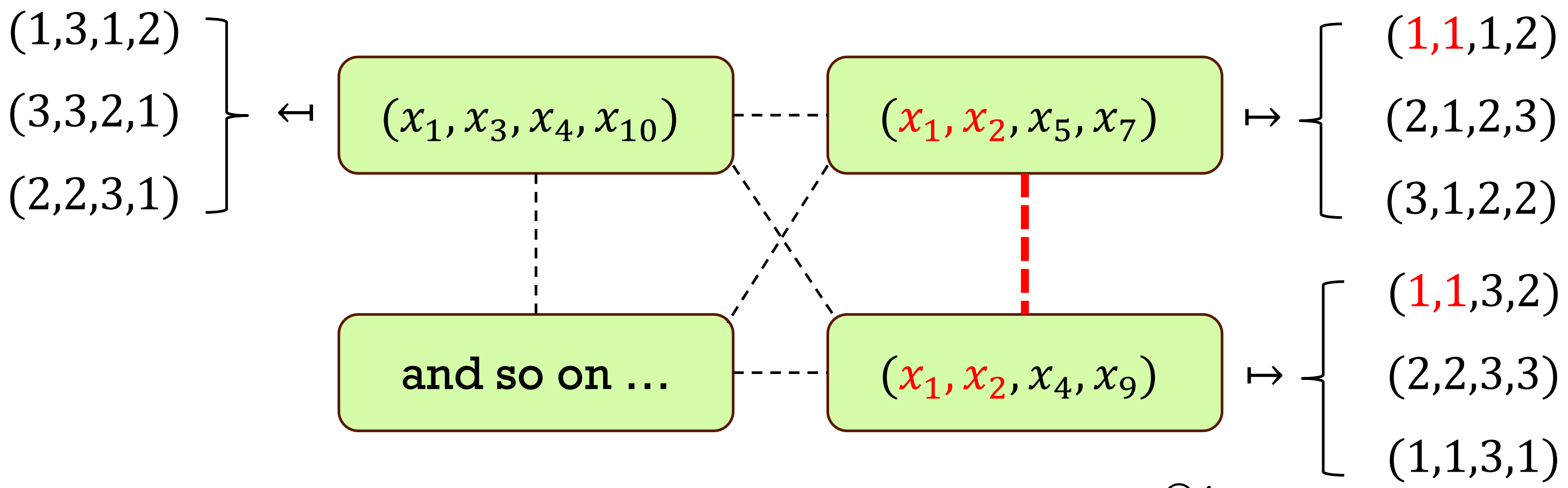
A 3-list satisfying assignment for $\Pi^{\odot 4}$

Proof Overview



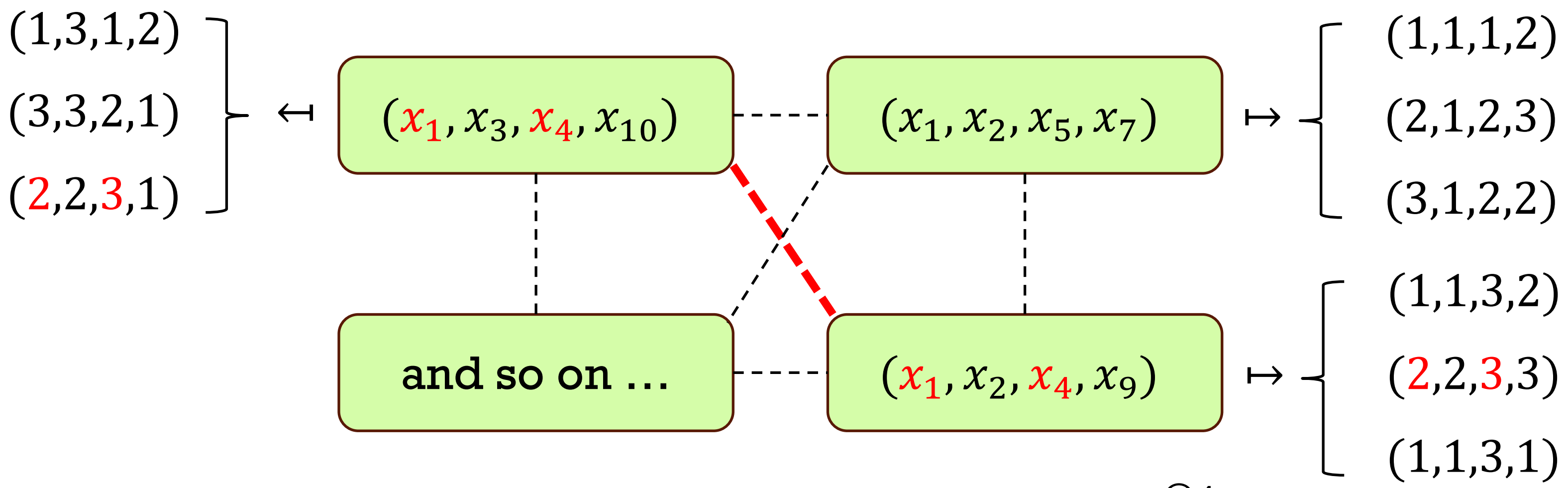
A 3-list satisfying assignment for $\Pi^{\odot 4}$

Proof Overview



A 3-list satisfying assignment for $\Pi^{\odot 4}$

Proof Overview



A 3-list satisfying assignment for $\Pi^{\odot 4}$

Proof Overview

2-CSP
$\Pi = (X, \Sigma, \Phi)$

$\longrightarrow$

t-wise Direct Product 2-CSP
$\Pi^{\odot t} = \left(\binom{X}{t}, \Sigma^t, \Phi'\right)$

- For some sufficiently large $t = t(r)$,
  - given an r-list satisfying multi-assignment σ of $\Pi^{\odot t}$,
  - want to construct an $(r-1)$-list satisfying multi-assignment σ' of $\Pi^{\odot t'}$, for some $t' < t$.

Proof Overview

2-CSP
$\Pi = (X, \Sigma, \Phi)$

$\longrightarrow$

t-wise Direct Product 2-CSP
$\Pi^{\odot t} = \left(\binom{X}{t}, \Sigma^t, \Phi'\right)$

- For some sufficiently large $t = t(r)$,
  - given an r-list satisfying multi-assignment σ of $\Pi^{\odot t}$,
  - want to construct an $(r-1)$-list satisfying multi-assignment σ' of $\Pi^{\odot t'}$, for some $t' < t$.

- If we end up with the 1-list satisfiability of $\Pi^{\odot(\geq 2)}$, then we are done!

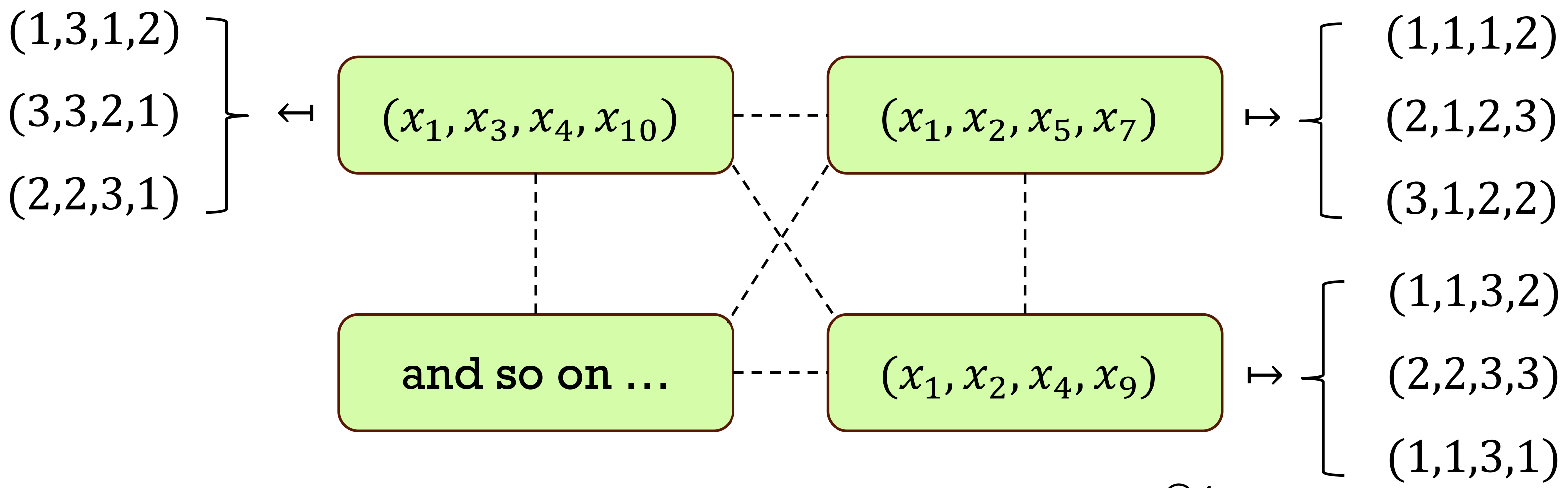
Proof Overview

2-CSP		t-wise Direct Product 2-CSP
$\Pi = (X, \Sigma, \Phi)$	$\longrightarrow$	$\Pi^{\odot t} = \left(\binom{X}{t}, \Sigma^t, \Phi'\right)$

- For some sufficiently large $t = t(r)$,
  - given an r-list satisfying multi-assignment σ of $\Pi^{\odot t}$,
  - want to construct an $(r-1)$-list satisfying multi-assignment σ' of $\Pi^{\odot t'}$, for some $t' < t$.
    - for each set $S \in \binom{X}{t'}$, choose a set $T \in \binom{X}{t}$ with $S \subseteq T$
    - the list $\sigma'(S)$ is inherited from the list $\sigma(T)$ (at the hope of decreasing the list size by 1)

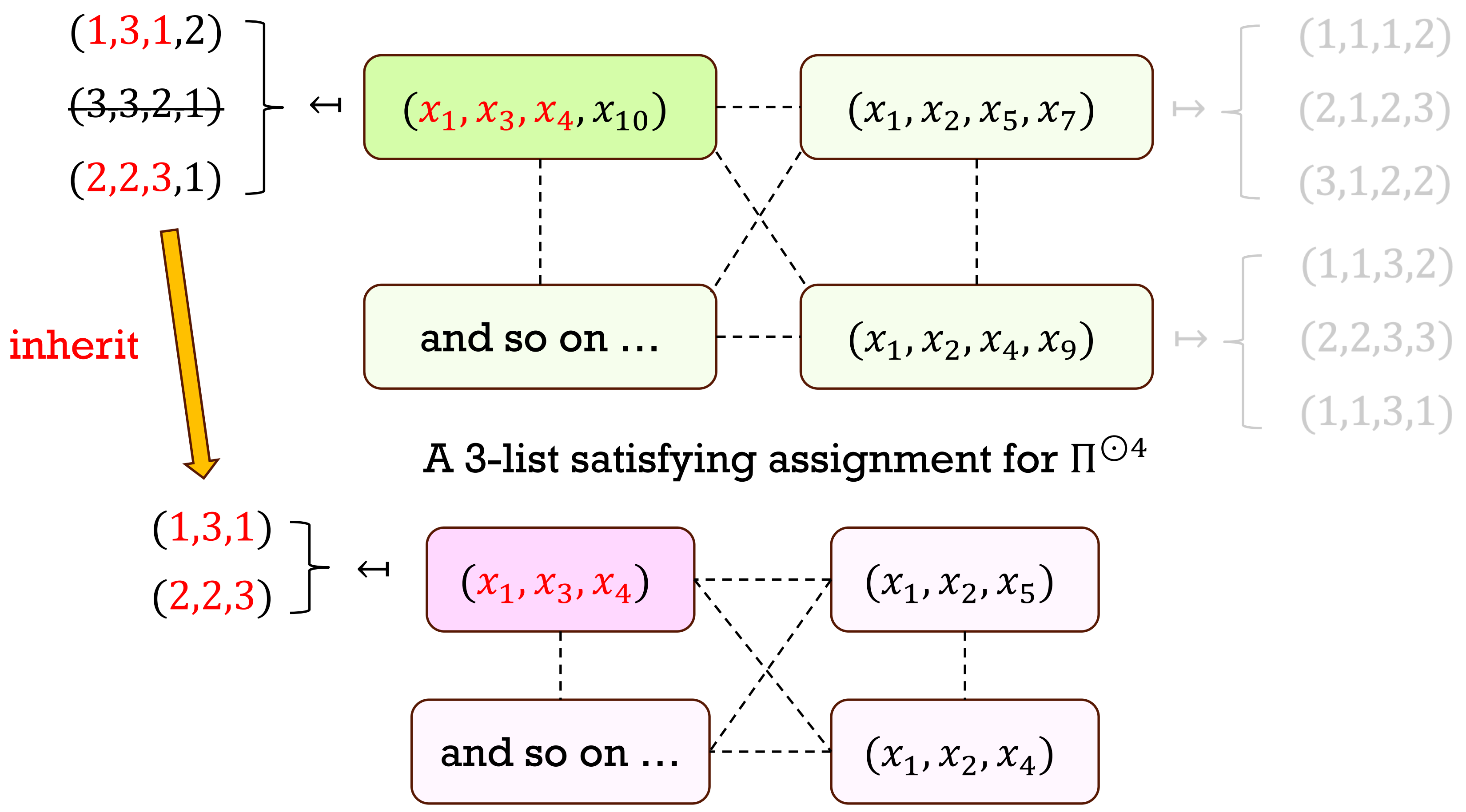
- If we end up with the 1-list satisfiability of $\Pi^{\odot(\geq 2)}$, then we are done!

Proof Overview

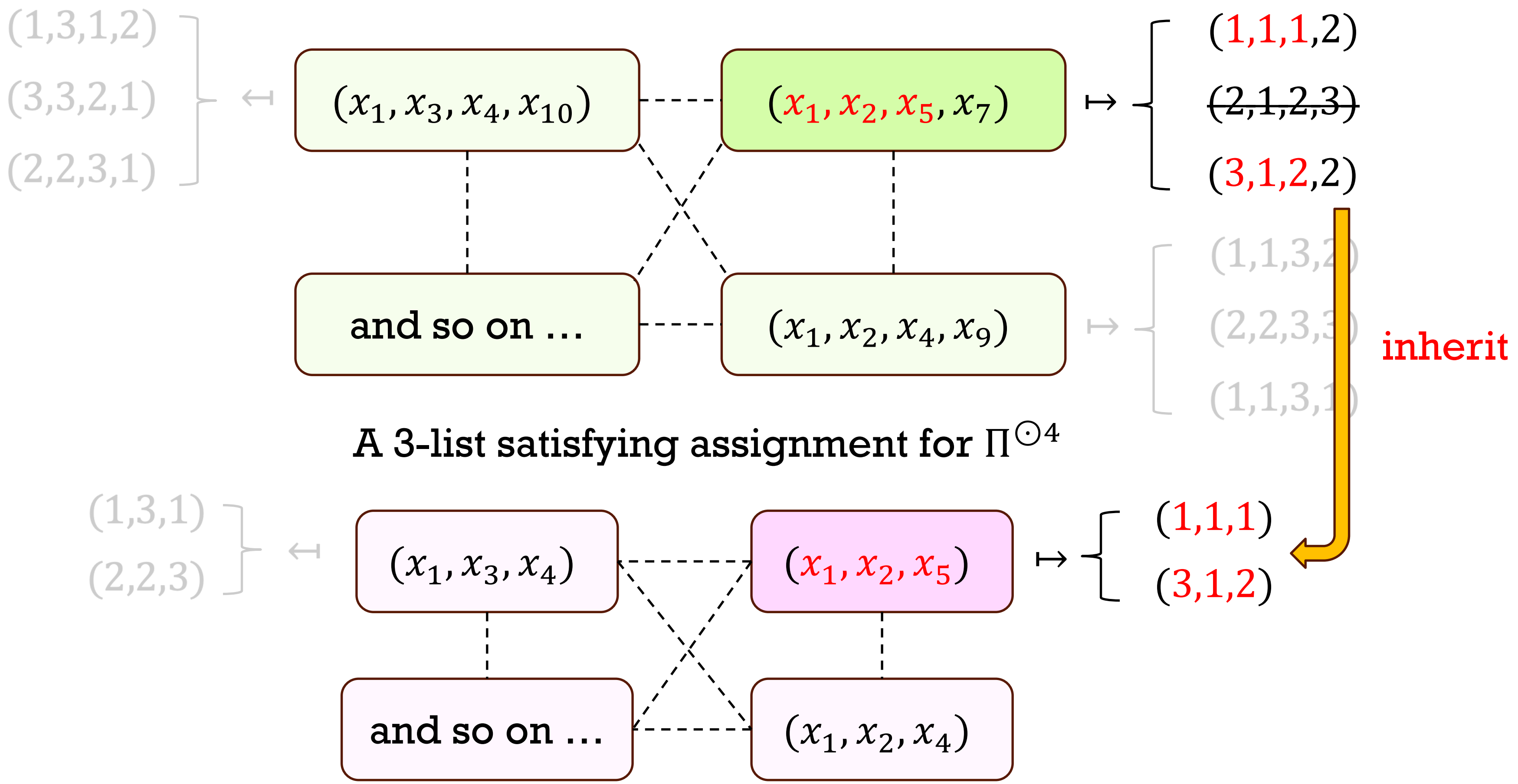


A 3-list satisfying assignment for $\Pi^{\odot 4}$

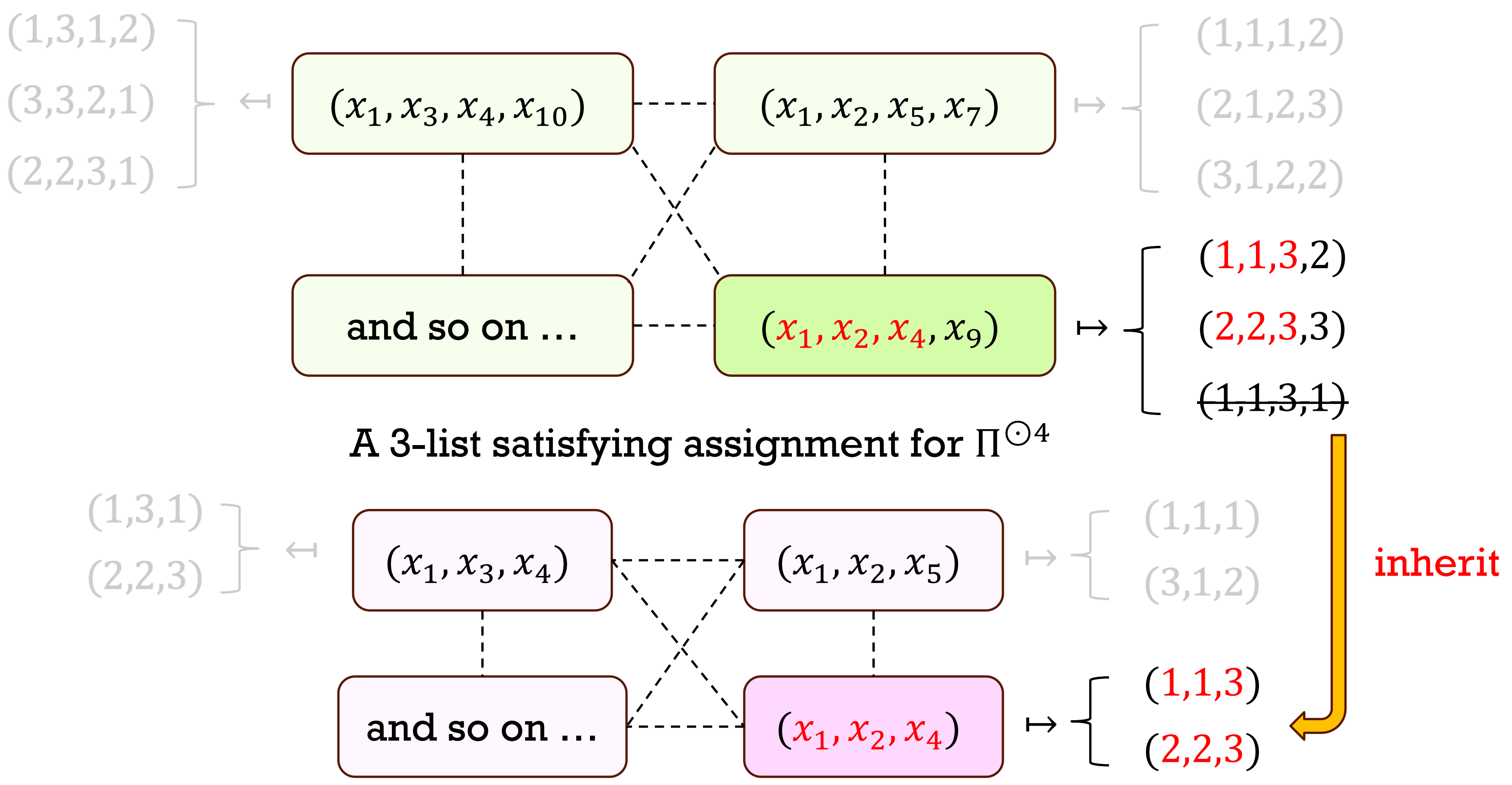
Proof Overview



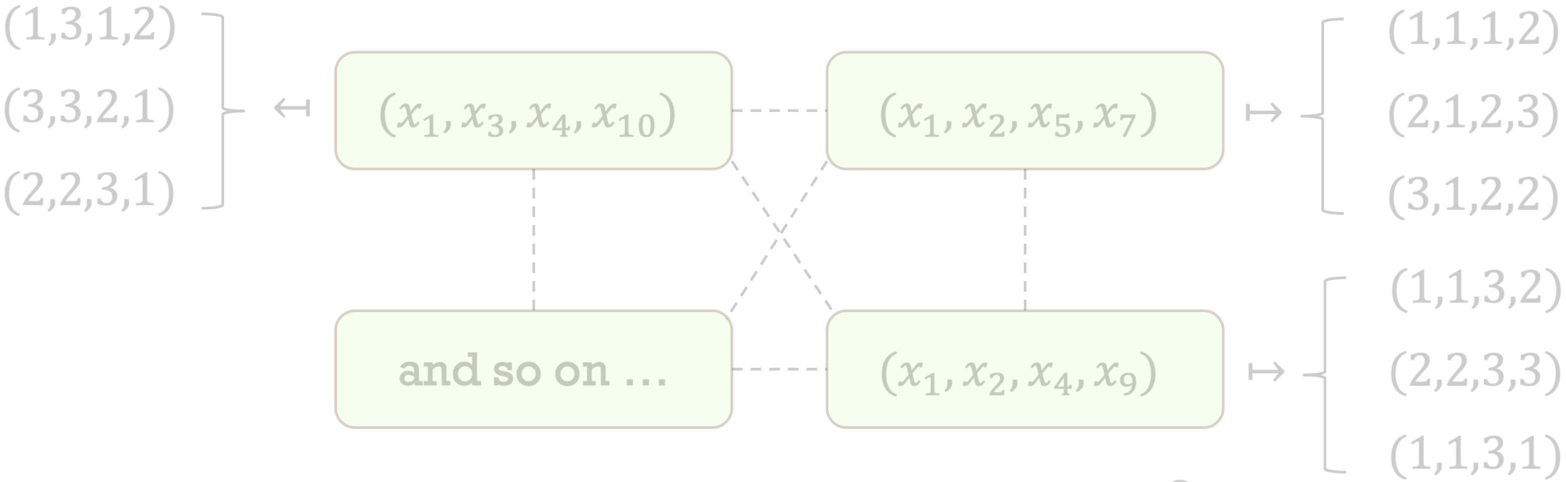
Proof Overview



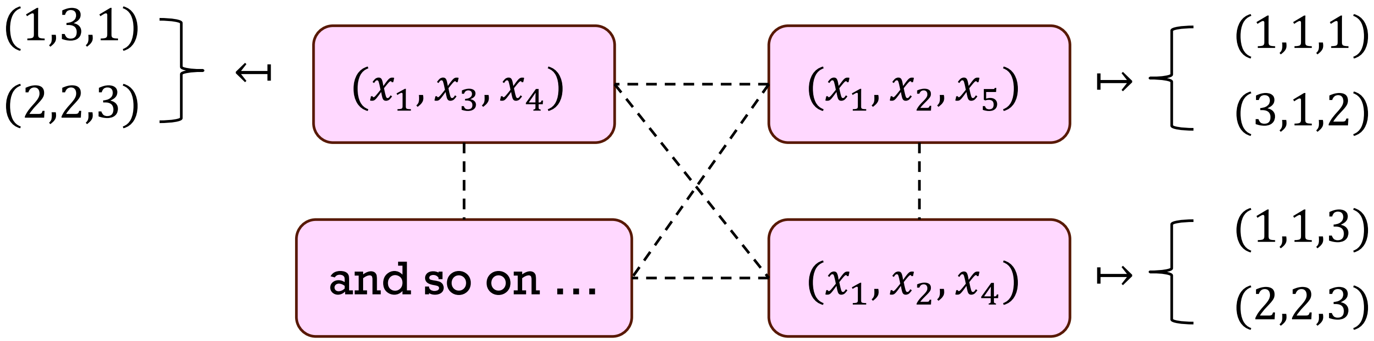
Proof Overview



Proof Overview



A 3-list satisfying assignment for $\Pi^{\odot 4}$



A 2-list satisfying assignment for $\Pi^{\odot 3}$?

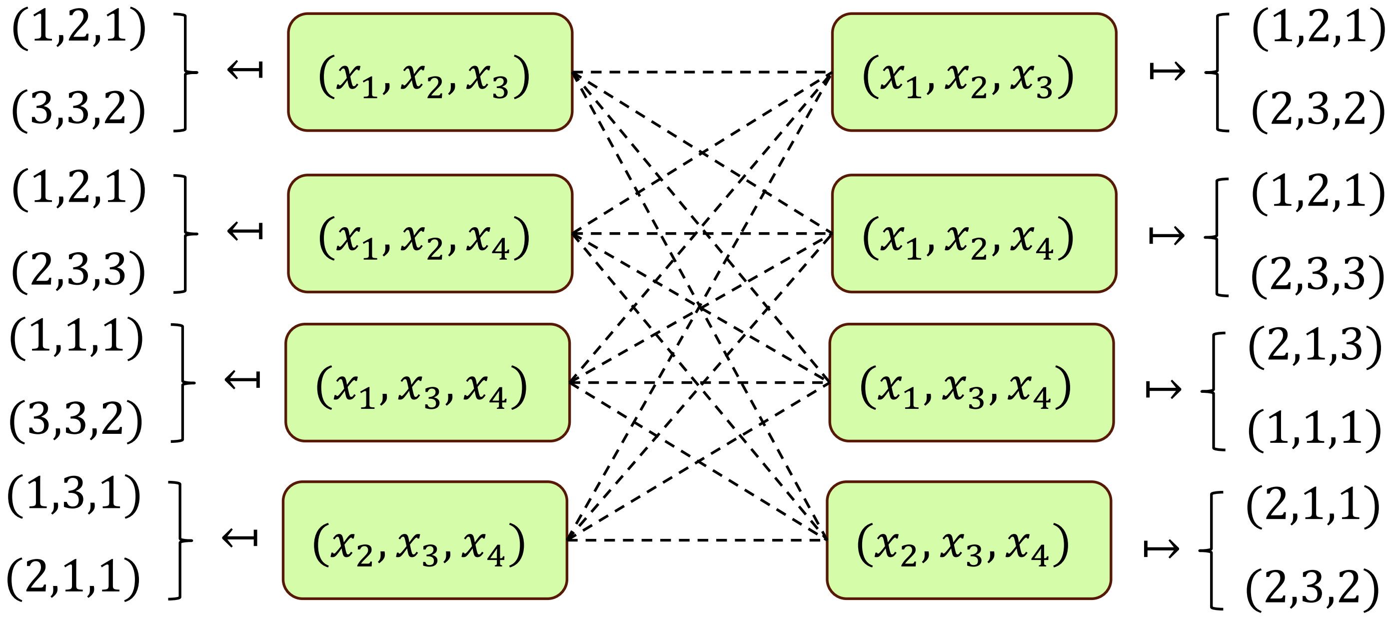
Proof Overview

- How can we discard one assignment safely?
  - the one that is never used to meet any consistency constraints!

Proof Overview

- How can we discard one assignment safely?
  - the one that is never used to meet any consistency constraints!

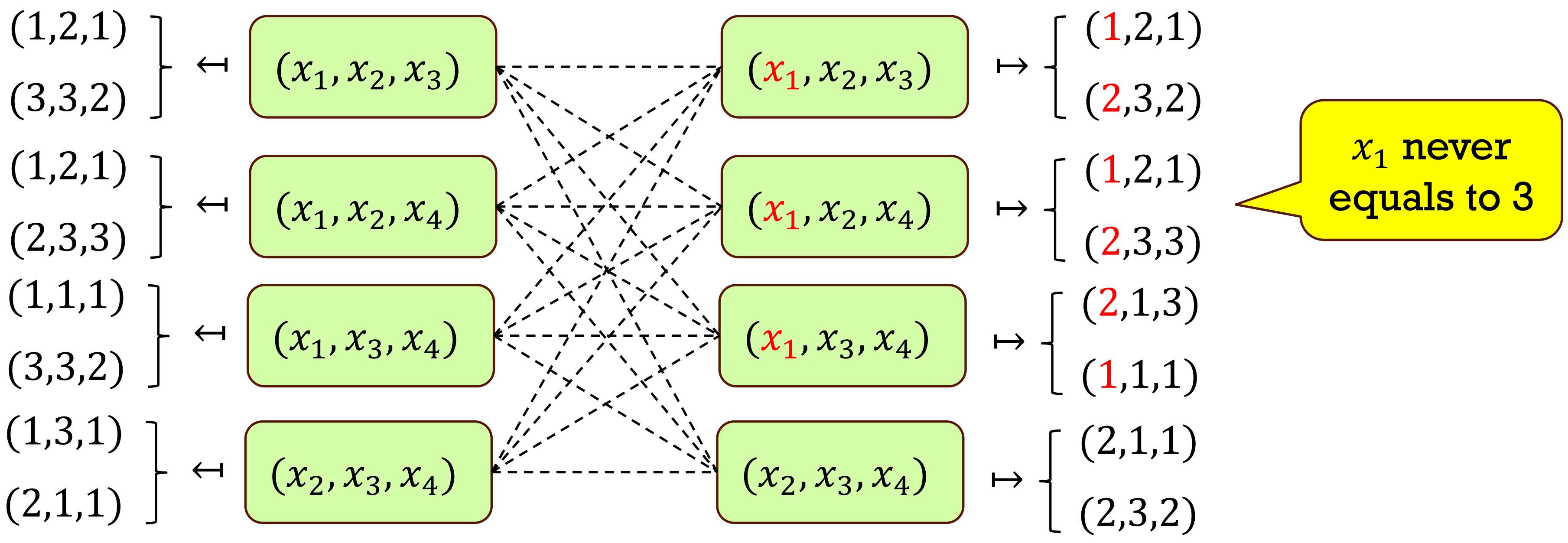
- Suppose we have the following *bipartite* direct product instance:



Proof Overview

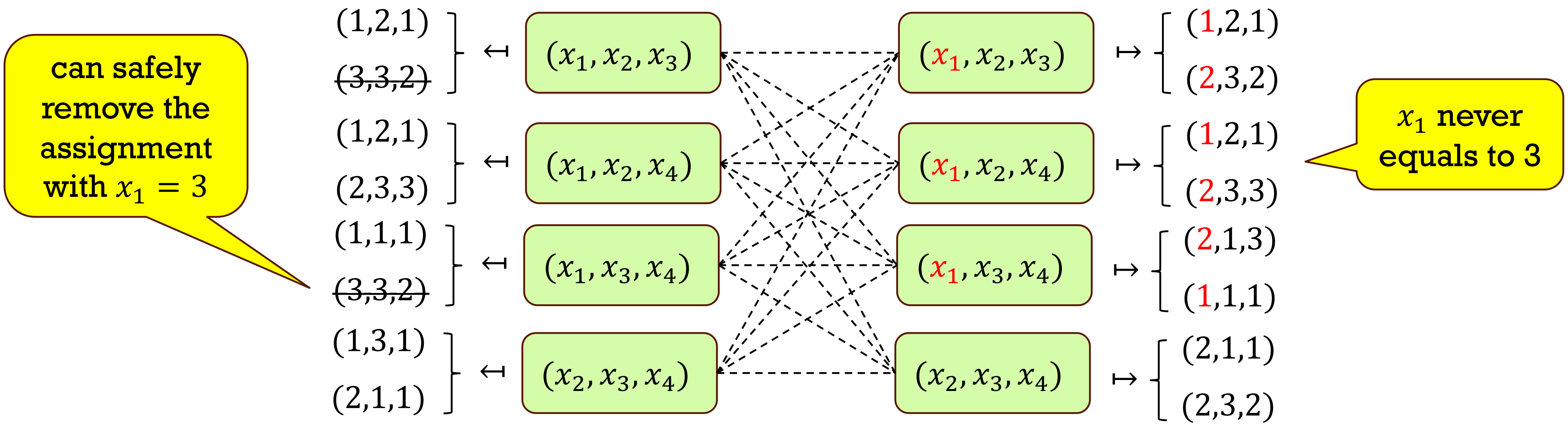
- How can we discard one assignment safely?
  - the one that is never used to meet any consistency constraints!

- Suppose we have the following *bipartite* direct product instance:



Proof Overview

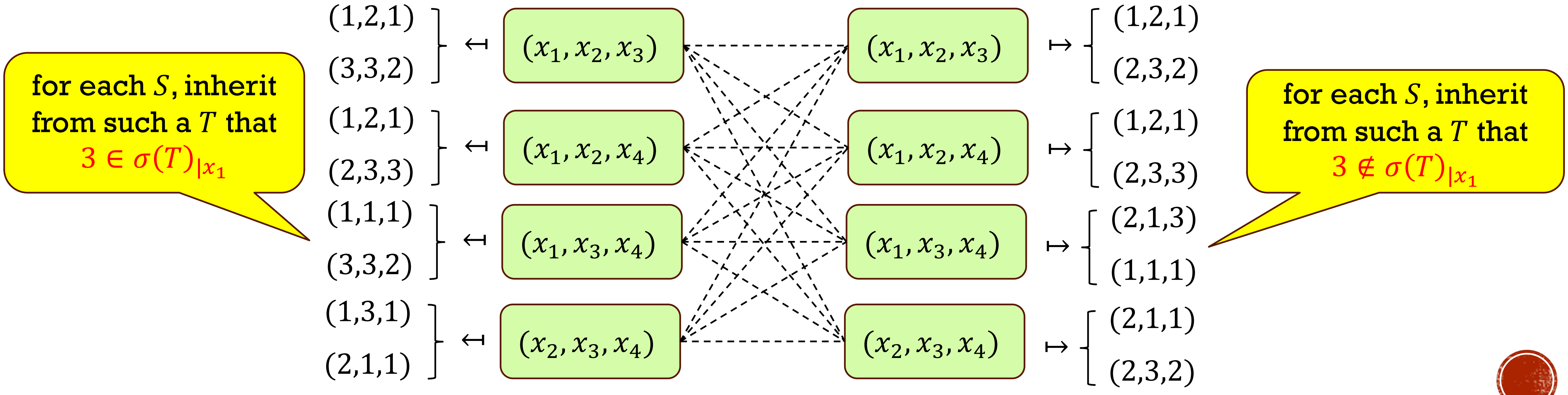
- How can we discard one assignment safely?
  - the one that is never used to meet any consistency constraints!
- Suppose we have the following *bipartite* direct product instance:



Proof Overview

- How can we discard one assignment safely?
  - the one that is never used to meet any consistency constraints!

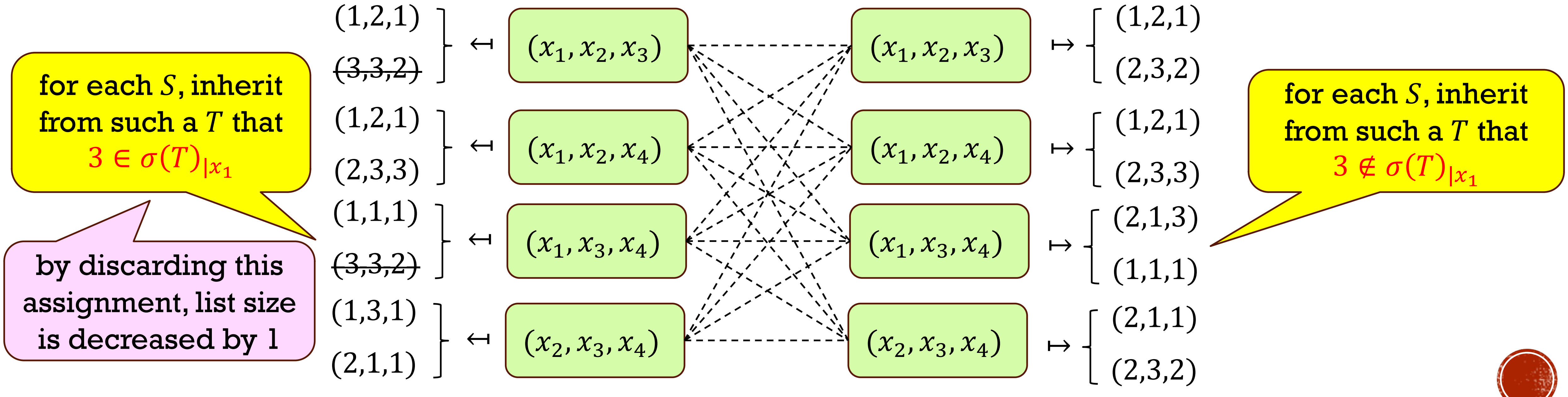
- Suppose we have the following *bipartite* direct product instance:



Proof Overview

- How can we discard one assignment safely?
  - the one that is never used to meet any consistency constraints!

- Suppose we have the following *bipartite* direct product instance:



Proof Overview

- Bipartite $(r, 1)$-case

$\uparrow$

- Bipartite (r, q)-case

$\uparrow$

- Non-bipartite r-case

Proof Overview

- Bipartite $(r, 1)$-case

$\uparrow$

- Bipartite (r, q)-case

$\uparrow$

- Non-bipartite r-case

Takeaway

- Parameterized Inapproximability Hypothesis – parameterized analog of PCP
- Baby PIH – inapproximability of the list-satisfiability of (parameterized) 2CSP
  - W[1]-hard to distinguish between [1-list satisfiable] and [not even r-list satisfiable]
  - Proof Idea: induction on the list size

Takeaway

- Parameterized Inapproximability Hypothesis – parameterized analog of PCP
- Baby PIH – inapproximability of the list-satisfiability of (parameterized) 2CSP
  - W[1]-hard to distinguish between [1-list satisfiable] and [not even r-list satisfiable]
  - Proof Idea: induction on the list size
- Average Baby PIH?
  - $\Rightarrow$ constant inapproximability of k-ExactCover

Takeaway

- Parameterized Inapproximability Hypothesis – parameterized analog of PCP
- Baby PIH – inapproximability of the list-satisfiability of (parameterized) 2CSP
  - W[1]-hard to distinguish between [1-list satisfiable] and [not even r-list satisfiable]
  - Proof Idea: induction on the list size

- Average Baby PIH?
  - ⇒ constant inapproximability of k-ExactCover

- Thanks!