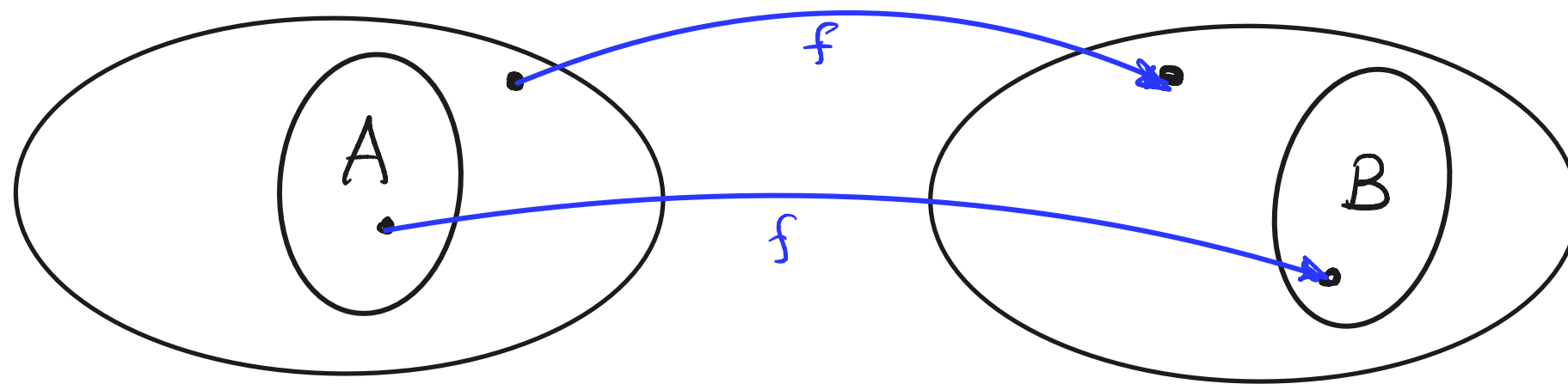


Lecture 22

- HW5 due Sunday 11 am
- Review problems posted for Test 2
(on course webpage - Supplementary Materials)
will post SOLNS SUNDAY OR MONDAY
- Review Session tentatively Mon Dec 1 6pm

NP-completeness

Definition Language A is **polynomial-time (mapping) reducible** to B (written $A \leq_p B$) if there is a polynomial-time computable function $f: \Sigma^* \rightarrow \Sigma^*$ such that $w \in A \iff f(w) \in B$



Definition

- A language $B \subseteq \{0,1\}^*$ is **NP-hard** if for every $A \in \text{NP}$ there is a polynomial time reduction from A to B ($A \leq_p B$)
- $B \subseteq \{0,1\}^*$ is **NP-complete** if:
 - (i) B is in NP and
 - (ii) B is NP-hard

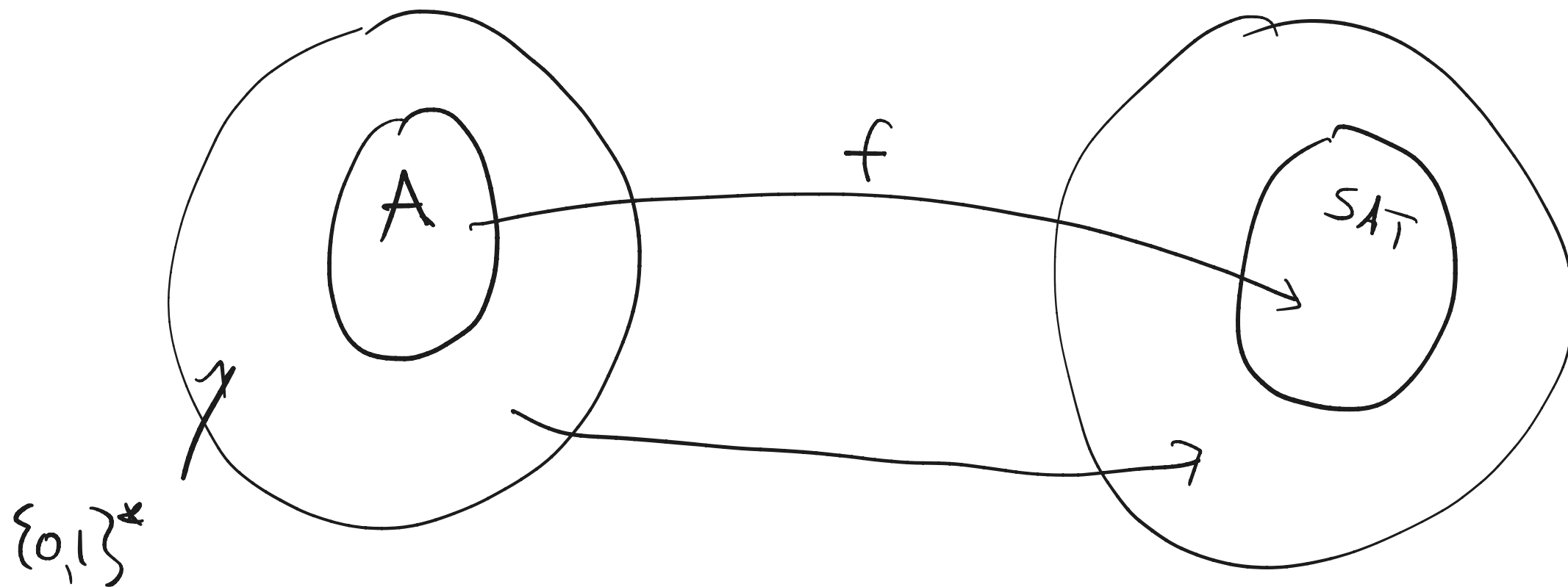
NP-completeness

Cook-Levin theorem

SAT is NP-complete

} Proof Today

$\forall A \in \{0,1\}^*$ in NP

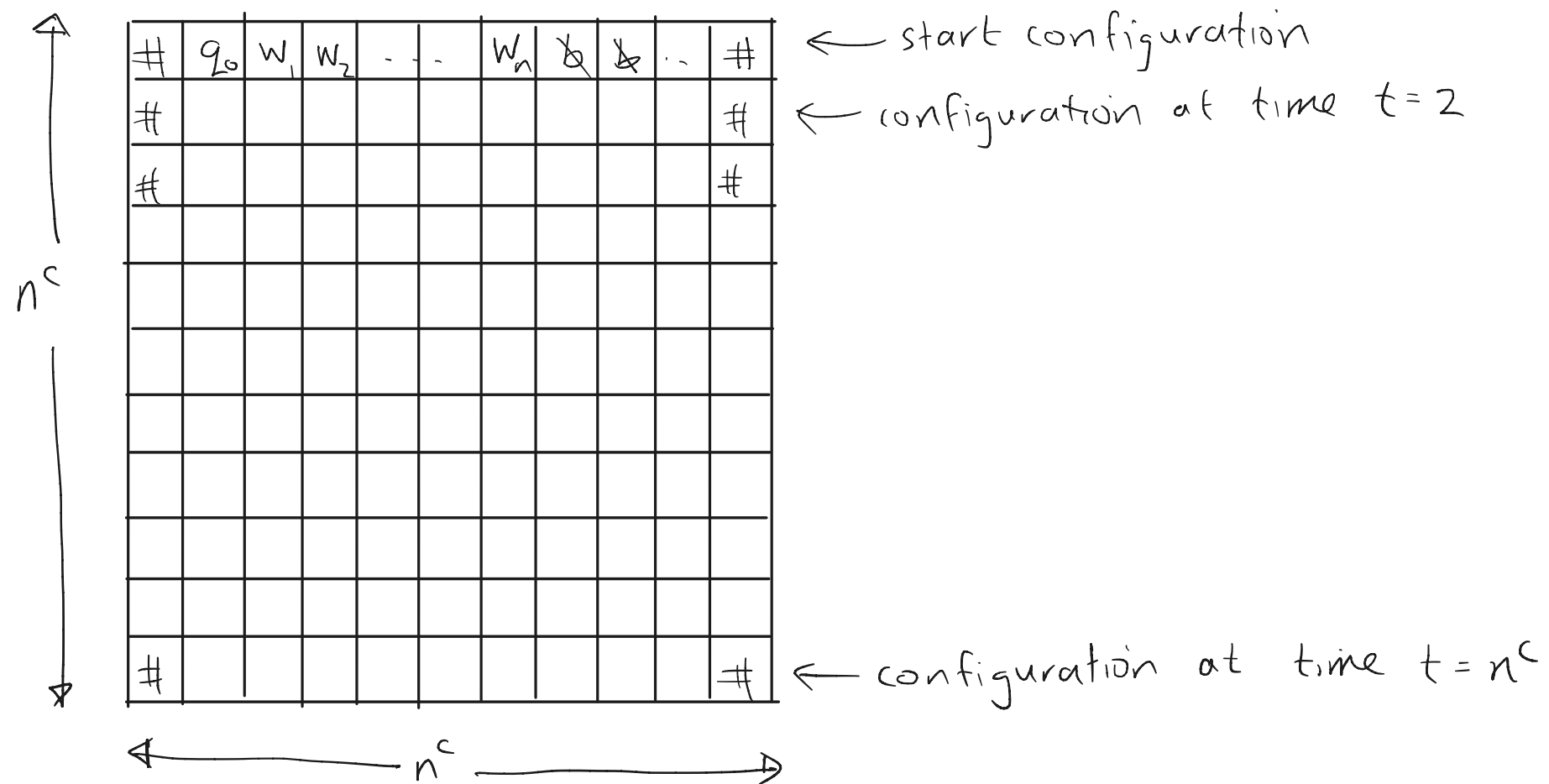


Cook-Levin Theorem : KSAT is NP-complete

1. KSAT $\in$ NP (already did)
2. To prove KSAT is NP-hard we have to prove:
For every language $A \in$ NP, $A \leq_p$ KSAT

Let $A \in$ NP and let N be a Nondet. TM accepting A in polynomial time, n^c , $c \geq 0$.

A tableaux for N on input w :



High Level Idea :

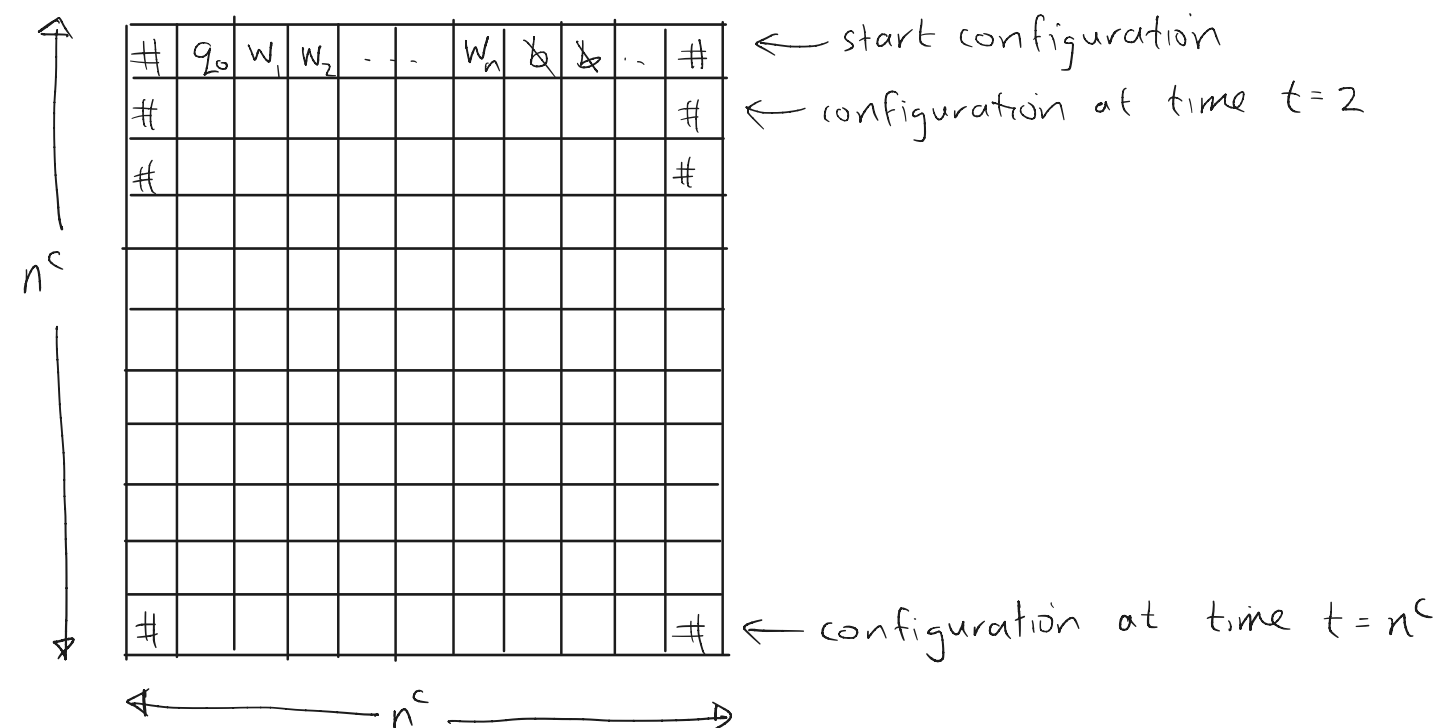
A tableaux for N on input w :

$w \in A \iff \exists$ accepting
computation path
of N on w

$\iff \exists$ accepting tableaux
for N on w

$\iff \underbrace{\exists \vec{X}}_{\text{description of tableaux}}$ such that $\vec{X}$ is legal tableaux of N on w
and $\vec{X}$ is accepting

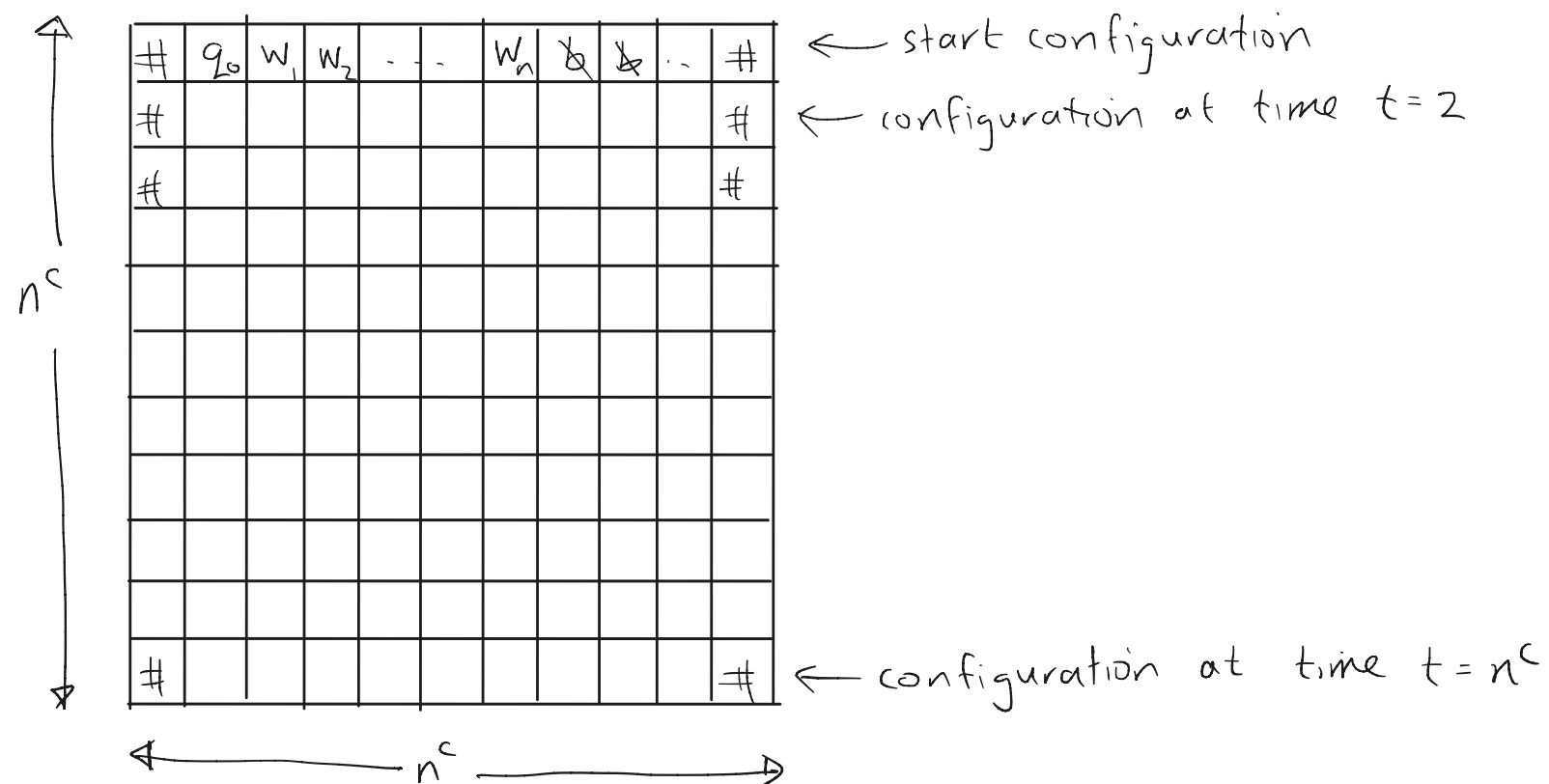
$\iff \underbrace{\phi(\vec{X})}_{\text{CNF}}$ is satisfiable



Cook-Levin Theorem : KSAT is NP-complete

Let $A \in \text{NP}$ and let N be a nondet. TM accepting A in polynomial time, n^c , $c \geq 0$.

A tableaux for N on input w :



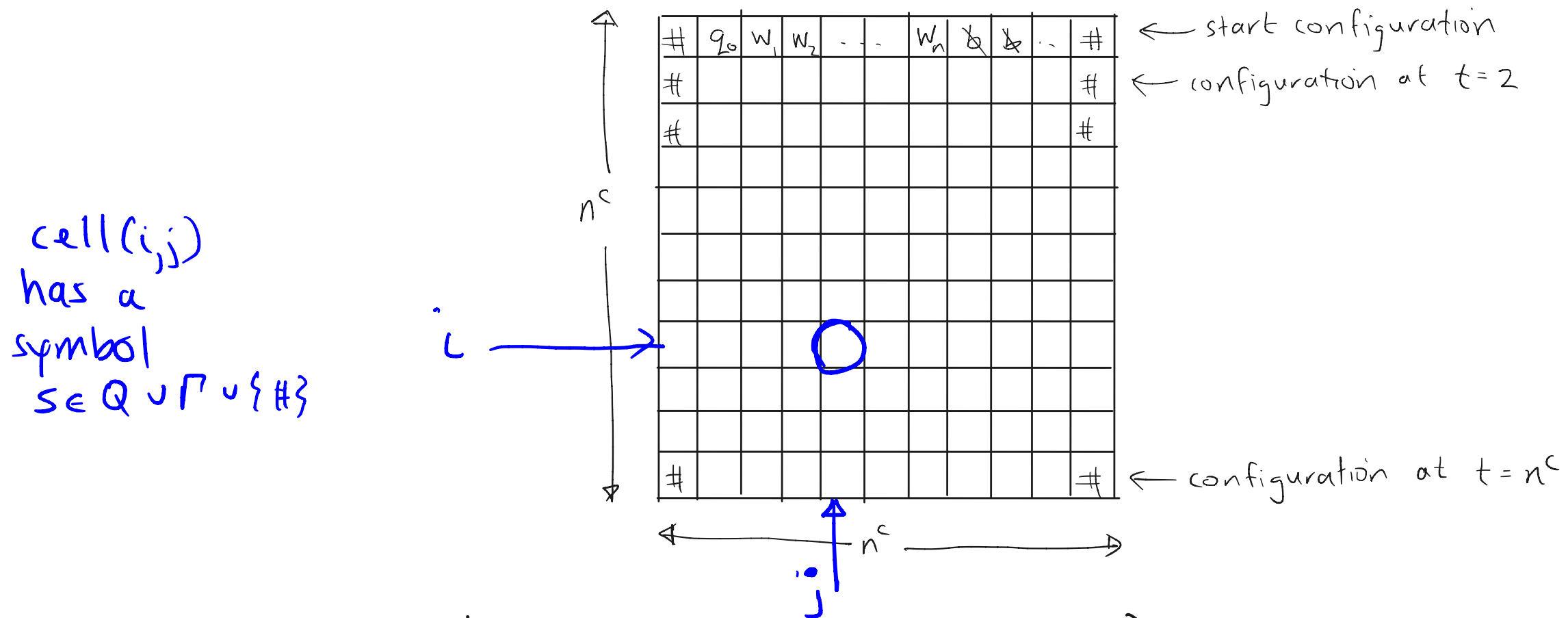
A tableaux is **accepting** if for some $t \leq n^c$ configuration at time t is accepting
(the state at time t is an accept state)

We want $f: w \in \Sigma^* \longrightarrow \text{CNF } \phi$

such that $w \in A$ iff $f(w)$ satisfiable

Cook-Levin Theorem: KSAT is NP-complete

We want $f: w \rightarrow \phi$ such that ϕ is satisfiable iff there is an accepting tableau of N on input w .



Variables of ϕ : $x_{i,j,s}$, $i, j \in \{1, \dots, n^c\}$, $s \in Q \cup \Gamma \cup \{\#\}$

($x_{i,j,s} = 1$ iff cell (i, j) contains symbol s)

Q = states of N
 Γ = tape alphabet

$$\phi = \underbrace{\phi_{\text{cell}}} \wedge \underbrace{\phi_{\text{start}}} \wedge \underbrace{\phi_{\text{move}}} \wedge \underbrace{\phi_{\text{accept}}}$$

Cook-Levin Theorem : KSAT is NP-complete

Variables of ϕ : $x_{i,j,s}$, $i,j \in \{1, \dots, n^c\}$, $s \in Q \cup \Gamma \cup \{\#\}$

($x_{i,j,s} = 1$ iff cell (i,j) contains symbol s)

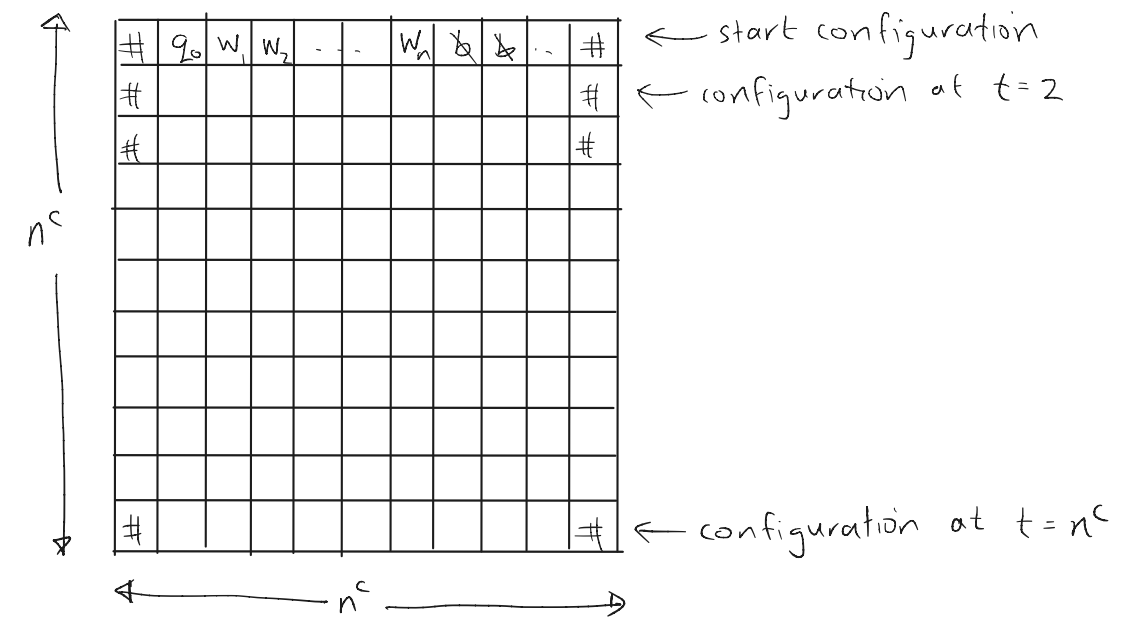
$$\phi = \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \phi_{\text{move}} \wedge \phi_{\text{accept}}$$

ϕ_{cell} : states that every cell contains exactly one symbol $s \in Q \cup \Gamma \cup \{\#\}$

ϕ_{start} : start config is $\# q_0 w_1 \dots w_n \#$

ϕ_{accept} : some cell contains an accept state q_{accept}

ϕ_{move} : each row (configuration at time t) follows from previous row (config at time $t-1$) by a valid transition according to N 's transition function



Cook-Levin Theorem : KSAT is NP-complete

Variables of ϕ : $x_{i,j,s}$, $i,j \in \{1, \dots, n^c\}$, $s \in \underbrace{Q \cup \Gamma \cup \{\#\}}_C$

($x_{i,j,s} = 1$ iff cell (i,j) contains symbol s)

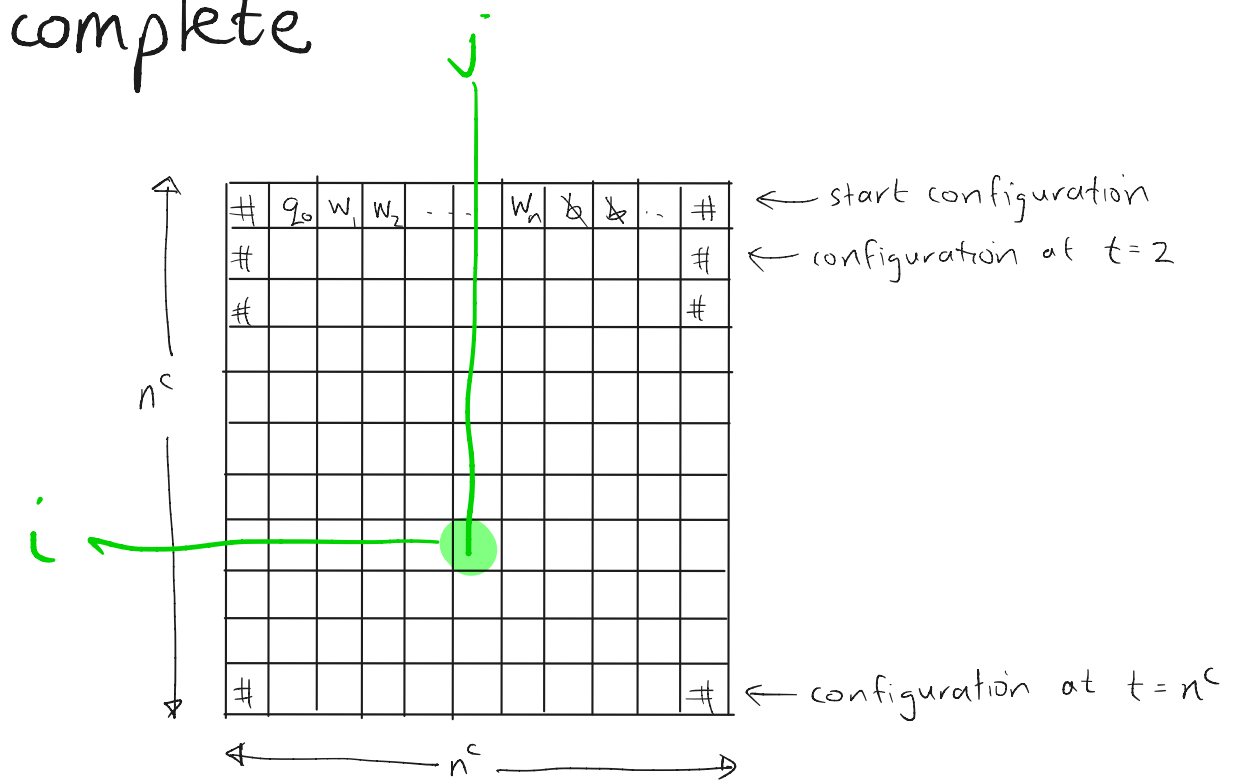
$$\phi = \boxed{\phi_{\text{cell}}} \wedge \phi_{\text{start}} \wedge \phi_{\text{move}} \wedge \phi_{\text{accept}}$$

$\boxed{\phi_{\text{cell}}}$: states that every cell contains exactly one symbol $s \in Q \cup \Gamma \cup \{\#\}$

$$= \bigwedge_{1 \leq i,j \leq n^c} \left[\left(\bigvee_{s \in C} x_{i,j,s} \right) \wedge \left(\bigwedge_{\substack{s,t \in C \\ s \neq t}} \overline{x_{i,j,s}} \vee \overline{x_{i,j,t}} \right) \right]$$

every cell (i,j)
contains a symbol

every cell (i,j)
contains at most
one symbol



cell (i,j) contains
 $x_{i,j,q_0} = 1$
 $x_{i,j,t} = 0$
 $\forall t \neq q_0$

Cook-Levin Theorem : K SAT is NP-complete

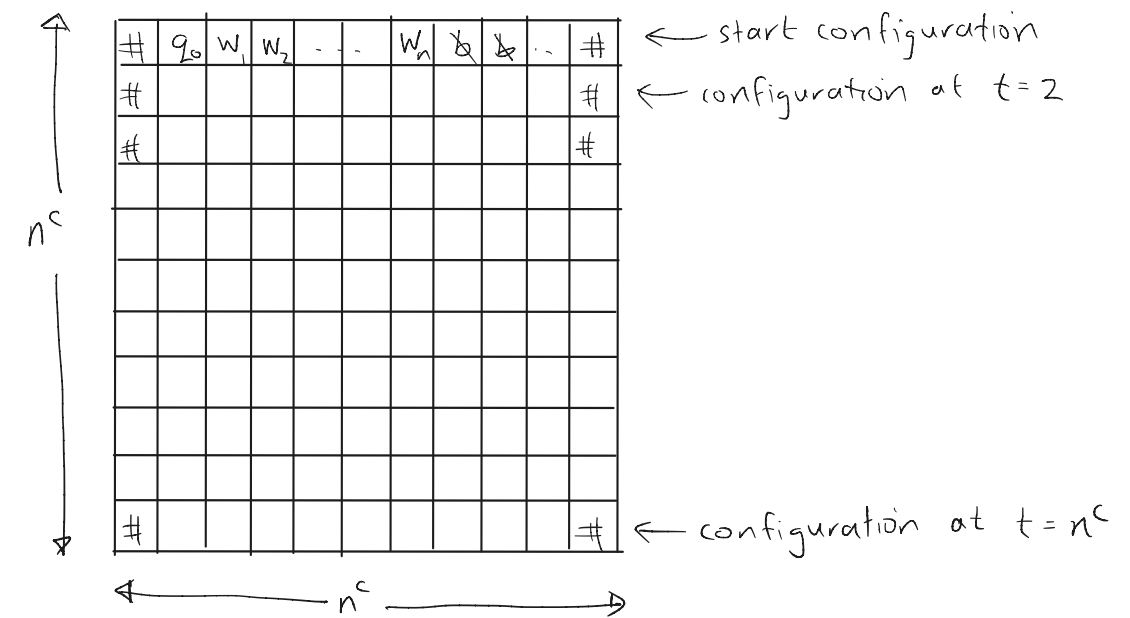
Variables of ϕ : $x_{i,j,s}$, $i,j \in \{1, \dots, n^c\}$, $s \in Q \cup \Gamma \cup \{\#\}$

($x_{i,j,s} = 1$ iff cell (i,j) contains symbol s)

$$\phi = \phi_{\text{cell}} \wedge \boxed{\phi_{\text{start}}} \wedge \phi_{\text{move}} \wedge \phi_{\text{accept}}$$

$\boxed{\phi_{\text{start}}}$: start config is $\# q_0 w_1 \dots w_n \& \dots \& \#$

$$= \underbrace{x_{1,1,\#}}_{\text{starts with \#}} \wedge \underbrace{x_{1,2,q_0}}_{\text{then } q_0} \wedge \underbrace{x_{1,3,w_1} \wedge x_{1,4,w_2} \wedge \dots \wedge x_{1,n+2,w_n}}_{\text{then } w_1 \dots w_n} \wedge \underbrace{x_{1,n+3,\&} \wedge \dots \wedge x_{1,n^c-1,\&}}_{\text{blanks}} \wedge \underbrace{x_{1,n^c,\#}}_{\text{(last symbol is \#)}}$$



Cook-Levin Theorem : KSAT is NP-complete

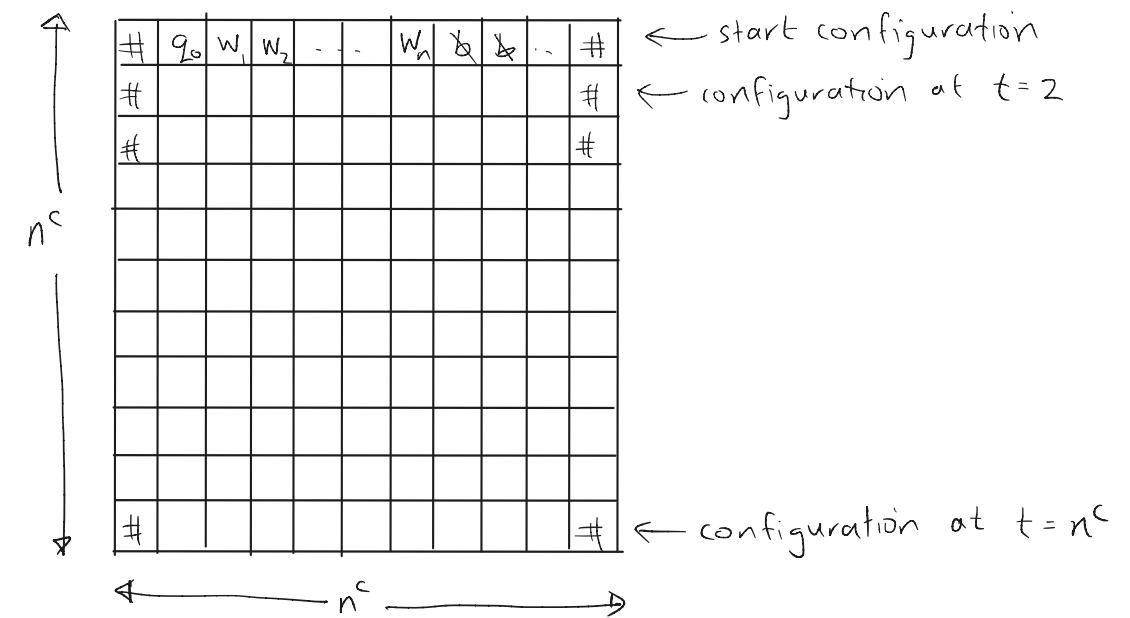
Variables of ϕ : $x_{i,j,s}$, $i,j \in \{1, \dots, n^c\}$, $s \in Q \cup \Gamma \cup \{\#\}$

($x_{i,j,s} = 1$ iff cell (i,j) contains symbol s)

$$\phi = \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \phi_{\text{move}} \wedge \boxed{\phi_{\text{accept}}}$$

$\boxed{\phi_{\text{accept}}}$: some cell contains an accept state q_{accept}

$$= \bigvee_{1 \leq i, j \leq n^c} x_{i,j, q_{\text{accept}}}$$



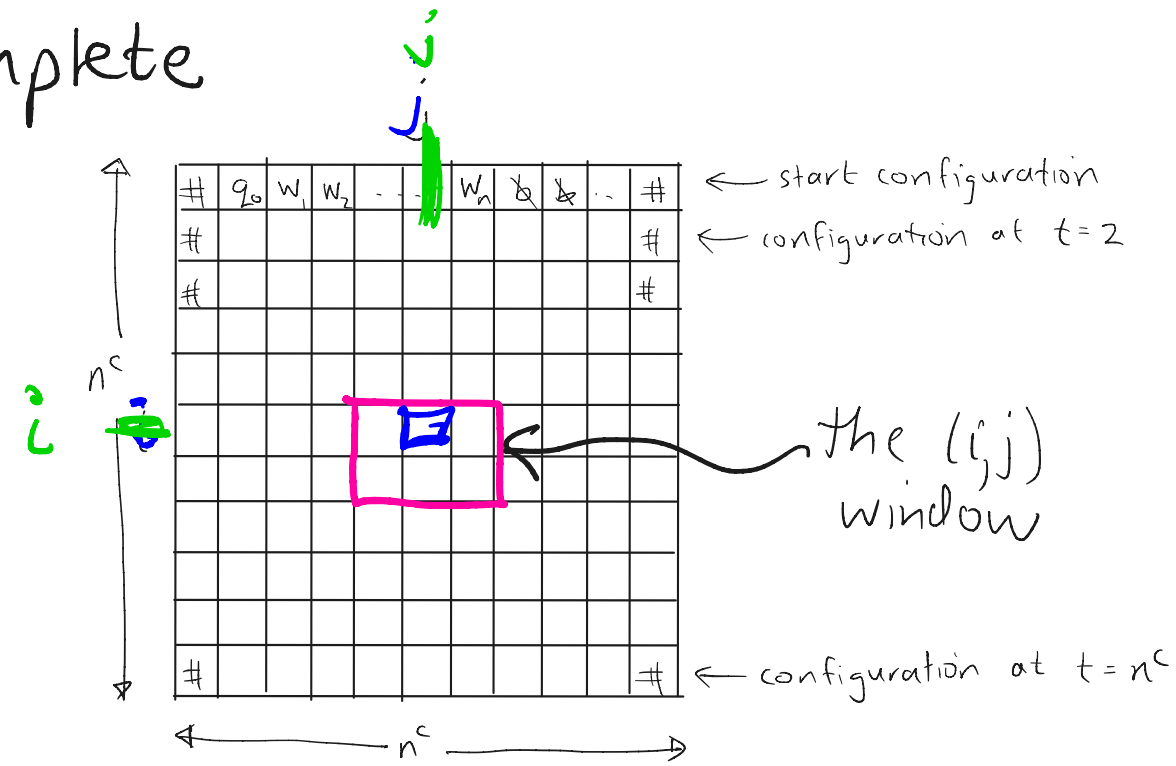
Cook-Levin Theorem : K SAT is NP-complete

Variables of ϕ : $x_{i,j,s}$, $i,j \in \{1, \dots, n^c\}$, $s \in Q \cup \Gamma \cup \{\#\}$

$$(x_{i,j,s} = 1 \text{ iff cell } (i,j) \text{ contains symbol } s)$$

$$\phi = \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \boxed{\phi_{\text{move}}} \wedge \phi_{\text{accept}}$$

$\phi_{\text{move}} :$ $\bigwedge_{1 \leq i, j \leq n^c}$ (the (i, j) window is legal)



that is, $\forall (i, j)$ the cell (i, j) is consistent with a legal transition from the configuration at time $i-1$.

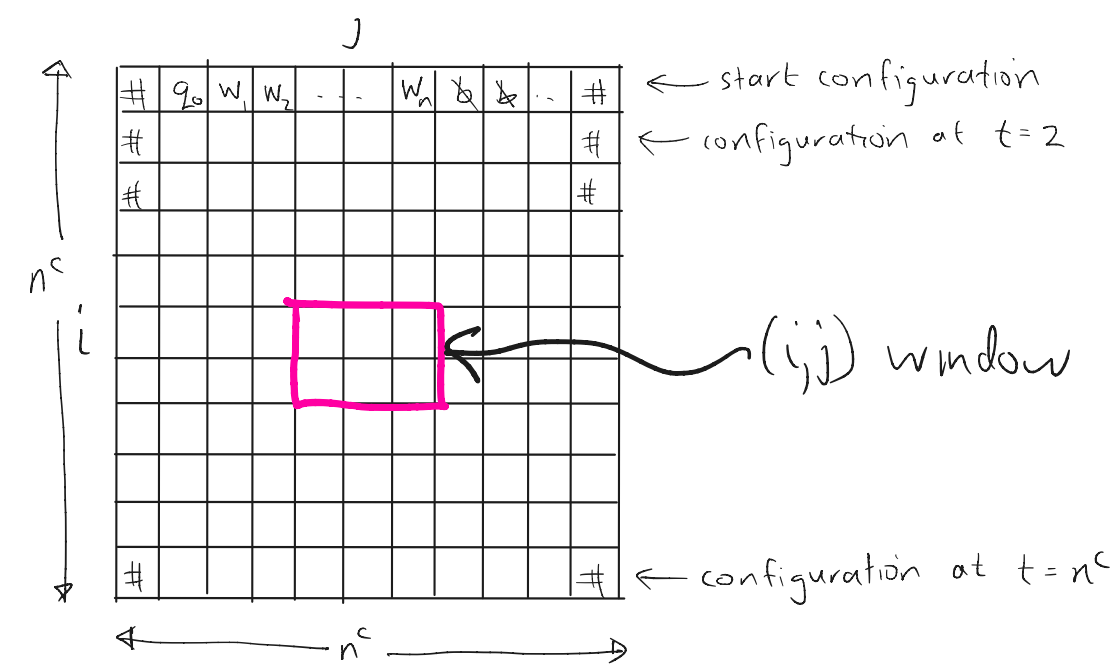
Cook-Levin Theorem : KSAT is NP-complete

Variables of ϕ : $x_{i,j,s}$, $i,j \in \{1, \dots, n^c\}$, $s \in Q \cup \Gamma \cup \{\#\}$

($x_{i,j,s} = 1$ iff cell (i,j) contains symbol s)

$$\phi = \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \boxed{\phi_{\text{move}}} \wedge \phi_{\text{accept}}$$

$$\boxed{\phi_{\text{move}}} : \bigwedge_{1 \leq i,j \leq n^c} (\text{the } (i,j) \text{ window is legal})$$



key idea: we can check that entire tableau is legal by locally checking every 2×3 window

A 2×3 window is legal if it doesn't violate N 's transition function.

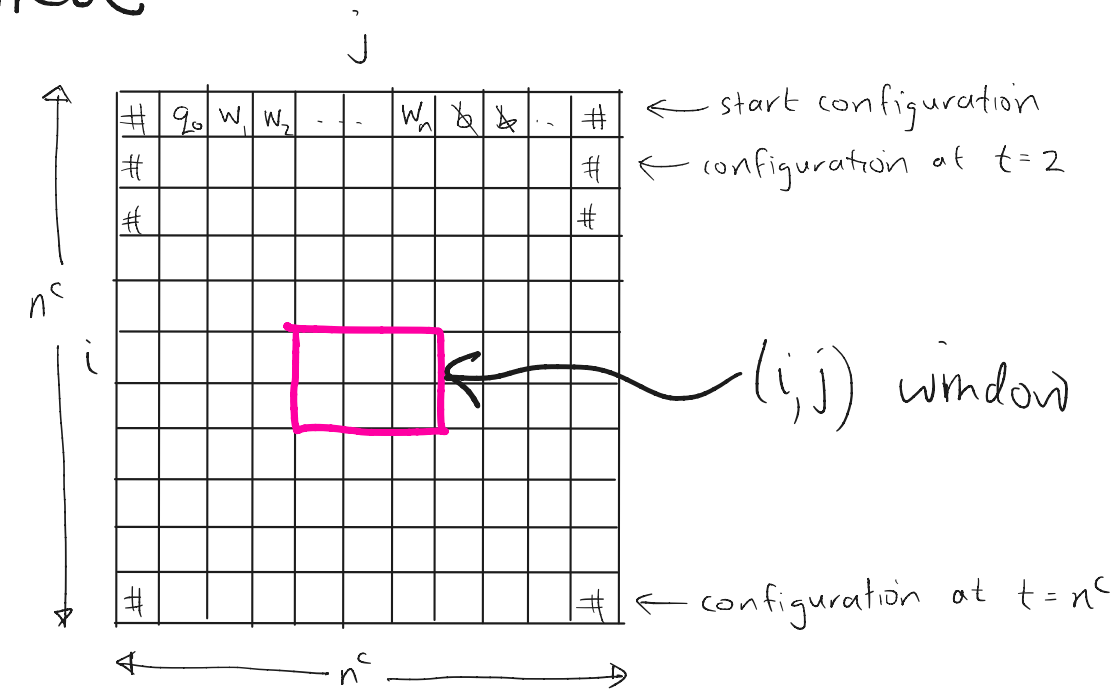
Cook-Levin Theorem : KSAT is NP-complete

Variables of ϕ : $x_{i,j,s}$, $i,j \in \{1, \dots, n^c\}$, $s \in Q \cup \Gamma \cup \{\#\}$

($x_{i,j,s} = 1$ iff cell (i,j) contains symbol s)

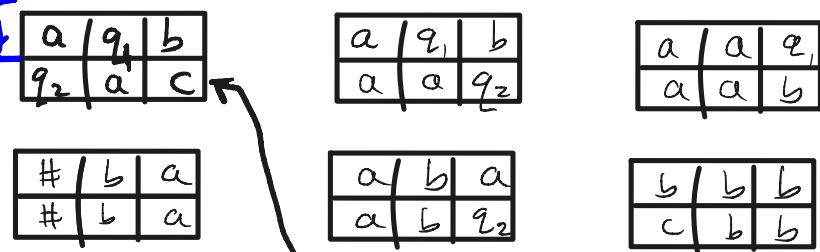
$$\phi = \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \boxed{\phi_{\text{move}}} \wedge \phi_{\text{accept}}$$

$$\boxed{\phi_{\text{move}}} : \bigwedge_{1 \leq i,j \leq n^c} (\text{the } (i,j) \text{ window is legal})$$

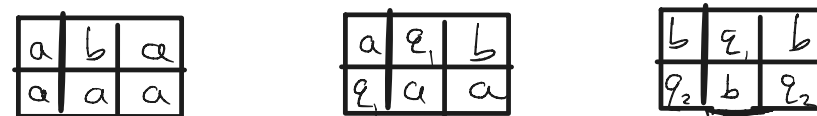


Example : $\delta(q_1, a) = \{(q_1, b, R)\}$, $\delta(q_1, b) = \{(q_2, c, L), (q_2, a, R)\}$

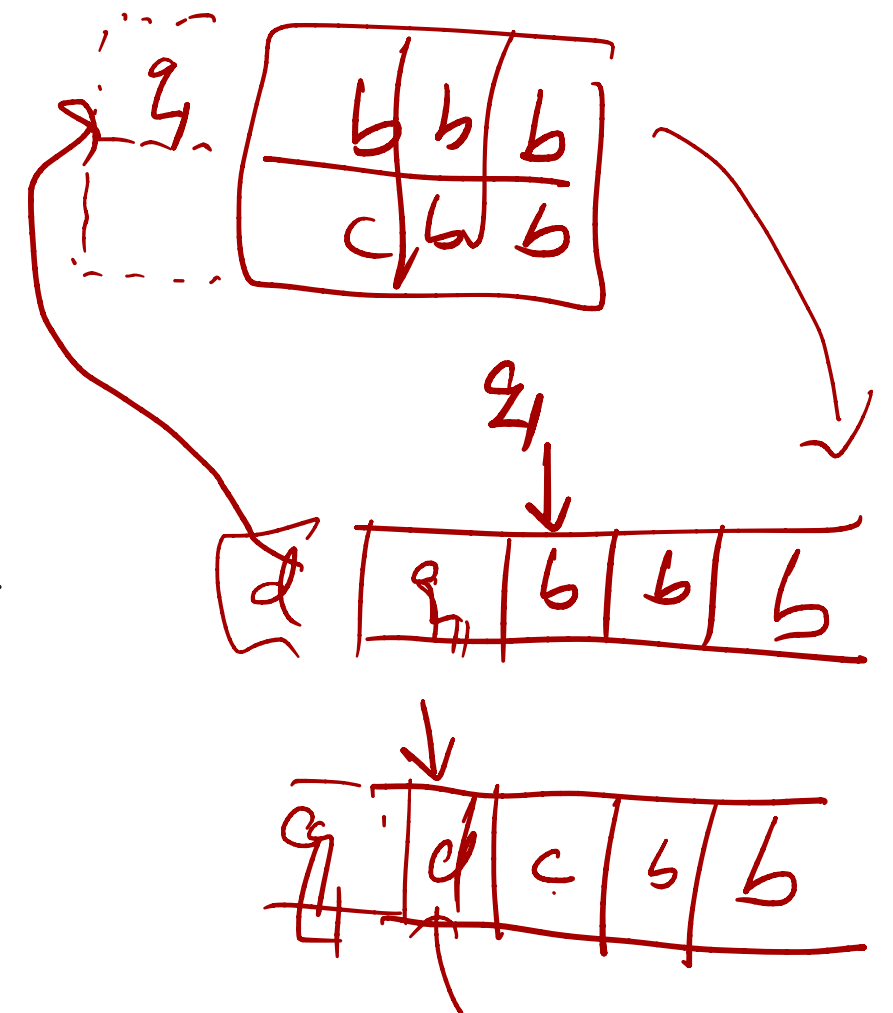
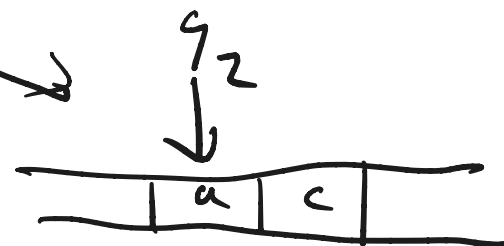
Legal windows :



Some Illegal Windows :



corresponds to : [a, b]



	$j=3$	$j=4$	$j=5$
$i=2$	a	b	a
$i=3$	a	a	a

Specifically not legal by:

$$\left(\overline{X_{2,3,a}} \vee \overline{X_{2,4,b}} \vee \overline{X_{2,5,a}} \vee \overline{X_{3,3,a}} \vee \overline{X_{3,4,a}} \vee \overline{X_{3,5,a}} \right)$$

$$\underbrace{\phi(x_1 \dots x_n)}$$

$$\phi(\vec{x})$$

CNF underlying
variables $x_1 \dots x_n$

$$C_1 \wedge C_2 \wedge \dots \wedge C_m$$

$$C_i : (x_1 \vee \bar{x}_2 \vee x_7 \vee \dots)$$



Cook-Levin Theorem : KSAT is NP-complete

Variables of ϕ : $x_{i,j,s}$, $i,j \in \{1, \dots, n^c\}$, $s \in Q \cup \Gamma \cup \{\#\}$

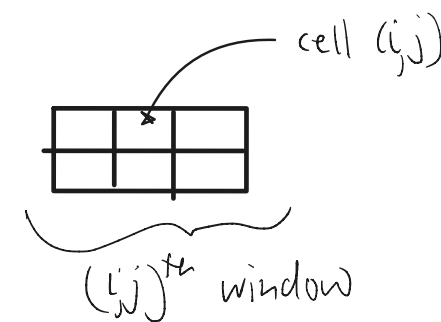
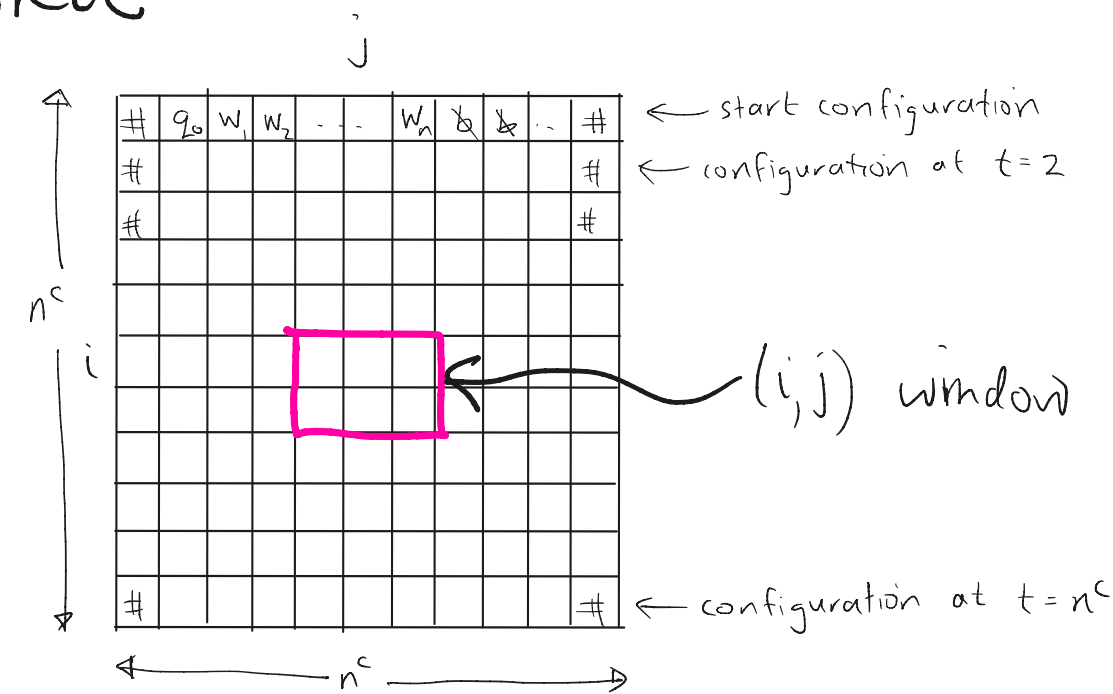
($x_{i,j,s} = 1$ iff cell (i,j) contains symbol s)

$$\phi = \phi_{\text{cell}} \wedge \phi_{\text{start}} \wedge \boxed{\phi_{\text{move}}} \wedge \phi_{\text{accept}}$$

$$\boxed{\phi_{\text{move}}} : \bigwedge_{1 \leq i,j \leq n^c} (\text{the } (i,j) \text{ window is legal})$$

$$\phi_{\text{move}} = \bigwedge_{1 \leq i,j \leq n^c} (\text{the } (i,j)^{\text{th}} \text{ window is legal})$$

$$= \bigwedge_{1 \leq i,j \leq n^c} \bigvee_{\substack{a_1, a_2, \dots, a_6 \\ \text{is a legal} \\ \text{window}}} (x_{i-1,j,a_1} \wedge x_{i,j,a_2} \wedge x_{i+1,j,a_3} \wedge x_{i-1,j+1,a_4} \wedge x_{i,j+1,a_5} \wedge x_{i+1,j+1,a_6})$$



So we've defined $f: w \rightarrow \phi(\vec{x})$ $\phi = \phi_{\text{start}} \wedge \phi_{\text{env}} \wedge \phi_{\text{mem}} \wedge \phi_{\text{accept}}$

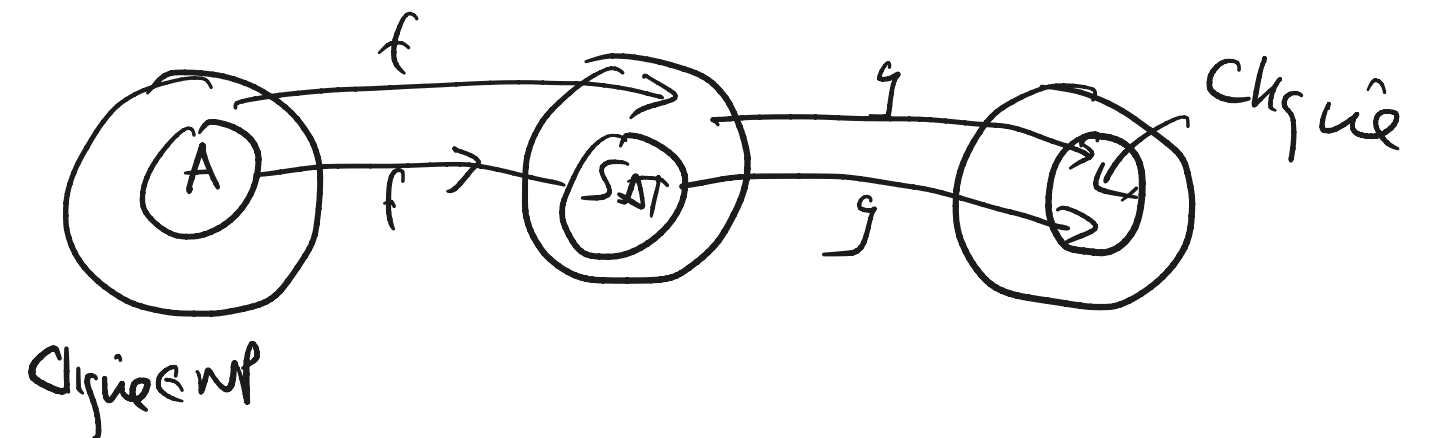
such that :

- ① $f(w) = \phi$ Building ϕ is polytime
- ② ϕ is satisfiable (there is an assignment to variables $x_{i,j,s}$)
iff there is an accepting computation (tableaux)
of N on w iff $w \in A$

↖ see proof in book

Last class we showed these languages NP-Complete
assuming SAT NP-COMPLETE :

- CLIQUE
- INDEPENDENT SET
- HITTING SET



OTHER NP-COMPLETE PROBLEMS :

- COLORING
- SUBSET SUM
- HAMILTONIAN PATH

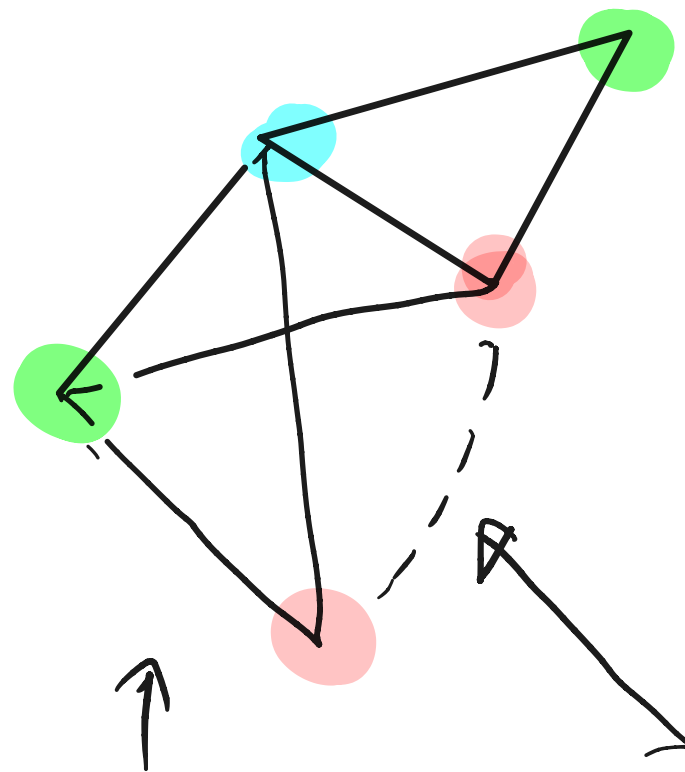
} Proofs of NP-completeness
in book

COLORING (OR GRAPH-COLOR)

Input: (g, k)

Accept if we can properly color g using colors $1, 2, \dots, k$

no 2 adjacent
vertices have same color



↑
3 colorable

with this edge added
need 4 colors

Theorem COLORING is NP-COMplete

(1) in NP (did earlier)

(2) NP-hard

COLORING IS NP-HARD

We will show $3SAT \leq_P \text{Coloring}$

$f(\phi) \longrightarrow \langle G_\phi, 3 \rangle$ where G_ϕ is a graph

3CNF
n vars
m clauses

m.n

such that

ϕ is satisfiable iff

G_ϕ is 3-colorable.

$(x_1 \vee x_2 \vee x_3) \leftarrow$ free constraint

that ~~is~~ a satisfying ass.

$(\bar{x}_1 \vee x_4) \quad x_1 = 1 \quad x_4 = 0$

~~is~~ can't have $x_1 = 0 \quad x_2 = 0 \quad \& \quad x_3 = 0$

COLORING IS NP-HARD

We will show $3SAT \leq_m \text{Coloring}$

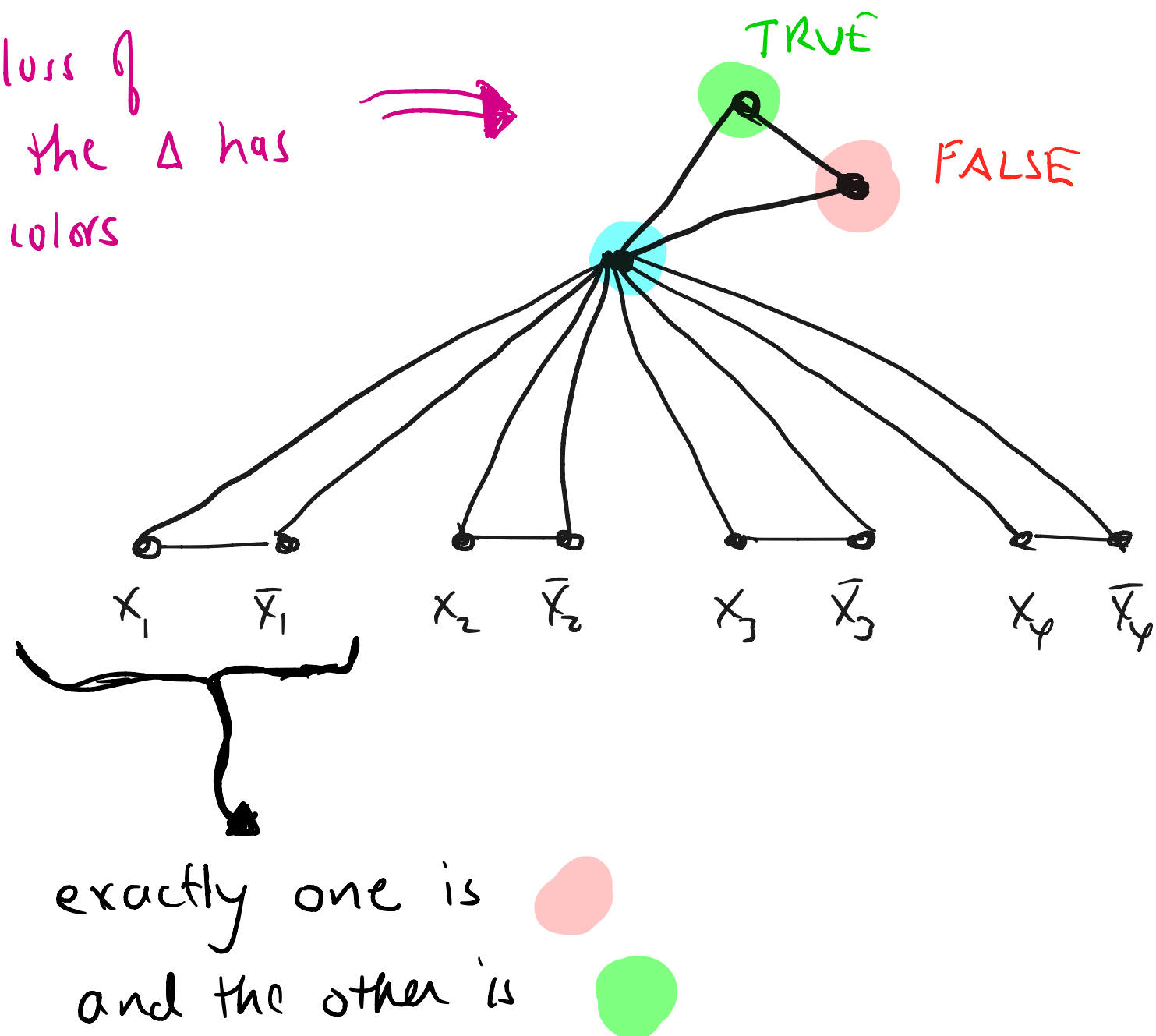
$$\begin{array}{l} f(\phi) \\ \text{3CNF} \\ n \text{ vars} \\ m \text{ clauses} \end{array} \longrightarrow \langle G_\phi, 3 \rangle \quad \text{where } G_\phi \text{ is a graph}$$

Example $\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_4) \wedge (\bar{x}_2 \vee x_3)$

We want to simulate 0/1 assignment to the variables
using 3 colors

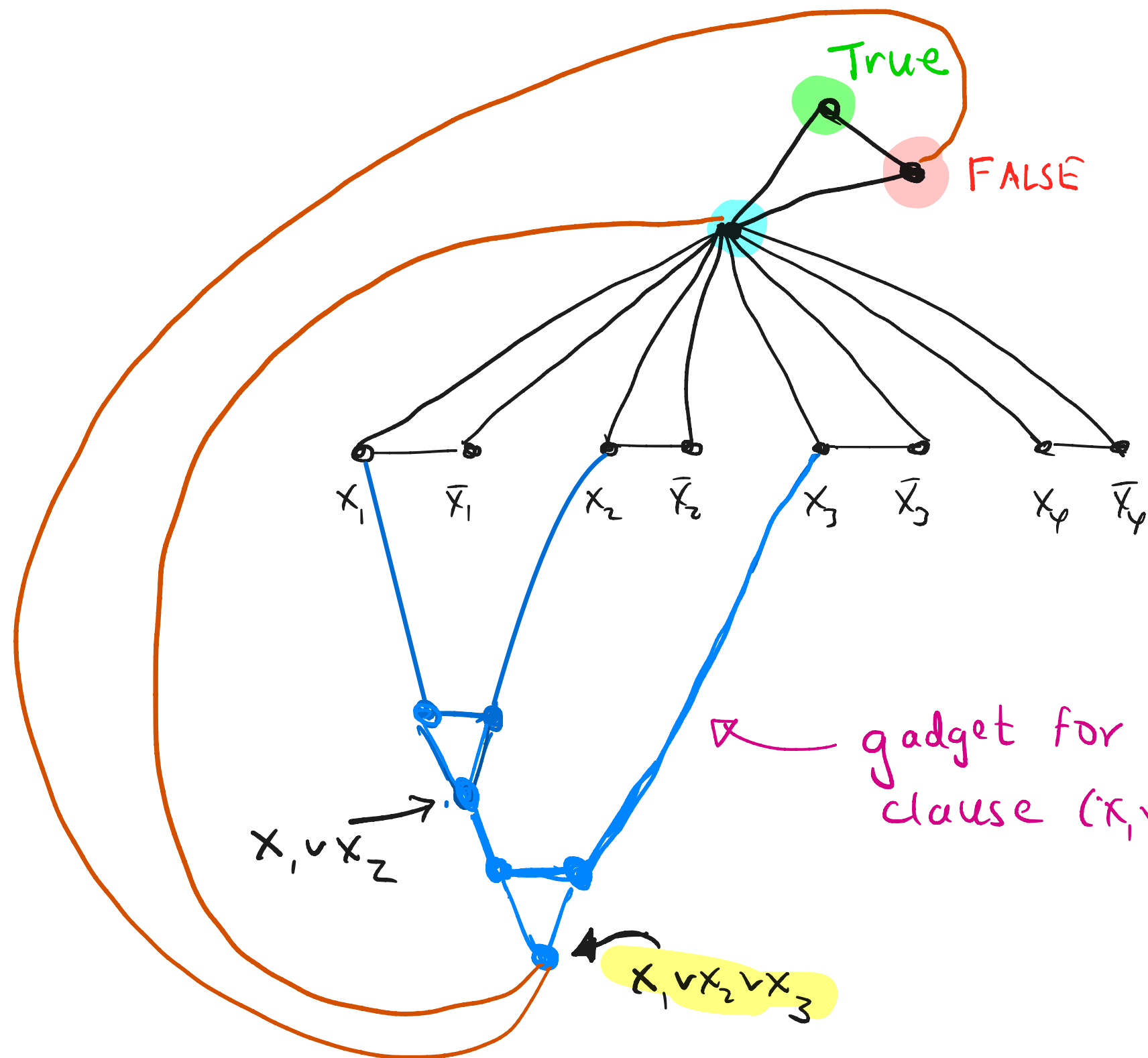
Example $\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_4) \wedge (\bar{x}_2 \vee x_3)$

Without loss of generality the Δ has these 3 colors



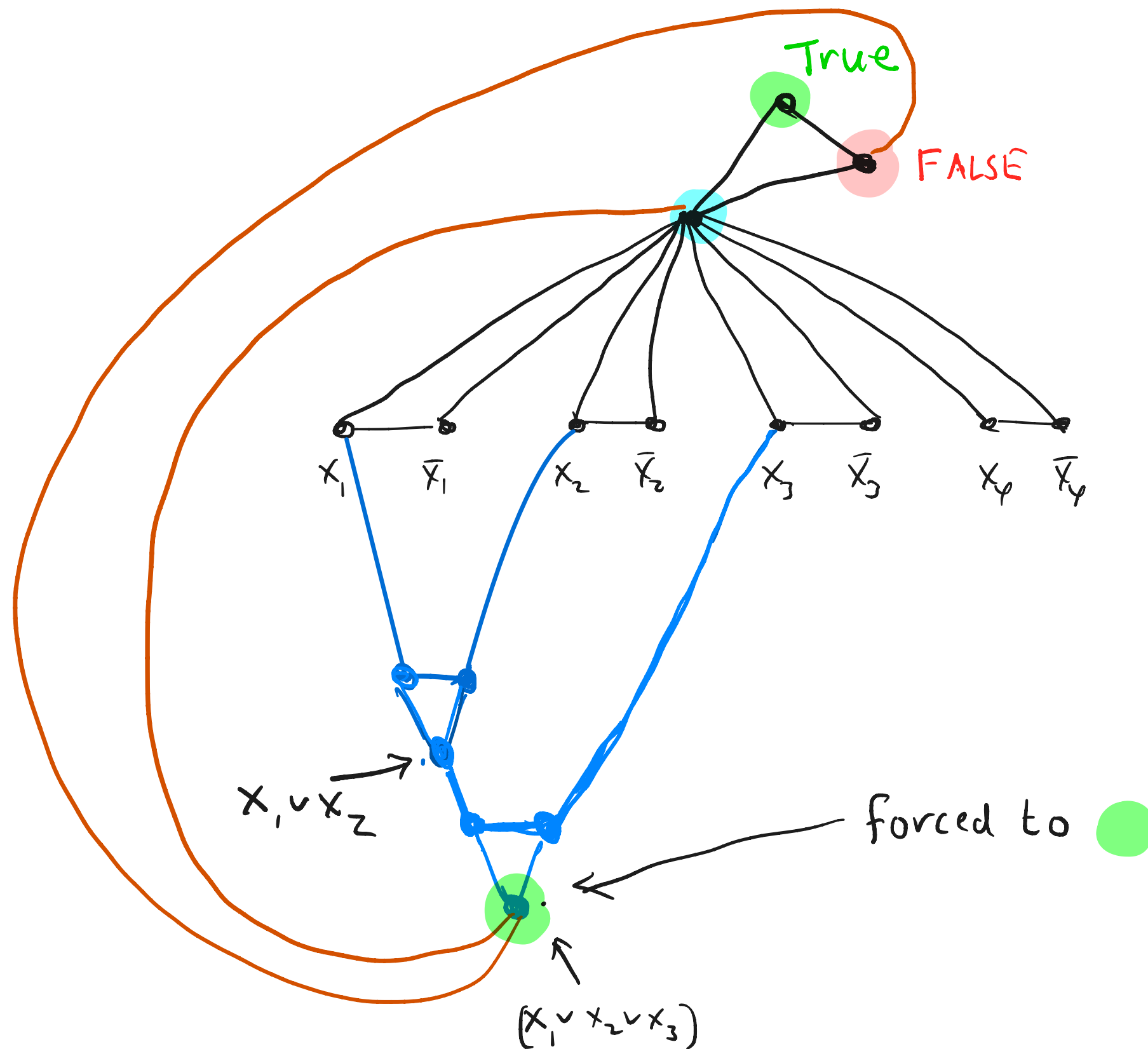
"gadget" that forces colors to correspond to 0/1 assignment

Example $\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_4) \wedge (\bar{x}_2 \vee x_3)$



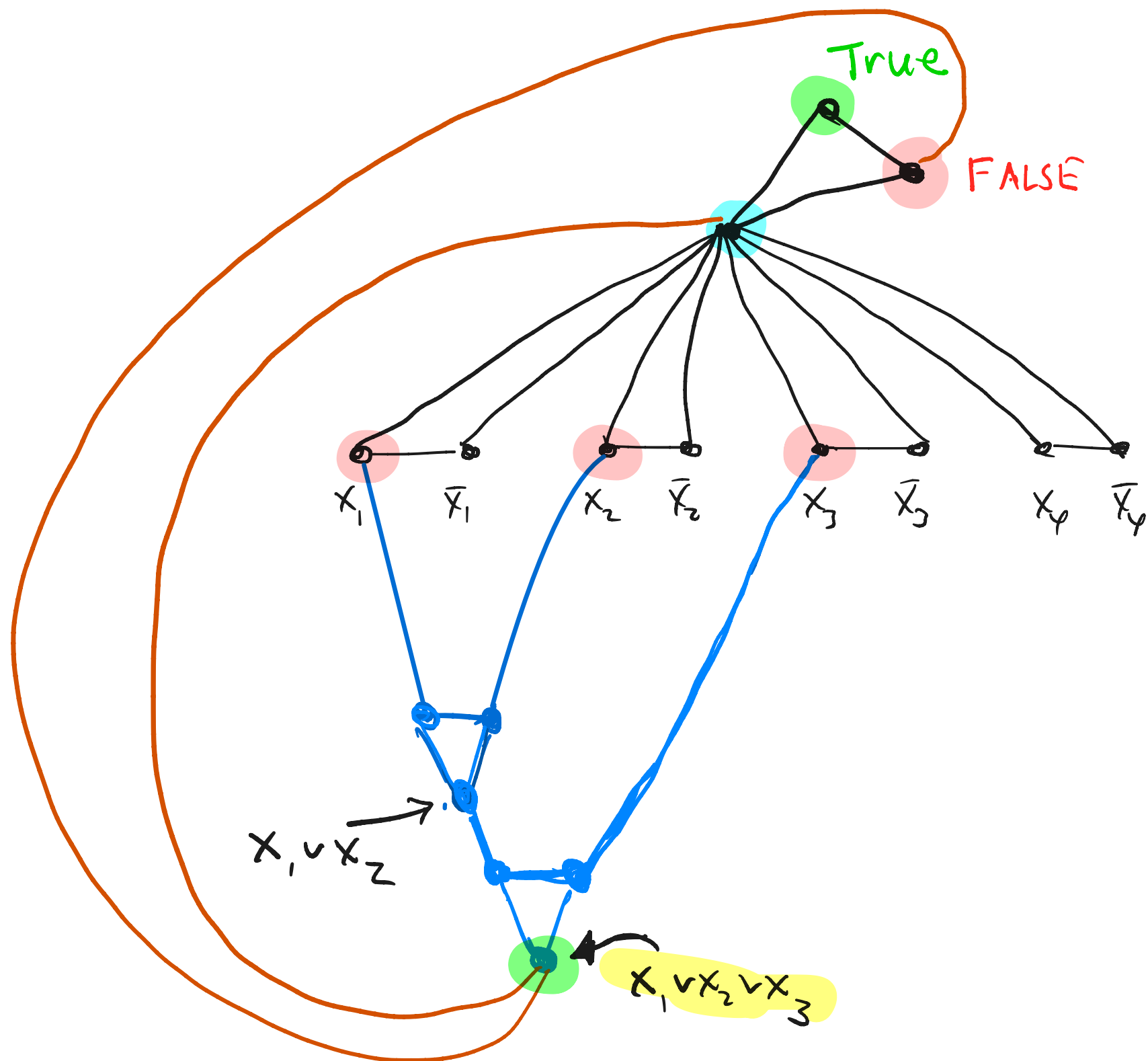
Next :
Clause "gadgets"
to force constraints
on variables

Example $\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_4) \wedge (\bar{x}_2 \vee x_3)$



Next :
 Clause "gadgets"
 to force constraints
 on variables

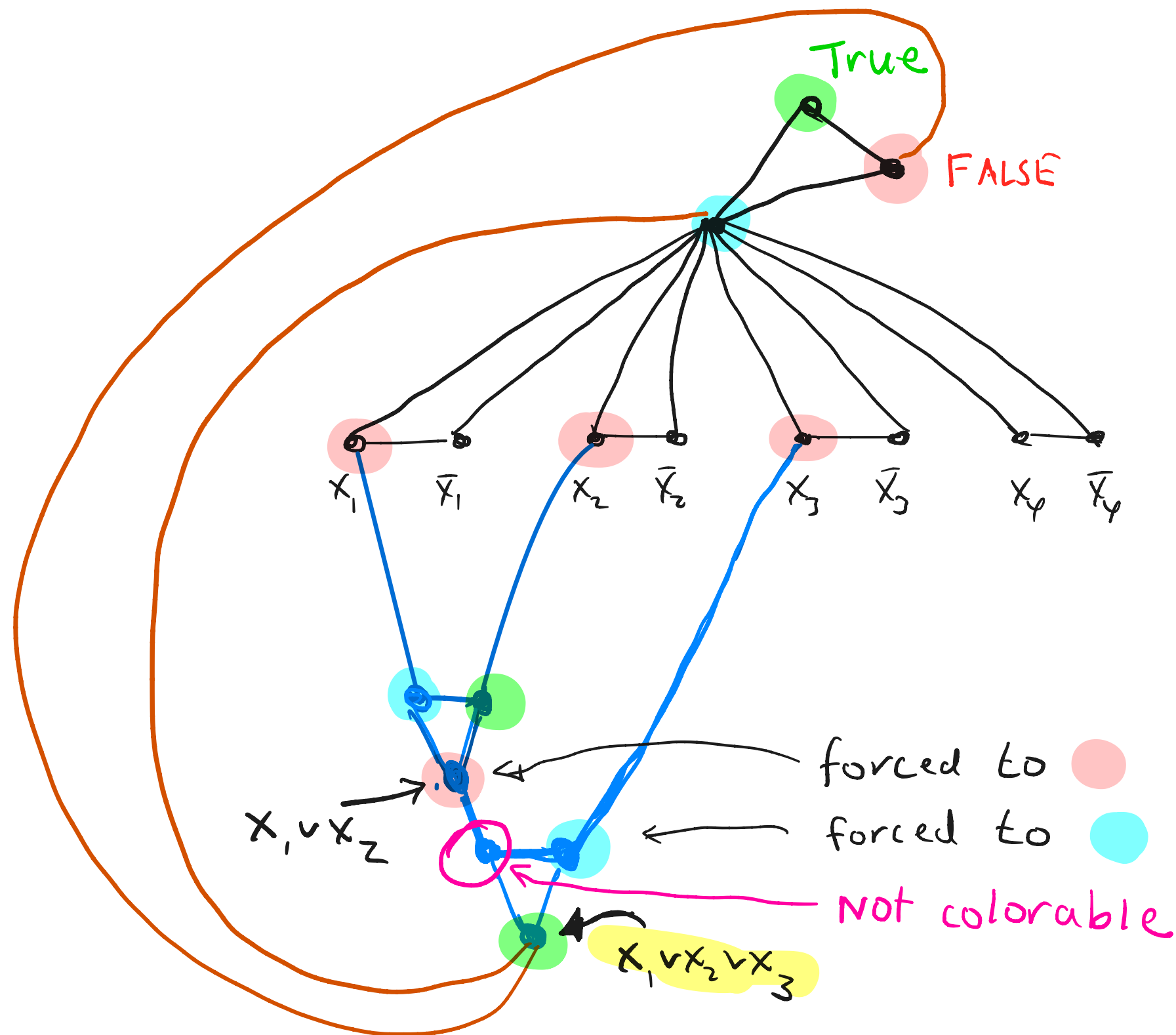
Example $\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_4) \wedge (\bar{x}_2 \vee x_3)$



Next :
Clause "gadgets"
to force constraints
on variables

Bad case :
 x_1, x_2, x_3 all ●

Example $\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_4) \wedge (\bar{x}_2 \vee x_3)$



Next :

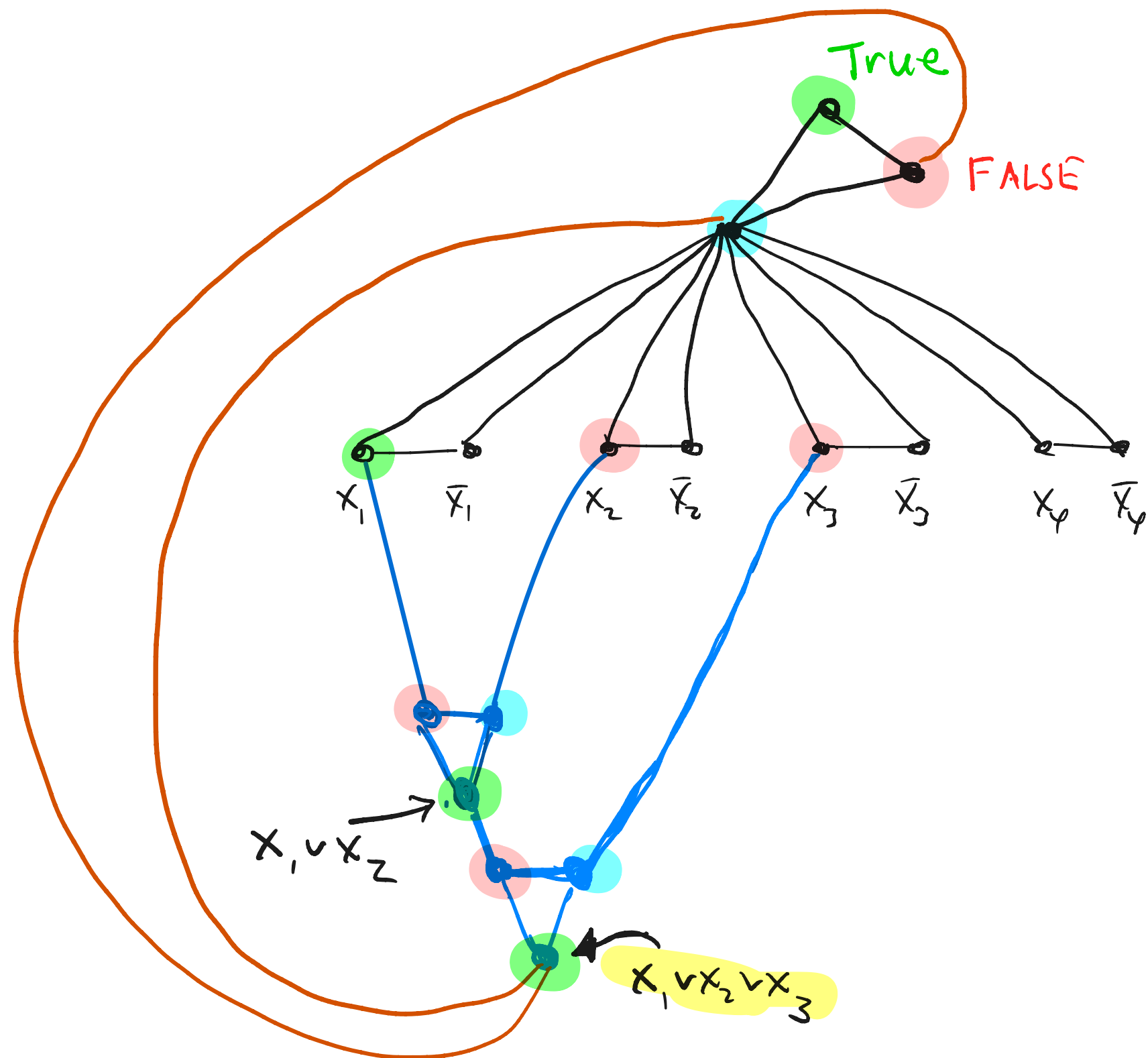
Clause "gadgets"
to force constraints
on variables

Bad case:

x_1, x_2, x_3 all ●

$\therefore$ If all literals set to ●
(all false), not
3-colorable

Example $\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_4) \wedge (\bar{x}_2 \vee x_3)$

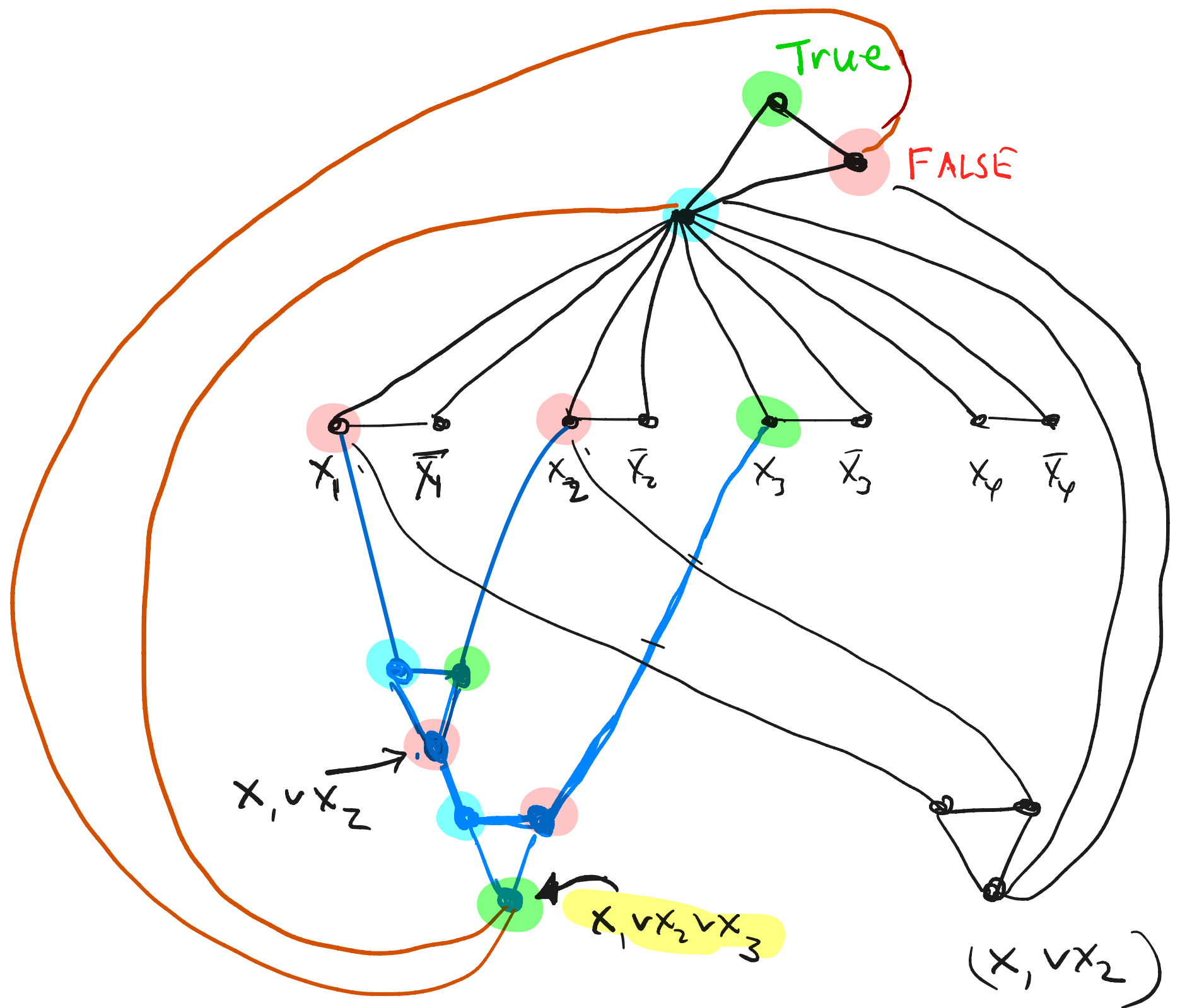


* As long as one of the 3 literals is ● we can properly 3-color this gadget

Example:

x_1 ●
 x_2, x_3 ●

Example $\phi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_4) \wedge (\bar{x}_2 \vee x_3)$



* As long as one of the 3 literals is ● we can properly 3-color this gadget

Example 2

x_1, x_2 ●

x_3 ●

Use the same gadget trick for each clause
to build G_ϕ

- Can Build G_ϕ from ϕ in polytime
- Can show G_ϕ is 3-colorable iff
 ϕ has a satisfying assignment

Subset SUM is NP COMPLETE

Input $\langle S, t \rangle$ where S is a set of positive integers, t is a positive integer.

Accept iff there is a subset of S that sums to t

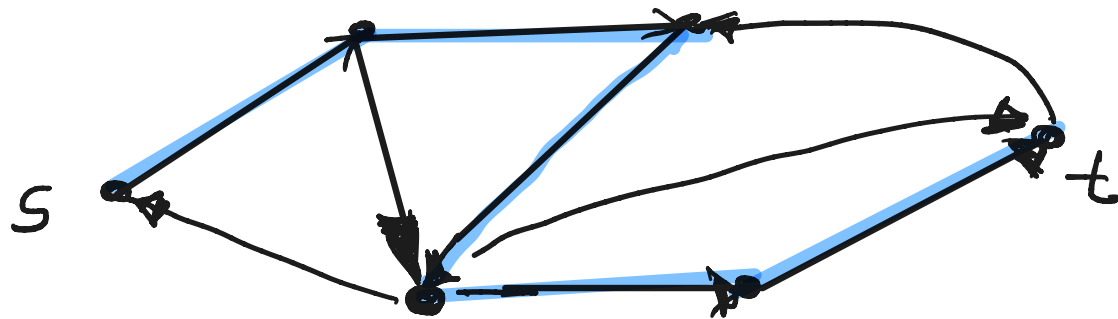
Example $S = \{1, 2, 2, 5, 13, 15\}$ $t = 17$ ✓

$t = 11$ ✗

3SAT $\rightarrow$ Subset Sum

$\boxed{\text{HAMPATH}} = \{ (g, s, t) \mid g \text{ is a directed graph containing a Hamilton path (visits all vertices once) from } s \text{ to } t \}$

Example



Verifier V on input (g, s, t, p) :

check if p encodes a Hamiltonian path from s to t .
If yes $\rightarrow$ accept; otherwise $\rightarrow$ reject

Theorem HAMPATH is NP-complete

Proof

1. HAMPATH in NP (already did this)

2. We will show $3SAT \leq_p \text{HAMPATH}$ (and thus HAMPATH is NP-hard)

Let $\phi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \dots \wedge (a_m \vee b_m \vee c_m)$ each a_i, b_i, c_i is a literal.

$f: \phi \rightarrow (g_\phi, s, t)$

$\text{HAMPATH} = \left\{ (g, s, t) \mid \begin{array}{l} g \text{ is a directed graph} \\ \text{with a hamiltonian path} \\ \text{(visits every vertex in } g \text{ exactly once)} \\ \text{from } s \text{ to } t \end{array} \right\}$

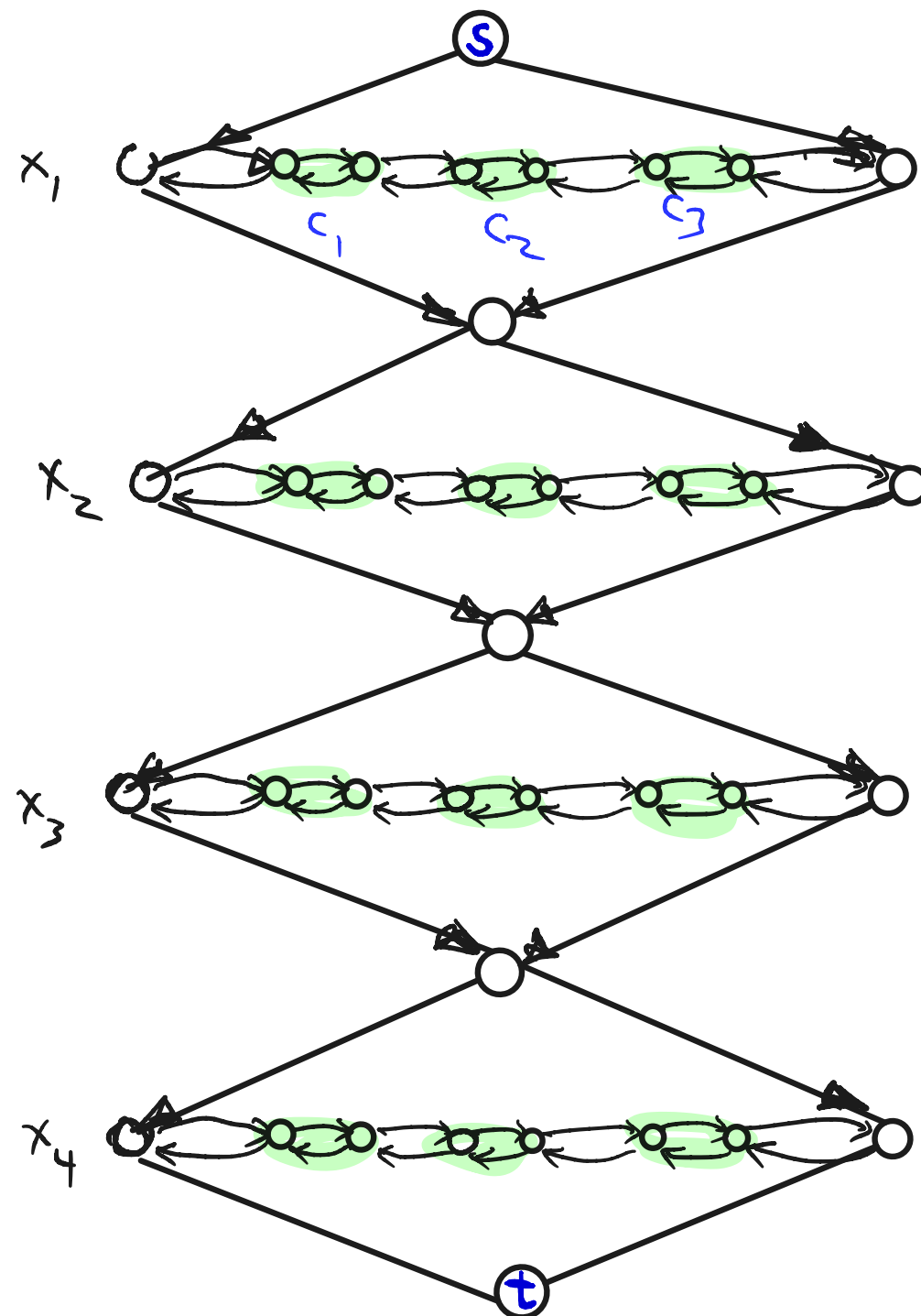
$$\text{Let } \phi = \underbrace{(x_1 \vee x_3 \vee \bar{x}_4)}_{c_1} \wedge \underbrace{(x_2 \vee \bar{x}_3 \vee x_4)}_{c_2} \wedge \underbrace{(\bar{x}_1 \vee x_2 \vee x_3)}_{c_3}$$

$$f: \phi \rightarrow (g_\phi, s, t)$$

$$n = \# \text{ vars} = 4$$

$$m = \# \text{ clauses} = 3$$

g_ϕ :



c_1

c_2

c_3

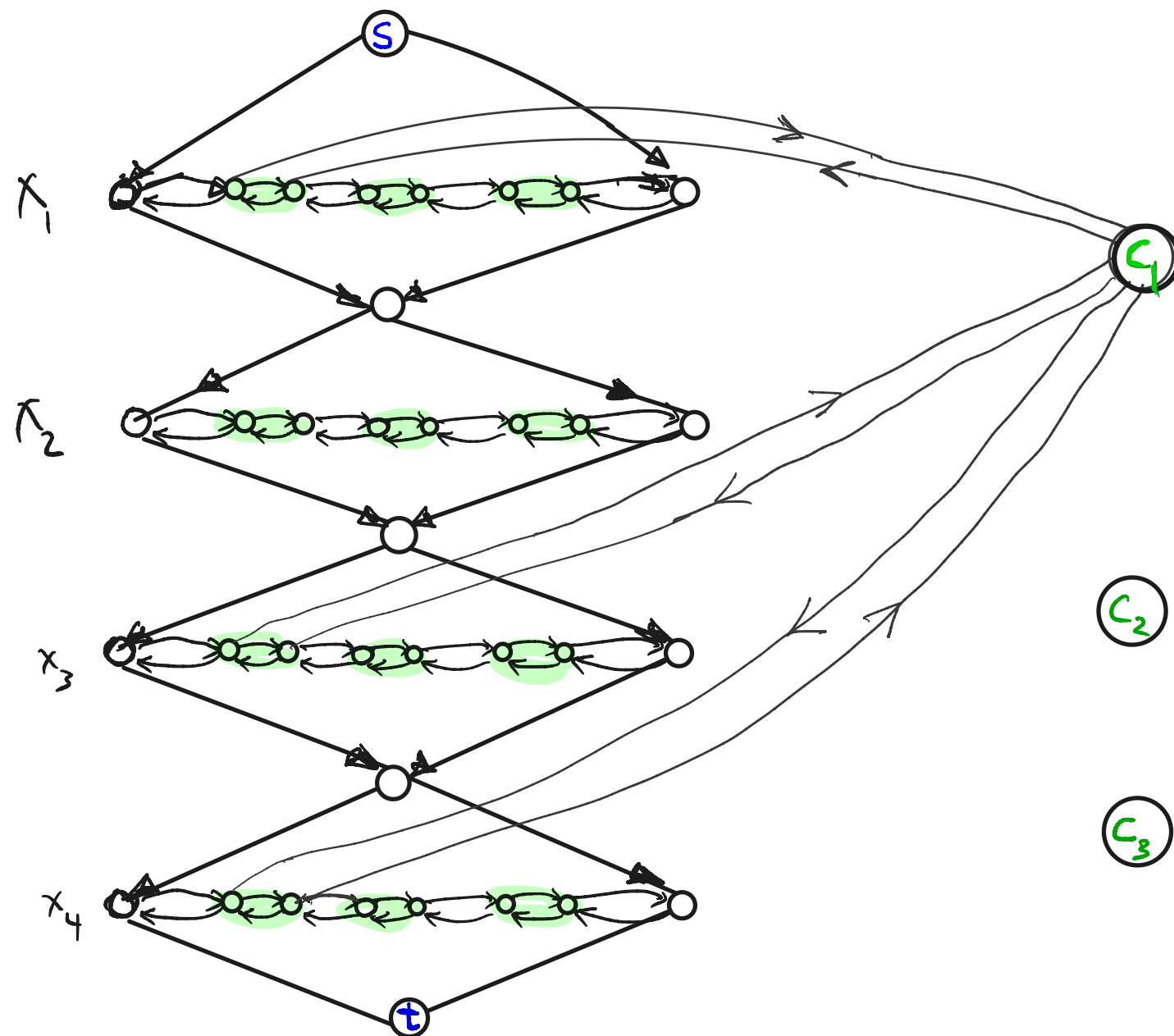
$$\text{Let } \phi = \underbrace{(x_1 \vee x_3 \vee \bar{x}_4)}_{c_1} \wedge \underbrace{(x_2 \vee \bar{x}_3 \vee x_4)}_{c_2} \wedge \underbrace{(\bar{x}_1 \vee x_2 \vee x_3)}_{c_3}$$

$$f: \phi \rightarrow (g_\phi, s, t)$$

$$n = \# \text{ vars} = 4$$

$$m = \# \text{ clauses} = 3$$

g_ϕ :



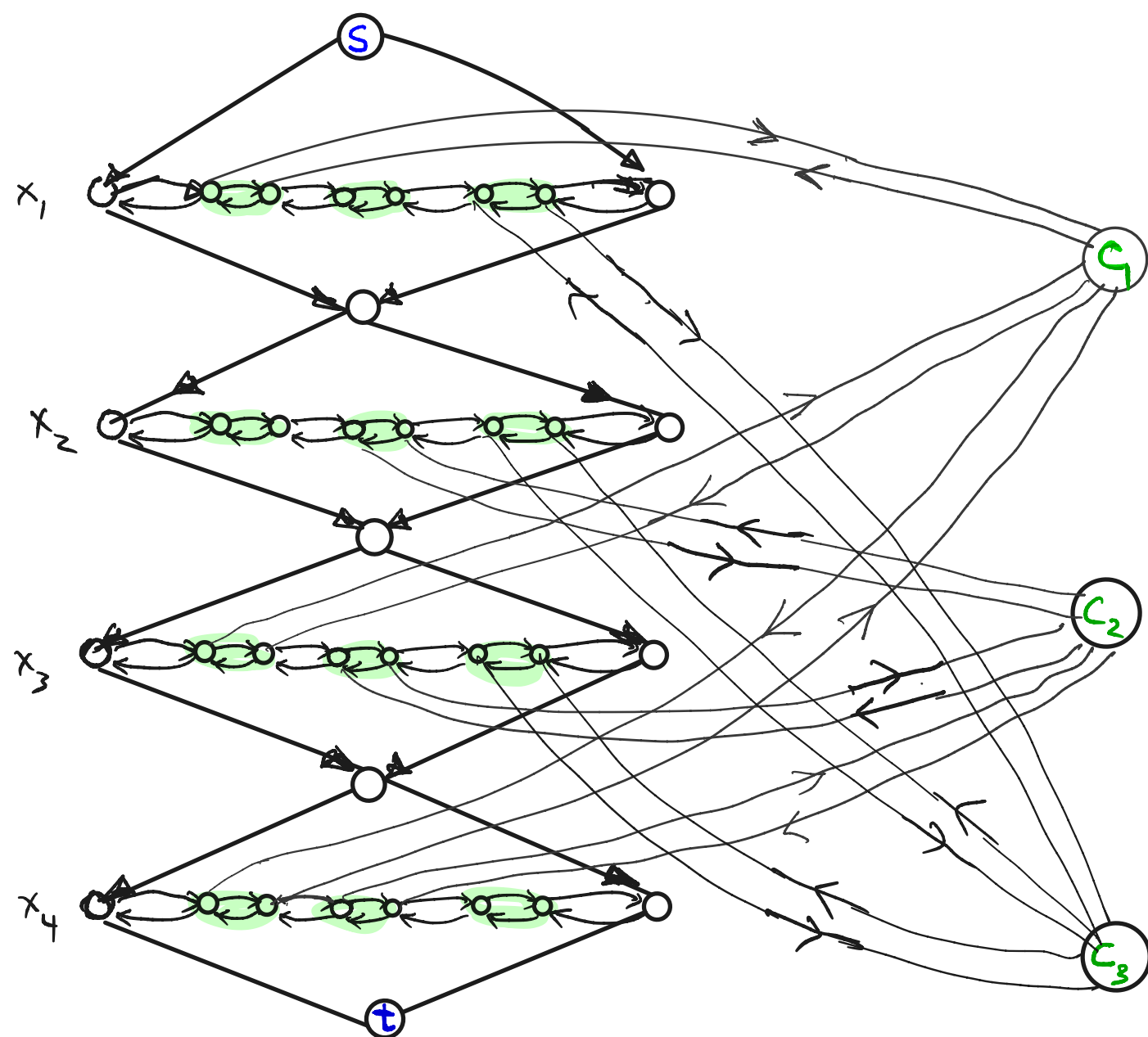
$$\text{Let } \phi = \underbrace{(x_1 \vee x_3 \vee \bar{x}_4)}_{C_1} \wedge \underbrace{(x_2 \vee \bar{x}_3 \vee x_4)}_{C_2} \wedge \underbrace{(\bar{x}_1 \vee x_2 \vee x_3)}_{C_3}$$

$$f: \phi \rightarrow (g_\phi, s, t)$$

$$n = \# \text{ vars} = 4$$

$$m = \# \text{ clauses} = 3$$

g_ϕ :



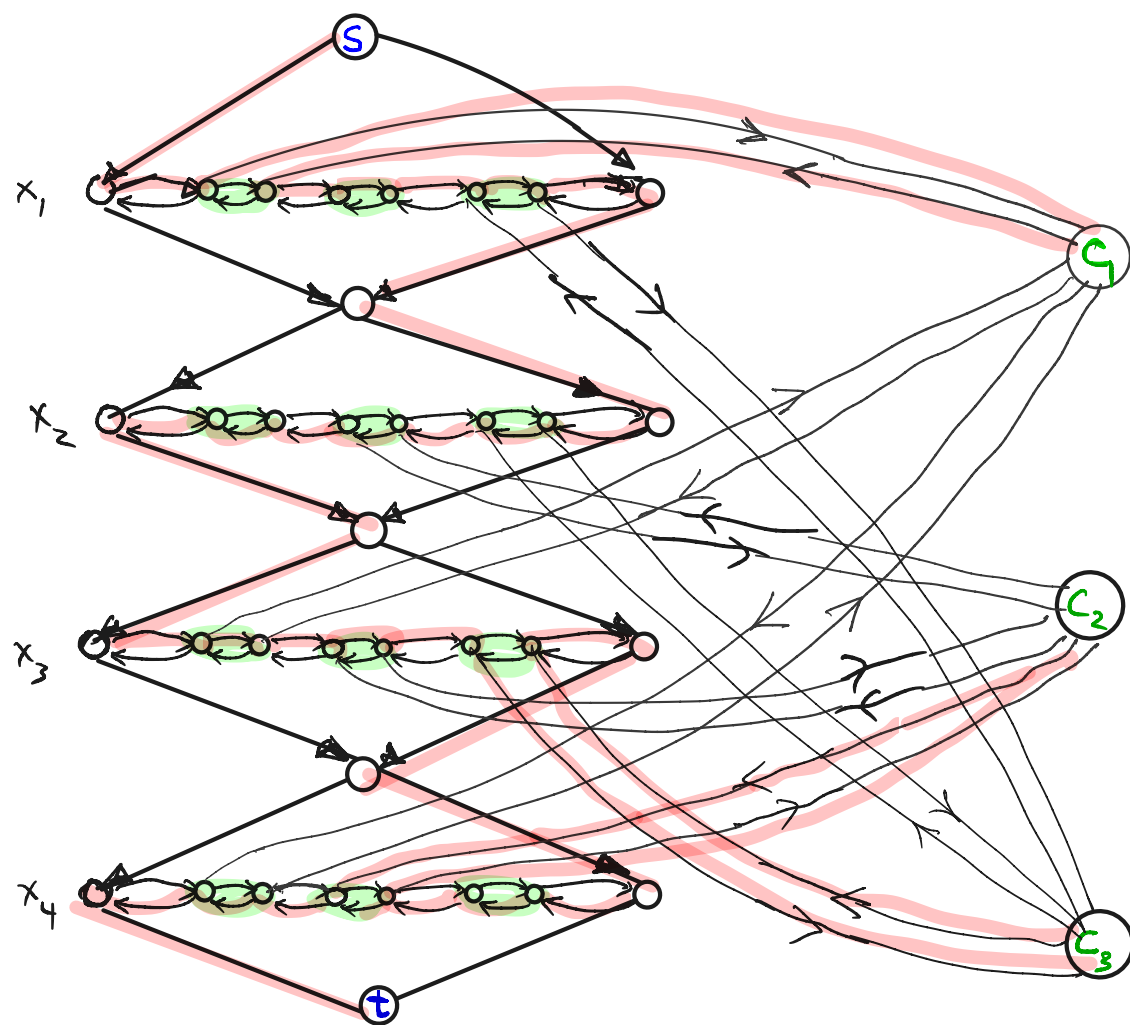
$$\text{Let } \phi = \underbrace{(x_1 \vee x_3 \vee \bar{x}_4)}_{C_1} \wedge \underbrace{(x_2 \vee \bar{x}_3 \vee \bar{x}_4)}_{C_2} \wedge \underbrace{(\bar{x}_1 \vee x_2 \vee x_3)}_{C_3}$$

$$f: \phi \rightarrow (g_\phi, s, t)$$

$$n = \# \text{ vars} = 4$$

$$m = \# \text{ clauses} = 3$$

g_ϕ :



Claim ϕ is satisfiable iff g_ϕ has a Hamiltonian path from s to t

$$\alpha: \underbrace{x_1=1}_{C_1} \quad x_2=0 \quad \underbrace{x_3=1}_{C_3} \quad \underbrace{x_4=0}_{C_2}$$