METHODS FOR EFFICIENT SAT SOLVING

Aalok Thakkar

Indian Statistical Institute, Kolkata

October 28, 2024

METHODS FOR EFFICIENT SAT SOLVING

Aalok Thakkar

Our lab is offering PhD, visiting researcher, and internship positions.
Contact me at aalok.thakkar@ashoka.edu.in for more details.

Indian Statistical Institute, Kolkata

October 28, 2024

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

What are $\vee$, $\leftarrow$, $\rightarrow$, $\neg$, $\oplus$, $\leftrightarrow$?

What are variables (atoms)?

What are assignments (models)?

What does satisfaction mean?

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

Is there a satisfying assignment?

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

Is there a satisfying assignment?

Check all assignments!

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

Is there a satisfying assignment?

Check all assignments!

Can we do better?

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

Is there a satisfying assignment?

Check all assignments!

Can we do something smarter?

Conjunctive Normal Form (CNF)

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

Conjunctive Normal Form (CNF)

$$\{A \lor B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \lor B, A \lor \neg(\neg C \lor D), \neg((D \lor A) \land \neg(D \land A)), (B \land D) \lor (\neg B \land \neg D)\}$$

$$p = c_1 \land c_2 \land \ldots \land c_m$$

$$c_i = l_{i,1} \lor l_{i,2} \lor \ldots \lor l_{i,n_i}$$

$l_{i,j} = A$ or $l_{i,j} = \neg A$

Conjunctive Normal Form (CNF)

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

$$p = c_1 \wedge c_2 \wedge \ldots \wedge c_m$$

$$c_i = l_{i,1} \vee l_{i,2} \vee \ldots \vee l_{i,n_i}$$

$$l_{i,j} = A \text{ or } l_{i,j} = \neg A$$

Normalisation Theorem: For every propositional formula, there exists an *equivalent* formula in conjunctive normal form.

Conjunctive Normal Form (CNF)

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

$$p = c_1 \wedge c_2 \wedge \ldots \wedge c_m$$

$$c_i = l_{i,1} \vee l_{i,2} \vee \ldots \vee l_{i,n_i}$$

$$l_{i,j} = A \text{ or } l_{i,j} = \neg A$$

Tseitin's Theorem: For every propositional formula, there exists a <u>polynomial size</u> *equisatisfiable* formula in conjunctive normal form.

Conjunctive Normal Form (CNF)

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

$A \vee B$

Conjunctive Normal Form (CNF)

$$\{A \lor B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \lor B, A \lor \neg(\neg C \lor D), \neg((D \lor A) \land \neg(D \land A)), (B \land D) \lor (\neg B \land \neg D)\}$$

$A \lor B$

$A \lor \neg(\neg C \lor D)$

Conjunctive Normal Form (CNF)

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

$$A \vee B$$

$$A \vee \neg(\neg C \vee D) \implies A \vee (C \wedge \neg D) \implies (A \vee C) \wedge (A \vee \neg D)$$

Conjunctive Normal Form (CNF)

$$\{A \lor B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \lor B, A \lor \neg(\neg C \lor D), \neg((D \lor A) \land \neg(D \land A)), (B \land D) \lor (\neg B \land \neg D)\}$$

$$A \lor B$$

$$A \lor \neg(\neg C \lor D) \implies A \lor (C \land \neg D) \implies (A \lor C) \land (A \lor \neg D)$$

$$\neg((D \lor A) \land \neg(D \land A))$$

Conjunctive Normal Form (CNF)

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

$A \vee B$

$$A \vee \neg(\neg C \vee D) \implies A \vee (C \wedge \neg D) \implies (A \vee C) \wedge (A \vee \neg D)$$

$$\neg((D \vee A) \wedge \neg(D \wedge A)) \implies (\neg D \wedge \neg A) \vee (D \wedge A) \implies$$

$$((\neg D \wedge \neg A) \vee D) \wedge ((\neg D \wedge \neg A) \vee A) \implies$$

$$(\neg D \vee D) \wedge (\neg A \vee D) \wedge (\neg D \vee A) \wedge (\neg A \vee A)$$

Conjunctive Normal Form (CNF)

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

$$A \vee B$$

$$A \vee \neg(\neg C \vee D) \implies A \vee (C \wedge \neg D) \implies (A \vee C) \wedge (A \vee \neg D)$$

$$\neg((D \vee A) \wedge \neg(D \wedge A)) \implies (\neg D \wedge \neg A) \vee (D \wedge A) \implies$$

$$((\neg D \wedge \neg A) \vee D) \wedge ((\neg D \wedge \neg A) \vee A) \implies$$

$$(\neg D \vee D) \wedge (\neg A \vee D) \wedge (\neg D \vee A) \wedge (\neg A \vee A)$$

$$\ldots$$

Conjunctive Normal Form (CNF)

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

$$\{(A \vee B), (A \vee C) \wedge (A \vee \neg D), (\neg A \vee D) \wedge (\neg D \vee A), (B \vee \neg D) \wedge (D \vee \neg B)\}$$

$$\{(A \vee B), (A \vee C), (A \vee \neg D), (\neg A \vee D), (\neg D \vee A), (B \vee \neg D), (D \vee \neg B)\}$$

Can find a satisfying assignment in polynomial time?

Conjunctive Normal Form (CNF)

$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

$$\{(A \vee B), (A \vee C) \wedge (A \vee \neg D), (\neg A \vee D) \wedge (\neg D \vee A), (B \vee \neg D) \wedge (D \vee \neg B)\}$$

$$\{(A \vee B), (A \vee C), (A \vee \neg D), (\neg A \vee D), (\neg D \vee A), (B \vee \neg D), (D \vee \neg B)\}$$

$$\neg A \rightarrow B$$

$$\neg A \rightarrow C$$

$$D \rightarrow A$$

$$A \rightarrow D$$

$$B \rightarrow D$$

$$D \rightarrow B$$

Conjunctive Normal Form (CNF)

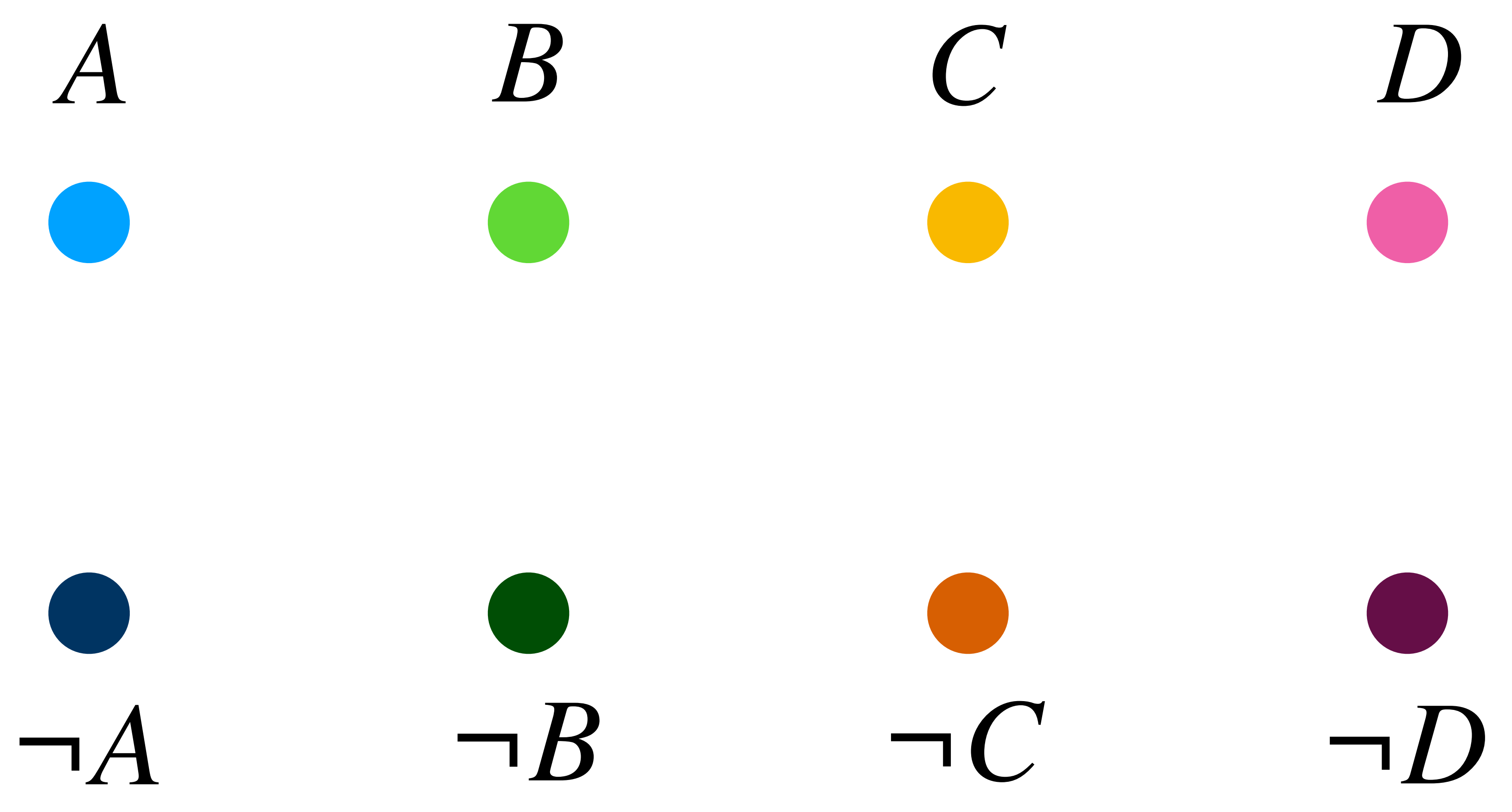
$$\{A \vee B, A \leftarrow (C \rightarrow D), \neg(D \oplus A), B \leftrightarrow D\}$$

$$\{A \vee B, A \vee \neg(\neg C \vee D), \neg((D \vee A) \wedge \neg(D \wedge A)), (B \wedge D) \vee (\neg B \wedge \neg D)\}$$

$$\{(A \vee B), (A \vee C) \wedge (A \vee \neg D), (\neg A \vee D) \wedge (\neg D \vee A), (B \vee \neg D) \wedge (D \vee \neg B)\}$$

$$\{(A \vee B), (A \vee C), (A \vee \neg D), (\neg A \vee D), (\neg D \vee A), (B \vee \neg D), (D \vee \neg B)\}$$

$\neg A \rightarrow B$	$\neg B \rightarrow A$
$\neg A \rightarrow C$	$\neg C \rightarrow A$
$D \rightarrow A$	$\neg A \rightarrow \neg D$
$A \rightarrow D$	$\neg D \rightarrow \neg A$
$B \rightarrow D$	$\neg D \rightarrow \neg B$
$D \rightarrow B$	$\neg B \rightarrow \neg D$



$\neg A \rightarrow B$ $\qquad$ $\neg B \rightarrow A$

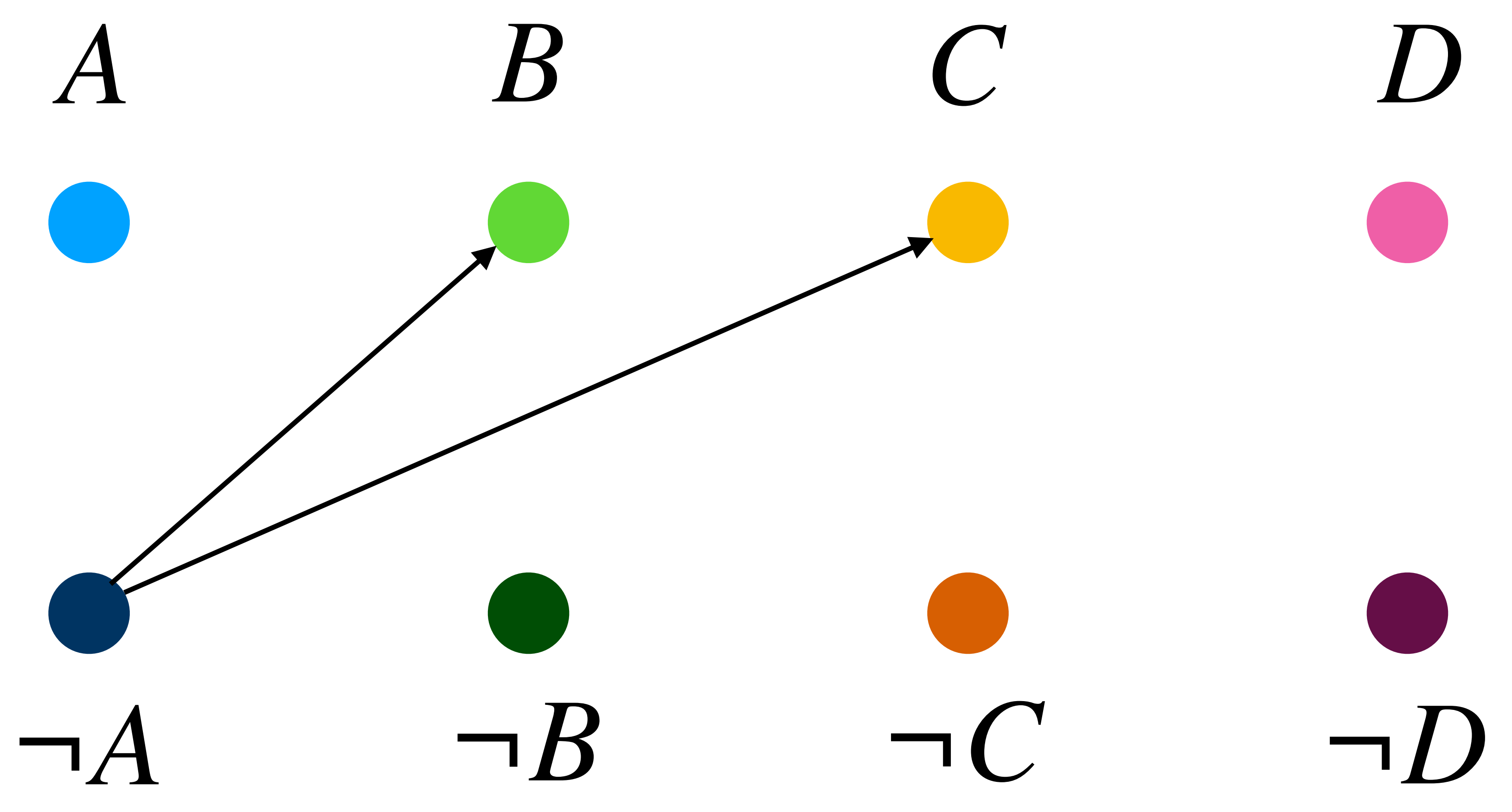
$\neg A \rightarrow C$ $\qquad$ $\neg C \rightarrow A$

$D \rightarrow A$ $\qquad$ $\neg A \rightarrow \neg D$

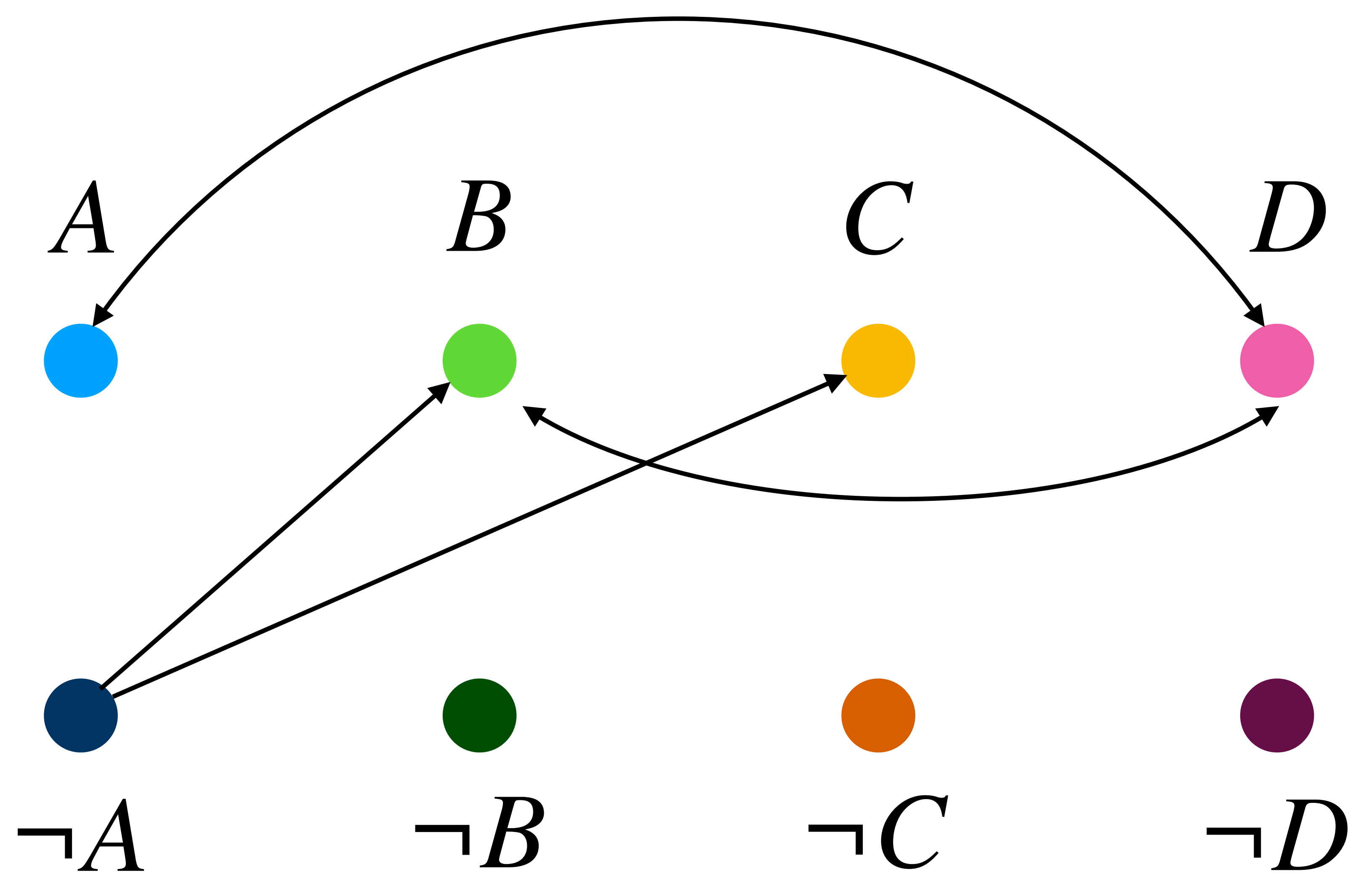
$A \rightarrow D$ $\qquad$ $\neg D \rightarrow \neg A$

$B \rightarrow D$ $\qquad$ $\neg D \rightarrow \neg B$

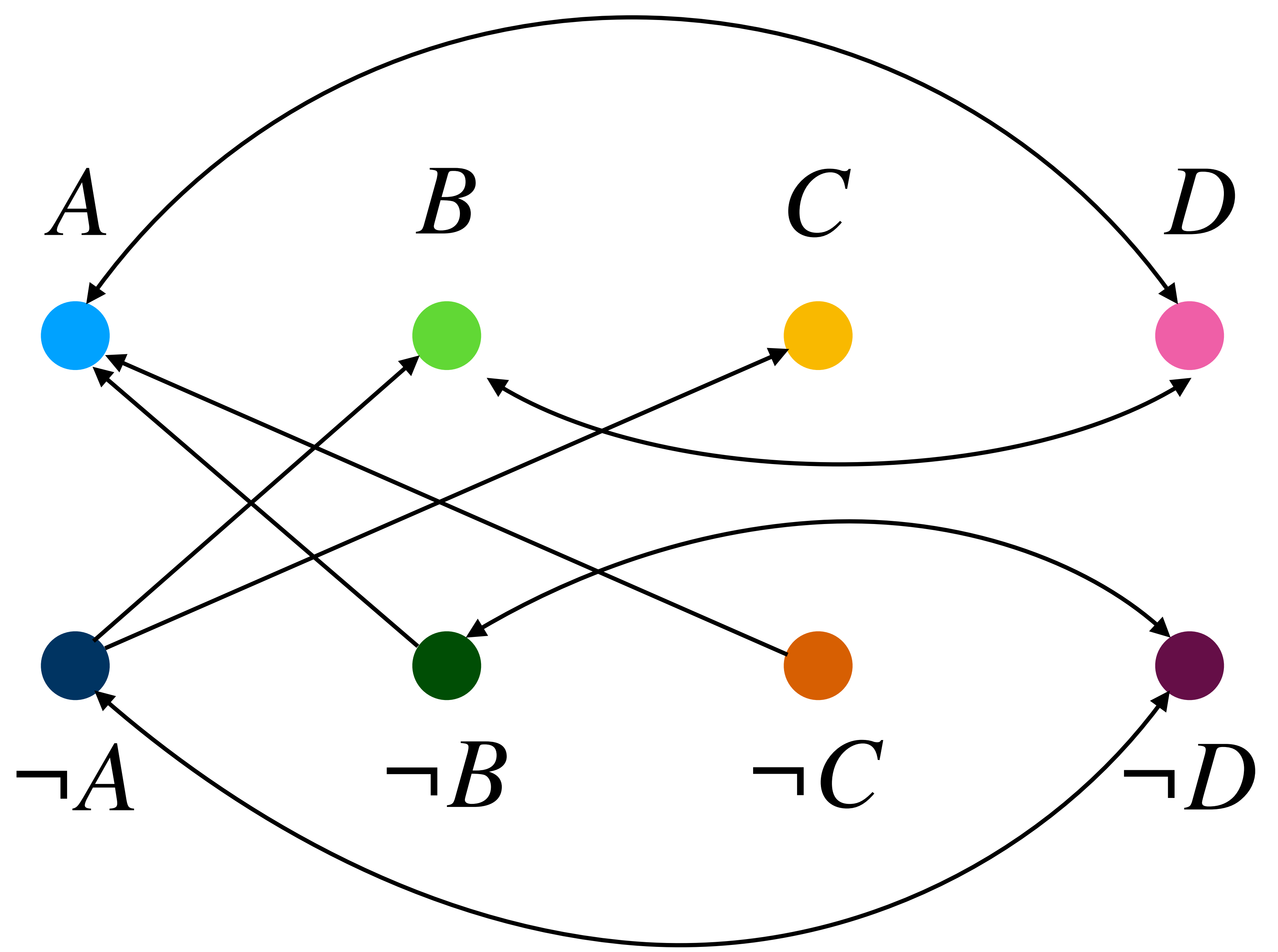
$D \rightarrow B$ $\qquad$ $\neg B \rightarrow \neg D$



$\neg A \to B$	$\neg B \to A$
$\neg A \to C$	$\neg C \to A$
$D \to A$	$\neg A \to \neg D$
$A \to D$	$\neg D \to \neg A$
$B \to D$	$\neg D \to \neg B$
$D \to B$	$\neg B \to \neg D$



$\neg A \rightarrow B$	$\neg B \rightarrow A$
$\neg A \rightarrow C$	$\neg C \rightarrow A$
$D \rightarrow A$	$\neg A \rightarrow \neg D$
$A \rightarrow D$	$\neg D \rightarrow \neg A$
$B \rightarrow D$	$\neg D \rightarrow \neg B$
$D \rightarrow B$	$\neg B \rightarrow \neg D$



$\neg A \rightarrow B$

$\neg A \rightarrow C$

$D \rightarrow A$

$A \rightarrow D$

$B \rightarrow D$

$D \rightarrow B$

$\neg B \rightarrow A$

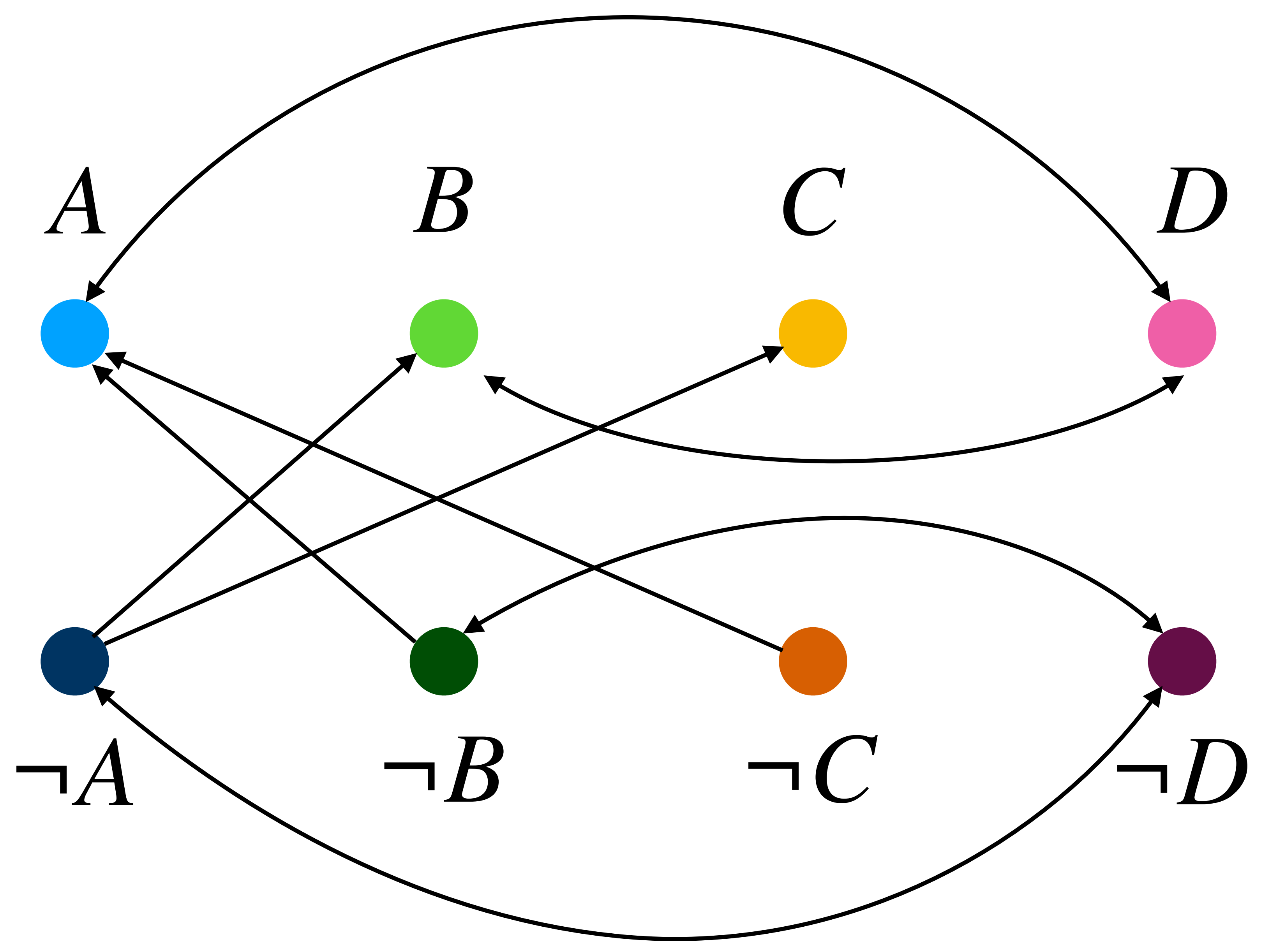
$\neg C \rightarrow A$

$\neg A \rightarrow \neg D$

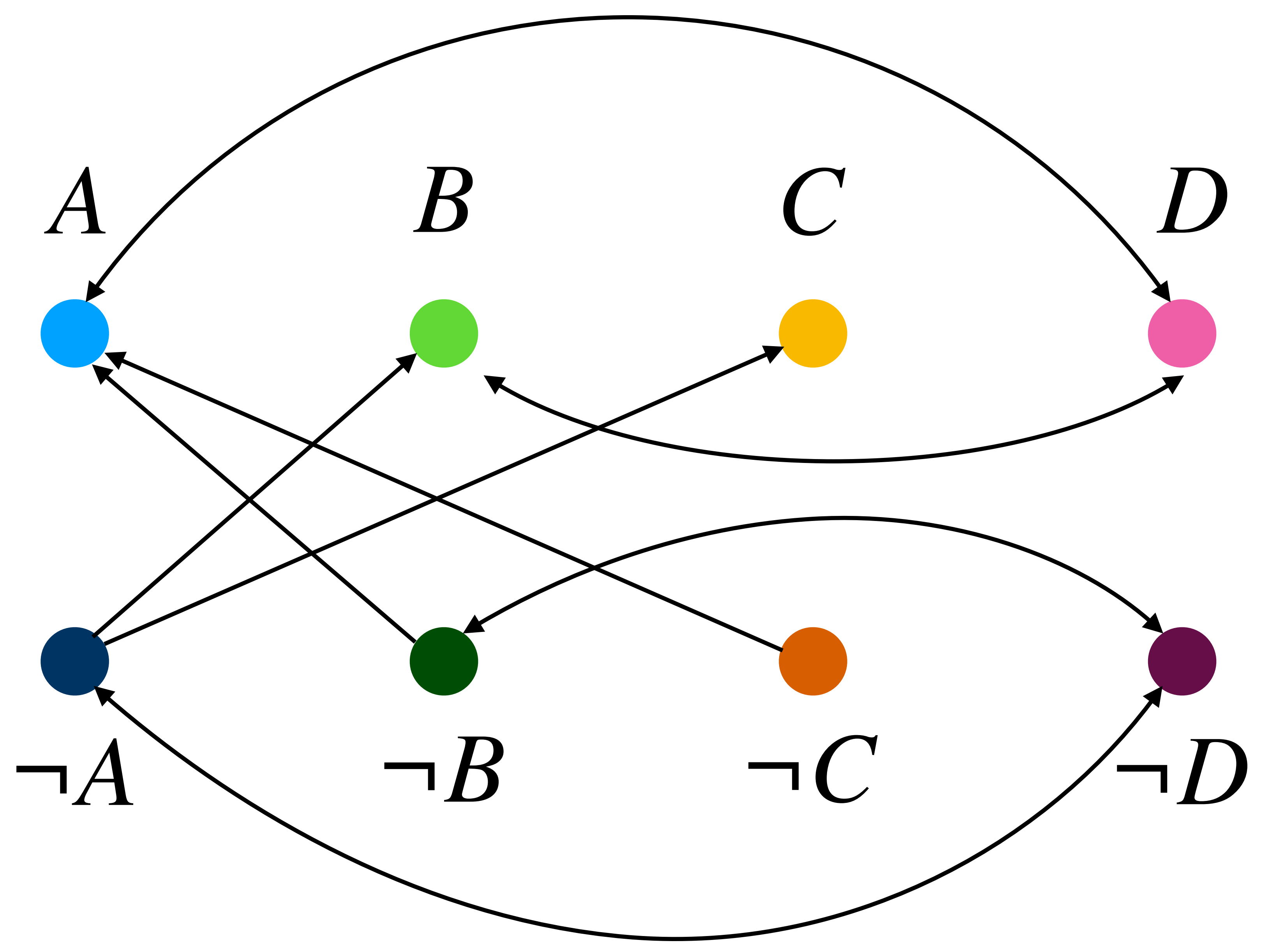
$\neg D \rightarrow \neg A$

$\neg D \rightarrow \neg B$

$\neg B \rightarrow \neg D$



Aspvall, Plass, Tarjan (1979): For any variable X, the vertices for X and $\neg X$ exist in a strongly connected component of the implication graph if and only if the set is not satisfiable.



Aspvall, Plass, Tarjan (1979): For any variable X, the vertices for X and $\neg X$ exist in a strongly connected component of the implication graph if and only if the set is not satisfiable.

When is a SAT problem in P?

Schaefer's dichotomy theorem:

Given a finite set of variables, and a conjunction of constraints, a class of SAT instances is in P if and only if all constraints are:

1. Satisfied when all the variables are true (or when all variables are false).

2. Binary clauses

3. Horn clauses or dual-Horn clauses

4. Affine clauses

What do we want?

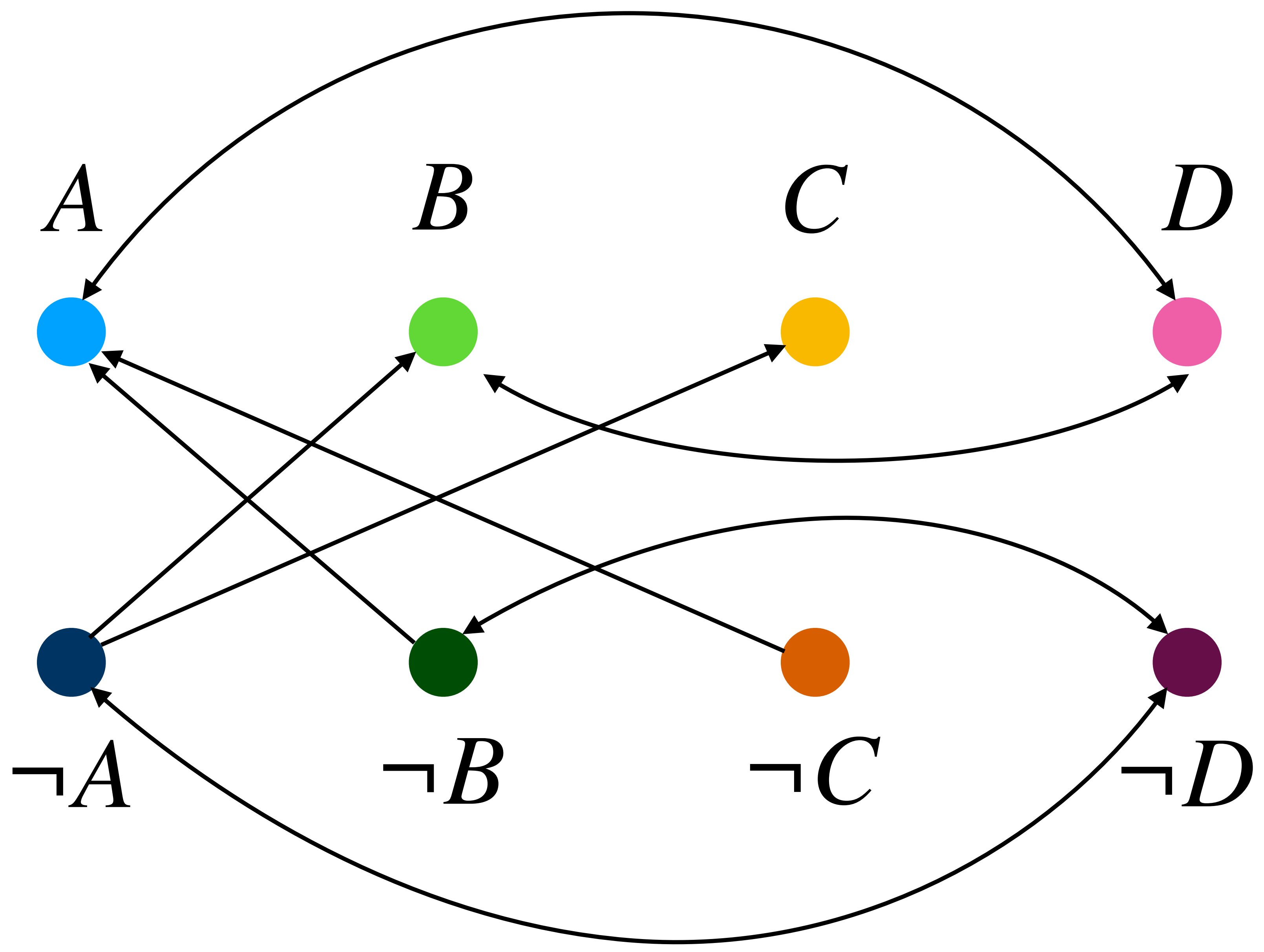
$$A \wedge B \wedge C \rightarrow D$$

$$\texttt{true} \rightarrow A$$

$$D \rightarrow \texttt{false}$$

$$B \wedge C \rightarrow E$$

$$E \wedge C \rightarrow B$$



For certain classes of propositional logic, we have efficient algorithms.

For all of propositional logic, a *somewhat efficient* algorithm?

Davis–Putnam–Logemann–Loveland (DPLL) Algorithm

Partial assignment: $m : \{x_1, \ldots x_n\} \rightarrow \{0,1,?\}$

State of a literal: l is true under m if $m(l) = 1$,
and l is false under m if $m(l) = 0$.

State of a clause: c is true under m if for some $l \in c, m(l) = 1$,
and c is true under m if for all $l \in c, m(l) = 0$.

Unit clause: c is a unit clause under m if exactly one $l \in c$ is unassigned
and the rest are assigned 0. Such an l is called a unit literal.

Davis–Putnam–Logemann–Loveland (DPLL) Algorithm

Input: CNF f, and partial assignment m

Chose an unassigned variable a, and assign it $b \in \{0,1\}$.

If $DPLL\left(f, m[a \to b]\right) = SAT$, return $m[a \to b]$

Else, return $DPLL\left(f, m[a \to 1 - b]\right)$

Davis–Putnam–Logemann–Loveland (DPLL) Algorithm

Input: CNF f, and partial assignment m

If f is true under m, return m.

If f is false under m, return $\perp$.

Chose an unassigned variable a, and assign it $b \in \{0,1\}$.

If $DPLL\left(f, m[a \to b]\right) = SAT$, return $m[a \to b]$

Else, return $DPLL\left(f, m[a \to 1 - b]\right)$

Davis–Putnam–Logemann–Loveland (DPLL) Algorithm

Input: CNF f, and partial assignment m

If f is true under m, return m.

If f is false under m, return $\bot$.

If $\exists$ unit literal p under m, then return $DPLL\left(f, m[p \to 1]\right)$.

Chose an unassigned variable a, and assign it $b \in \{0,1\}$.

If $DPLL\left(f, m[a \to b]\right) = SAT$, return $m[a \to b]$

Else, return $DPLL\left(f, m[a \to 1 - b]\right)$

Davis–Putnam–Logemann–Loveland (DPLL) Algorithm

Input: CNF f, and partial assignment m

If f is true under m, return m.

If f is false under m, return $\bot$.

If $\exists$ unit literal p under m, then return $DPLL\left(f, m[p \to 1]\right)$.

If $\exists$ unit literal $\neg p$ under m, then return $DPLL\left(f, m[p \to 0]\right)$.

Chose an unassigned variable a, and assign it $b \in \{0,1\}$.

If $DPLL\left(f, m[a \to b]\right) = SAT$, return $m[a \to b]$

Else, return $DPLL\left(f, m[a \to 1 - b]\right)$

$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

$$c_4 = (\neg x_3 \vee \neg x_4)$$

$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$

$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

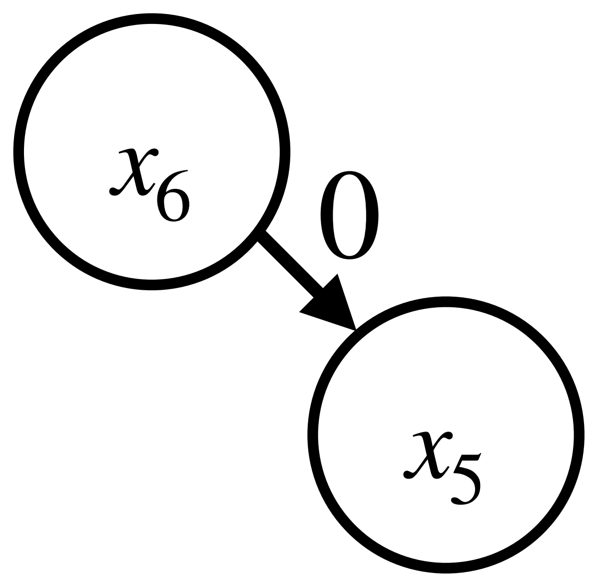
$$c_4 = (\neg x_3 \vee \neg x_4)$$

$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$



Davis–Putnam–Logemann–Loveland (DPLL) Algorithm

Input: CNF f, and partial assignment m

If f is true under m, return m.

If f is false under m, return $\bot$.

If $\exists$ unit literal p under m, then return $DPLL\left(f, m[p \to 1]\right)$.

If $\exists$ unit literal $\neg p$ under m, then return $DPLL\left(f, m[p \to 0]\right)$.

Chose an unassigned variable a, and assign it $b \in \{0,1\}$.

If $DPLL\left(f, m[a \to b]\right) = SAT$, return $m[a \to b]$

Else, return $DPLL\left(f, m[a \to 1 - b]\right)$

$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

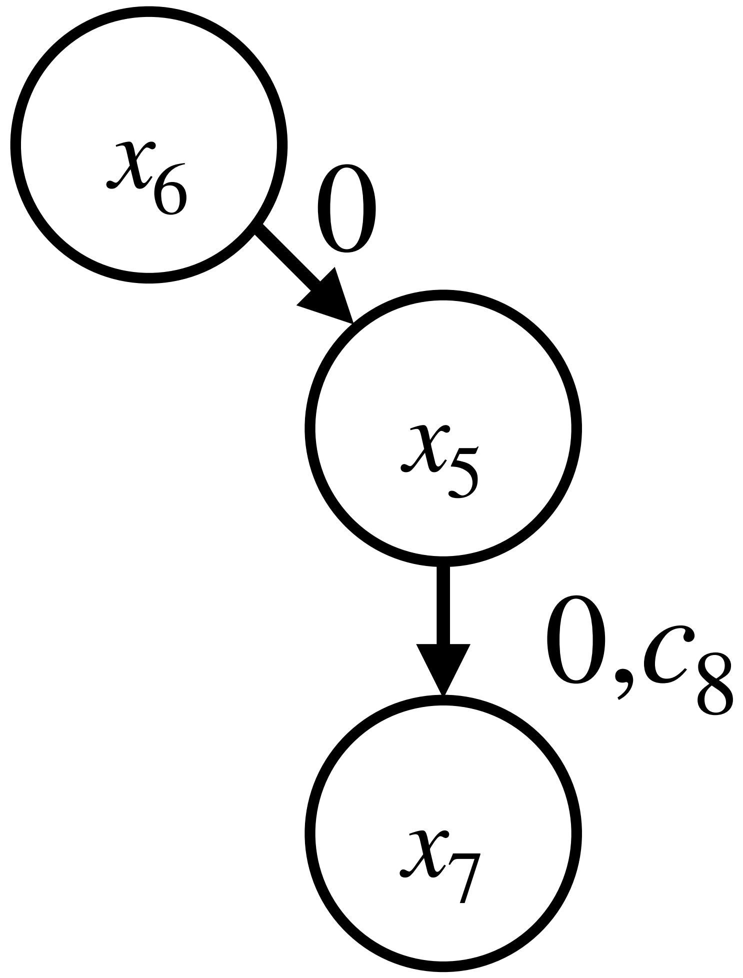
$$c_4 = (\neg x_3 \vee \neg x_4)$$

$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$



Davis–Putnam–Logemann–Loveland (DPLL) Algorithm

Input: CNF f, and partial assignment m

If f is true under m, return m.

If f is false under m, return $\perp$.

If $\exists$ unit literal p under m, then return $DPLL\left(f, m[p \to 1]\right)$.

If $\exists$ unit literal $\neg p$ under m, then return $DPLL\left(f, m[p \to 0]\right)$.

Chose an unassigned variable a, and assign it $b \in \{0,1\}$.

If $DPLL\left(f, m[a \to b]\right) = SAT$, return $m[a \to b]$

Else, return $DPLL\left(f, m[a \to 1 - b]\right)$

$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

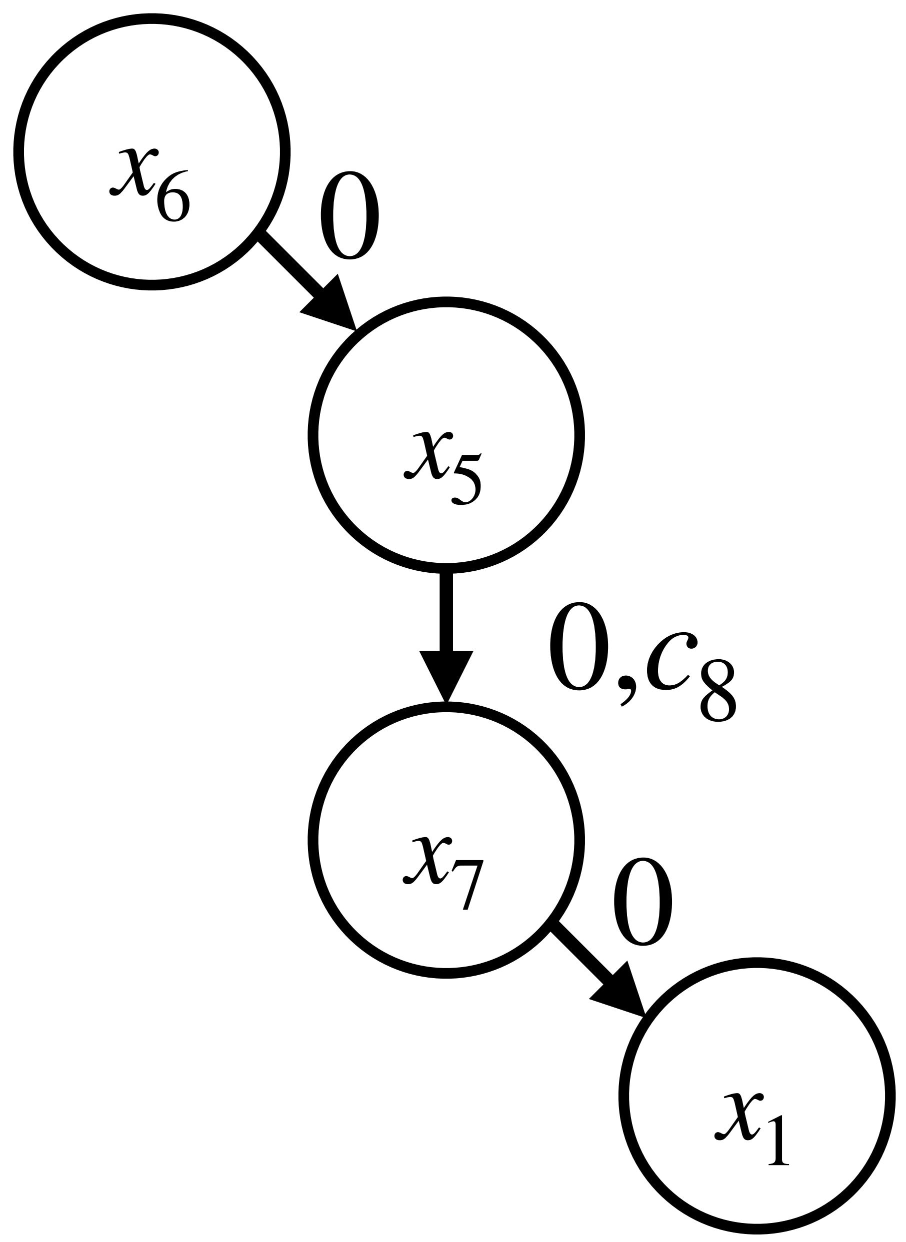
$$c_4 = (\neg x_3 \vee \neg x_4)$$

$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$



$$c_1 = (\neg x_1 \lor x_2)$$

$$c_2 = (\neg x_1 \lor x_3 \lor x_5)$$

$$c_3 = (\neg x_2 \lor x_4)$$

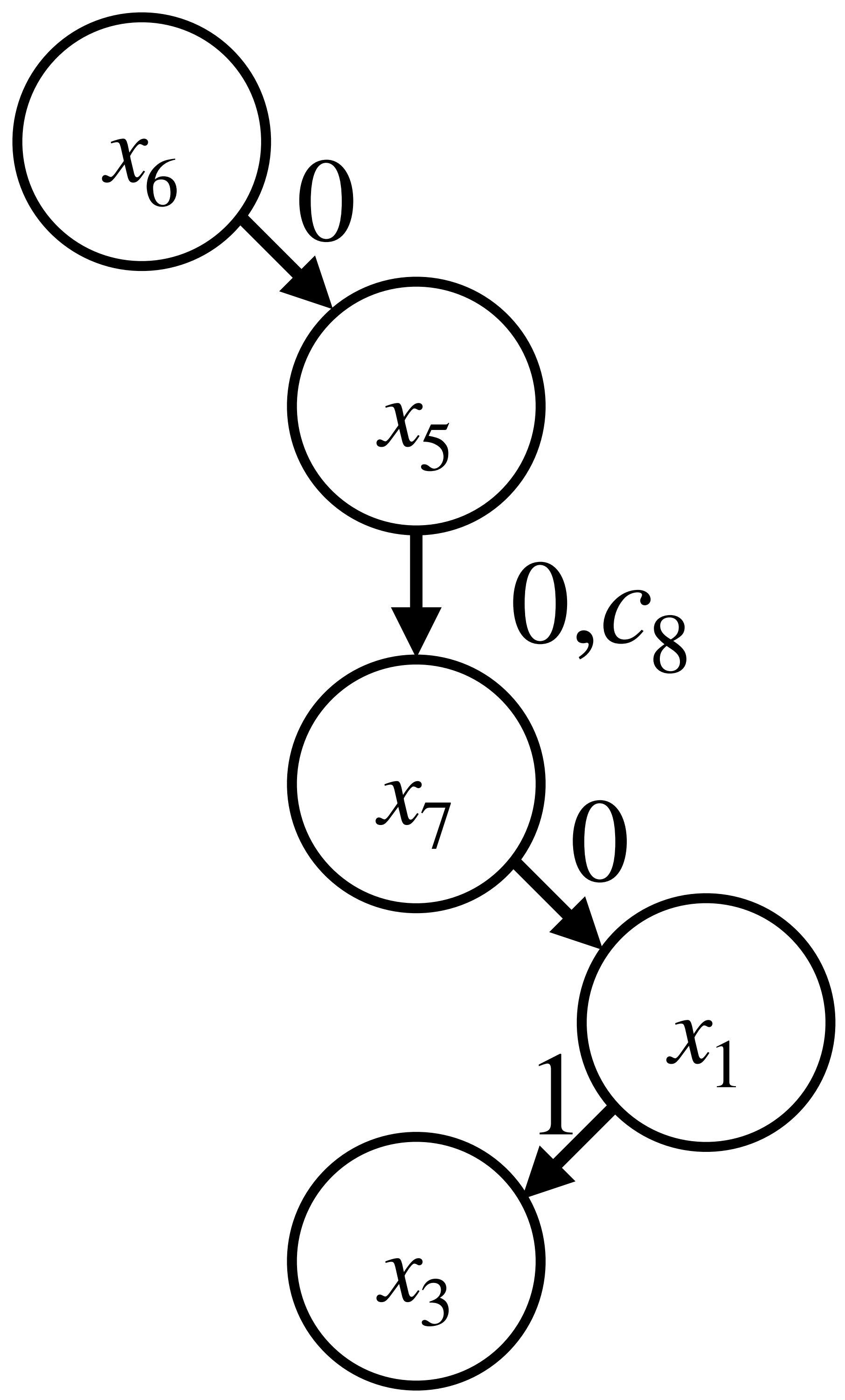
$$c_4 = (\neg x_3 \lor \neg x_4)$$

$$c_5 = (x_1 \lor x_5 \lor \neg x_2)$$

$$c_6 = (x_2 \lor x_3)$$

$$c_7 = (x_2 \lor \neg x_3 \lor x_7)$$

$$c_8 = (x_6 \lor \neg x_5)$$



$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

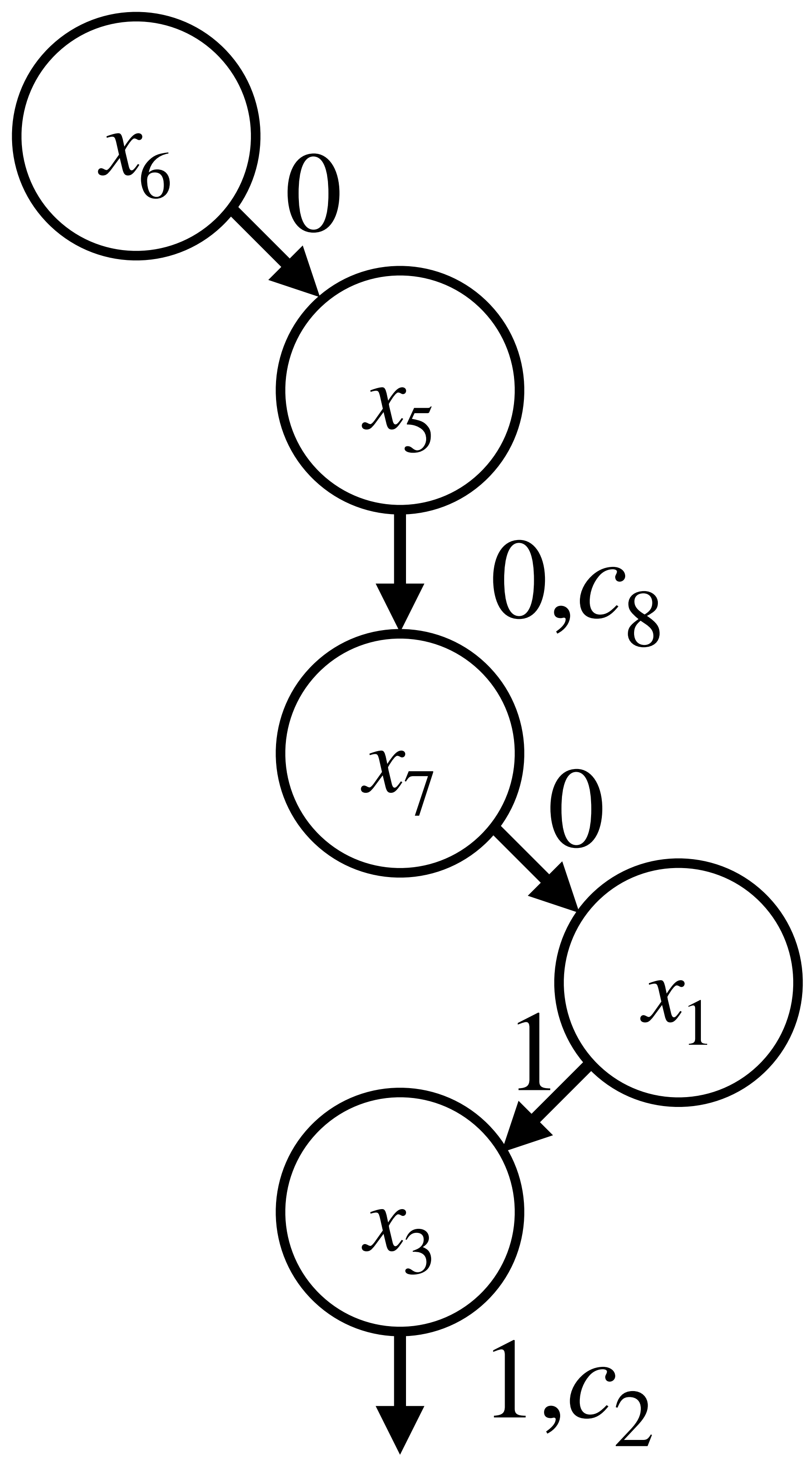
$$c_4 = (\neg x_3 \vee \neg x_4)$$

$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$



$$c_1 = (\neg x_1 \lor x_2)$$

$$c_2 = (\neg x_1 \lor x_3 \lor x_5)$$

$$c_3 = (\neg x_2 \lor x_4)$$

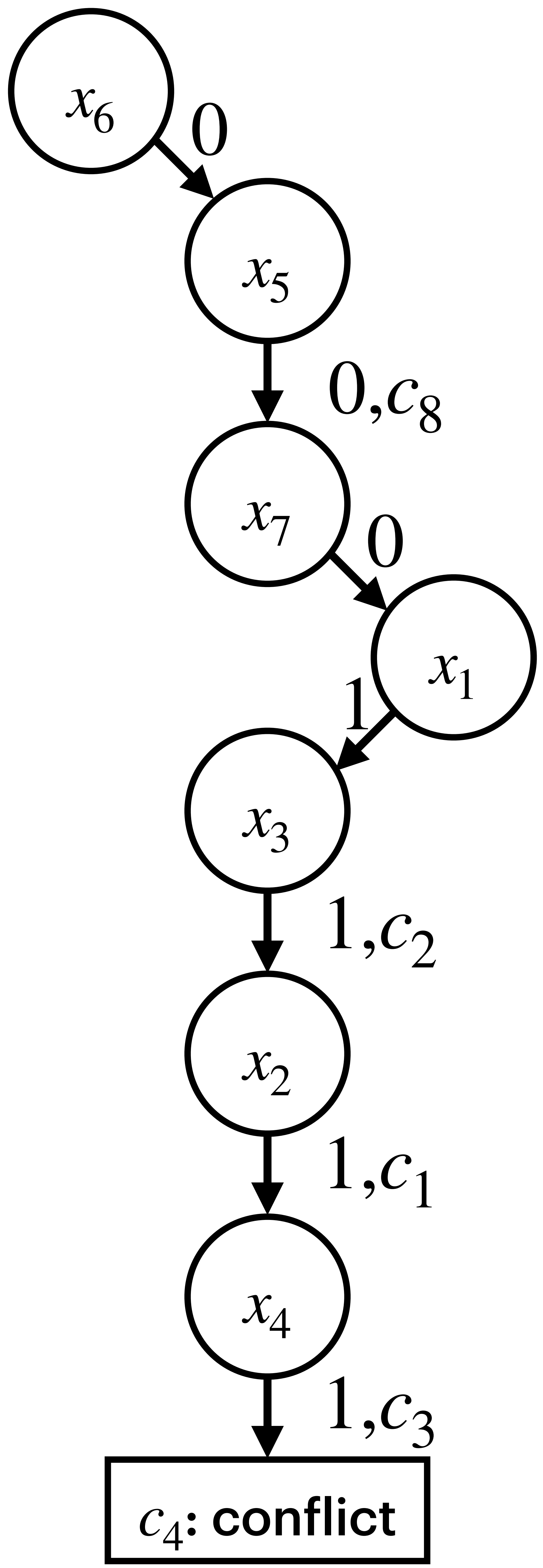
$$c_4 = (\neg x_3 \lor \neg x_4)$$

$$c_5 = (x_1 \lor x_5 \lor \neg x_2)$$

$$c_6 = (x_2 \lor x_3)$$

$$c_7 = (x_2 \lor \neg x_3 \lor x_7)$$

$$c_8 = (x_6 \lor \neg x_5)$$



Davis–Putnam–Logemann–Loveland (DPLL) Algorithm

Input: CNF f, and partial assignment m

If f is true under m, return m.

If f is false under m, return $\bot$.

If $\exists$ unit literal p under m, then return $DPLL\left(f, m[p \to 1]\right)$.

If $\exists$ unit literal $\neg p$ under m, then return $DPLL\left(f, m[p \to 0]\right)$.

Chose an unassigned variable a, and assign it $b \in \{0,1\}$.

If $DPLL\left(f, m[a \to b]\right) = SAT$, return $m[a \to b]$

Else, return $DPLL\left(f, m[a \to 1 - b]\right)$

$$c_1 = (\neg x_1 \lor x_2)$$

$$c_2 = (\neg x_1 \lor x_3 \lor x_5)$$

$$c_3 = (\neg x_2 \lor x_4)$$

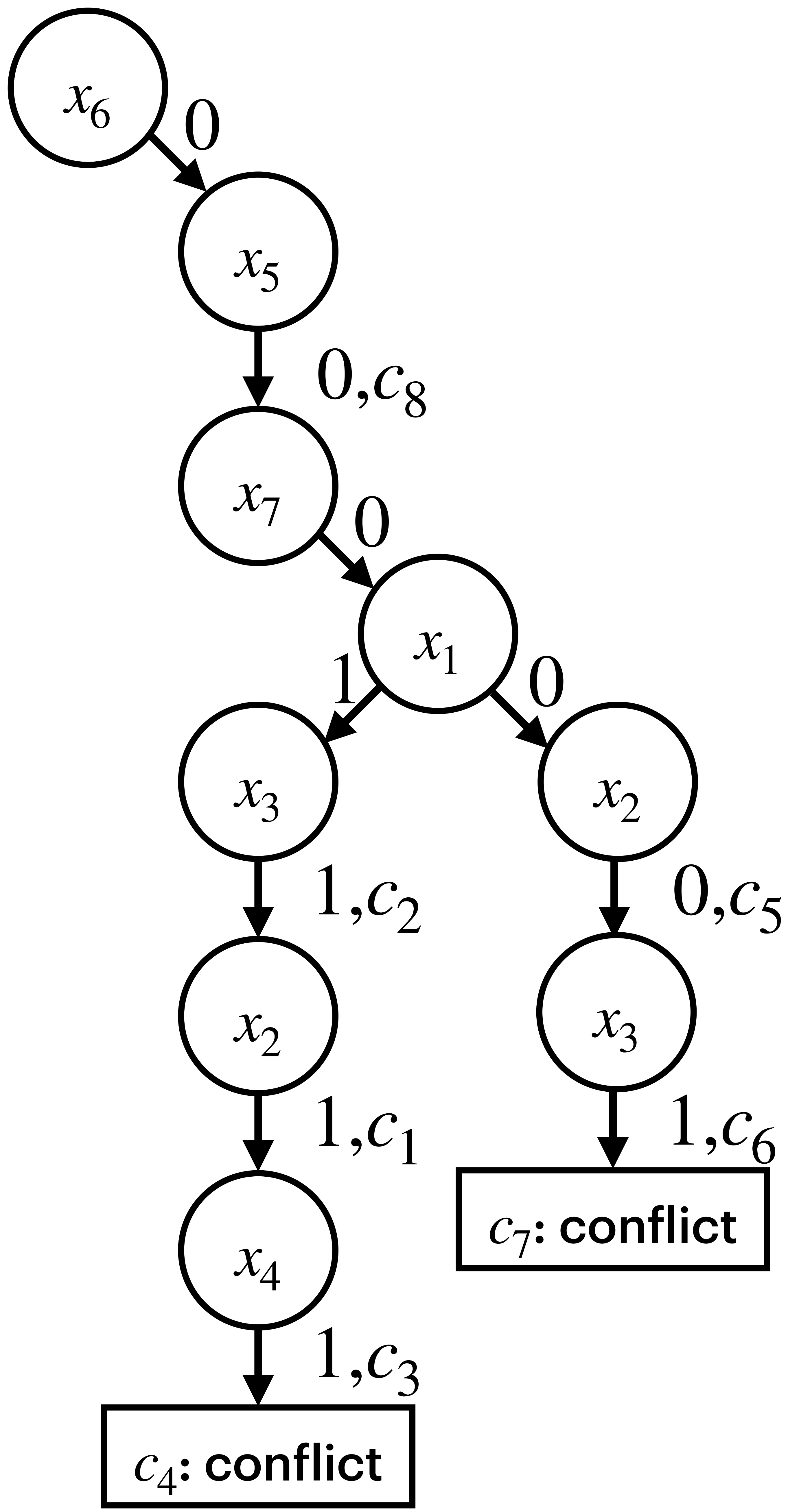
$$c_4 = (\neg x_3 \lor \neg x_4)$$

$$c_5 = (x_1 \lor x_5 \lor \neg x_2)$$

$$c_6 = (x_2 \lor x_3)$$

$$c_7 = (x_2 \lor \neg x_3 \lor x_7)$$

$$c_8 = (x_6 \lor \neg x_5)$$



$$c_1 = (\neg x_1 \lor x_2)$$

$$c_2 = (\neg x_1 \lor x_3 \lor x_5)$$

$$c_3 = (\neg x_2 \lor x_4)$$

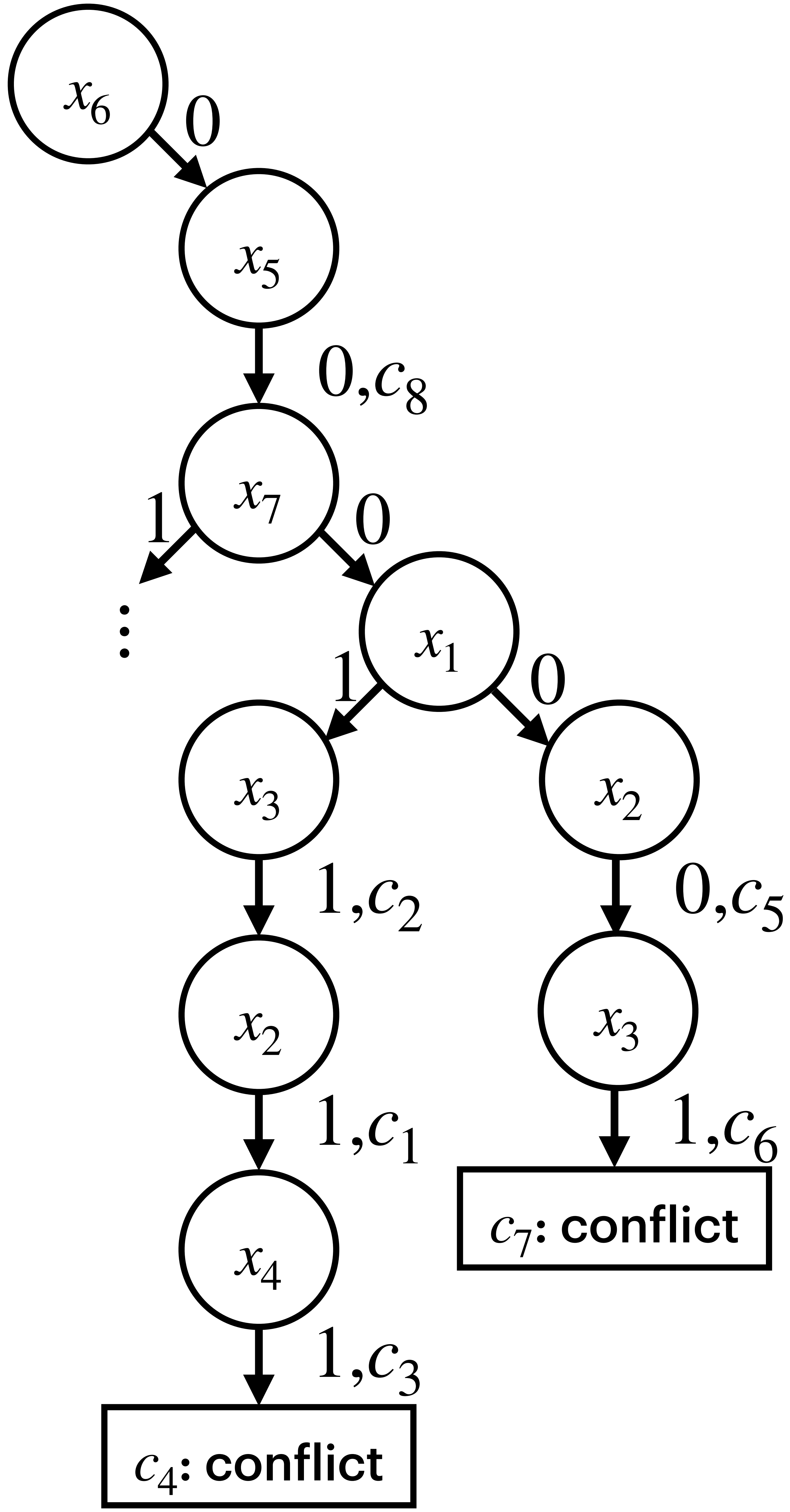
$$c_4 = (\neg x_3 \lor \neg x_4)$$

$$c_5 = (x_1 \lor x_5 \lor \neg x_2)$$

$$c_6 = (x_2 \lor x_3)$$

$$c_7 = (x_2 \lor \neg x_3 \lor x_7)$$

$$c_8 = (x_6 \lor \neg x_5)$$



Time to Code!

$c_1 = (\neg x_1 \vee x_2)$

$c_2 = (\neg x_1 \vee x_3 \vee x_5)$

$c_3 = (\neg x_2 \vee x_4)$

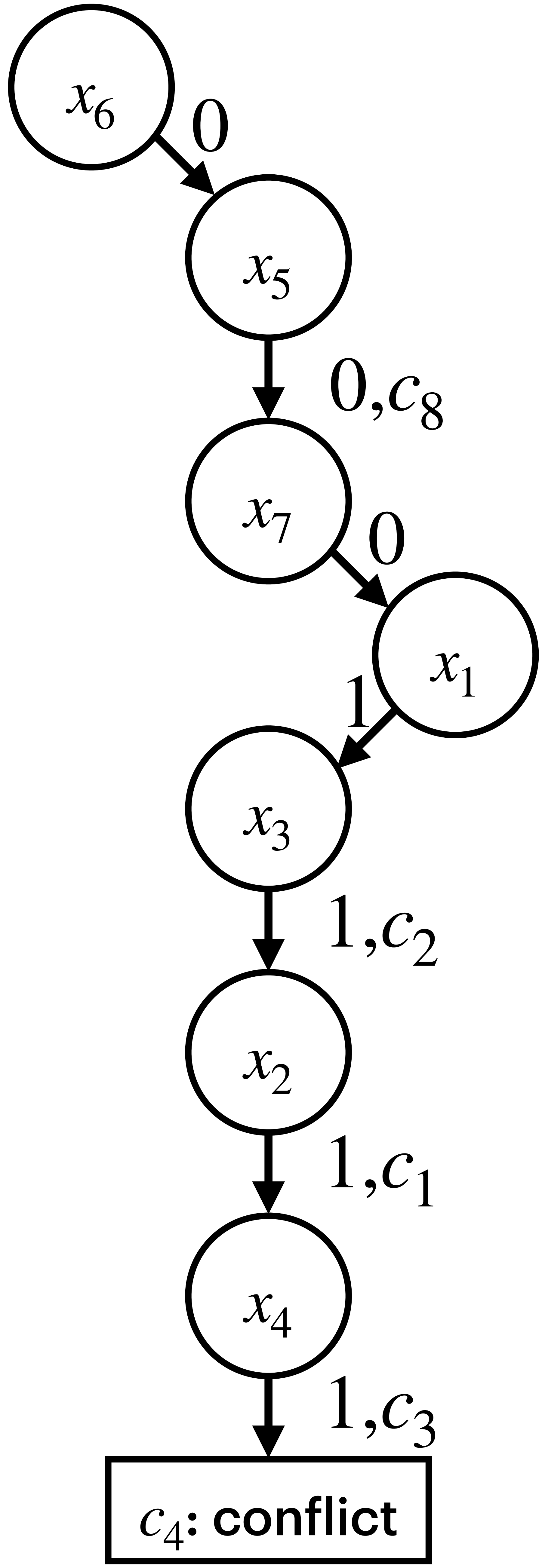
$c_4 = (\neg x_3 \vee \neg x_4)$

$c_5 = (x_1 \vee x_5 \vee \neg x_2)$

$c_6 = (x_2 \vee x_3)$

$c_7 = (x_2 \vee \neg x_3 \vee x_7)$

$c_8 = (x_6 \vee \neg x_5)$



$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

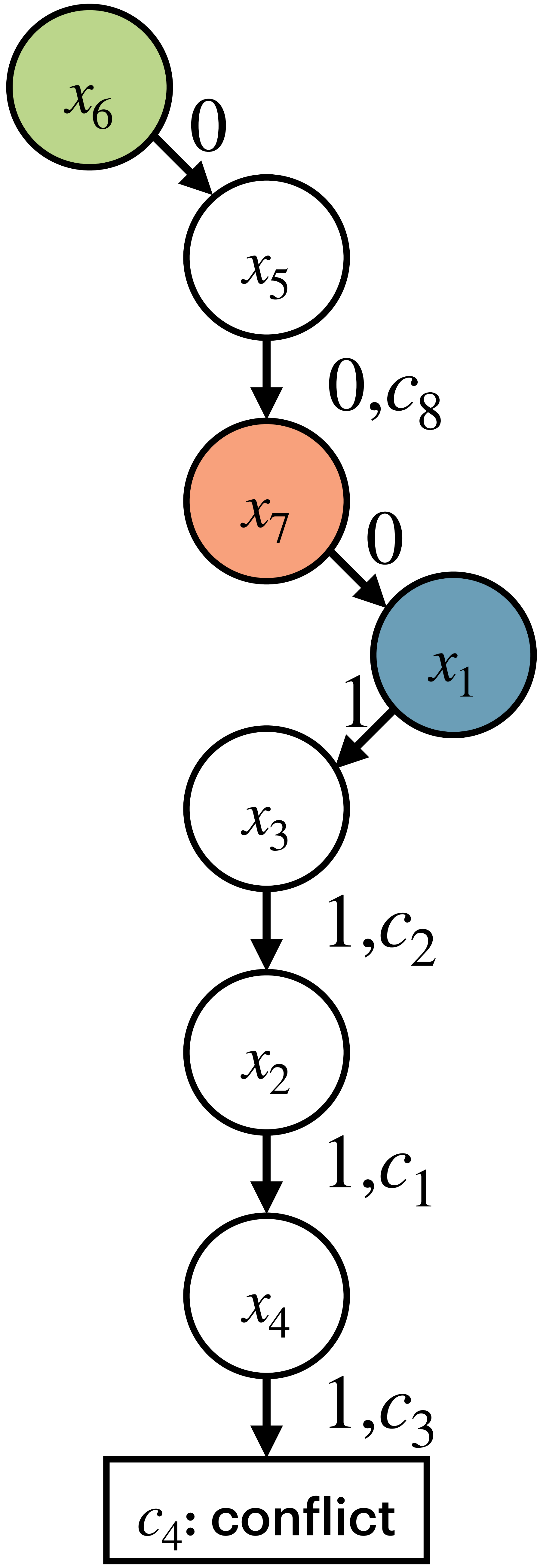
$$c_4 = (\neg x_3 \vee \neg x_4)$$

$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$



$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

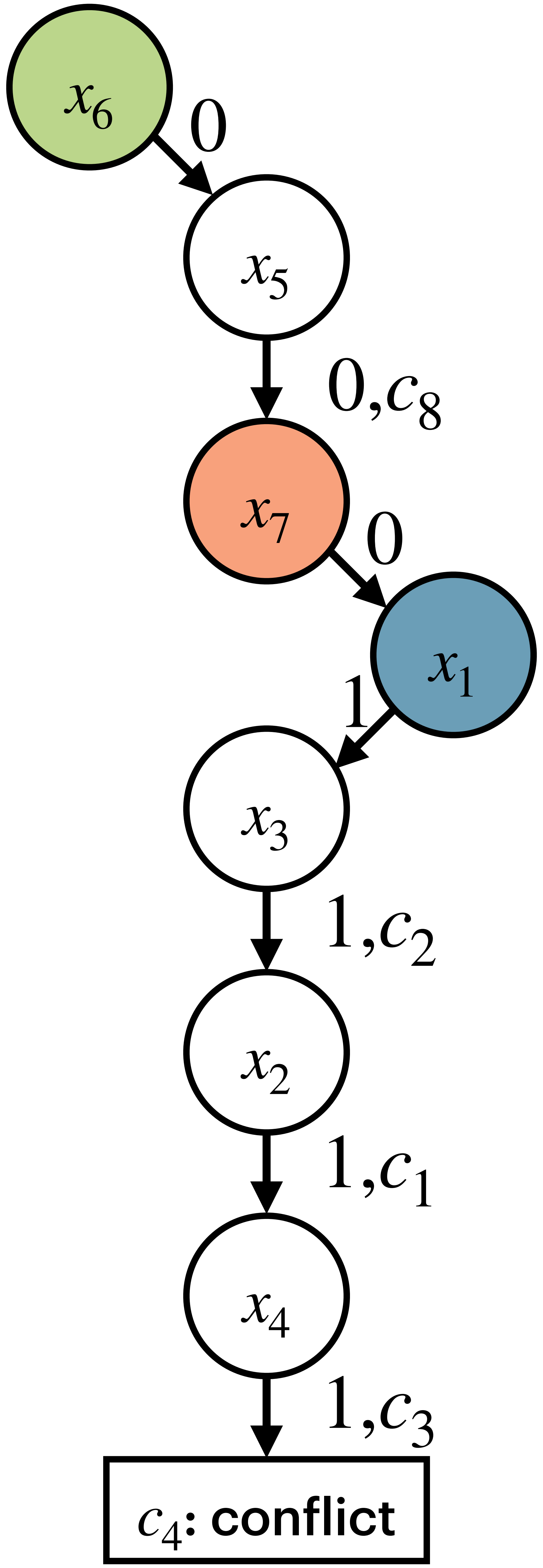
$$c_4 = (\neg x_3 \vee \neg x_4)$$

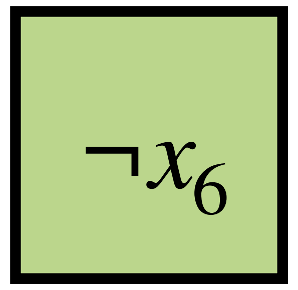
$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$





$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

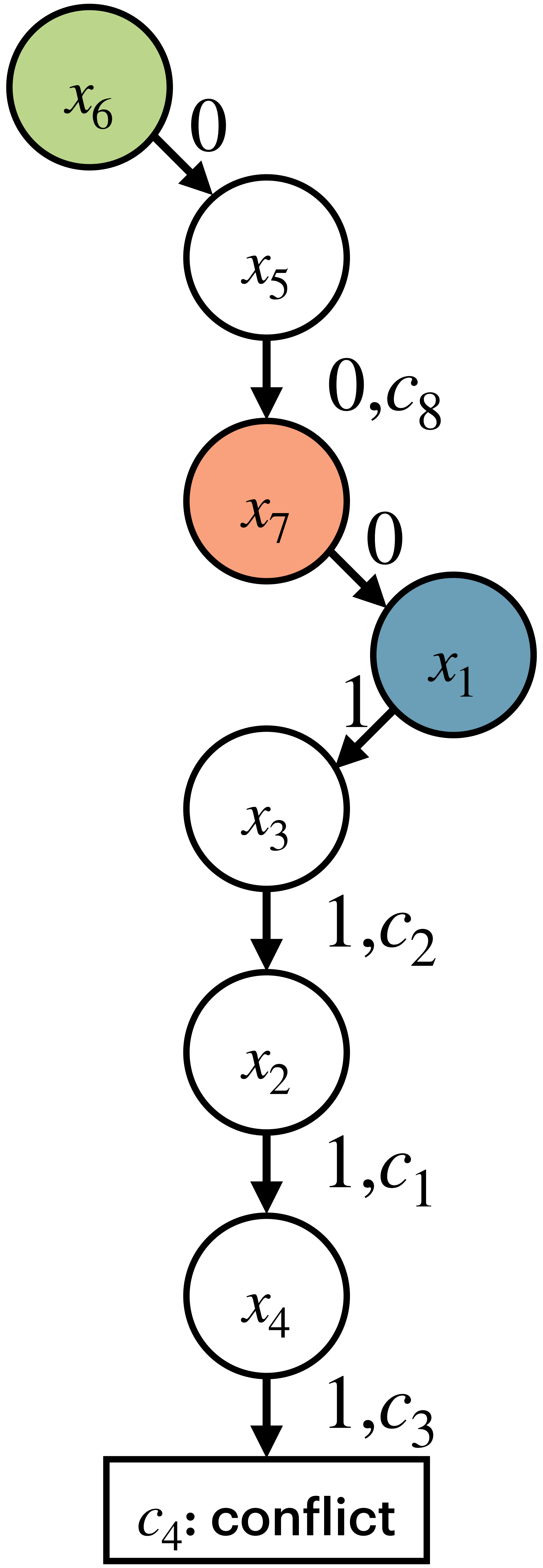
$$c_4 = (\neg x_3 \vee \neg x_4)$$

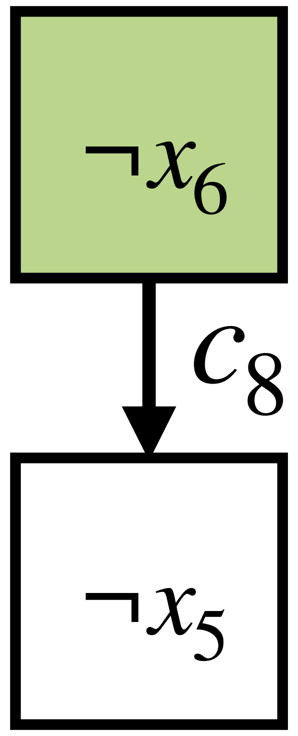
$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$





$$c_1 = (\neg x_1 \lor x_2)$$

$$c_2 = (\neg x_1 \lor x_3 \lor x_5)$$

$$c_3 = (\neg x_2 \lor x_4)$$

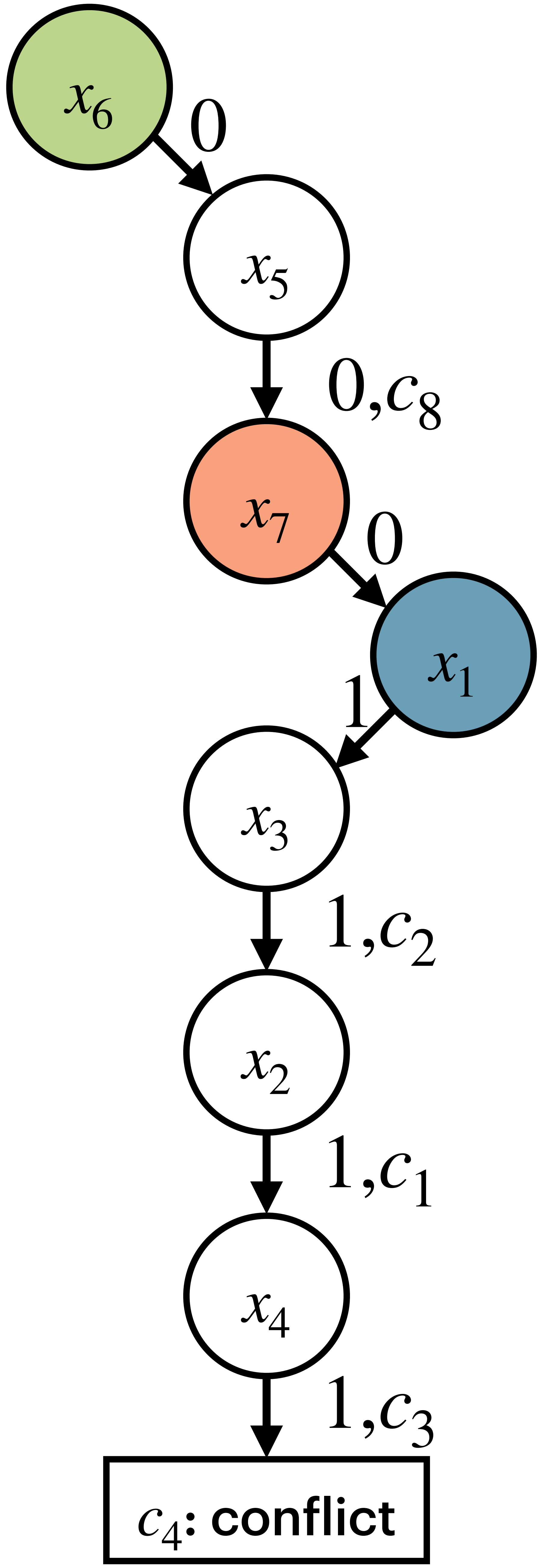
$$c_4 = (\neg x_3 \lor \neg x_4)$$

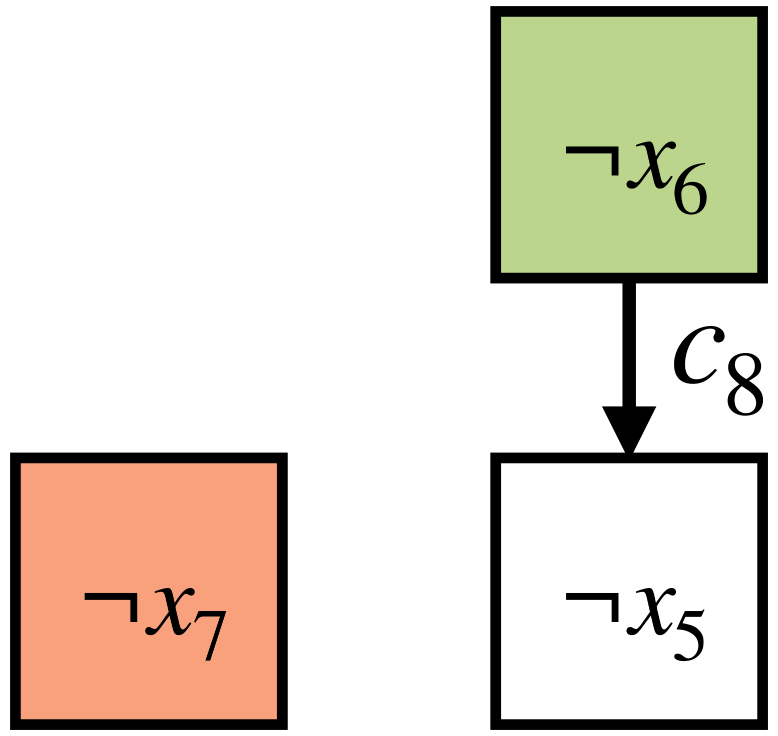
$$c_5 = (x_1 \lor x_5 \lor \neg x_2)$$

$$c_6 = (x_2 \lor x_3)$$

$$c_7 = (x_2 \lor \neg x_3 \lor x_7)$$

$$c_8 = (x_6 \lor \neg x_5)$$





$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

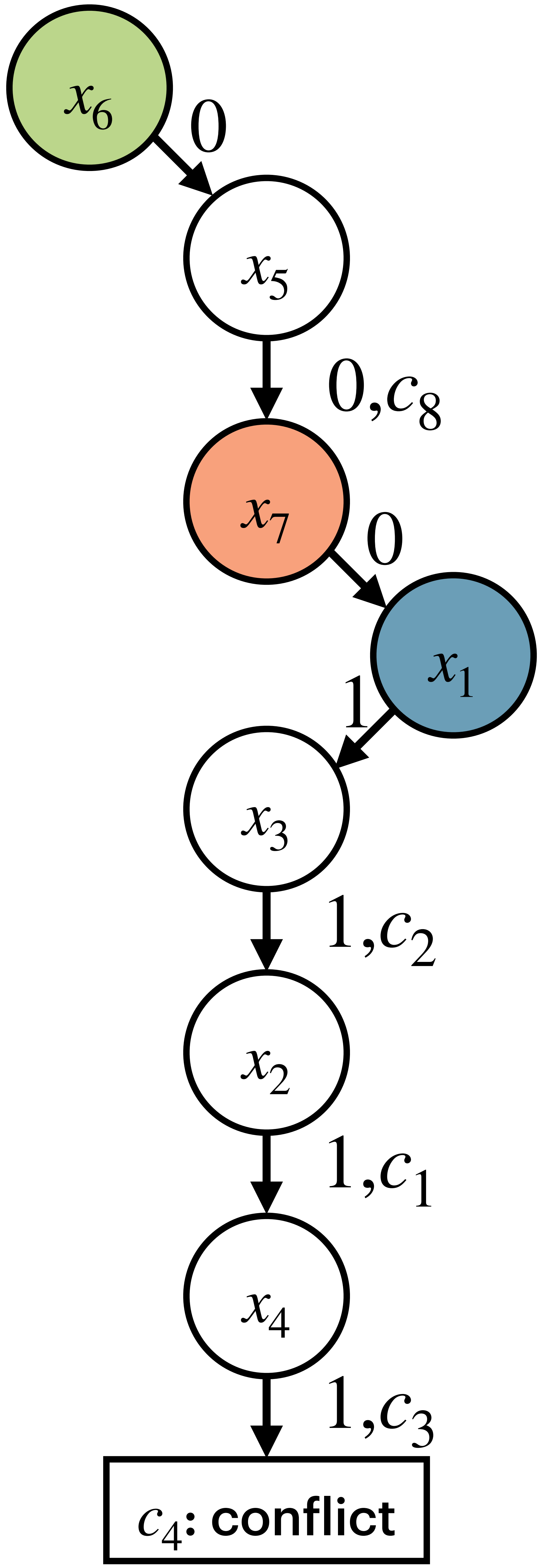
$$c_4 = (\neg x_3 \vee \neg x_4)$$

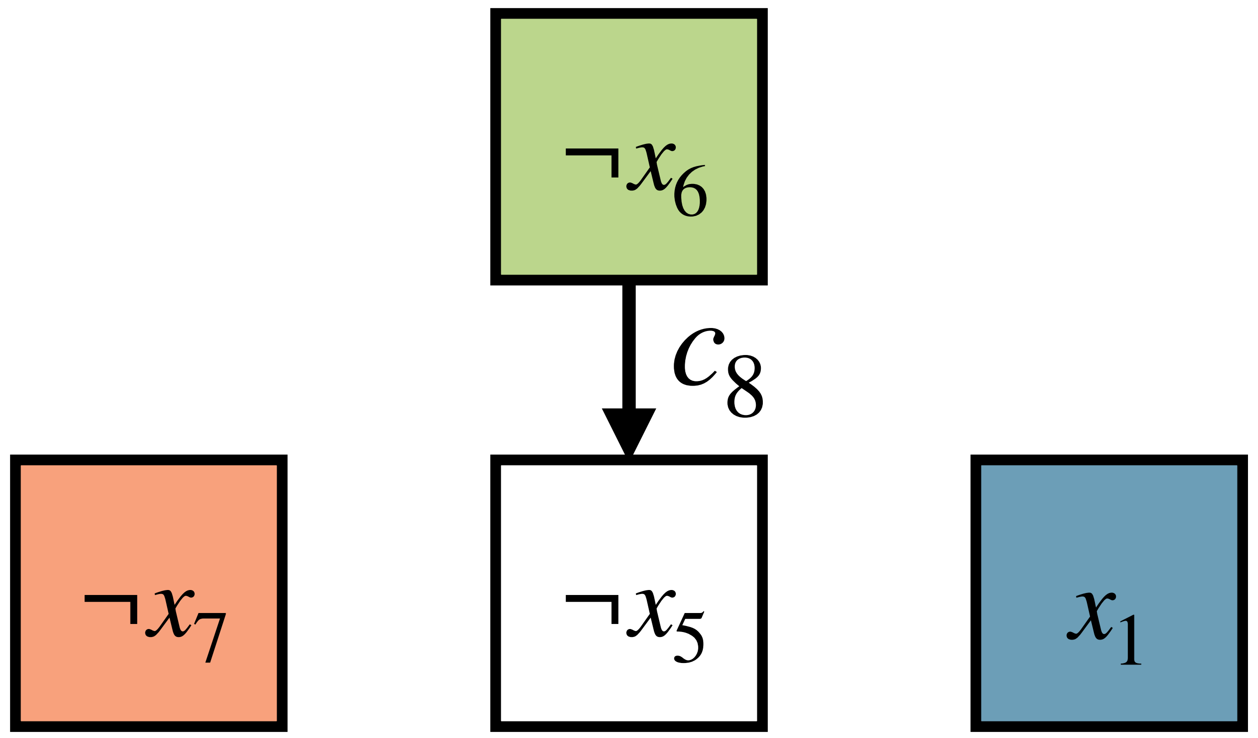
$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

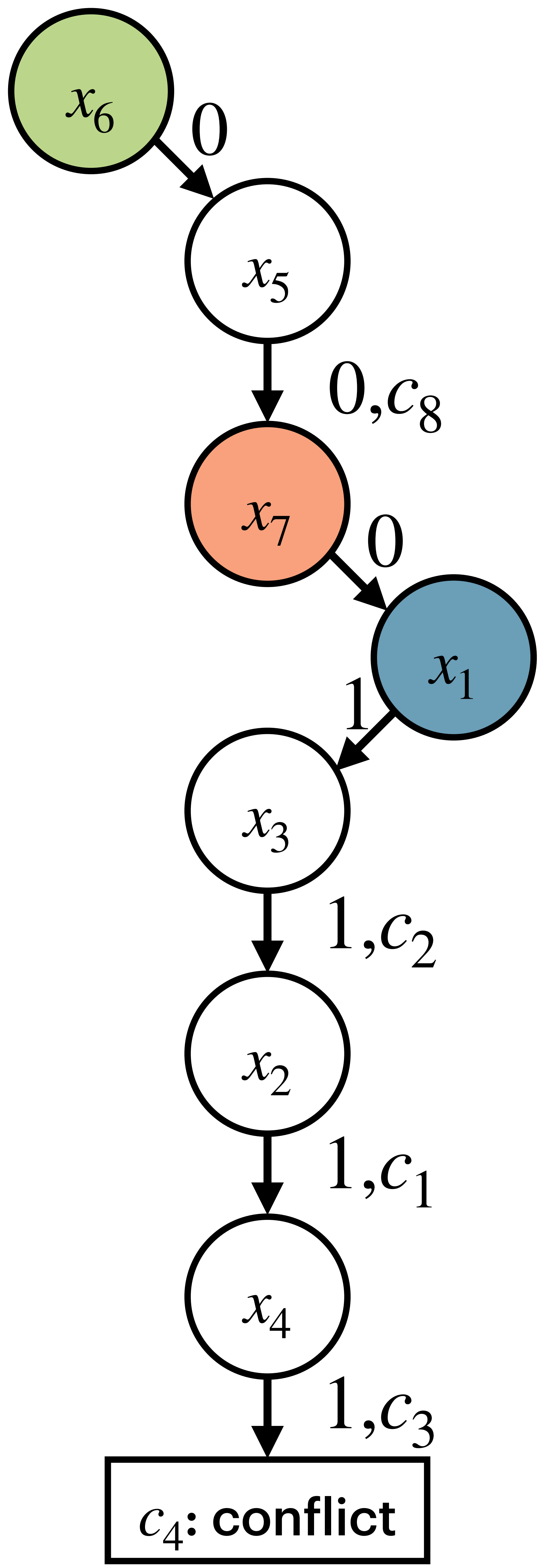
$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

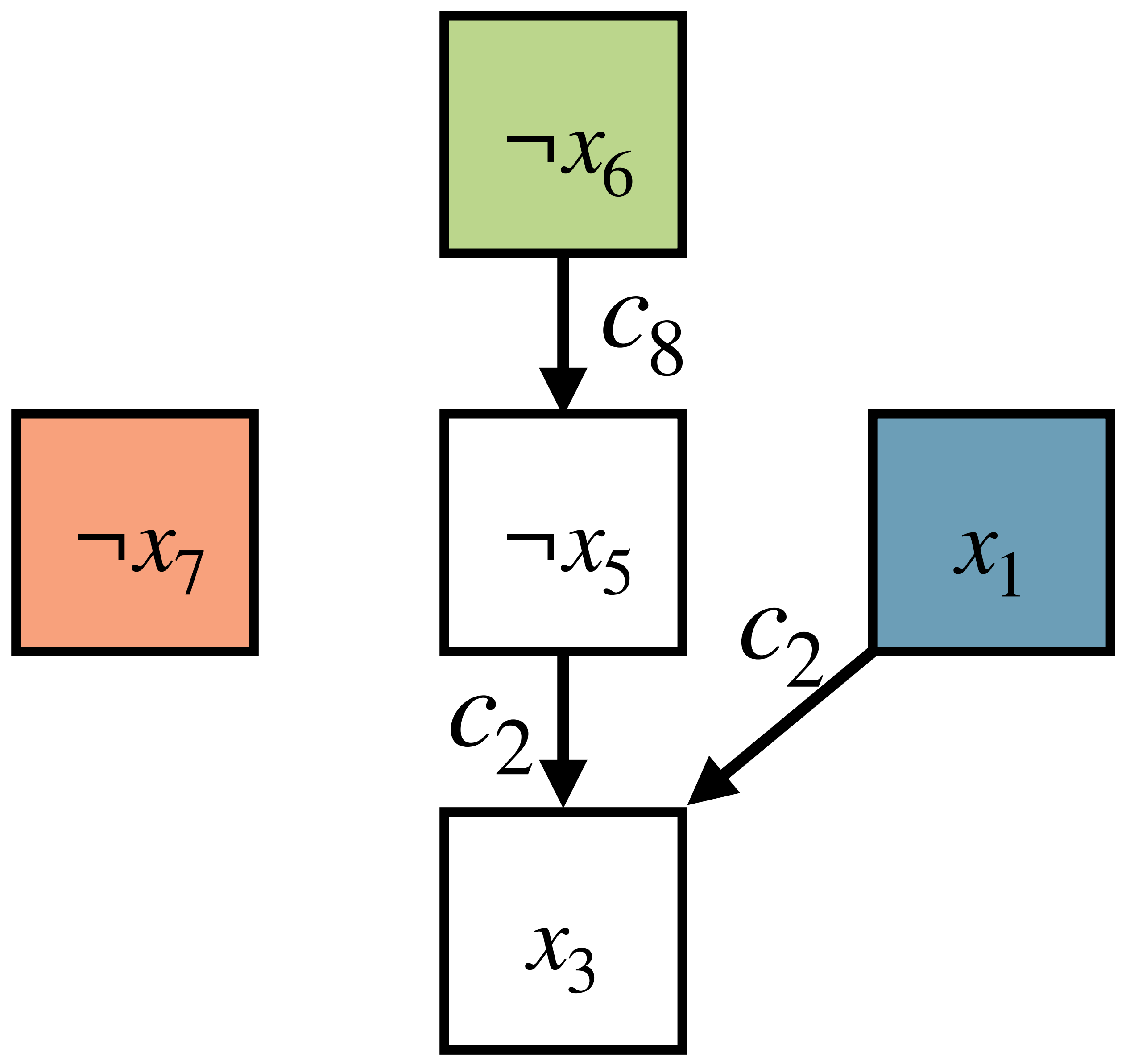
$$c_8 = (x_6 \vee \neg x_5)$$



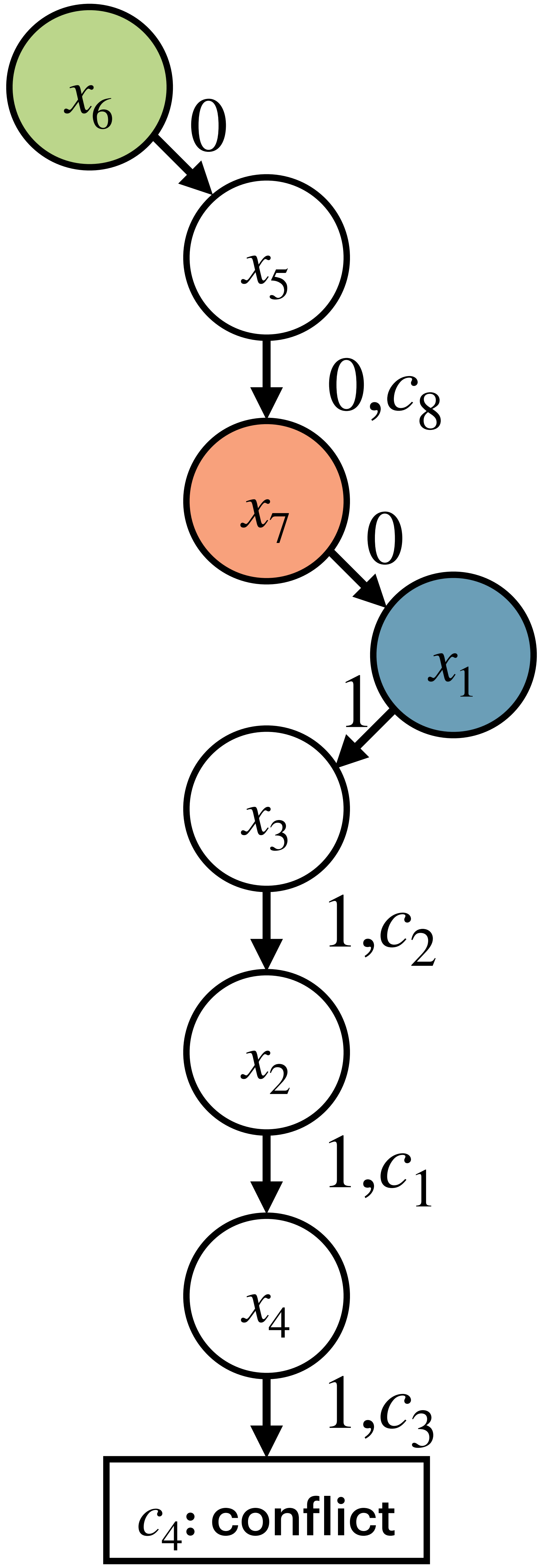


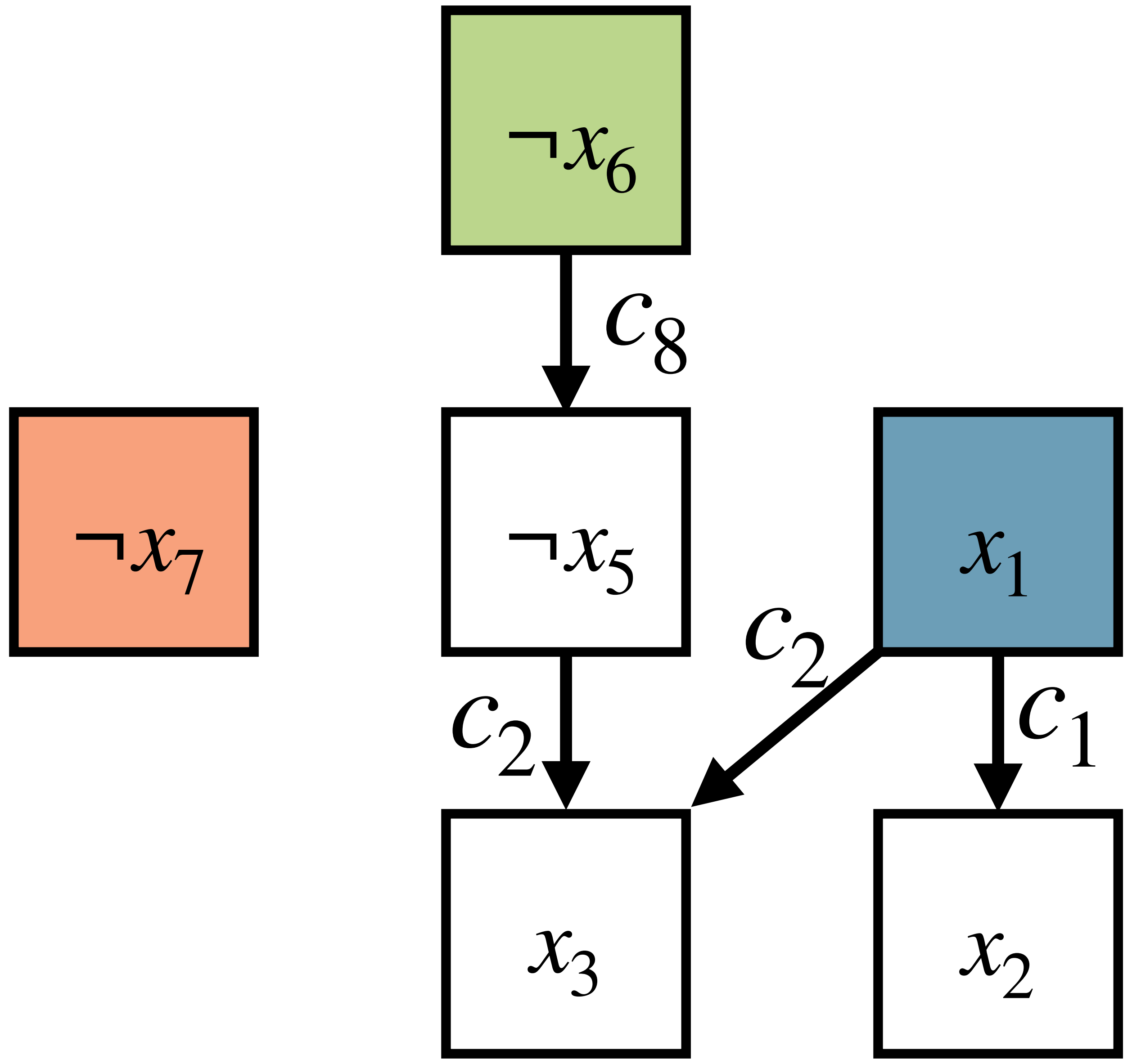
$$c_1 = (\neg x_1 \lor x_2)$$
$$c_2 = (\neg x_1 \lor x_3 \lor x_5)$$
$$c_3 = (\neg x_2 \lor x_4)$$
$$c_4 = (\neg x_3 \lor \neg x_4)$$
$$c_5 = (x_1 \lor x_5 \lor \neg x_2)$$
$$c_6 = (x_2 \lor x_3)$$
$$c_7 = (x_2 \lor \neg x_3 \lor x_7)$$
$$c_8 = (x_6 \lor \neg x_5)$$





$$c_1 = (\neg x_1 \lor x_2)$$
$$c_2 = (\neg x_1 \lor x_3 \lor x_5)$$
$$c_3 = (\neg x_2 \lor x_4)$$
$$c_4 = (\neg x_3 \lor \neg x_4)$$
$$c_5 = (x_1 \lor x_5 \lor \neg x_2)$$
$$c_6 = (x_2 \lor x_3)$$
$$c_7 = (x_2 \lor \neg x_3 \lor x_7)$$
$$c_8 = (x_6 \lor \neg x_5)$$





$$c_1 = (\neg x_1 \lor x_2)$$

$$c_2 = (\neg x_1 \lor x_3 \lor x_5)$$

$$c_3 = (\neg x_2 \lor x_4)$$

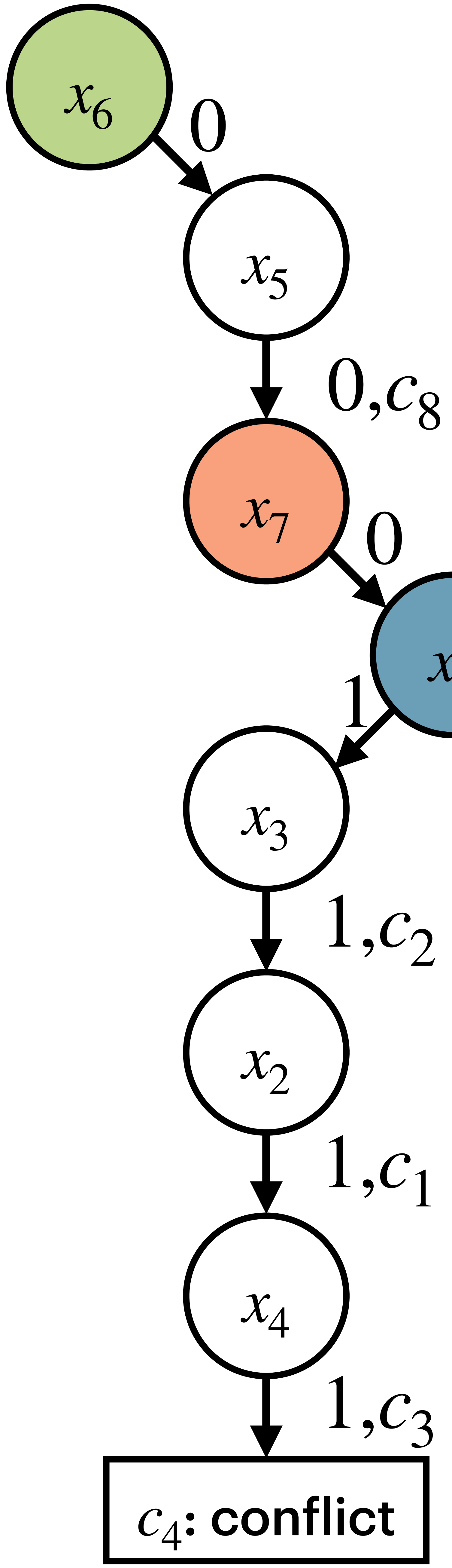
$$c_4 = (\neg x_3 \lor \neg x_4)$$

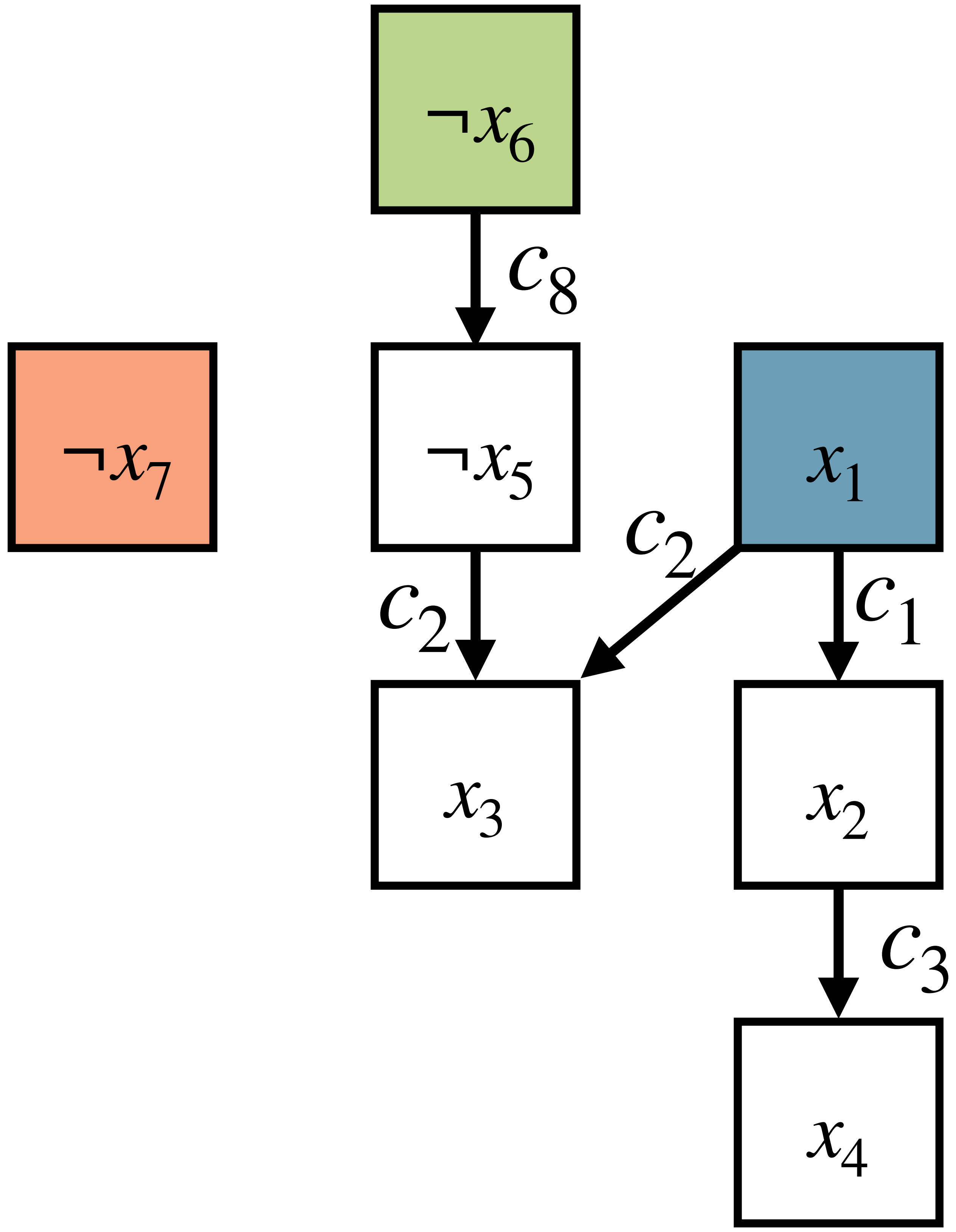
$$c_5 = (x_1 \lor x_5 \lor \neg x_2)$$

$$c_6 = (x_2 \lor x_3)$$

$$c_7 = (x_2 \lor \neg x_3 \lor x_7)$$

$$c_8 = (x_6 \lor \neg x_5)$$





$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

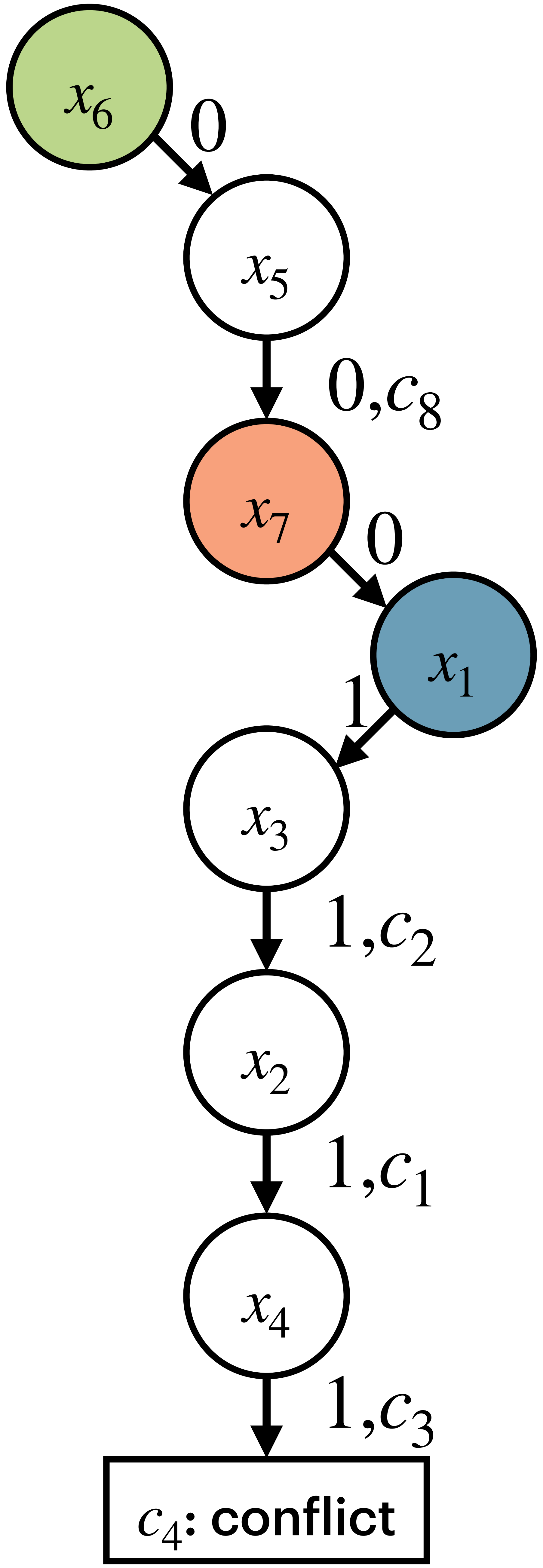
$$c_4 = (\neg x_3 \vee \neg x_4)$$

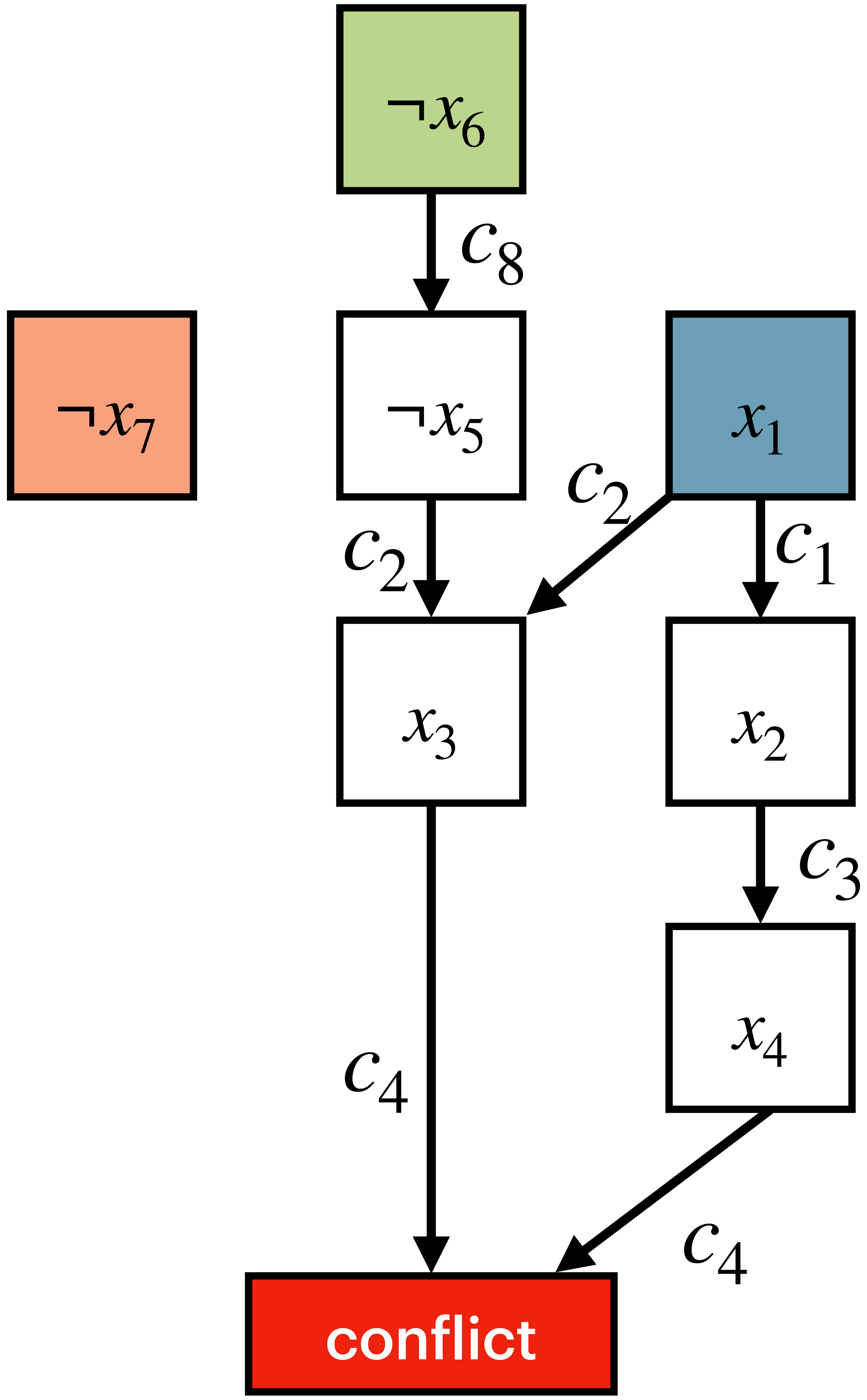
$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$





$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

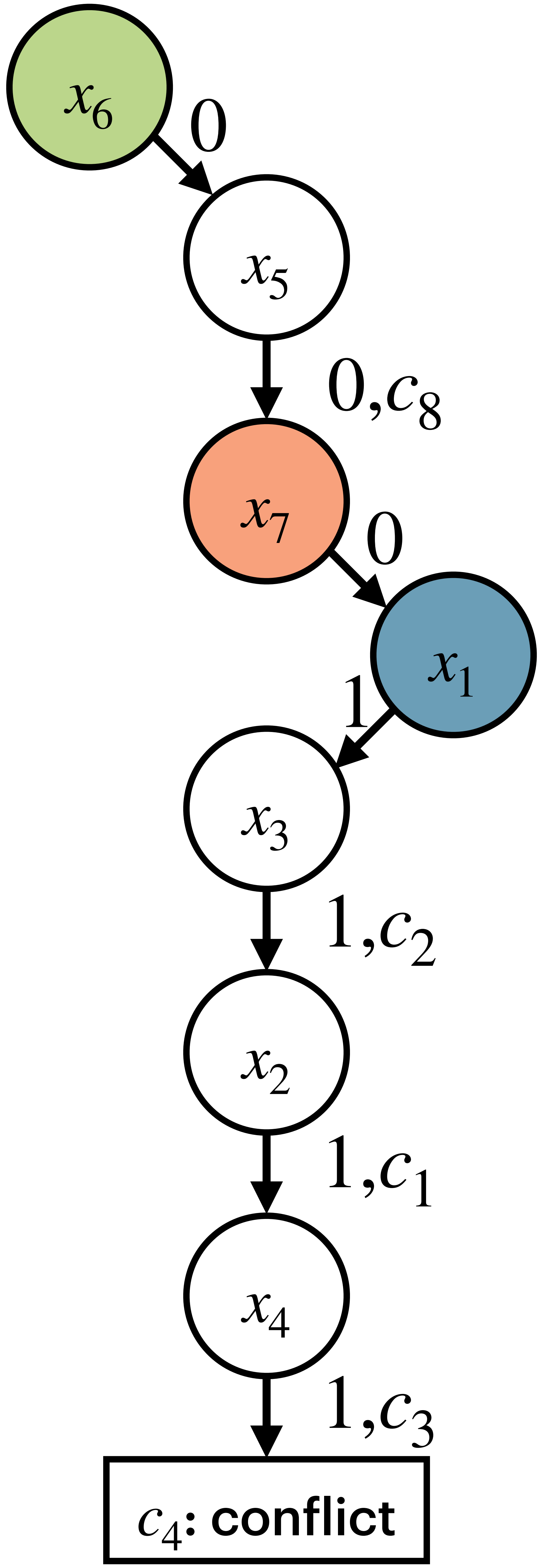
$$c_4 = (\neg x_3 \vee \neg x_4)$$

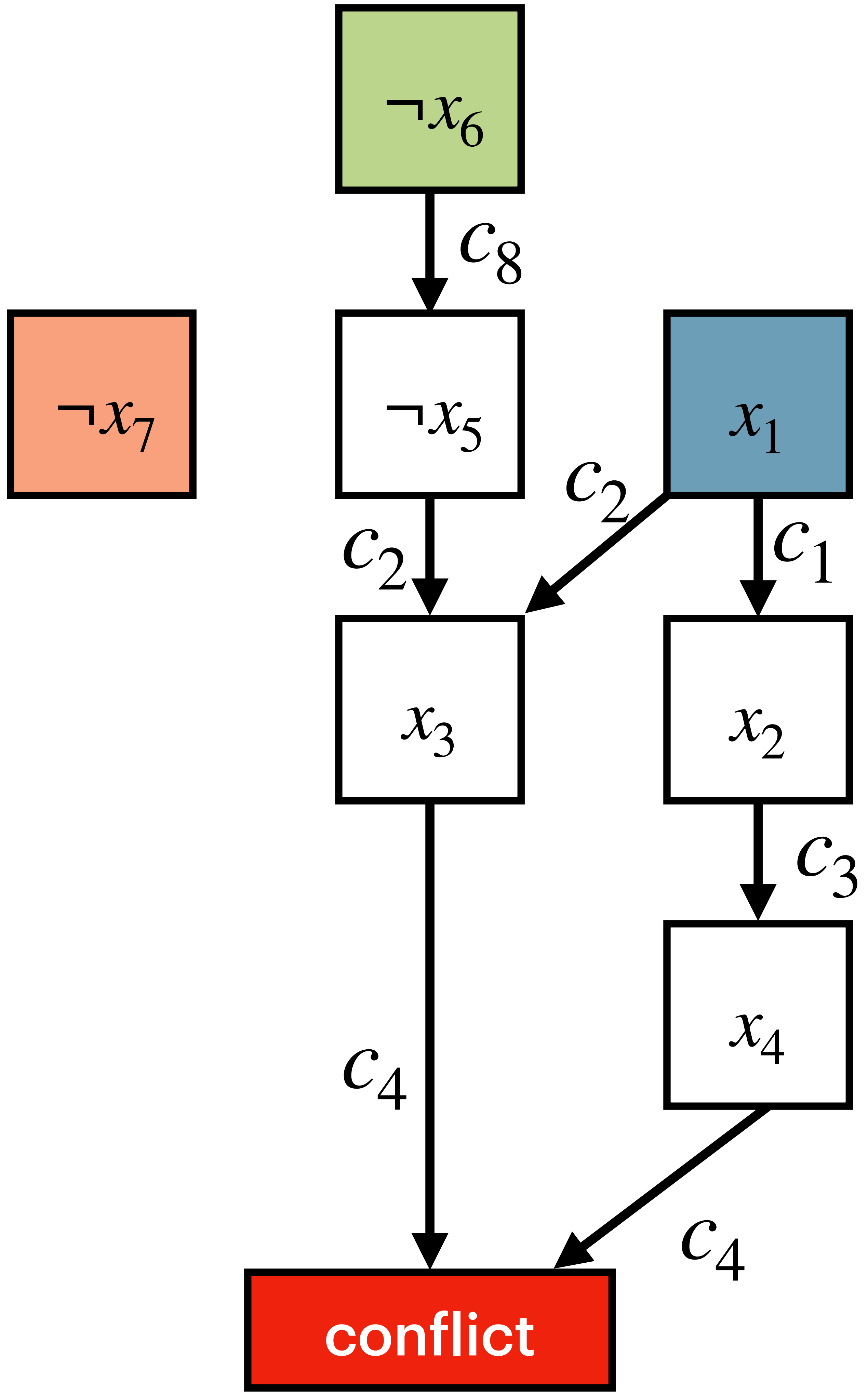
$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$





$$\neg x_6 \wedge x_1 \rightarrow \bot$$

$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

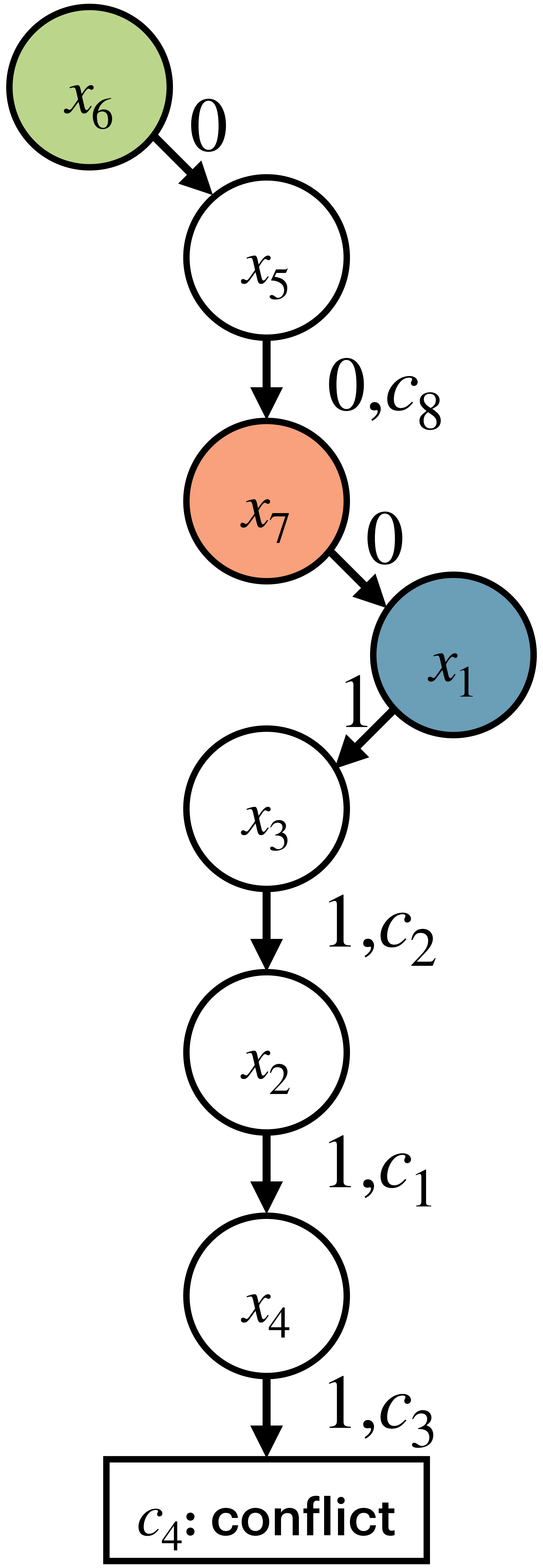
$$c_4 = (\neg x_3 \vee \neg x_4)$$

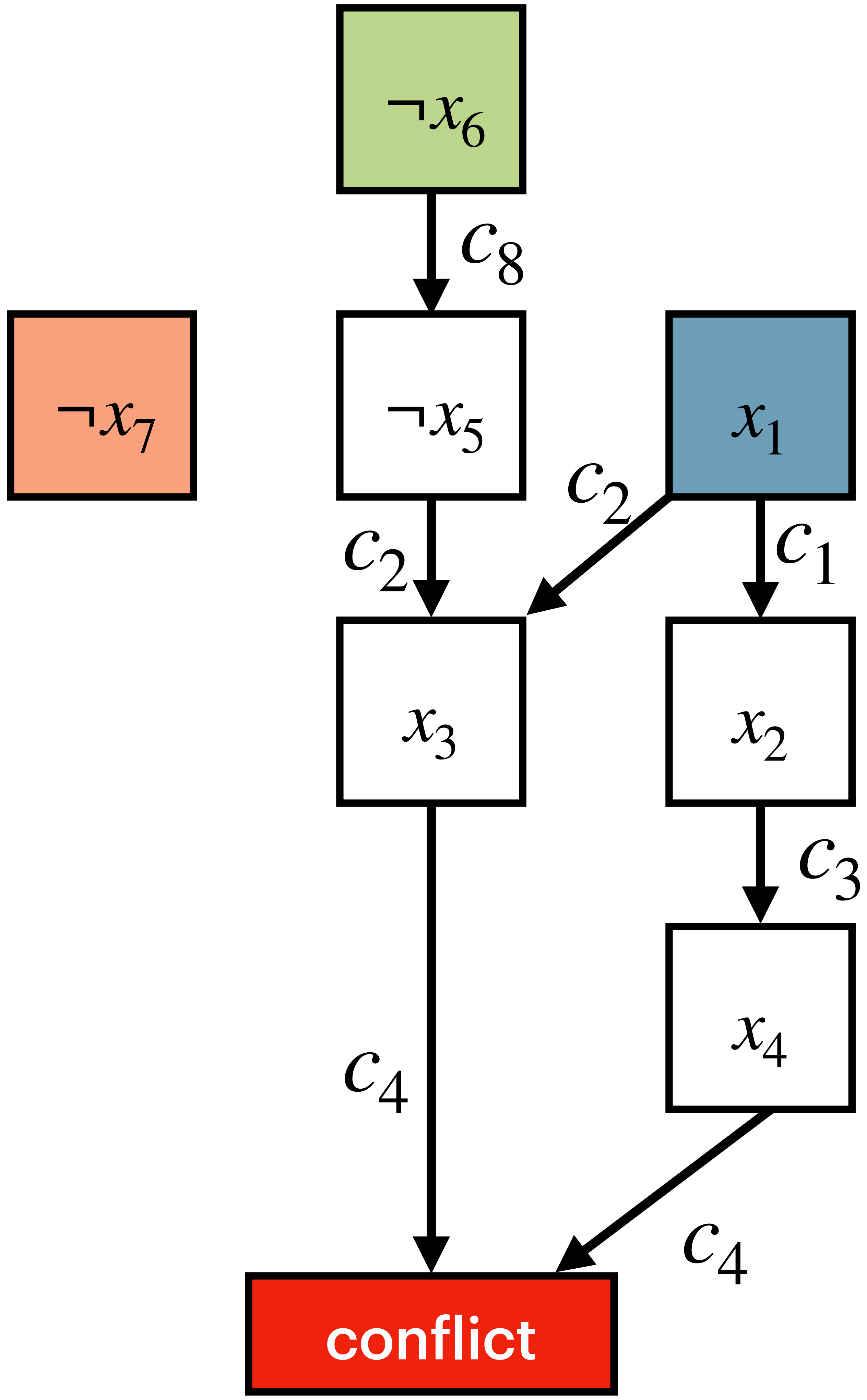
$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$





$$x_6 \vee \neg x_1$$

$$c_1 = (\neg x_1 \lor x_2)$$

$$c_2 = (\neg x_1 \lor x_3 \lor x_5)$$

$$c_3 = (\neg x_2 \lor x_4)$$

$$c_4 = (\neg x_3 \lor \neg x_4)$$

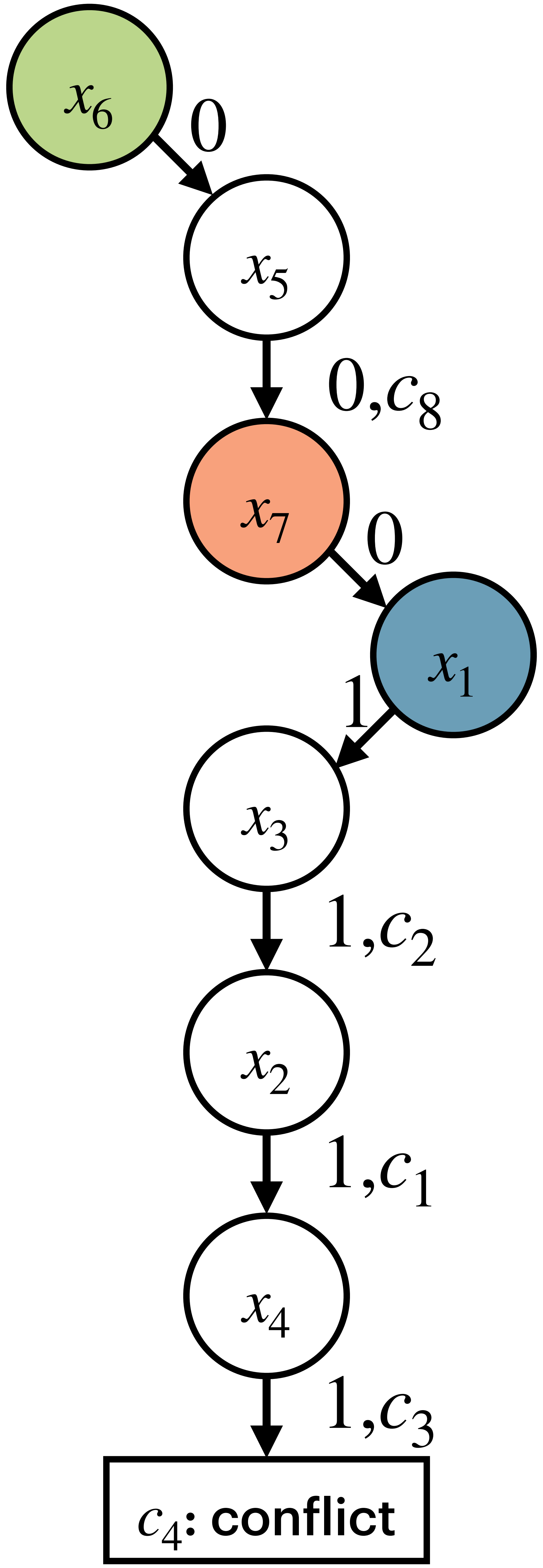
$$c_5 = (x_1 \lor x_5 \lor \neg x_2)$$

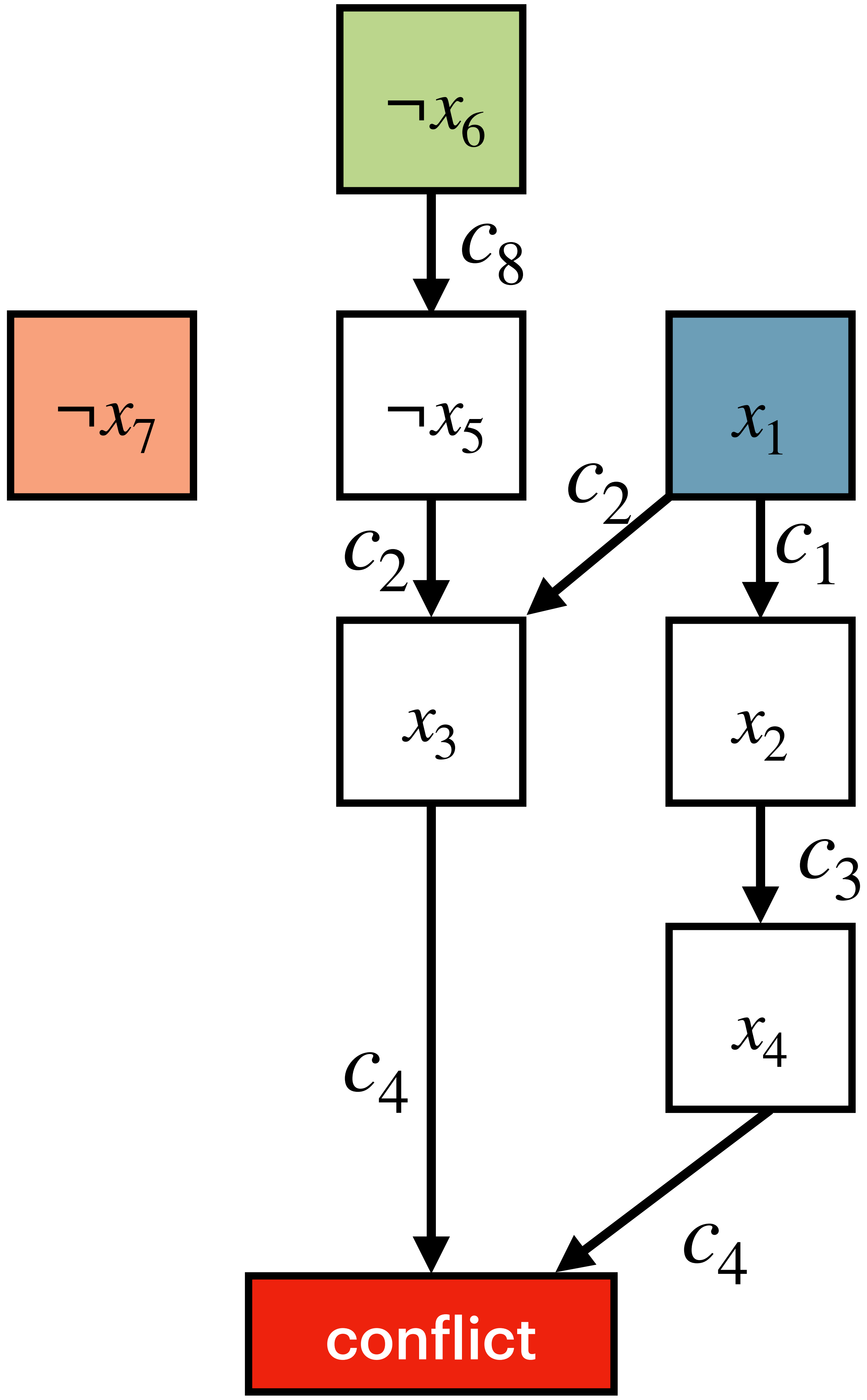
$$c_6 = (x_2 \lor x_3)$$

$$c_7 = (x_2 \lor \neg x_3 \lor x_7)$$

$$c_8 = (x_6 \lor \neg x_5)$$

$$c_9 = x_6 \lor \neg x_1$$





$c_1 = (\neg x_1 \lor x_2)$

$c_2 = (\neg x_1 \lor x_3 \lor x_5)$

$c_3 = (\neg x_2 \lor x_4)$

$c_4 = (\neg x_3 \lor \neg x_4)$

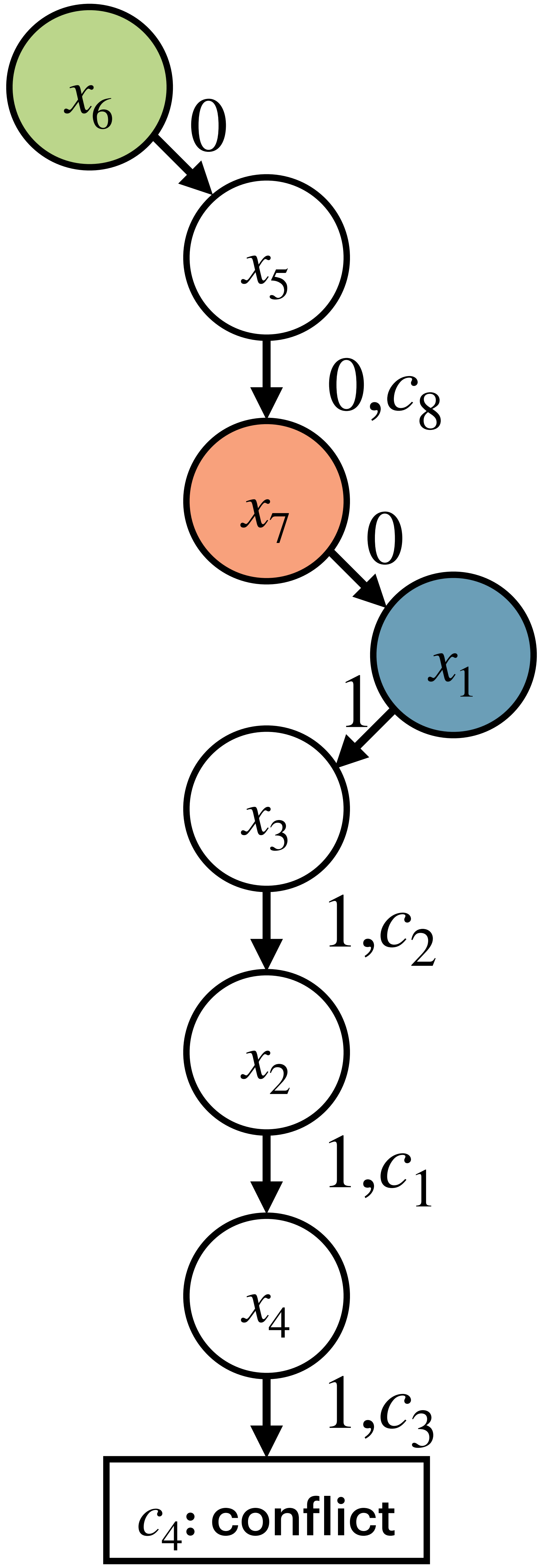
$c_5 = (x_1 \lor x_5 \lor \neg x_2)$

$c_6 = (x_2 \lor x_3)$

$c_7 = (x_2 \lor \neg x_3 \lor x_7)$

$c_8 = (x_6 \lor \neg x_5)$

$c_9 = x_6 \lor \neg x_1$



$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

$$c_4 = (\neg x_3 \vee \neg x_4)$$

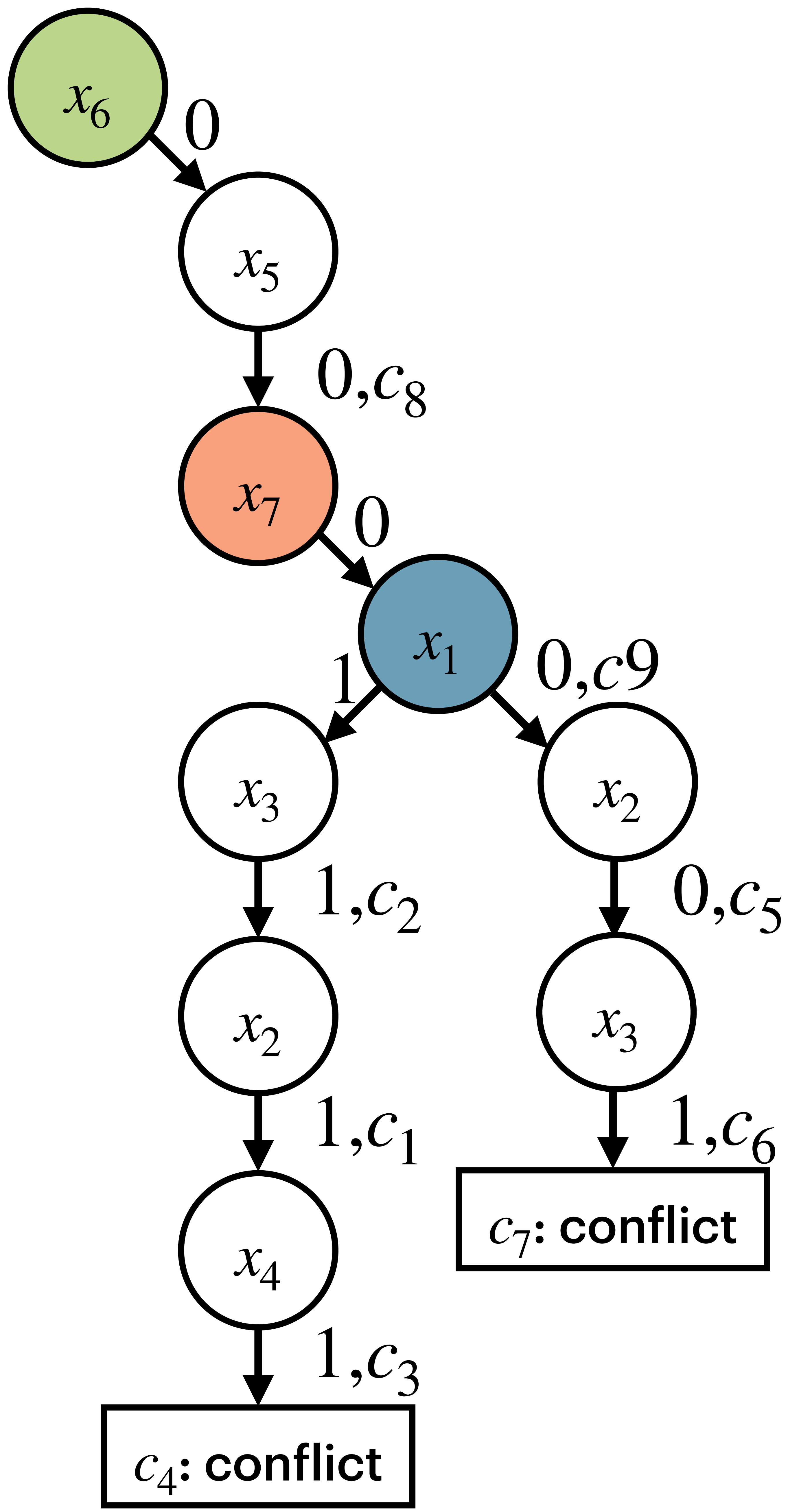
$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$

$$c_9 = x_6 \vee \neg x_1$$



$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

$$c_4 = (\neg x_3 \vee \neg x_4)$$

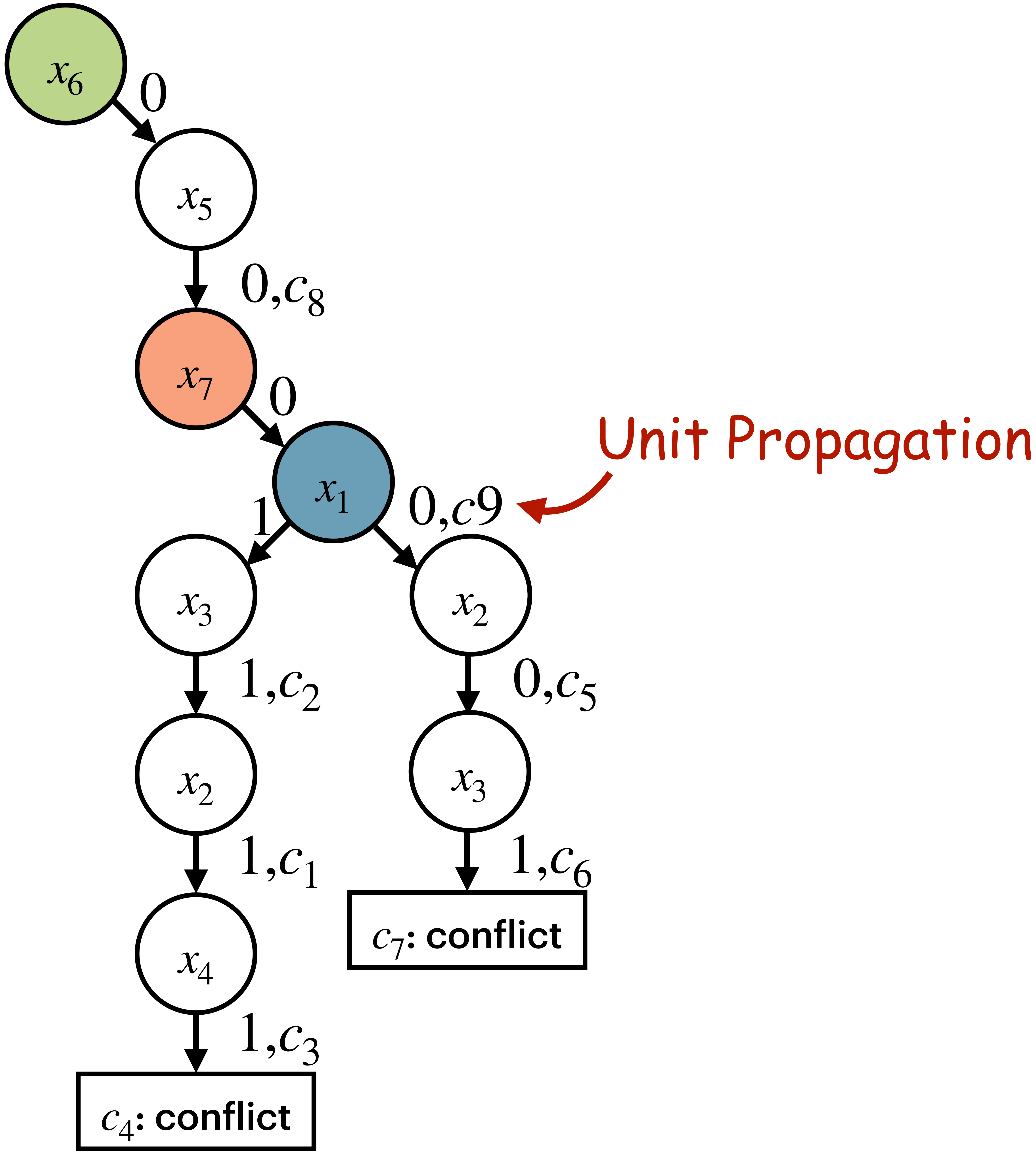
$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$

$$c_9 = x_6 \vee \neg x_1$$



$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

$$c_4 = (\neg x_3 \vee \neg x_4)$$

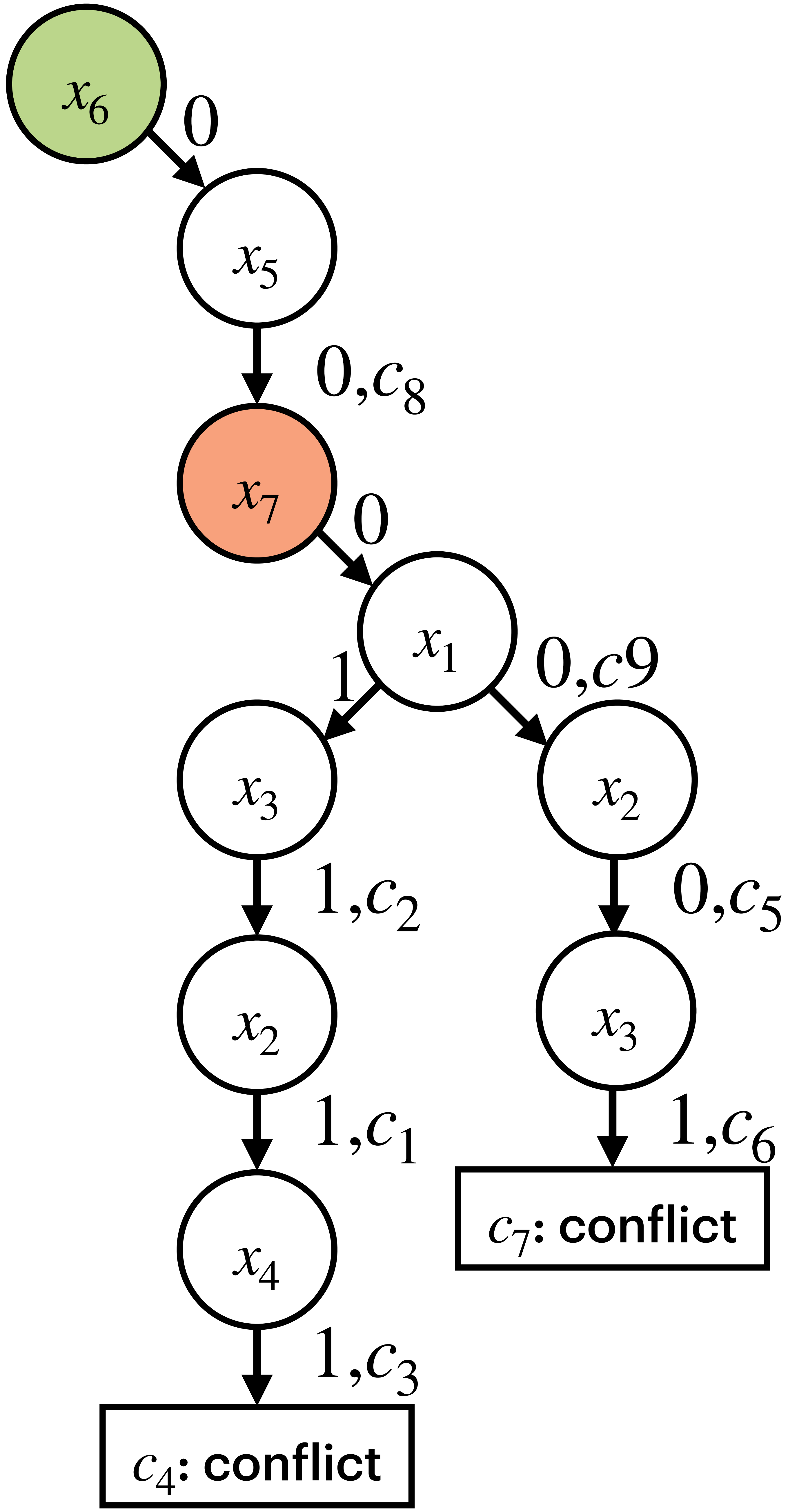
$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$

$$c_9 = x_6 \vee \neg x_1$$



$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

$$c_4 = (\neg x_3 \vee \neg x_4)$$

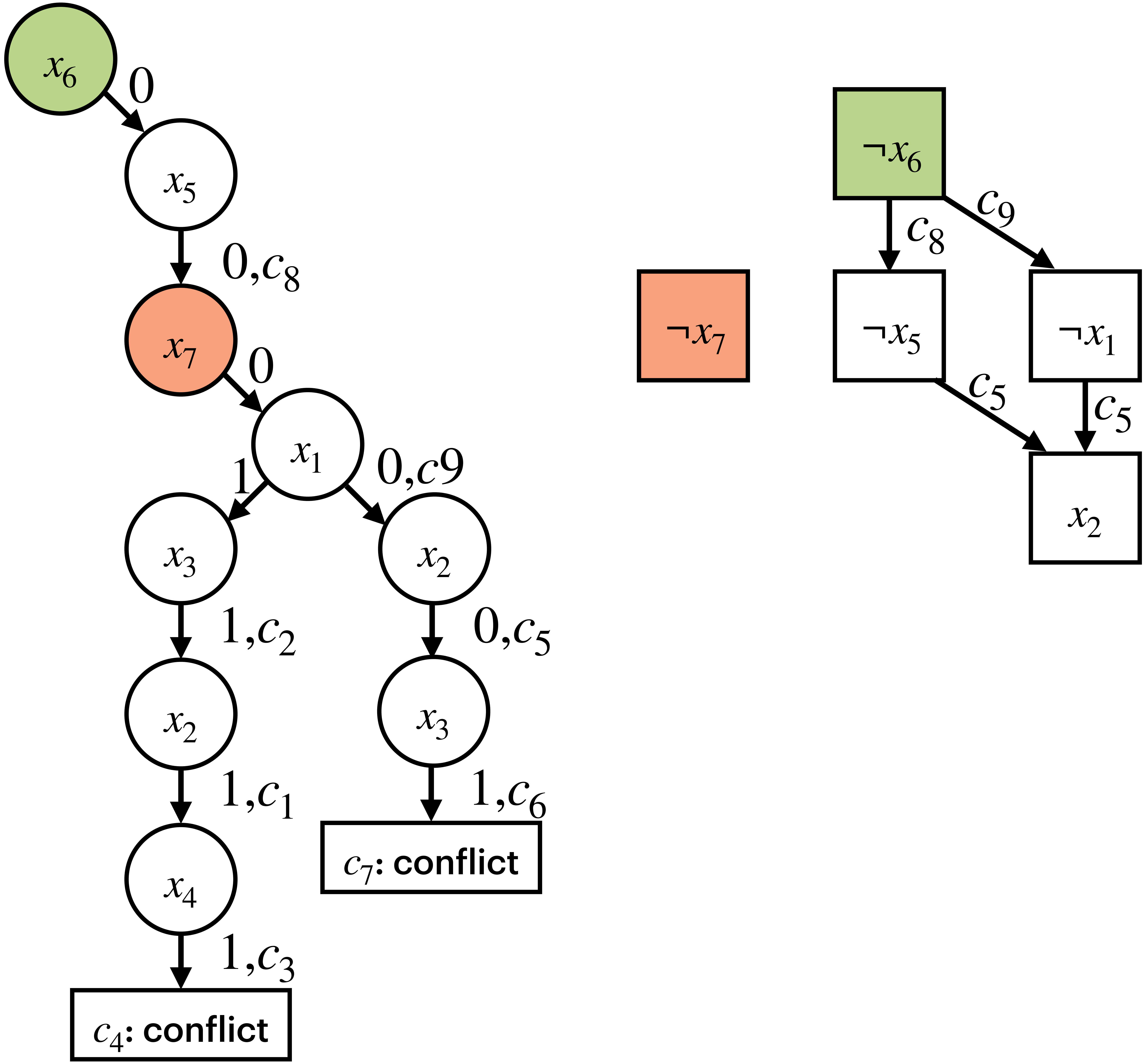
$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$

$$c_9 = x_6 \vee \neg x_1$$



$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

$$c_4 = (\neg x_3 \vee \neg x_4)$$

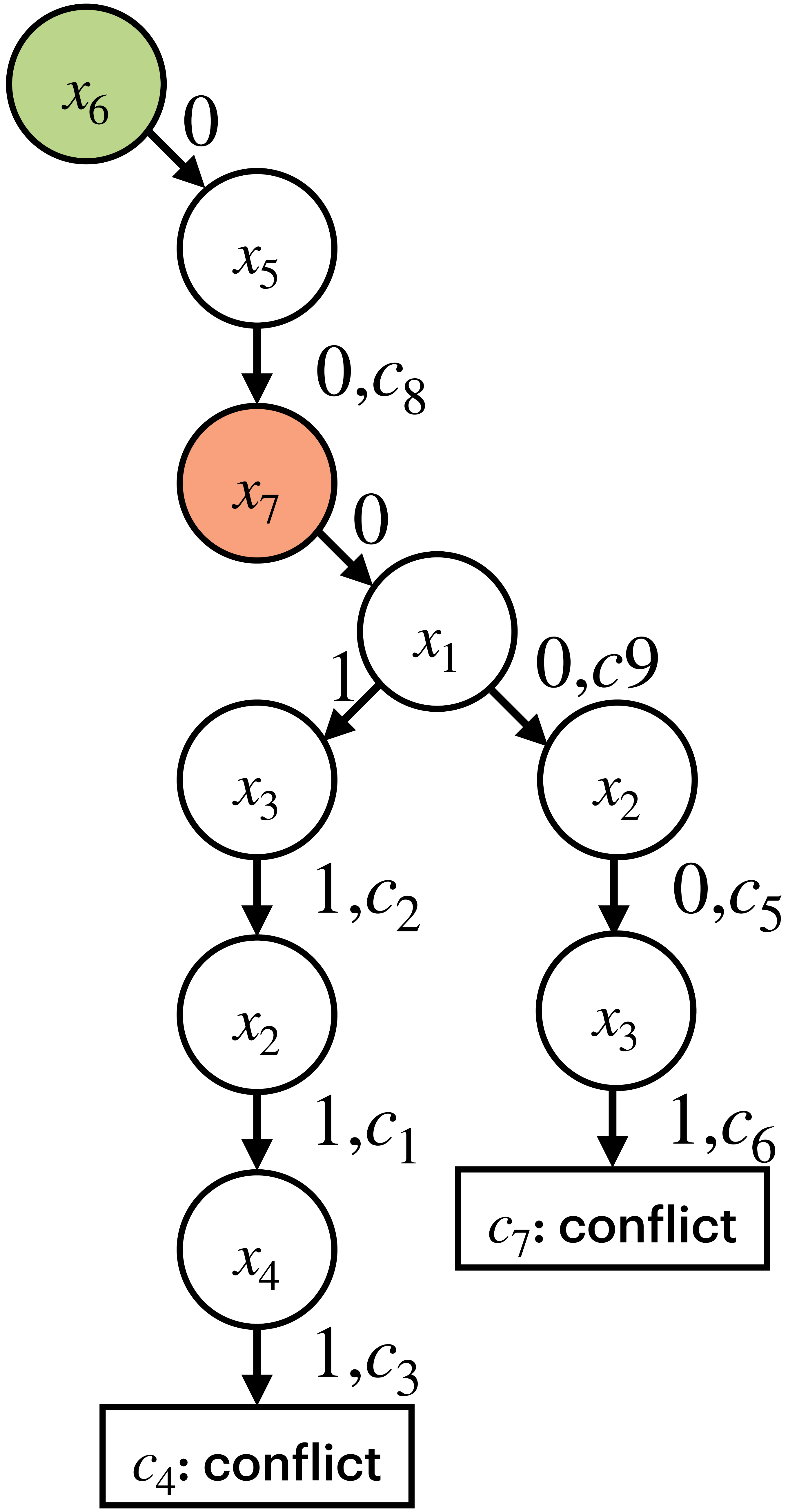
$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

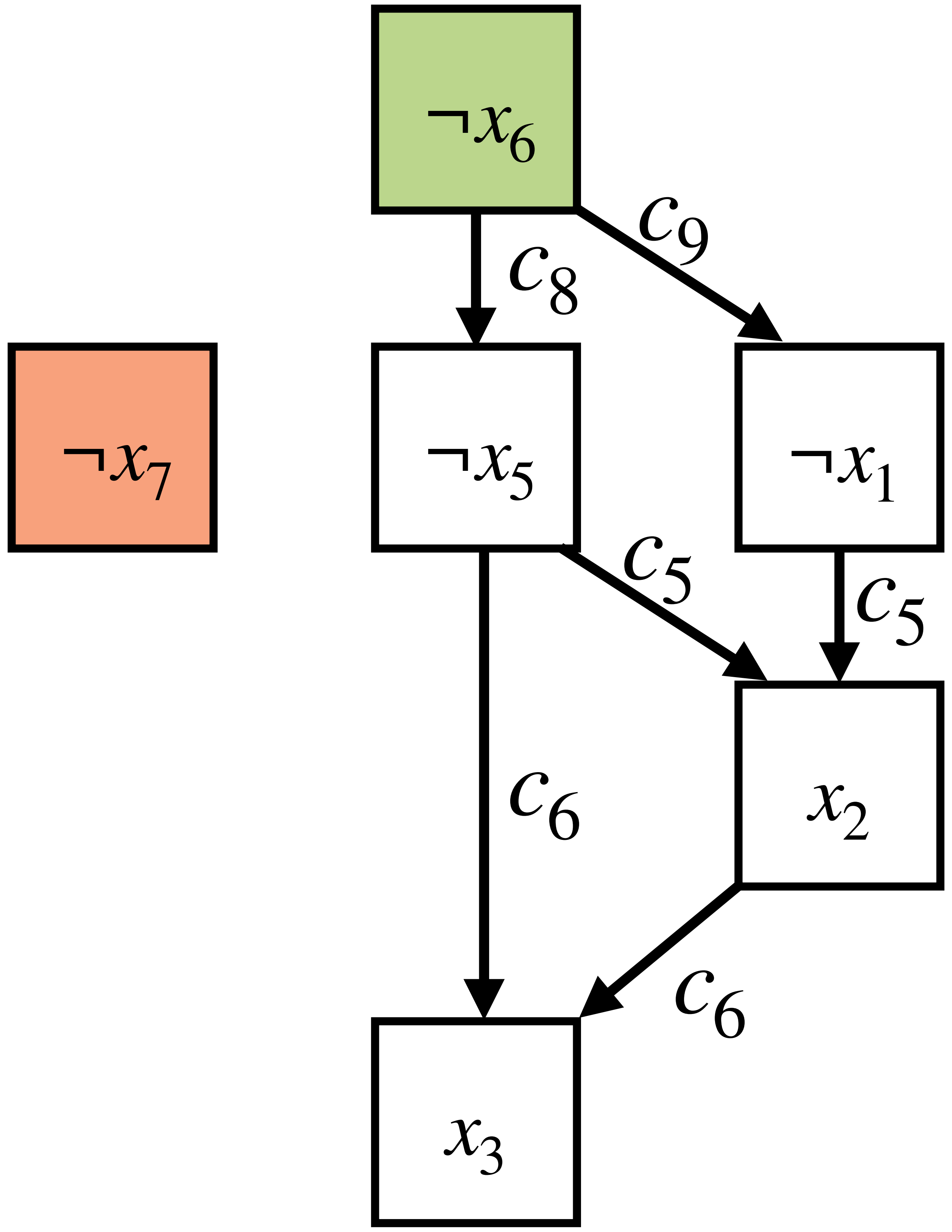
$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$

$$c_9 = x_6 \vee \neg x_1$$





$$c_1 = (\neg x_1 \lor x_2)$$

$$c_2 = (\neg x_1 \lor x_3 \lor x_5)$$

$$c_3 = (\neg x_2 \lor x_4)$$

$$c_4 = (\neg x_3 \lor \neg x_4)$$

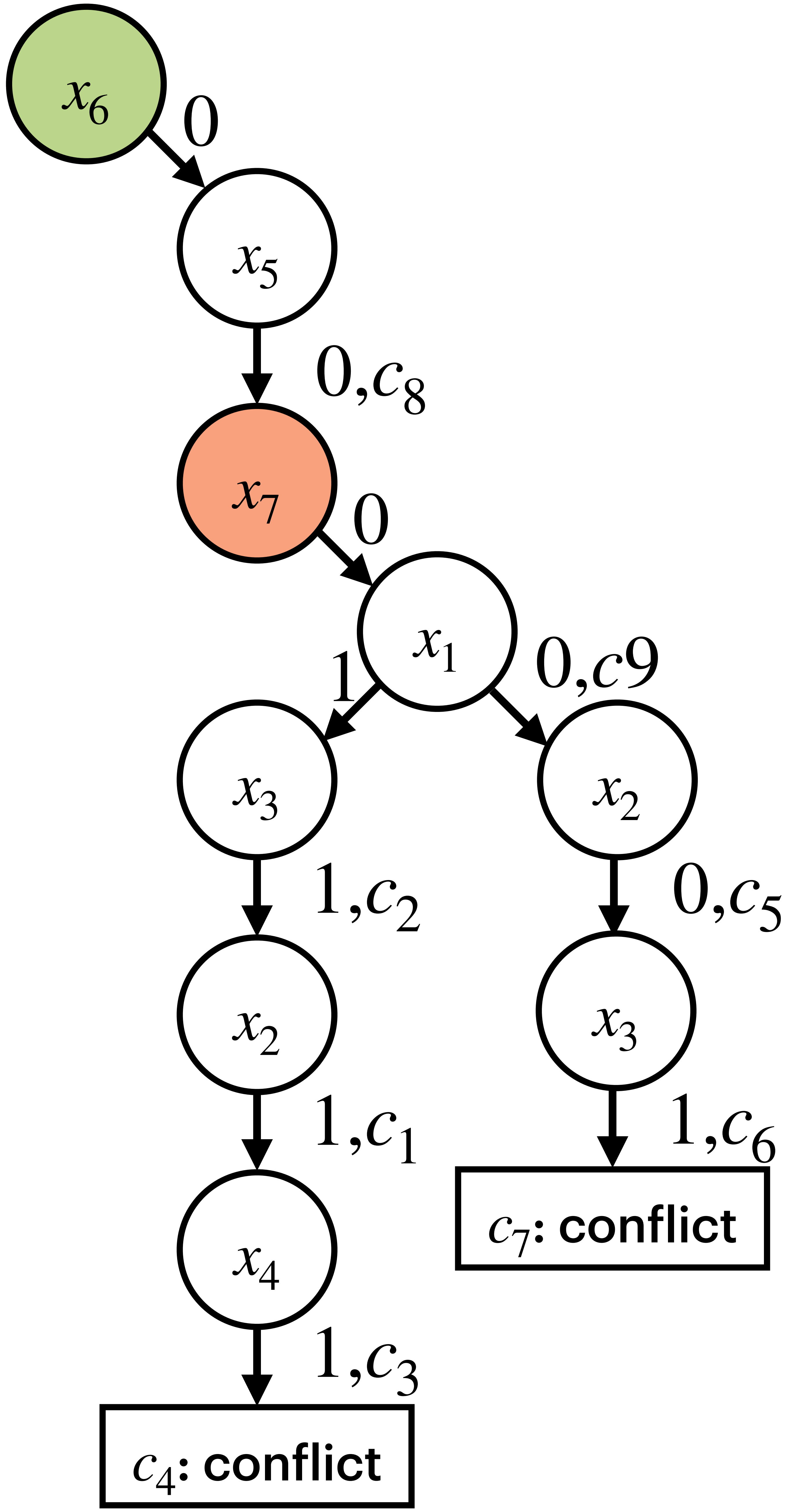
$$c_5 = (x_1 \lor x_5 \lor \neg x_2)$$

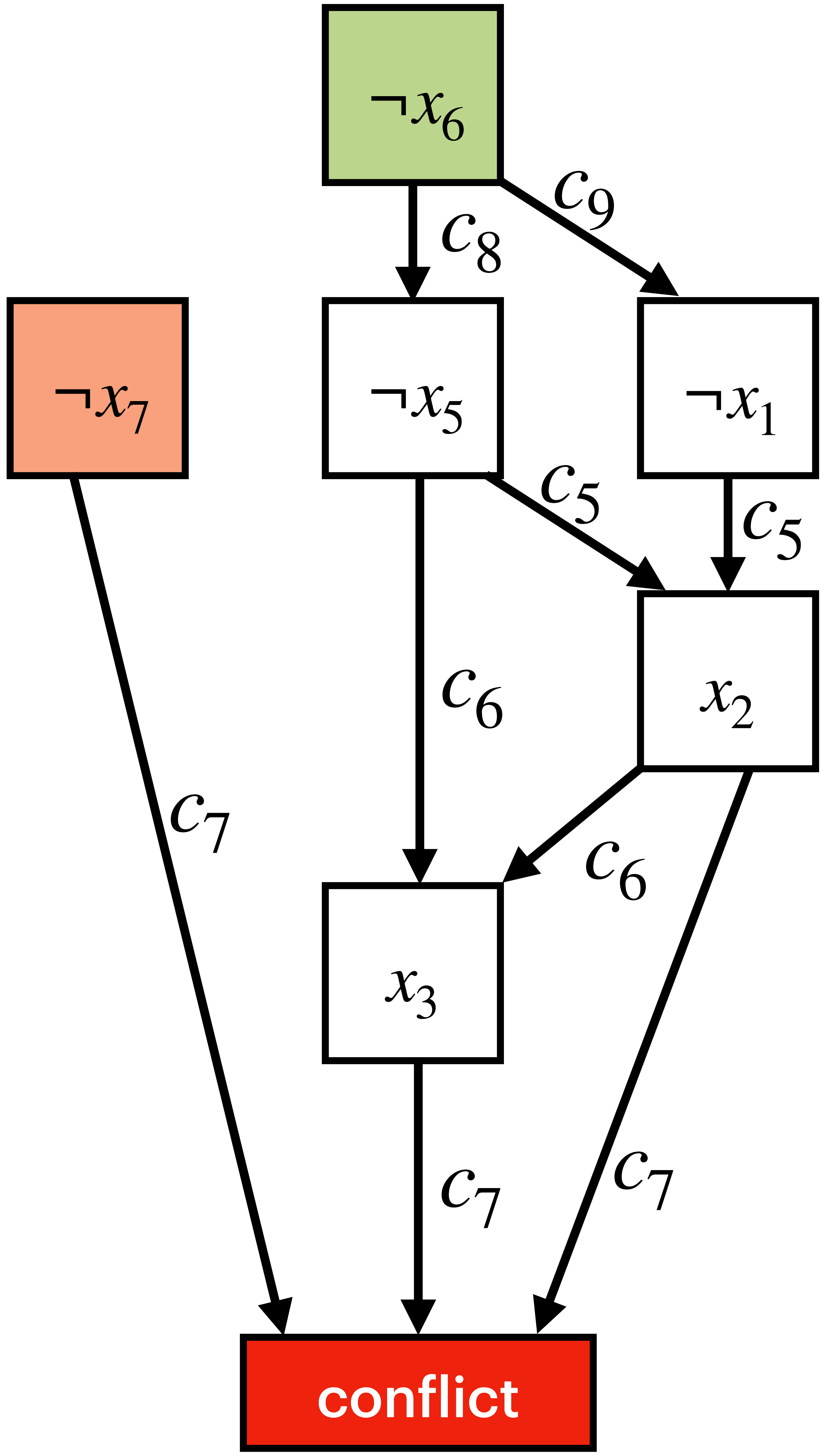
$$c_6 = (x_2 \lor x_3)$$

$$c_7 = (x_2 \lor \neg x_3 \lor x_7)$$

$$c_8 = (x_6 \lor \neg x_5)$$

$$c_9 = x_6 \lor \neg x_1$$





$$c_1 = (\neg x_1 \lor x_2)$$

$$c_2 = (\neg x_1 \lor x_3 \lor x_5)$$

$$c_3 = (\neg x_2 \lor x_4)$$

$$c_4 = (\neg x_3 \lor \neg x_4)$$

$$c_5 = (x_1 \lor x_5 \lor \neg x_2)$$

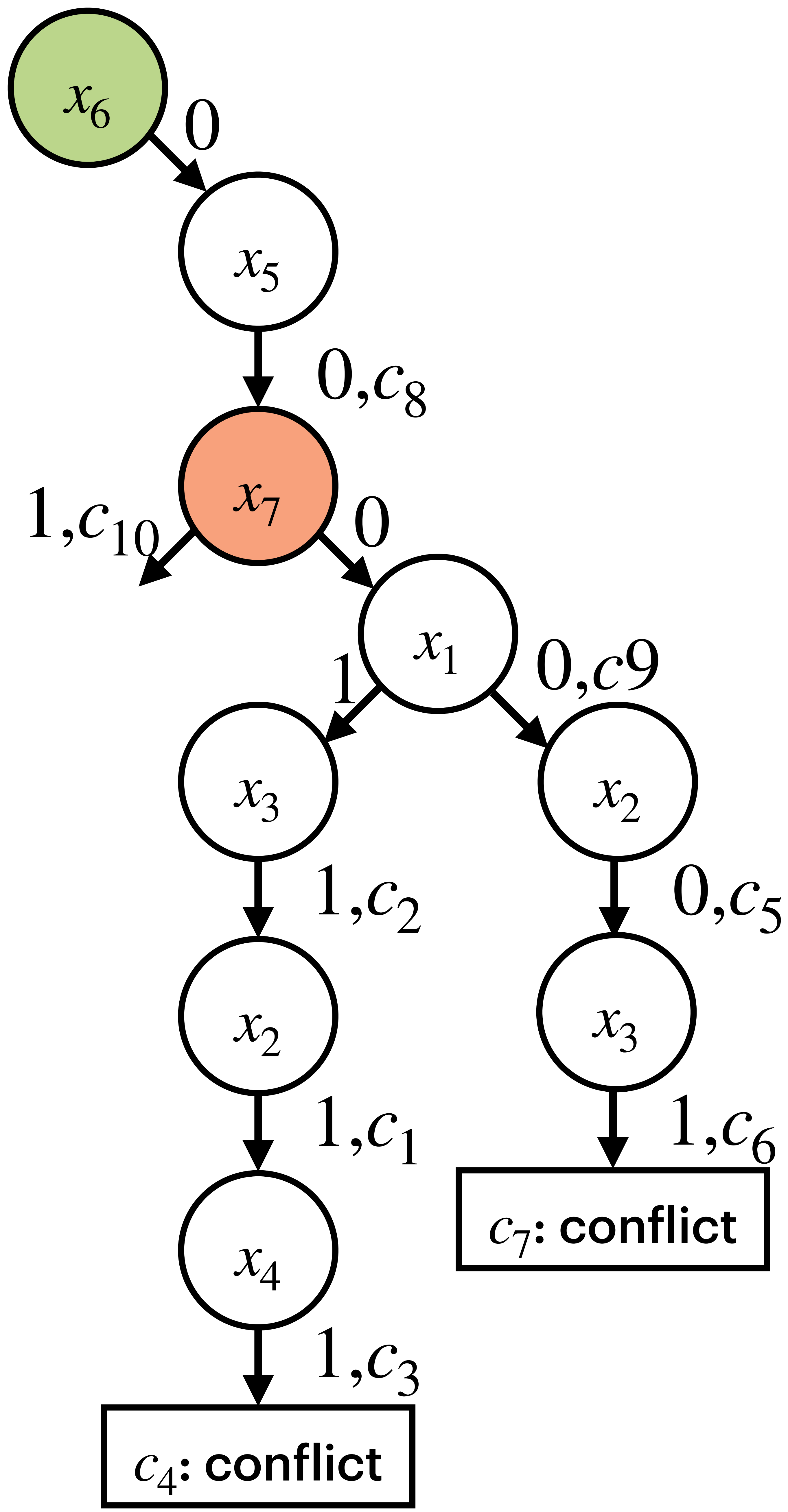
$$c_6 = (x_2 \lor x_3)$$

$$c_7 = (x_2 \lor \neg x_3 \lor x_7)$$

$$c_8 = (x_6 \lor \neg x_5)$$

$$c_9 = x_6 \lor \neg x_1$$

$$c_{10} = x_7 \lor x_6$$



$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

$$c_4 = (\neg x_3 \vee \neg x_4)$$

$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

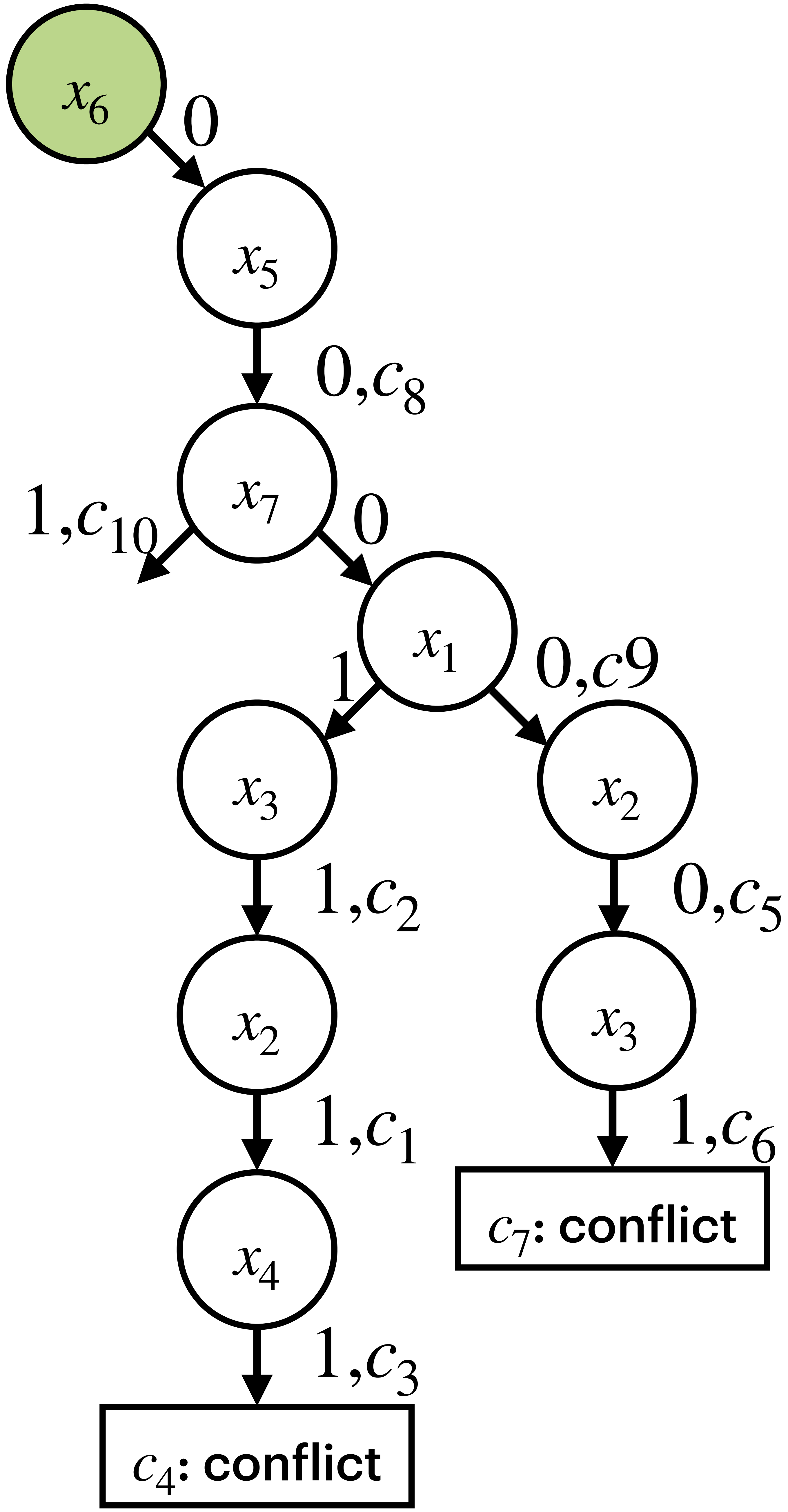
$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$

$$c_9 = x_6 \vee \neg x_1$$

$$c_{10} = x_7 \vee x_6$$



$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

$$c_4 = (\neg x_3 \vee \neg x_4)$$

$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

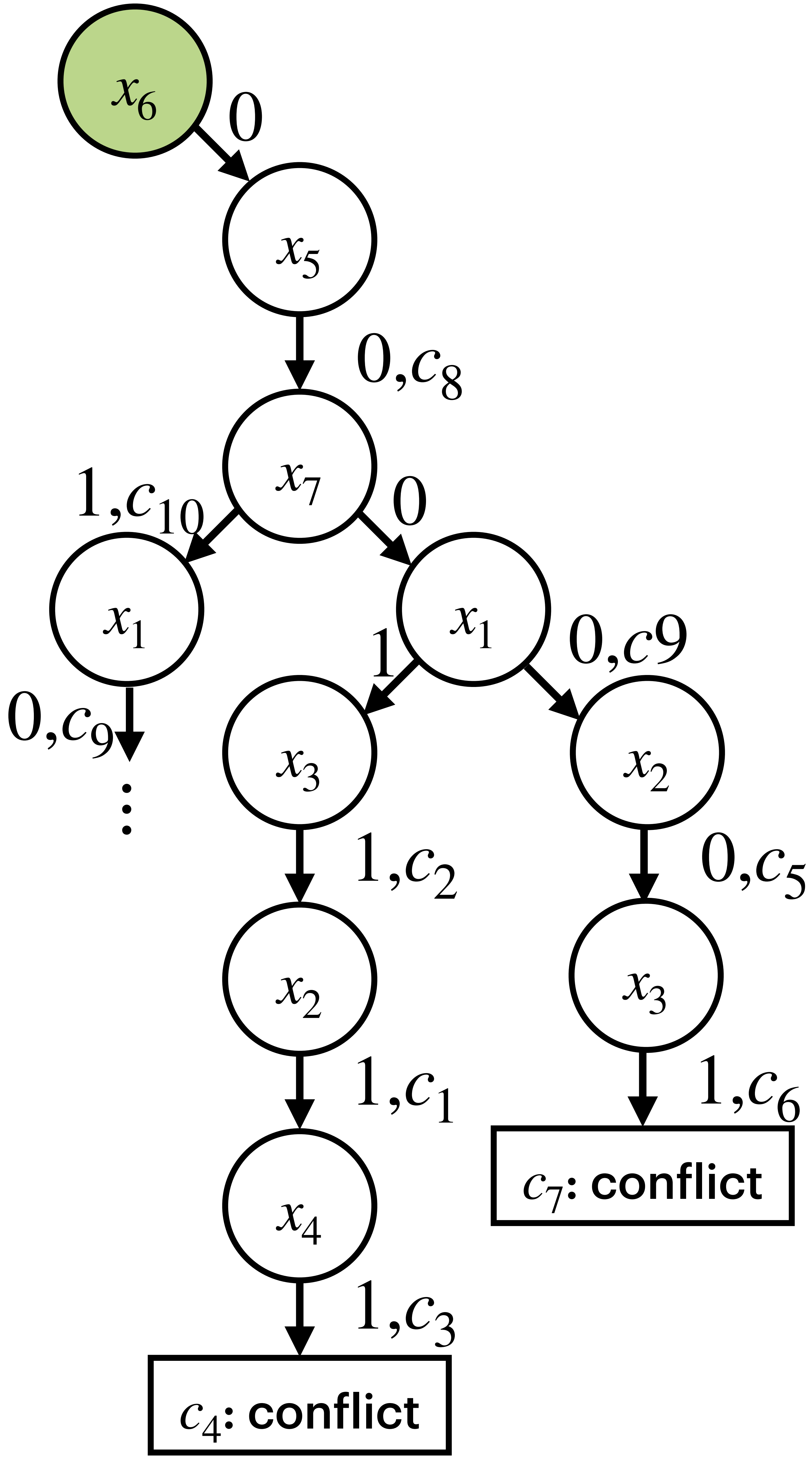
$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$

$$c_9 = x_6 \vee \neg x_1$$

$$c_{10} = x_7 \vee x_6$$



$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

$$c_4 = (\neg x_3 \vee \neg x_4)$$

$$c_5 = (x_1 \vee x_5 \vee \neg x_2)$$

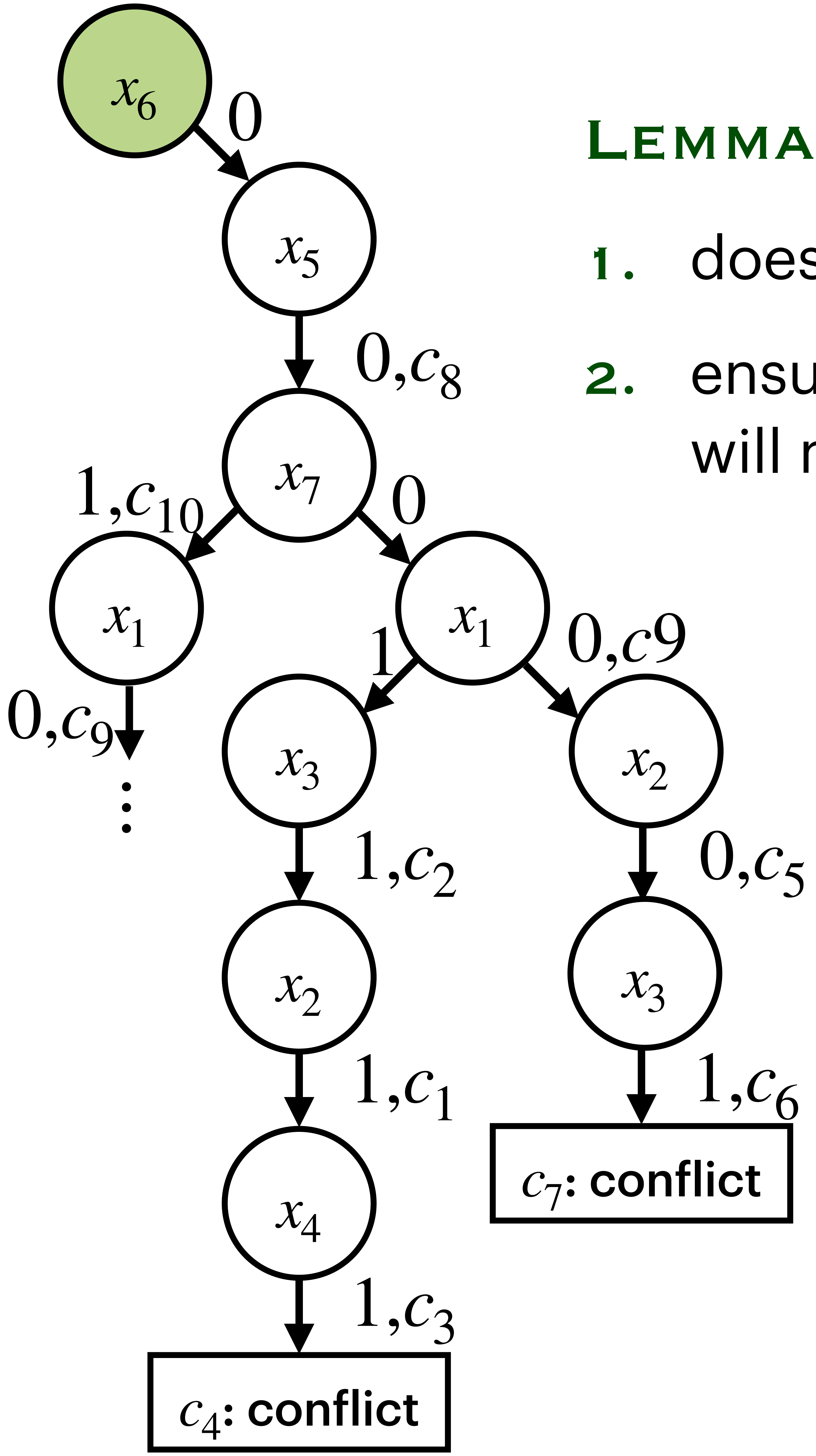
$$c_6 = (x_2 \vee x_3)$$

$$c_7 = (x_2 \vee \neg x_3 \vee x_7)$$

$$c_8 = (x_6 \vee \neg x_5)$$

$$c_9 = x_6 \vee \neg x_1$$

$$c_{10} = x_7 \vee x_6$$



LEMMA: adding conflict clauses

1. does not change satisfying assignments
2. ensures conflicting partial assignments will not be retried.

Davis–Putnam–Logemann–Loveland (DPLL) Algorithm

Input: CNF f, and partial assignment m

If f is true under m, return m.

If f is false under m, return $\bot$.

If $\exists$ unit literal p under m, then return $DPLL\left(f, m[p \to 1]\right)$.

If $\exists$ unit literal $\neg p$ under m, then return $DPLL\left(f, m[p \to 0]\right)$.

Chose an unassigned variable a, and assign it $b \in \{0,1\}$.

If $DPLL\left(f, m[a \to b]\right) = SAT$, return $m[a \to b]$

Else, return $DPLL\left(f, m[a \to 1 - b]\right)$

Conflict Driven Clause Learning (CDCL) Algorithm

Input: CNF f, and partial assignment m

If f is true under m, return m.

If f is false under m, return $\bot$.

If $\exists$ unit literal p under m, then return $DPLL\left(f, m[p \to 1]\right)$.

If $\exists$ unit literal $\neg p$ under m, then return $DPLL\left(f, m[p \to 0]\right)$.

Chose an unassigned variable a, and assign it $b \in \{0,1\}$.

If $DPLL\left(f, m[a \to b]\right) = SAT$, return $m[a \to b]$

Else, return $DPLL\left(f, m[a \to 1 - b]\right)$

Conflict Driven Clause Learning (CDCL) Algorithm

Input: CNF f, and partial assignment m

If f is true under m, return m.

If f is false under m, return $\bot$.

UnitPropagation(m, f)

Chose an unassigned variable a, and assign it $b \in \{0,1\}$.

If $DPLL\left(f, m[a \rightarrow b]\right) = SAT$, return $m[a \rightarrow b]$

Else, return $DPLL\left(f, m[a \rightarrow 1 - b]\right)$

Conflict Driven Clause Learning (CDCL) Algorithm

Input: CNF f, and partial assignment m

If f is true under m, return m.

If f is false under m:

UnitPropagation(m, f)

Chose an unassigned variable a, and assign it $b \in \{0,1\}$.

If $DPLL\left(f, m[a \to b]\right) = SAT$, return $m[a \to b]$

Else, return $DPLL\left(f, m[a \to 1 - b]\right)$

Conflict Driven Clause Learning (CDCL) Algorithm

Input: CNF f, and partial assignment m

If f is true under m, return m.

If f is false under m:

Let $c =$ Analyse conflict(m, f); $f := f \cup \{c\}$.

UnitPropagation(m, f)

Chose an unassigned variable a, and assign it $b \in \{0,1\}$.

If $DPLL\left(f, m[a \to b]\right) = SAT$, return $m[a \to b]$

Else, return $DPLL\left(f, m[a \to 1 - b]\right)$

Conflict Driven Clause Learning (CDCL) Algorithm

Input: CNF f

$m := 0;\ dl := 0;\ dstack := \{\}$

$m =$ UnitPropagation(m, f).

While $(m \not\models f)$ or (m is partial):

Conflict Driven Clause Learning (CDCL) Algorithm

Input: CNF f

$m := 0;\ dl := 0;\ dstack := \{\}$

m = UnitPropagation(m, f).

While ($m \not\models f$) or (m is partial):

While $m \not\models f$:

If $dl = 0$, return $\bot$

(c, dl) = AnalyseConflict(m, f). $f := f \cup \{c\}$.

m = UnitPropagation(m, f).

Conflict Driven Clause Learning (CDCL) Algorithm

Input: CNF f

$m := 0;\ dl := 0;\ dstack := \{\}$

m = UnitPropagation(m, f).

While $(m \not\models f)$ or (m is partial):

While $m \not\models f$:

If $dl = 0$, return $\perp$

(c, dl) = AnalyseConflict(m, f). $f := f \cup \{c\}$.

m = UnitPropagation(m, f).

If m is partial:

(m, dl) = Decide(m, f).

m = UnitPropagation(m, f).

Conflict Driven Clause Learning (CDCL) Algorithm

Input: CNF f

$m := 0;\ dl := 0;\ dstack := \{\}$

$m =$ UnitPropagation(m, f).

While $(m \nvDash f)$ or (m is partial):

While $m \nvDash f$:

If $dl = 0$, return $\bot$

$(c, dl) =$ AnalyseConflict(m, f). $f := f \cup \{c\}$.

$m =$ UnitPropagation(m, f).

If m is partial:

$(m, dl) =$ Decide(m, f).

$m =$ UnitPropagation(m, f).

DLIS (Dynamic Largest Individual Sum): Chooses a literal l with maximal occurrences in f.

DLCS (Dynamic Largest Clause Sum): Chooses a literal l with maximal occurrences of l and $\neg l$ in f.

MOM (Maximum Occurrence in Minimal Size Clauses): Let k be the shortest size clause in f. Choose a literal l with maximal occurrences of l and $\neg l$ in k-sized clauses of f.

Time to Code!

Greedy Search?

Start with any assignment.

Flip the variable that minimises the number of unsatisfied clauses.

Greedy Search?

Start with any assignment.

Flip the variable that minimises the number of unsatisfied clauses.

When does this not work?

Randomized Greedy Search?

Start with any assignment.

With probability p,
Flip the variable that minimises the number of unsatisfied clauses.

With probability $(1 - p)$,
Flip any variable in an unsatisfied clause.

Randomized Greedy Search?

Start with any assignment.

With probability p,
Flip the variable that minimises the number of unsatisfied clauses.

With probability $(1 - p)$,
Flip any variable in an unsatisfied clause.

Random Walk

WalkSAT Algorithm:

Start with any assignment.

With probability p,
Flip the variable that minimises the number of unsatisfied clauses.

With probability $(1 - p)$,
Flip any variable in an unsatisfied clause.

WalkSAT Algorithm:

Start with any assignment.

With probability p,
Flip the variable that minimises the number of unsatisfied clauses.

With probability $(1 - p)$,
Flip any variable in an unsatisfied clause.

WalkSAT Algorithm:

Start with any assignment.

If there is a variable that can be flipped such that it does not turn any satisfied clauses into unsatisfied, flip it. Otherwise:

With probability p,
Flip the variable that minimises the number of unsatisfied clauses.

With probability $(1 - p)$,
Flip any variable in an unsatisfied clause.

WalkSAT Algorithm:

Start with any assignment.

If there is a variable that can be flipped such that it does not turn any satisfied clauses into unsatisfied, flip it. Otherwise:

With probability p,
Flip the variable that minimises the number of unsatisfied clauses.

With probability $(1 - p)$,
Flip any variable in an unsatisfied clause.

Hill Climbing + Random Walk

An Extension:

$$p := A \mid p \wedge p \mid p \vee p \mid \neg p$$

An Extension:

$$p := A \mid p \wedge p \mid p \vee p \mid \neg p$$

$$A := (e = e)$$

An Extension:

$$p := A \mid p \wedge p \mid p \vee p \mid \neg p$$

$$A := (e = e)$$

$$e \in \mathbb{R} \cup V \qquad e := e \backsim e$$

$$\backsim \; := + \mid -$$

An Example:

$$p : \neg(x = 0) \wedge \big((x + y = 3.5) \vee (y - x = 2)\big)$$

Are there (x, y) such that p can be satisfied?

An Example:

$$p : \neg(x = 0) \wedge \big((x + y = 3.5) \vee (y - x = 2)\big)$$

$$A \qquad\qquad\qquad B \qquad\qquad\qquad C$$

Are there (x, y) such that p can be satisfied?

An Example:

$$p : \neg(x = 0) \wedge \big((x + y = 3.5) \vee (y - x = 2)\big)$$

$$p' : \neg A \wedge (B \vee C)$$

Are there (x, y) such that p can be satisfied?

Conflict Driven Clause Learning (CDCL) Algorithm

Input: CNF f

$m := 0;\ dl := 0;\ dstack := \{\}$

m = UnitPropagation(m, f).

While $(m \not\models f)$ or (m is partial):

While $m \not\models f$:

If $dl = 0$, return $\bot$

(c, dl) = AnalyseConflict(m, f). $f := f \cup \{c\}$.

m = UnitPropagation(m, f).

If m is partial:

(m, dl) = Decide(m, f).

m = UnitPropagation(m, f).

An Example:

$$p : \neg(x = 0) \wedge \big((x + y = 3.5) \vee (y - x = 2)\big)$$

$$p' : \neg A \wedge (B \vee C)$$

Are there (x, y) such that p can be satisfied?

$$\{A : 0, B : 1, C : 0\}$$

$$\{A : 0, B : 0, C : 1\}$$

$$\{A : 0, B : 1, C : 1\}$$

An Example:

$$p : \neg(x = 0) \wedge \big((x + y = 3.5) \vee (y - x = 2)\big)$$

$$p' : \neg A \wedge (B \vee C)$$

Are there (x, y) such that p can be satisfied?

$$\big\{(x, y) \,|\, x \neq 0,\, x + y = 3.5,\, y - x = 2\big\}$$

$$\{A : 0,\, B : 1,\, C : 1\}$$

An Example:

$$p : \neg(x = 0) \wedge \big((x + y = 3.5) \vee (y - x = 2)\big)$$

$$p' : \neg A \wedge (B \vee C)$$

Are there (x, y) such that p can be satisfied?

$$\big\{(x, y) \,|\, x \neq 0,\ x + y = 3.5,\ y - x = 2\big\}$$

$$\big\{(0.75, 2.75)\big\}$$

$$\{A : 0,\ B : 1,\ C : 1\}$$

An Example:

$$p : \neg(x = 0) \wedge \big((x + y = 3.5) \vee (y + x = 2)\big)$$

$$p' : \neg A \wedge (B \vee C)$$

Are there (x, y) such that p can be satisfied?

$$\big\{(x, y) \,|\, x \neq 0,\ x + y = 3.5,\ y + x = 2\big\}$$

$$\{A : 0,\ B : 1,\ C : 1\}$$

An Example:

$$p : \neg(x = 0) \land \big((x + y = 3.5) \lor (y + x = 2)\big)$$

$$p' : \neg A \land (B \lor C)$$

Are there (x, y) such that p can be satisfied?

$$\{A : 0,\ B : 1,\ C : 0\}$$

$$\{A : 0,\ B : 0,\ C : 1\}$$

~~$\{A : 0,\ B : 1,\ C : 1\}$~~

An Example:

$$p : \neg(x = 0) \wedge \big((x + y = 3.5) \vee (y + x = 2)\big)$$

$$p' : \neg A \wedge (B \vee C)$$

Are there (x, y) such that p can be satisfied?

$$\{A : 0,\ B : 1,\ C : 0\}$$

$$\{A : 0,\ B : 0,\ C : 1\}$$

$$\sout{\{A : 0,\ B : 1,\ C : 1\}}$$

Conflict Driven Clause Learning (CDCL) Algorithm

Input: CNF f

$m := 0;\ dl := 0;\ dstack := \{\}$

$m =$ UnitPropagation(m, f).

While $(m \not\models f)$ or (m is partial):

While $m \not\models f$:

If $dl = 0$, return $\bot$

$(c, dl) =$ AnalyseConflict(m, f). $f := f \cup \{c\}$.

$m =$ UnitPropagation(m, f).

If m is partial:

$(m, dl) =$ Decide(m, f).

$m =$ UnitPropagation(m, f).

Why does this work?

$$p : \neg(x = 0) \wedge \big((x + y = 3.5) \vee (y - x = 2)\big)$$

$$p' : \neg A \wedge (B \vee C)$$

Are there (x, y) such that p can be satisfied?

$$\big\{(x, y) \,|\, x \neq 0,\, x + y = 3.5,\, y - x = 2\big\}$$

$$\big\{(0.75,2.75)\big\}$$

$$\{A : 0,\, B : 1,\, C : 1\}$$

Why does this work?

$$p : \neg(x = 0) \wedge \big((x + y = 3.5) \vee (y - x = 2)\big)$$

$$p' : \neg A \wedge (B \vee C)$$

Are there (x, y) such that p can be satisfied?

$$\big\{(x, y) \,|\, x \neq 0,\, x + y = 3.5,\, y - x = 2\big\}$$ Decidability of SLE

$$\big\{(0.75, 2.75)\big\}$$

$$\{A : 0,\, B : 1,\, C : 1\}$$

More Theories

$$p_1 : \big(y = g(y)\big) \wedge \big(f(x) = f(g(x))\big)$$

$$p_2 : (i \neq j) \rightarrow read(write(a, i, v), j) = read(a, j)$$

$$\big(read(A, x) = y\big) \wedge \big(f(x) = f(y)\big) \wedge \big(2x > y\big)$$

Theory of Equality Logic with Uninterpreted Functions

$$p = A \mid p \wedge p \mid p \vee p \mid p \rightarrow p \mid \neg p$$

$$A := e = e$$

$$e := f(e) \mid x$$

Theory of Equality Logic with Uninterpreted Functions

$$p = A \mid p \wedge p \mid p \vee p \mid p \rightarrow p \mid \neg p$$

$$A := e = e$$

$$e := f(e) \mid x$$

$$\forall x \,.\, (x = x)$$

$$\forall x, y \,.\, (x = y) \rightarrow (y = x)$$

$$\forall x, y, z \,.\, (x = y) \wedge (y = z) \rightarrow (x = z)$$

$$\forall x, y \,.\, (x = y) \rightarrow \big(f(x) = f(y)\big)$$

Theory of Equality Logic with Uninterpreted Functions

$$f(f(f(a))) = a \quad \land \quad f(f(f(f(f(a))))) = a \quad \land \quad \neg\big(f(a) = a\big)$$

$$f(f(f(a))) = a$$

Theory of Equality Logic with Uninterpreted Functions

$$f(f(f(a))) = a \quad \wedge \quad f(f(f(f(f(a))))) = a \quad \wedge \quad \neg\big(f(a) = a\big)$$

$$f(f(f(f(a)))) = f(a)$$

Theory of Equality Logic with Uninterpreted Functions

$$f(f(f(a))) = a \quad \land \quad f(f(f(f(f(a))))) = a \quad \land \quad \neg\big(f(a) = a\big)$$

$$f(f(f(f(f(a))))) = f(f(a))$$

Theory of Equality Logic with Uninterpreted Functions

$$f(f(f(a))) = a \quad \wedge \quad f(f(f(f(f(a))))) = a \quad \wedge \quad \neg\big(f(a) = a\big)$$

$$a = f(f(a))$$

Theory of Equality Logic with Uninterpreted Functions

$$f(f(f(a))) = a \quad \wedge \quad f(f(f(f(f(a))))) = a \quad \wedge \quad \neg\left(f(a) = a\right)$$

$$f(a) = f(f(f(a)))$$

Theory of Equality Logic with Uninterpreted Functions

$$f(f(f(a))) = a \quad \wedge \quad f(f(f(f(f(a))))) = a \quad \wedge \quad \neg\big(f(a) = a\big)$$

$$f(a) = a$$

Theory of Equality Logic with Uninterpreted Functions

$$f(f(f(a))) = a \quad \wedge \quad f(f(f(f(f(a))))) = a \quad \wedge \quad \neg\big(f(a) = a\big)$$

$$\bot$$

Theory of Equality Logic with Uninterpreted Functions

$$p = A \mid p \wedge p \mid p \vee p \mid p \rightarrow p \mid \neg p$$

$$A := e = e$$

$$e := f(e) \mid x$$

$$\forall x \,.\, (x = x)$$

$$\forall x, y \,.\, (x = y) \rightarrow (y = x)$$

$$\forall x, y, z \,.\, (x = y) \wedge (y = z) \rightarrow (x = z)$$

$$\forall x, y \,.\, (x = y) \rightarrow \big(f(x) = f(y)\big)$$

Theory of Equality Logic with Uninterpreted Functions

Design an algorithm to decide if a given statement in EUF is a theorem.

Theory of Equality Logic with Uninterpreted Functions

```
int a = f(x);
int b = g(a, y);
int a = h(b, z);
return a;
```

```
int a = h(g(f(x),y), z);
return a;
```

Are these two equivalent?

Examples

$$p_1 : \big(7 = g(7)\big) \wedge \big(f(x) = f(g(x))\big)$$

$$p_2 : (i \neq j) \rightarrow read(write(a, i, v), j) = read(a, j)$$

$$\big(read(A, x) = y\big) \wedge \big(f(x) = f(y)\big) \wedge \big(2x > y\big)$$

Examples

$$p_1 : \big(7 = g(7)\big) \wedge \big(f(x) = f(g(x))\big)$$

$$p_2 : (i \neq j) \rightarrow read(write(a, i, v), j) = read(a, j)$$

$$\big(read(A, x) = y\big) \wedge \big(f(x) = f(y)\big) \wedge \big(2x > y\big)$$

Not Always!

Nelson-Oppen Theory Combination

Convert F into $F_1 \wedge F_2 \wedge \ldots \wedge F_n$ such that F_i has only terms from theory T_i

$$read(x + y, A) = write(x, y, A)$$

Nelson-Oppen Theory Combination

Convert F into $F_1 \wedge F_2 \wedge \ldots \wedge F_n$ such that F_i has only terms from theory T_i

$$read(x + y, A) = write(x, y, A)$$

$$\big(read(z, A) = write(x, y, A)\big) \wedge (z = x + y)$$

Nelson-Oppen Theory Combination

Convert F into $F_1 \wedge F_2 \wedge \ldots \wedge F_n$ such that F_i has only terms from theory T_i

$$0 \leq x \leq 1 \wedge \big(f(x) \neq f(0)\big) \wedge \big(f(x) \neq f(1)\big)$$

Nelson-Oppen Theory Combination

Convert F into $F_1 \wedge F_2 \wedge \ldots \wedge F_n$ such that F_i has only terms from theory T_i

$$0 \le x \le 1 \wedge \big(f(x) \neq f(0)\big) \wedge \big(f(x) \neq f(1)\big)$$

$$0 \le x \le 1 \wedge y = 0 \wedge z = 1 \wedge \big(f(x) \neq f(y)\big) \wedge \big(f(x) \neq f(z)\big)$$

Nelson-Oppen Theory Combination

Convert F into $F_1 \wedge F_2 \wedge \ldots \wedge F_n$ such that F_i has only terms from theory T_i

Nelson-Oppen Theory Combination

Convert F into $F_1 \wedge F_2 \wedge \ldots \wedge F_n$ such that F_i has only terms from theory T_i

Let DP_i be a decision procedure for T_i. If $DP_i(F_i)$ returns $\bot$, then F is unsatisfiable.

Nelson-Oppen Theory Combination

Convert F into $F_1 \wedge F_2 \wedge \ldots \wedge F_n$ such that F_i has only terms from theory T_i

Let DP_i be a decision procedure for T_i. If $DP_i(F_i)$ returns $\perp$, then F is unsatisfiable.

If all $DP_i(F_i)$ return $\top$, then F may still be unsatisfiable.

Nelson-Oppen Theory Combination

Nelson-Oppen Theory Combination

$$F[\sim] := \bigwedge \{t = s \mid t \sim s \text{ and } t, s \in S\} \wedge \bigwedge \{t \neq s \mid t \nsim s \text{ and } t, s \in S\}$$

Nelson-Oppen Theory Combination

Let T_1 and T_2 be two theories with disjoint signature.
Let F be a conjunction of literals for theory $C\left(T_1 \cup T_2\right)$.

1. Convert F into $F_1 \wedge F_2$.
2. Guess an equivalence relation $\sim$ over $vars\left(F_1\right) \cap vars\left(F_2\right)$.
3. Check $DP_1\left(F_1 \wedge F[\sim]\right)$.
4. Check $DP_2\left(F_2 \wedge F[\sim]\right)$.

Nelson-Oppen Theory Combination

$$0 \le x \le 1 \wedge \big(f(x) \neq f(0)\big) \wedge \big(f(x) \neq f(1)\big)$$

Nelson-Oppen Theory Combination

$$0 \leq x \leq 1 \land y = 0 \land z = 1 \land \big(f(x) \neq f(y)\big) \land \big(f(x) \neq f(z)\big)$$

Nelson-Oppen Theory Combination

$$0 \leq x \leq 1 \wedge y = 0 \wedge z = 1 \wedge \big(f(x) \neq f(y)\big) \wedge \big(f(x) \neq f(z)\big)$$

1. $x = y \quad \wedge \quad y = z \quad \wedge \quad z = x$
2. $x \neq y \quad \wedge \quad y \neq z \quad \wedge \quad z = x$
3. $x = y \quad \wedge \quad y \neq z \quad \wedge \quad z \neq x$
4. $x \neq y \quad \wedge \quad y = z \quad \wedge \quad z \neq x$
5. $x \neq y \quad \wedge \quad y \neq z \quad \wedge \quad z \neq x$

Nelson-Oppen Theory Combination

$$0 \leq x \leq 1 \land y = 0 \land z = 1 \land \big(f(x) \neq f(y)\big) \land \big(f(x) \neq f(z)\big)$$

1. $x = y \quad \land \quad y = z \quad \land \quad z = x$ ✗
2. $x \neq y \quad \land \quad y \neq z \quad \land \quad z = x$
3. $x = y \quad \land \quad y \neq z \quad \land \quad z \neq x$
4. $x \neq y \quad \land \quad y = z \quad \land \quad z \neq x$
5. $x \neq y \quad \land \quad y \neq z \quad \land \quad z \neq x$

Nelson-Oppen Theory Combination

$$0 \leq x \leq 1 \land y = 0 \land z = 1 \land \big(f(x) \neq f(y)\big) \land \big(f(x) \neq f(z)\big)$$

1. $x = y \quad \land \quad y = z \quad \land \quad z = x$ ✗
2. $x \neq y \quad \land \quad y \neq z \quad \land \quad z = x$ ✗
3. $x = y \quad \land \quad y \neq z \quad \land \quad z \neq x$
4. $x \neq y \quad \land \quad y = z \quad \land \quad z \neq x$
5. $x \neq y \quad \land \quad y \neq z \quad \land \quad z \neq x$

Nelson-Oppen Theory Combination

$$0 \leq x \leq 1 \land y = 0 \land z = 1 \land \big(f(x) \neq f(y)\big) \land \big(f(x) \neq f(z)\big)$$

1. $x = y \quad \land \quad y = z \quad \land \quad z = x$ ✗
2. $x \neq y \quad \land \quad y \neq z \quad \land \quad z = x$ ✗
3. $x = y \quad \land \quad y \neq z \quad \land \quad z \neq x$ ✗
4. $x \neq y \quad \land \quad y = z \quad \land \quad z \neq x$
5. $x \neq y \quad \land \quad y \neq z \quad \land \quad z \neq x$

Nelson-Oppen Theory Combination

$$0 \leq x \leq 1 \land y = 0 \land z = 1 \land \big(f(x) \neq f(y)\big) \land \big(f(x) \neq f(z)\big)$$

1. $x = y \quad \land \quad y = z \quad \land \quad z = x$ ✗
2. $x \neq y \quad \land \quad y \neq z \quad \land \quad z = x$ ✗
3. $x = y \quad \land \quad y \neq z \quad \land \quad z \neq x$ ✗
4. $x \neq y \quad \land \quad y = z \quad \land \quad z \neq x$ ✗
5. $x \neq y \quad \land \quad y \neq z \quad \land \quad z \neq x$

Nelson-Oppen Theory Combination

$$0 \leq x \leq 1 \land y = 0 \land z = 1 \land \big(f(x) \neq f(y)\big) \land \big(f(x) \neq f(z)\big)$$

1. $x = y \quad \land \quad y = z \quad \land \quad z = x$ ✗
2. $x \neq y \quad \land \quad y \neq z \quad \land \quad z = x$ ✗
3. $x = y \quad \land \quad y \neq z \quad \land \quad z \neq x$ ✗
4. $x \neq y \quad \land \quad y = z \quad \land \quad z \neq x$ ✗
5. $x \neq y \quad \land \quad y \neq z \quad \land \quad z \neq x$?

Nelson-Oppen Theory Combination

Let T_1 and T_2 be two theories with disjoint signature.
Let F be a conjunction of literals for theory $C\left(T_1 \cup T_2\right)$.

1. Convert F into $F_1 \wedge F_2$.
2. Guess an equivalence relation $\sim$ over $vars\left(F_1\right) \cap vars\left(F_2\right)$.
3. Check $DP_1\left(F_1 \wedge F[\sim]\right)$.
4. Check $DP_2\left(F_2 \wedge F[\sim]\right)$.

Time to Code!


Pre-doctoral Research Workshop
January 9-11, 2025