COL:750

Foundations of Automatic Verification

Instructor: Priyanka Golia

Course Webpage

https://priyanka-golia.github.io/teaching/COL-750/index.html

Boolean /propositional formulas ——> SAT Solvers

If formula is SATisfiable, gives an satisfying assignment

UNSAT

DP algorithm for SAT Solving (Martin Davis - Hilary Putnam 1960)

1. Start with F_{CNF}
2. For every clause C in F_{CNF} that either contains both l and $\neg l$ or has pure literal do:
   1. $F_{CNF} \leftarrow remove\_from\_formula(C, F_{CNF})$
3. If F_{CNF} is empty
   1. Return SAT
4. If F_{CNF} has empty clause then
   1. Return UNSAT
5. Pick a literal l that occurs with both polarities in F_{CNF}.
   1. $F_{CNF} \leftarrow Resolution(C, l, F_{CNF})$
6. For every clause C that contains l or $\neg l$ do :
   1. $F_{CNF} \leftarrow remove\_from\_formula(C, F_{CNF})$

DP algorithm

$$F = (p \lor q) \land (p \lor \neg q) \land (\neg p \lor r) \land (\neg p \lor \neg r)$$

DP algorithm

$$F = (p \lor q) \land (p \lor \neg q) \land (\neg p \lor r) \land (\neg p \lor \neg r)$$

* No pure literal, no clause with $l \lor \neg l$

DP algorithm

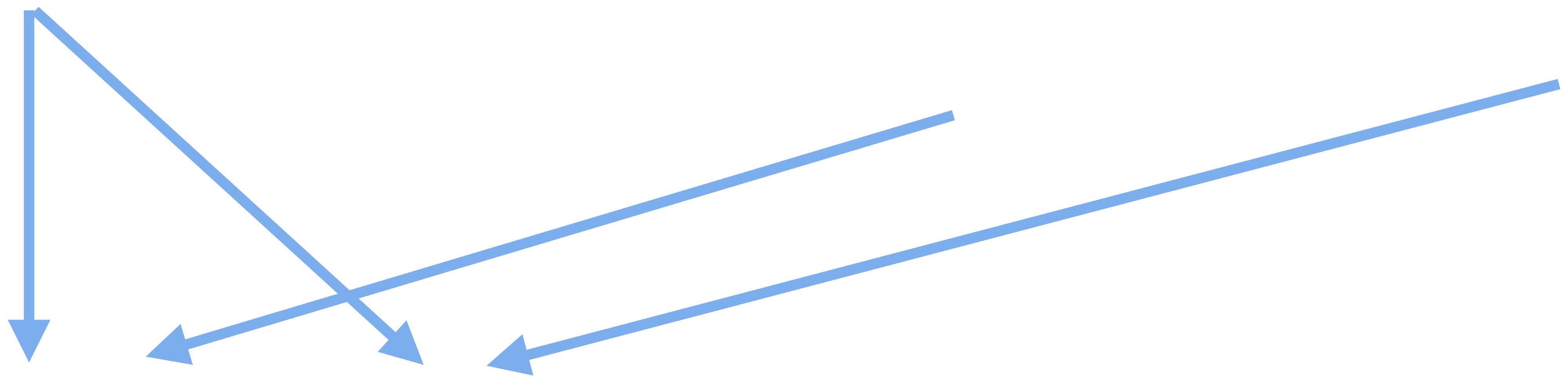
$$F = (p \lor q) \land (p \lor \neg q) \land (\neg p \lor r) \land (\neg p \lor \neg r)$$

* No pure literal, no clause with $l \lor \neg l$

Pick literal p

DP algorithm

$$F = (p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee r) \wedge (\neg p \vee \neg r)$$

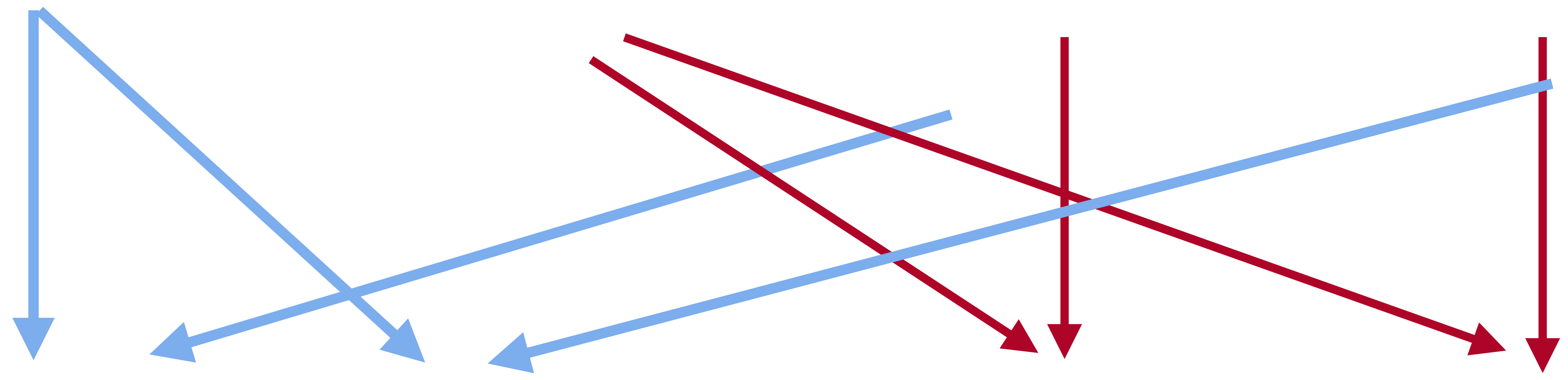


* No pure literal, no clause with $l \vee \neg l$

Pick literal p

DP algorithm

$$F = (p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee r) \wedge (\neg p \vee \neg r)$$

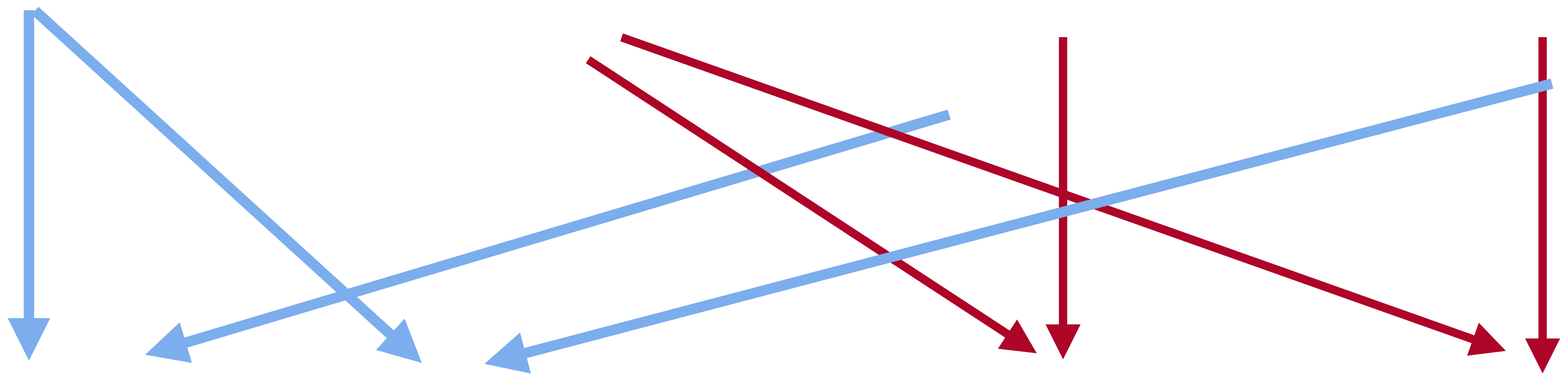


* No pure literal, no clause with $l \vee \neg l$

Pick literal p

DP algorithm

$$F = (p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee r) \wedge (\neg p \vee \neg r)$$



$$(q \vee r) \;\wedge\; (q \vee \neg r) \;\wedge\; (\neg q \vee r) \;\wedge\; (\neg q \vee \neg r)$$

* No pure literal, no clause with $l \vee \neg l$

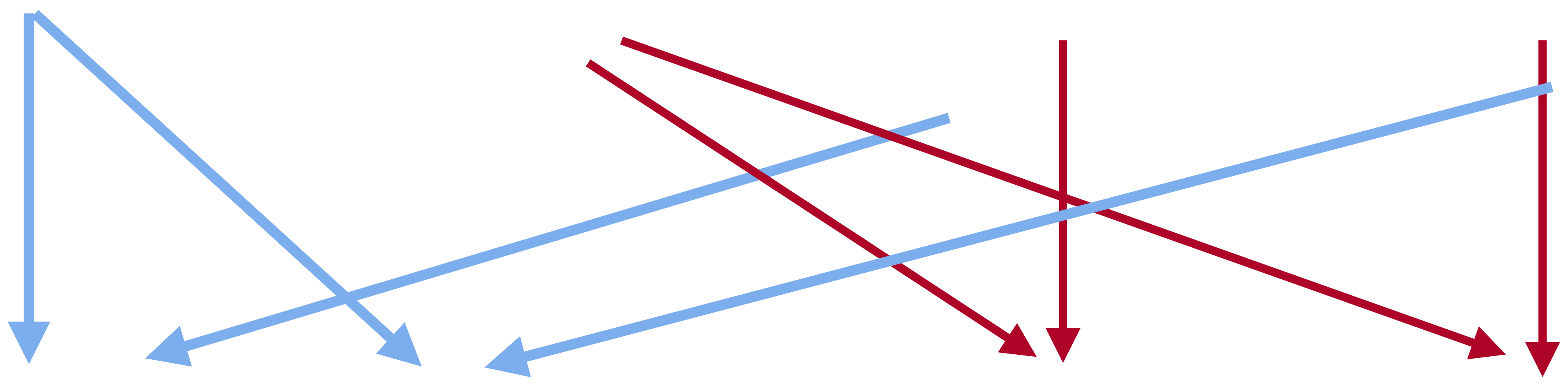
Pick literal p

DP algorithm

$$F = (p \lor q) \land (p \lor \neg q) \land (\neg p \lor r) \land (\neg p \lor \neg r)$$

* No pure literal, no clause with $l \lor \neg l$

Pick literal p



$$(q \lor r) \ \land \ (q \lor \neg r) \ \land \ (\neg q \lor r) \ \land \ (\neg q \lor \neg r)$$

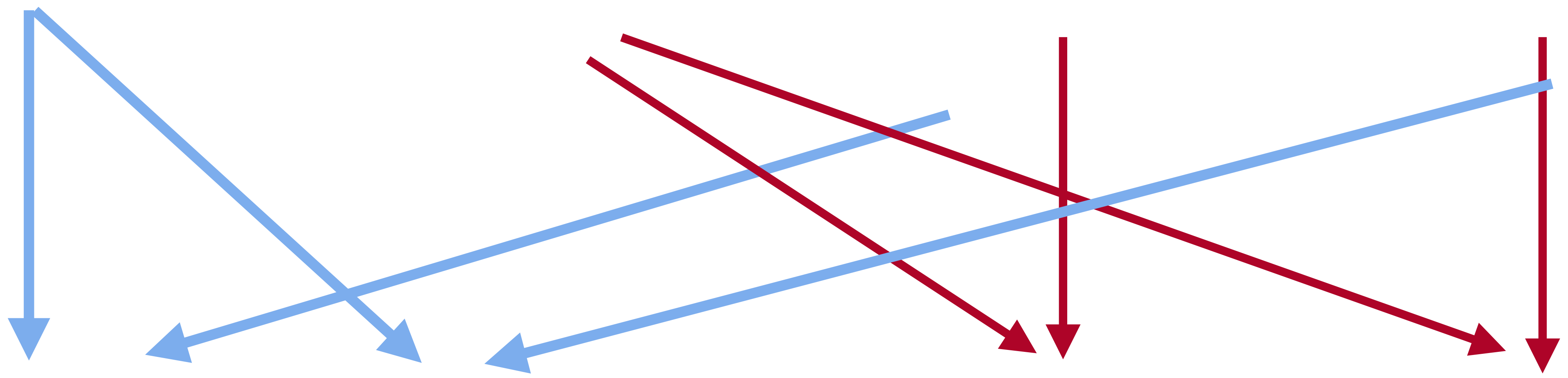
* No pure literal, no clause with $l \lor \neg l$

DP algorithm

$$F = (p \lor q) \land (p \lor \neg q) \land (\neg p \lor r) \land (\neg p \lor \neg r)$$

* No pure literal, no clause with $l \lor \neg l$

  Pick literal p



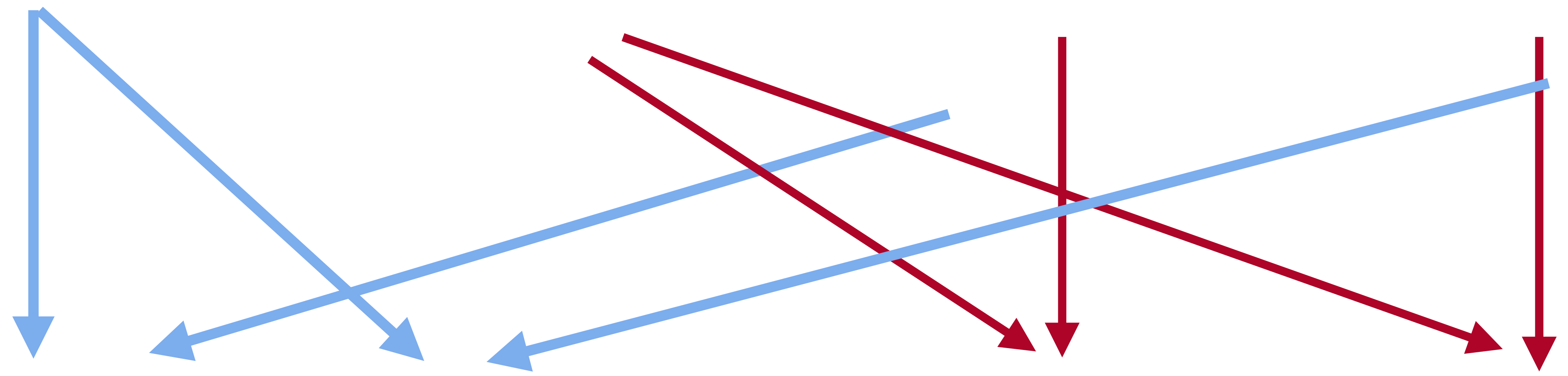
$$(q \lor r) \;\land\; (q \lor \neg r) \;\land\; (\neg q \lor r) \;\land\; (\neg q \lor \neg r)$$

* No pure literal, no clause with $l \lor \neg l$

  Pick literal q

DP algorithm

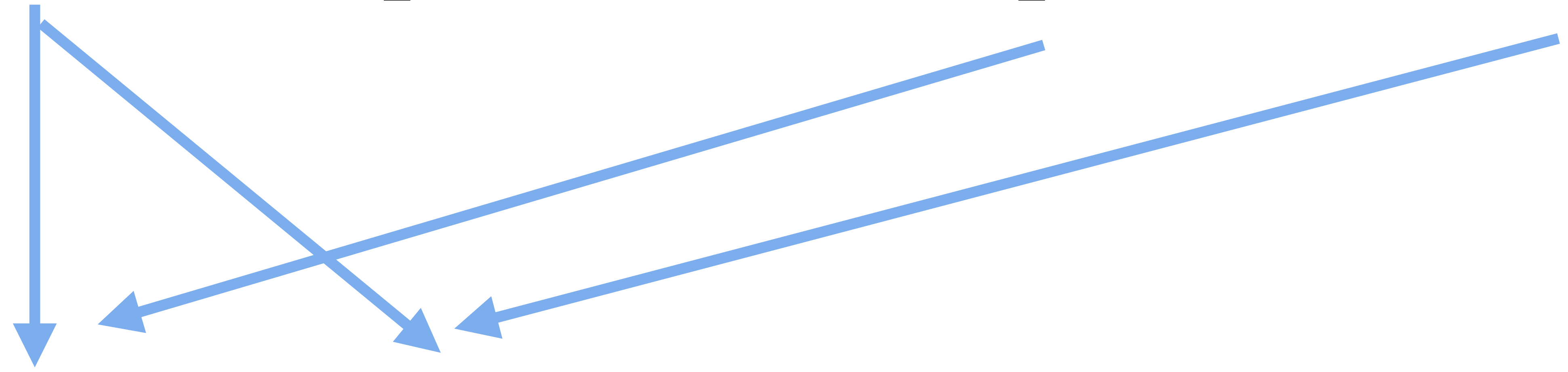
$$F = (p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee r) \wedge (\neg p \vee \neg r)$$



* No pure literal, no clause with $l \vee \neg l$

Pick literal p

$$(q \vee r) \;\wedge\; (q \vee \neg r) \;\wedge\; (\neg q \vee r) \;\wedge\; (\neg q \vee \neg r)$$

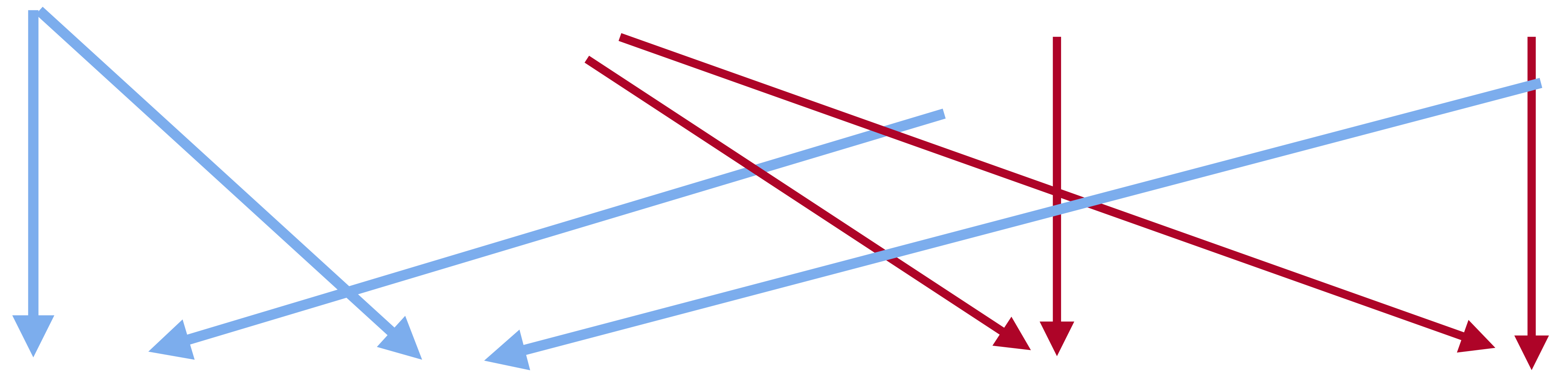


* No pure literal, no clause with $l \vee \neg l$

Pick literal q

DP algorithm

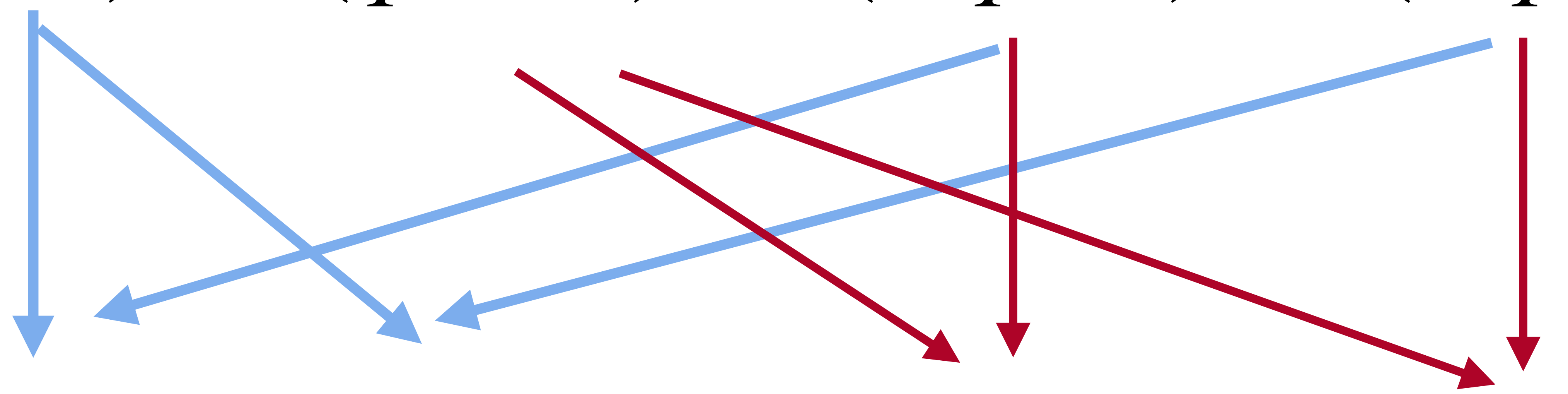
$$F = (p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee r) \wedge (\neg p \vee \neg r)$$



* No pure literal, no clause with $l \vee \neg l$

  Pick literal p

$$(q \vee r) \;\wedge\; (q \vee \neg r) \;\wedge\; (\neg q \vee r) \;\wedge\; (\neg q \vee \neg r)$$

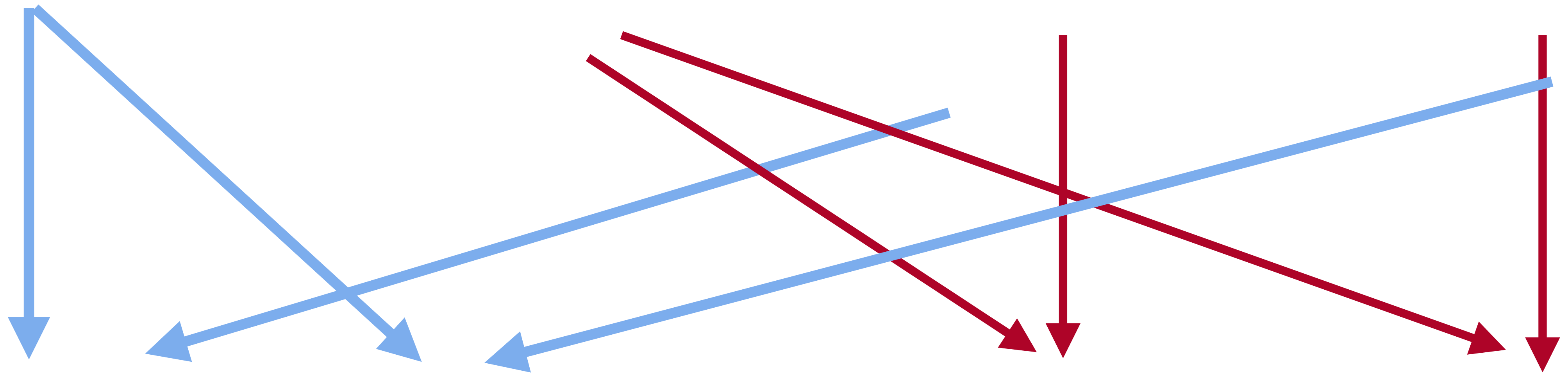


* No pure literal, no clause with $l \vee \neg l$

  Pick literal q

DP algorithm

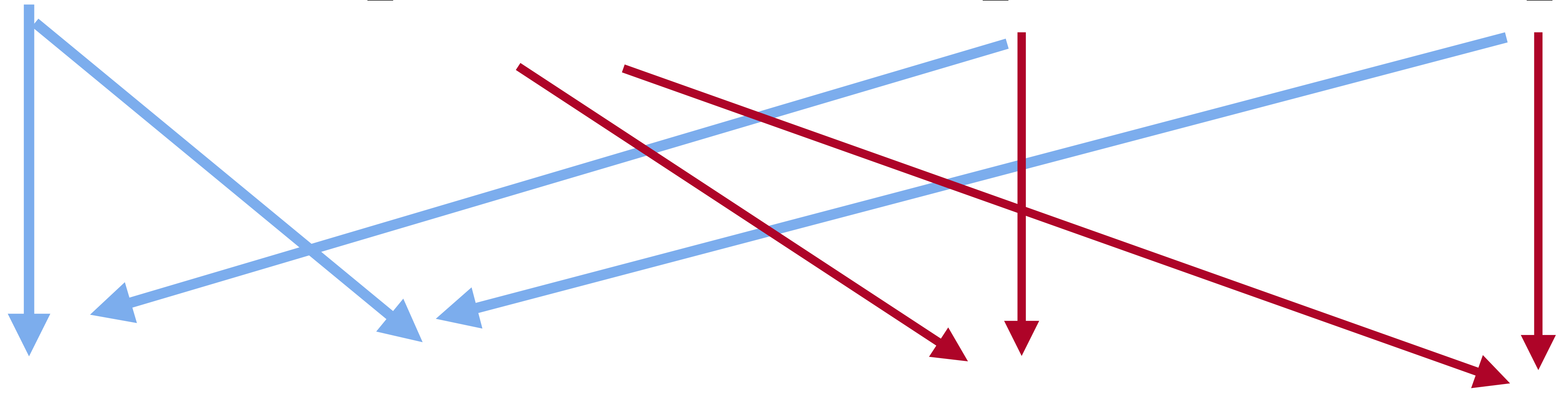
$$F = (p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee r) \wedge (\neg p \vee \neg r)$$



* No pure literal, no clause with $l \vee \neg l$

Pick literal p

$$(q \vee r) \;\wedge\; (q \vee \neg r) \;\wedge\; (\neg q \vee r) \;\wedge\; (\neg q \vee \neg r)$$



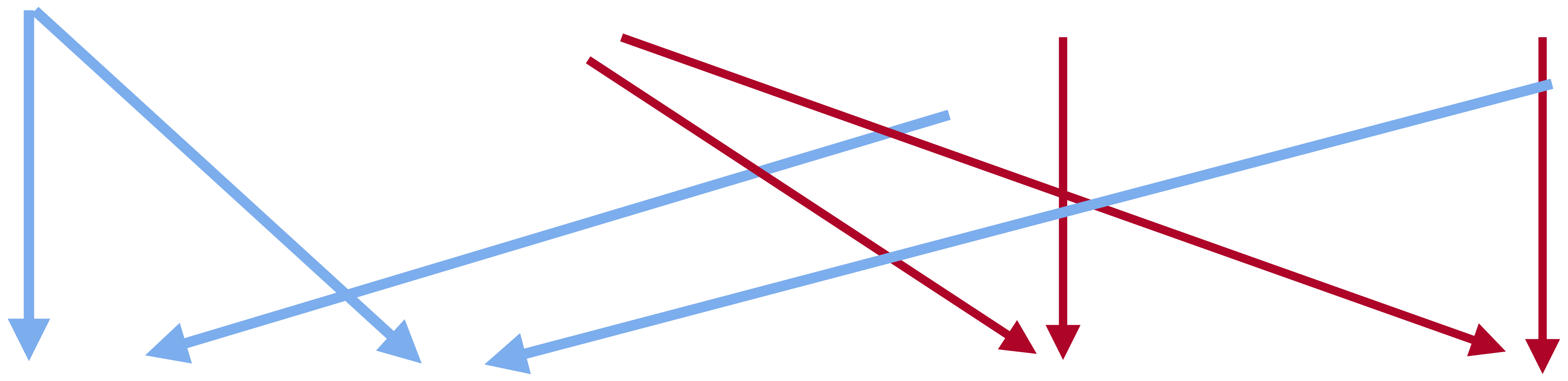
* No pure literal, no clause with $l \vee \neg l$

Pick literal q

$$(r) \;\wedge\; (r \vee \neg r) \;\wedge\; (\neg r \vee r) \;\wedge\; (\neg r)$$

DP algorithm

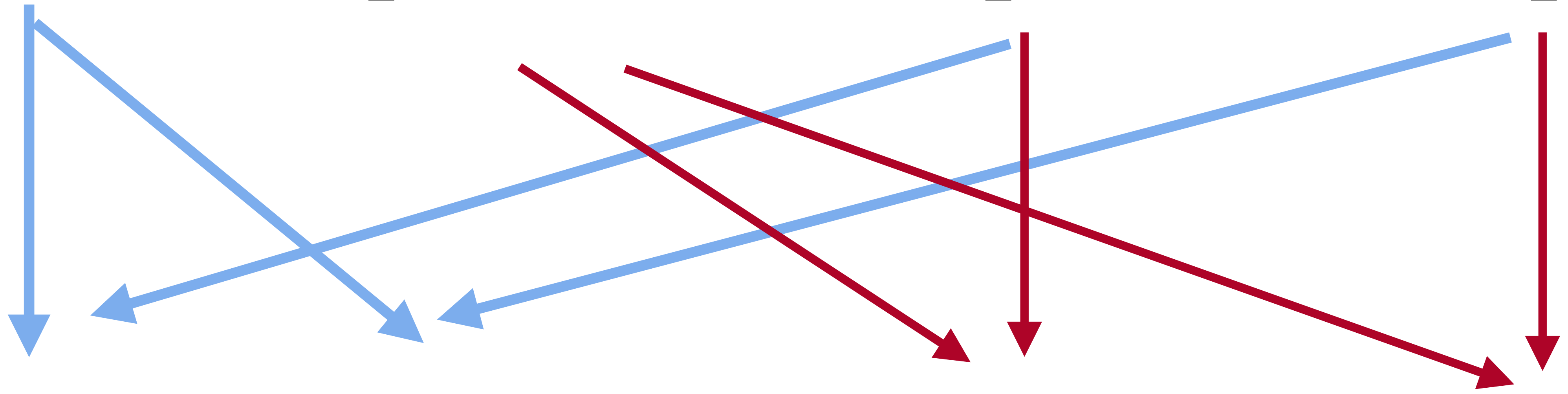
$$F = (p \lor q) \land (p \lor \neg q) \land (\neg p \lor r) \land (\neg p \lor \neg r)$$



* No pure literal, no clause with $l \lor \neg l$

Pick literal p

$$(q \lor r) \ \land \ (q \lor \neg r) \ \land \ (\neg q \lor r) \ \land \ (\neg q \lor \neg r)$$



* No pure literal, no clause with $l \lor \neg l$

Pick literal q

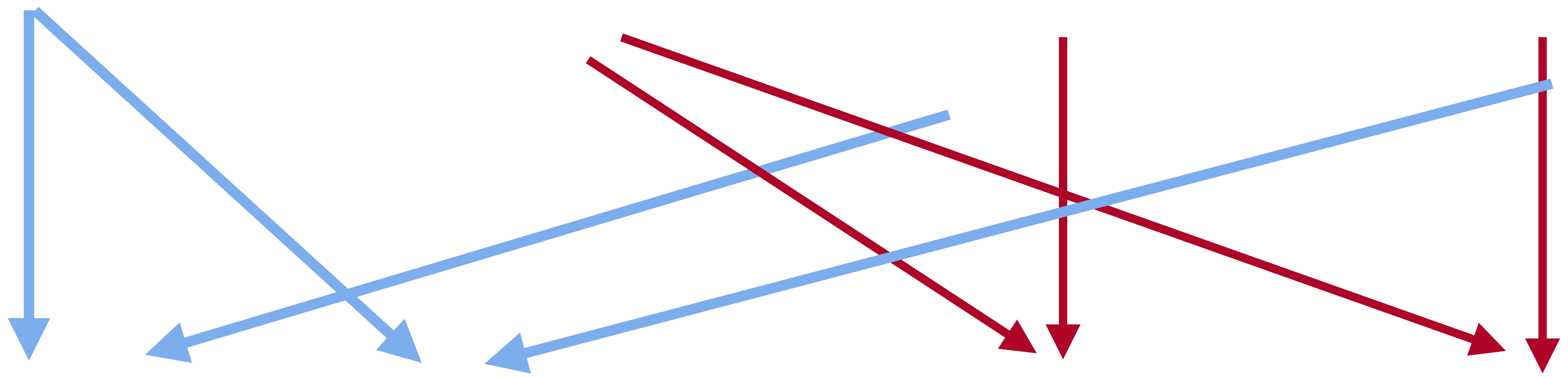
$$(r) \ \land \ (r \lor \neg r) \ \land \ (\neg r \lor r) \ \land \ (\neg r)$$

*remove clauses with $l \lor \neg l$

$$(r) \land (\neg r)$$

DP algorithm

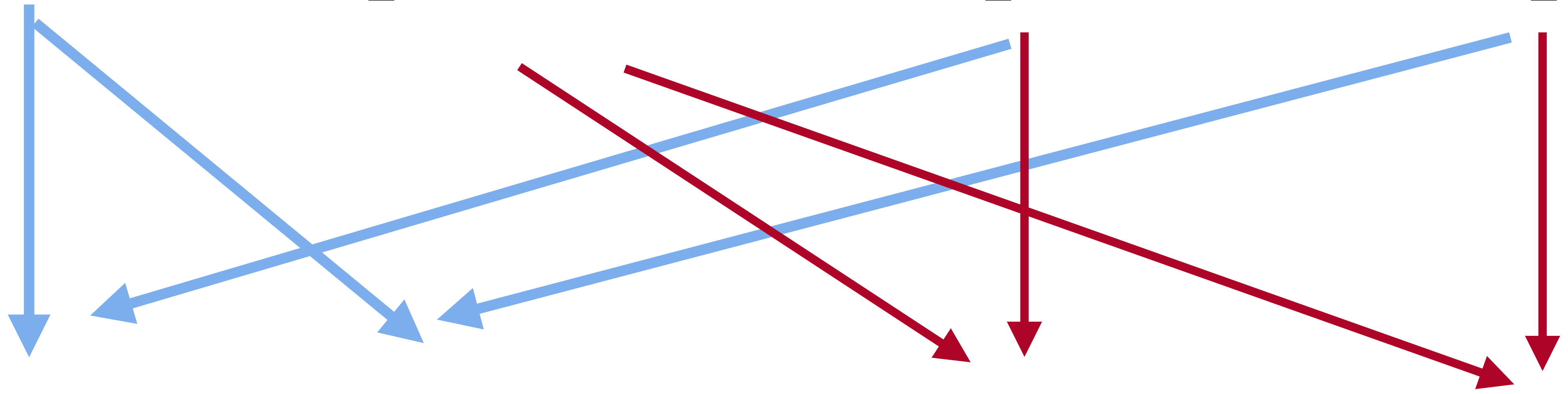
$$F = (p \lor q) \land (p \lor \neg q) \land (\neg p \lor r) \land (\neg p \lor \neg r)$$



* No pure literal, no clause with $l \lor \neg l$

Pick literal p

$$(q \lor r) \;\land\; (q \lor \neg r) \;\land\; (\neg q \lor r) \;\land\; (\neg q \lor \neg r)$$



* No pure literal, no clause with $l \lor \neg l$

Pick literal q

$$(r) \;\land\; (r \lor \neg r) \;\land\; (\neg r \lor r) \;\land\; (\neg r)$$

*remove clauses with $l \lor \neg l$

$$(r) \land (\neg r)$$

Pick literal r

DP algorithm

$$F = (p \lor q) \land (p \lor \neg q) \land (\neg p \lor r) \land (\neg p \lor \neg r)$$

* No pure literal, no clause with $l \lor \neg l$

Pick literal p

$$(q \lor r) \;\land\; (q \lor \neg r) \;\land\; (\neg q \lor r) \;\land\; (\neg q \lor \neg r)$$

* No pure literal, no clause with $l \lor \neg l$

Pick literal q

$$(r) \;\land\; (r \lor \neg r) \;\land\; (\neg r \lor r) \;\land\; (\neg r)$$

*remove clauses with $l \lor \neg l$

$$(r) \land (\neg r)$$

Pick literal r

$$\square$$

DP algorithm

$$F = (p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee r) \wedge (\neg p \vee \neg r)$$

* No pure literal, no clause with $l \vee \neg l$

Pick literal p

$$(q \vee r) \;\wedge\; (q \vee \neg r) \;\wedge\; (\neg q \vee r) \;\wedge\; (\neg q \vee \neg r)$$

* No pure literal, no clause with $l \vee \neg l$

Pick literal q

$$(r) \;\wedge\; (r \vee \neg r) \;\wedge\; (\neg r \vee r) \;\wedge\; (\neg r)$$

*remove clauses with $l \vee \neg l$

$$(r) \wedge (\neg r)$$

Pick literal r

$$\square$$

F has empty clause — UNSAT

DP algorithm

$$F = (p \vee q \vee r) \wedge (q \vee \neg r \vee \neg s) \wedge (\neg q \vee s) \wedge (\neg p \vee \neg s)$$

DP algorithm

$$F = (p \lor q \lor r) \land (q \lor \neg r \lor \neg s) \land (\neg q \lor s) \land (\neg p \lor \neg s)$$

$$(q \lor r \lor \neg s) \land (q \lor \neg r \lor \neg s) \land (\neg q \lor s)$$

$$(r \lor \neg s \lor s) \land (\neg r \lor \neg s \lor s)$$

$$\{\}$$

Empty Formula, return SAT

DP algorithm

$$F = (p \lor q \lor r) \land (q \lor \neg r \lor \neg s) \land (\neg q \lor s) \land (\neg p \lor \neg s)$$

* No pure literal, no clause with $l \lor \neg l$

$$(q \lor r \lor \neg s) \land (q \lor \neg r \lor \neg s) \land (\neg q \lor s)$$

$$(r \lor \neg s \lor s) \land (\neg r \lor \neg s \lor s)$$

$$\{\}$$

Empty Formula, return SAT

DP algorithm

$$F = (p \vee q \vee r) \wedge (q \vee \neg r \vee \neg s) \wedge (\neg q \vee s) \wedge (\neg p \vee \neg s)$$

* No pure literal, no clause with $l \vee \neg l$

Pick literal p

$$(q \vee r \vee \neg s) \wedge (q \vee \neg r \vee \neg s) \wedge (\neg q \vee s)$$

$$(r \vee \neg s \vee s) \wedge (\neg r \vee \neg s \vee s)$$

$$\{\}$$

Empty Formula, return SAT

DP algorithm

$$F = (p \vee q \vee r) \wedge (q \vee \neg r \vee \neg s) \wedge (\neg q \vee s) \wedge (\neg p \vee \neg s)$$

* No pure literal, no clause with $l \vee \neg l$
  Pick literal p

$$(q \vee r \vee \neg s) \wedge (q \vee \neg r \vee \neg s) \wedge (\neg q \vee s)$$

* No pure literal, no clause with $l \vee \neg l$

$$(r \vee \neg s \vee s) \wedge (\neg r \vee \neg s \vee s)$$

$$\{\}$$

Empty Formula, return SAT

DP algorithm

$$F = (p \vee q \vee r) \wedge (q \vee \neg r \vee \neg s) \wedge (\neg q \vee s) \wedge (\neg p \vee \neg s)$$

* No pure literal, no clause with $l \vee \neg l$

Pick literal p

$$(q \vee r \vee \neg s) \wedge (q \vee \neg r \vee \neg s) \wedge (\neg q \vee s)$$

* No pure literal, no clause with $l \vee \neg l$

Pick literal q

$$(r \vee \neg s \vee s) \wedge (\neg r \vee \neg s \vee s)$$

$$\{\}$$

Empty Formula, return SAT

DP algorithm

$$F = (p \vee q \vee r) \wedge (q \vee \neg r \vee \neg s) \wedge (\neg q \vee s) \wedge (\neg p \vee \neg s)$$

* No pure literal, no clause with $l \vee \neg l$

  Pick literal p

$$(q \vee r \vee \neg s) \wedge (q \vee \neg r \vee \neg s) \wedge (\neg q \vee s)$$

* No pure literal, no clause with $l \vee \neg l$

  Pick literal q

$$(r \vee \neg s \vee s) \wedge (\neg r \vee \neg s \vee s)$$

* remove clauses with $l \vee \neg l$

$$\{\}$$

Empty Formula, return SAT

$$F = (p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee r) \wedge (\neg p \vee \neg r)$$

$$F = (p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee r) \wedge (\neg p \vee \neg r) \;\; \wedge \neg p$$

$$F = (p \lor q) \land (p \lor \neg q) \land (\neg p \lor r) \land (\neg p \lor \neg r) \land \neg p$$

Unit clause

$$F = (p \lor q) \land (p \lor \neg q) \land \underline{(\neg p \lor r) \land (\neg p \lor \neg r) \land \neg p}$$

Unit clause

$$F = (p \lor q) \land (p \lor \neg q) \land (\neg p \lor r) \land (\neg p \lor \neg r) \land \neg p$$

Unit clause

p has to take value 0, $(\neg p \lor r) \land (\neg p \lor \neg r)$ are True

$$F = (p \lor q) \land (p \lor \neg q) \land (\neg p \lor r) \land (\neg p \lor \neg r) \land \neg p$$ Unit clause

p has to take value 0, $(\neg p \lor r) \land (\neg p \lor \neg r)$ are True

Can we remove all clauses that have $\neg p$?

$$F = (p \lor q) \land (p \lor \neg q) \land (\neg p \lor r) \land (\neg p \lor \neg r) \land \neg p$$ Unit clause

p has to take value 0, $(\neg p \lor r) \land (\neg p \lor \neg r)$ are True

Can we remove all clauses that have $\neg p$?

$$F = (p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee r) \wedge (\neg p \vee \neg r) \wedge \neg p$$

Unit clause

p has to take value 0, $(\neg p \vee r) \wedge (\neg p \vee \neg r)$ are True

Can we remove all clauses that have $\neg p$?

$$F = (p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee r) \wedge (\neg p \vee \neg r) \wedge \neg p$$

Unit clause

p has to take value 0, $(\neg p \vee r) \wedge (\neg p \vee \neg r)$ are True

Can we remove all clauses that have $\neg p$?

$$(\neg p) \wedge (p \vee q) \equiv_{SAT} q$$

$$F = (p \lor q) \land (p \lor \neg q) \land (\neg p \lor r) \land (\neg p \lor \neg r) \land \neg p$$

Unit clause

p has to take value 0, $(\neg p \lor r) \land (\neg p \lor \neg r)$ are True

Can we remove all clauses that have $\neg p$?

$$(\neg p) \land (p \lor q) \equiv_{SAT} q$$

$$(\neg p) \land (p \lor \neg q) \equiv_{SAT} \neg q$$

Unit Propagation

While F contains a unit clause (*l*) do:

For every clause C in F that has *l* do:

$$F_{CNF} \leftarrow remove\_from\_formula(C, F_{CNF})$$

For every clause C in F that has $\neg l$ do:

$$F_{CNF} \leftarrow remove\_from\_formula(C, F_{CNF})$$

$$F_{CNF} \leftarrow add\_to\_formula(C \setminus \neg l, F_{CNF})$$

DP algorithm for SAT Solving (Martin Davis - Hilary Putnam 1960)

1. Start with F_{CNF}
2. For every clause C in F_{CNF} that either contains both l and $\neg l$ or has pure literal do:
   1. $F_{CNF} \leftarrow remove\_from\_formula(C, F_{CNF})$
3. $F_{CNF} \leftarrow$ UnitPropagation(F_{CNF})
4. If F_{CNF} is empty
   1. Return SAT
5. If F_{CNF} has empty clause then
   1. Return UNSAT
6. Pick a literal l that occurs with both polarities in F_{CNF}.
   1. $F_{CNF} \leftarrow Resolution(C, l, F_{CNF})$
7. For every clause C that contains l or $\neg l$ do :
   1. $F_{CNF} \leftarrow remove\_from\_formula(C, F_{CNF})$

DPLL algorithm (Davis -Putnam-Logemann-Loveland 1960)

Complete and Sound algorithm & takes linear space in worst case.

Still the basis of SAT solver

zChaff Solver — efficient implementation of DPLL.

Won test of time award at CAV 2001.

Notations

Partial Model: subset of elements of Vars(F) maps to $\{0,1\}$

Under partial model m,

A literal l is True if $m(l)$= 1

A literal l is False if $m(l)$ = 0

Otherwise:

l is unassigned.

Example: $F = (x_1 \vee x_2 \vee \neg x_3); m = \{x_1 \mapsto 1, x_3 \mapsto 1\}$

x_1 is True, x_2 is unassigned, x_3 is False.

Notations

Partial Model: subset of elements of Vars(F) maps to {0,1}

Under partial model m,

Clause C is True if there is a $l \in C$, such that l is True.

Clause C is False if for each literal $l \in C$, l is False

Otherwise:

C is unassigned.

Example: $m = \{x_1 \mapsto 1, x_3 \mapsto 1\}$

Notations

Partial Model: subset of elements of Vars(F) maps to $\{0,1\}$

Under partial model m,

Clause C is True if there is a $l \in C$, such that l is True.

Clause C is False if for each literal $l \in C$, l is False

Otherwise:

C is unassigned.

$C = (x_1 \vee x_2 \vee x_3)$ — True

Example: $m = \{x_1 \mapsto 1, x_3 \mapsto 1\}$

Notations

Partial Model: subset of elements of Vars(F) maps to $\{0,1\}$

Under partial model m,

Clause C is True if there is a $l \in C$, such that l is True.

Clause C is False if for each literal $l \in C$, l is False

Otherwise:

C is unassigned.

Example: $m = \{x_1 \mapsto 1, x_3 \mapsto 1\}$

$C = (x_1 \vee x_2 \vee x_3)$ — True

$C = (\neg x_1 \vee x_2 \vee \neg x_3)$ — Unassigned

Notations

Partial Model: subset of elements of Vars(F) maps to $\{0,1\}$

Under partial model m,

Clause C is True if there is a $l \in C$, such that l is True.

Clause C is False if for each literal $l \in C$, l is False

Otherwise:

C is unassigned.

Example: $m = \{x_1 \mapsto 1, x_3 \mapsto 1\}$

$C = (x_1 \vee x_2 \vee x_3)$ — True

$C = (\neg x_1 \vee x_2 \vee \neg x_3)$ — Unassigned

$C = (\neg x_1 \vee \neg x_3)$ — False

Notations

Partial Model: subset of elements of Vars(F) maps to $\{0,1\}$

Under partial model m,

F_{CNF} is True if for each $C \in F_{CNF}$, C is True.

F_{CNF} is False if there is a $C \in F_{CNF}$ such that C is False

Otherwise:

F_{CNF} is unassigned.

Notations

Partial Model: subset of elements of Vars(F) maps to $\{0,1\}$

Under partial model m,

F_{CNF} is True if for each $C \in F_{CNF}$, C is True.

F_{CNF} is False if there is a $C \in F_{CNF}$ such that C is False

Otherwise:

F_{CNF} is unassigned.

Unit Clause (updated): C is a unit clause under partial model m if there is exactly one literal l in C which is unassigned, and rest all literals of C are False.

Notations

Partial Model: subset of elements of Vars(F) maps to $\{0,1\}$

Under partial model m,

F_{CNF} is True if for each $C \in F_{CNF}$, C is True.

F_{CNF} is False if there is a $C \in F_{CNF}$ such that C is False

Otherwise:

F_{CNF} is unassigned.

Unit Clause (updated): C is a unit clause under partial model m if there is exactly one literal l in C which is unassigned, and rest all literals of C are False.

Example: $C = (x_1 \vee \neg x_3 \vee \neg x_2); m = \{x_1 \mapsto 0, x_2 \mapsto 1\}$

Notations

Partial Model: subset of elements of Vars(F) maps to $\{0,1\}$

Under partial model m,

F_{CNF} is True if for each $C \in F_{CNF}$, C is True.

F_{CNF} is False if there is a $C \in F_{CNF}$ such that C is False

Otherwise:

F_{CNF} is unassigned.

Unit Clause (updated): C is a unit clause under partial model m if there is exactly one literal l in C which is unassigned, and rest all literals of C are False.

Example: $C = (x_1 \vee \neg x_3 \vee \neg x_2); m = \{x_1 \mapsto 0, x_2 \mapsto 1\}$ C is unit clause under m.

DPLL algorithm (Davis -Putnam-Logemann-Loveland 1960)

1. Maintains a partial model, initially $\varnothing$
2. Assign unassigned variables either 0 or 1
    1. (Randomly one after the other)
3. Sometime forced to make a decision due to unit clause

DPLL algorithm (Davis -Putnam-Logemann-Loveland 1960)

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

Initially m is $\emptyset$

Pick a variable, say x_3, and assign it a Boolean value, say 1. Partial model $m = \{x_3 \mapsto 1\}$

$C_1 : (x_1 \vee \neg x_2)$ unassigned.

$C_2 : (\neg x_1 \vee x_2 \vee \neg x_3)$ unassigned.

Pick another variable, say x_1, and assign it a Boolean value, say 0.

Partial model $m = \{x_1 \mapsto 0, x_3 \mapsto 1\}$

$C_1 : (x_1 \vee \neg x_2)$ Unit clause, forced decision $(x_2 \mapsto 0)$

$C_2 : (\neg x_1 \vee x_2 \vee \neg x_3)$ True.

$m = \{x_1 \mapsto 0, x_2 \mapsto 0, x_3 \mapsto 1\}$ and $m \models F$

DPLL algorithm (Davis -Putnam-Logemann-Loveland 1960)

1. Maintains a partial model, initially $\emptyset$
2. Assign unassigned variables either 0 or 1
    1. (Randomly one after the other)
3. Sometime forced to make a decision due to unit clause

What to do if F is False under partial model m?

Backtracking

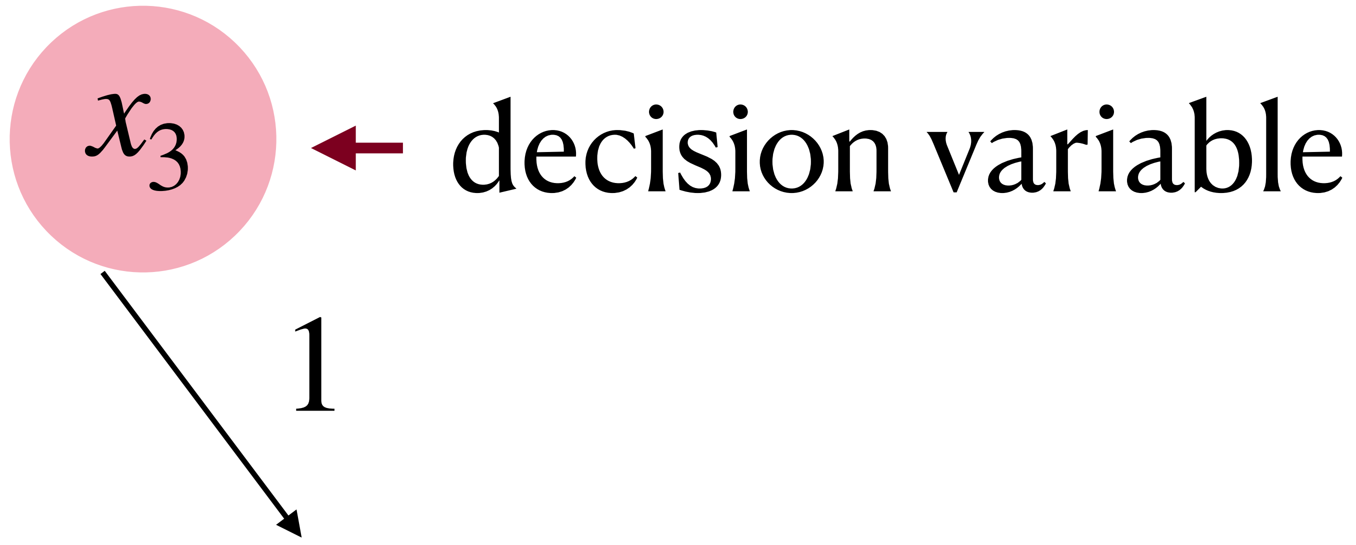
$$F = (\neg x_1 \lor \neg x_2 \lor \neg x_3) \land (\neg x_3 \lor x_2) \land (\neg x_3 \lor x_1)$$

Backtracking

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

Pick a variable, say x_3, and assign it a Boolean value, say 1.

Partial model $m = \{x_3 \mapsto 1\}$



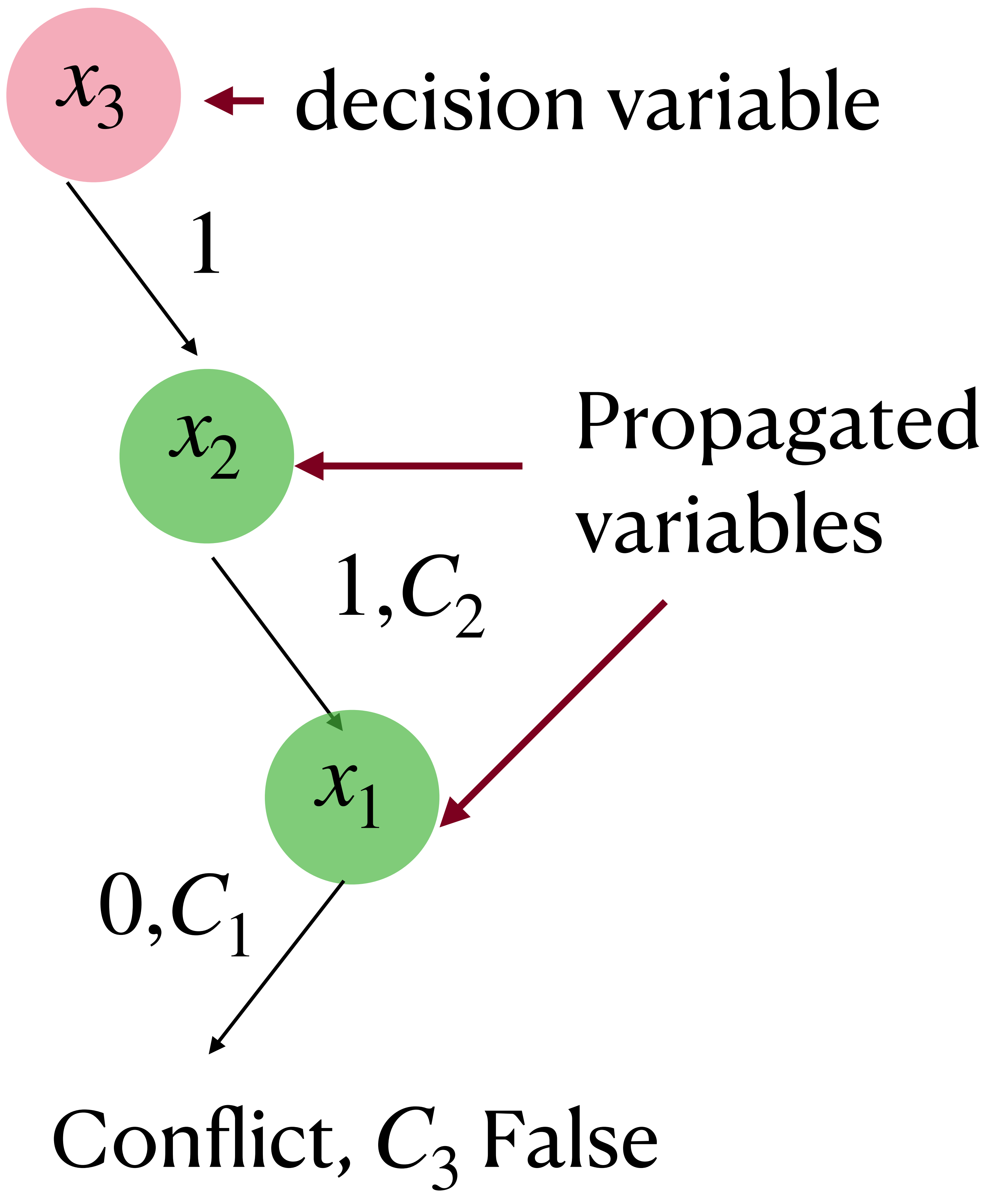
Backtracking

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

Pick a variable, say x_3, and assign it a Boolean value, say 1.

Partial model $m = \{x_3 \mapsto 1\}$

$(\neg x_3 \vee x_2) \wedge (\neg x_3 \vee x_1)$ — unit clauses



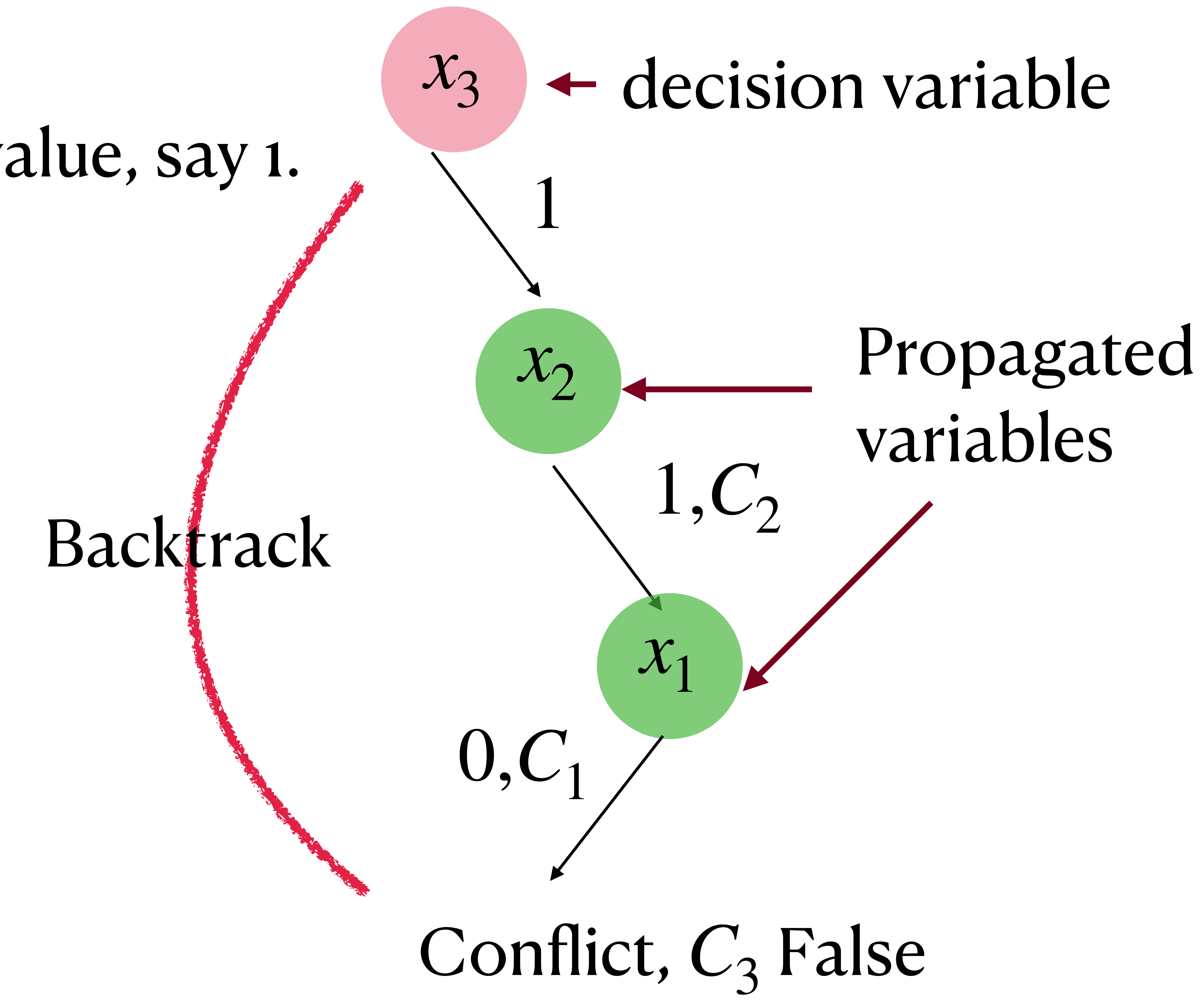
Backtracking

$F = (\neg x_1 \lor \neg x_2 \lor \neg x_3) \land (\neg x_3 \lor x_2) \land (\neg x_3 \lor x_1)$

Pick a variable, say x_3, and assign it a Boolean value, say 1.

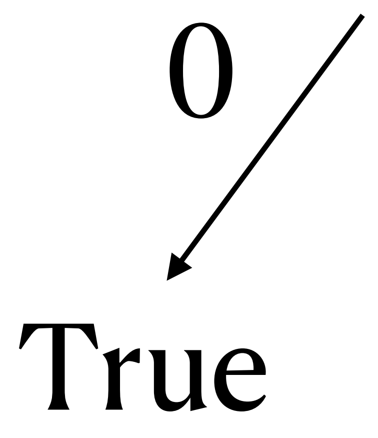
Partial model $m = \{x_3 \mapsto 1\}$

$(\neg x_3 \lor x_2) \land (\neg x_3 \lor x_1)$ — unit clauses



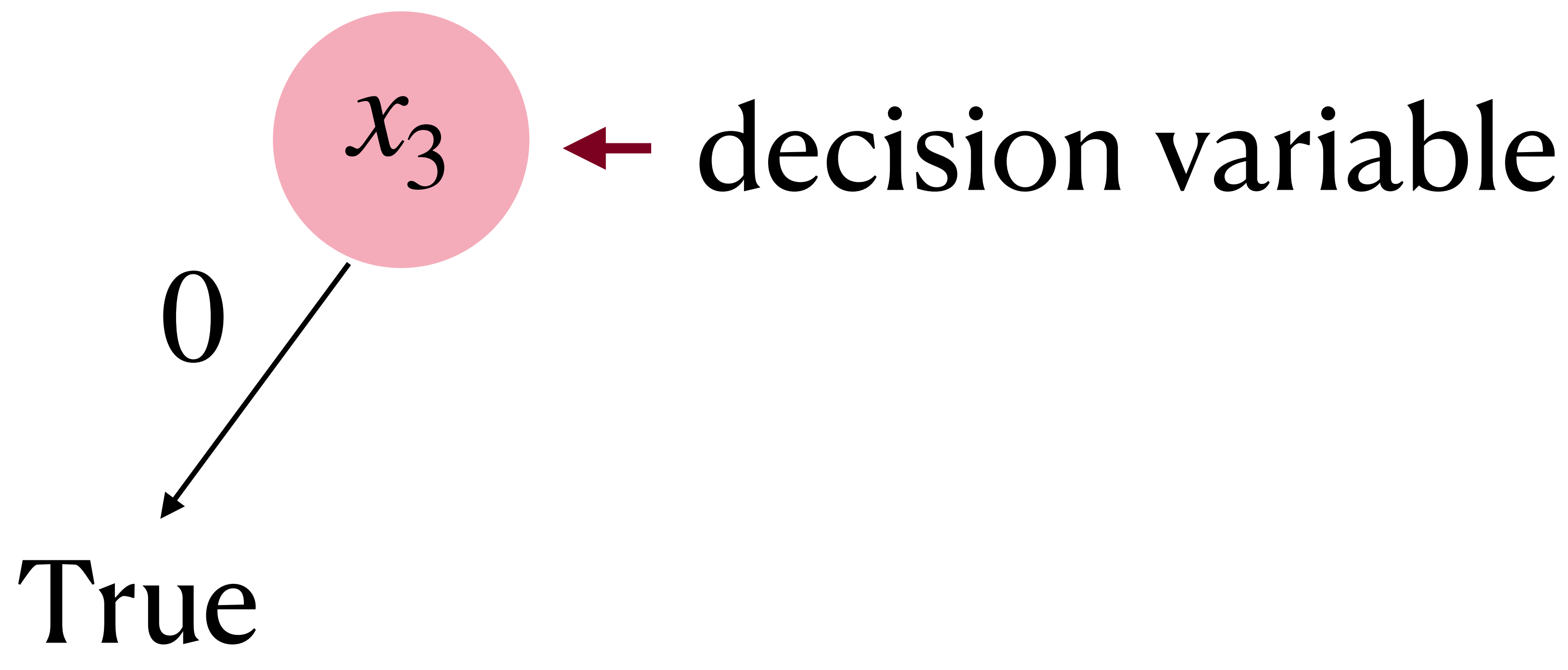
Backtracking

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



Backtracking

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

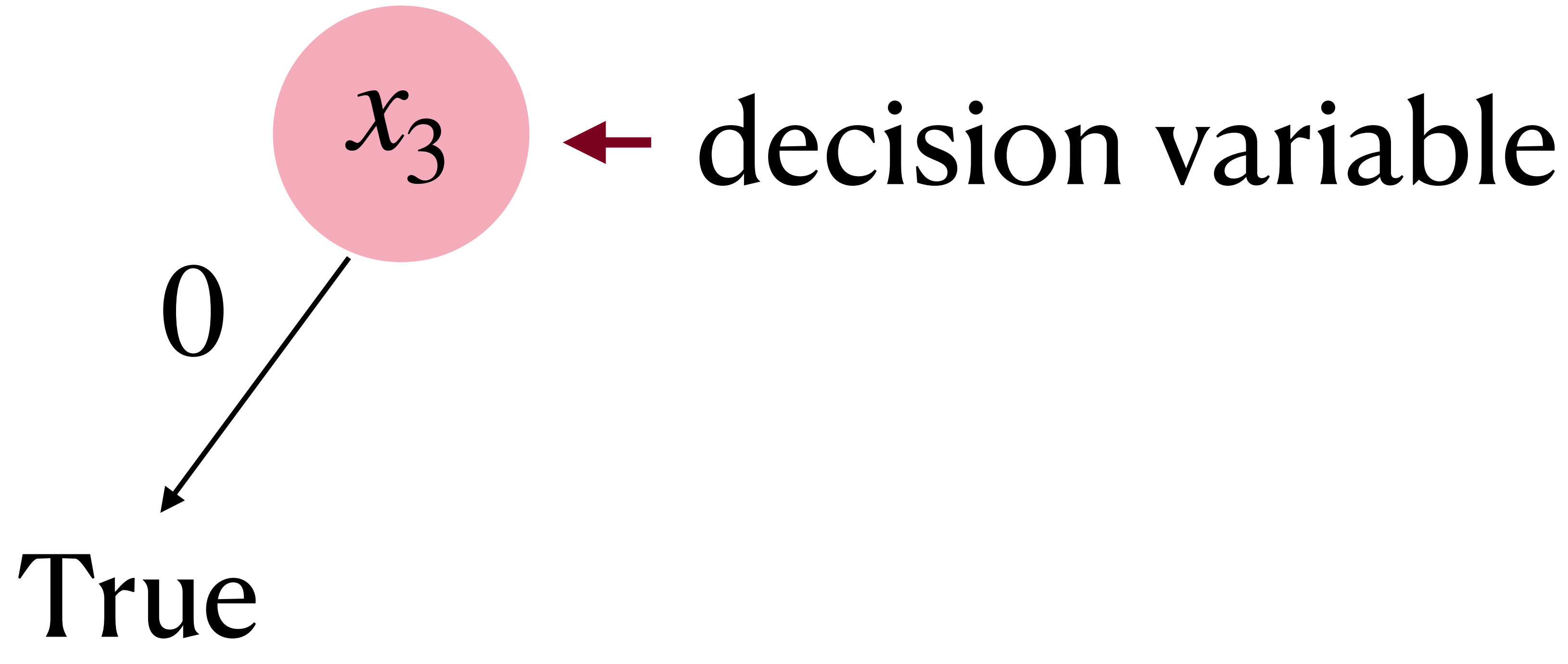


Backtracking

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

Backtrack to last decision, and change the polarity.

Partial model $m = \{x_3 \mapsto 0\}$



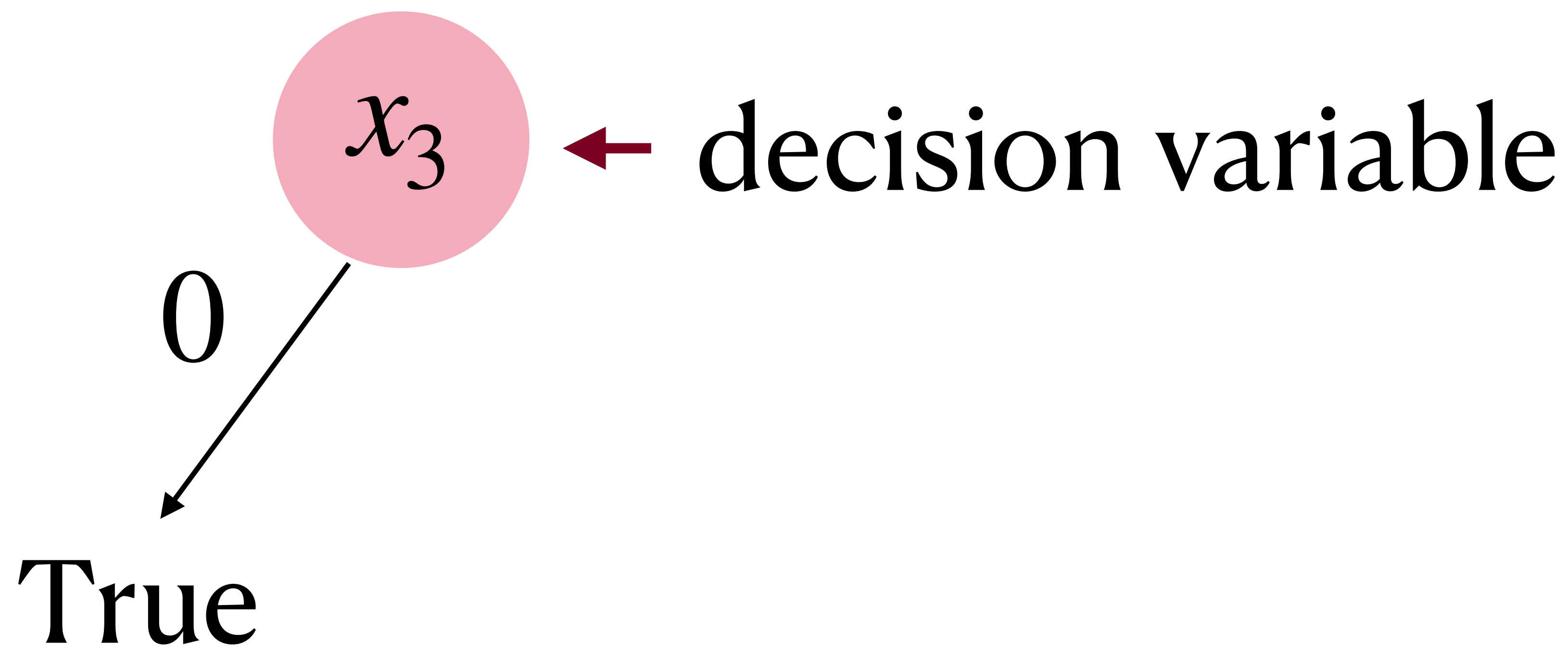
Backtracking

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

Backtrack to last decision, and change the polarity.

Partial model $m = \{x_3 \mapsto 0\}$

All clauses are True, hence F is True



DPLL algorithm (Davis -Putnam-Logemann-Loveland 1960)

$DPLL(F, m = \emptyset)\{$

1. If F is True under m then Return SAT
2. If F is False under m then Return UNSAT
3. If there is a unit literal l under m then Return $DPLL(F, m[l \mapsto 1])$
4. If there is a unit literal $\neg l$ under m then Return $DPLL(F, m[l \mapsto 0])$

   Choose an unassigned variable p, and random bit $b \in \{0,1\}$

5. If $DPLL(F, m[p \mapsto b]) == $ SAT then Return SAT

   Else Return $DPLL(F, m[p \mapsto 1 - b])$

$\}$

DPLL algorithm (Davis -Putnam-Logemann-Loveland 1960)

$DPLL(F, m = \varnothing)\{$

1. If F is True under m then Return SAT
2. If F is False under m then Return UNSAT
3. If there is a unit literal l under m then Return $DPLL(F, m[l \mapsto 1])$
4. If there is a unit literal $\neg l$ under m then Return $DPLL(F, m[l \mapsto 0])$

Unit Propagation

Choose an unassigned variable p, and random bit $b \in \{0,1\}$

5. If $DPLL(F, m[p \mapsto b]) == \text{SAT}$ then Return SAT

   Else Return $DPLL(F, m[p \mapsto 1 - b])$

$\}$

DPLL algorithm (Davis -Putnam-Logemann-Loveland 1960)

$DPLL(F, m = \emptyset)\{$

1. If F is True under m then Return SAT
2. If F is False under m then Return UNSAT
3. If there is a unit literal l under m then Return $DPLL(F, m[l \mapsto 1])$
4. If there is a unit literal $\neg l$ under m then Return $DPLL(F, m[l \mapsto 0])$

   Choose an unassigned variable p, and random bit $b \in \{0,1\}$

5. If $DPLL(F, m[p \mapsto b]) == \text{SAT}$ then Return SAT

   Else Return $DPLL(F, m[p \mapsto 1 - b])$

$\}$

Backtracking at conflict

Unit Propagation

DPLL algorithm (Davis -Putnam-Logemann-Loveland 1960)

1. Maintains a partial model, initially $\emptyset$
2. Assign unassigned variables either 0 or 1
    1. (Randomly one after the other)
3. Sometime forced to make a decision due to unit clause

DPLL run consists of

- Decision
- Unit propagation
- Backtracking

$C_1 = (\neg p_1 \vee p_2)$

$C_2 = (\neg p_1 \vee p_3 \vee p_5)$

$C_3 = (\neg p_2 \vee p_4)$

$C_4 = (\neg p_3 \vee \neg p_4)$

$C_5 = (p_1 \vee p_5 \vee \neg p_2)$

$C_6 = (p_2 \vee p_3)$

$C_7 = (p_2 \vee \neg p_3 \vee p_7)$

$C_8 = (p_6 \vee \neg p_5)$

decision variable

propagated variables

$C_1 = (\neg p_1 \vee p_2)$

$C_2 = (\neg p_1 \vee p_3 \vee p_5)$

$C_3 = (\neg p_2 \vee p_4)$

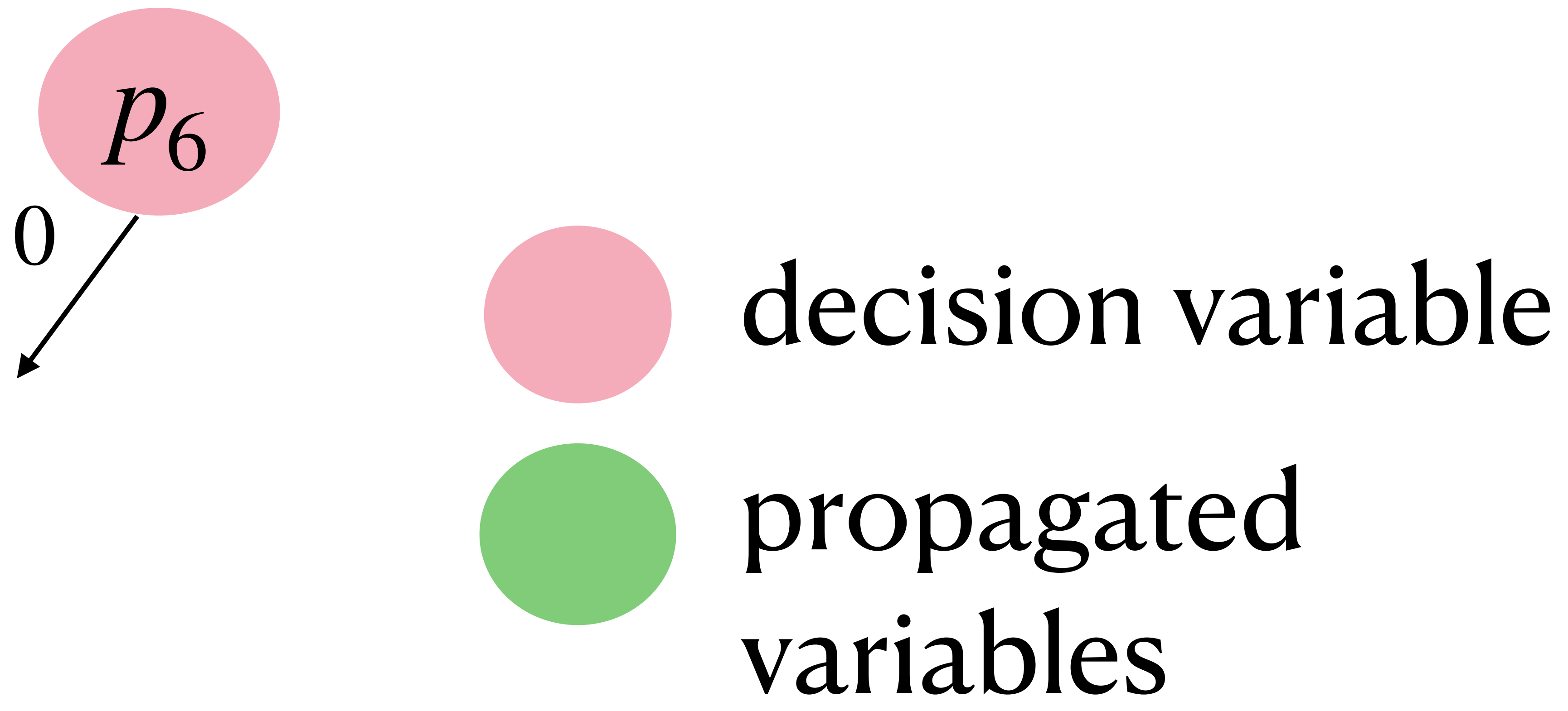
$C_4 = (\neg p_3 \vee \neg p_4)$

$C_5 = (p_1 \vee p_5 \vee \neg p_2)$

$C_6 = (p_2 \vee p_3)$

$C_7 = (p_2 \vee \neg p_3 \vee p_7)$

$C_8 = (p_6 \vee \neg p_5)$



$C_1 = (\neg p_1 \lor p_2)$

$C_2 = (\neg p_1 \lor p_3 \lor p_5)$

$C_3 = (\neg p_2 \lor p_4)$

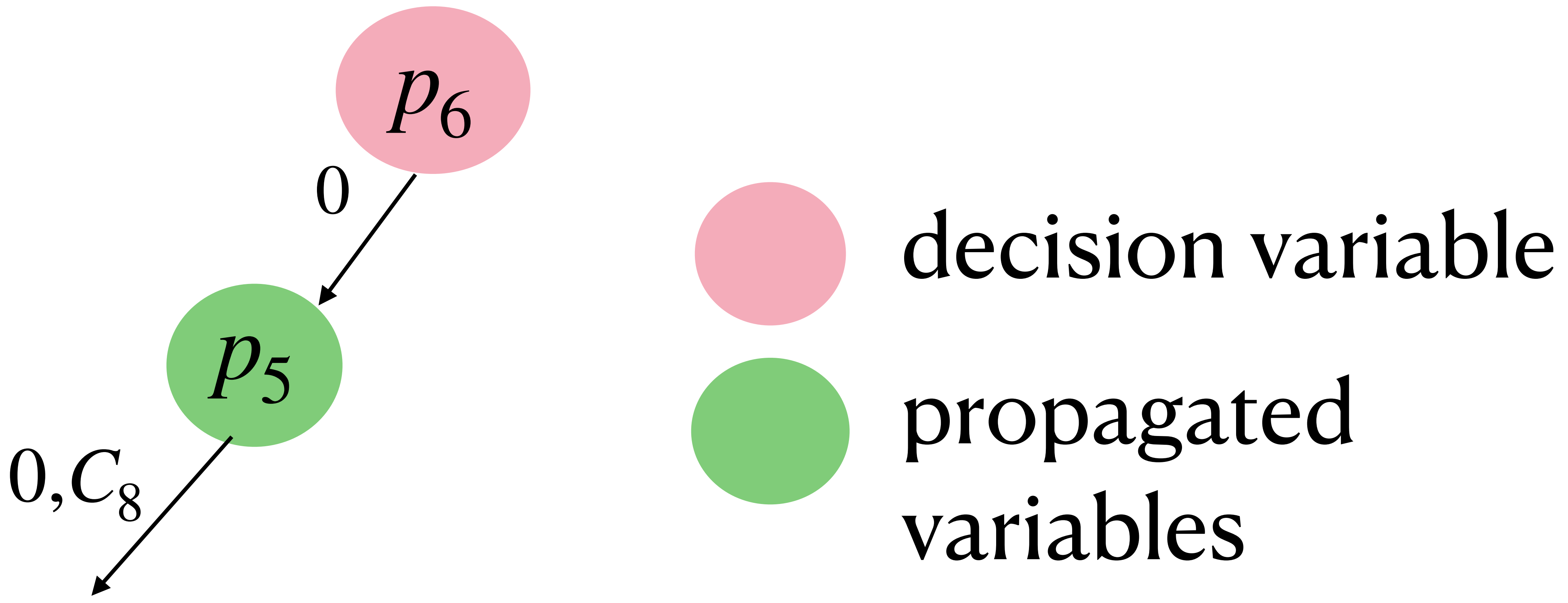
$C_4 = (\neg p_3 \lor \neg p_4)$

$C_5 = (p_1 \lor p_5 \lor \neg p_2)$

$C_6 = (p_2 \lor p_3)$

$C_7 = (p_2 \lor \neg p_3 \lor p_7)$

$C_8 = (p_6 \lor \neg p_5)$



$C_1 = (\neg p_1 \lor p_2)$

$C_2 = (\neg p_1 \lor p_3 \lor p_5)$

$C_3 = (\neg p_2 \lor p_4)$

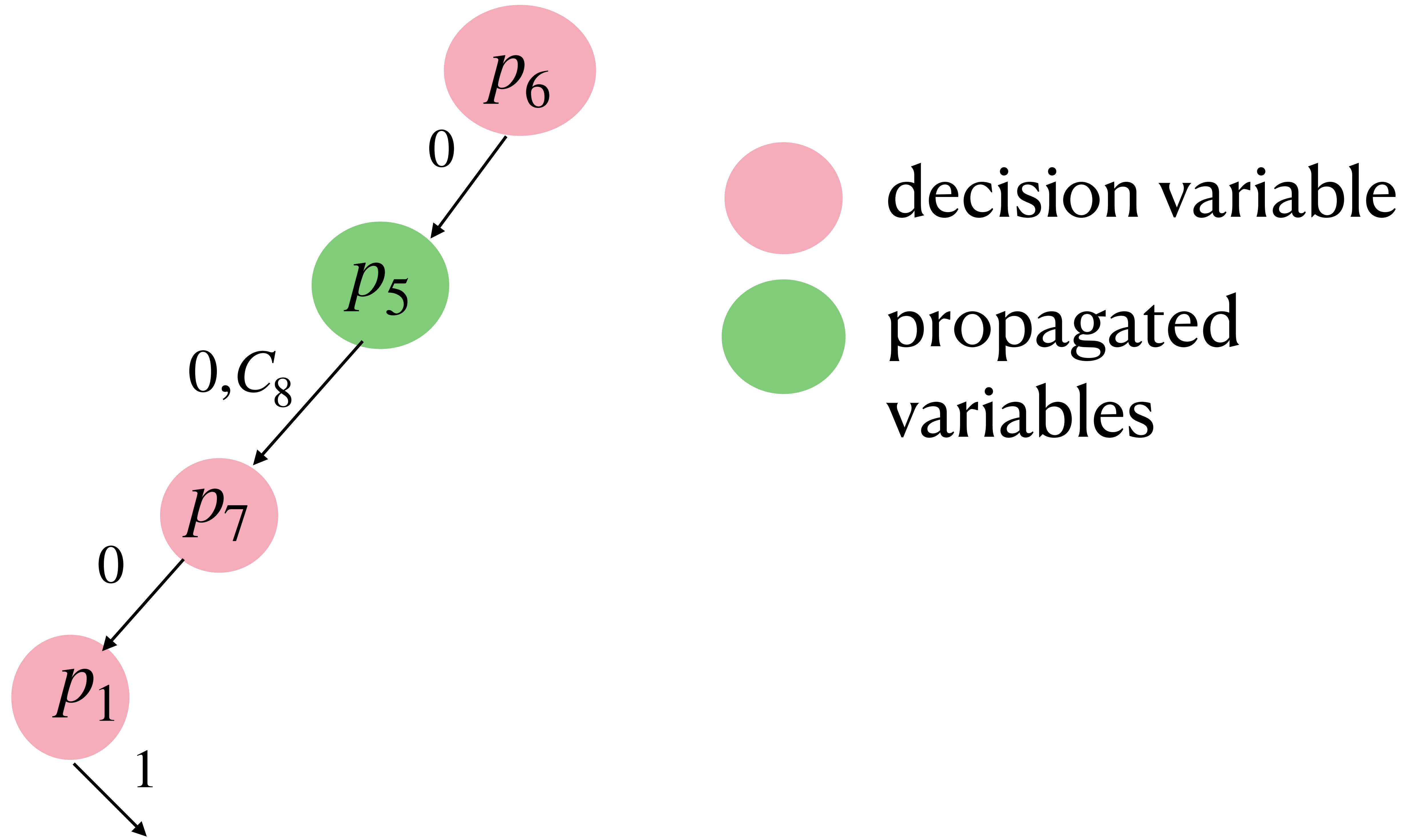
$C_4 = (\neg p_3 \lor \neg p_4)$

$C_5 = (p_1 \lor p_5 \lor \neg p_2)$

$C_6 = (p_2 \lor p_3)$

$C_7 = (p_2 \lor \neg p_3 \lor p_7)$

$C_8 = (p_6 \lor \neg p_5)$



$C_1 = (\neg p_1 \vee p_2)$

$C_2 = (\neg p_1 \vee p_3 \vee p_5)$

$C_3 = (\neg p_2 \vee p_4)$

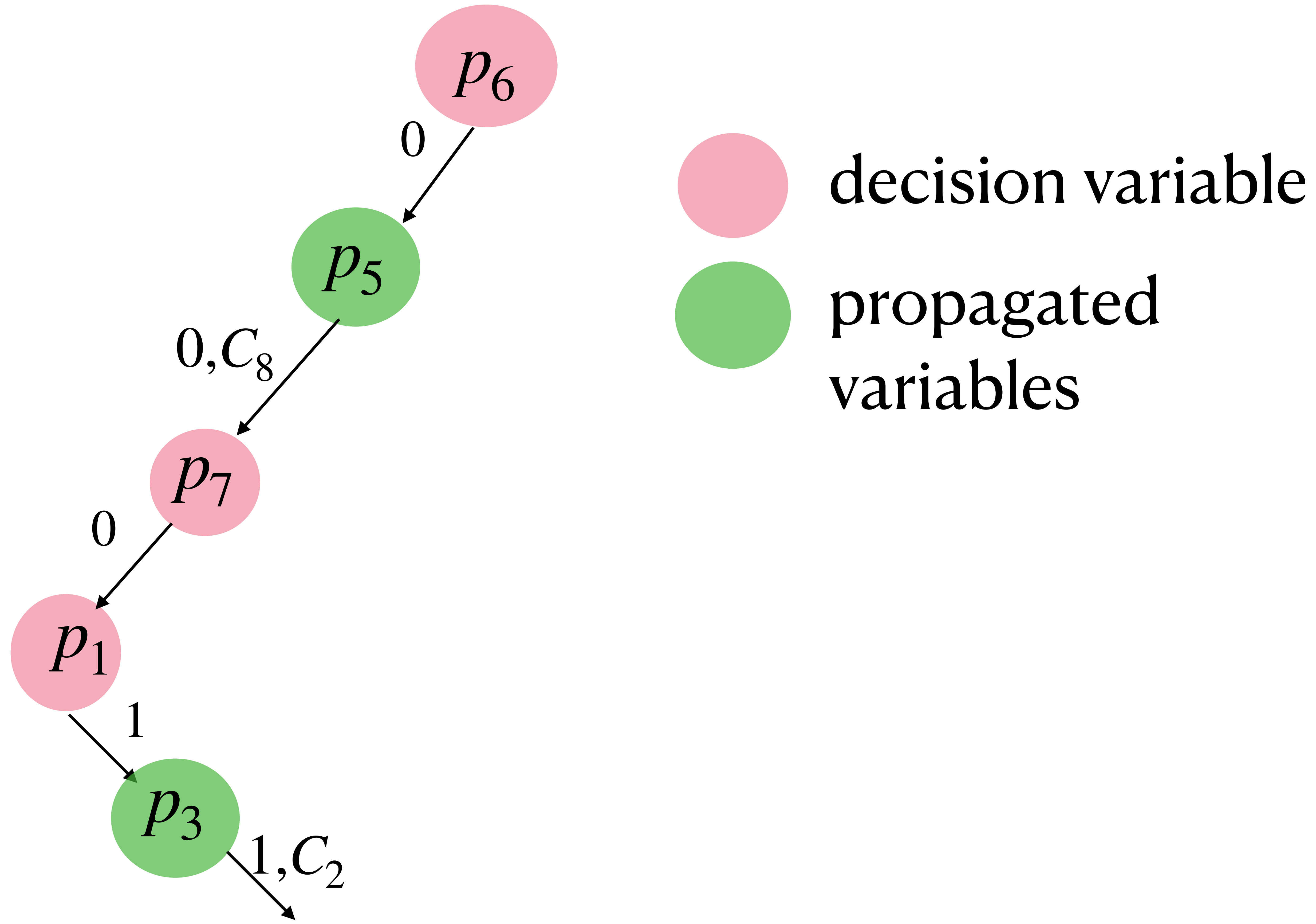
$C_4 = (\neg p_3 \vee \neg p_4)$

$C_5 = (p_1 \vee p_5 \vee \neg p_2)$

$C_6 = (p_2 \vee p_3)$

$C_7 = (p_2 \vee \neg p_3 \vee p_7)$

$C_8 = (p_6 \vee \neg p_5)$



$C_1 = (\neg p_1 \vee p_2)$

$C_2 = (\neg p_1 \vee p_3 \vee p_5)$

$C_3 = (\neg p_2 \vee p_4)$

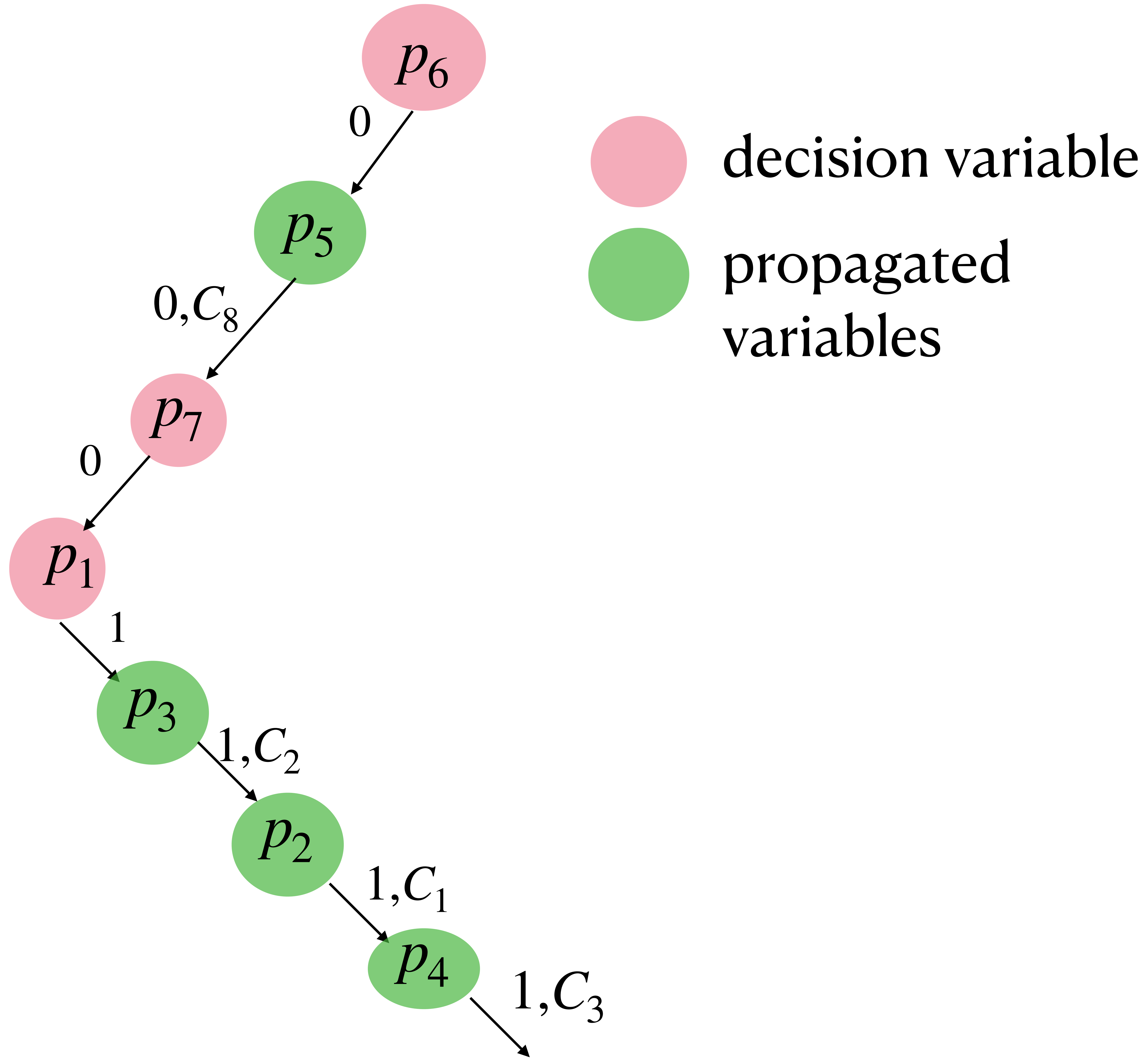
$C_4 = (\neg p_3 \vee \neg p_4)$

$C_5 = (p_1 \vee p_5 \vee \neg p_2)$

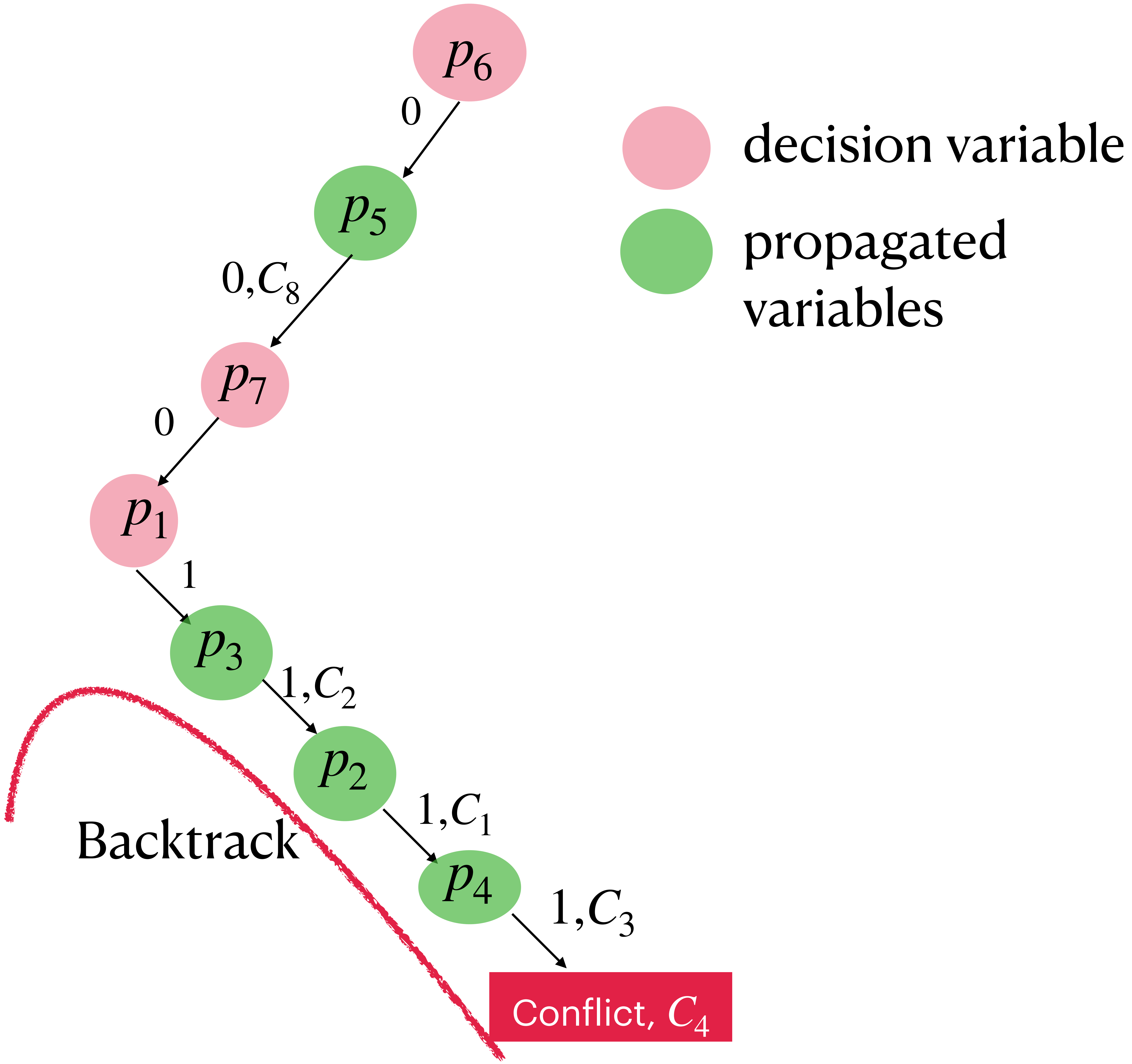
$C_6 = (p_2 \vee p_3)$

$C_7 = (p_2 \vee \neg p_3 \vee p_7)$

$C_8 = (p_6 \vee \neg p_5)$



$$C_1 = (\neg p_1 \vee p_2)$$
$$C_2 = (\neg p_1 \vee p_3 \vee p_5)$$
$$C_3 = (\neg p_2 \vee p_4)$$
$$C_4 = (\neg p_3 \vee \neg p_4)$$
$$C_5 = (p_1 \vee p_5 \vee \neg p_2)$$
$$C_6 = (p_2 \vee p_3)$$
$$C_7 = (p_2 \vee \neg p_3 \vee p_7)$$
$$C_8 = (p_6 \vee \neg p_5)$$



$C_1 = (\neg p_1 \vee p_2)$

$C_2 = (\neg p_1 \vee p_3 \vee p_5)$

$C_3 = (\neg p_2 \vee p_4)$

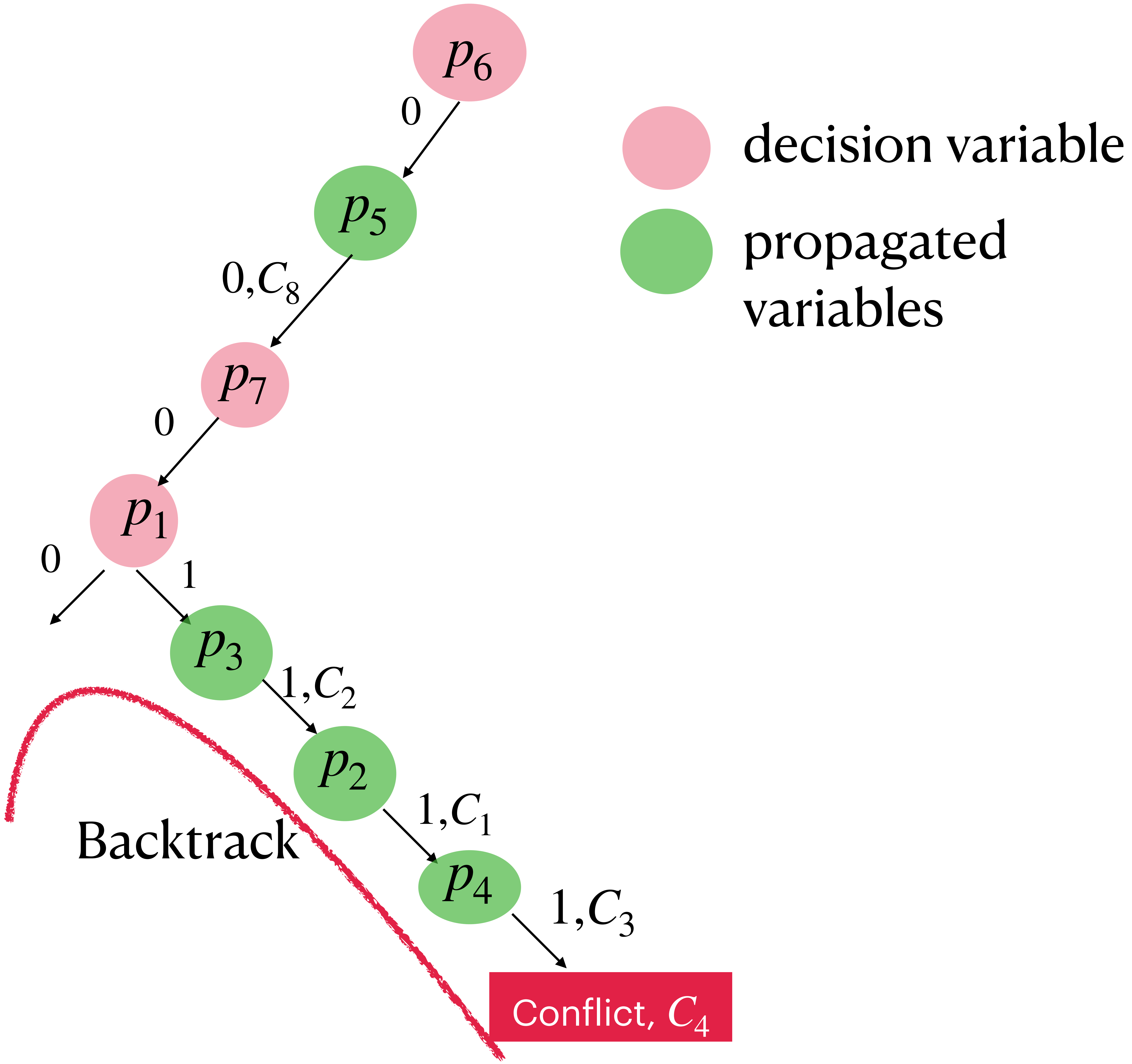
$C_4 = (\neg p_3 \vee \neg p_4)$

$C_5 = (p_1 \vee p_5 \vee \neg p_2)$

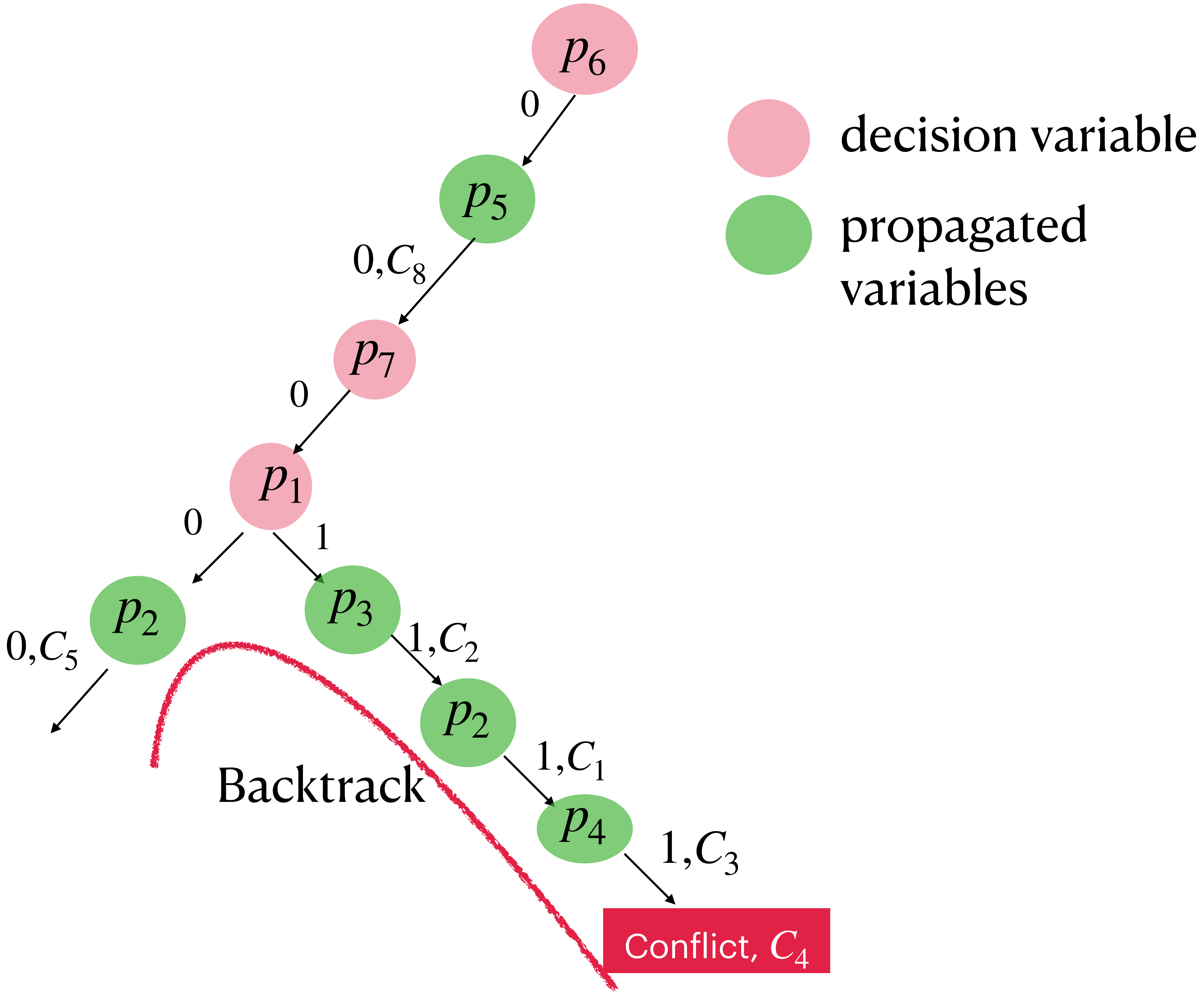
$C_6 = (p_2 \vee p_3)$

$C_7 = (p_2 \vee \neg p_3 \vee p_7)$

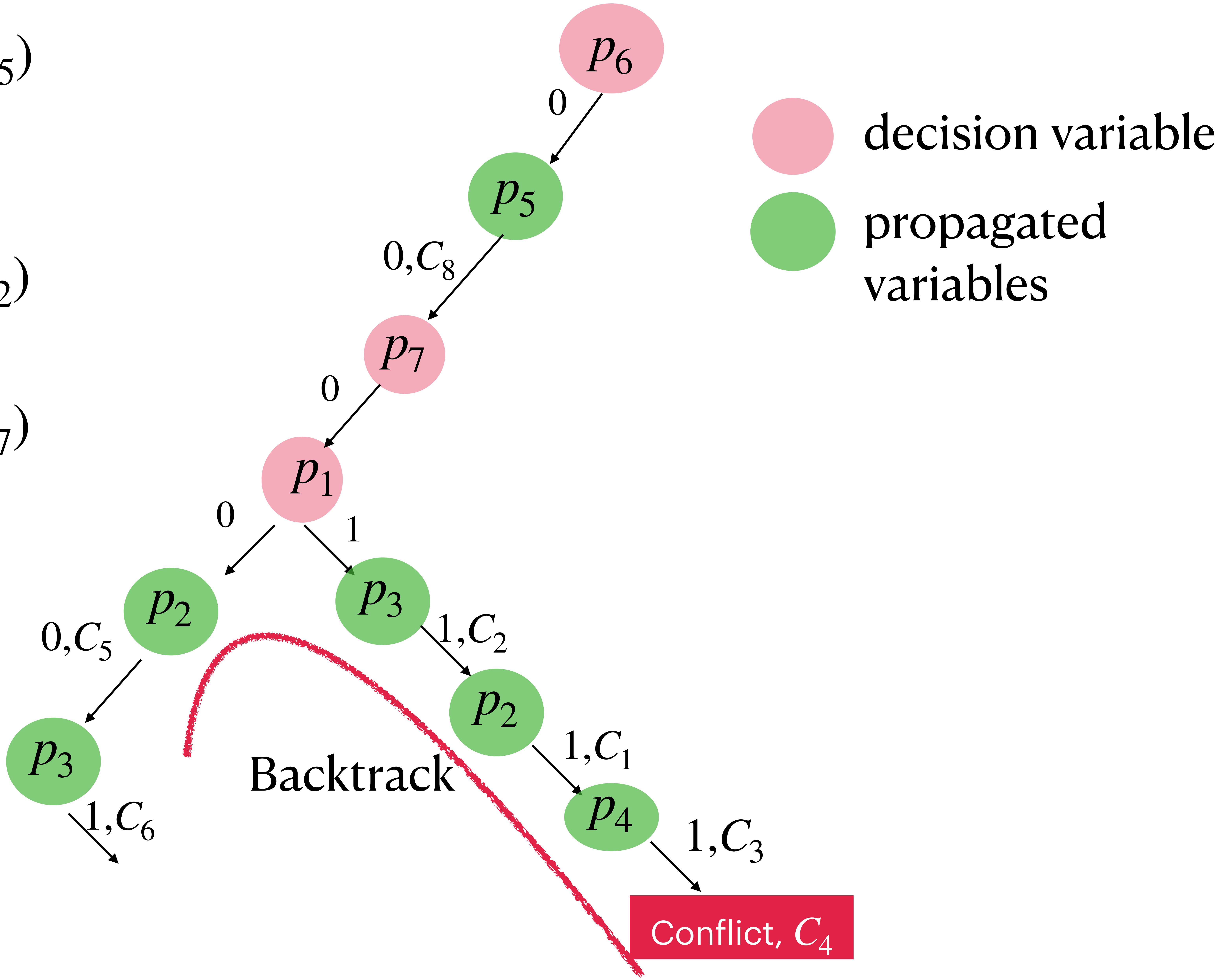
$C_8 = (p_6 \vee \neg p_5)$



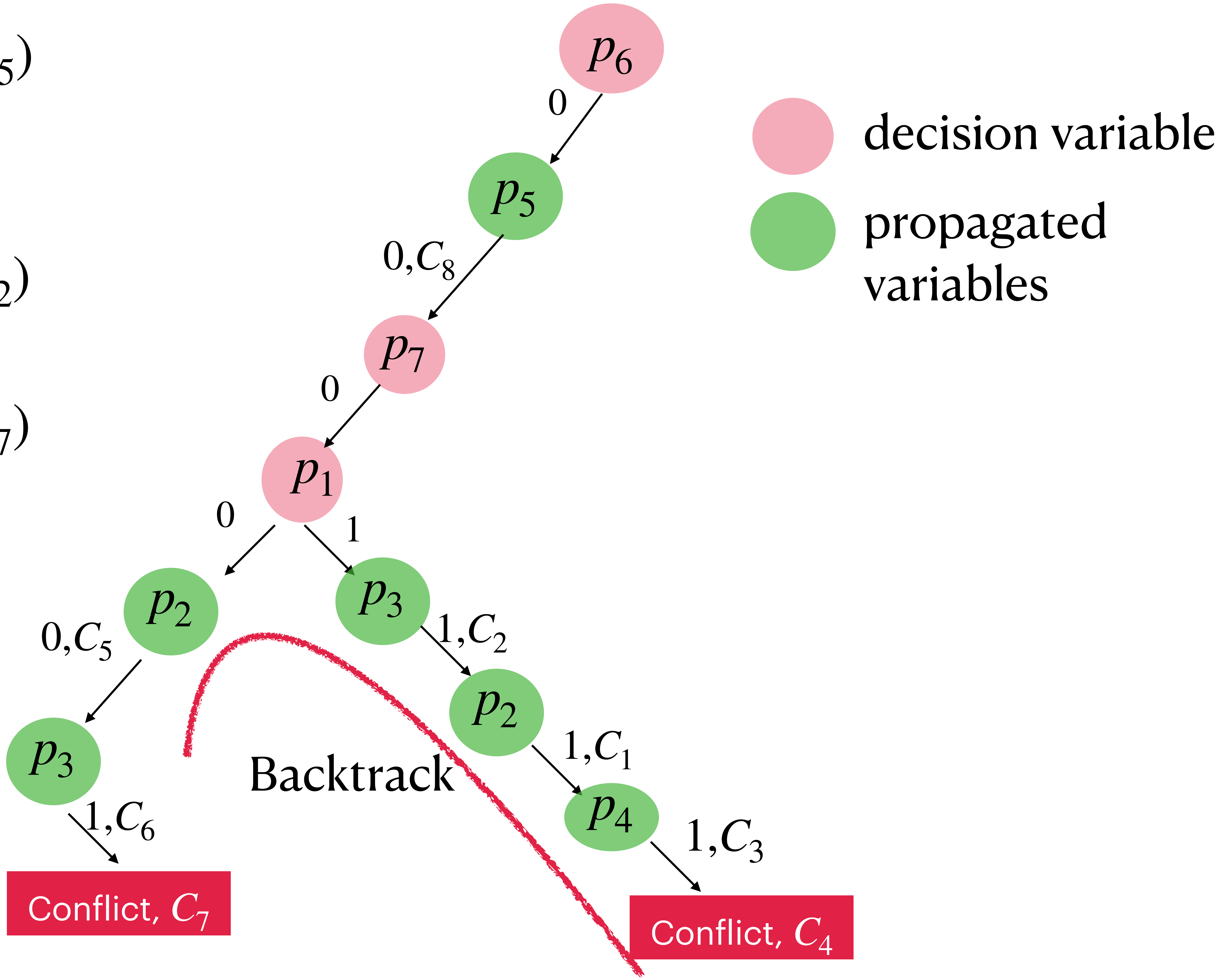
$$C_1 = (\neg p_1 \lor p_2)$$
$$C_2 = (\neg p_1 \lor p_3 \lor p_5)$$
$$C_3 = (\neg p_2 \lor p_4)$$
$$C_4 = (\neg p_3 \lor \neg p_4)$$
$$C_5 = (p_1 \lor p_5 \lor \neg p_2)$$
$$C_6 = (p_2 \lor p_3)$$
$$C_7 = (p_2 \lor \neg p_3 \lor p_7)$$
$$C_8 = (p_6 \lor \neg p_5)$$



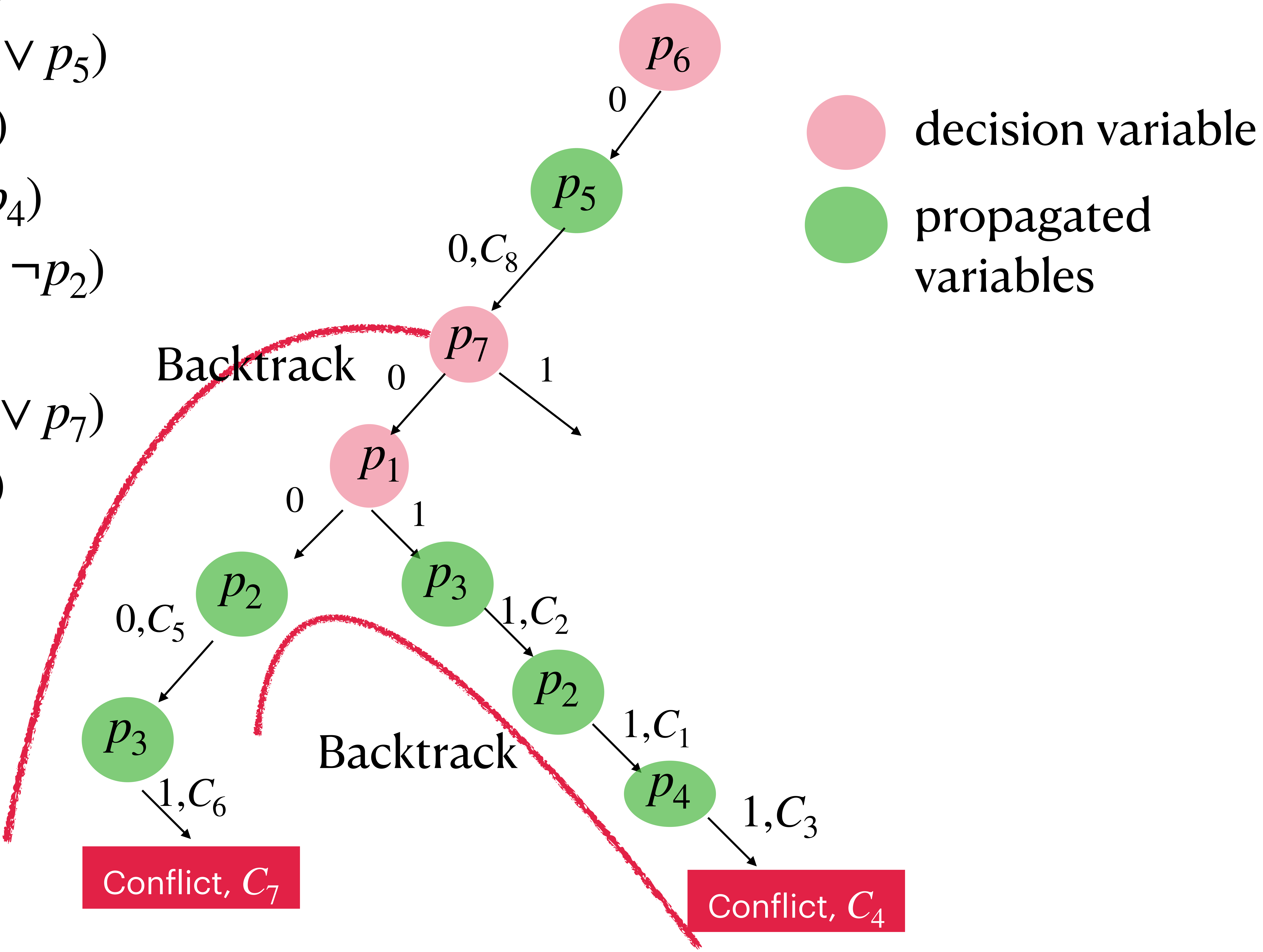
$C_1 = (\neg p_1 \vee p_2)$
$C_2 = (\neg p_1 \vee p_3 \vee p_5)$
$C_3 = (\neg p_2 \vee p_4)$
$C_4 = (\neg p_3 \vee \neg p_4)$
$C_5 = (p_1 \vee p_5 \vee \neg p_2)$
$C_6 = (p_2 \vee p_3)$
$C_7 = (p_2 \vee \neg p_3 \vee p_7)$
$C_8 = (p_6 \vee \neg p_5)$



$$C_1 = (\neg p_1 \vee p_2)$$
$$C_2 = (\neg p_1 \vee p_3 \vee p_5)$$
$$C_3 = (\neg p_2 \vee p_4)$$
$$C_4 = (\neg p_3 \vee \neg p_4)$$
$$C_5 = (p_1 \vee p_5 \vee \neg p_2)$$
$$C_6 = (p_2 \vee p_3)$$
$$C_7 = (p_2 \vee \neg p_3 \vee p_7)$$
$$C_8 = (p_6 \vee \neg p_5)$$



$$C_1 = (\neg p_1 \vee p_2)$$
$$C_2 = (\neg p_1 \vee p_3 \vee p_5)$$
$$C_3 = (\neg p_2 \vee p_4)$$
$$C_4 = (\neg p_3 \vee \neg p_4)$$
$$C_5 = (p_1 \vee p_5 \vee \neg p_2)$$
$$C_6 = (p_2 \vee p_3)$$
$$C_7 = (p_2 \vee \neg p_3 \vee p_7)$$
$$C_8 = (p_6 \vee \neg p_5)$$



CDCL: Conflict Driven Clause Learning

An optimization of DPLL:

As we decide and propagate, we can observe the run, and avoid unnecessary backtracking.

CDCL: Conflict Driven Clause Learning

An optimization of DPLL:

As we decide and propagate, we can observe the run, and avoid unnecessary backtracking.

Construct a data structure to avoid unnecessary backtracking.

CDCL: Conflict Driven Clause Learning

An optimization of DPLL:

As we decide and propagate, we can observe the run, and avoid unnecessary backtracking.

Construct a data structure to avoid unnecessary backtracking.

Implication Graph.

$$C_1 = (\neg p_1 \vee p_2)$$
$$C_2 = (\neg p_1 \vee p_3 \vee p_5)$$
$$C_3 = (\neg p_2 \vee p_4)$$
$$C_4 = (\neg p_3 \vee \neg p_4)$$
$$C_5 = (p_1 \vee p_5 \vee \neg p_2)$$
$$C_6 = (p_2 \vee p_3)$$
$$C_7 = (p_2 \vee \neg p_3 \vee p_7)$$
$$C_8 = (p_6 \vee \neg p_5)$$

$$C_1 = (\neg p_1 \vee p_2)$$

$$C_2 = (\neg p_1 \vee p_3 \vee p_5)$$

$$C_3 = (\neg p_2 \vee p_4)$$

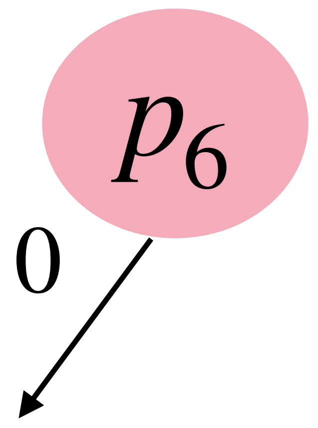
$$C_4 = (\neg p_3 \vee \neg p_4)$$

$$C_5 = (p_1 \vee p_5 \vee \neg p_2)$$

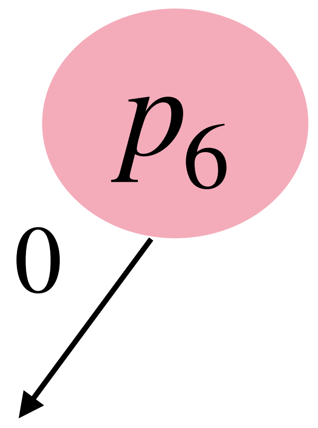
$$C_6 = (p_2 \vee p_3)$$

$$C_7 = (p_2 \vee \neg p_3 \vee p_7)$$

$$C_8 = (p_6 \vee \neg p_5)$$



$$C_1 = (\neg p_1 \vee p_2)$$
$$C_2 = (\neg p_1 \vee p_3 \vee p_5)$$
$$C_3 = (\neg p_2 \vee p_4)$$
$$C_4 = (\neg p_3 \vee \neg p_4)$$
$$C_5 = (p_1 \vee p_5 \vee \neg p_2)$$
$$C_6 = (p_2 \vee p_3)$$
$$C_7 = (p_2 \vee \neg p_3 \vee p_7)$$
$$C_8 = (p_6 \vee \neg p_5)$$



$\neg p_6 @ 1$

$C_1 = (\neg p_1 \vee p_2)$

$C_2 = (\neg p_1 \vee p_3 \vee p_5)$

$C_3 = (\neg p_2 \vee p_4)$

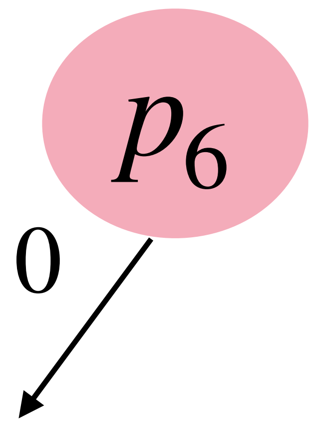
$C_4 = (\neg p_3 \vee \neg p_4)$

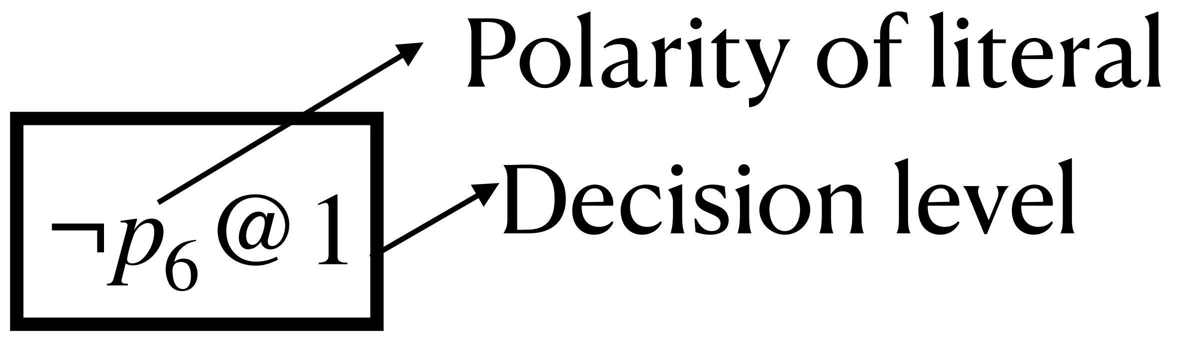
$C_5 = (p_1 \vee p_5 \vee \neg p_2)$

$C_6 = (p_2 \vee p_3)$

$C_7 = (p_2 \vee \neg p_3 \vee p_7)$

$C_8 = (p_6 \vee \neg p_5)$





$C_1 = (\neg p_1 \lor p_2)$

$C_2 = (\neg p_1 \lor p_3 \lor p_5)$

$C_3 = (\neg p_2 \lor p_4)$

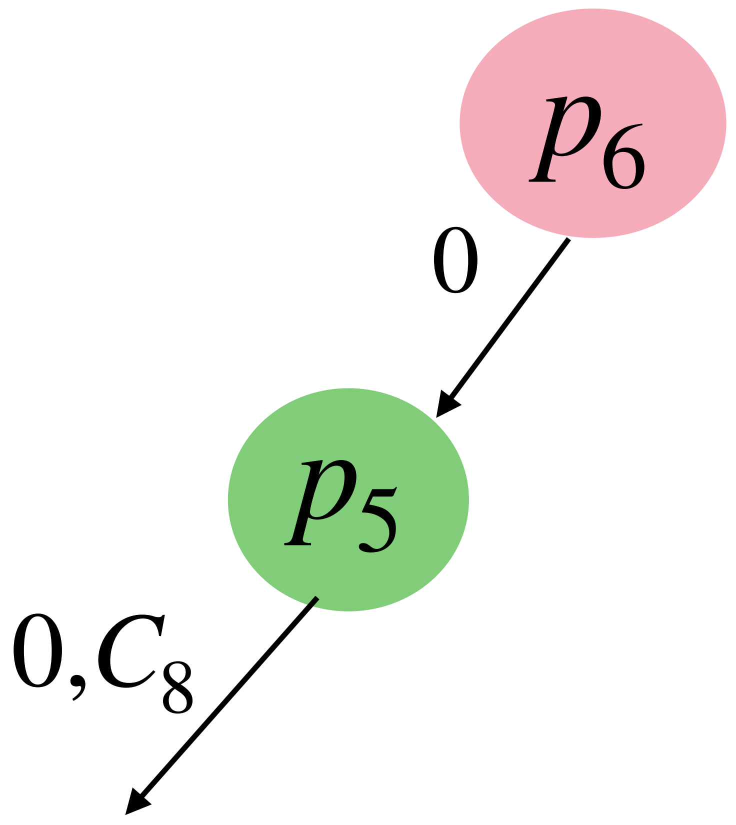
$C_4 = (\neg p_3 \lor \neg p_4)$

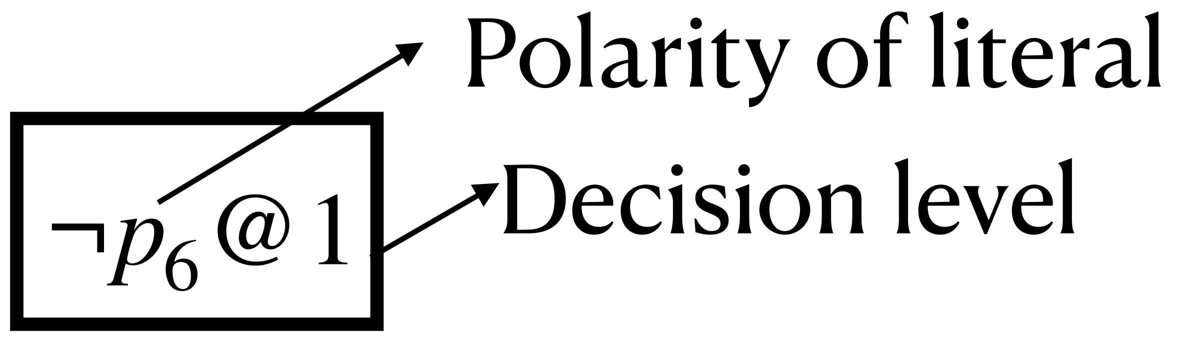
$C_5 = (p_1 \lor p_5 \lor \neg p_2)$

$C_6 = (p_2 \lor p_3)$

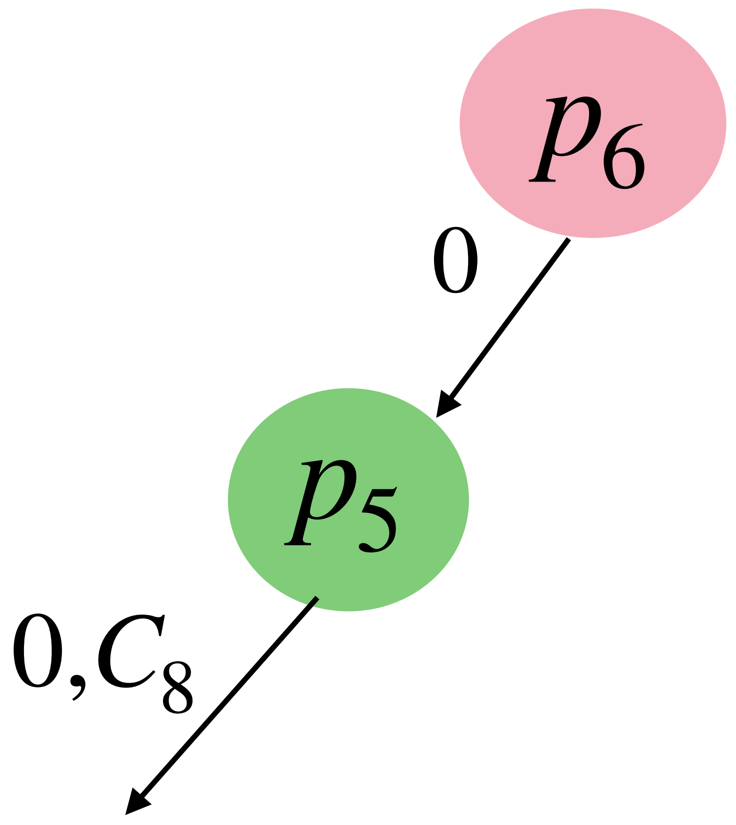
$C_7 = (p_2 \lor \neg p_3 \lor p_7)$

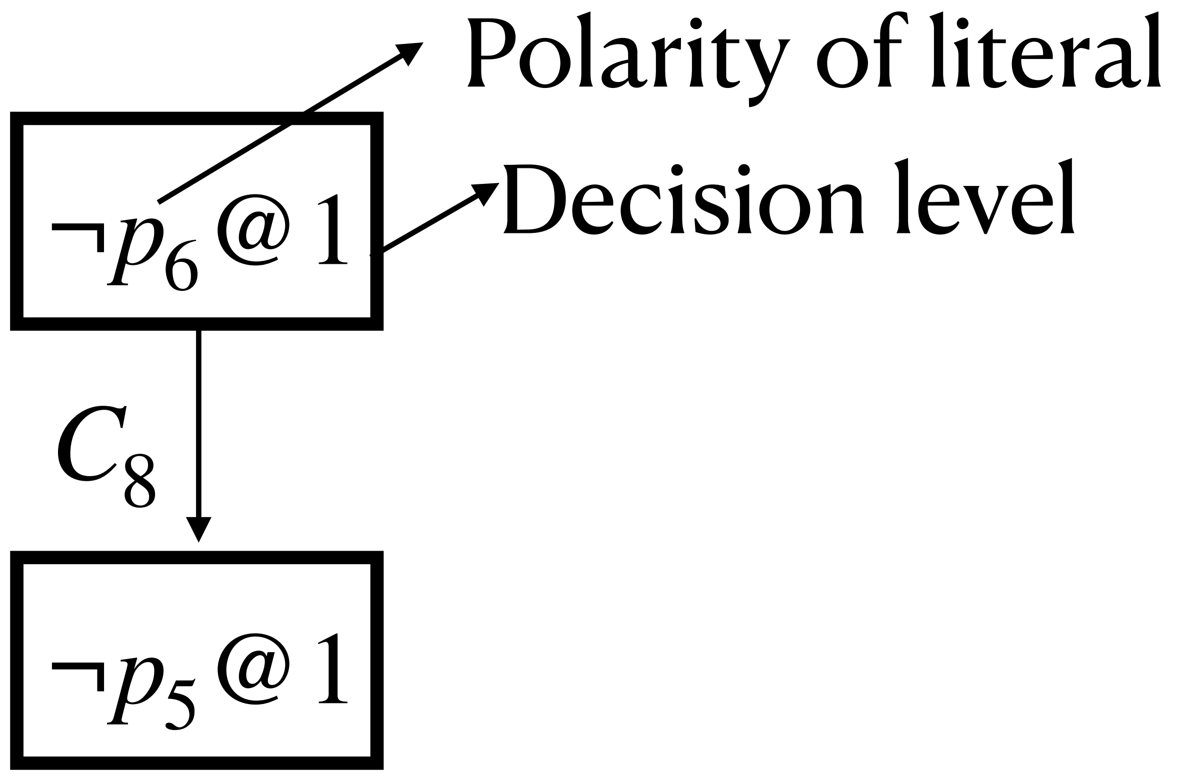
$C_8 = (p_6 \lor \neg p_5)$





$$C_1 = (\neg p_1 \vee p_2)$$
$$C_2 = (\neg p_1 \vee p_3 \vee p_5)$$
$$C_3 = (\neg p_2 \vee p_4)$$
$$C_4 = (\neg p_3 \vee \neg p_4)$$
$$C_5 = (p_1 \vee p_5 \vee \neg p_2)$$
$$C_6 = (p_2 \vee p_3)$$
$$C_7 = (p_2 \vee \neg p_3 \vee p_7)$$
$$C_8 = (p_6 \vee \neg p_5)$$





$C_1 = (\neg p_1 \vee p_2)$

$C_2 = (\neg p_1 \vee p_3 \vee p_5)$

$C_3 = (\neg p_2 \vee p_4)$

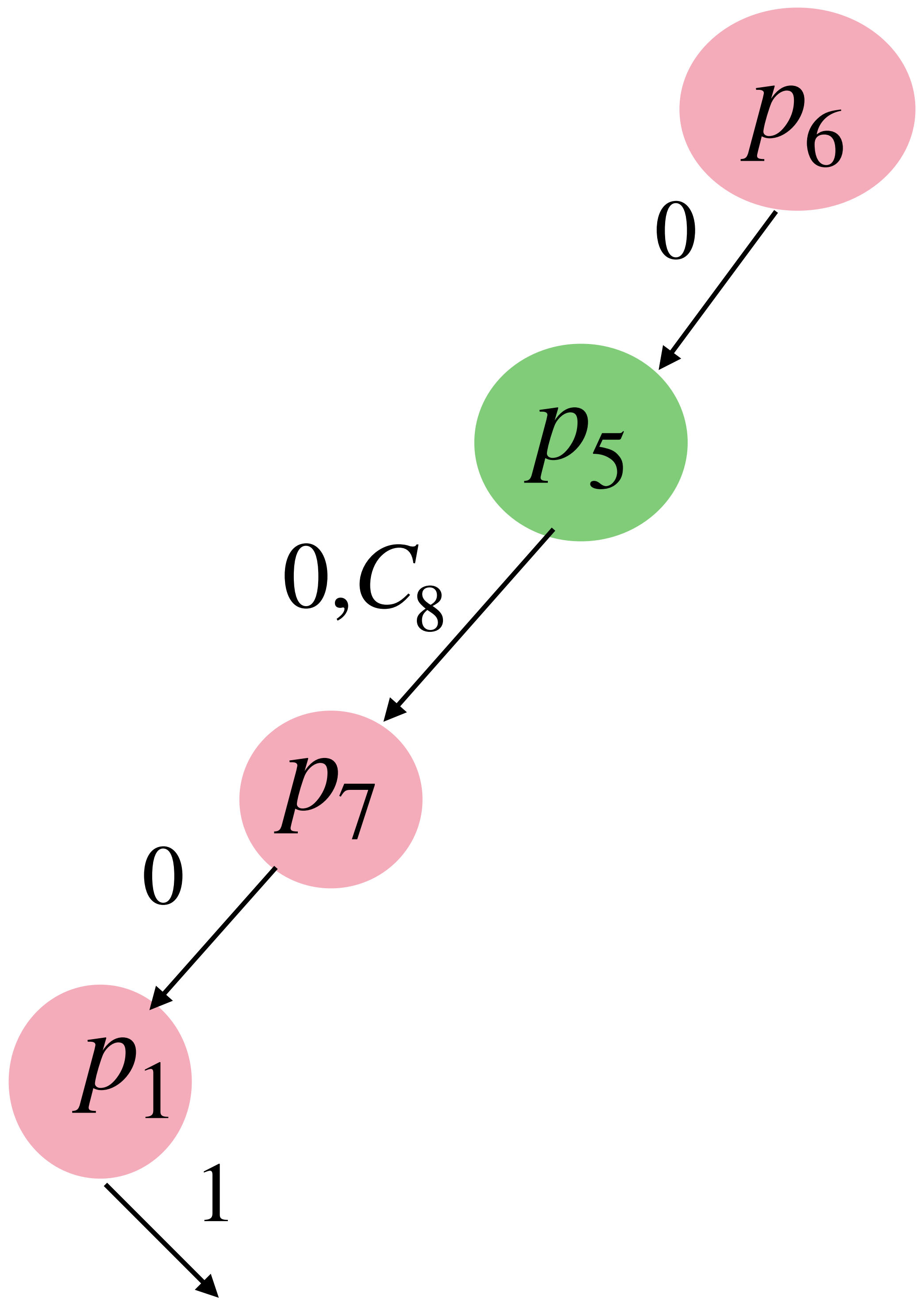
$C_4 = (\neg p_3 \vee \neg p_4)$

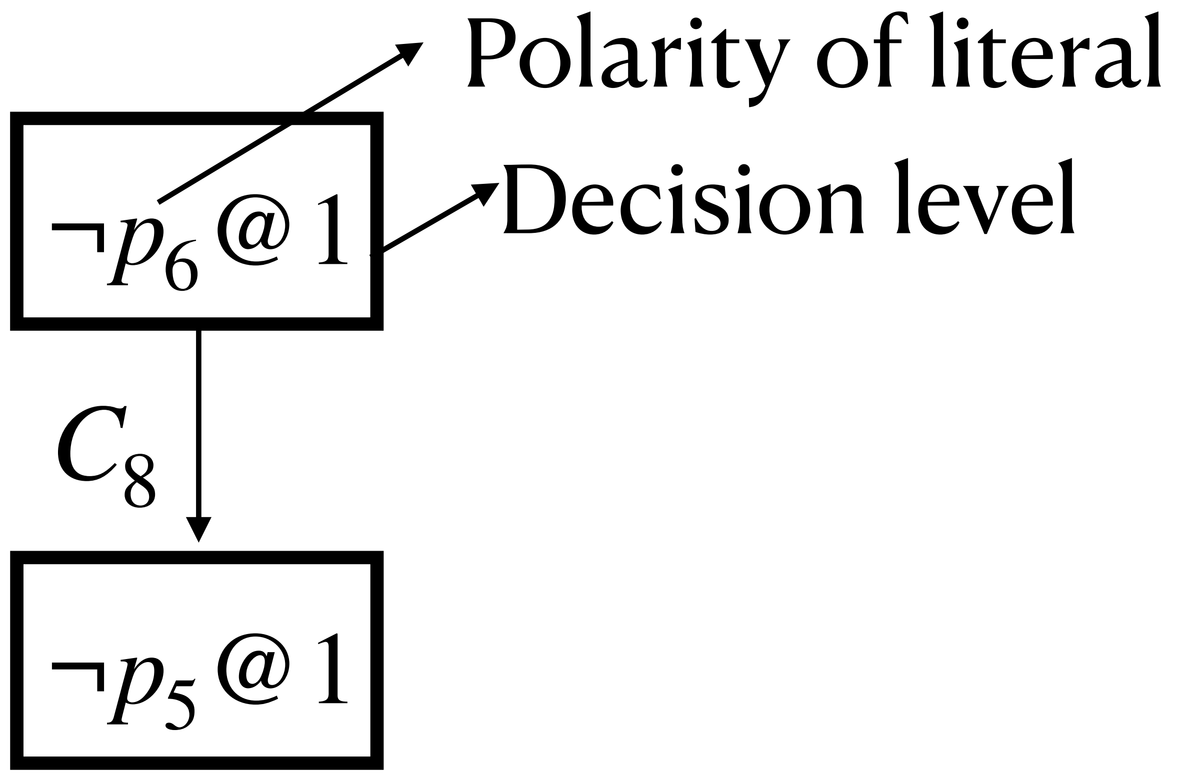
$C_5 = (p_1 \vee p_5 \vee \neg p_2)$

$C_6 = (p_2 \vee p_3)$

$C_7 = (p_2 \vee \neg p_3 \vee p_7)$

$C_8 = (p_6 \vee \neg p_5)$





$C_1 = (\neg p_1 \vee p_2)$

$C_2 = (\neg p_1 \vee p_3 \vee p_5)$

$C_3 = (\neg p_2 \vee p_4)$

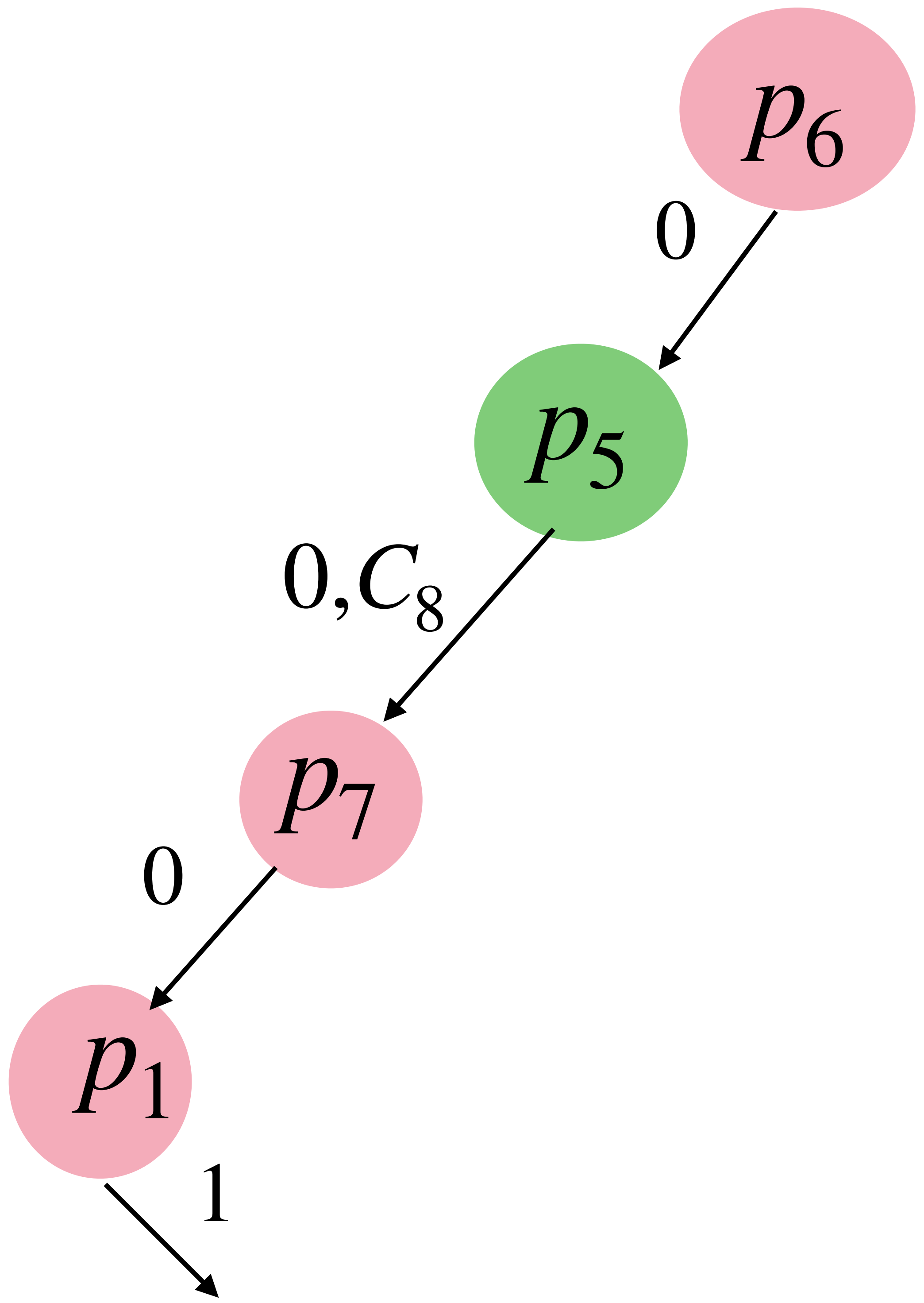
$C_4 = (\neg p_3 \vee \neg p_4)$

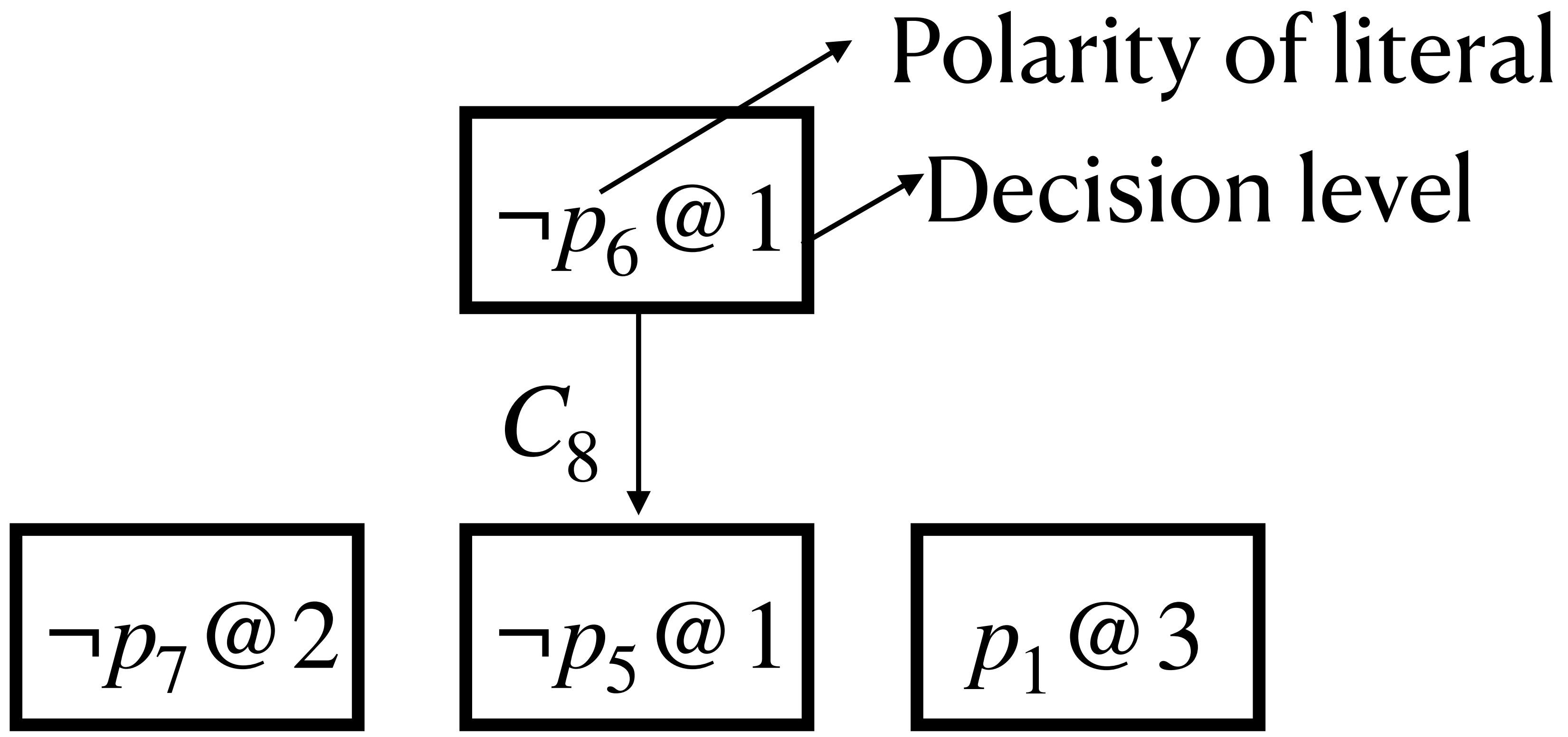
$C_5 = (p_1 \vee p_5 \vee \neg p_2)$

$C_6 = (p_2 \vee p_3)$

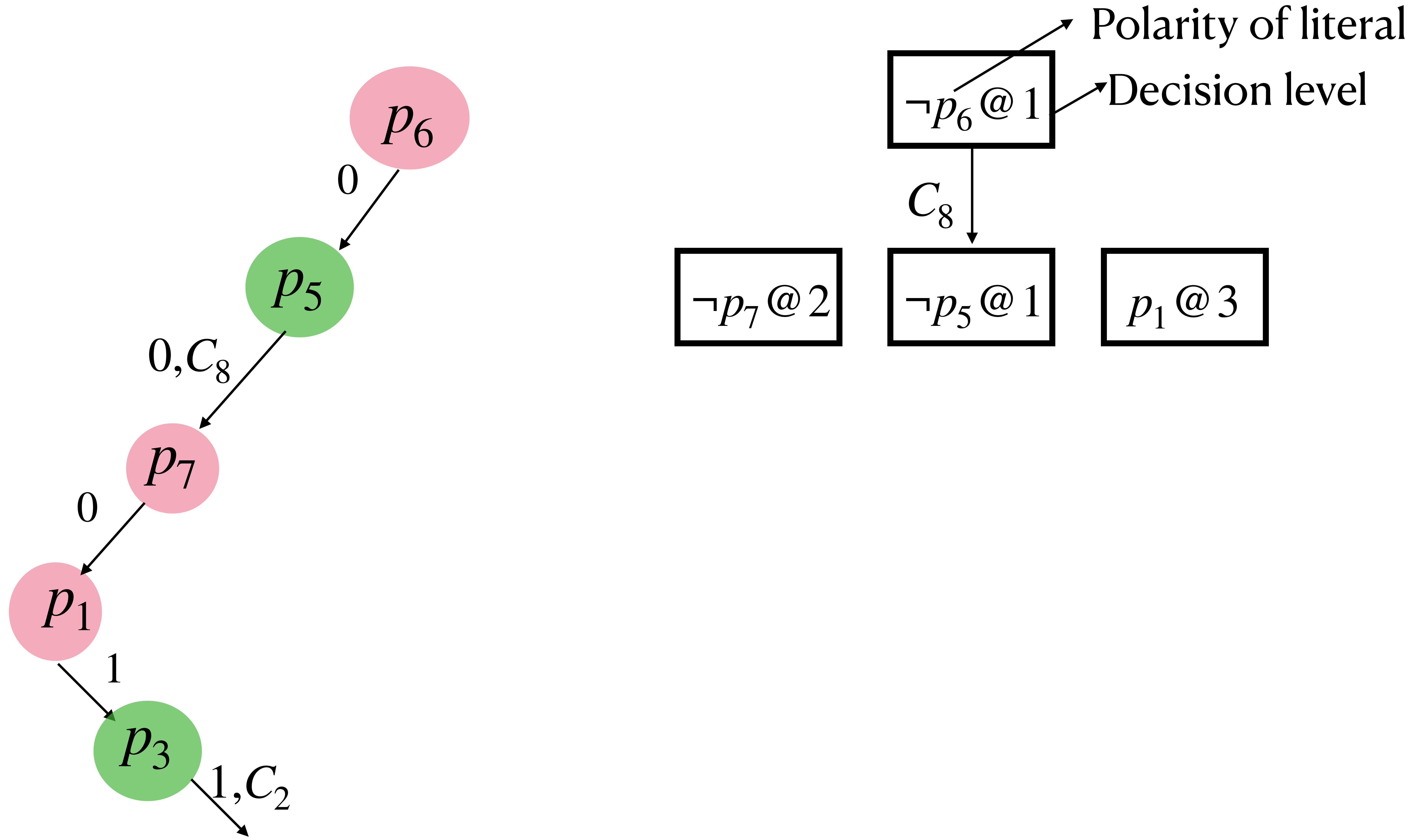
$C_7 = (p_2 \vee \neg p_3 \vee p_7)$

$C_8 = (p_6 \vee \neg p_5)$





$$C_1 = (\neg p_1 \lor p_2)$$
$$C_2 = (\neg p_1 \lor p_3 \lor p_5)$$
$$C_3 = (\neg p_2 \lor p_4)$$
$$C_4 = (\neg p_3 \lor \neg p_4)$$
$$C_5 = (p_1 \lor p_5 \lor \neg p_2)$$
$$C_6 = (p_2 \lor p_3)$$
$$C_7 = (p_2 \lor \neg p_3 \lor p_7)$$
$$C_8 = (p_6 \lor \neg p_5)$$



$C_1 = (\neg p_1 \lor p_2)$

$C_2 = (\neg p_1 \lor p_3 \lor p_5)$

$C_3 = (\neg p_2 \lor p_4)$

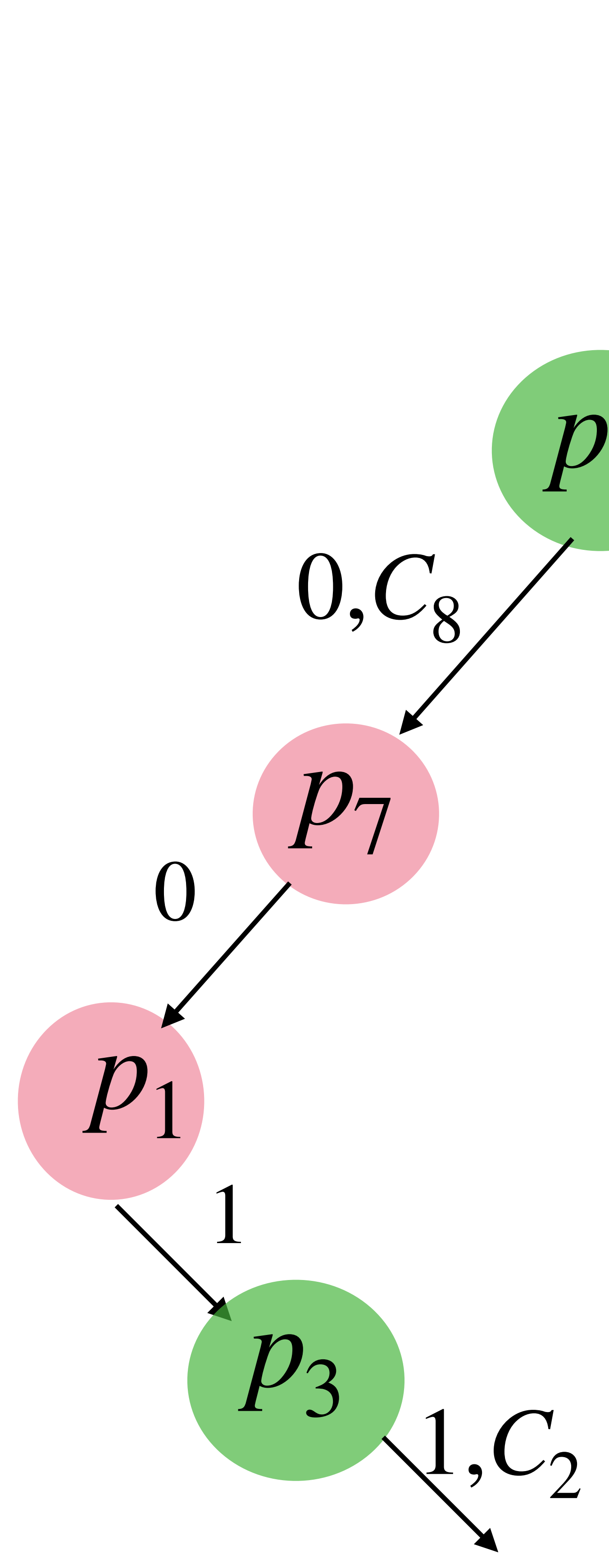
$C_4 = (\neg p_3 \lor \neg p_4)$

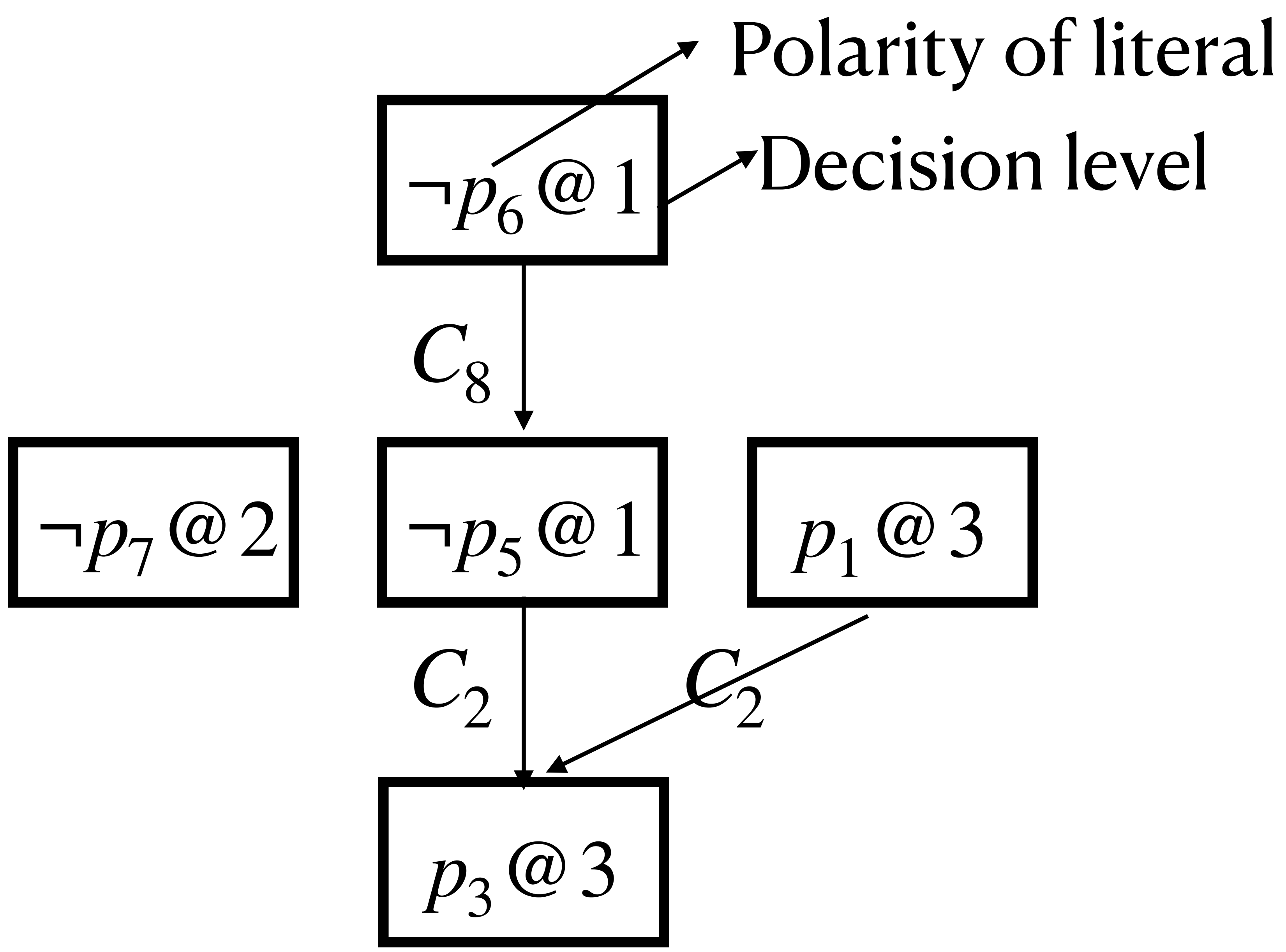
$C_5 = (p_1 \lor p_5 \lor \neg p_2)$

$C_6 = (p_2 \lor p_3)$

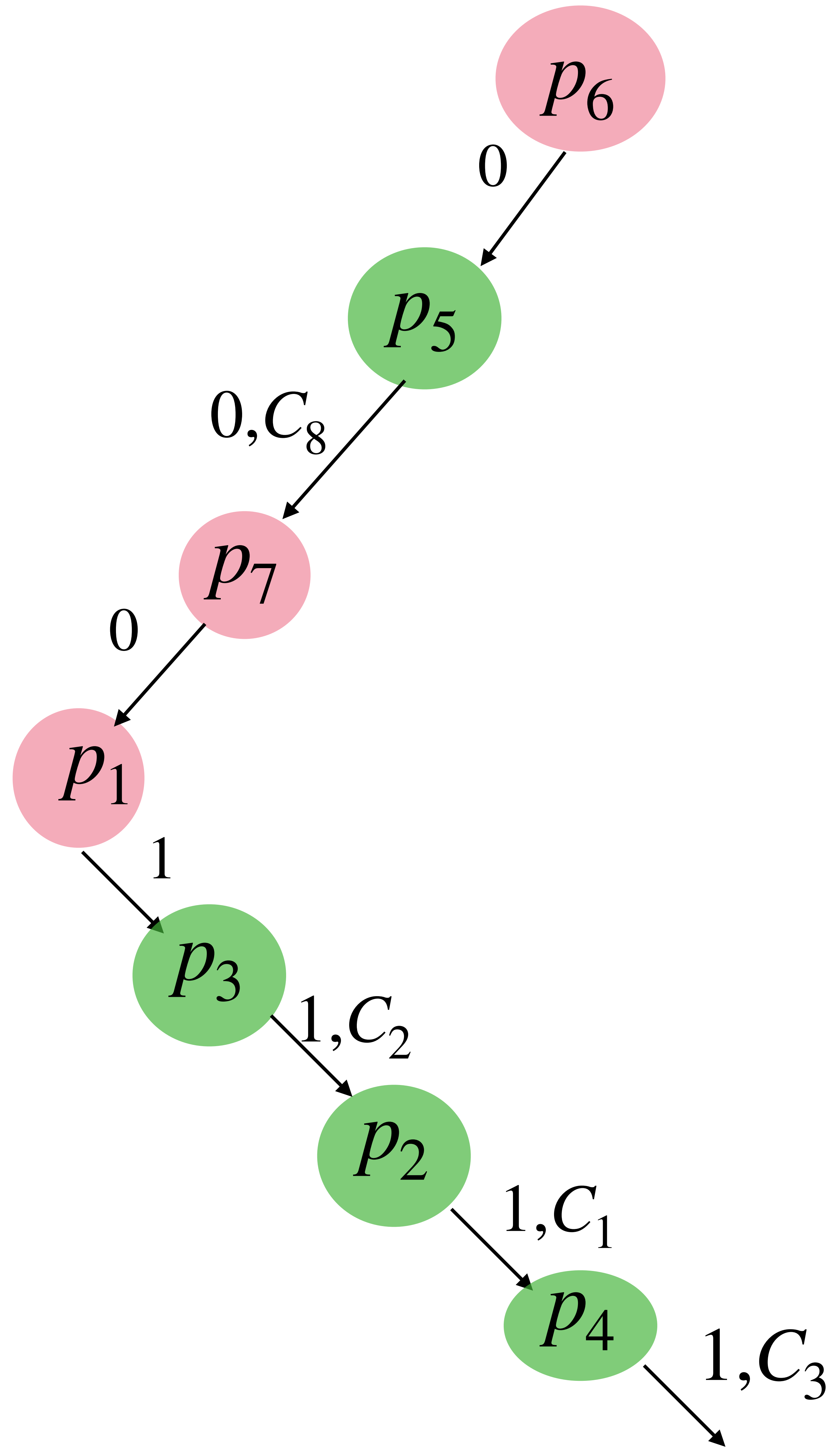
$C_7 = (p_2 \lor \neg p_3 \lor p_7)$

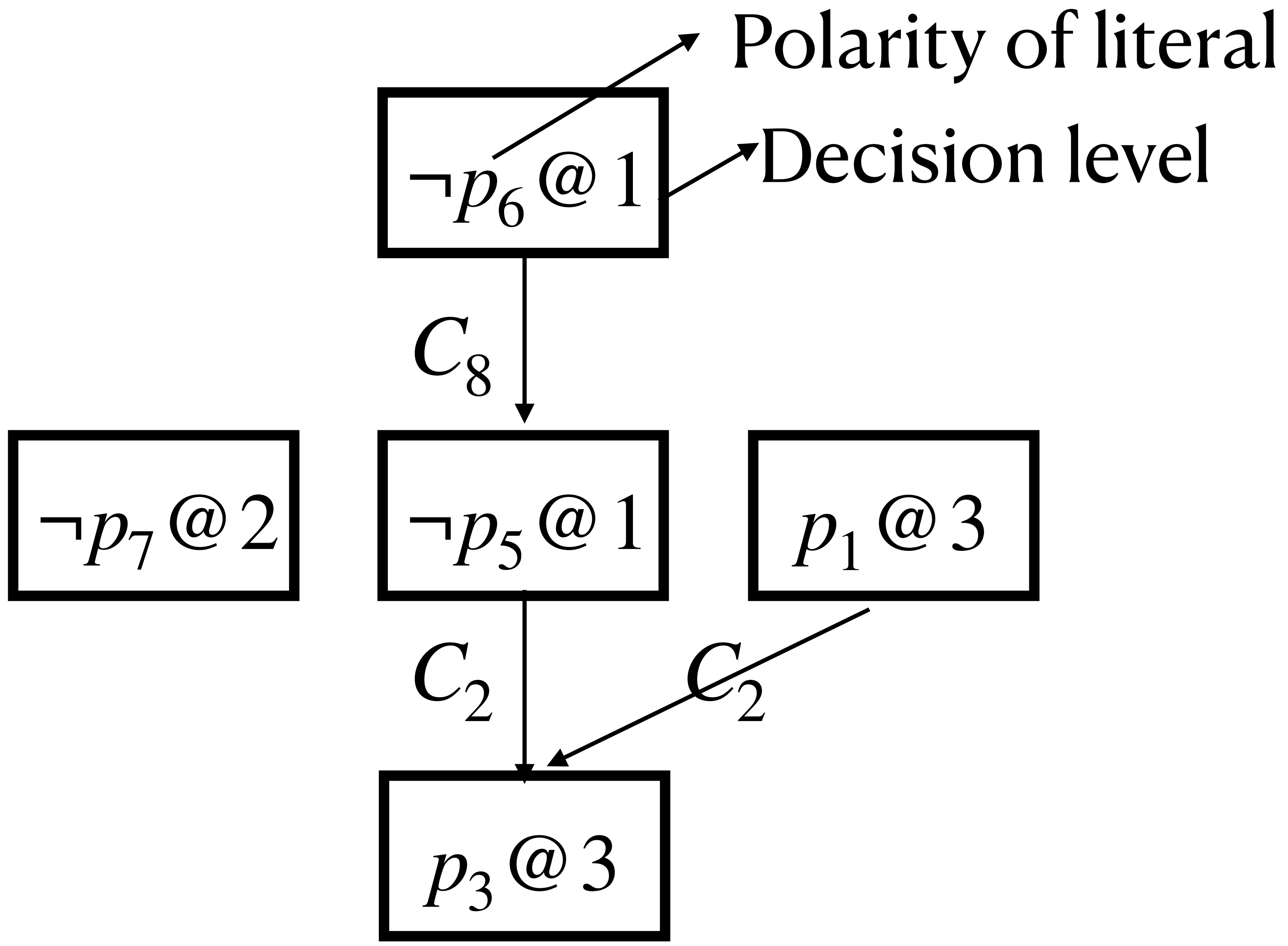
$C_8 = (p_6 \lor \neg p_5)$





$$C_1 = (\neg p_1 \vee p_2)$$
$$C_2 = (\neg p_1 \vee p_3 \vee p_5)$$
$$C_3 = (\neg p_2 \vee p_4)$$
$$C_4 = (\neg p_3 \vee \neg p_4)$$
$$C_5 = (p_1 \vee p_5 \vee \neg p_2)$$
$$C_6 = (p_2 \vee p_3)$$
$$C_7 = (p_2 \vee \neg p_3 \vee p_7)$$
$$C_8 = (p_6 \vee \neg p_5)$$





$C_1 = (\neg p_1 \vee p_2)$

$C_2 = (\neg p_1 \vee p_3 \vee p_5)$

$C_3 = (\neg p_2 \vee p_4)$

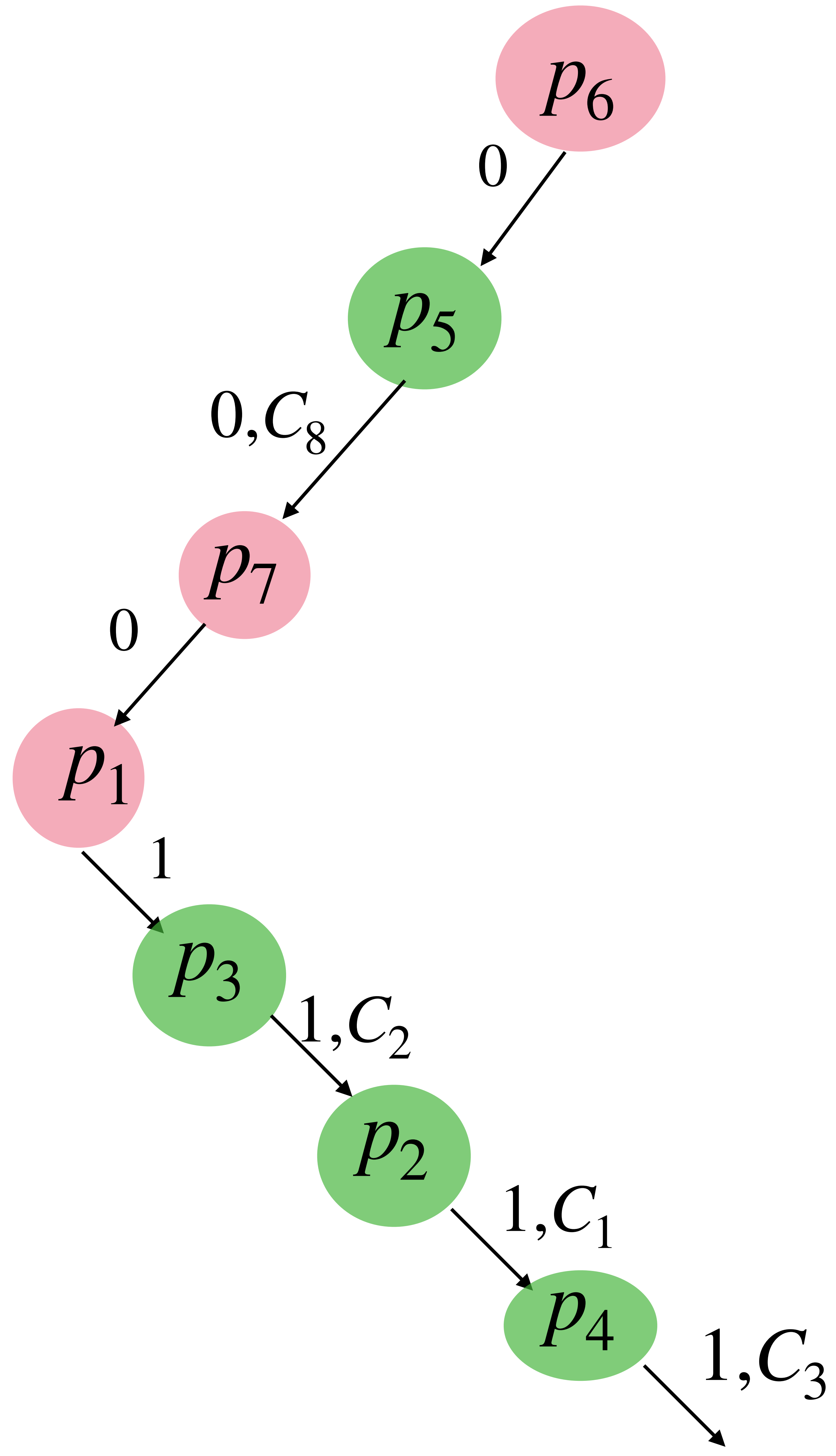
$C_4 = (\neg p_3 \vee \neg p_4)$

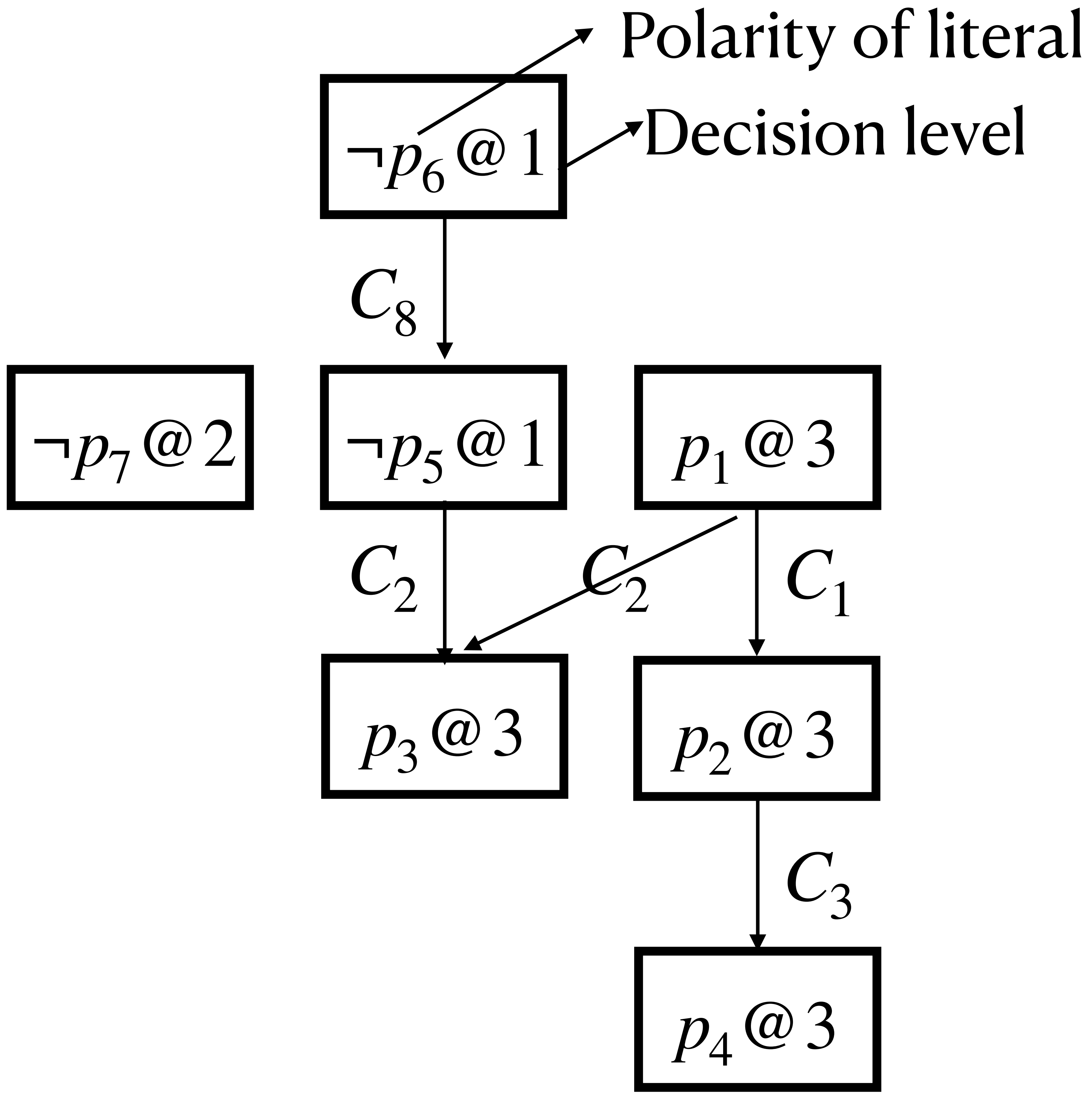
$C_5 = (p_1 \vee p_5 \vee \neg p_2)$

$C_6 = (p_2 \vee p_3)$

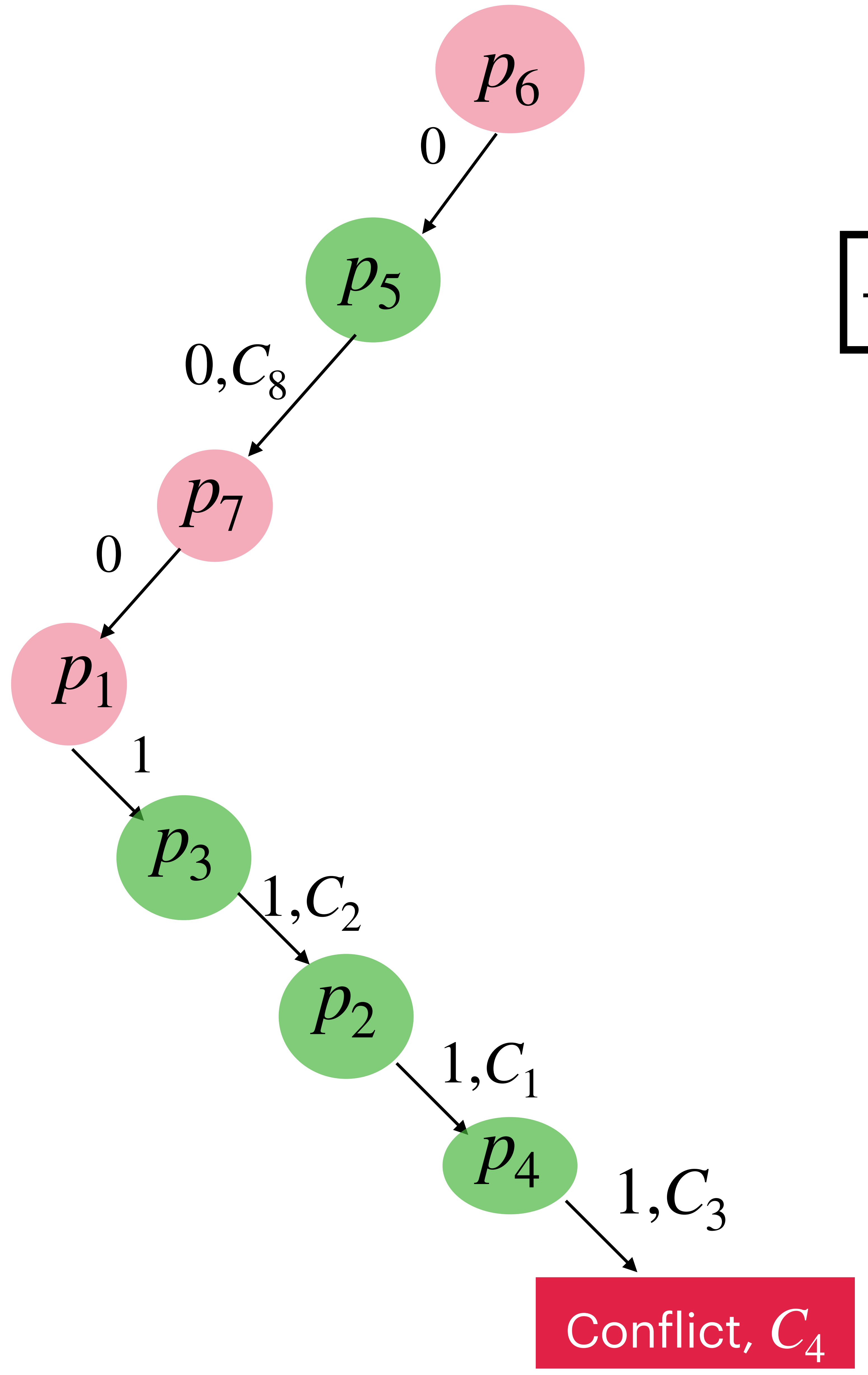
$C_7 = (p_2 \vee \neg p_3 \vee p_7)$

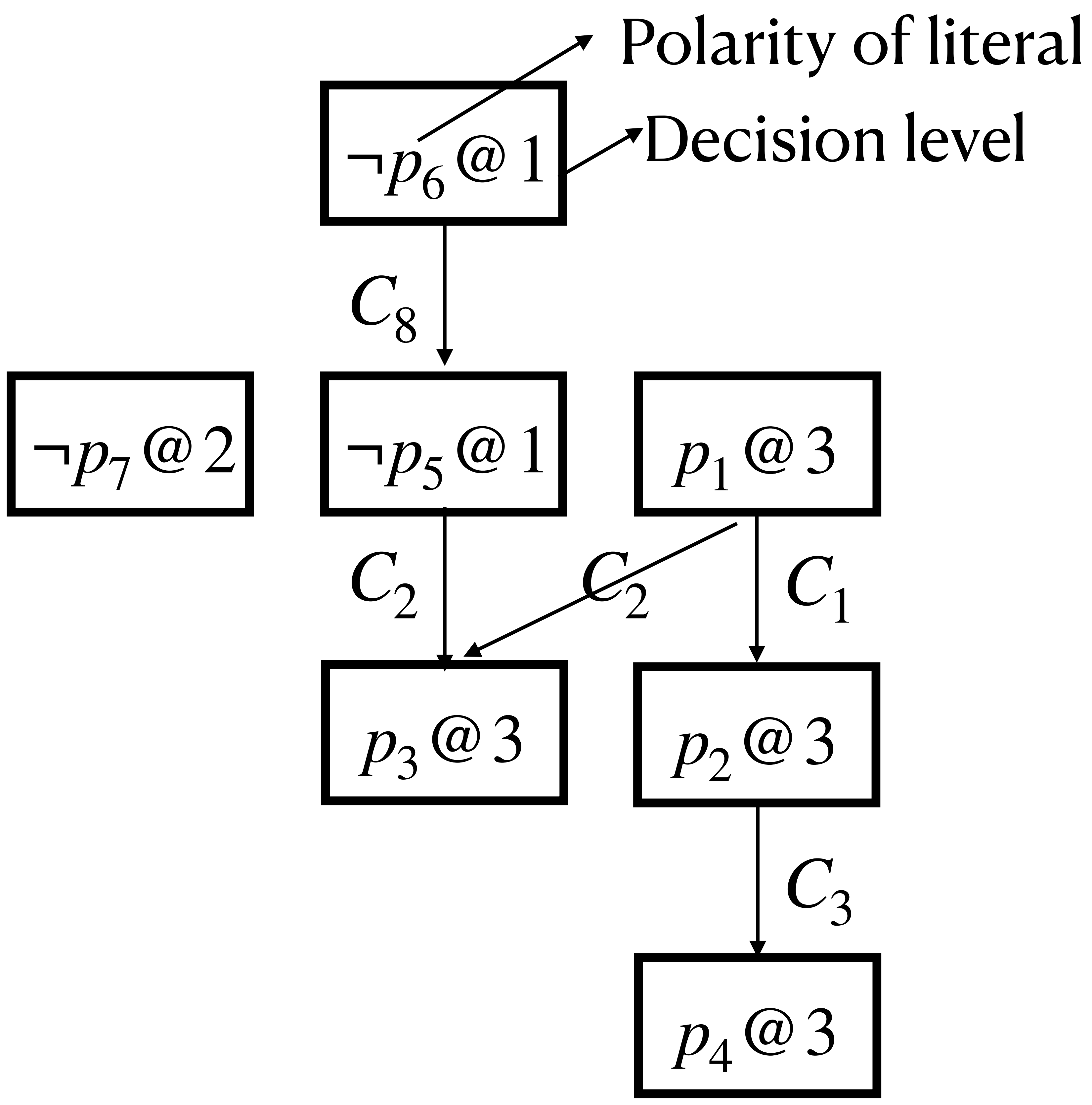
$C_8 = (p_6 \vee \neg p_5)$



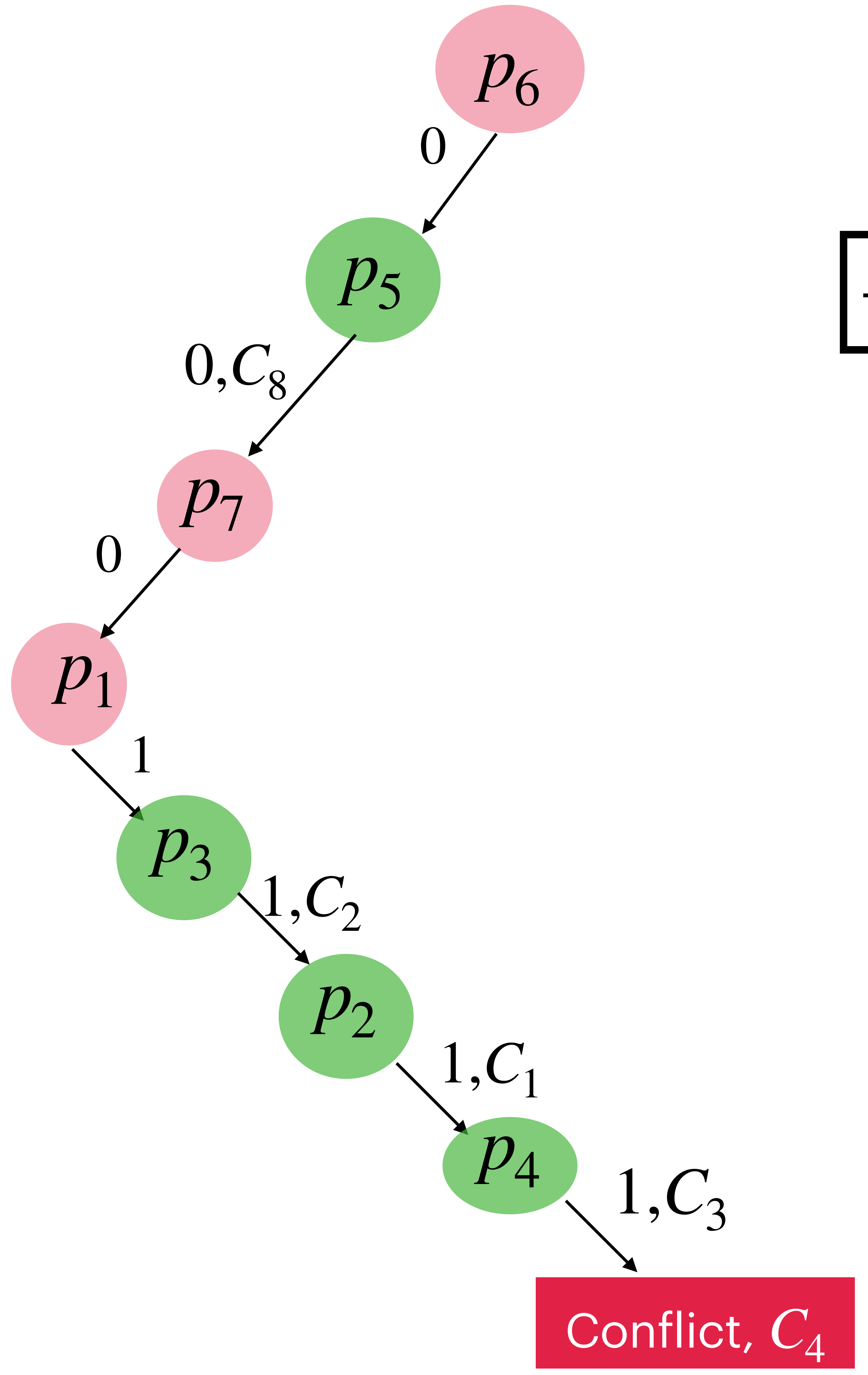


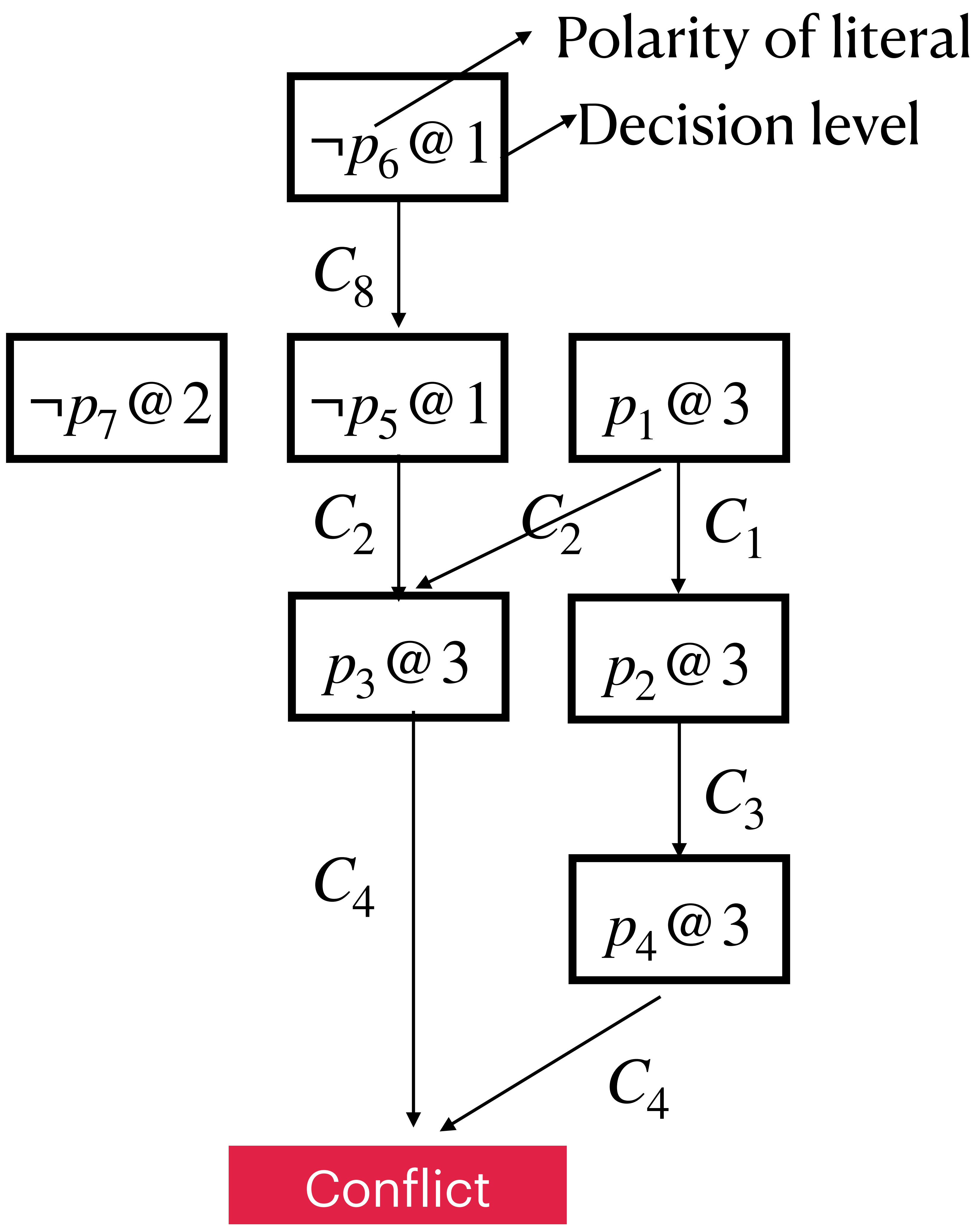
$C_1 = (\neg p_1 \vee p_2)$
$C_2 = (\neg p_1 \vee p_3 \vee p_5)$
$C_3 = (\neg p_2 \vee p_4)$
$C_4 = (\neg p_3 \vee \neg p_4)$
$C_5 = (p_1 \vee p_5 \vee \neg p_2)$
$C_6 = (p_2 \vee p_3)$
$C_7 = (p_2 \vee \neg p_3 \vee p_7)$
$C_8 = (p_6 \vee \neg p_5)$



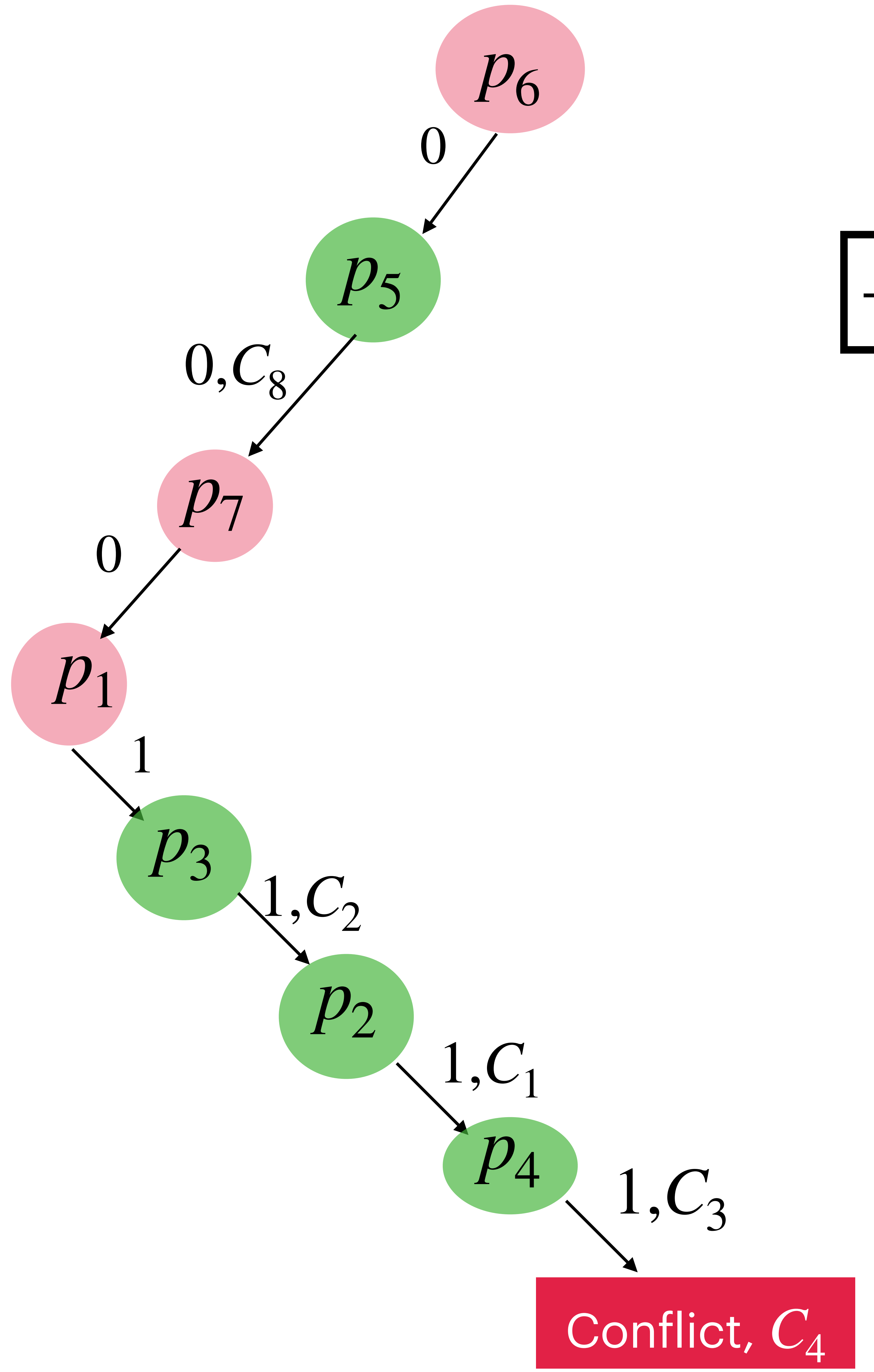


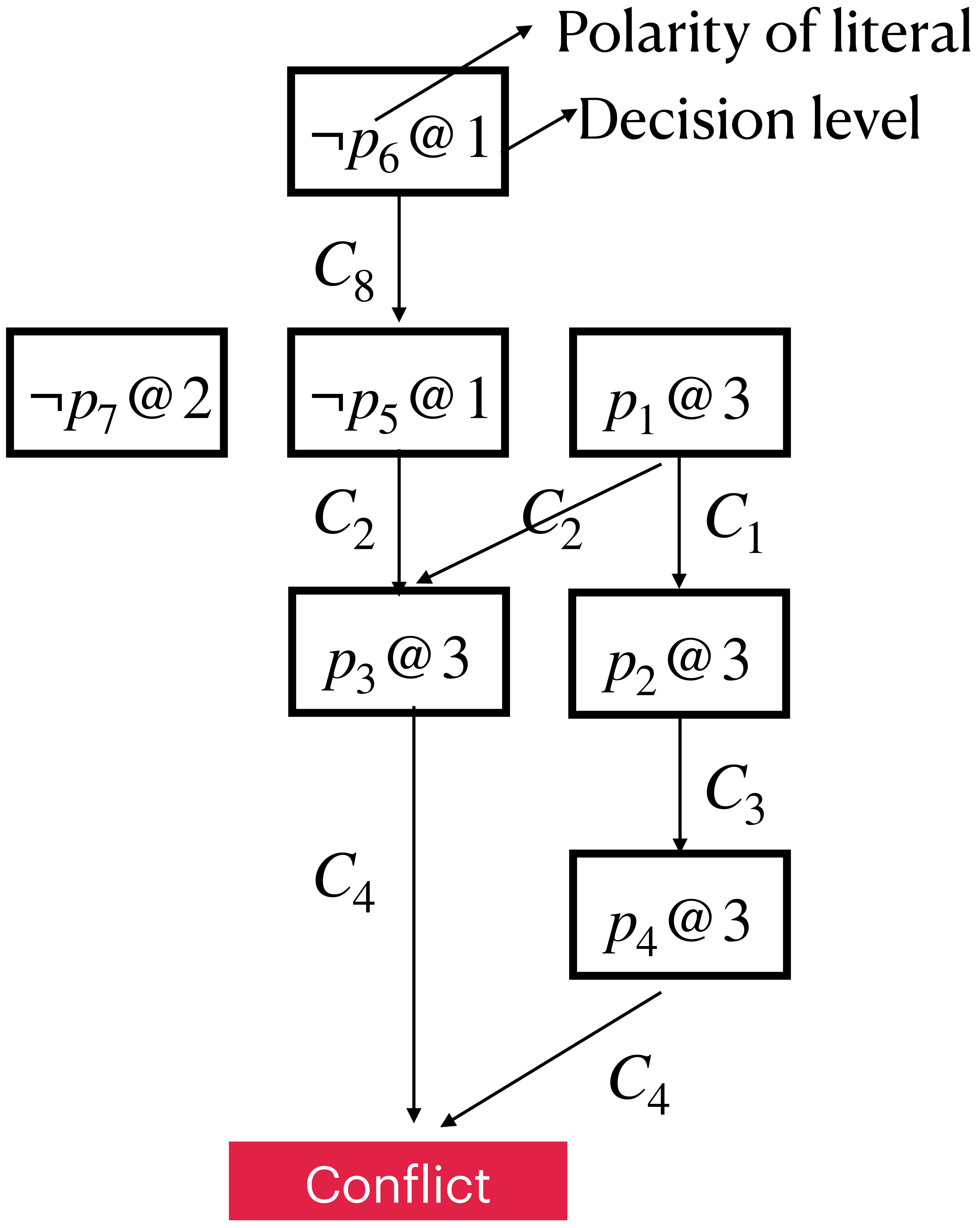
$$C_1 = (\neg p_1 \vee p_2)$$
$$C_2 = (\neg p_1 \vee p_3 \vee p_5)$$
$$C_3 = (\neg p_2 \vee p_4)$$
$$C_4 = (\neg p_3 \vee \neg p_4)$$
$$C_5 = (p_1 \vee p_5 \vee \neg p_2)$$
$$C_6 = (p_2 \vee p_3)$$
$$C_7 = (p_2 \vee \neg p_3 \vee p_7)$$
$$C_8 = (p_6 \vee \neg p_5)$$





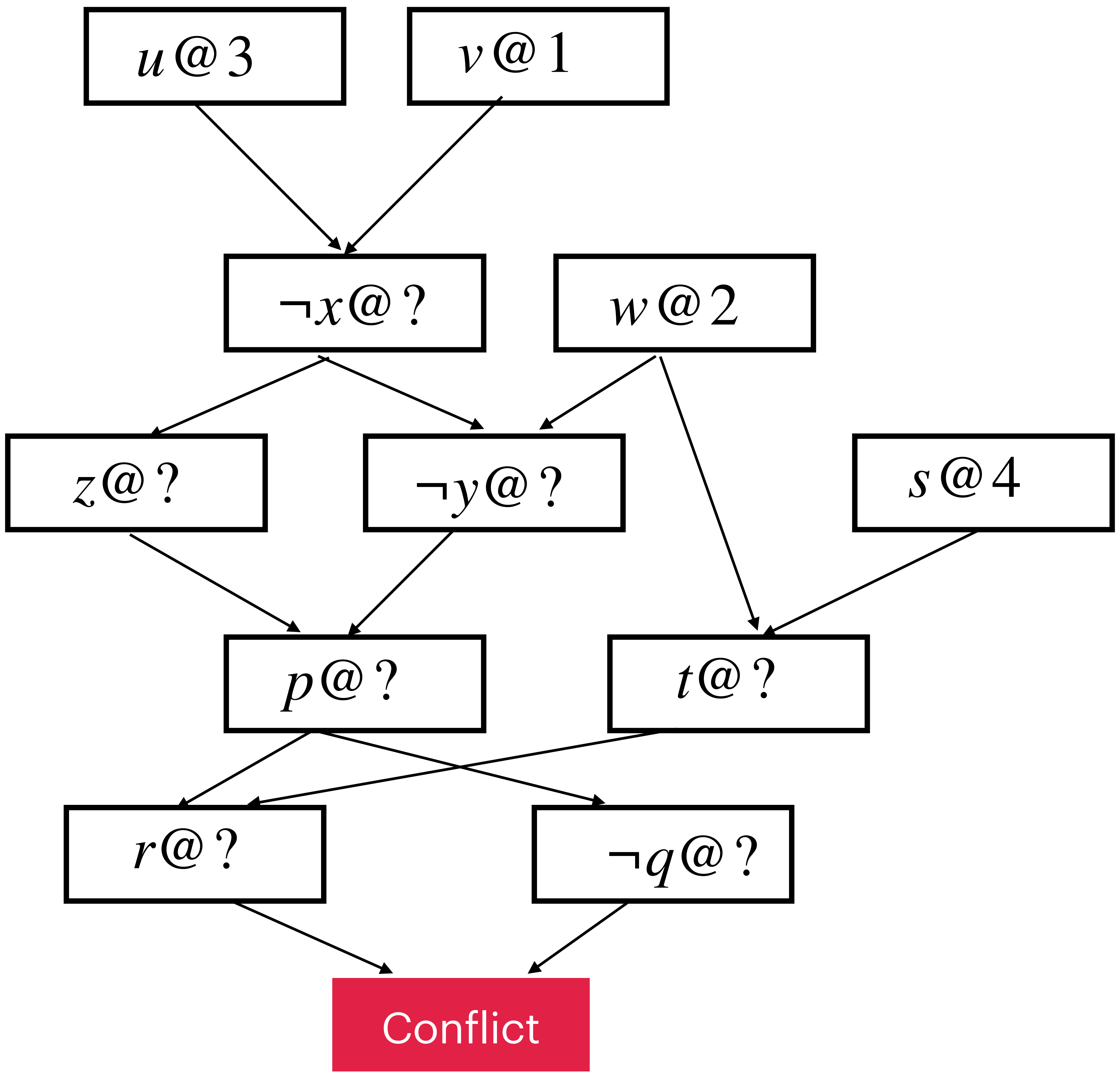
$$C_1 = (\neg p_1 \lor p_2)$$
$$C_2 = (\neg p_1 \lor p_3 \lor p_5)$$
$$C_3 = (\neg p_2 \lor p_4)$$
$$C_4 = (\neg p_3 \lor \neg p_4)$$
$$C_5 = (p_1 \lor p_5 \lor \neg p_2)$$
$$C_6 = (p_2 \lor p_3)$$
$$C_7 = (p_2 \lor \neg p_3 \lor p_7)$$
$$C_8 = (p_6 \lor \neg p_5)$$



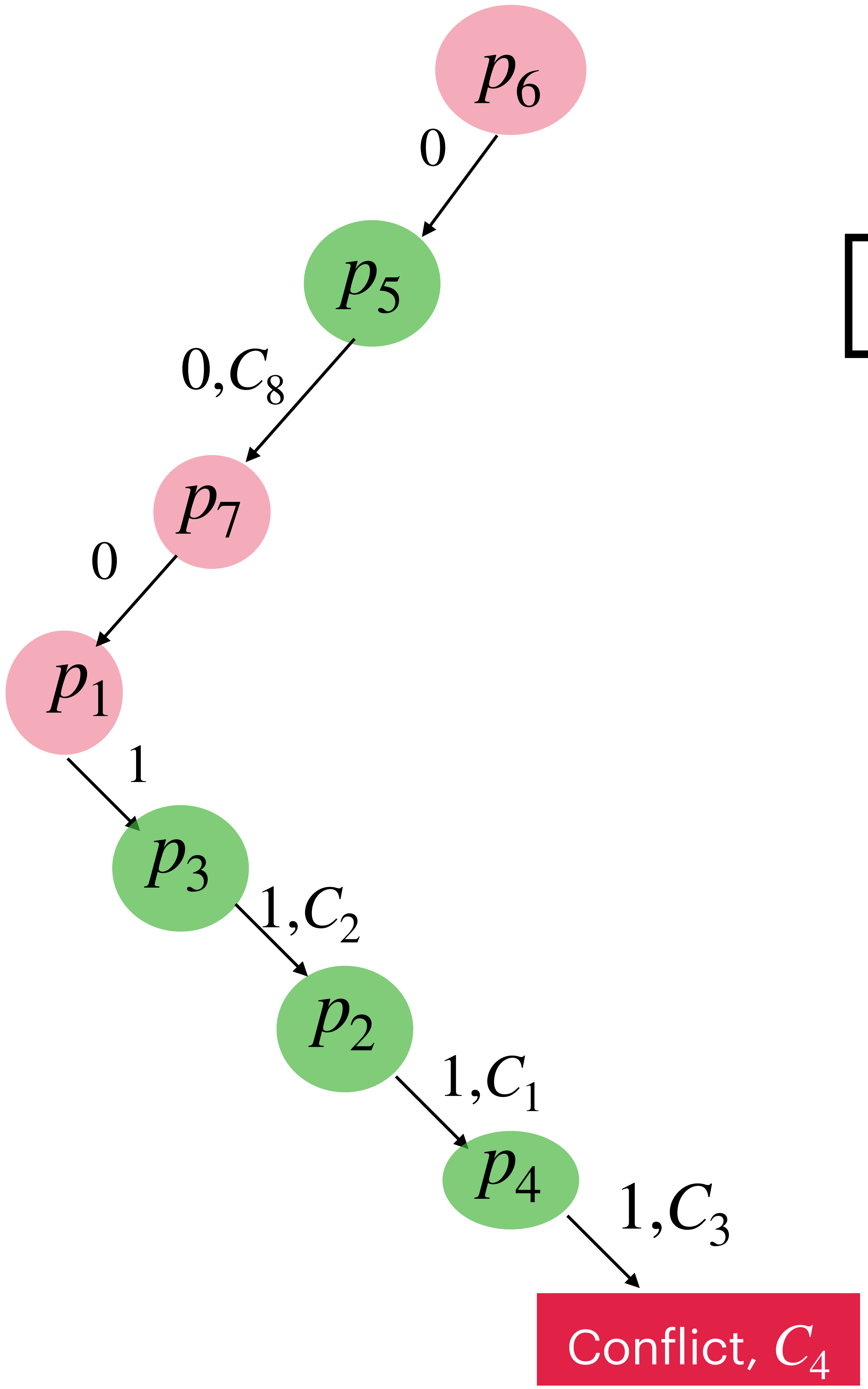


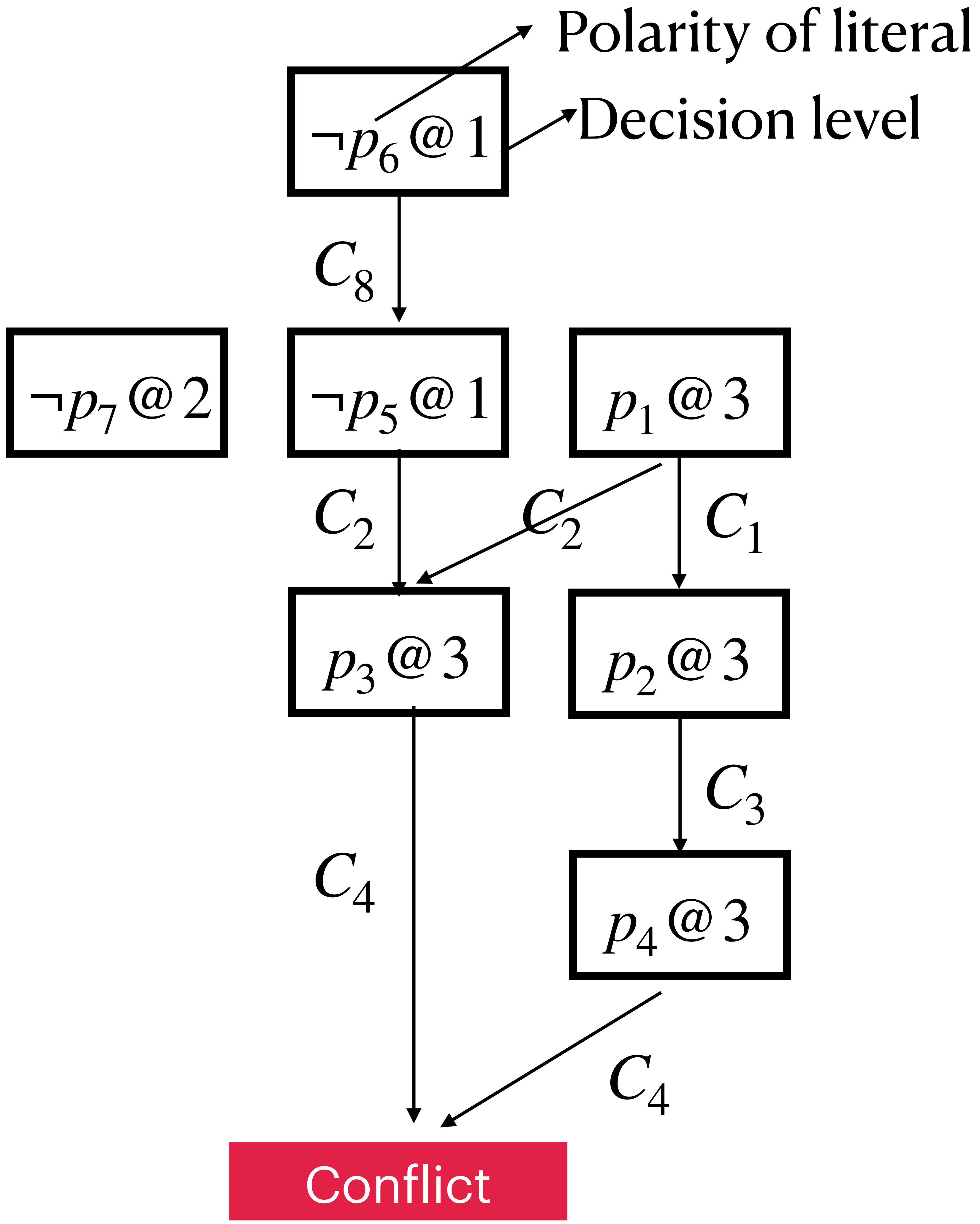
Implication Graph.

Assign decision level at every node in implication graph



$C_1 = (\neg p_1 \vee p_2)$
$C_2 = (\neg p_1 \vee p_3 \vee p_5)$
$C_3 = (\neg p_2 \vee p_4)$
$C_4 = (\neg p_3 \vee \neg p_4)$
$C_5 = (p_1 \vee p_5 \vee \neg p_2)$
$C_6 = (p_2 \vee p_3)$
$C_7 = (p_2 \vee \neg p_3 \vee p_7)$
$C_8 = (p_6 \vee \neg p_5)$

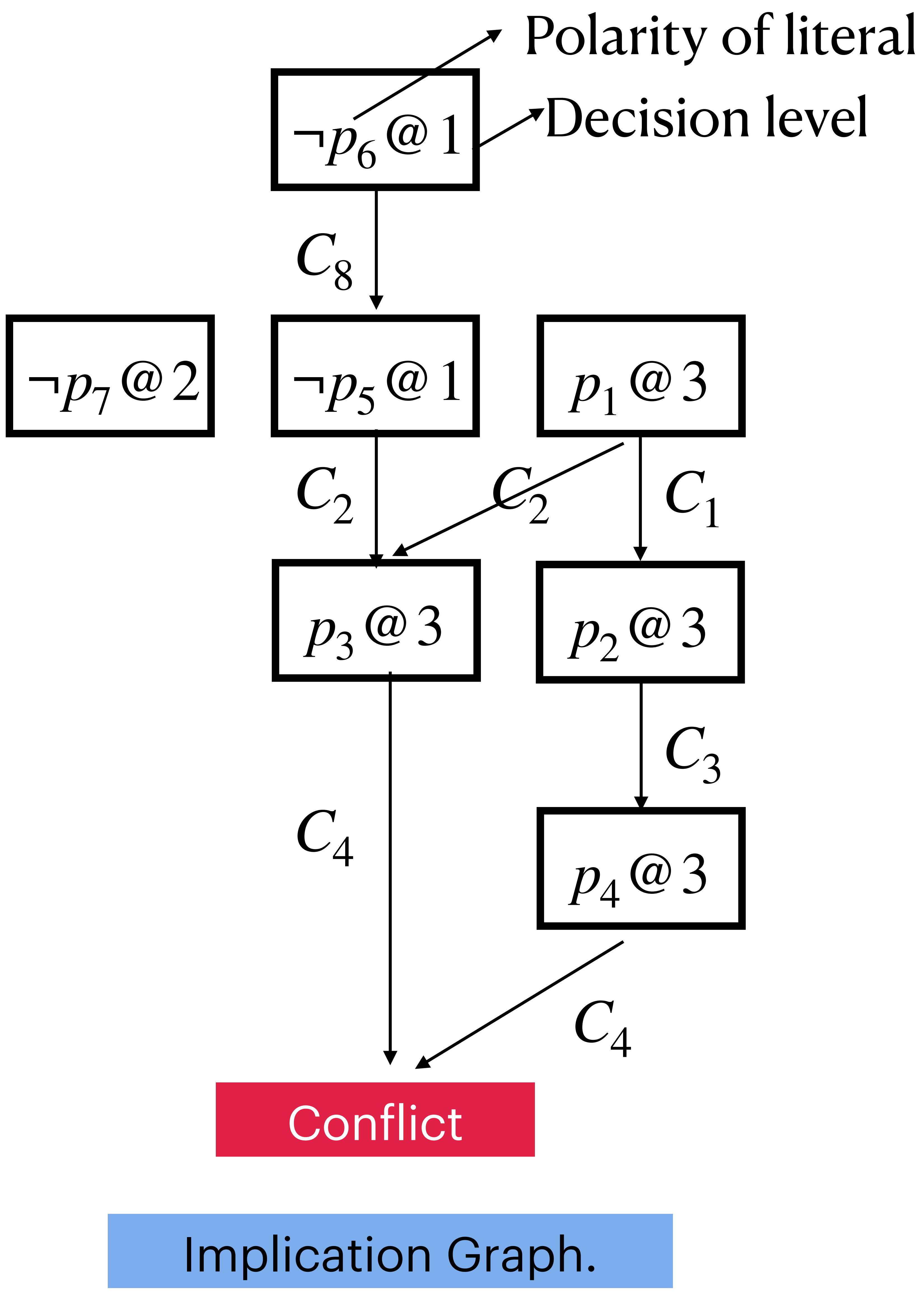




Implication Graph.

Conflict Clause

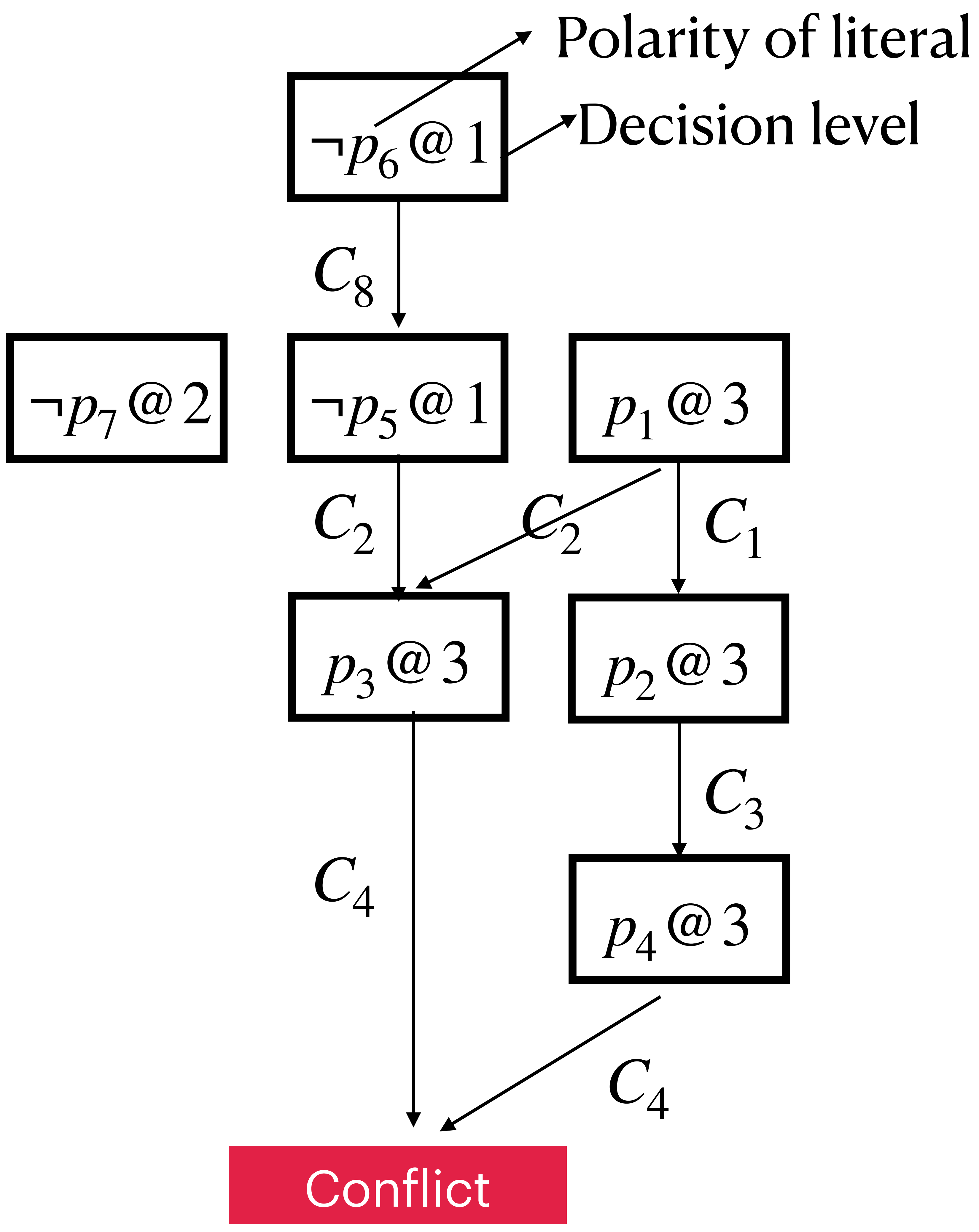
The clause of the negations of the causing decisions is called conflict clause.



Conflict Clause

The clause of the negations of the causing decisions is called conflict clause.

Mistake: $p_6 = 0$ and $p_1 = 1$



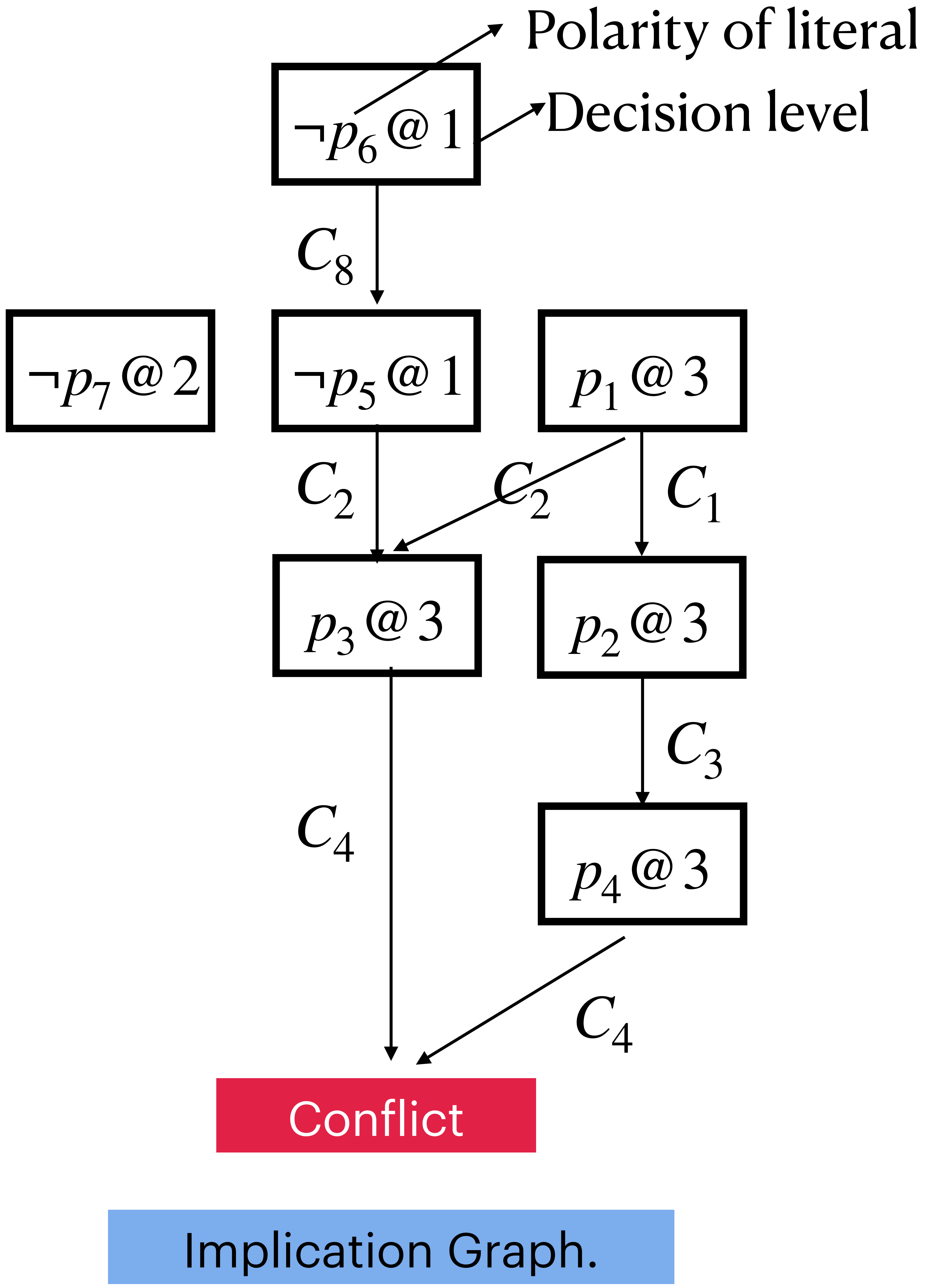
Implication Graph.

Conflict Clause

The clause of the negations of the causing decisions is called conflict clause.

Mistake: $p_6 = 0$ and $p_1 = 1$

Conflict clause : $\neg(\neg p_6 \wedge p_1) \equiv p_6 \vee \neg p_1$



Conflict Clause

The clause of the negations of the causing decisions is called conflict clause.

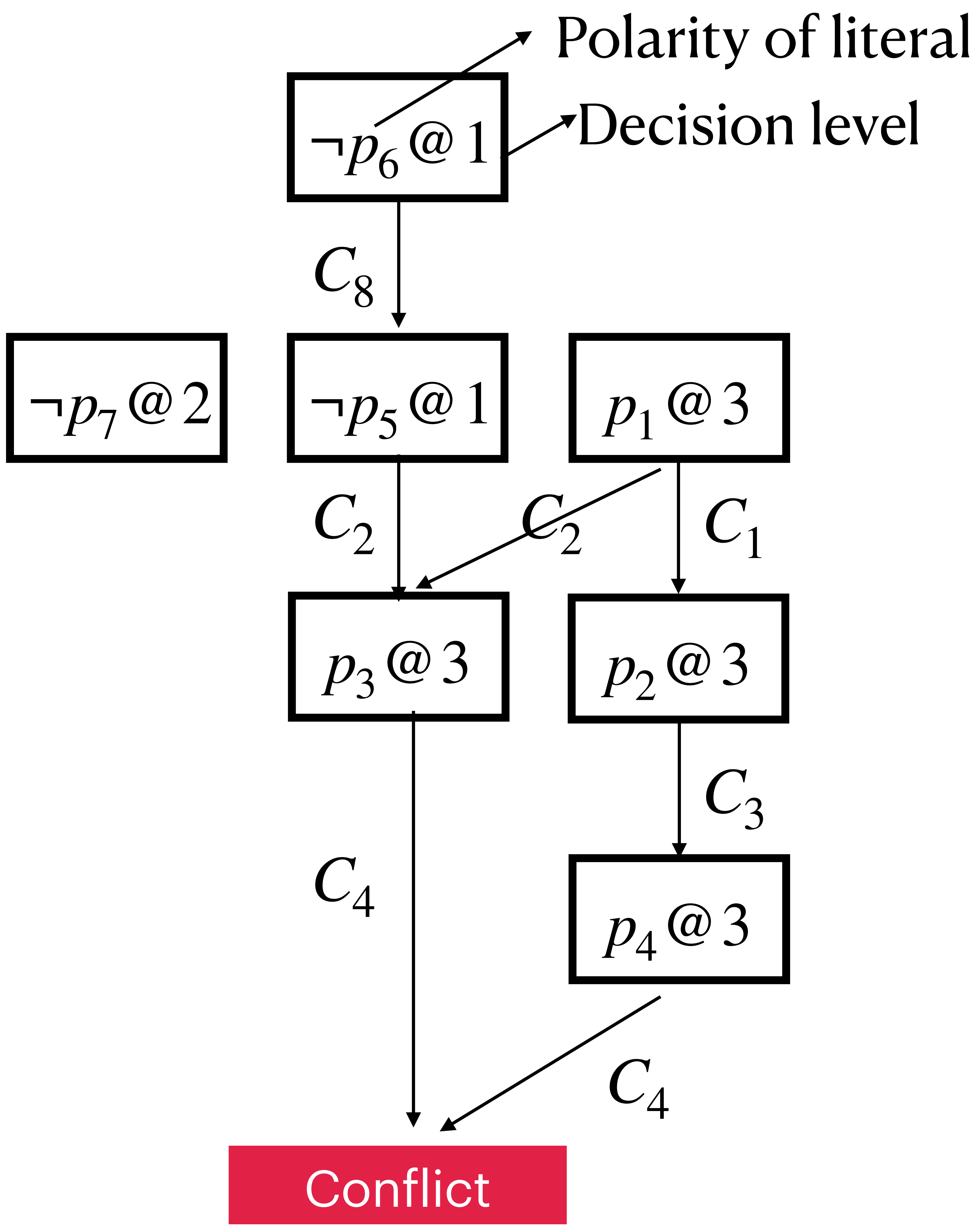
Mistake: $p_6 = 0$ and $p_1 = 1$

Conflict clause : $\neg(\neg p_6 \wedge p_1) \equiv p_6 \vee \neg p_1$

$m(p_6) = 0, m(p_7) = 1, m(p_1) = 1$

$m(p_6) = 0, m(p_1) = 1$

This will never be tried again!



Implication Graph.

Conflict Clause

The clause of the negations of the causing decisions is called conflict clause.

Mistake: $p_6 = 0$ and $p_1 = 1$

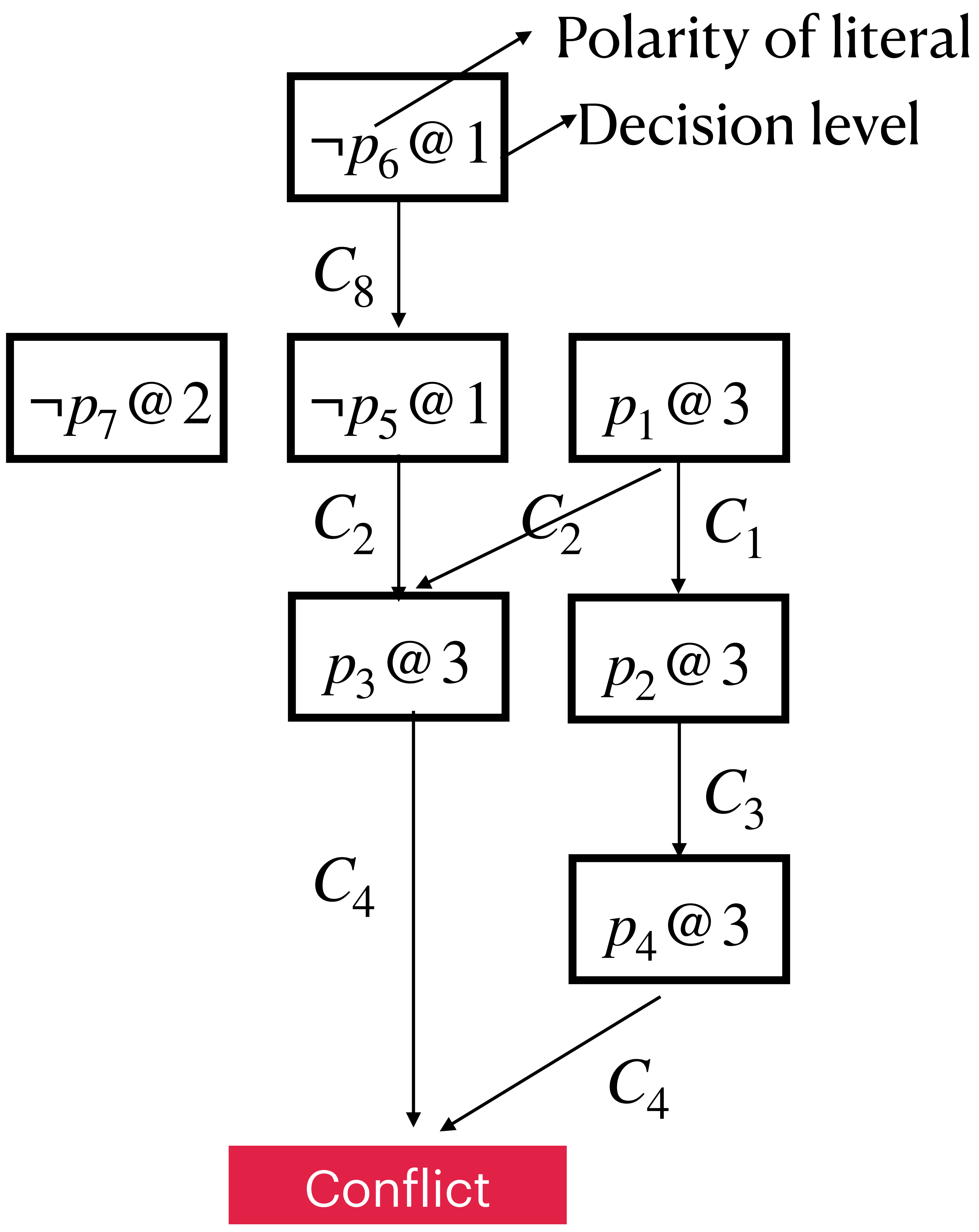
Conflict clause : $\neg(\neg p_6 \wedge p_1) \equiv p_6 \vee \neg p_1$

$m(p_6) = 0, m(p_7) = 1, m(p_1) = 1$

$m(p_6) = 0, m(p_1) = 1$

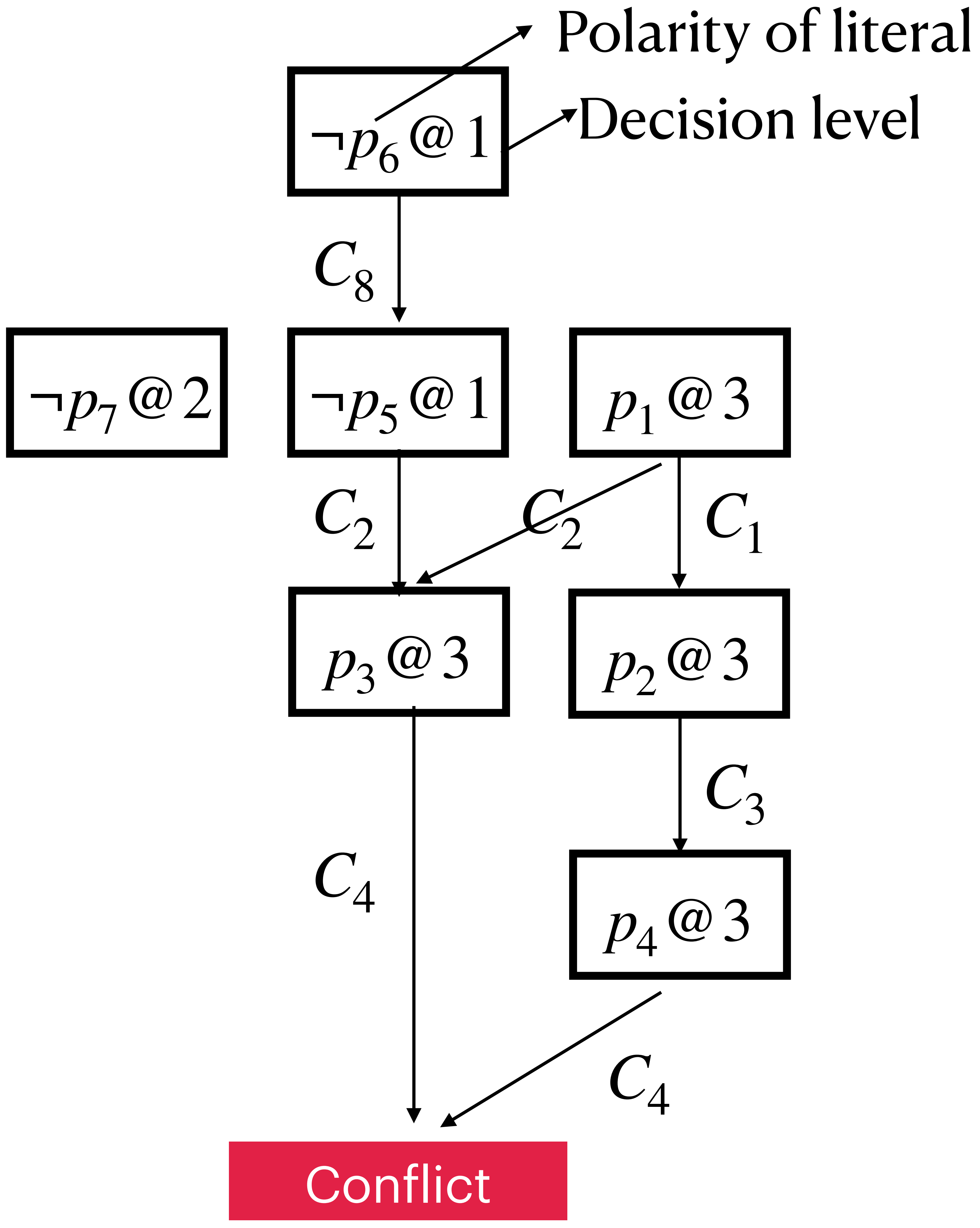
This will never be tried again!

CDCL: Conflict Driven Clause Learning



Implication Graph.

$$C_1 = (\neg p_1 \vee p_2)$$
$$C_2 = (\neg p_1 \vee p_3 \vee p_5)$$
$$C_3 = (\neg p_2 \vee p_4)$$
$$C_4 = (\neg p_3 \vee \neg p_4)$$
$$C_5 = (p_1 \vee p_5 \vee \neg p_2)$$
$$C_6 = (p_2 \vee p_3)$$
$$C_7 = (p_2 \vee \neg p_3 \vee p_7)$$
$$C_8 = (p_6 \vee \neg p_5)$$
$$C_9 = (p_6 \vee \neg p_1)$$



Implication Graph.

$C_1 = (\neg p_1 \vee p_2)$

$C_2 = (\neg p_1 \vee p_3 \vee p_5)$

$C_3 = (\neg p_2 \vee p_4)$

$C_4 = (\neg p_3 \vee \neg p_4)$

$C_5 = (p_1 \vee p_5 \vee \neg p_2)$

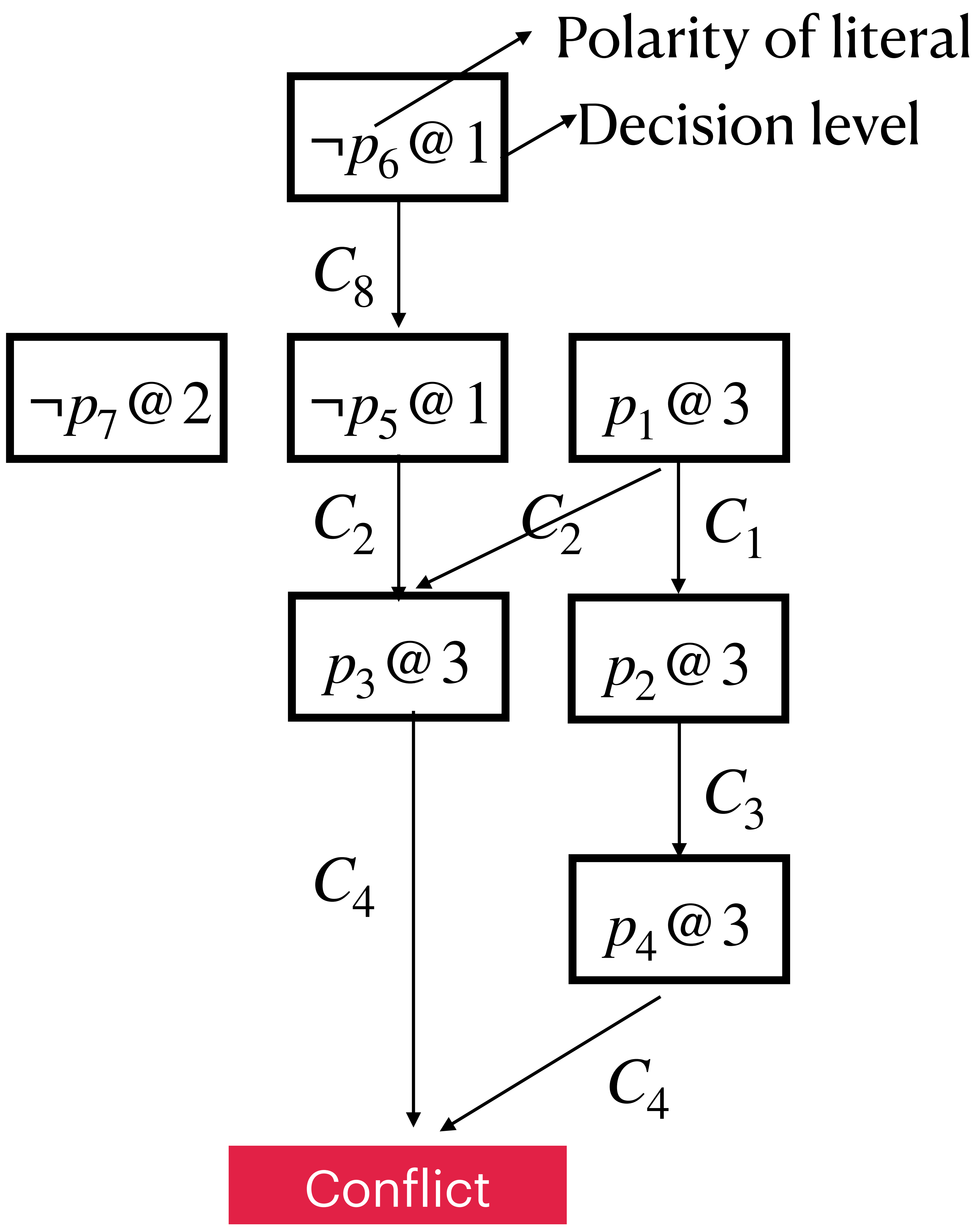
$C_6 = (p_2 \vee p_3)$

$C_7 = (p_2 \vee \neg p_3 \vee p_7)$

$C_8 = (p_6 \vee \neg p_5)$

$C_9 = (p_6 \vee \neg p_1)$

Added a new clause!
Where should we backtrack?



Implication Graph.

$C_1 = (\neg p_1 \lor p_2)$

$C_2 = (\neg p_1 \lor p_3 \lor p_5)$

$C_3 = (\neg p_2 \lor p_4)$

$C_4 = (\neg p_3 \lor \neg p_4)$

$C_5 = (p_1 \lor p_5 \lor \neg p_2)$

$C_6 = (p_2 \lor p_3)$

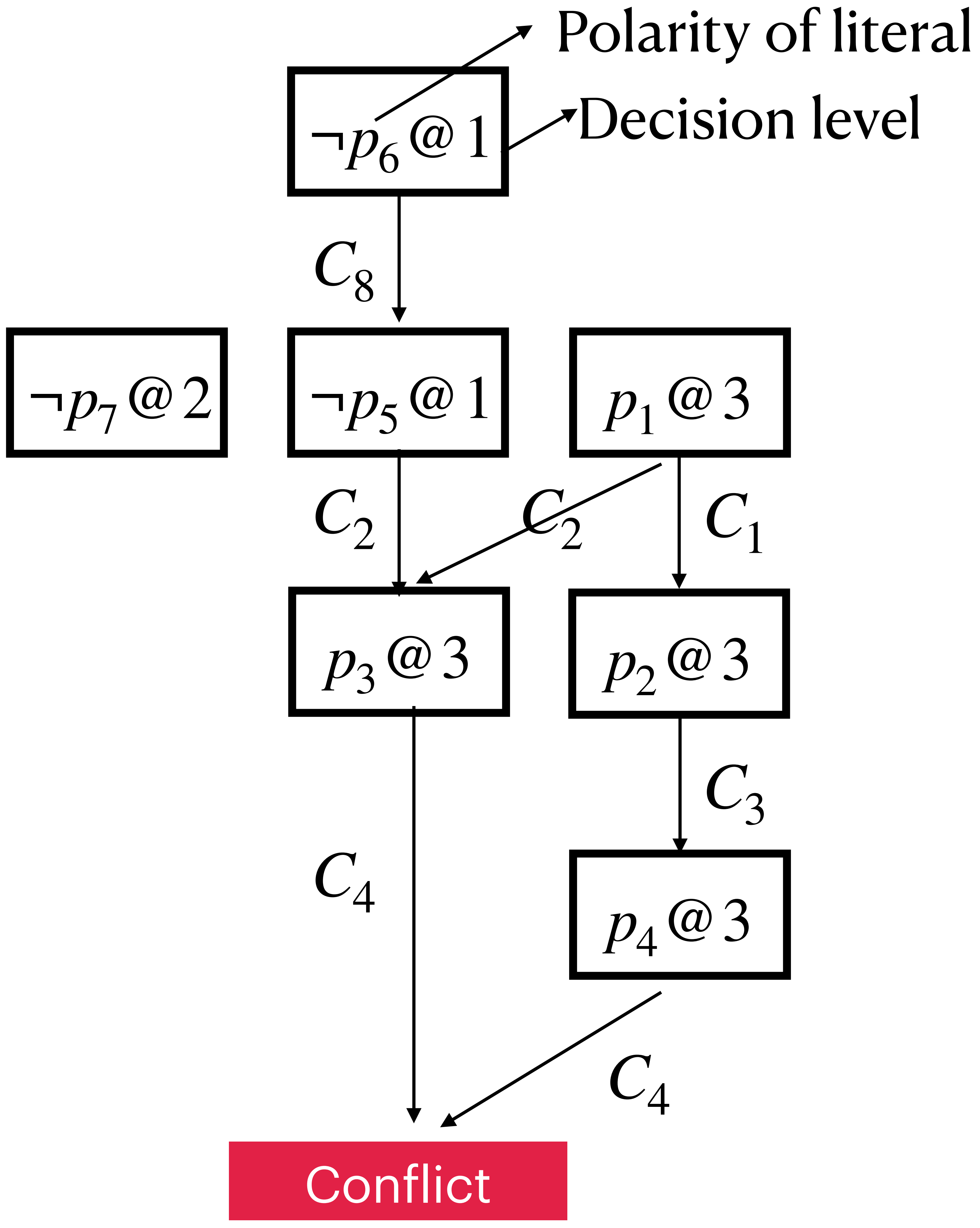
$C_7 = (p_2 \lor \neg p_3 \lor p_7)$

$C_8 = (p_6 \lor \neg p_5)$

$C_9 = (p_6 \lor \neg p_1)$

Added a new clause!
Where should we backtrack?

Backtrack to second largest decision in the conflict clause.



Implication Graph.

$C_1 = (\neg p_1 \vee p_2)$

$C_2 = (\neg p_1 \vee p_3 \vee p_5)$

$C_3 = (\neg p_2 \vee p_4)$

$C_4 = (\neg p_3 \vee \neg p_4)$

$C_5 = (p_1 \vee p_5 \vee \neg p_2)$

$C_6 = (p_2 \vee p_3)$

$C_7 = (p_2 \vee \neg p_3 \vee p_7)$

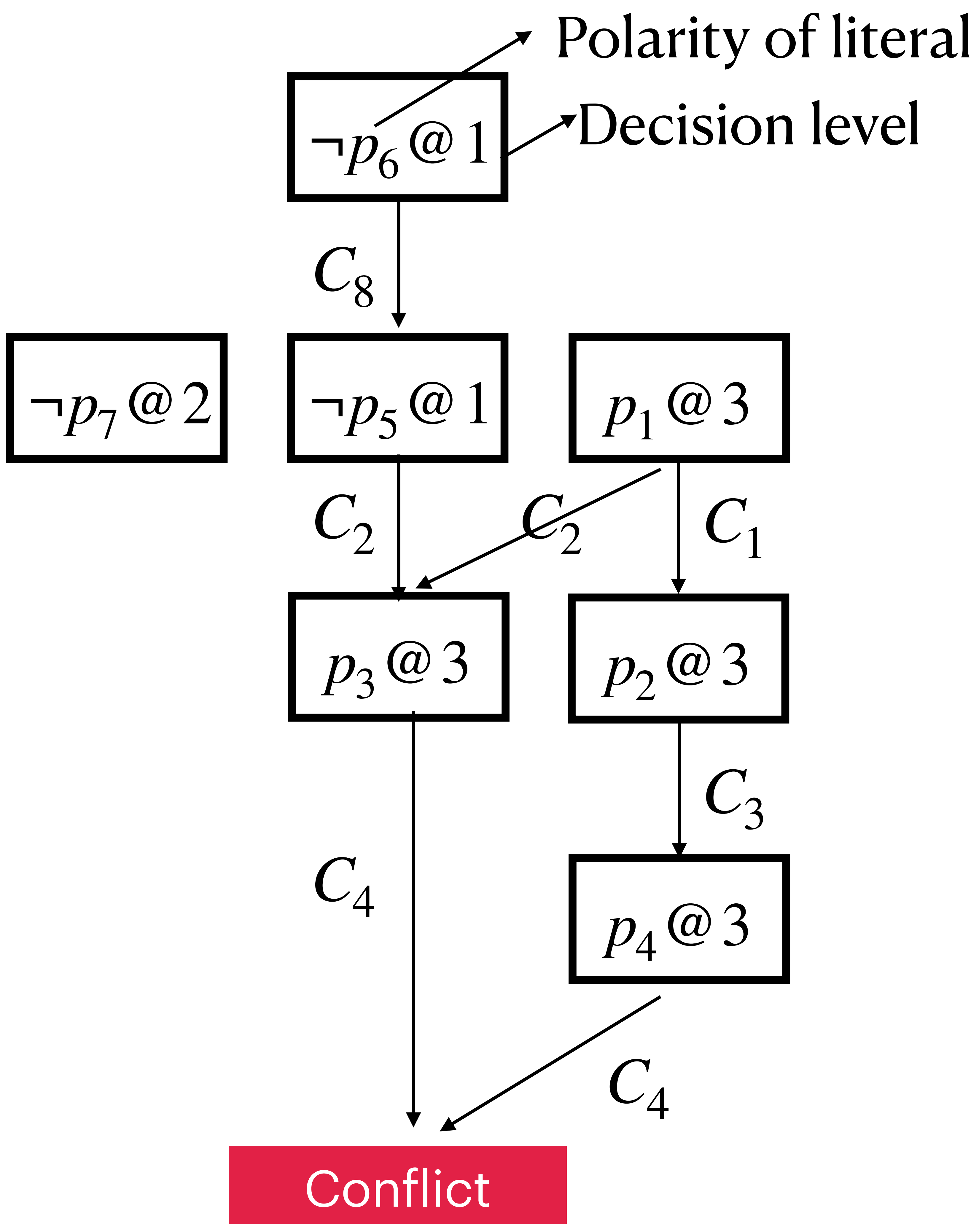
$C_8 = (p_6 \vee \neg p_5)$

$C_9 = (p_6 \vee \neg p_1)$

Added a new clause!
Where should we backtrack?

Backtrack to second largest decision in the conflict clause.

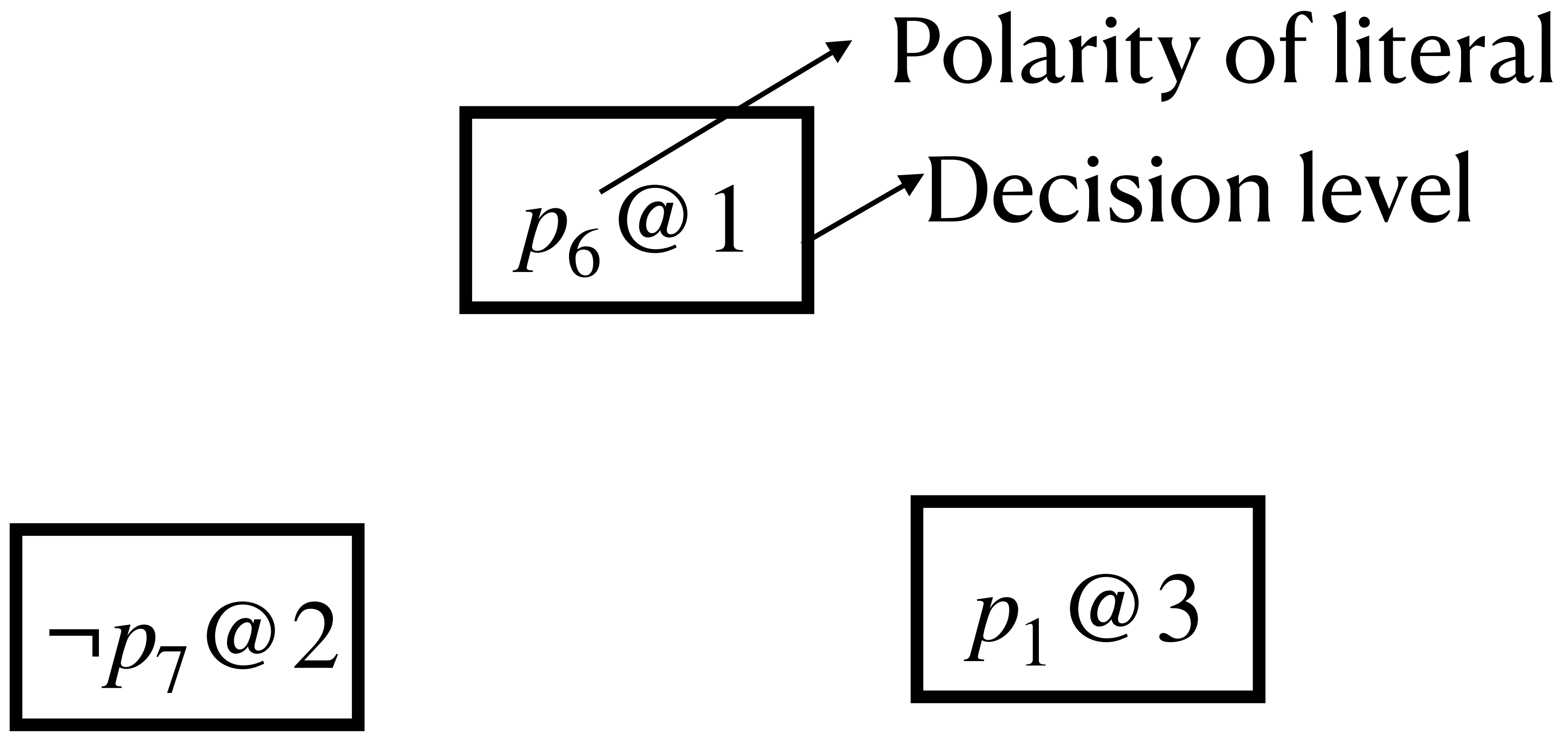
Here we should backtrack to decision level 1.



Implication Graph.

$$C_1 = (\neg p_1 \lor p_2)$$
$$C_2 = (\neg p_1 \lor p_3 \lor p_5)$$
$$C_3 = (\neg p_2 \lor p_4)$$
$$C_4 = (\neg p_3 \lor \neg p_4)$$
$$C_5 = (p_1 \lor p_5 \lor \neg p_2)$$
$$C_6 = (p_2 \lor p_3)$$
$$C_7 = (p_2 \lor \neg p_3 \lor p_7)$$
$$C_8 = (p_6 \lor \neg p_5)$$
$$C_9 = (p_6 \lor \neg p_1)$$

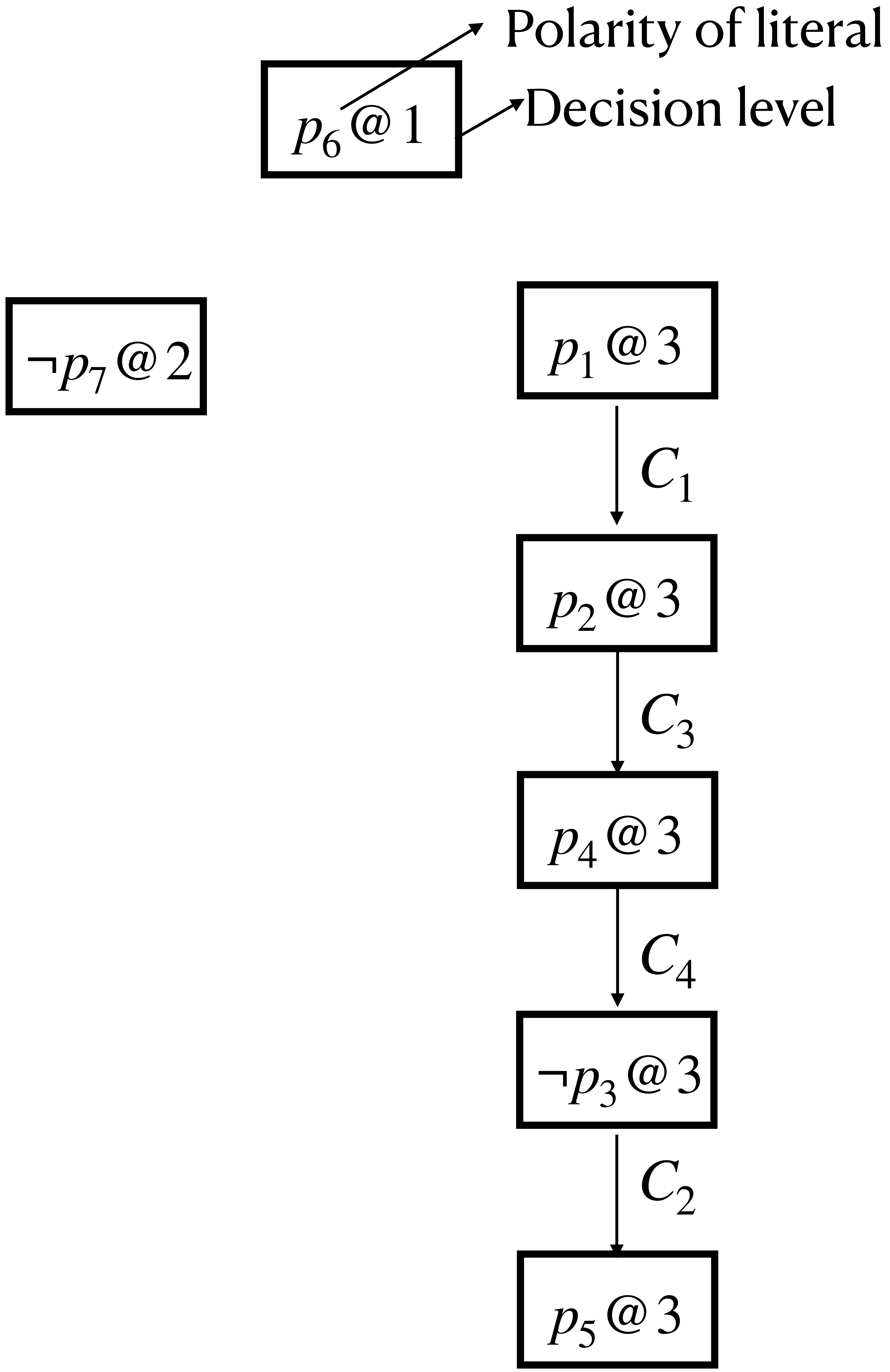
Here we should backtrack to decision level 1.



Implication Graph.

$$C_1 = (\neg p_1 \vee p_2)$$
$$C_2 = (\neg p_1 \vee p_3 \vee p_5)$$
$$C_3 = (\neg p_2 \vee p_4)$$
$$C_4 = (\neg p_3 \vee \neg p_4)$$
$$C_5 = (p_1 \vee p_5 \vee \neg p_2)$$
$$C_6 = (p_2 \vee p_3)$$
$$C_7 = (p_2 \vee \neg p_3 \vee p_7)$$
$$C_8 = (p_6 \vee \neg p_5)$$
$$C_9 = (p_6 \vee \neg p_1)$$

Here we should backtrack to decision level 1.



Implication Graph.

CDCL: Conflict Driven Clause Learning

1. UnitPropagation(m, F): applies unit propagation and extends m.

2. Decide(m, F): choose an unassigned variable in m and assign it a Boolean value.

3. ClauseLearning(m, F): returns a conflict clause learned using implication graph, and a decision level upto which the solver needs to backtrack.

Course Webpage

Thanks!