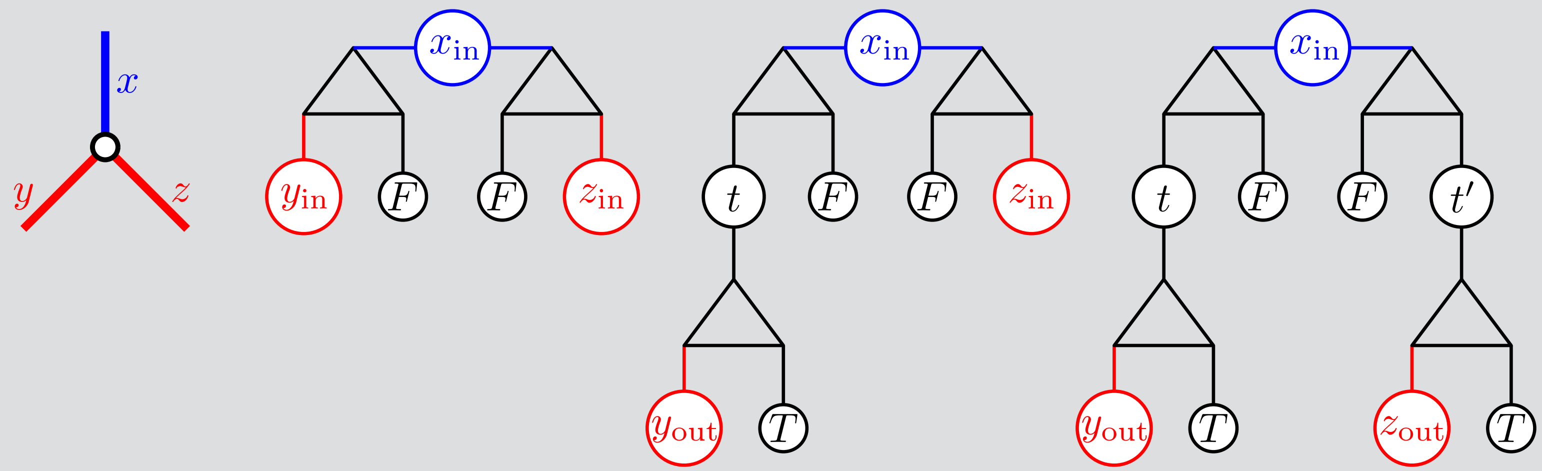
Reconfiguration of Satisfying Assignments and Subset Sums: Easy to Find, Hard to Connect



Jean Cardinal, Erik Demaine, David Eppstein, Robert Hearn, *Andrew Winslow*

ULB Université Libre de Bruxelles
UCI University of California, Irvine
Massachusetts Institute of Technology
The University of Texas Rio Grande Valley

Reconfiguration: a SAT Example

Formula: $(x_1 \lor \neg x_2) \land (\neg x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2 \lor \neg x_3)$

$x_1 = F$

$x_2 = F$ $\quad (F \lor \neg F) \land (\neg F \lor F \lor F) \land (\neg F \lor \neg F \lor \neg F)$

$x_3 = F$

Formula: $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$

$x_1 = F$

$x_2 = F$ $\quad (F \vee \neg F) \wedge (\neg F \vee F \vee F) \wedge (\neg F \vee \neg F \vee \neg F)$

$x_3 = F$

$\Downarrow$ flip x_3

$x_1 = F$

$x_2 = F$ $\quad (F \vee \neg F) \wedge (\neg F \vee F \vee T) \wedge (\neg F \vee \neg F \vee \neg T)$

$x_3 = T$

Formula: $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$

$x_1 = F$
$x_2 = F$ $\quad (F \vee \neg F) \wedge (\neg F \vee F \vee F) \wedge (\neg F \vee \neg F \vee \neg F)$
$x_3 = F$

$\Downarrow$ flip x_3

$x_1 = F$
$x_2 = F$ $\quad (F \vee \neg F) \wedge (\neg F \vee F \vee T) \wedge (\neg F \vee \neg F \vee \neg T)$
$x_3 = T$

$\Downarrow$ flip x_1

$x_1 = T$
$x_2 = F$ $\quad (T \vee \neg F) \wedge (\neg T \vee F \vee T) \wedge (\neg T \vee \neg F \vee \neg T)$
$x_3 = T$

Formula: $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$

$x_1 = F$

$x_2 = F$ $\quad (F \vee \neg F) \wedge (\neg F \vee F \vee F) \wedge (\neg F \vee \neg F \vee \neg F)$

$x_3 = F$

$\Downarrow$ flip x_3

$x_1 = F$

$x_2 = F$ $\quad (F \vee \neg F)$ Keeping formula satisfied $\vee \neg F \vee \neg T)$

$x_3 = T$

$\Downarrow$ flip x_1

$x_1 = T$

$x_2 = F$ $\quad (T \vee \neg F) \wedge (\neg T \vee F \vee T) \wedge (\neg T \vee \neg F \vee \neg T)$

$x_3 = T$

Formula: $(x_1 \lor \neg x_2) \land (\neg x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2 \lor \neg x_3)$

$x_1 = F$

$x_2 = F$ $\quad (F \lor \neg F) \land (\neg F \lor F \lor F) \land (\neg F \lor \neg F \lor \neg F)$

$x_3 = F$

$\Downarrow$

$\Downarrow$

$x_1 = T$

$x_2 = T$ $\quad (T \lor \neg T) \land (\neg T \lor T \lor F) \land (\neg T \lor \neg F \lor \neg F)$

$x_3 = F$

Formula: $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$

$x_1 = F$

$x_2 = F$ $\quad (F \vee \neg F) \wedge (\neg F \vee F \vee F) \wedge (\neg F \vee \neg F \vee \neg F)$

$x_3 = F$

$\Downarrow$

Impossible

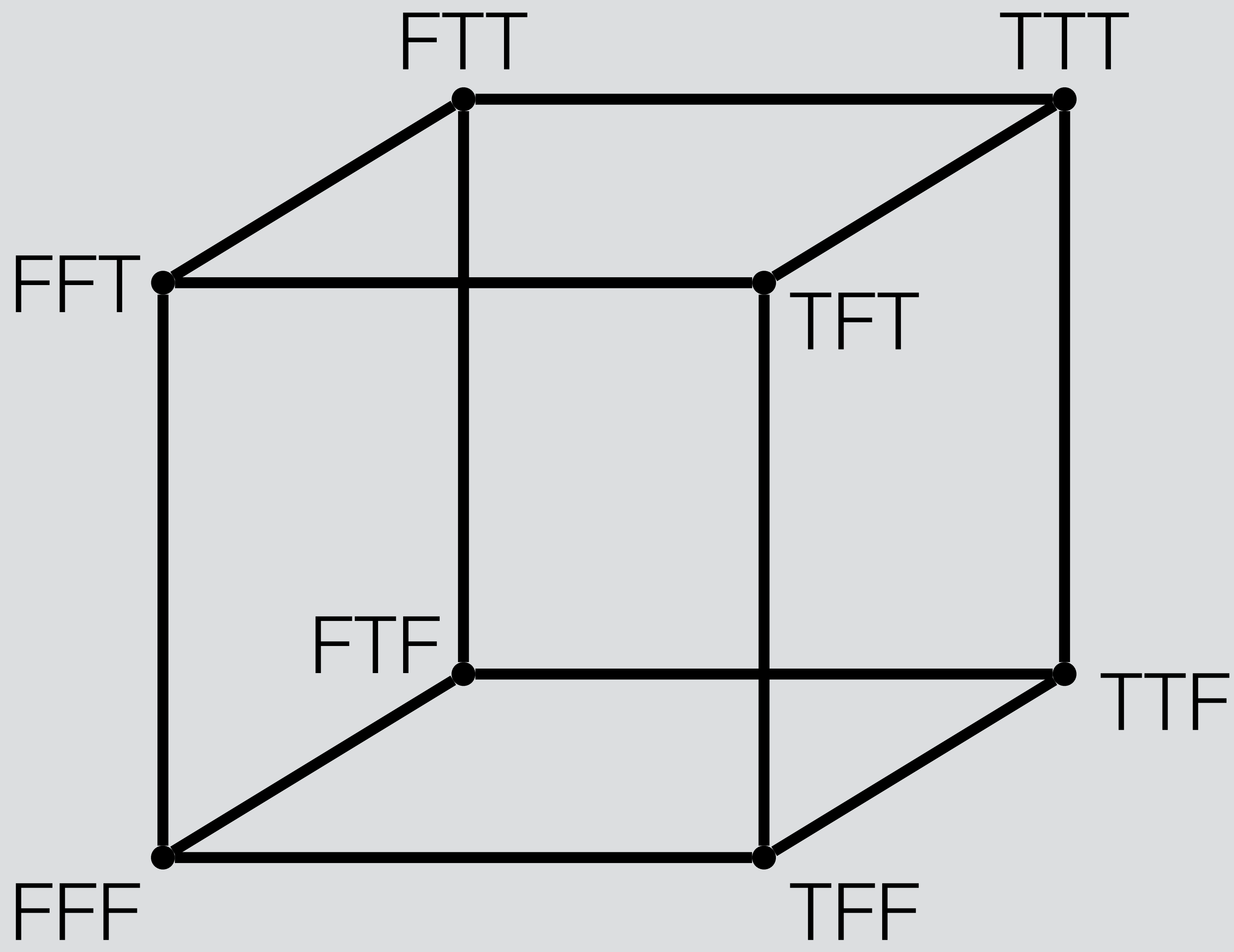
$\Downarrow$

$x_1 = T$

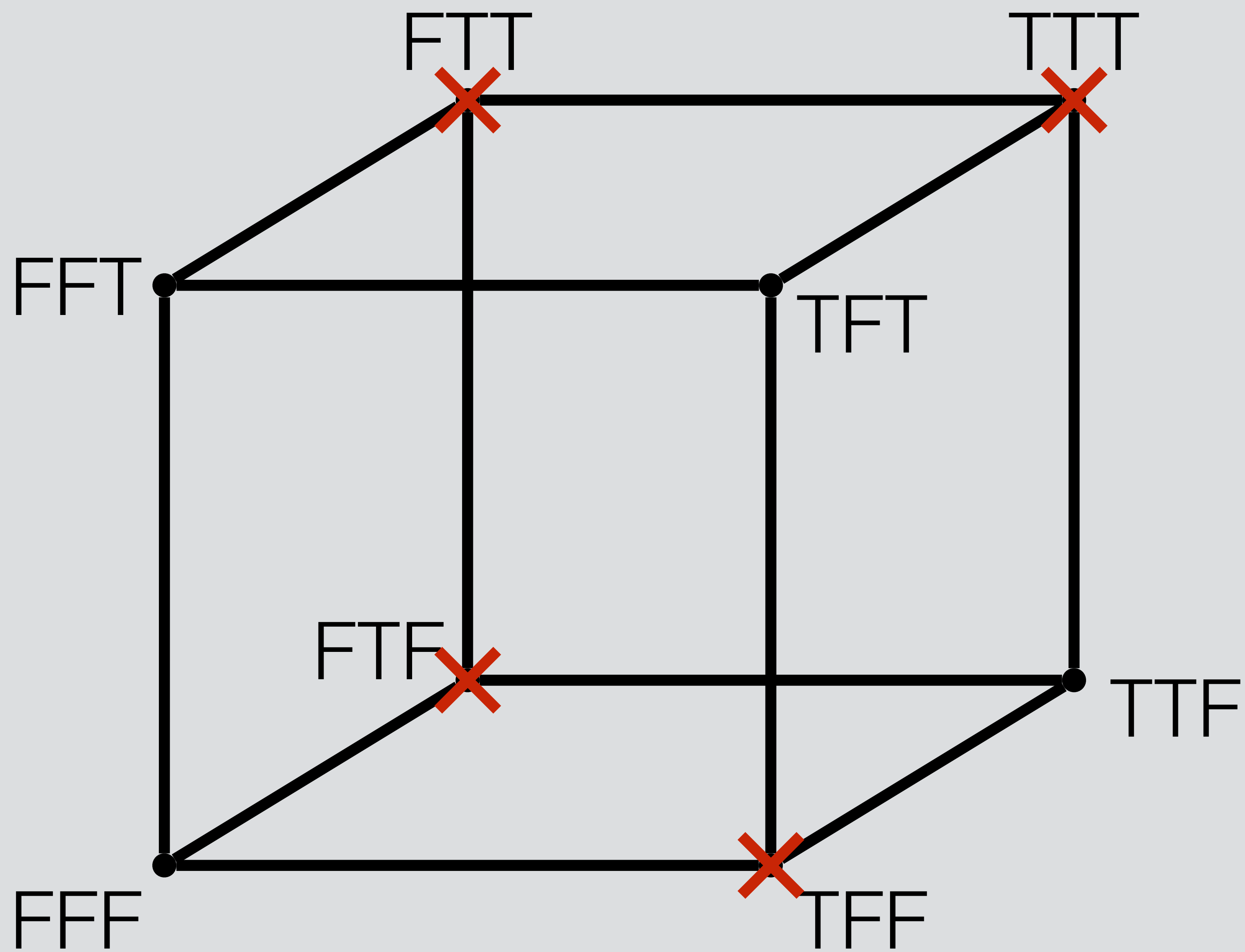
$x_2 = T$ $\quad (T \vee \neg T) \wedge (\neg T \vee T \vee F) \wedge (\neg T \vee \neg F \vee \neg F)$

$x_3 = F$

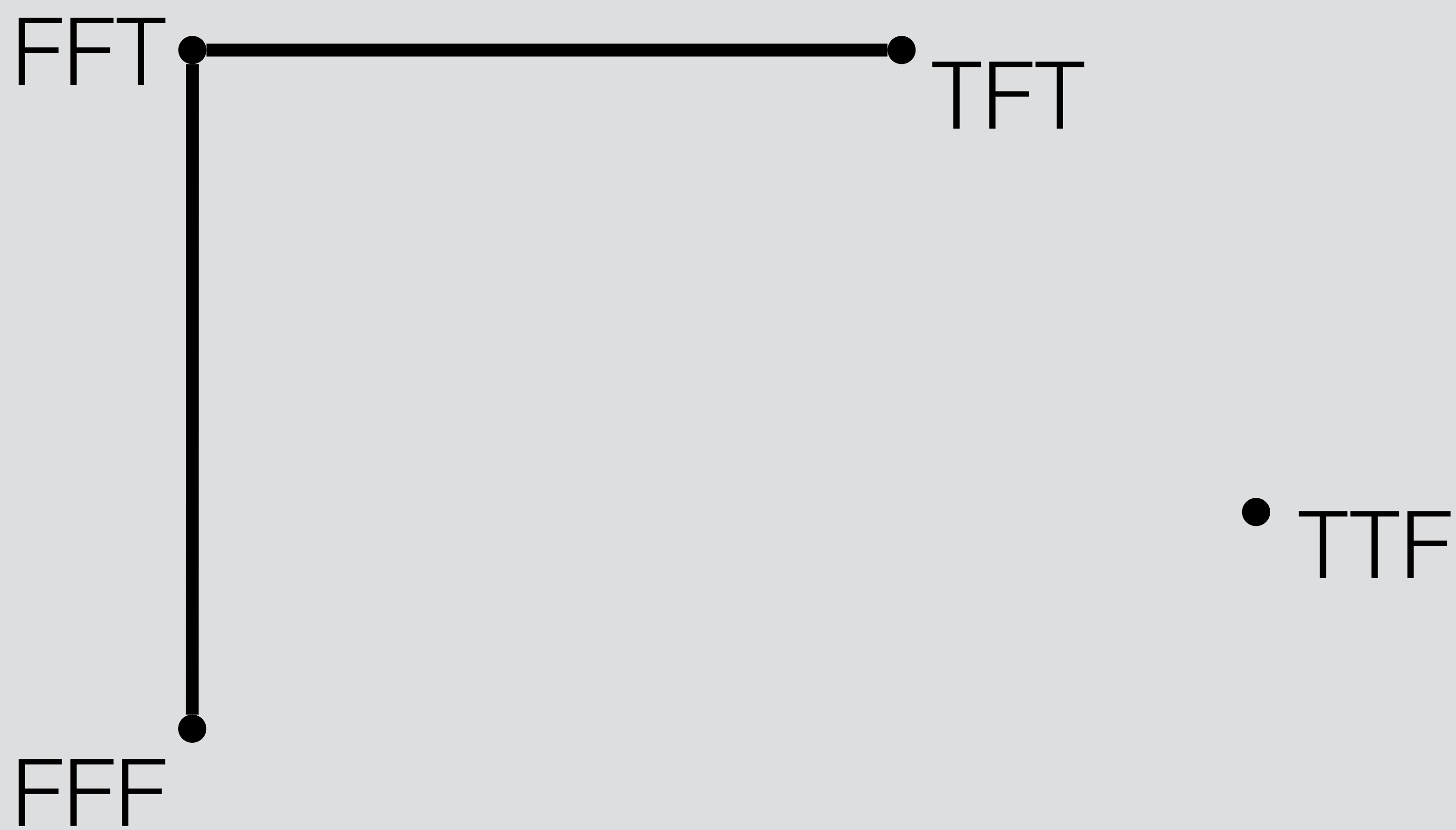
Formula: $(x_1 \lor \neg x_2) \land (\neg x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2 \lor \neg x_3)$



Formula: $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$



Formula: $(x_1 \lor \neg x_2) \land (\neg x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2 \lor \neg x_3)$



Reconfiguration:
a Subset Sum Example

Subset: $\{2, 3, 4, 5, 6, 7, 8\}$ Target sum: 12

$S = \{2, 3, 7\}$ $2 + 3 + 7 = 12$

Subset: {2, 3, 4, 5, 6, 7, 8} Target sum: 12

$S = \{2, 3, 7\}$ $2 + 3 + 7 = 12$

$\Downarrow$ swap 2, 3 with 5

$S = \{5, 7\}$ $5 + 7 = 12$

Subset: {2, 3, 4, 5, 6, 7, 8} Target sum: 12

$S = \{2, 3, 7\}$ $2 + 3 + 7 = 12$

$\Downarrow$ swap 2, 3 with 5

$S = \{5, 7\}$ $5 + 7 = 12$

$\Downarrow$ swap 7 with 3, 4

$S = \{5, 3, 4\}$ $5 + 3 + 4 = 12$

Subset: {2, 3, 4, 5, 6, 7, 8} Target sum: 12

$S = \{2, 3, 7\}$ $2 + 3 + 7 = 12$

$\Downarrow$ swap 2, 3 with 5

$S = \{5, 7\}$ $5 + 7 = 12$

$\Downarrow$ swap 7 with 3, 4

$S = \{5, 3, 4\}$ $5 + 3 + 4 = 12$

Keeping sum equal to target sum

Subset: {2, 3, 4, 5, 6, 7, 8} Target sum: 12

$S = \{2, 3, 7\}$ $2 + 3 + 7 = 12$

$\Downarrow$ swap 2, 3 with 5

$S = \{5, 7\}$ $5 + 7 = 12$

Keeping sum equal to target sum

$\Downarrow$ swap 7 with 3, 4

$S = \{5, 3, 4\}$ $5 + 3 + 4 = 12$

$S = \{2, 3, 7\}$ $2 + 3 + 7 = 12$

$\Downarrow$

$\Downarrow$

$S = \{2, 4, 6\}$ $5 + 3 + 4 = 12$

Subset: {2, 3, 4, 5, 6, 7, 8} Target sum: 12

S = {2, 3, 7} 2 + 3 + 7 = 12

⇓ swap 2, 3 with 5

S = {5, 7} 5 + 7 = 12

⇓ swap 7 with 3, 4

S = {5, 3, 4} 5 + 3 + 4 = 12

Keeping sum equal to target sum

S = {2, 3, 7} 2 + 3 + 7 = 12

⇓

✗ Impossible

⇓

S = {2, 4, 6} 5 + 3 + 4 = 12

3SAT Reconfiguration Problem

3SAT Reconfiguration Problem

Input:

- An instance of 3SAT Φ.
- A satisfying assignment A of Φ.
- A satisfying assignment B of Φ.

Output:

Whether A can be reconfigured into B.

SAT Reconfiguration Problem

Input:

- 3SAT formula $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$
- $x_1 = F$, $x_2 = F$, $x_3 = F$.
- $x_1 = T$, $x_2 = F$, $x_3 = T$.

Output: Yes (can be reconfigured).

Input:

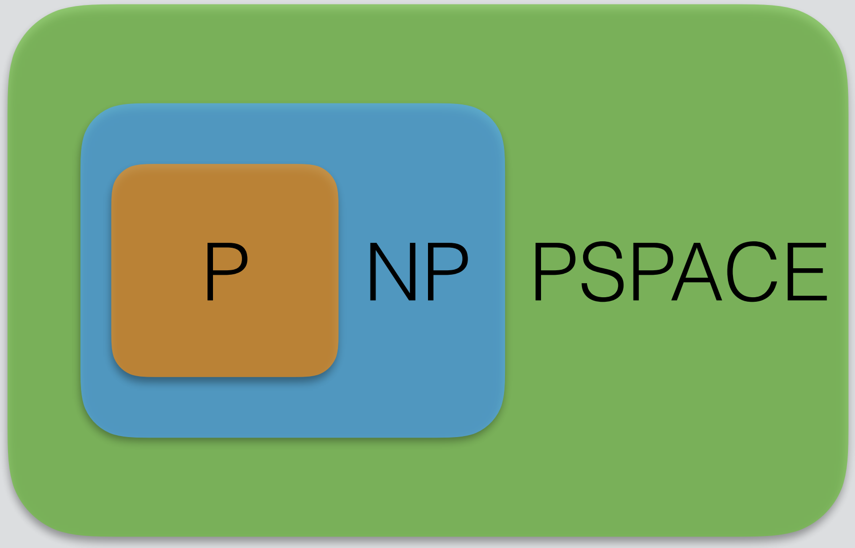
- 3SAT formula $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$
- $x_1 = F$, $x_2 = F$, $x_3 = F$.
- $x_1 = T$, $x_2 = T$, $x_3 = F$.

Output: No (cannot be reconfigured).

SAT Reconfiguration Problem

Theorem: the 3SAT reconfiguration problem is PSPACE-complete.

problems solvable in $n^{O(1)}$ space



Corollary: some reconfigurations require *exponentially* many variable flips.

SAT Variants

1-in-3SAT

One-in-three (1-in-3): satisfying assignment if 1 (but not 2 or 3) true literals per clause.

$$(x_1 \vee x_3 \vee x_4) \wedge (x_2 \vee x_2 \vee x_4) \wedge (x_1 \vee x_2 \vee x_4)$$

$x_1 = F$
$x_2 = F$
$x_3 = F$
$x_4 = T$

$$(F \vee F \vee T) \quad \wedge \quad (F \vee F \vee T) \quad \wedge \quad (F \vee F \vee T)$$

satisfied — satisfied — satisfied

$x_1 = T$
$x_2 = F$
$x_3 = F$
$x_4 = T$

$$(T \vee F \vee T) \quad \wedge \quad (F \vee F \vee T) \quad \wedge \quad (T \vee F \vee T)$$

not satisfied — satisfied — not satisfied

SAT Reconfiguration Problems

Theorem: the 2SAT reconfiguration problem is in P. [Gopalan et al. 2009]

Theorem: the 1-in-3SAT reconfiguration problem is in P (always "No").

Theorem: the 3SAT reconfiguration problem is PSPACE-complete.

SAT Reconfiguration Problems

Theorem: the 2SAT reconfiguration problem is in P. [Gopalan et al. 2009]

Theorem: the 1-in-3SAT reconfiguration problem is in P (always "No").

Theorem: the 3SAT reconfiguration problem is PSPACE-complete.

Is there a SAT variant whose:

- Solving problem ("Does a satisfying assignment exist?") is in P.
- Reconfiguration problem is PSPACE-complete.

SAT Reconfiguration Problems

Theorem: the 2SAT reconfiguration problem is in P. [Gopalan et al. 2009]

Theorem: the 1-in-3SAT reconfiguration problem is in P (always "No").

Theorem: the 3SAT reconfiguration problem is PSPACE-complete.

Is there a SAT variant whose:

- Solving problem ("Does a satisfying assignment exist?") is in P.
- Reconfiguration problem is PSPACE-complete.

Yes, monotone planar NAE 3SAT.

Monotone Planar NAE 3SAT

Monotone: no negated variables.

Planar: graph of variables and clauses is planar.

Not-All-Equal (NAE): satisfying assignment
if 1 or 2 (but not 3) true literals per clause.

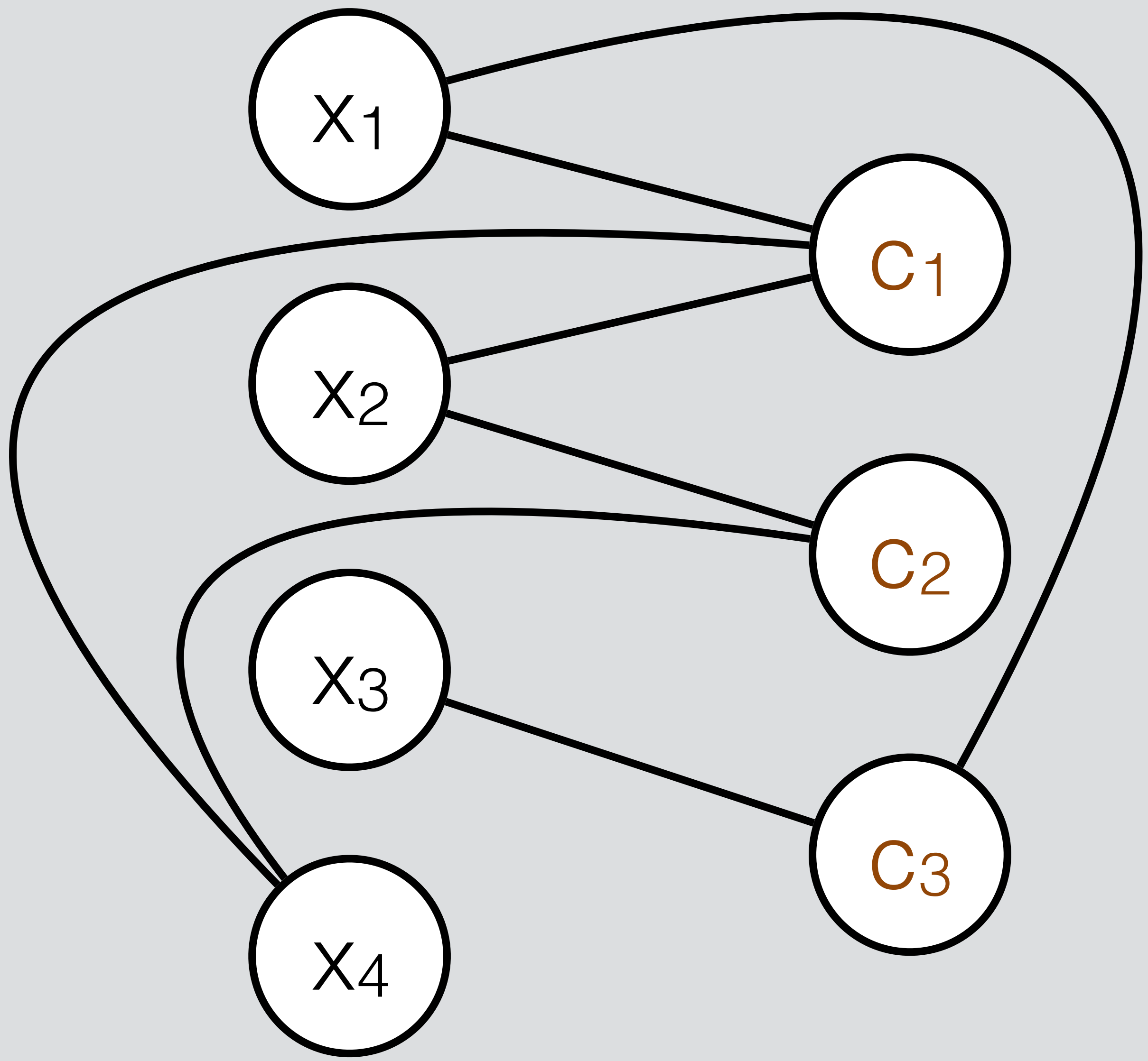
Monotone Planar NAE 3SAT

$(x_1 \lor x_2 \lor x_4) \land (x_2 \lor x_2 \lor x_4) \land (x_1 \lor x_2 \lor x_3)$

Monotone Planar NAE 3SAT

$(x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_2 \vee x_4) \wedge (x_1 \vee x_2 \vee x_3)$

c_1 c_2 c_3



Monotone Planar NAE 3SAT

$(x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_2 \vee x_4) \wedge (x_1 \vee x_2 \vee x_3)$

$x_1 = T$
$x_2 = F$
$x_3 = T$
$x_4 = T$

$(T \vee F \vee T) \wedge (F \vee F \vee T) \wedge (T \vee F \vee T)$

satisfied satisfied satisfied

$x_1 = T$
$x_2 = T$
$x_3 = T$
$x_4 = F$

$(T \vee T \vee F) \wedge (T \vee T \vee F) \wedge (T \vee T \vee T)$

satisfied satisfied not satisfied

Monotone Planar NAE 3SAT

Monotone planar NAE 3SAT solving is in P [Moret 1988]

Theorem: monotone planar NAE 3SAT reconfiguration is PSPACE-complete.

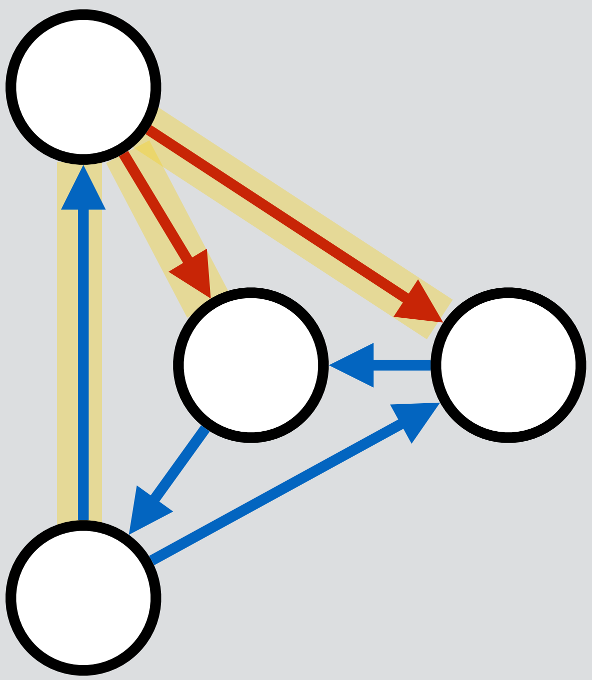
Reduction is from non-deterministic constraint logic (NCL)

Non-Deterministic Constraint Logic (NCL)

weight 1

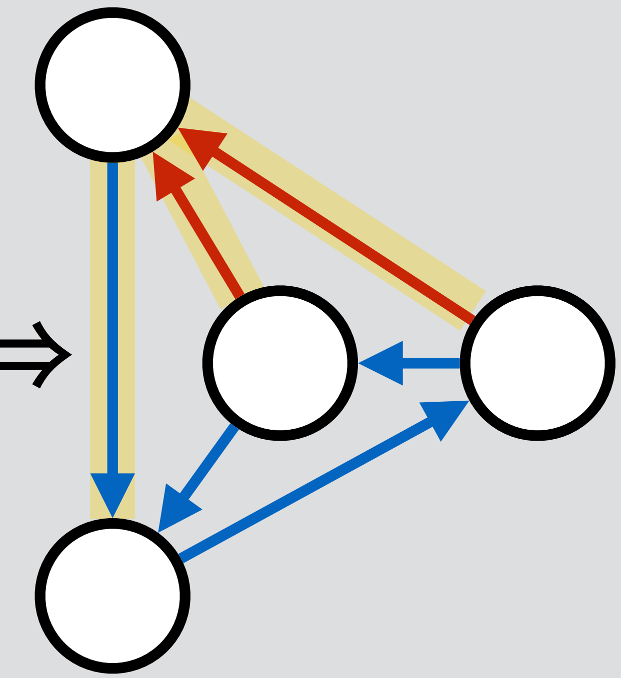
weight 2

Each node needs
incoming weight ≥2



⇒

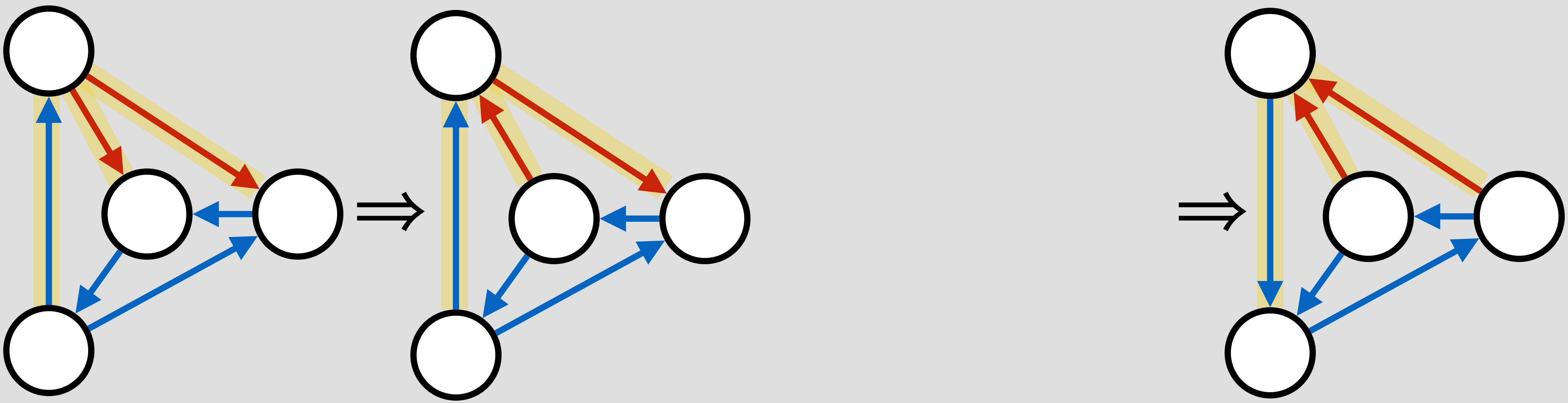
⇒



Non-Deterministic Constraint Logic (NCL)

weight 1 → (red)
weight 2 → (blue)

Each node needs incoming weight ≥2

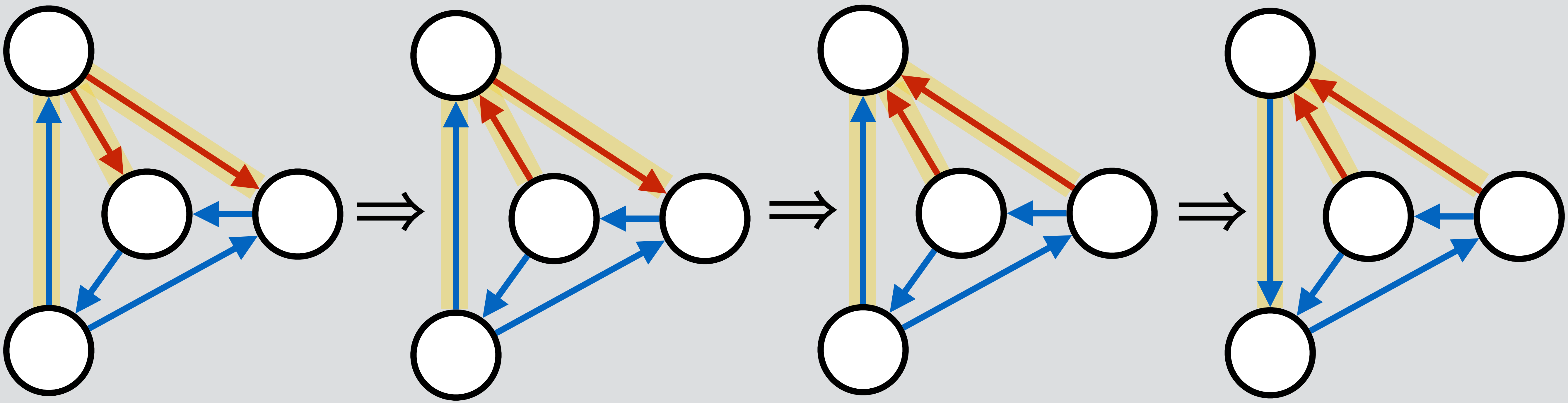


Non-Deterministic Constraint Logic (NCL)

weight 1 →

weight 2 →

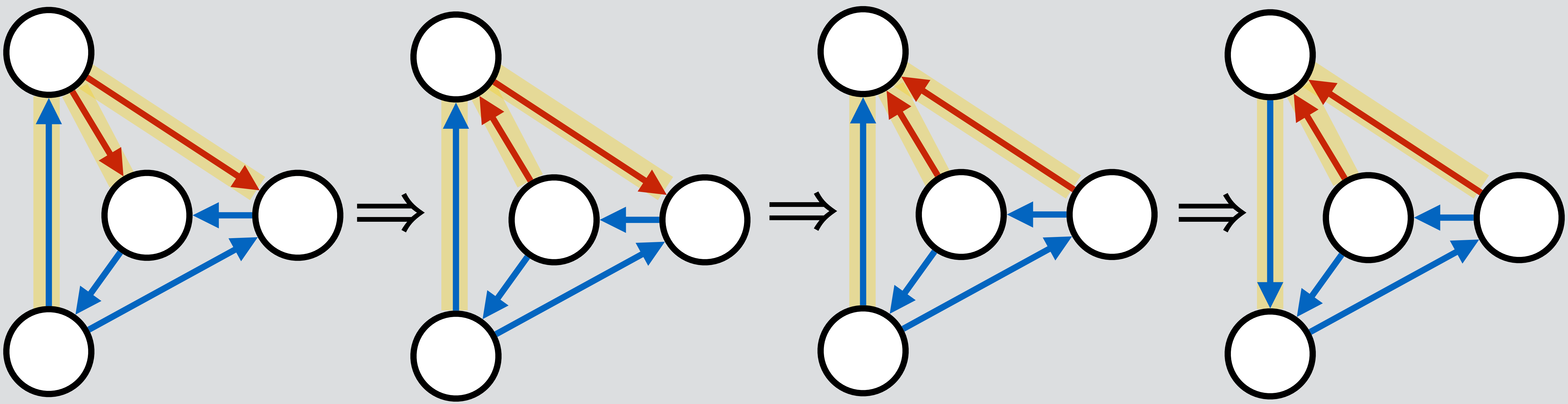
Each node needs incoming weight ≥2



Non-Deterministic Constraint Logic (NCL)

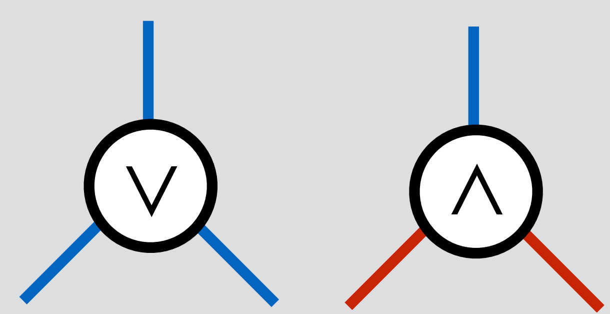
weight 1 →
weight 2 →

Each node needs incoming weight ≥2



NCL reconfiguration is PSPACE-complete, even for:

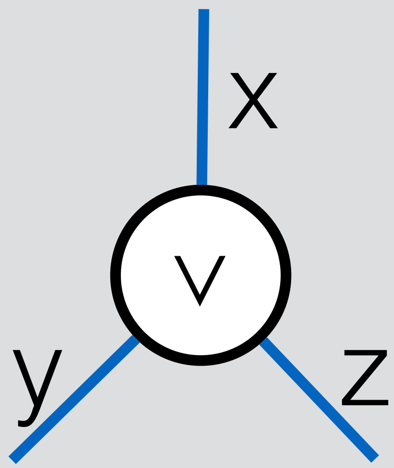
- Planar, degree-3 graphs.
- Only two types of nodes:
- Proved by [DH 2005]



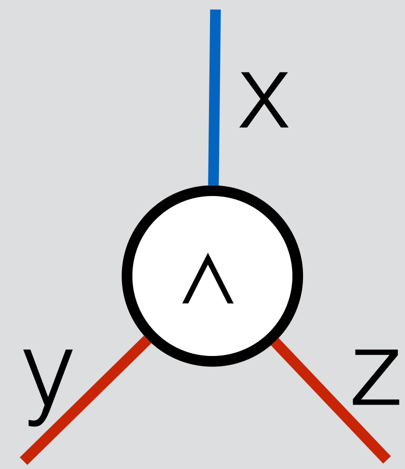
3SAT Reconfiguration is PSPACE-hard

Create a variable for orientation of each edge.

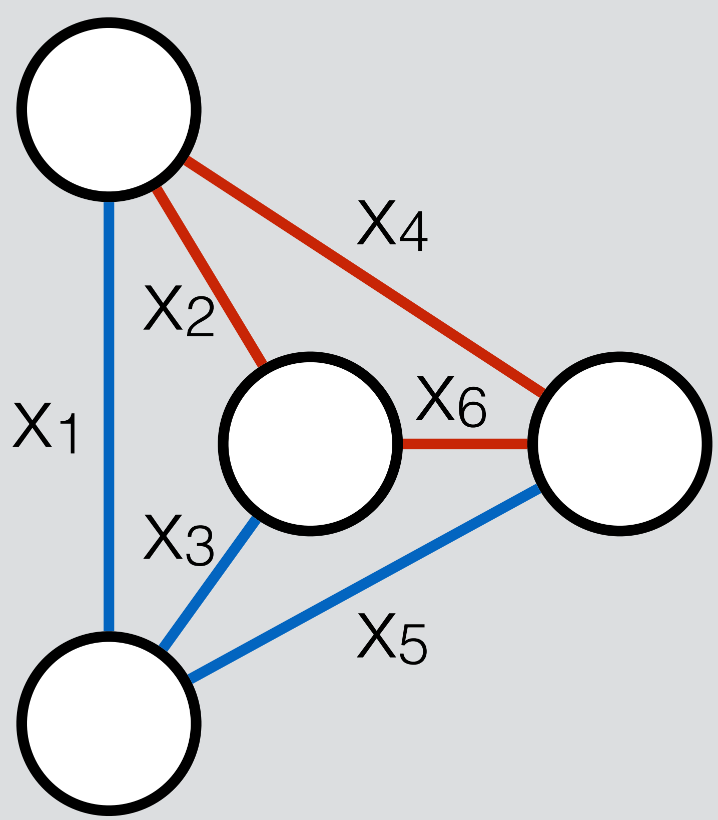
Create a clause set for each node.



$$\cong (x \vee y \vee z)$$



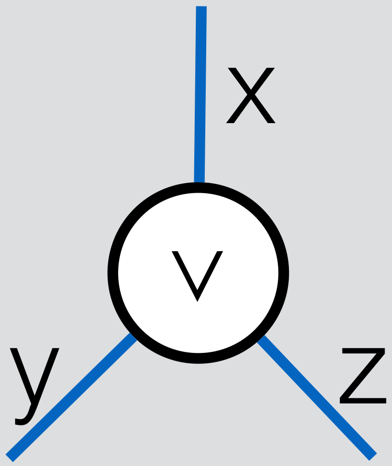
$$\cong (x \vee y) \wedge (x \vee z)$$



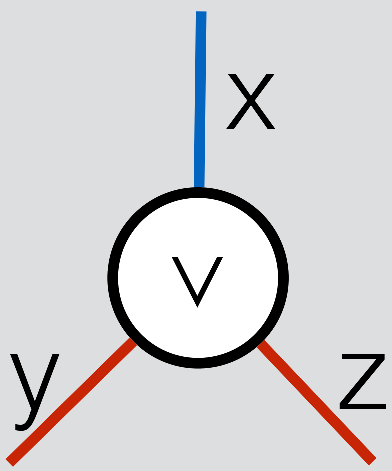
$$\cong (\neg x_1 \vee \neg x_2 \vee \neg x_5) \wedge (x_1 \vee x_2) \wedge (x_1 \vee x_4) \wedge (x_3 \vee \neg x_2) \wedge (x_3 \vee x_6) \wedge (x_5 \vee \neg x_4) \wedge (x_5 \vee \neg x_6)$$

<u>Theorem:</u> monotone planar NAE 3SAT reconfiguration is PSPACE-complete.

Reduction is from non-deterministic constraint logic (NCL).



$$\cong \quad (x \vee F \vee c) \wedge (a \vee b \vee c) \wedge (a \vee y \vee F) \wedge (b \vee z \vee F)$$

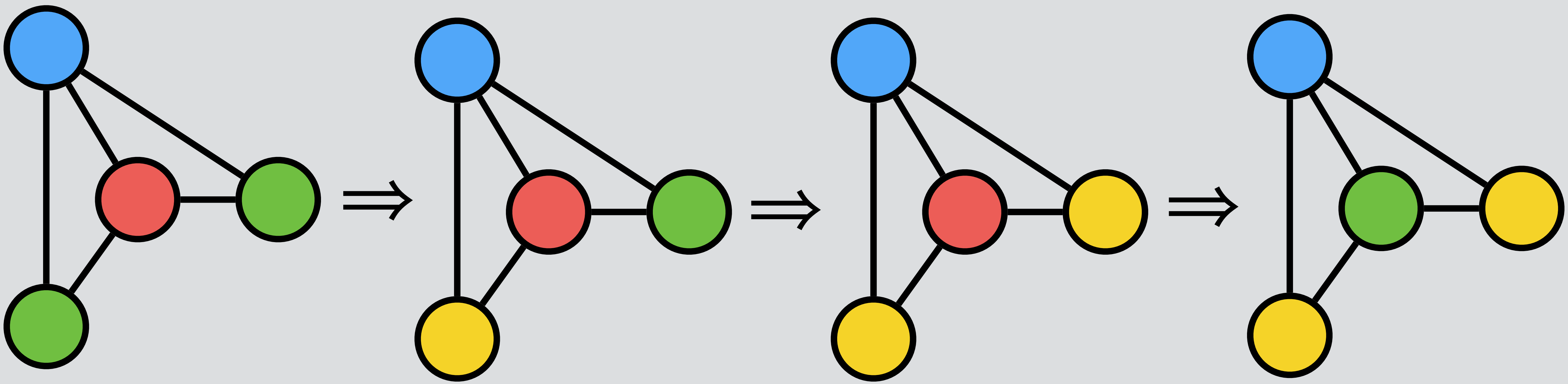


$$\cong \quad (x \vee y \vee F) \wedge (x \vee z \vee F)$$

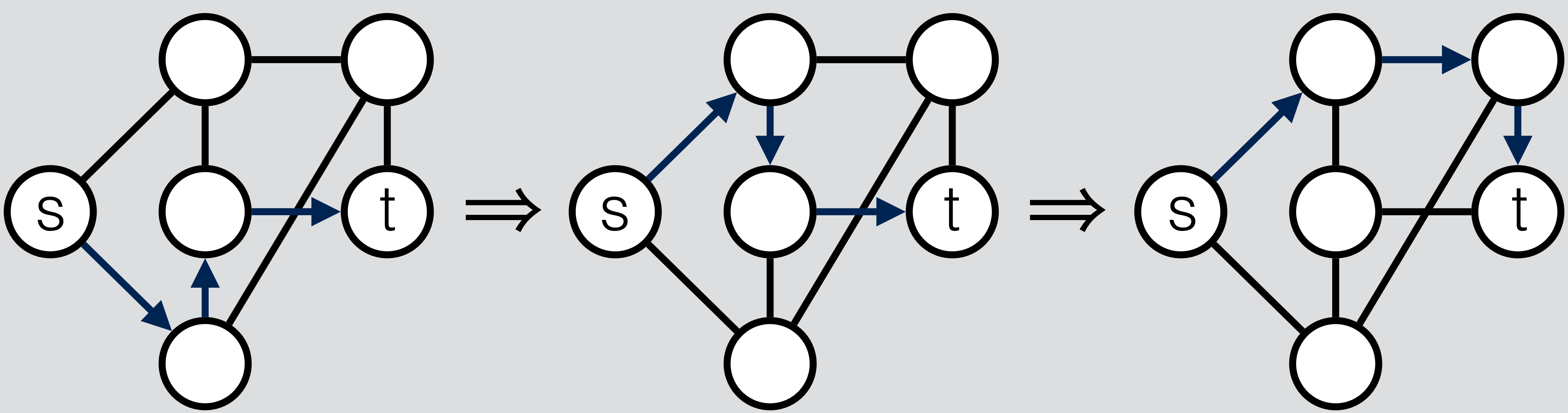
Other easy-to-solve, hard-to-connect problems

Easy-to-Solve Hard-to-Connect Problems

Reconfiguring planar graph 4-colorings. [Bonsma, Cerceda 2009]



Reconfiguring shortest paths. [Bonsma 2013]



Unary Input Subset Sum

Two options for subset sum reconfiguration:

1. Swap x, y and x+y, keep target sum.
2. Add/remove x, keep sum in target range.

Option 1	Option 2
S = {2, 3, 7}	S = {2, 3, 7}
⇓ swap 2, 3 with 5	⇓ remove 2
S = {5, 7}	S = {3, 7}
⇓ swap 7 with 3, 4	⇓ add 4
S = {5, 3, 4}	S = {3, 4, 7}

Unary Input Subset Sum

Two options for subset sum reconfiguration:

1. Swap x, y and x+y, keep target sum.
2. Add/remove x, keep sum in target range.

<u>Option 1</u>

S = {2, 3, 7}

⇓ swap 2, 3 with 5

S = {5, 7}

⇓ swap 7 with 3, 4

S = {5, 3, 4}

<u>Option 2</u>

S = {2, 3, 7}

NP-hard
[Ito, Demaine 2014]

⇓ add 4

S = {3, 4, 7}

Unary Input Subset Sum

Two options for subset sum reconfiguration:

1. Swap x, y and x+y, keep target sum.
2. Add/remove x, keep sum in target range.

Option 1

S = {2, 3, 7}

⇓ swap 2, 3 with 5

PSPACE-complete
(This work)

S = {5, 7}

⇓ swap 7 with 3, 4

S = {5, 3, 4}

Option 2

S = {2, 3, 7}

⇓ remove 2

NP-hard
[Ito, Demaine 2014]

S = {3, 7}

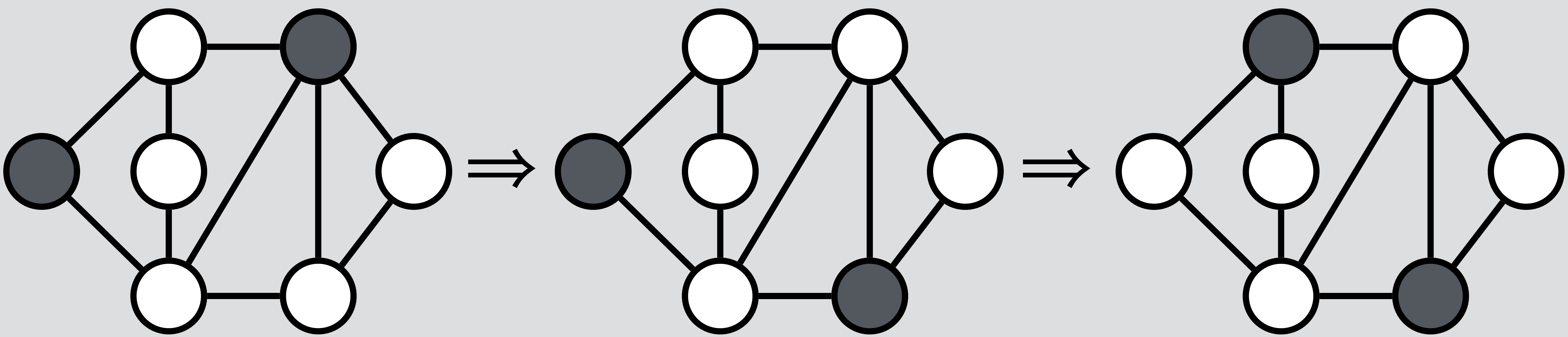
⇓ add 4

S = {3, 4, 7}

Subset Sum Reconfiguration

Theorem: subset sum reconfiguration via swapping x, y and x+y is strongly PSPACE-complete.
unary input

Reduction is from *token sliding*: reconfiguring independent sets via swapping adjacent vertices.



Reconfiguration problem is PSPACE-complete, even for 3-regular graphs [DH 2005]

Conclusion

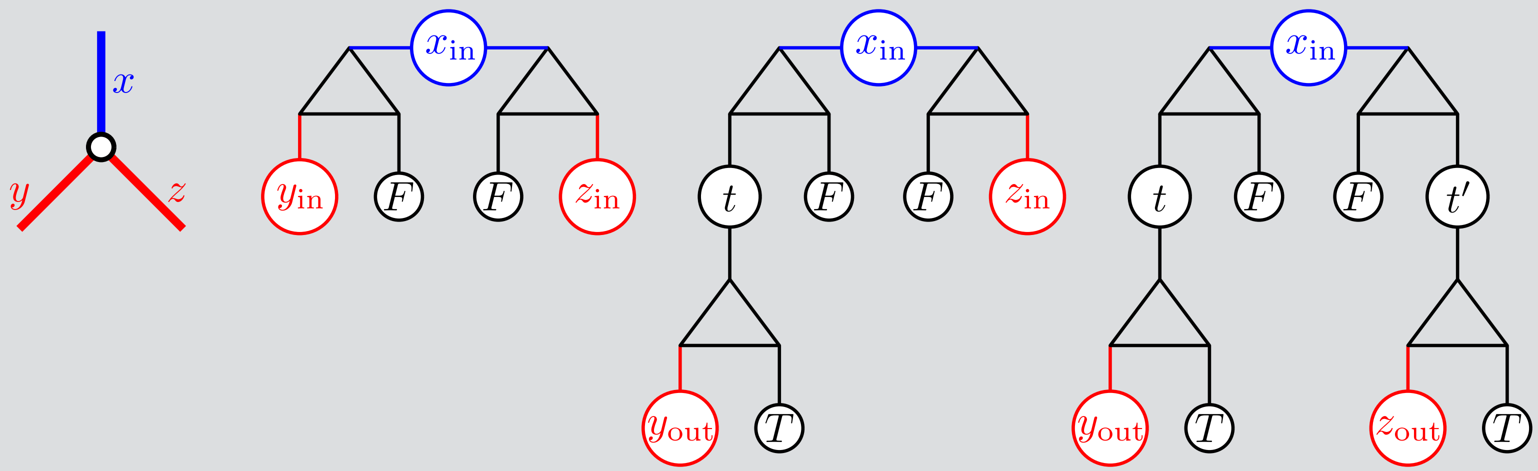
Two new “easy-to-solve, hard-to-connect” problems:

- monotone planar NAE 3SAT
- subset sum via swapping x, y and x+y.

Open:

- PSPACE-hardness of subset sum via add/remove x?
- Meta-theorems on reconfiguration for problems in P?
    - Dichotomy theorem for SAT [Gopalan et al. 2009]

Reconfiguration of Satisfying Assignments and Subset Sums: Easy to Find, Hard to Connect



Jean Cardinal, Erik Demaine, David Eppstein, Robert Hearn, *Andrew Winslow*

ULB Université Libre de Bruxelles
UCI University of California, Irvine
Massachusetts Institute of Technology
The University of Texas Rio Grande Valley