A Gentle Introduction to Modern SAT Solving — Part 2

Noah Fleming
University of California, San Diego

Last Time — Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

Derive new clauses from old using:

→ **Resolution rule:**

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

Goal: derive empty clause Λ

Resolution rule is **sound**

$\Longrightarrow$ Derivation of Λ certifies unsatisfiability

Last Time — Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

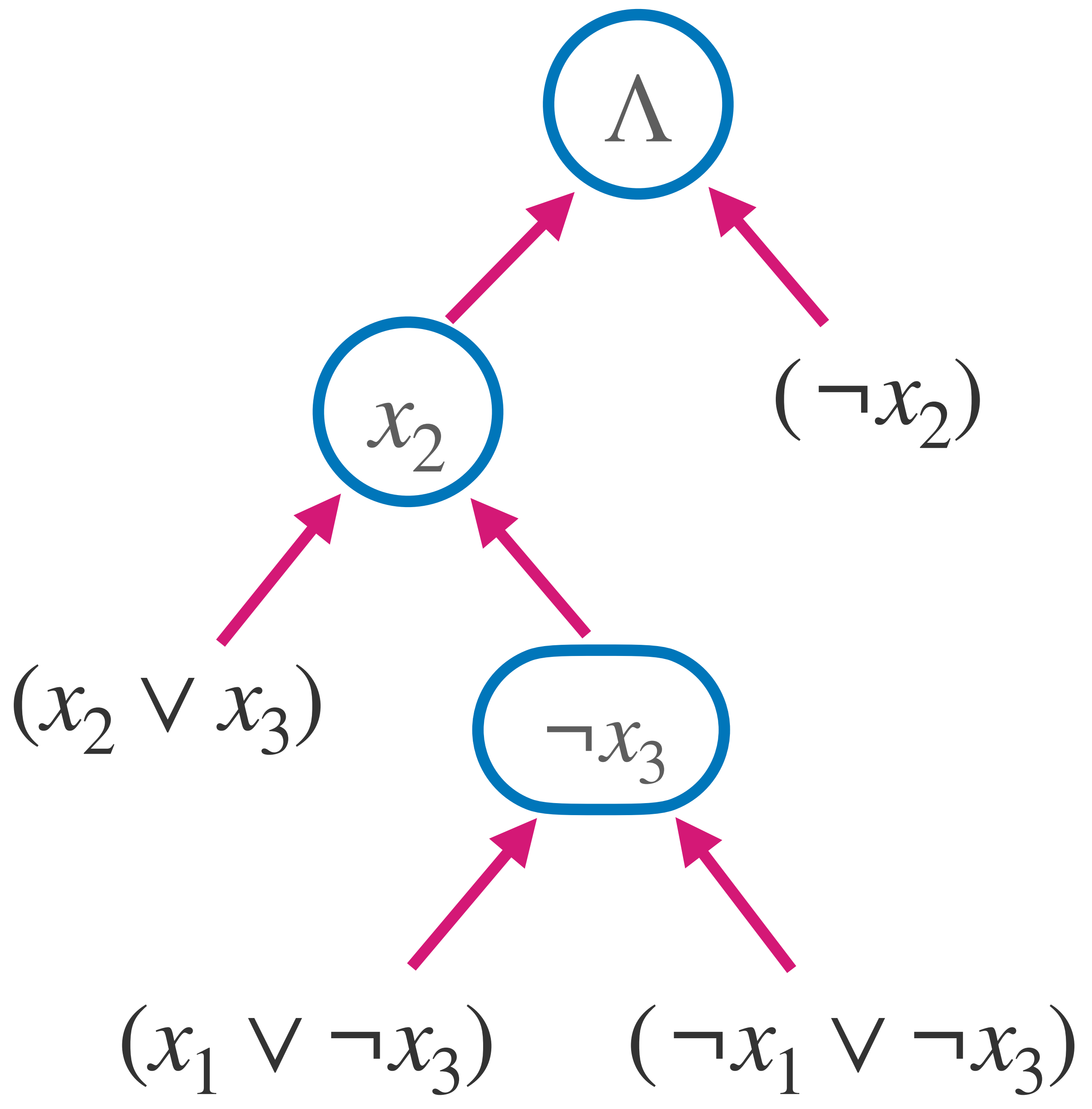
$$(x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3) \wedge (\neg x_2) \wedge (x_1 \vee \neg x_3)$$

Derive new clauses from old using:

$\rightarrow$ **Resolution rule:**

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

Goal: derive empty clause Λ



Resolution rule is **sound**

$\Longrightarrow$ Derivation of Λ certifies unsatisfiability

Last Time — Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

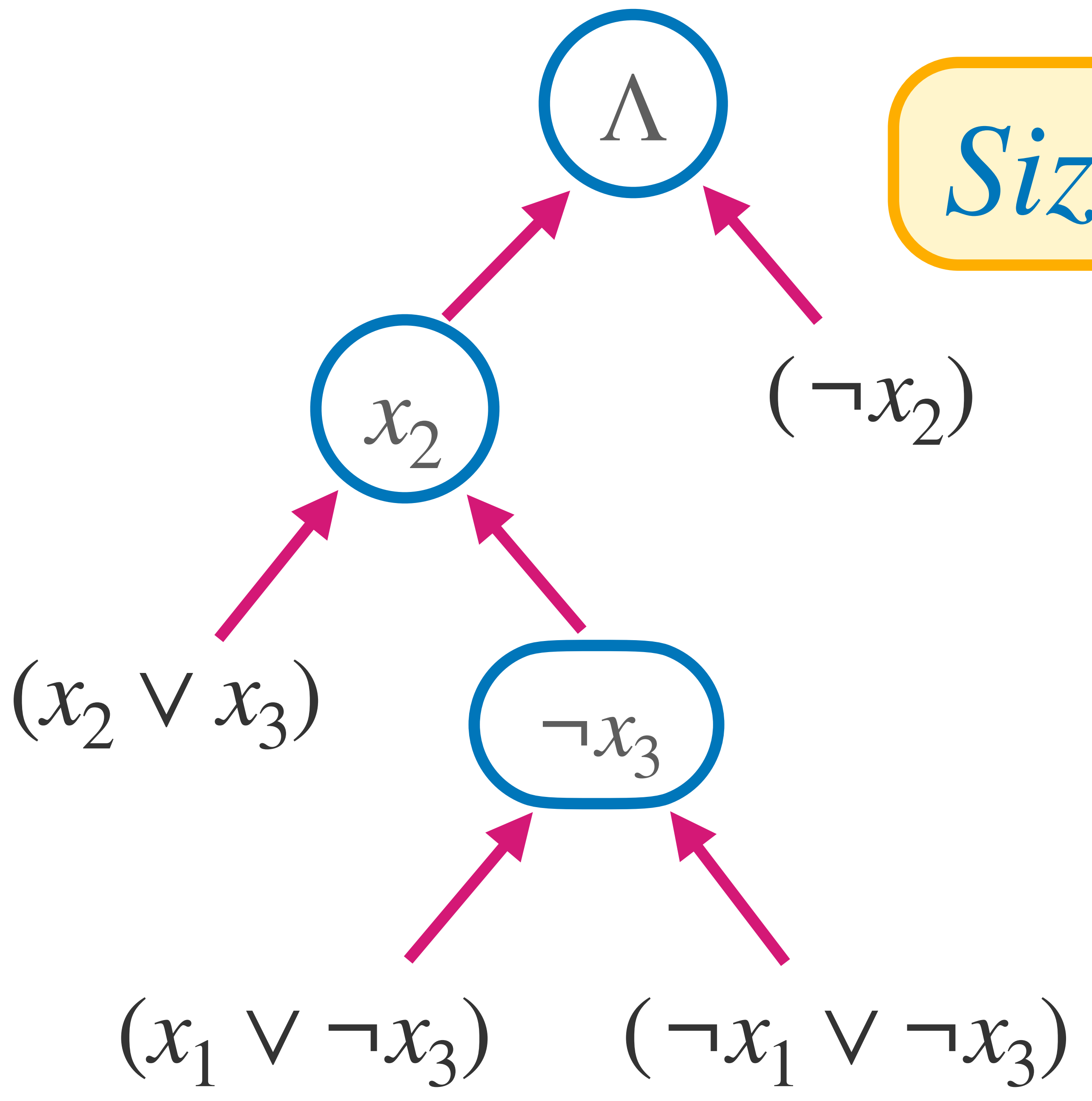
$$(x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3) \wedge (\neg x_2) \wedge (x_1 \vee \neg x_3)$$

Derive new clauses from old using:

$\rightarrow$ **Resolution rule:**

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

Goal: derive empty clause Λ



$Size_R(F)$: # of clauses

Resolution rule is **sound**

$\Longrightarrow$ Derivation of Λ certifies unsatisfiability

Last Time — Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

Derive new clauses from old using:

→ **Resolution rule:**

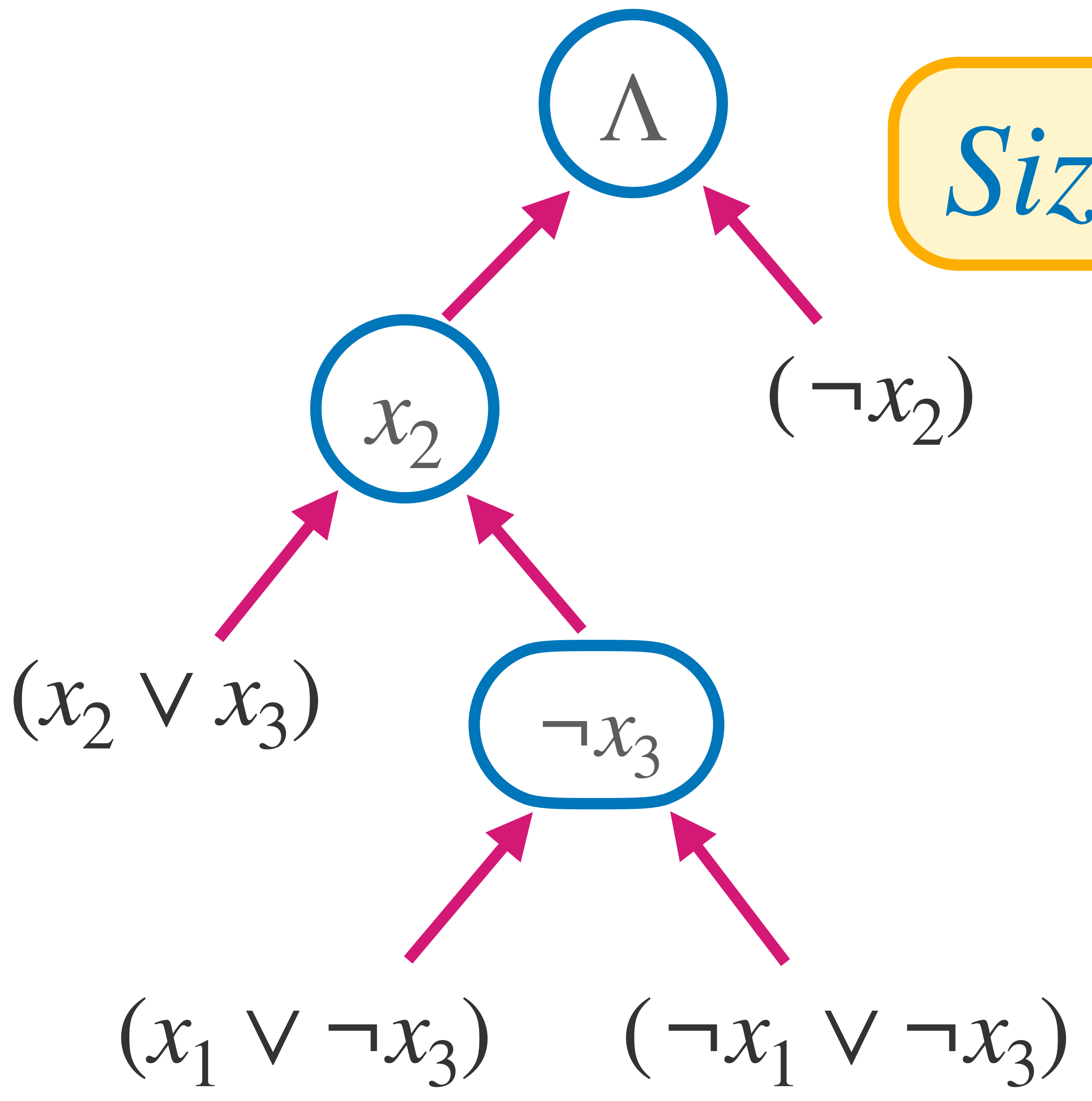
$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

Goal: derive empty clause Λ

Resolution rule is **sound**

$\Longrightarrow$ Derivation of Λ certifies unsatisfiability

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



$Size_R(F)$: # of clauses

(7)

Last Time — Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

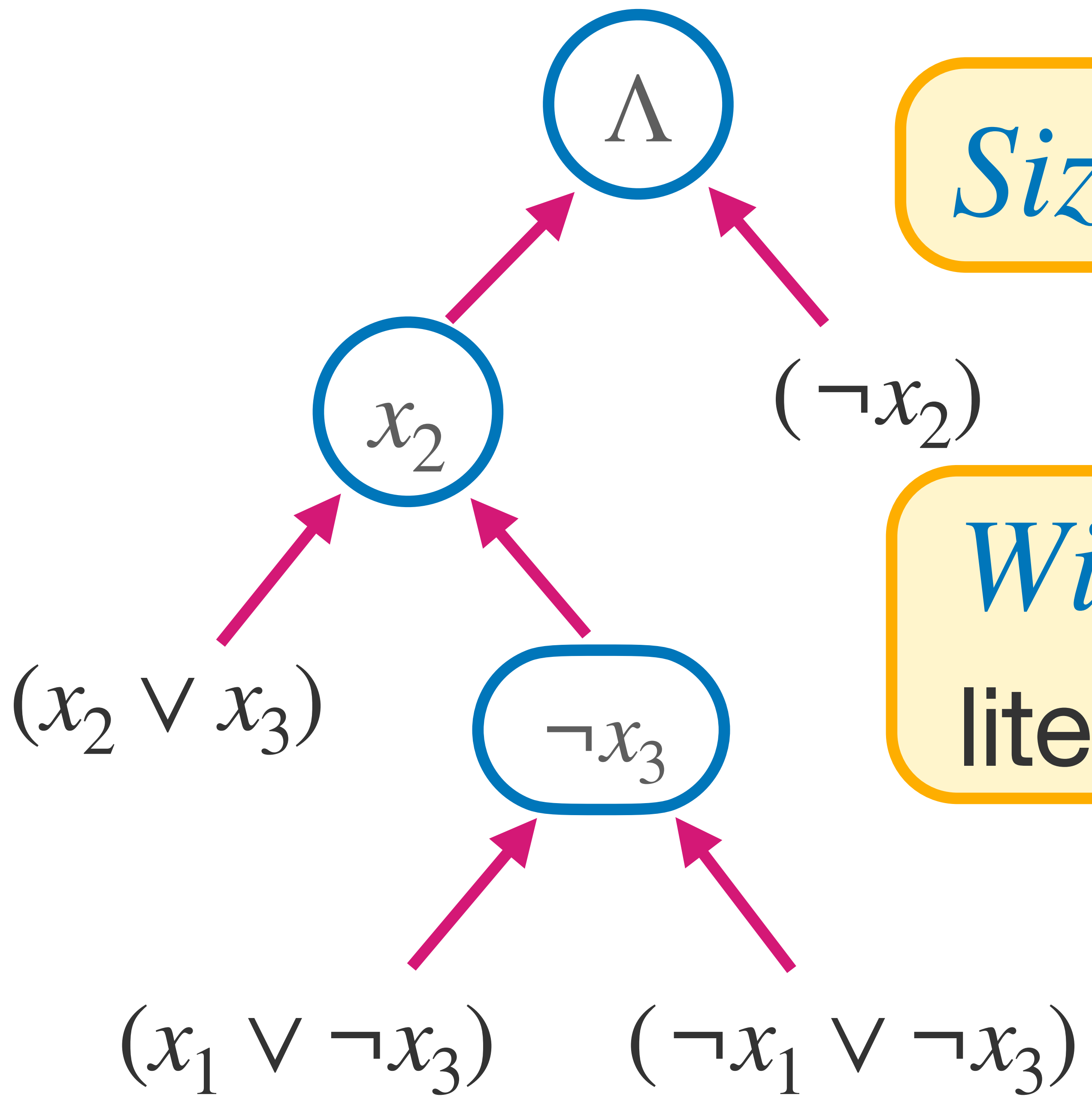
$$(x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3) \wedge (\neg x_2) \wedge (x_1 \vee \neg x_3)$$

Derive new clauses from old using:

$\rightarrow$ **Resolution rule:**

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

Goal: derive empty clause Λ



$Size_R(F)$**:** # of clauses

(7)

$Width_R(F)$**:** max # of literals in any clause

Resolution rule is **sound**

$\Longrightarrow$ Derivation of Λ certifies unsatisfiability

Last Time — Resolution

Resolution: A method for proving that a CNF formula is **unsatisfiable**

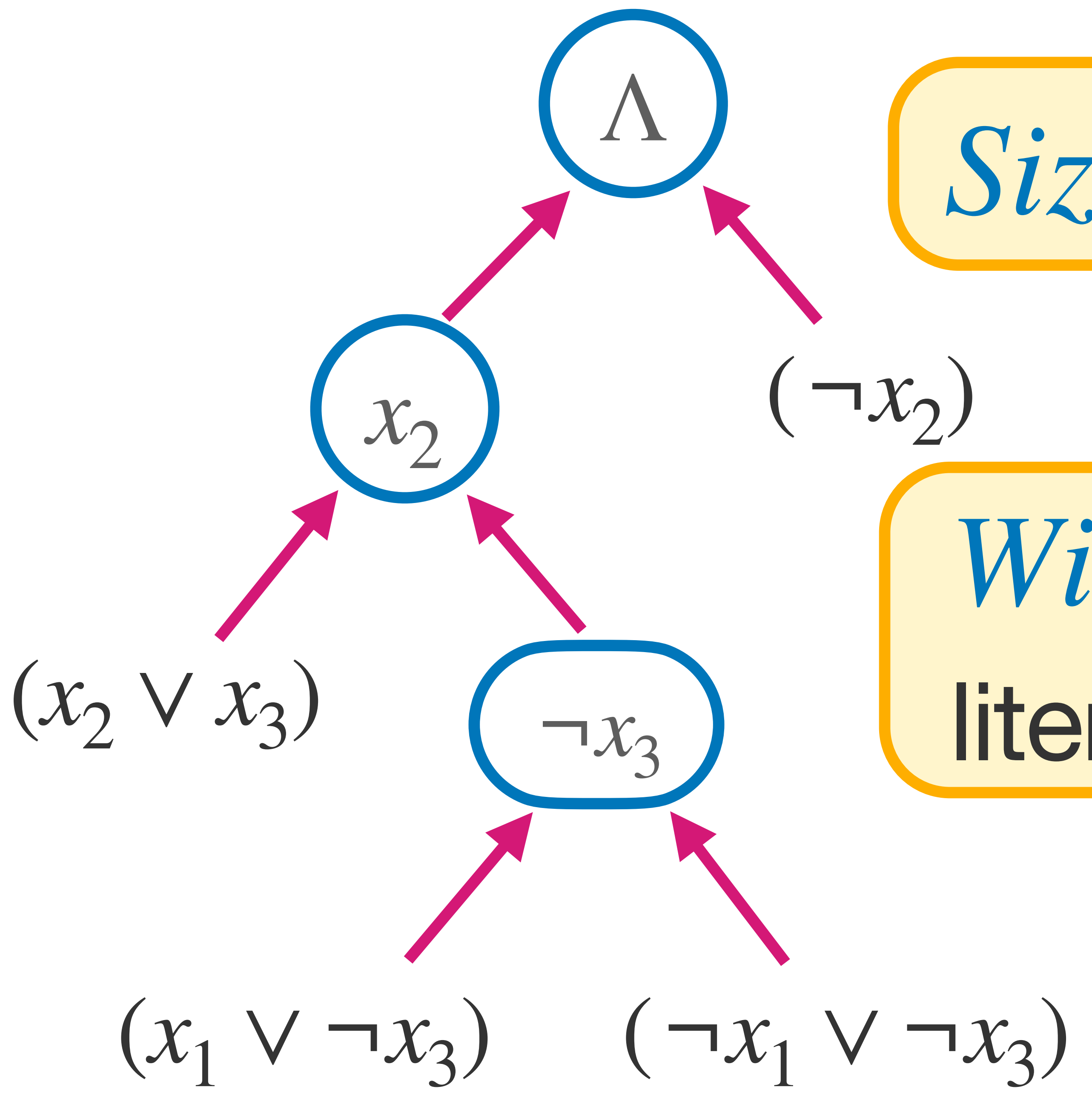
$$(x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3) \wedge (\neg x_2) \wedge (x_1 \vee \neg x_3)$$

Derive new clauses from old using:

$\rightarrow$ **Resolution rule:**

$$\frac{C_1 \vee x, \quad C_2 \vee \neg x}{C_1 \vee C_2}$$

Goal: derive empty clause Λ



$Size_R(F)$**:** # of clauses

(7)

$Width_R(F)$**:** max # of literals in any clause

(2)

Resolution rule is **sound**

$\Longrightarrow$ Derivation of Λ certifies unsatisfiability

Last Time

- Introduced the DPLL algorithm

  → Lower bounds on the runtime of DPLL follow from lower bounds on tree Resolution proofs

- Introduced the CDCL algorithm by extending DPLL with

  → Unit Propagation

  → Clause Learning

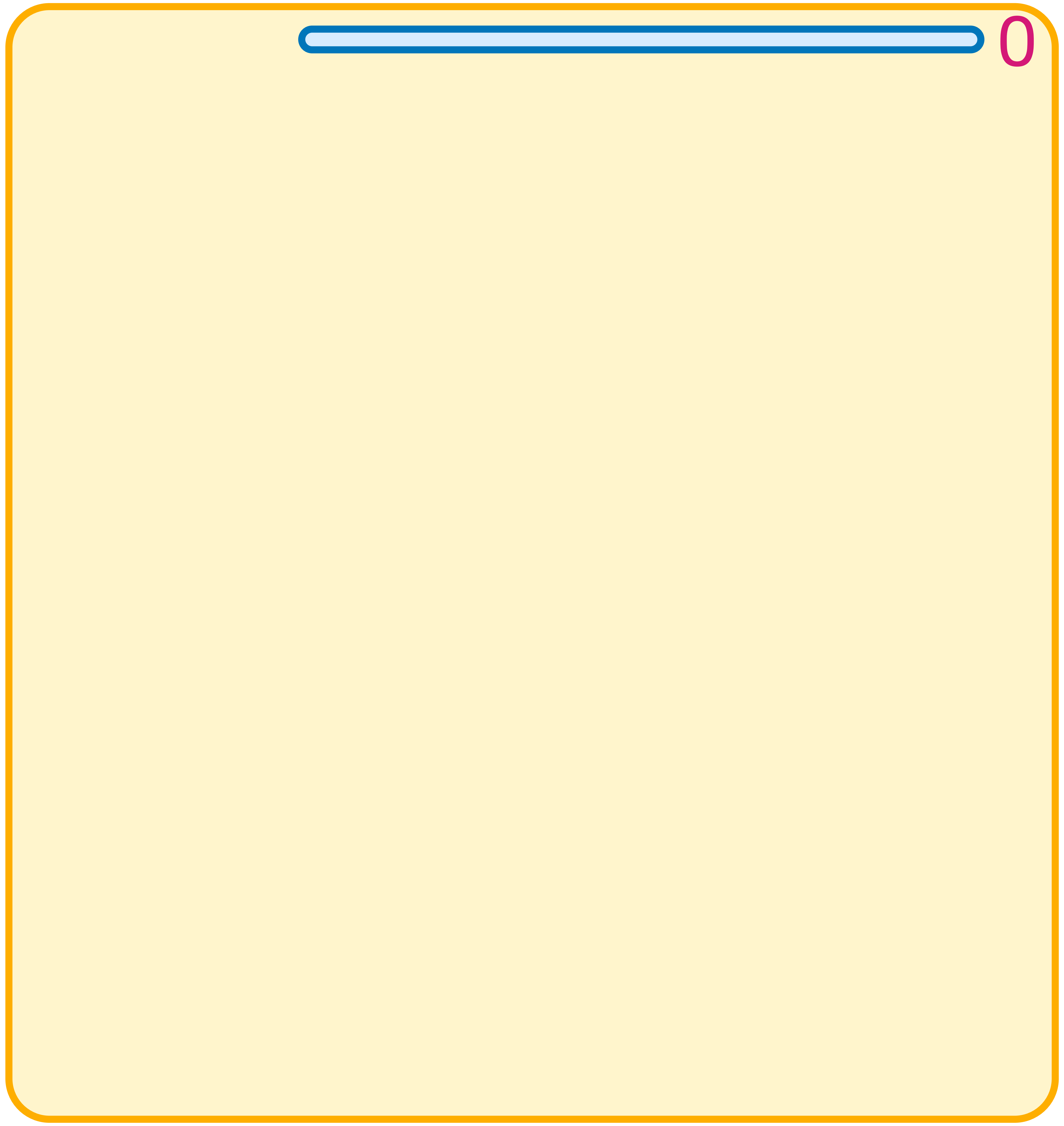
  → Restarts

Last Time — Unit Prop

$(x \vee y) \wedge (z \vee w) \wedge (h \vee \bar{z} \vee \bar{y}) \wedge (\bar{i} \vee \bar{z}) \wedge (i \vee \bar{z} \vee \bar{y})$

Unit clause: a clause containing a **single** literal ℓ

Unit Propagation: if F contains a unit clause (under the current assignment), set $\ell = 1$

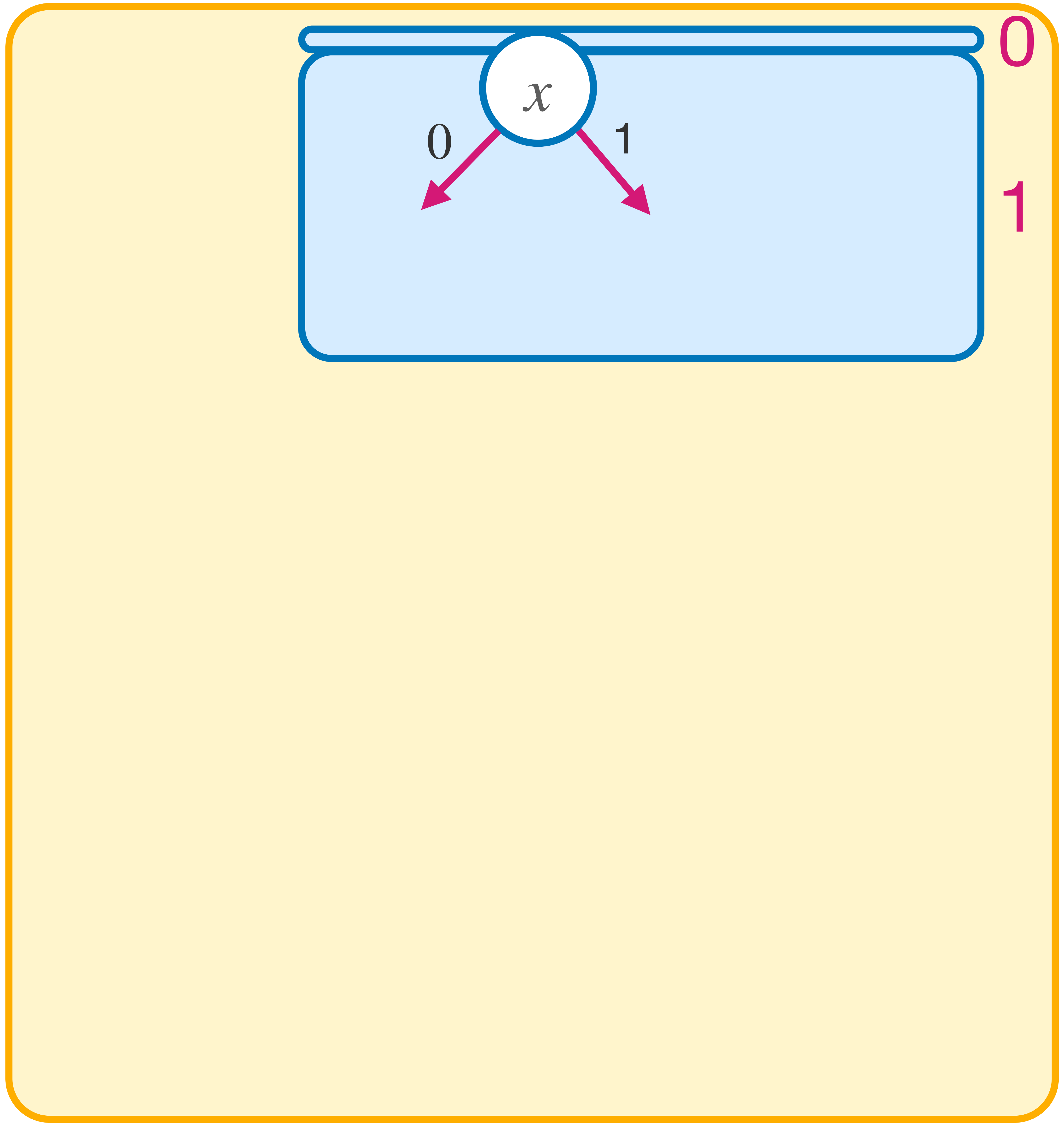


Last Time — Unit Prop

$$(x \vee y) \wedge (z \vee w) \wedge (h \vee \bar{z} \vee \bar{y}) \wedge (\bar{i} \vee \bar{z}) \wedge (i \vee \bar{z} \vee \bar{y})$$

Unit clause: a clause containing a **single** literal ℓ

Unit Propagation: if F contains a unit clause (under the current assignment), set $\ell = 1$

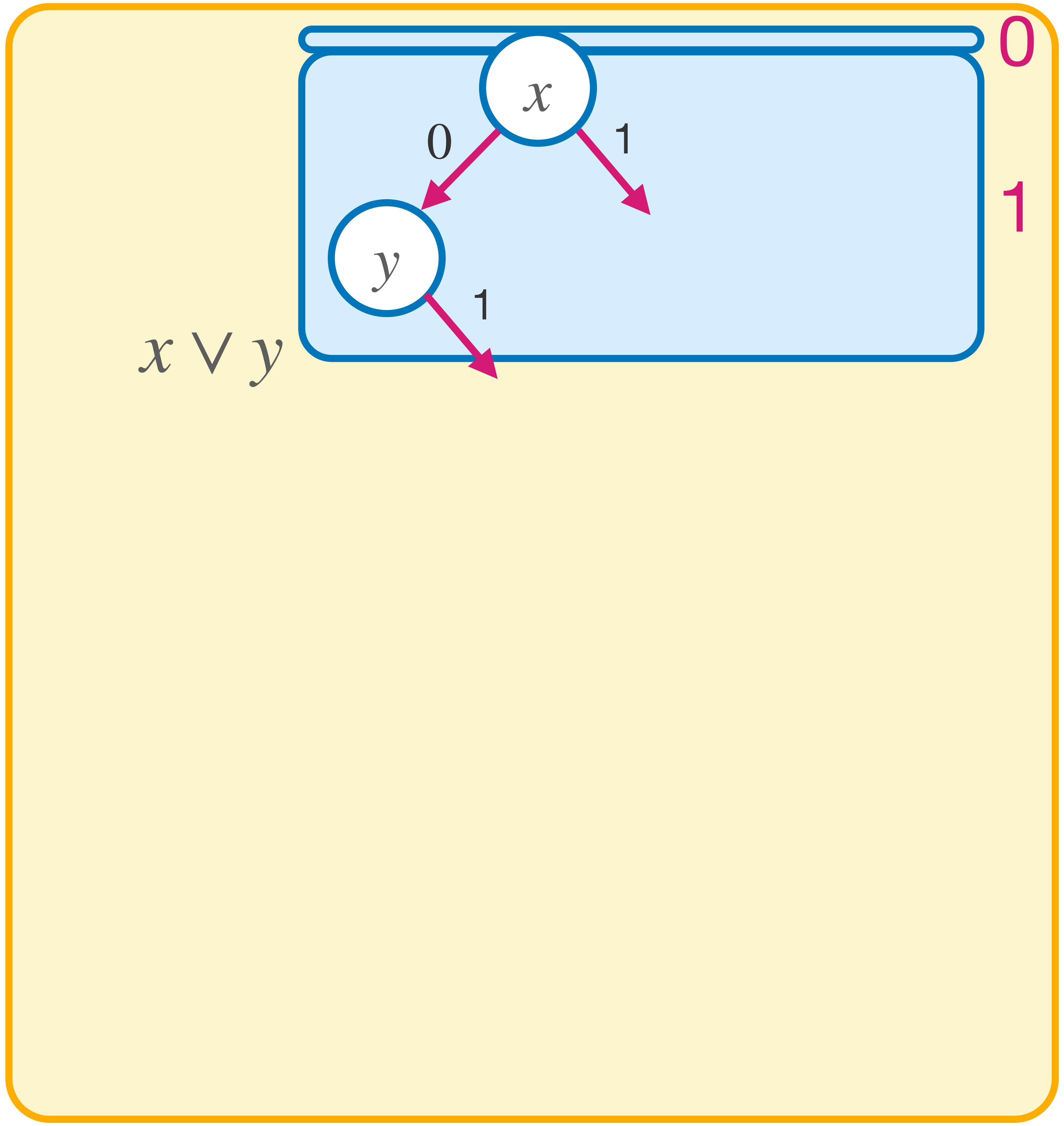


Last Time — Unit Prop

$$(x \vee y) \wedge (z \vee w) \wedge (h \vee \bar{z} \vee \bar{y}) \wedge (\bar{i} \vee \bar{z}) \wedge (i \vee \bar{z} \vee \bar{y})$$

Unit clause: a clause containing a **single** literal ℓ

Unit Propagation: if F contains a unit clause (under the current assignment), set $\ell = 1$

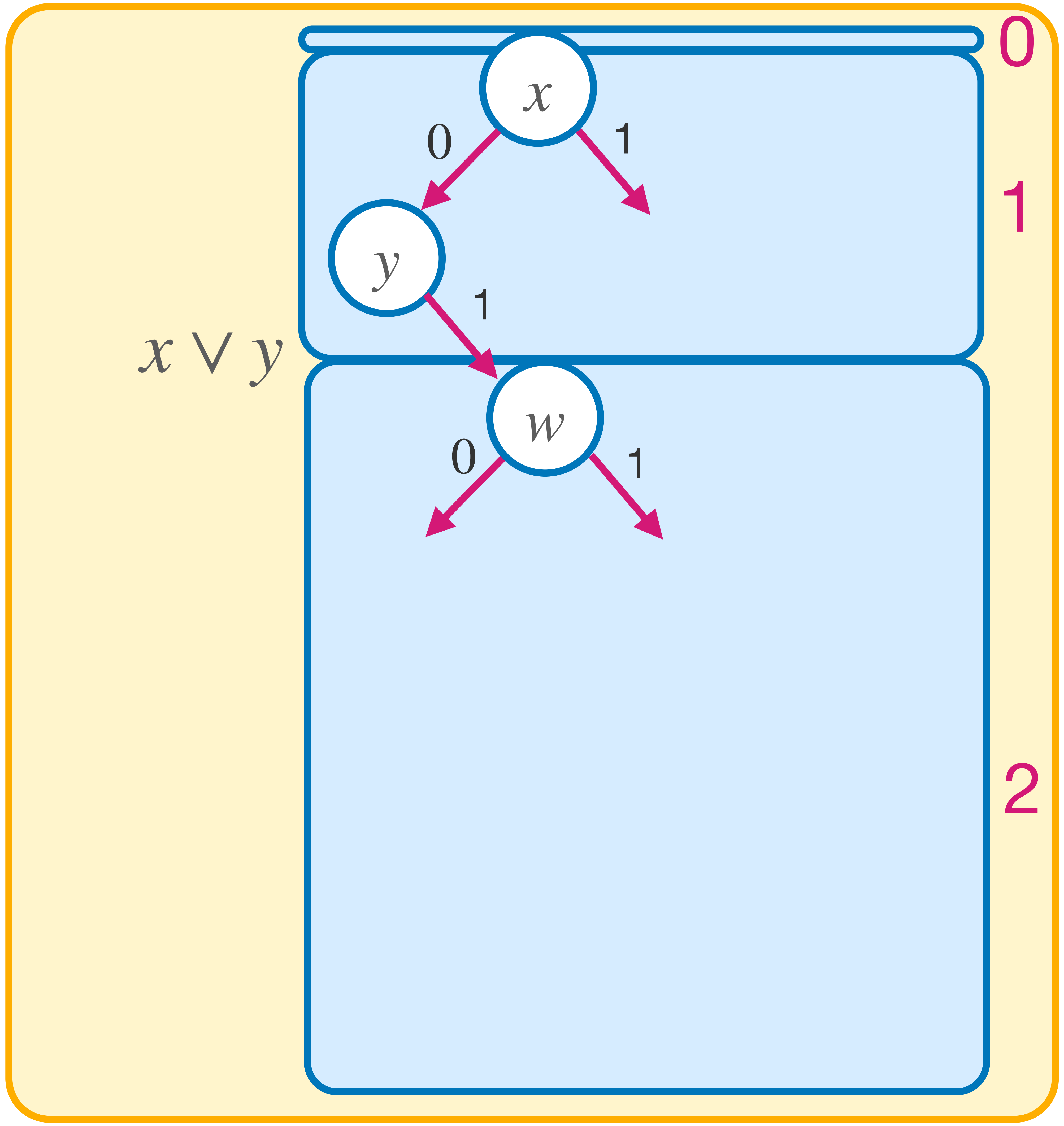


Last Time — Unit Prop

$(x \vee y) \wedge (z \vee w) \wedge (h \vee \bar{z} \vee \bar{y}) \wedge (\bar{i} \vee \bar{z}) \wedge (i \vee \bar{z} \vee \bar{y})$

Unit clause: a clause containing a **single** literal ℓ

Unit Propagation: if F contains a unit clause (under the current assignment), set $\ell = 1$

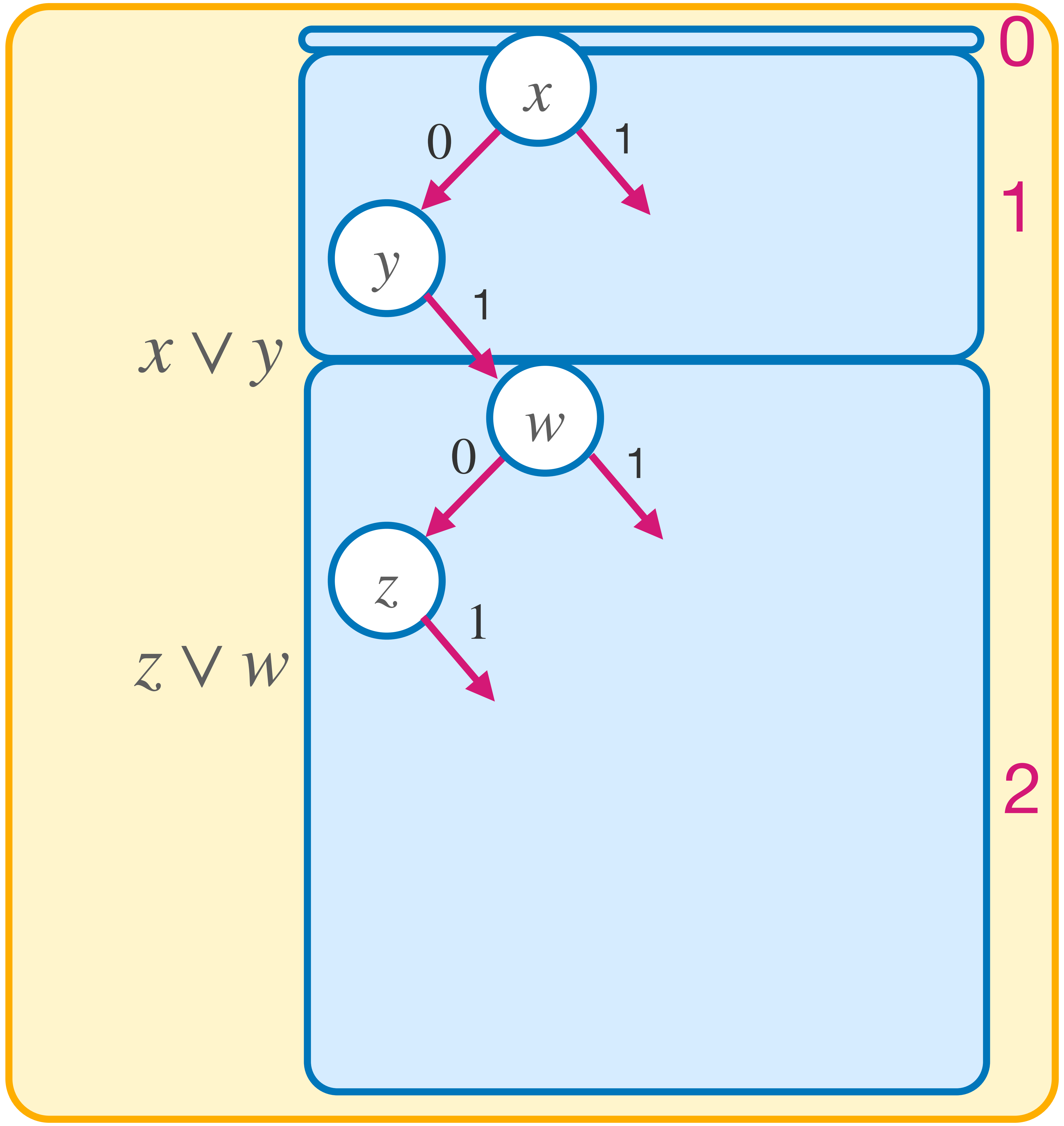


Last Time — Unit Prop

$$(x \vee y) \wedge (z \vee w) \wedge (h \vee \bar{z} \vee \bar{y}) \wedge (\bar{i} \vee \bar{z}) \wedge (i \vee \bar{z} \vee \bar{y})$$

Unit clause: a clause containing a **single** literal ℓ

Unit Propagation: if F contains a unit clause (under the current assignment), set $\ell = 1$

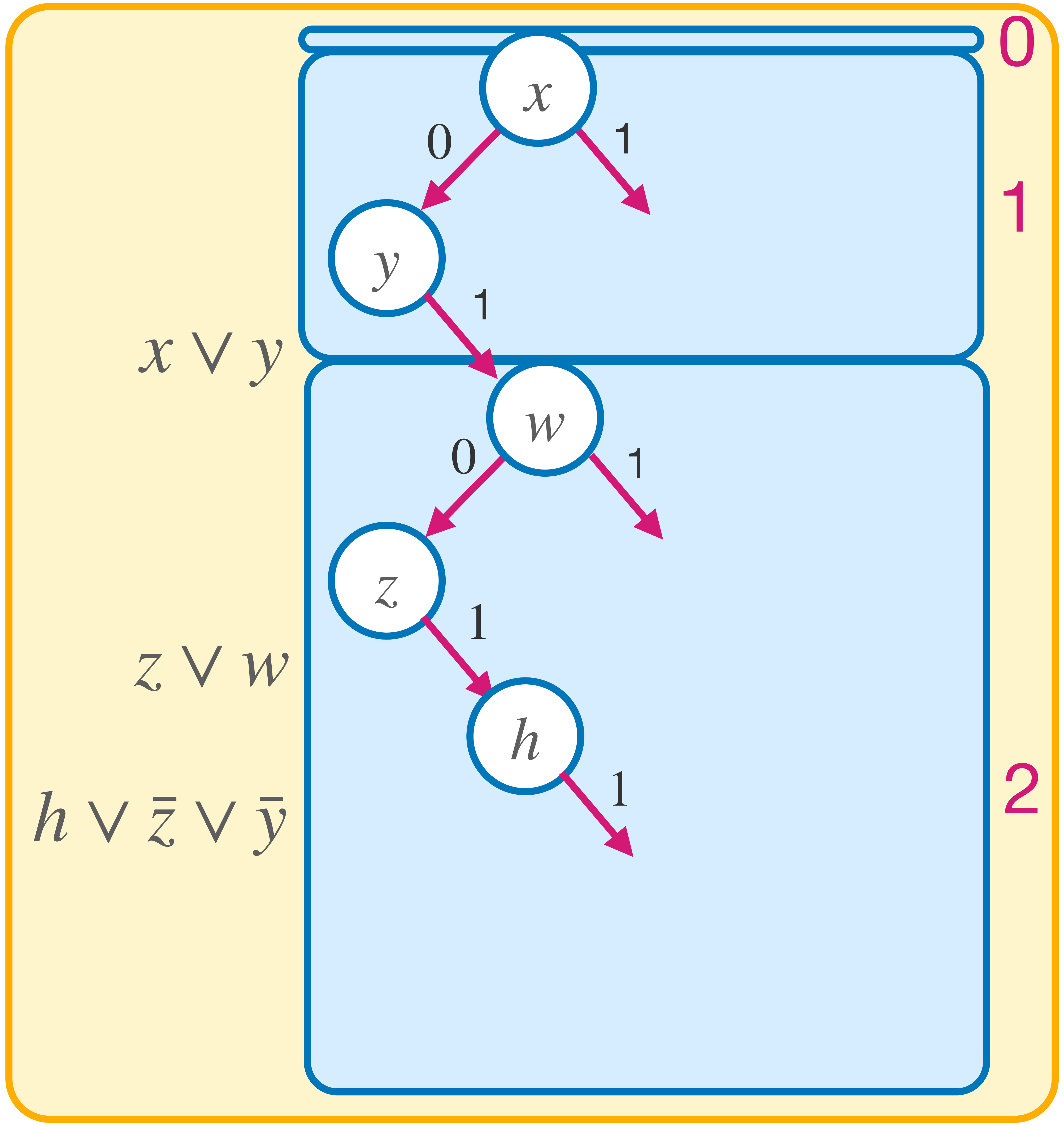


Last Time — Unit Prop

$(x \vee y) \wedge (z \vee w) \wedge (h \vee \bar{z} \vee \bar{y}) \wedge (\bar{i} \vee \bar{z}) \wedge (i \vee \bar{z} \vee \bar{y})$

Unit clause: a clause containing a **single** literal ℓ

Unit Propagation: if F contains a unit clause (under the current assignment), set $\ell = 1$

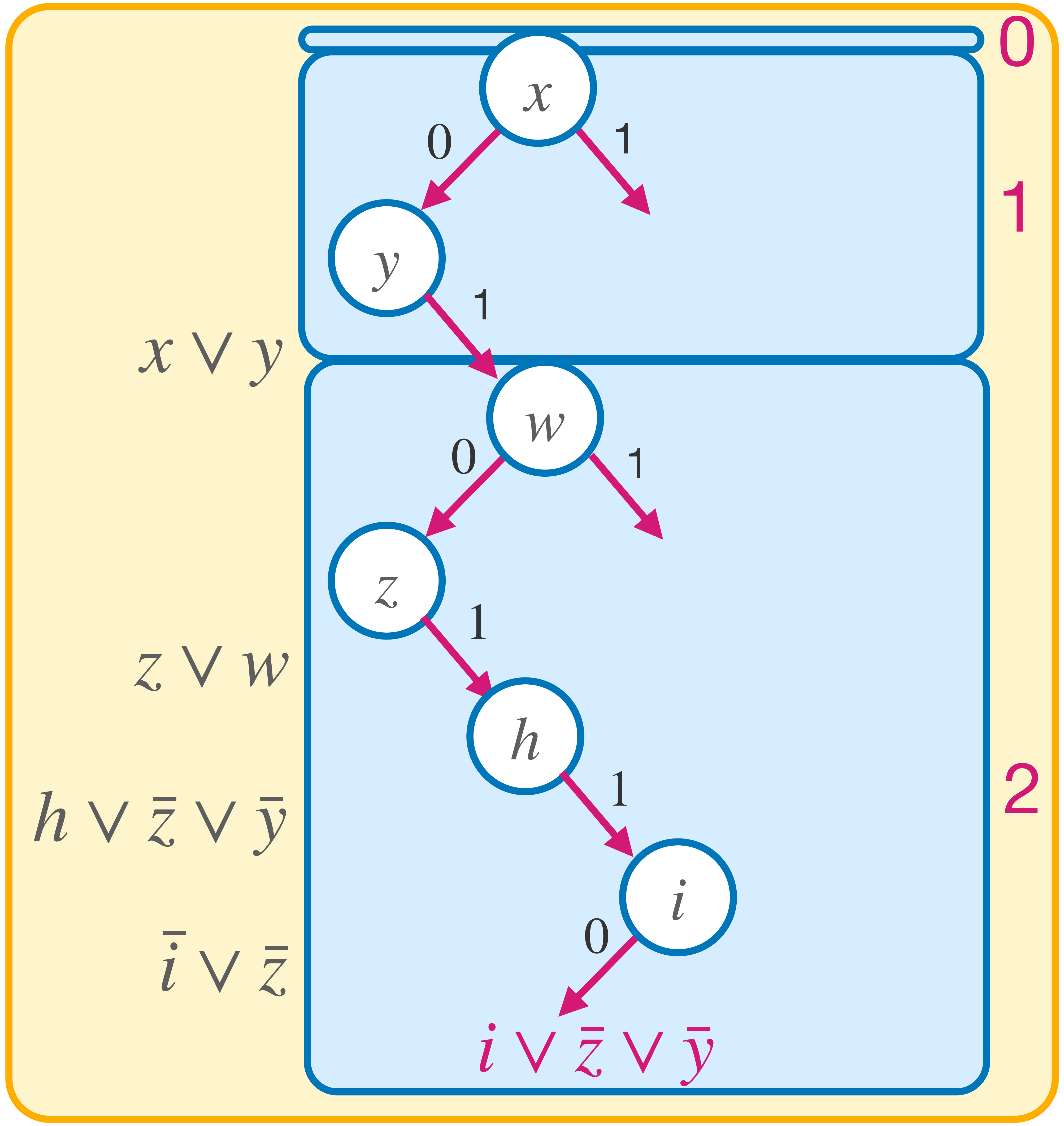


Last Time — Unit Prop

$$(x \vee y) \wedge (z \vee w) \wedge (h \vee \bar{z} \vee \bar{y}) \wedge (\bar{i} \vee \bar{z}) \wedge (i \vee \bar{z} \vee \bar{y})$$

Unit clause: a clause containing a **single** literal ℓ

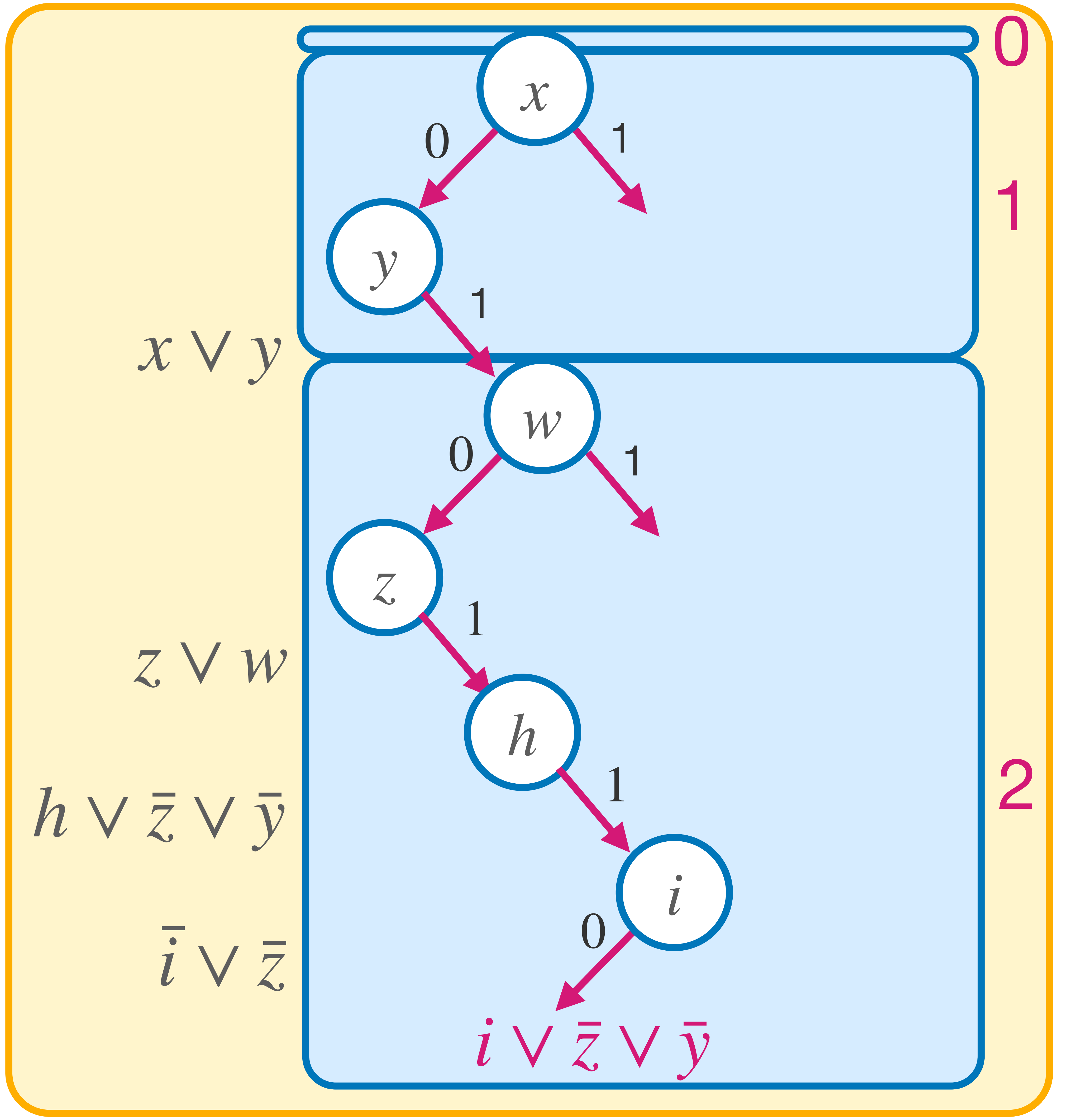
Unit Propagation: if F contains a unit clause (under the current assignment), set $\ell = 1$



Last Time — Clause Learning

$$(x \vee y) \wedge (z \vee w) \wedge (h \vee \bar{z} \vee \bar{y}) \wedge (\bar{i} \vee \bar{z}) \wedge (i \vee \bar{z} \vee \bar{y})$$

When a conflict, use Resolution to learn a clause



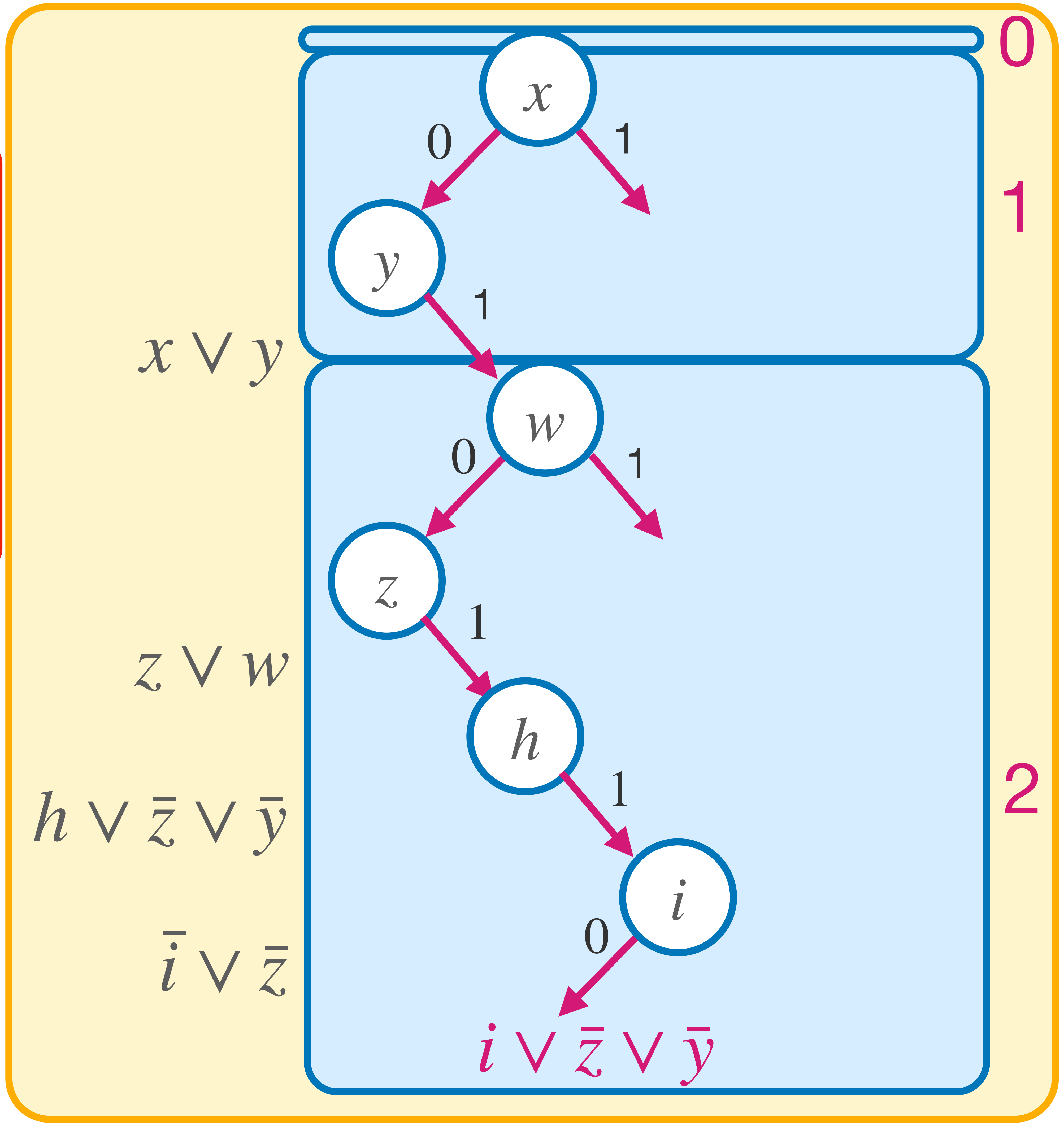
Last Time — Clause Learning

$$(x \lor y) \land (z \lor w) \land (h \lor \bar{z} \lor \bar{y}) \land (\bar{i} \lor \bar{z}) \land (i \lor \bar{z} \lor \bar{y})$$

When a conflict, use Resolution to learn a clause

1-UIP Clause

Obtained by resolving the conflict clause along the path until there is only one literal in the clause at the largest decision level



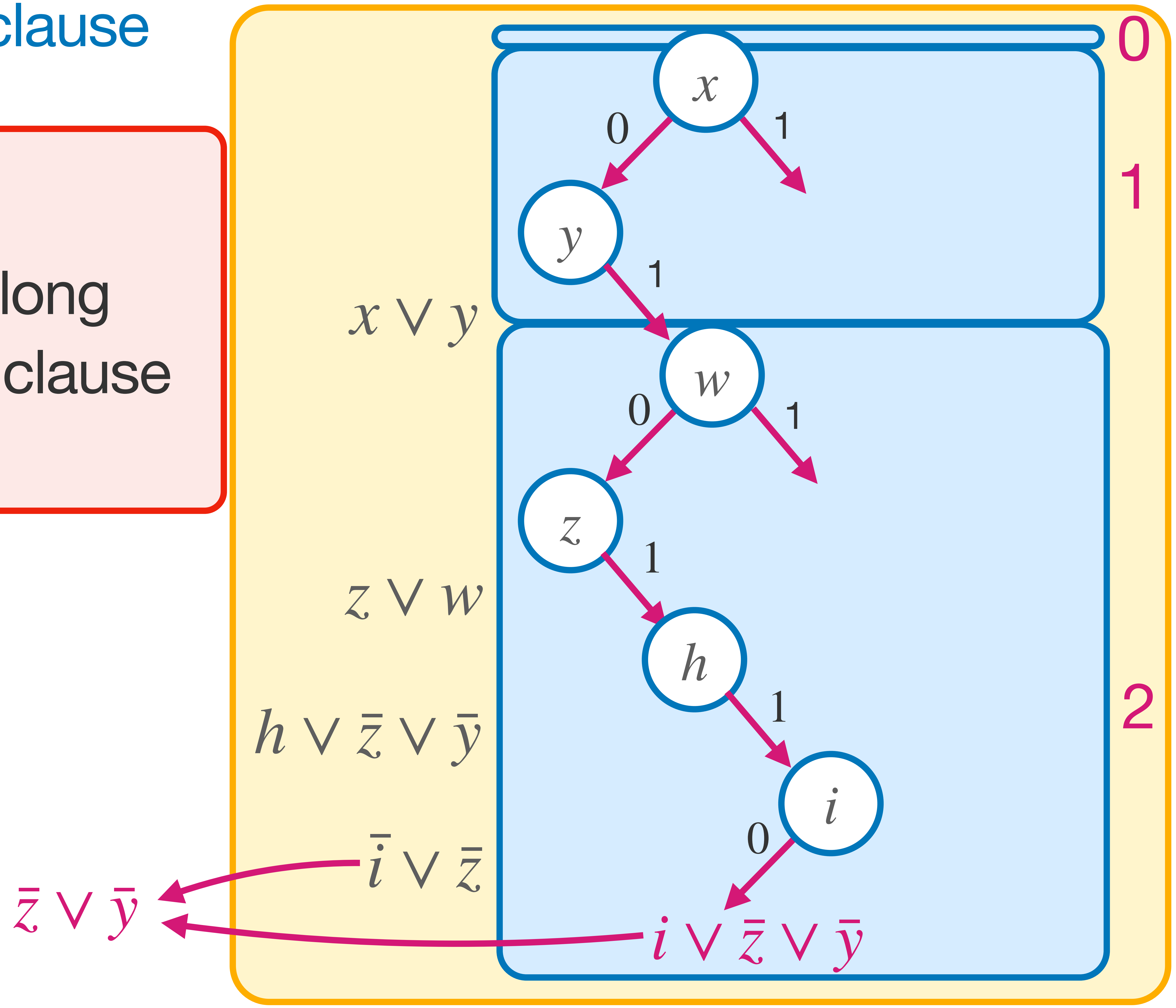
Last Time — Clause Learning

$(x \vee y) \wedge (z \vee w) \wedge (h \vee \bar{z} \vee \bar{y}) \wedge (\bar{i} \vee \bar{z}) \wedge (i \vee \bar{z} \vee \bar{y})$

When a conflict, use Resolution to learn a clause

1-UIP Clause

Obtained by resolving the conflict clause along the path until there is only one literal in the clause at the largest decision level



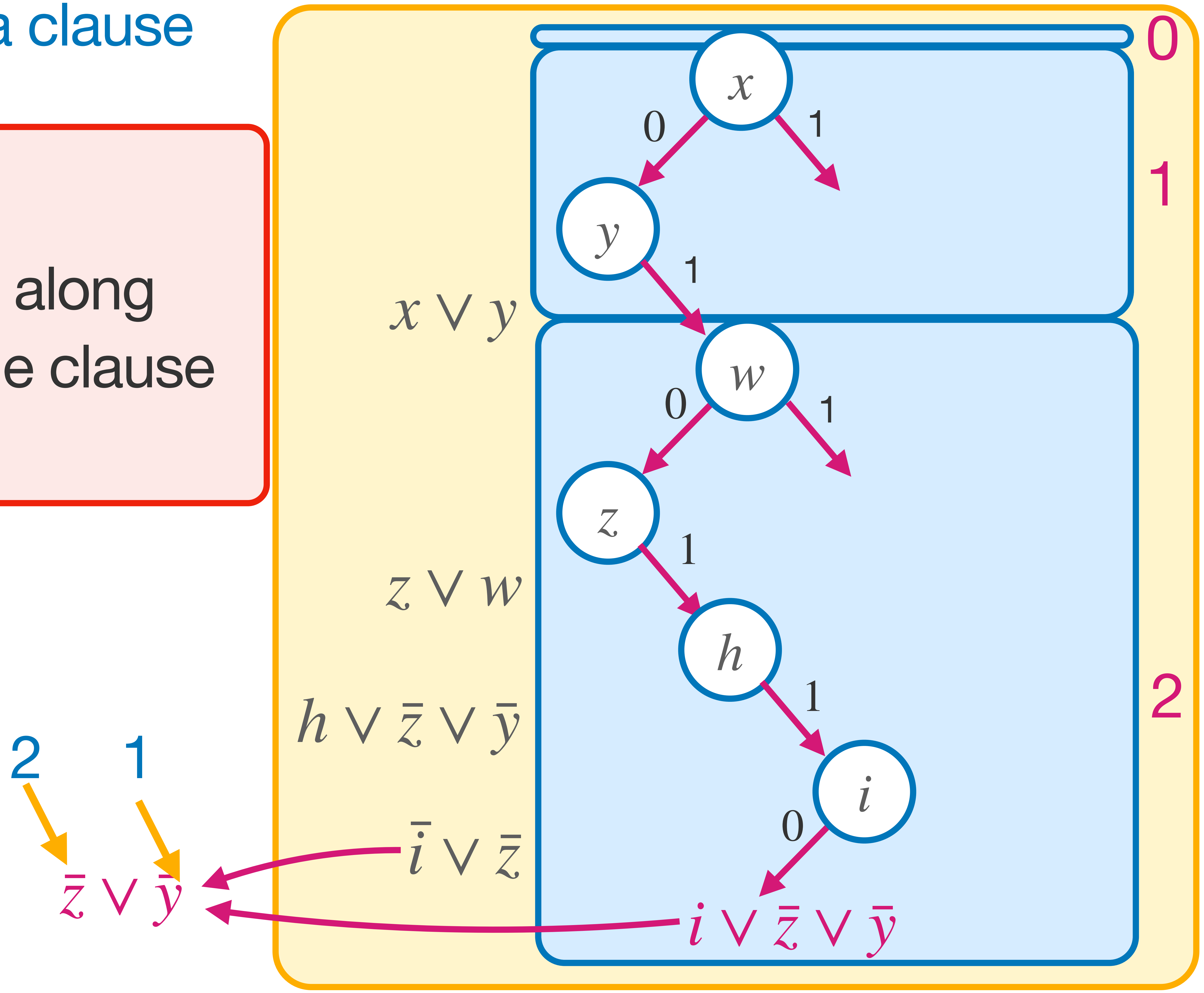
Last Time — Clause Learning

$(x \vee y) \wedge (z \vee w) \wedge (h \vee \bar{z} \vee \bar{y}) \wedge (\bar{i} \vee \bar{z}) \wedge (i \vee \bar{z} \vee \bar{y})$

When a conflict, use Resolution to learn a clause

1-UIP Clause

Obtained by resolving the conflict clause along the path until there is only one literal in the clause at the largest decision level



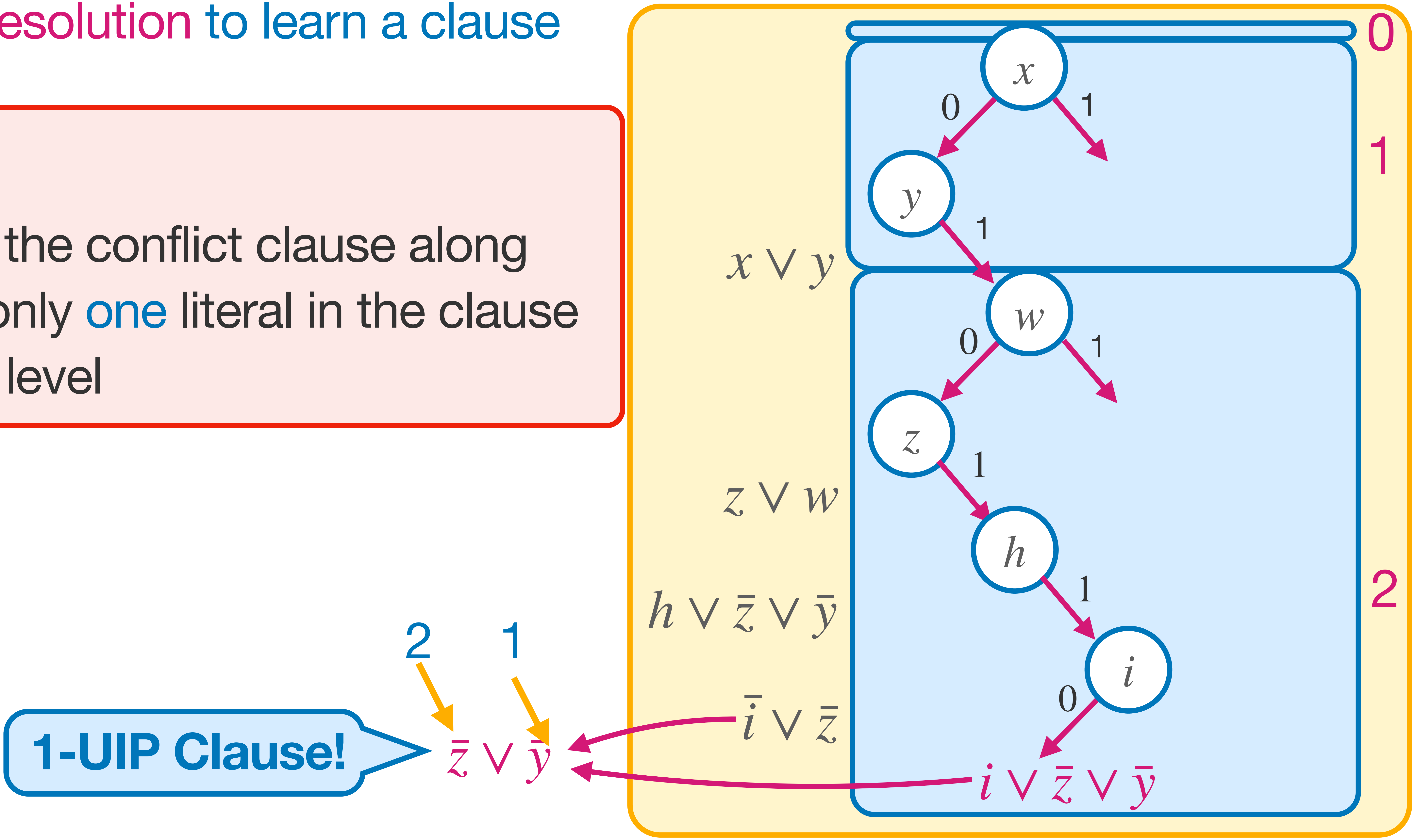
Last Time — Clause Learning

$(x \vee y) \wedge (z \vee w) \wedge (h \vee \bar{z} \vee \bar{y}) \wedge (\bar{i} \vee \bar{z}) \wedge (i \vee \bar{z} \vee \bar{y})$

When a conflict, use Resolution to learn a clause

1-UIP Clause

Obtained by resolving the conflict clause along the path until there is only one literal in the clause at the largest decision level



Last Time — Clause Learning

$(\bar{z} \vee \bar{y}) \wedge (x \vee y) \wedge (z \vee w) \wedge (h \vee \bar{z} \vee \bar{y}) \wedge (\bar{i} \vee \bar{z}) \wedge (i \vee \bar{z} \vee \bar{y})$

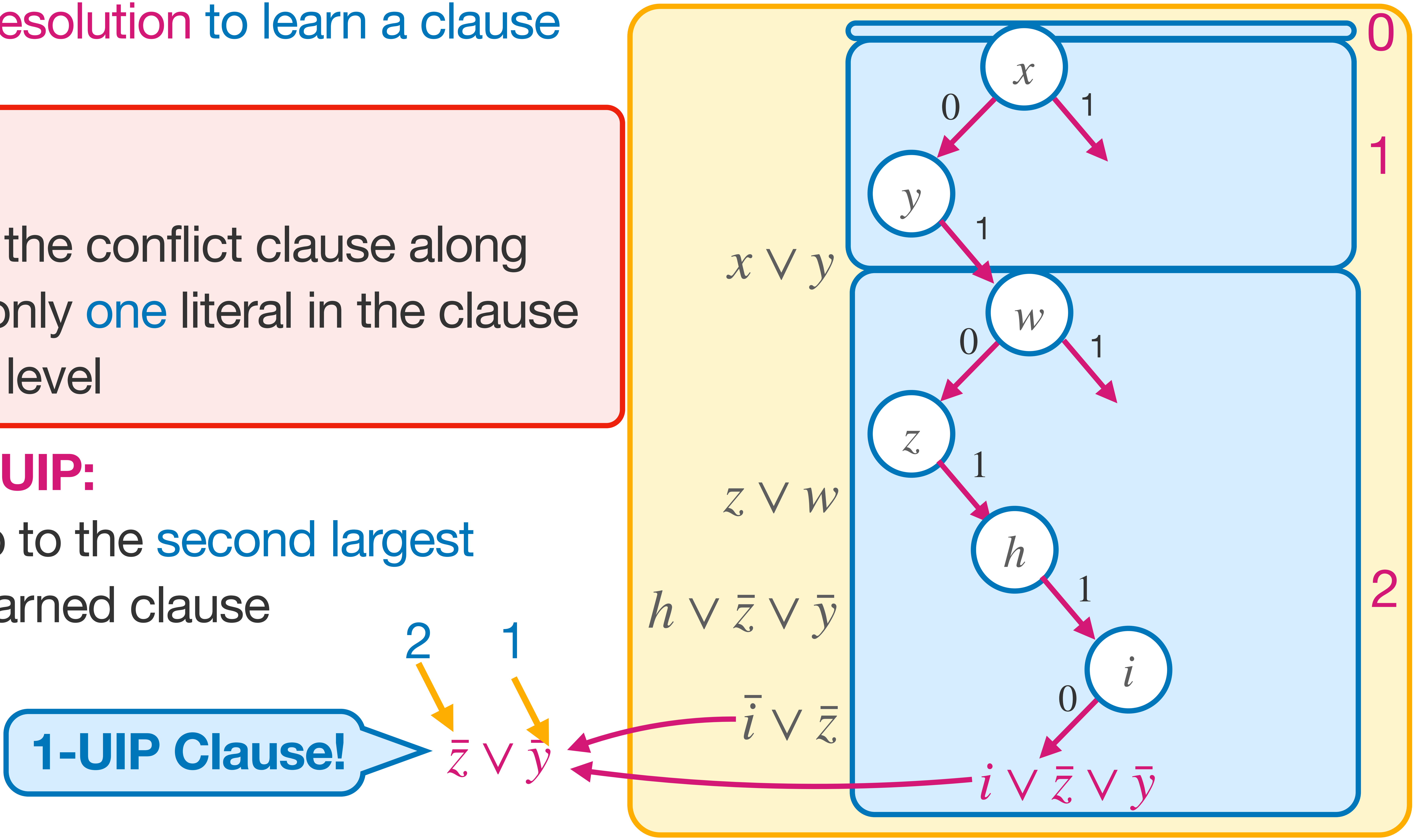
When a conflict, use Resolution to learn a clause

1-UIP Clause

Obtained by resolving the conflict clause along the path until there is only one literal in the clause at the largest decision level

Backtracking with 1-UIP:

Remove everything up to the second largest decision level in the learned clause



Last Time — Clause Learning

$(\bar{z} \vee \bar{y}) \wedge (x \vee y) \wedge (z \vee w) \wedge (h \vee \bar{z} \vee \bar{y}) \wedge (\bar{i} \vee \bar{z}) \wedge (i \vee \bar{z} \vee \bar{y})$

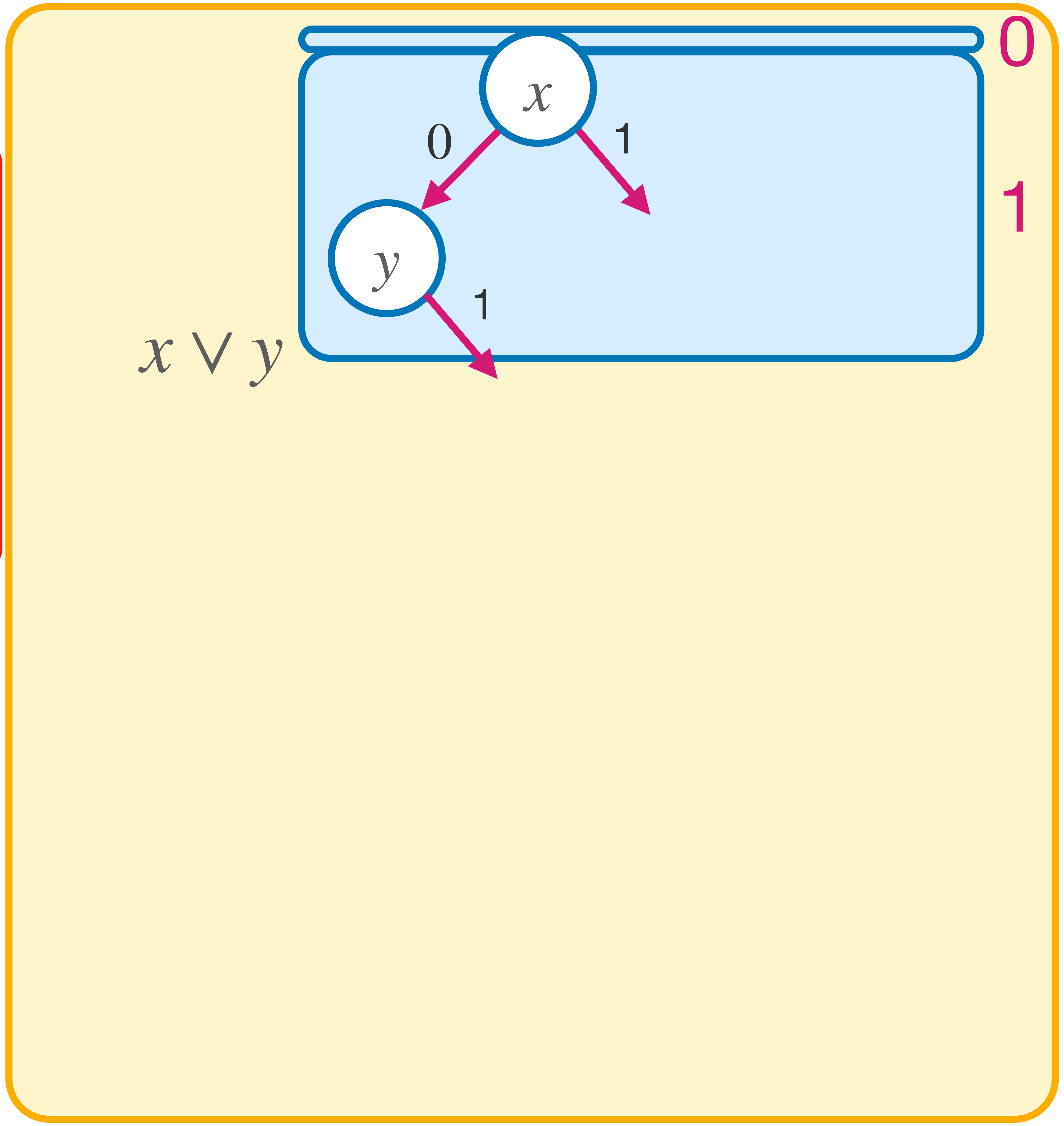
When a conflict, use Resolution to learn a clause

1-UIP Clause

Obtained by resolving the conflict clause along the path until there is only one literal in the clause at the largest decision level

Backtracking with 1-UIP:

Remove everything up to the second largest decision level in the learned clause



Analyzing CDCL

Q. Can we show CDCL doesn't solve SAT in polytime?

Analyzing CDCL

Q. Can we show CDCL doesn't solve SAT in polytime?

$\rightarrow$ We saw that DPLL when run on an unsatisfiable formula gives a tree Resolution proof

Analyzing CDCL

Q . Can we show CDCL doesn't solve SAT in polytime?

$\rightarrow$ We saw that DPLL when run on an unsatisfiable formula gives a tree Resolution proof

Theorem: Let F be an unsatisfiable CNF formula. If CDCL takes time s to solve F, then there is a size-s Resolution proof of F

Analyzing CDCL

Q. Can we show CDCL doesn't solve SAT in polytime?

$\rightarrow$ We saw that DPLL when run on an unsatisfiable formula gives a tree Resolution proof

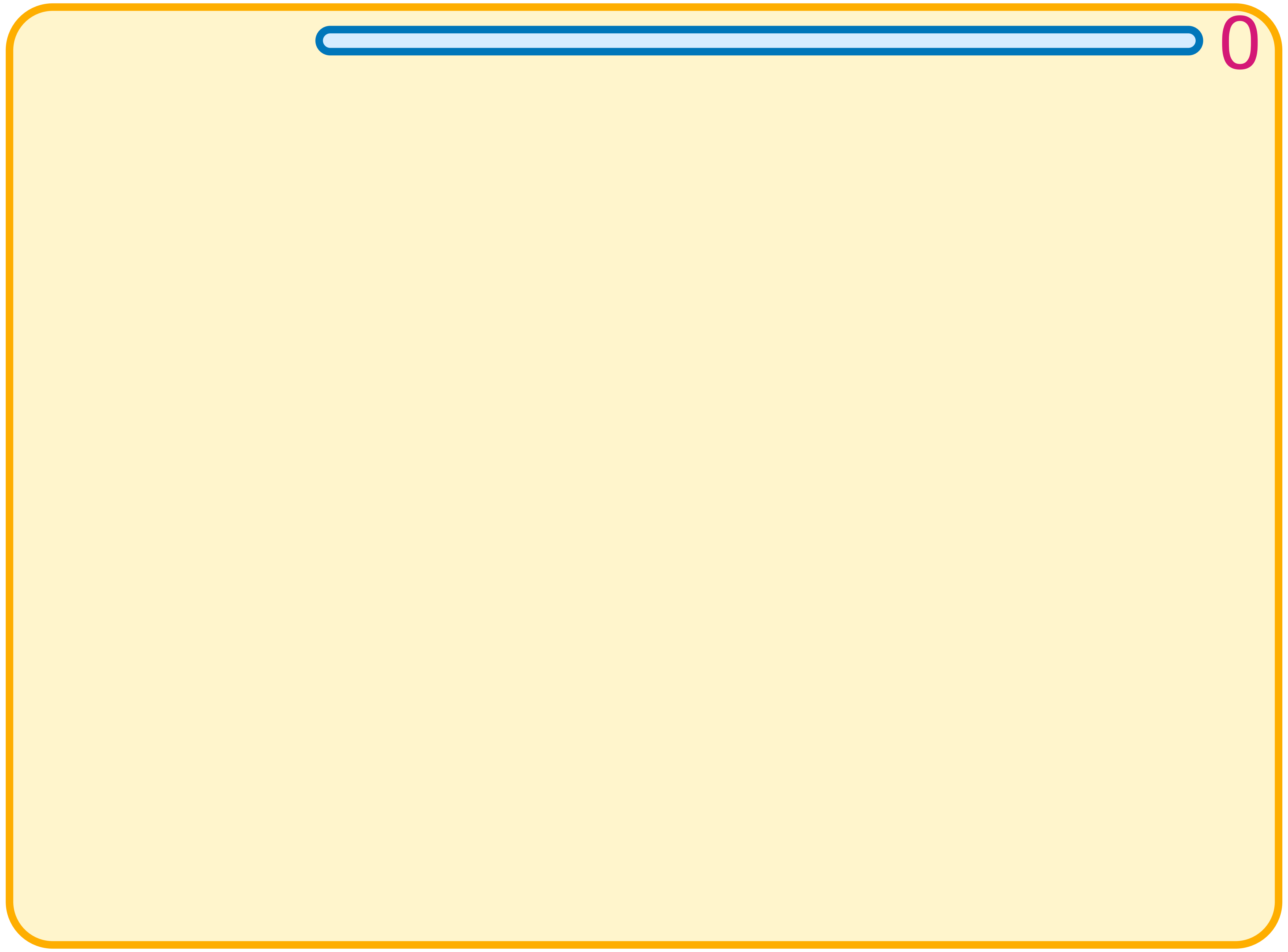
Theorem: Let F be an unsatisfiable CNF formula. If CDCL takes time s to solve F, then there is a size-s Resolution proof of F

Q. What do we learn if we run CDCL on an unsatisfiable formula?

Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

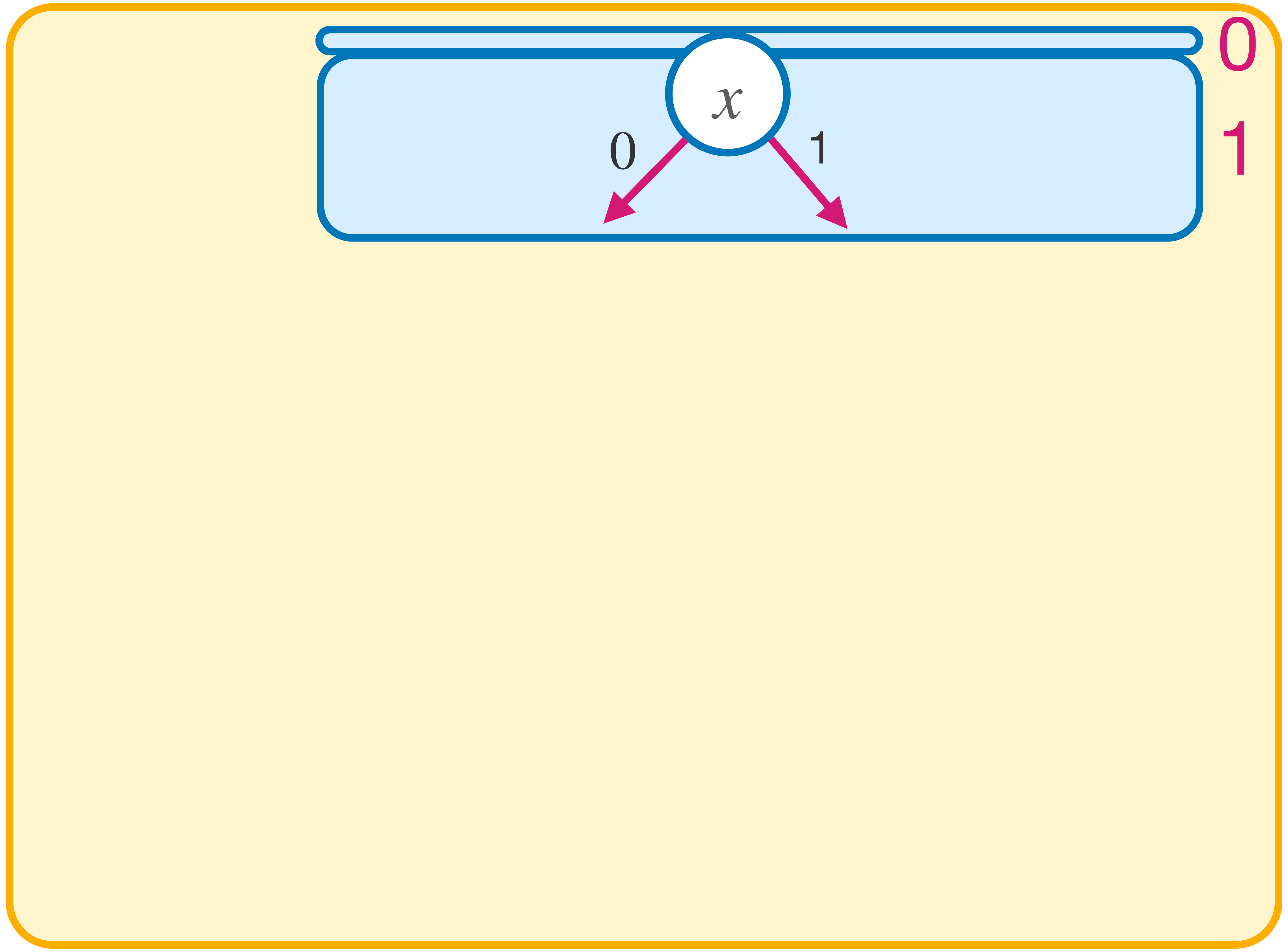
$$(y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

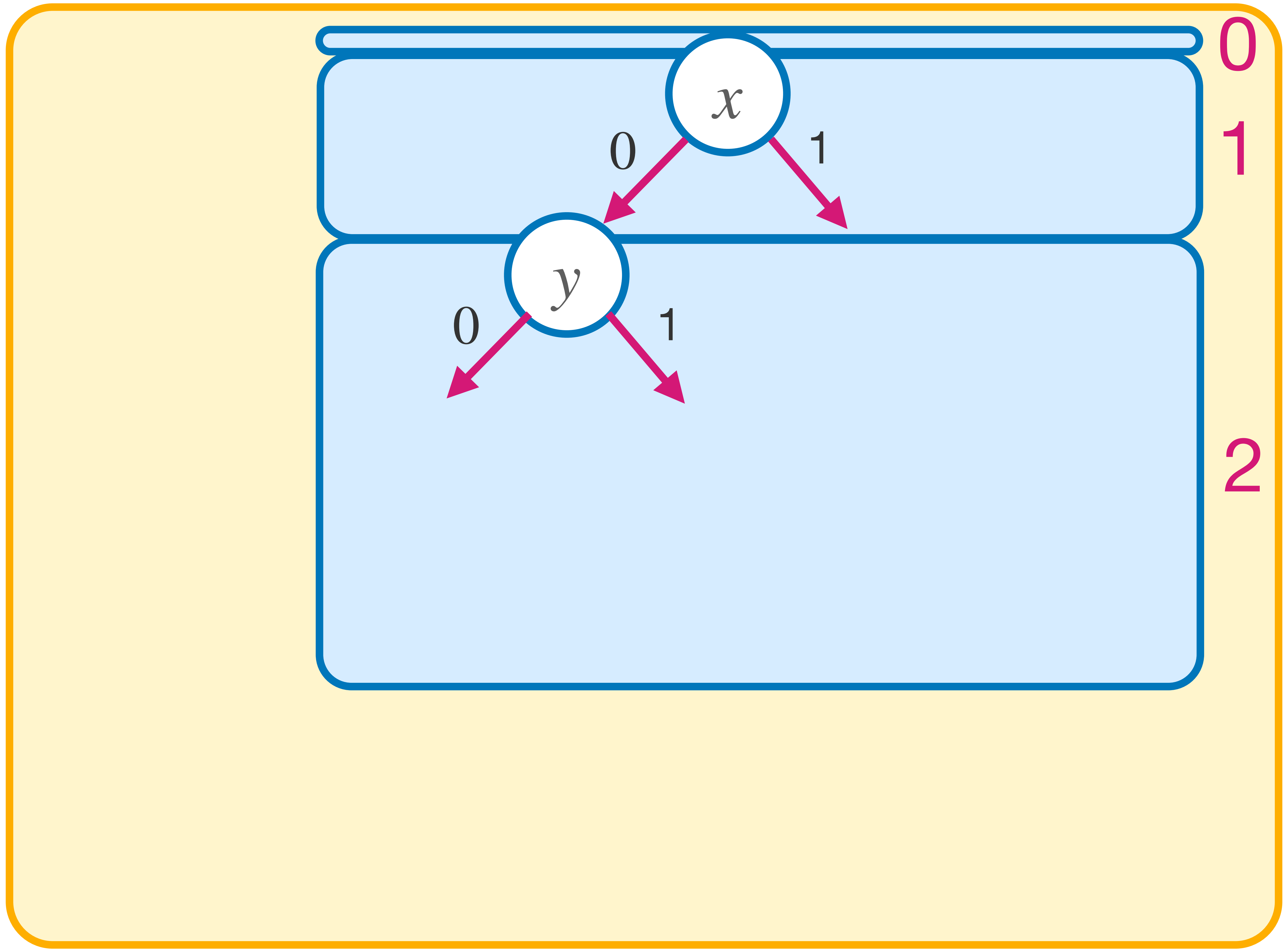
$$(y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

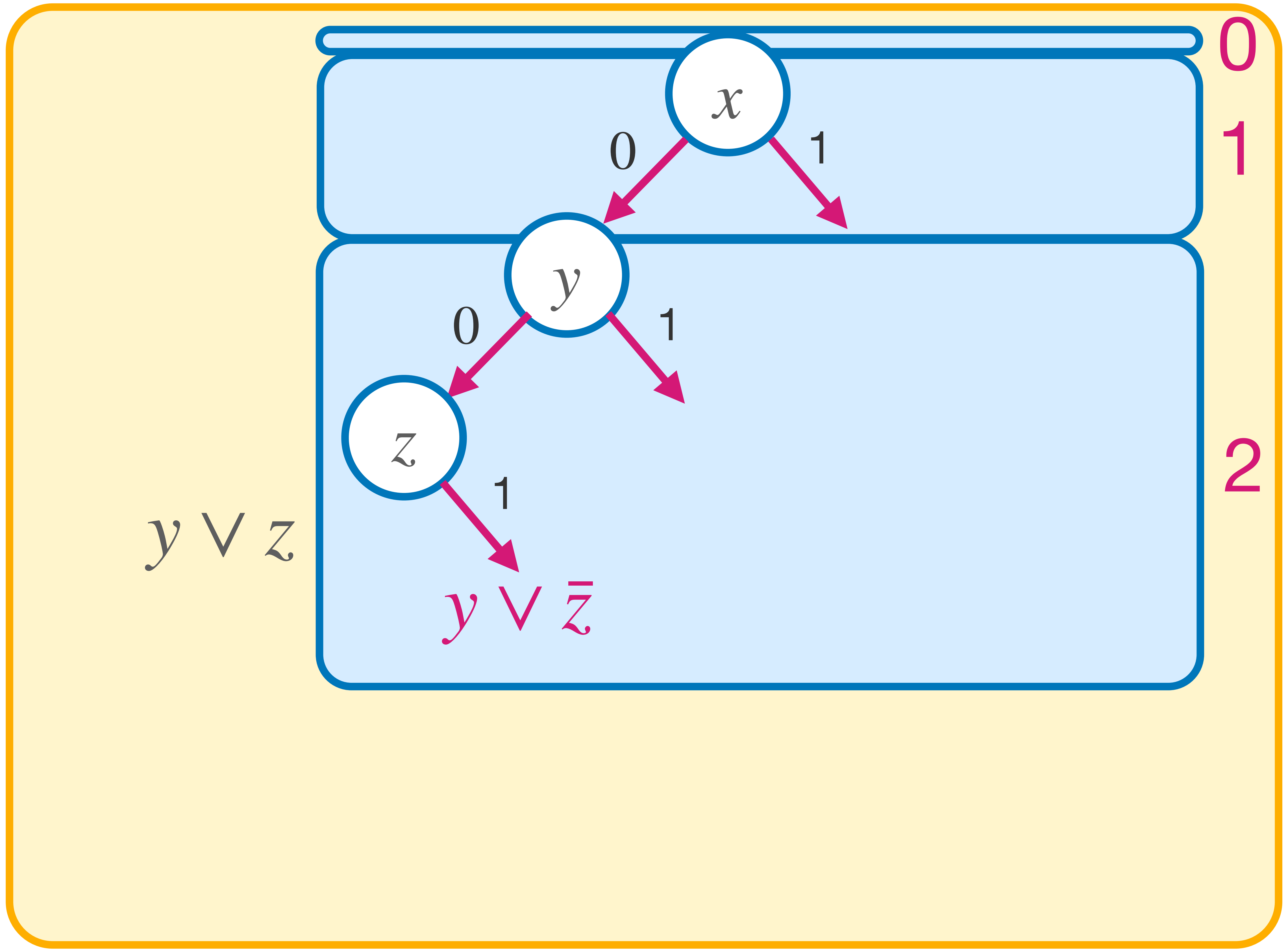
$$(y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

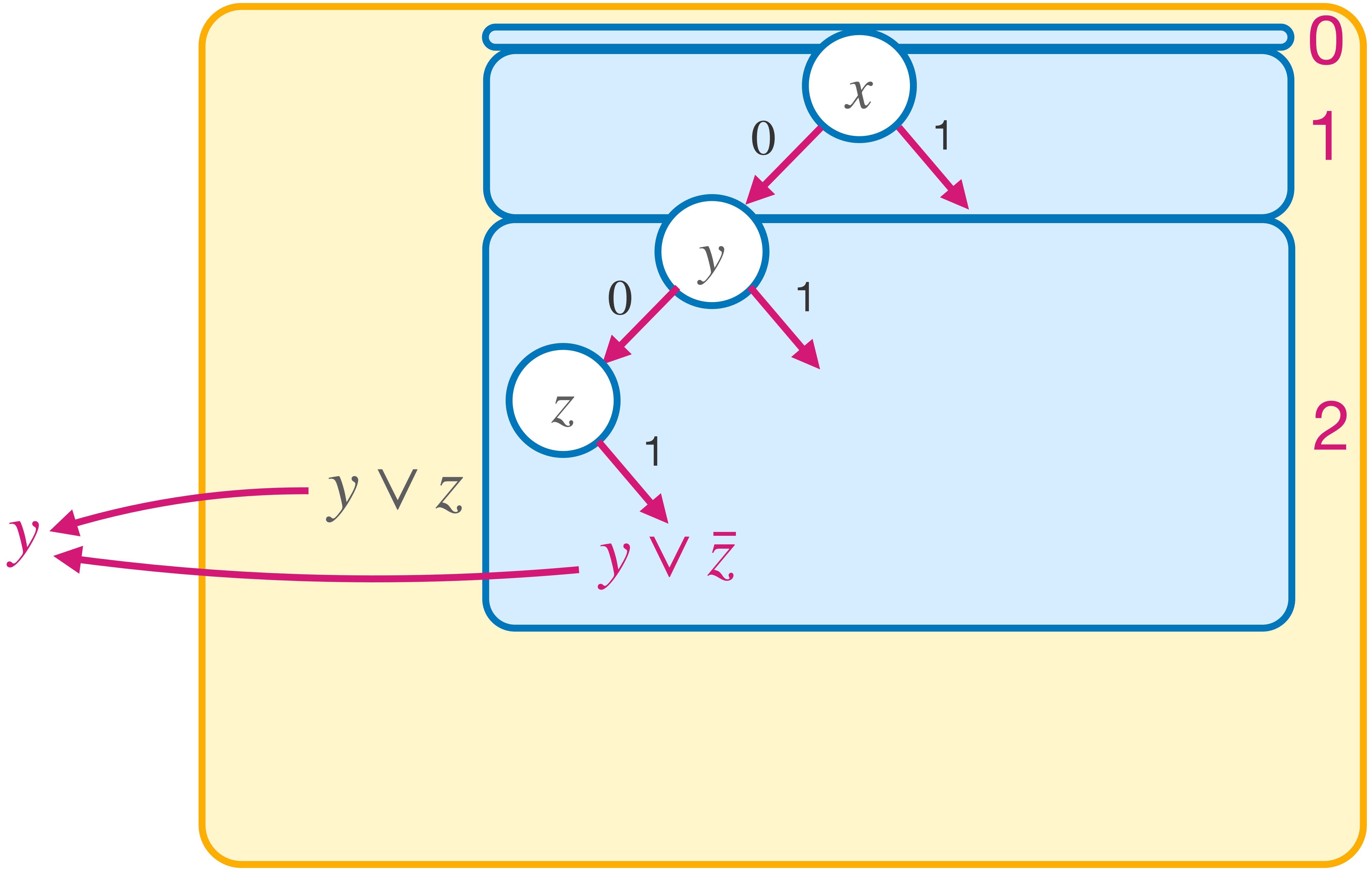
$$(y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

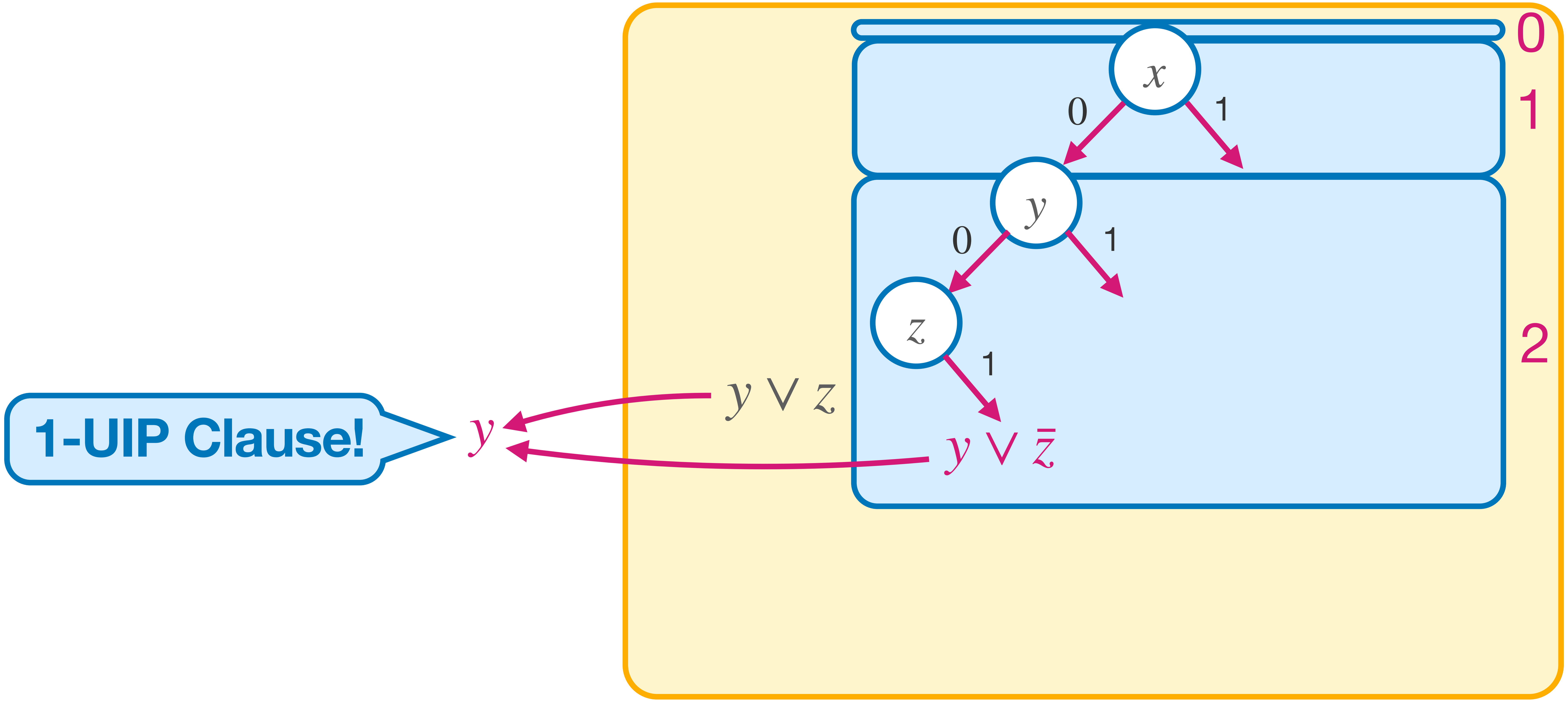
$$(y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

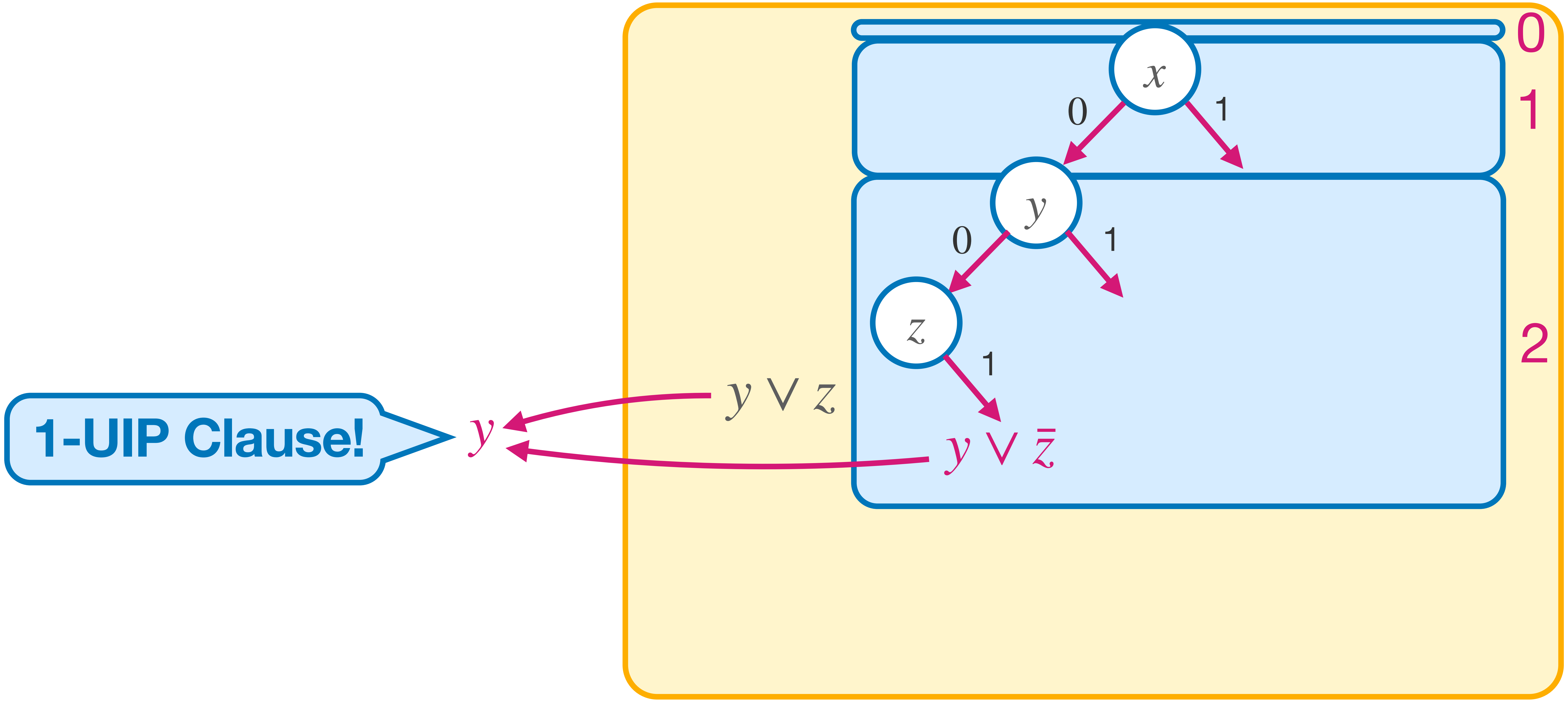
$$(y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

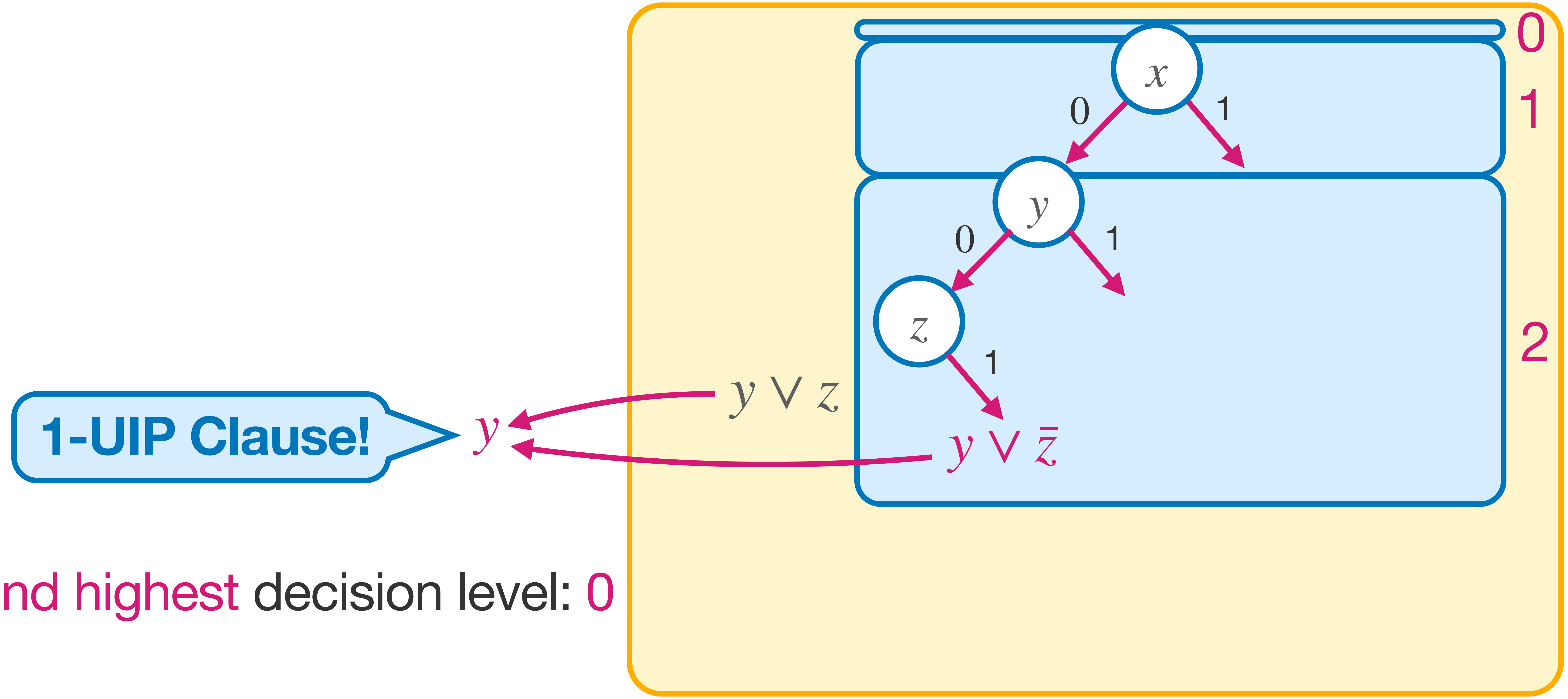
$$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

$$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$

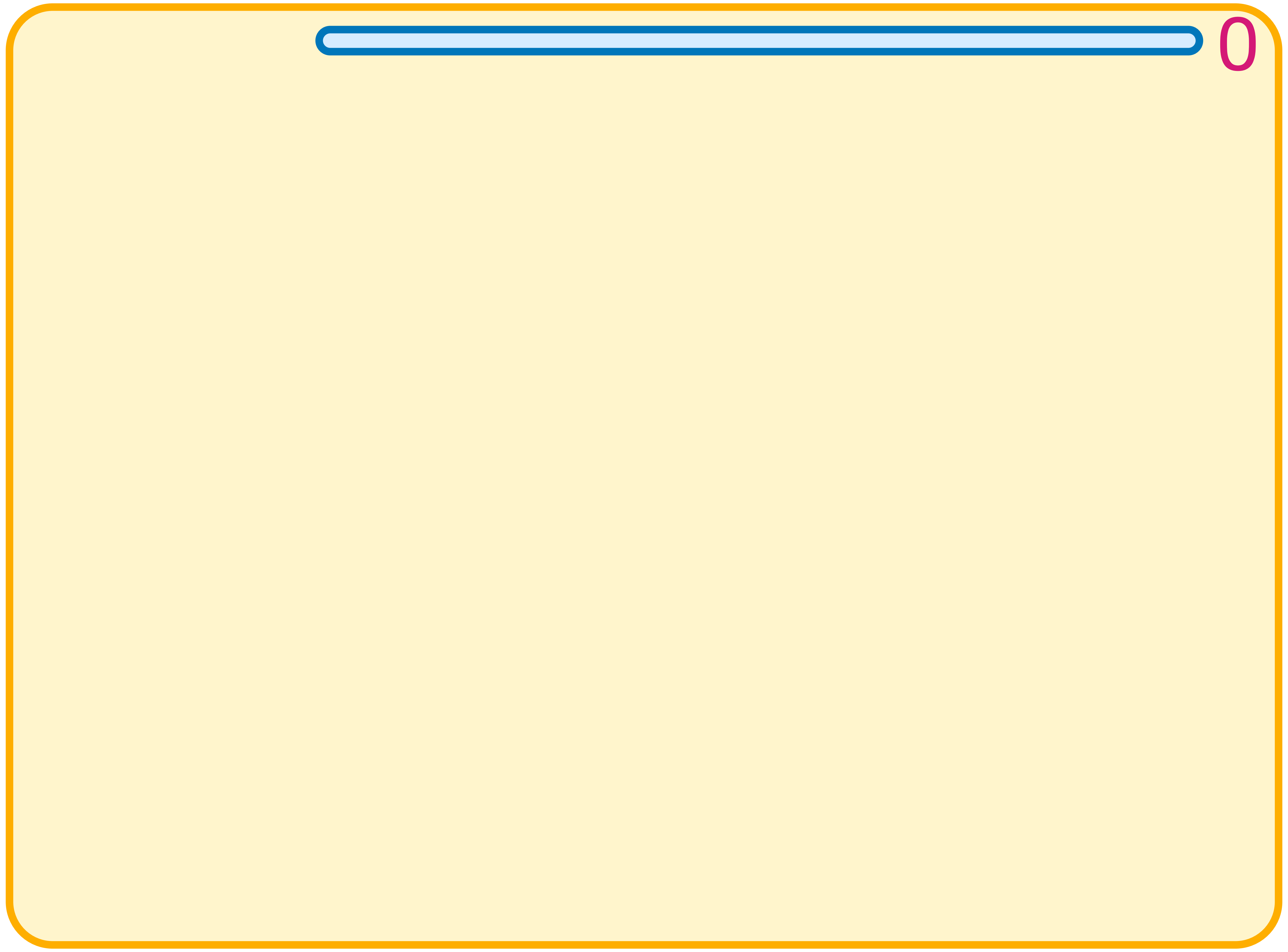


Backjump to second highest decision level: 0

Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

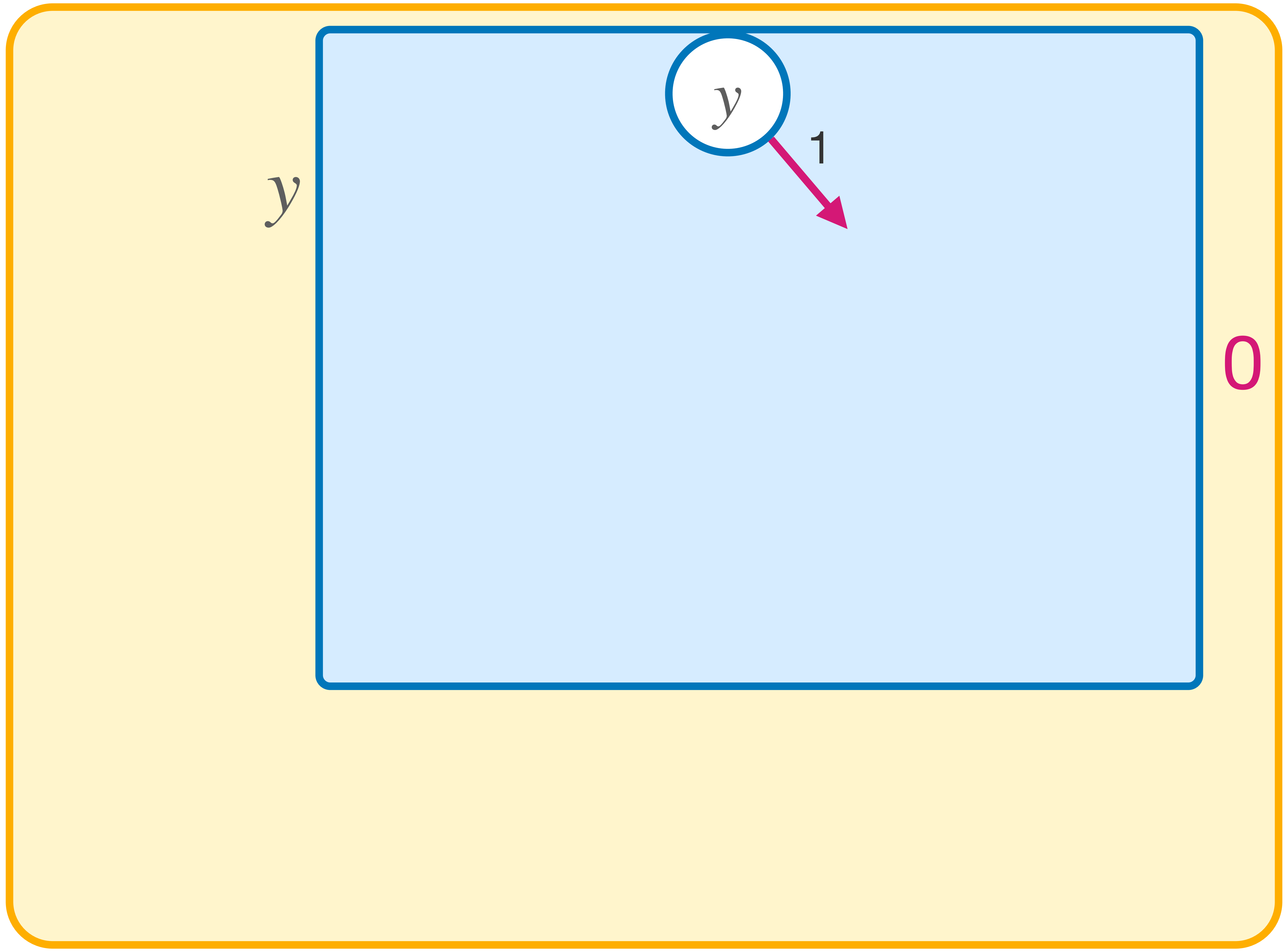
$$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

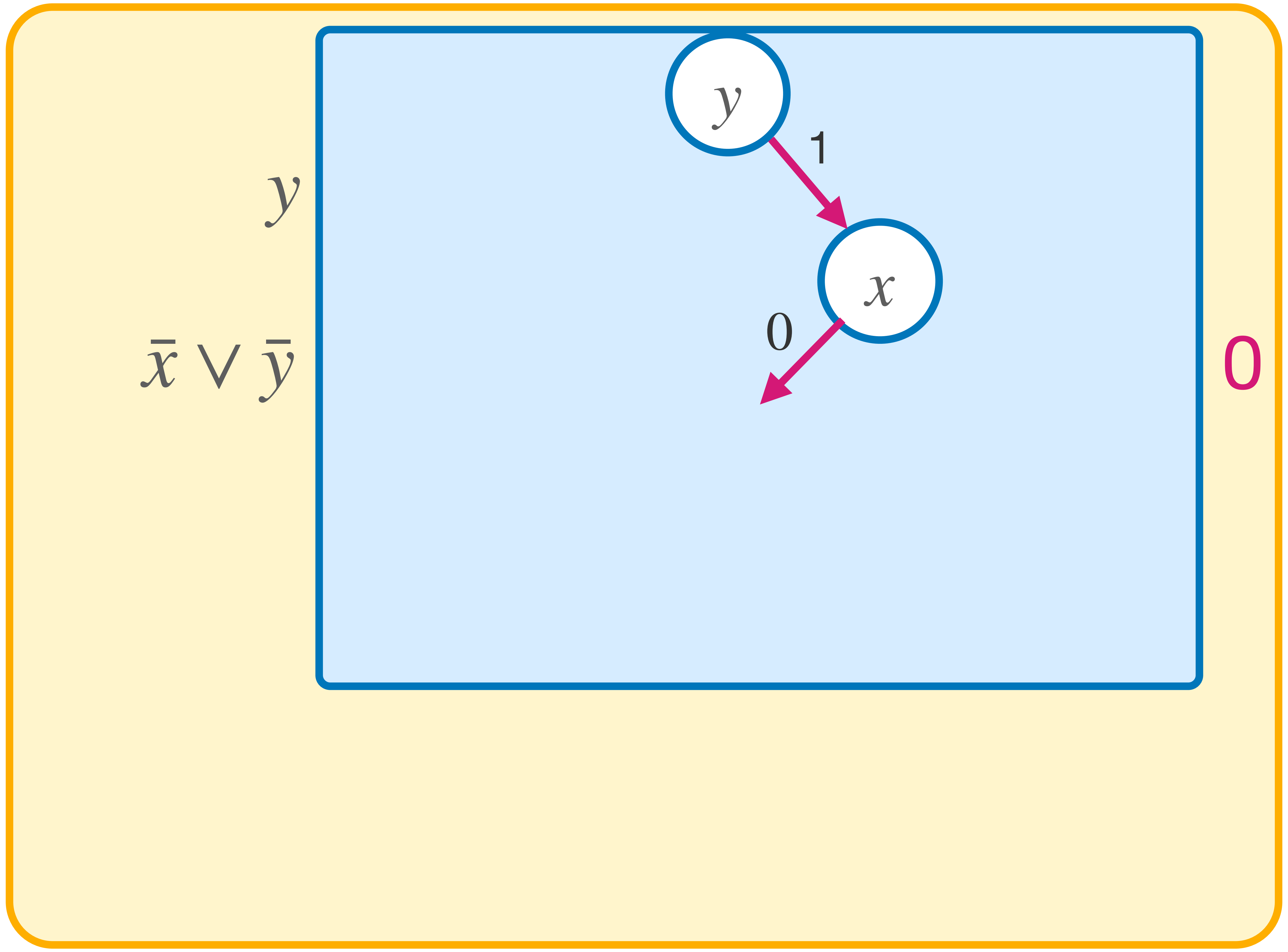
$$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

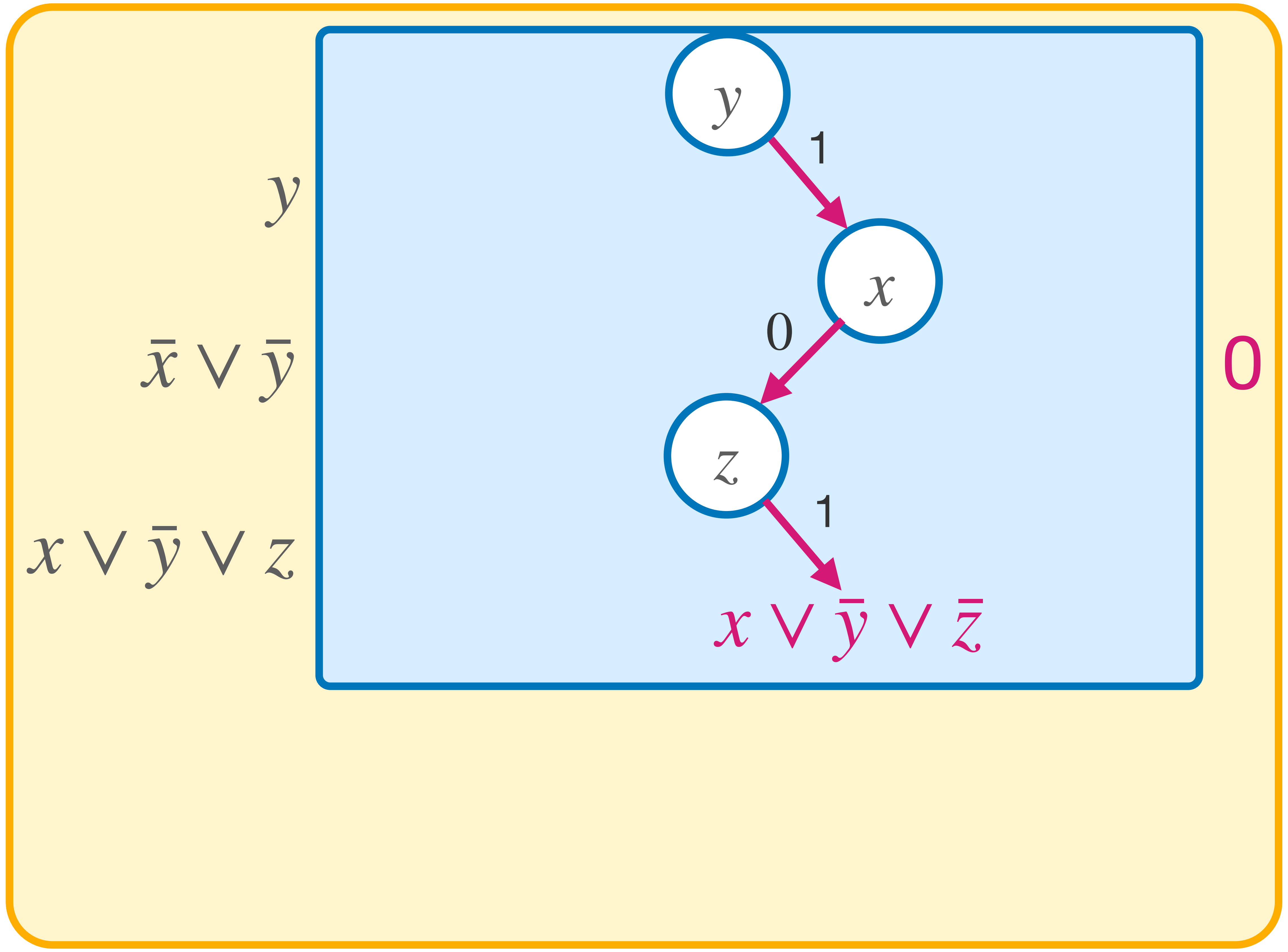
$$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

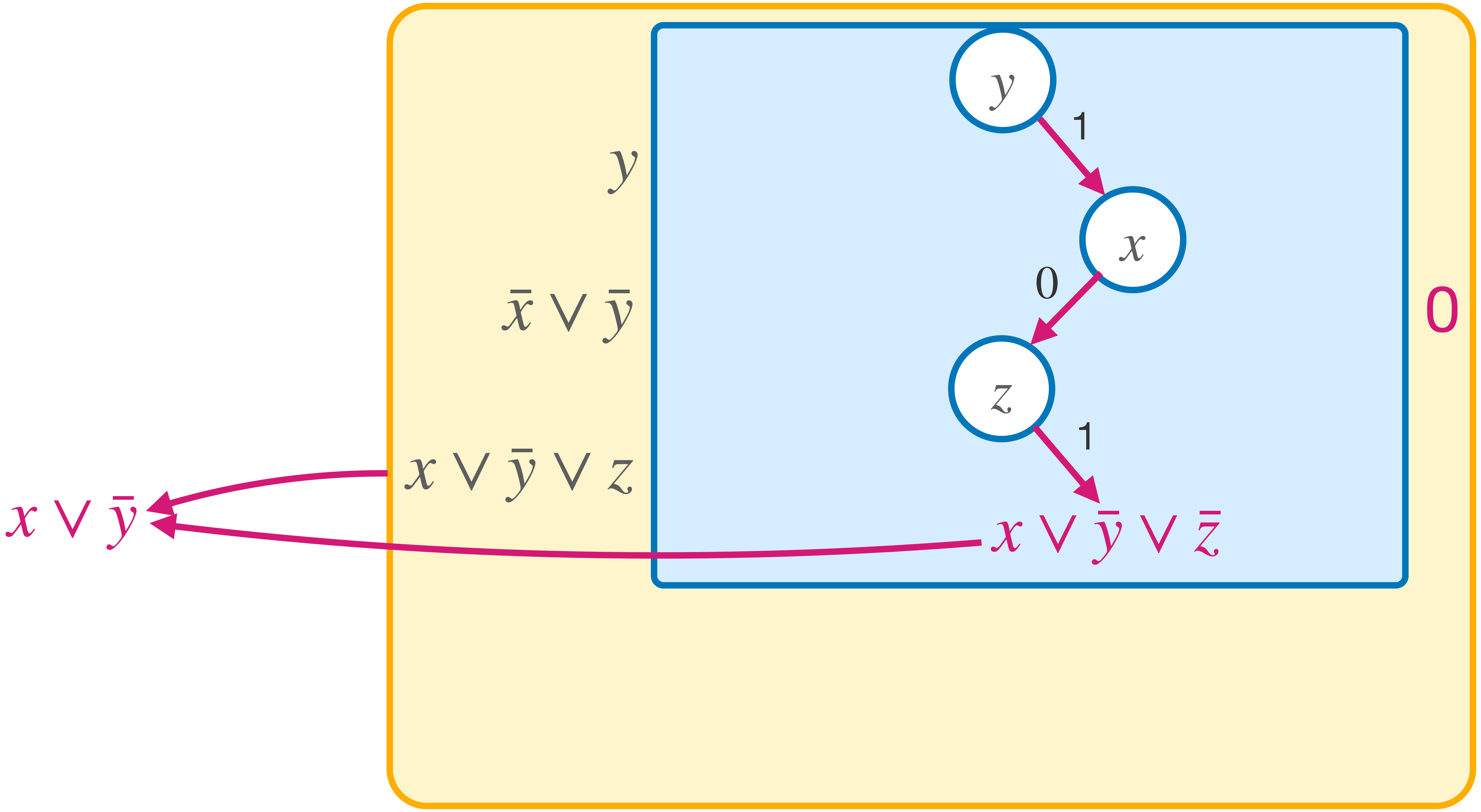
$$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

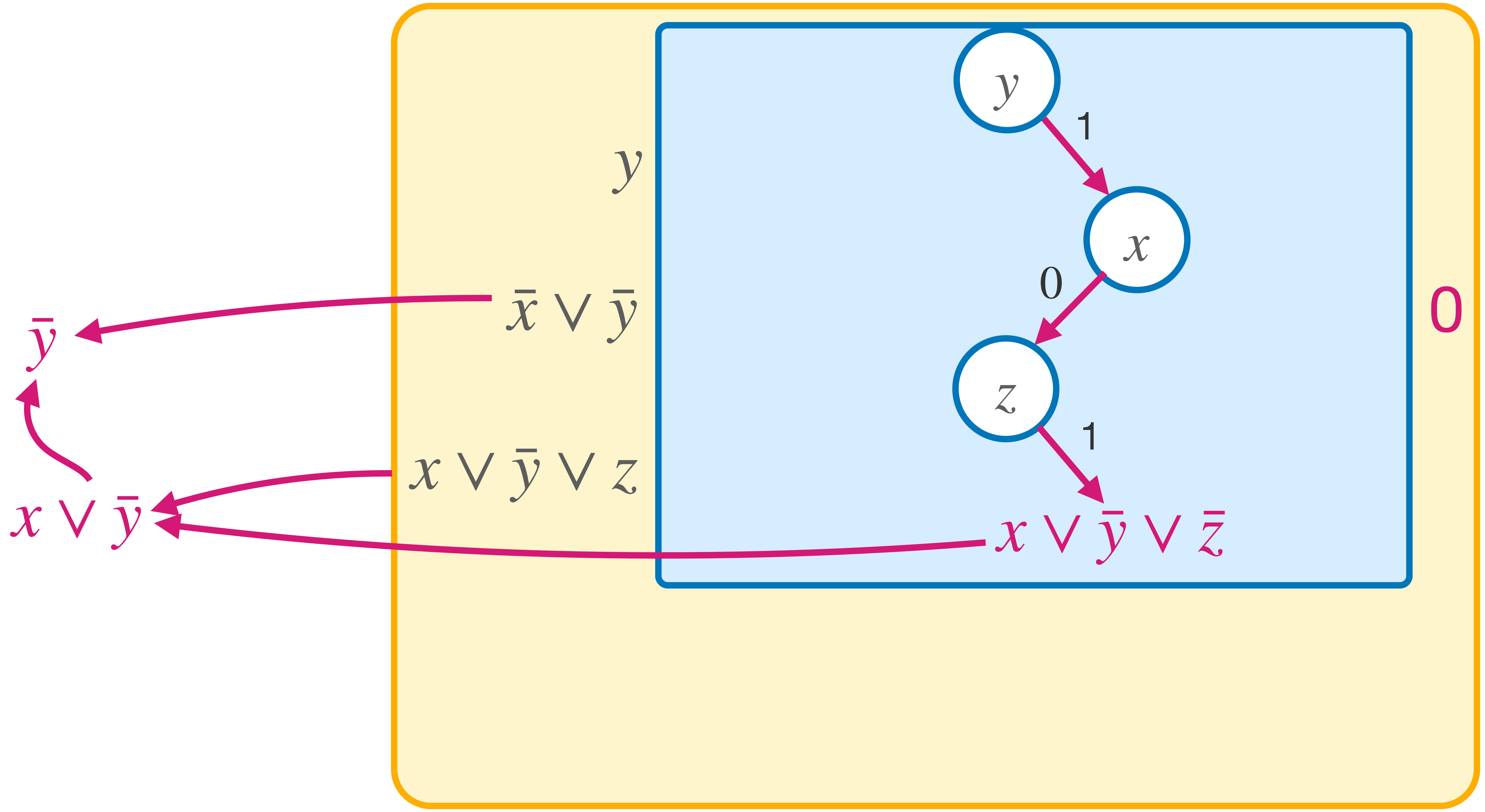
$$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

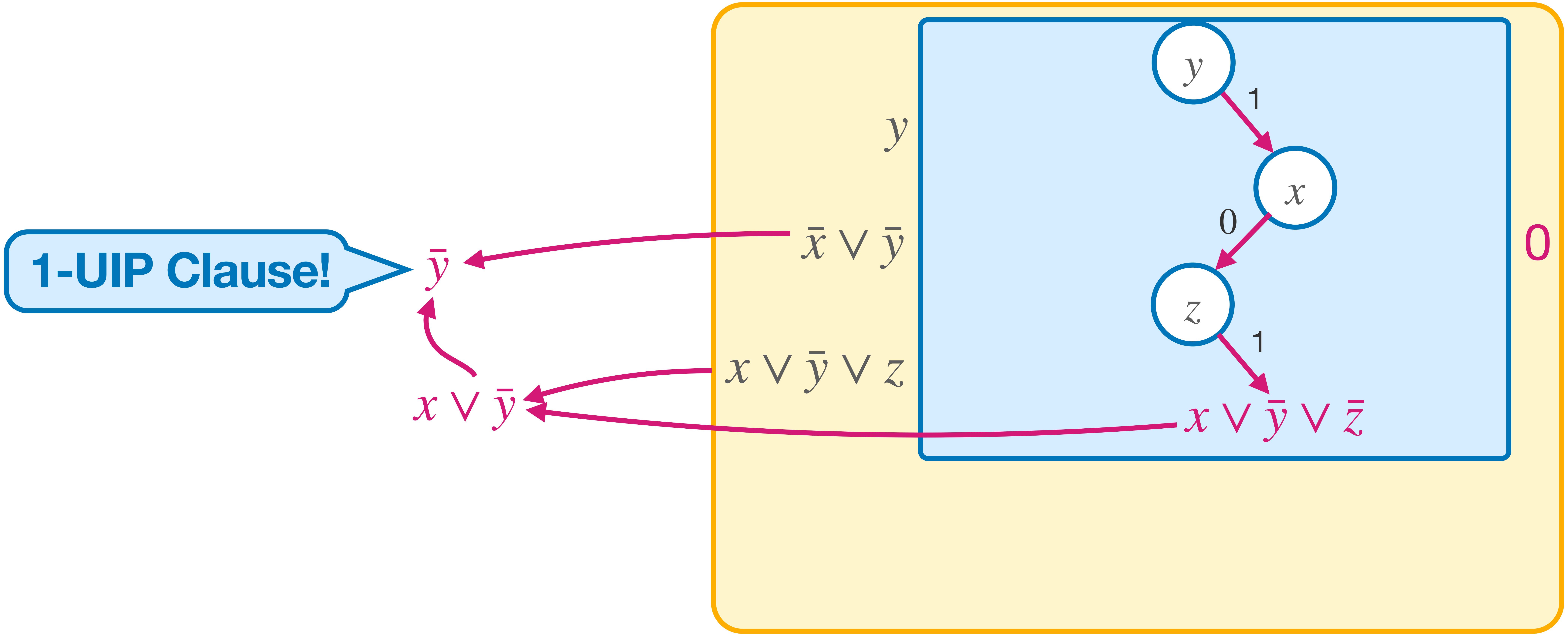
$$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

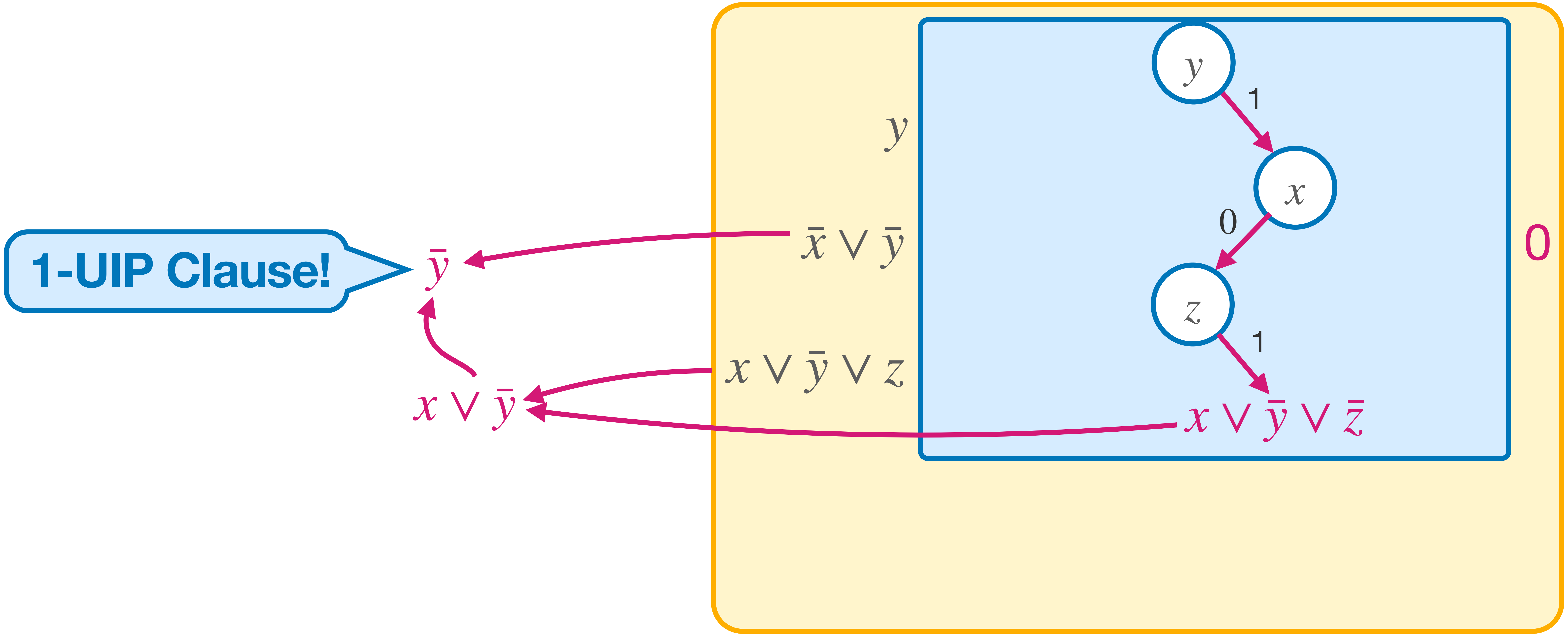
$$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

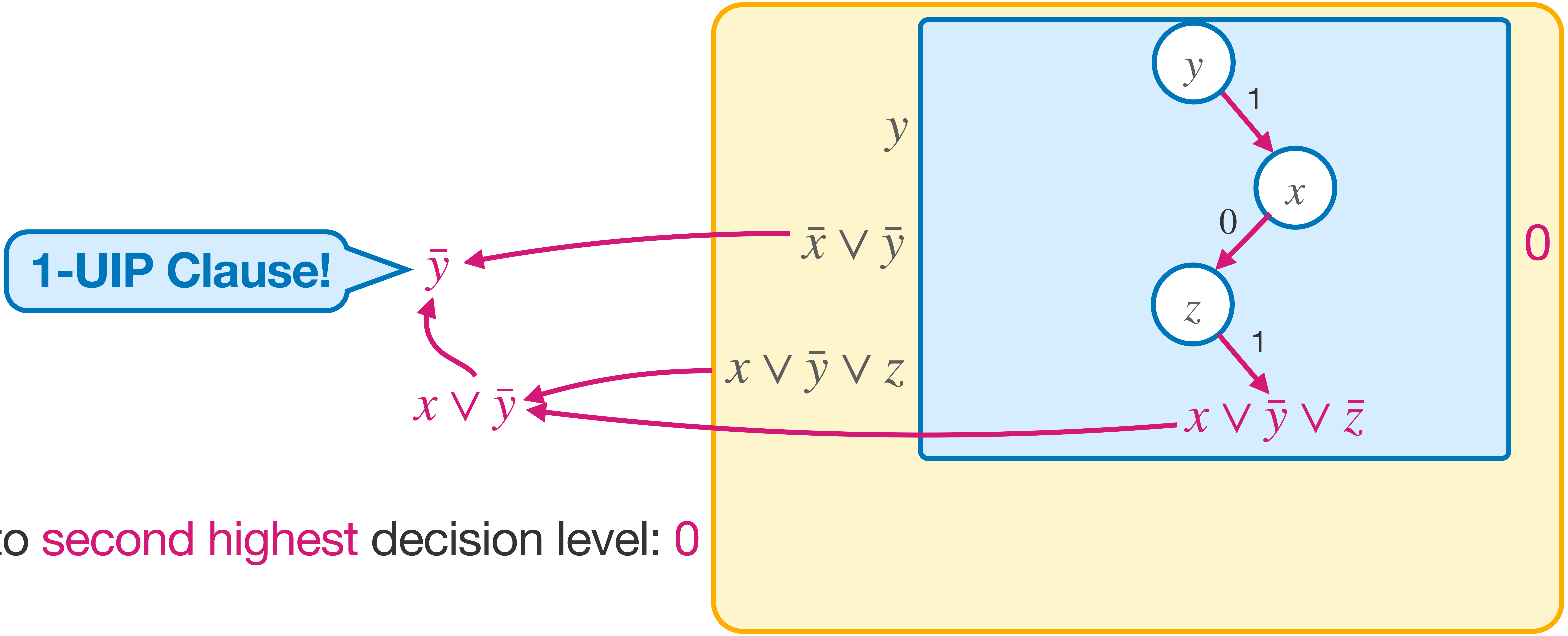
$$(\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

$$(\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$

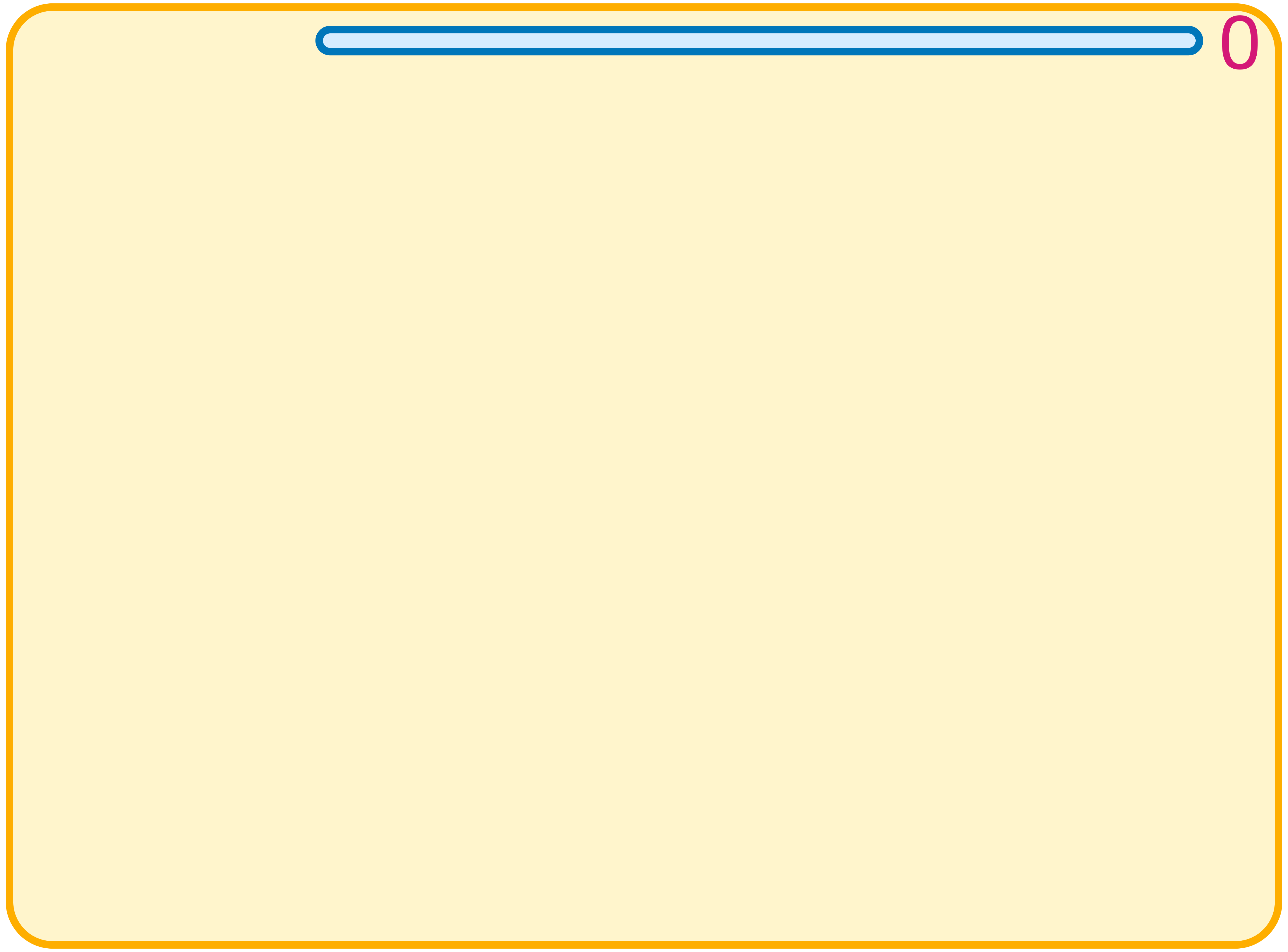


Backjump to second highest decision level: 0

Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

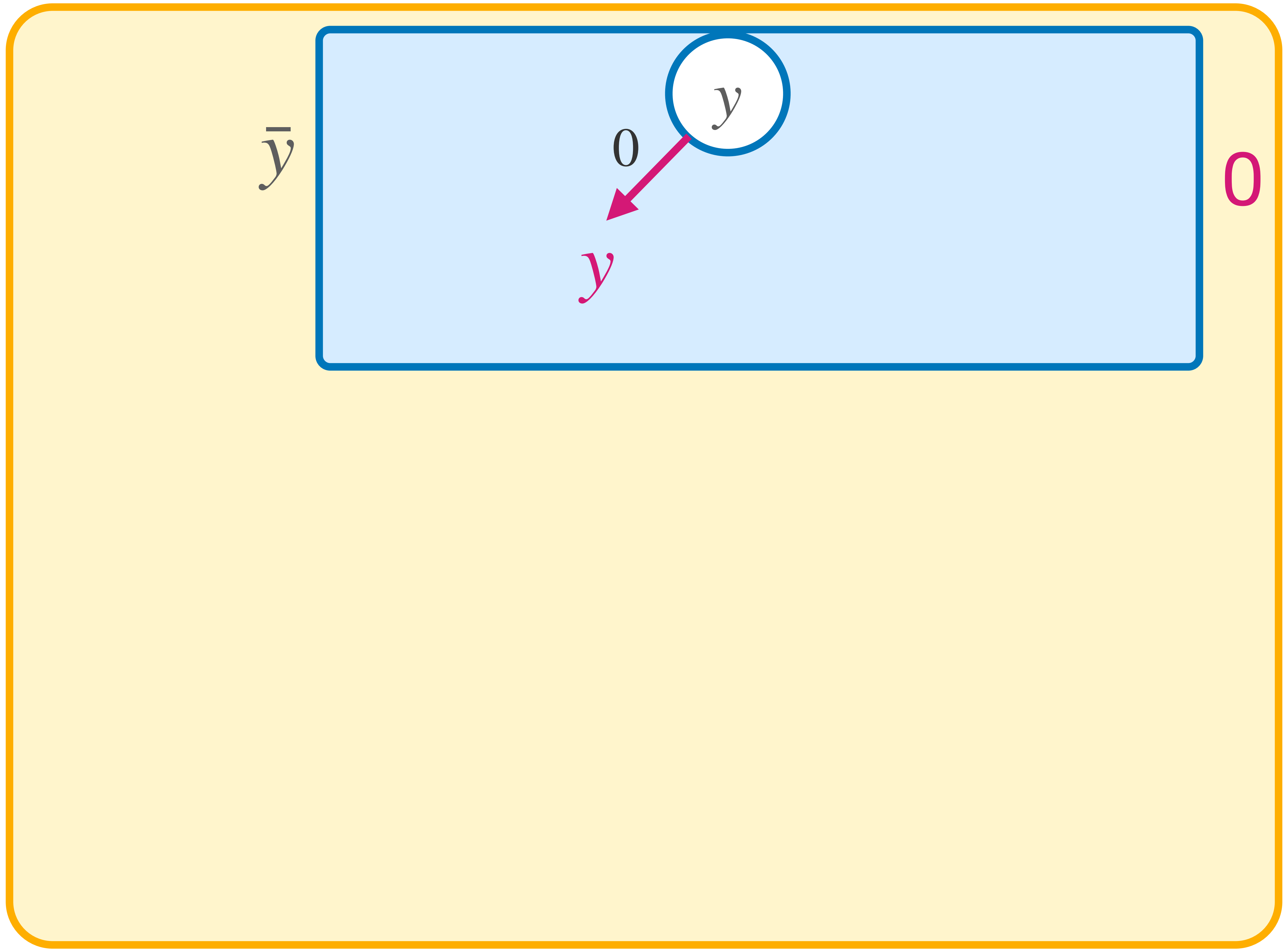
$$(\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

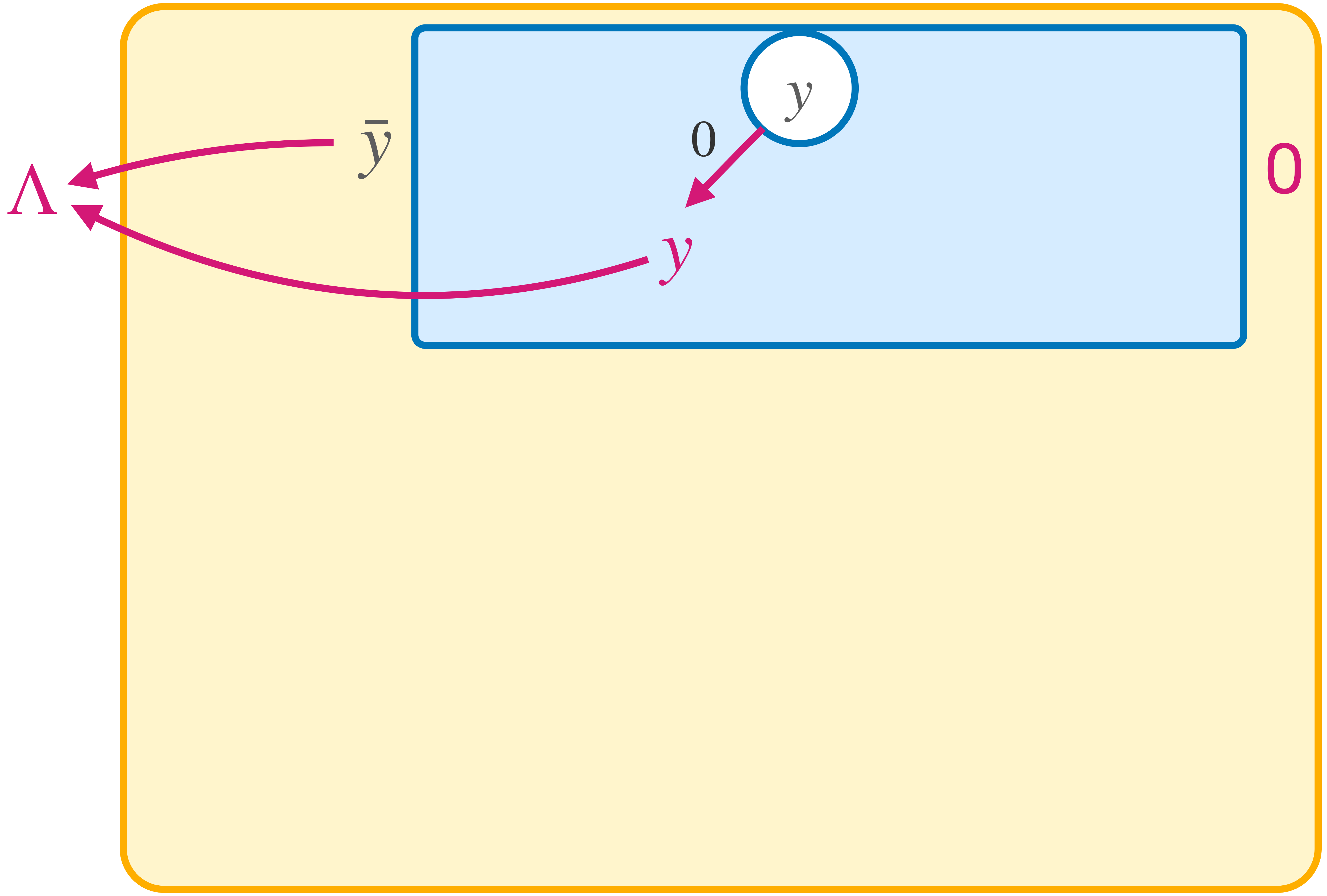
$$(\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

$$\Lambda \wedge (\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$



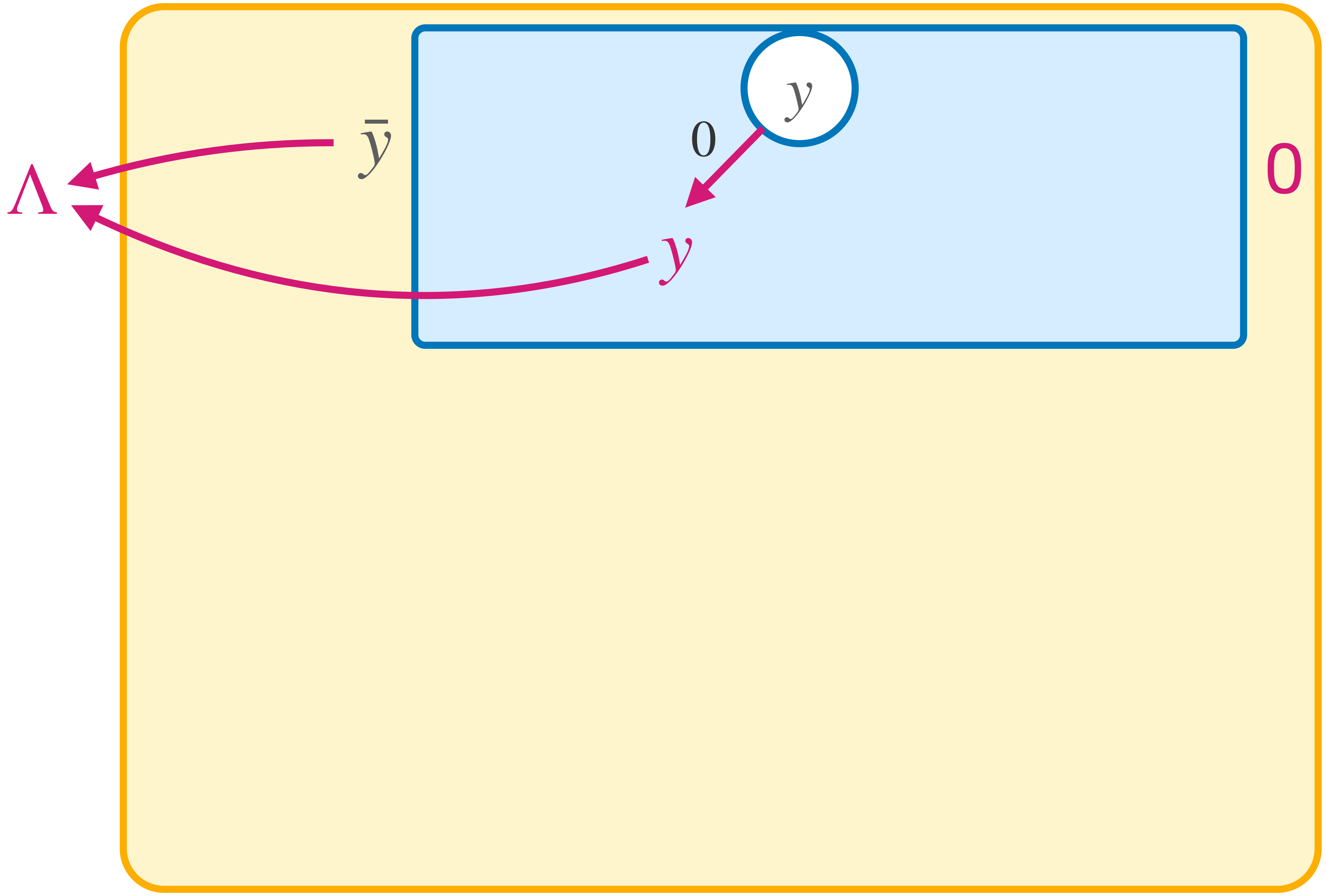
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

$$\Lambda \wedge (\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$

Learned empty clause Λ.

Halt: Unsatisfiable!



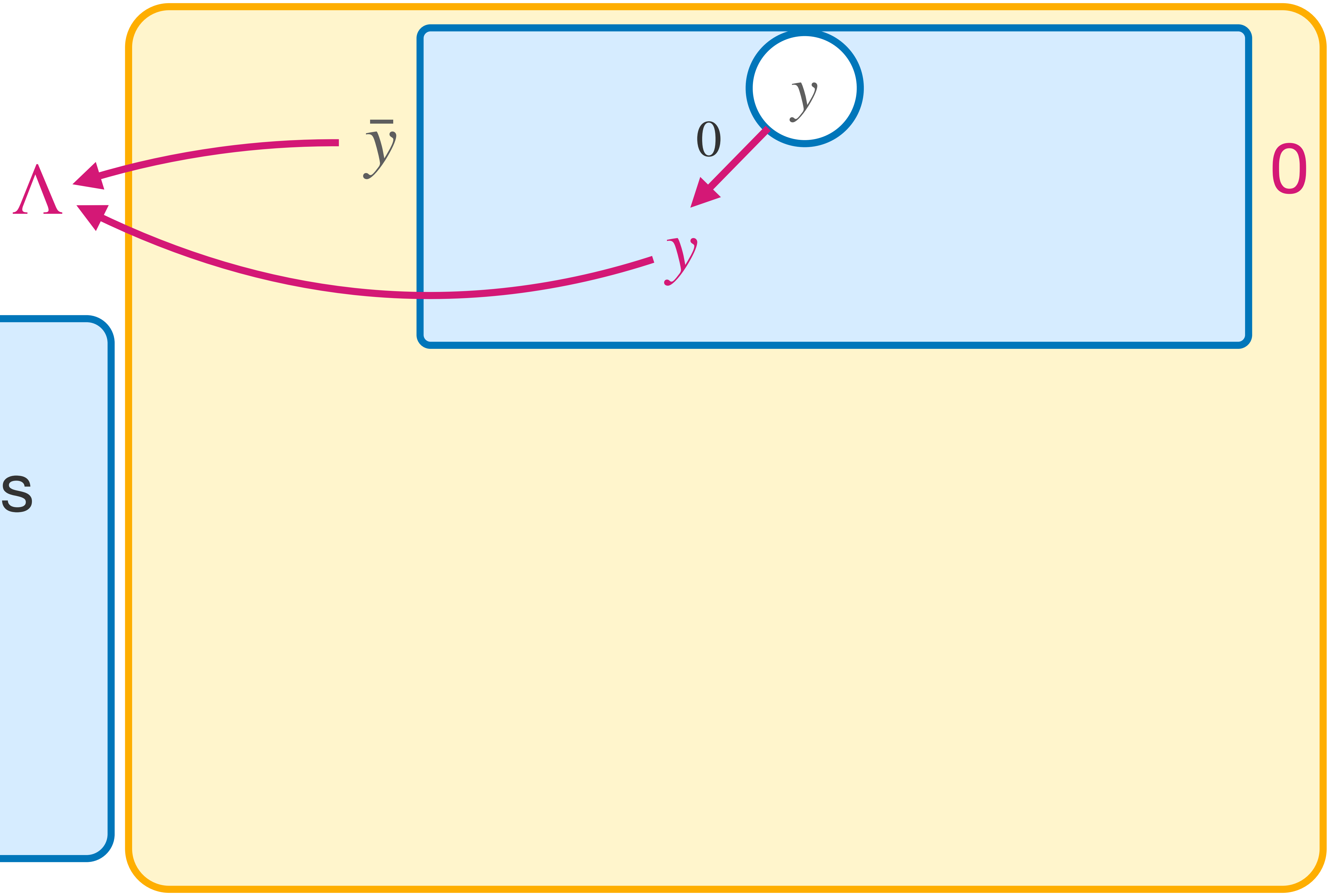
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

$$\Lambda \wedge (\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$

Learned empty clause Λ.

Halt: Unsatisfiable!



Takeaway:

CDCL run on an unsatisfiable formula halts when Λ is derived from clause learning

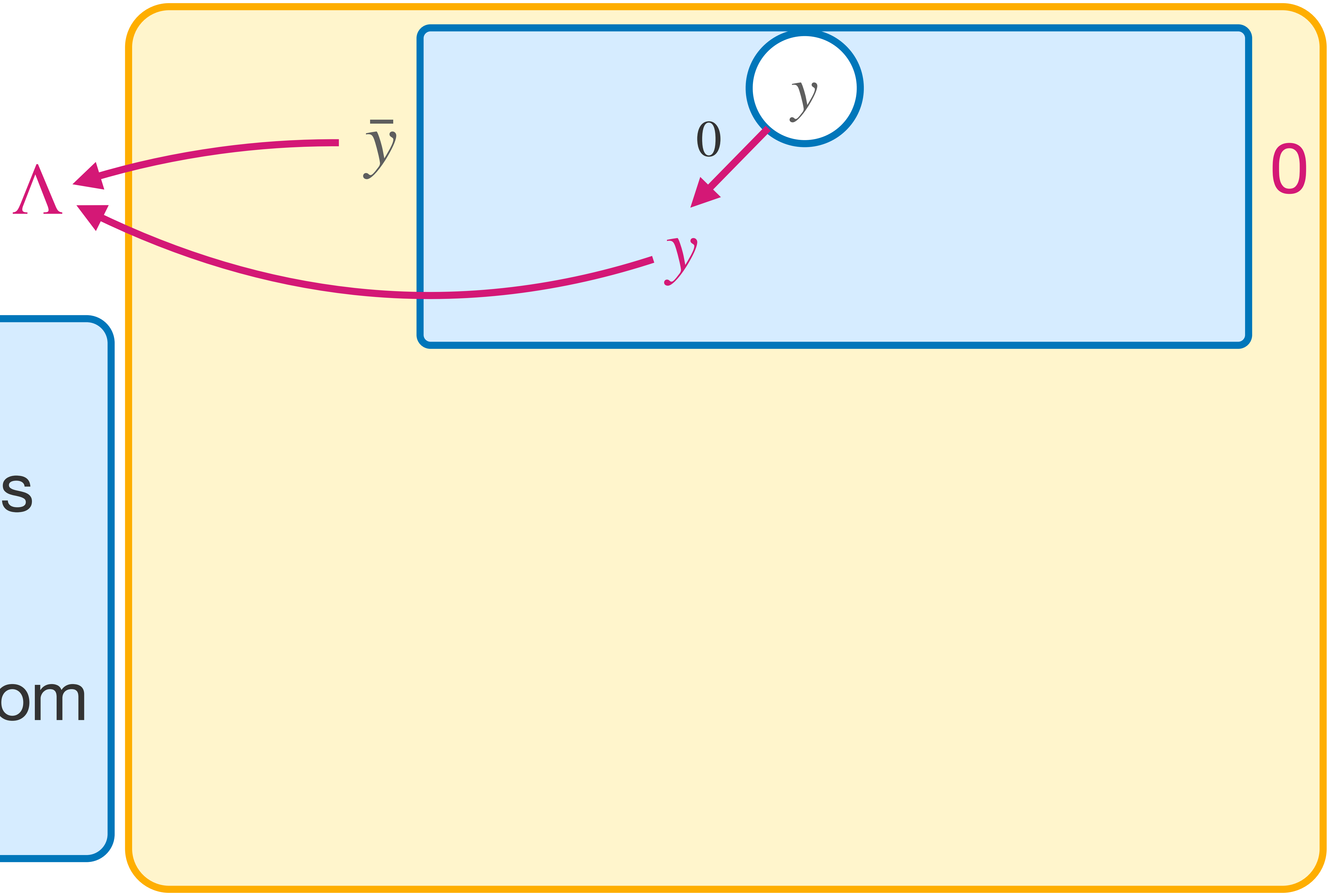
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

$$\Lambda \wedge (\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$

Learned empty clause Λ.

Halt: Unsatisfiable!



Takeaway:

CDCL run on an unsatisfiable formula halts when Λ is derived from clause learning

$\rightarrow$ Clause learning derives new clauses from old ones using Resolution

Analyzing CDCL

Theorem: Let F be an unsatisfiable CNF formula. If CDCL takes time s to solve F, then there is a size-s Resolution proof of F

Proof: Every time CDCL learns a clause, derive that clause in Resolution

Analyzing CDCL

Theorem: Let F be an unsatisfiable CNF formula. If CDCL takes time s to solve F, then there is a size-s Resolution proof of F

Proof: Every time CDCL learns a clause, derive that clause in Resolution

$\rightarrow$ Because CDCL halts when Λ is derived, we have a Resolution proof!

Analyzing CDCL

$$(y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$

Resolution

$(y \vee z), (y \vee \bar{z}), (x \vee \bar{y} \vee z), (x \vee \bar{y} \vee \bar{z}), (\bar{x} \vee \bar{y})$

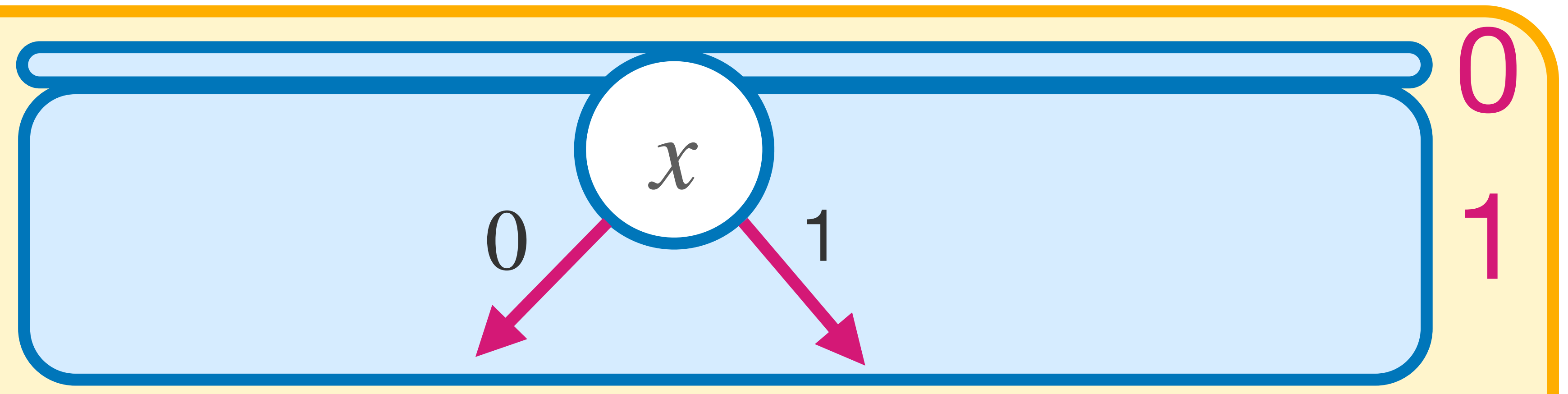
0

Analyzing CDCL

$$(y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})$$

Resolution

$(y \lor z), (y \lor \bar{z}), (x \lor \bar{y} \lor z), (x \lor \bar{y} \lor \bar{z}), (\bar{x} \lor \bar{y})$

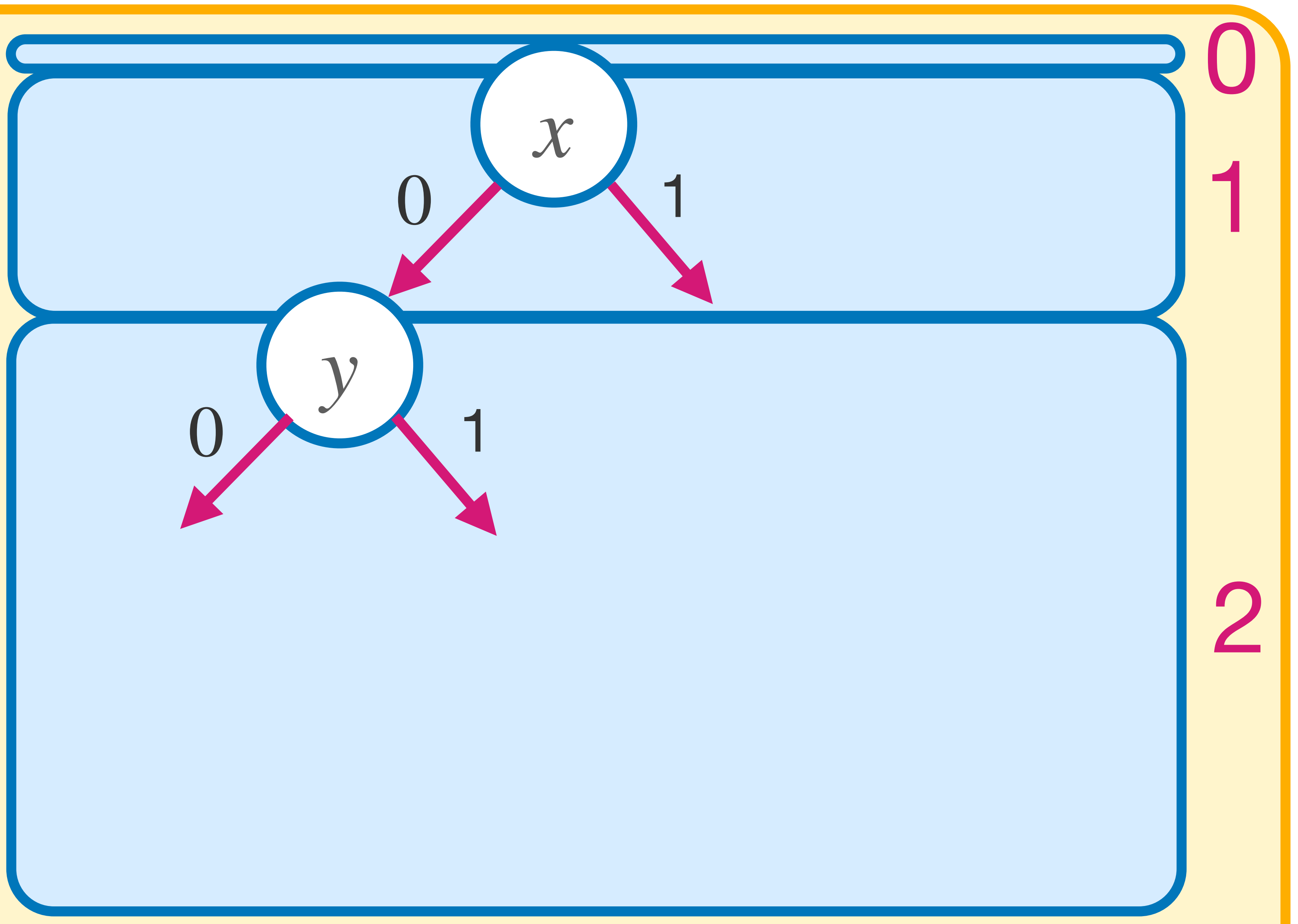


Analyzing CDCL

$$(y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$

Resolution

$(y \vee z), (y \vee \bar{z}), (x \vee \bar{y} \vee z), (x \vee \bar{y} \vee \bar{z}), (\bar{x} \vee \bar{y})$

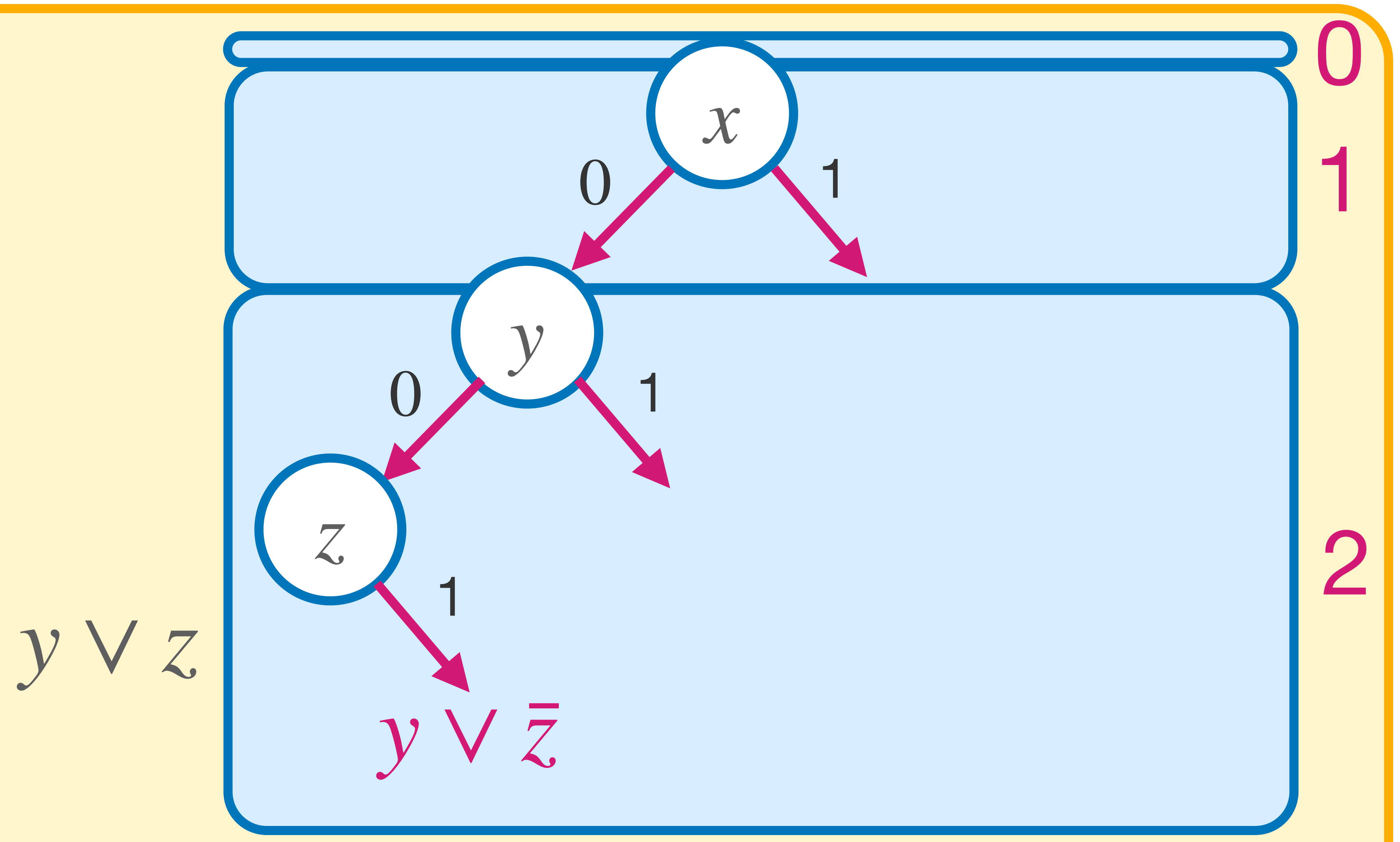


Analyzing CDCL

$$(y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$

Resolution

$(y \vee z), (y \vee \bar{z}), (x \vee \bar{y} \vee z), (x \vee \bar{y} \vee \bar{z}), (\bar{x} \vee \bar{y})$

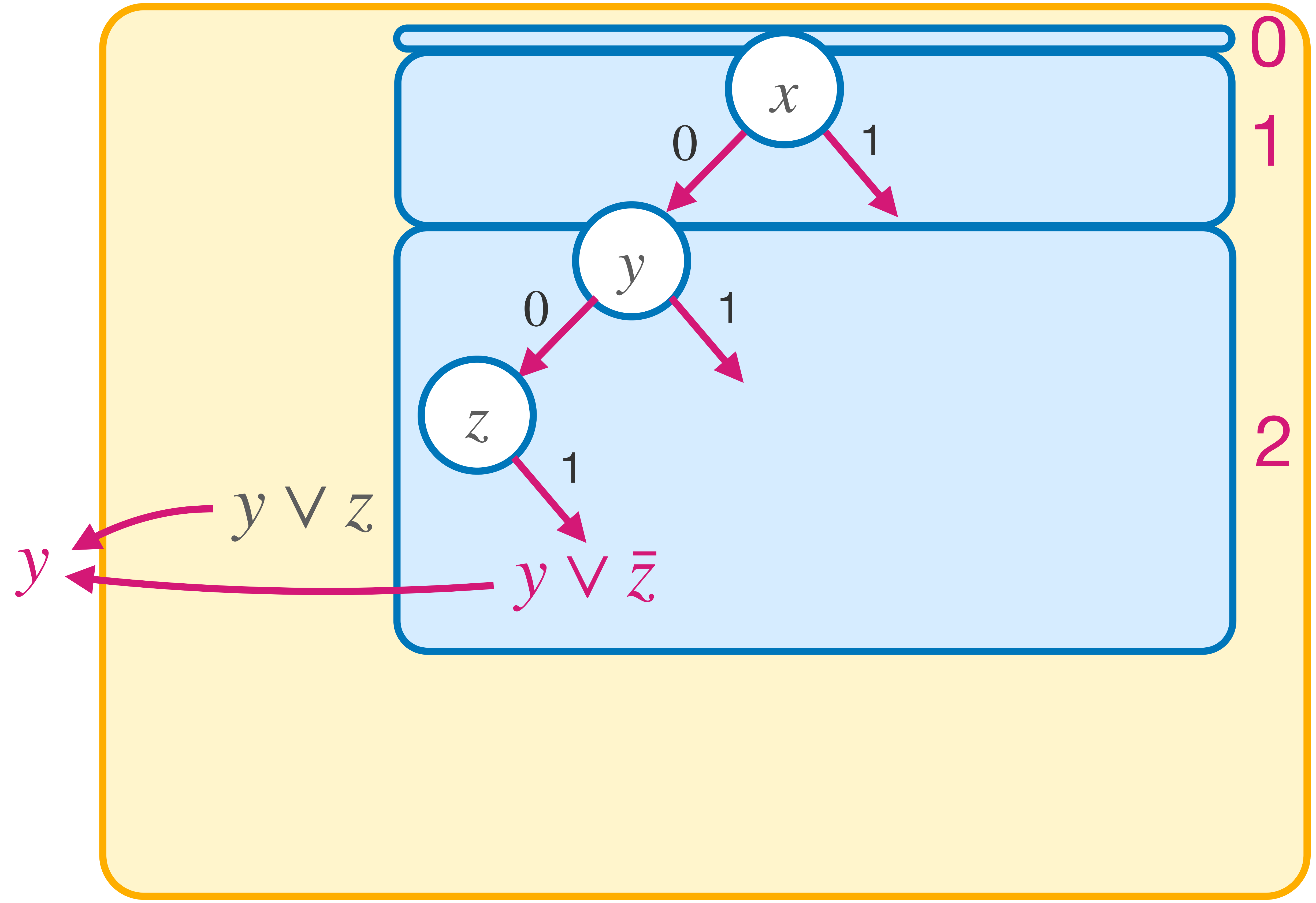


Analyzing CDCL

$$(y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$

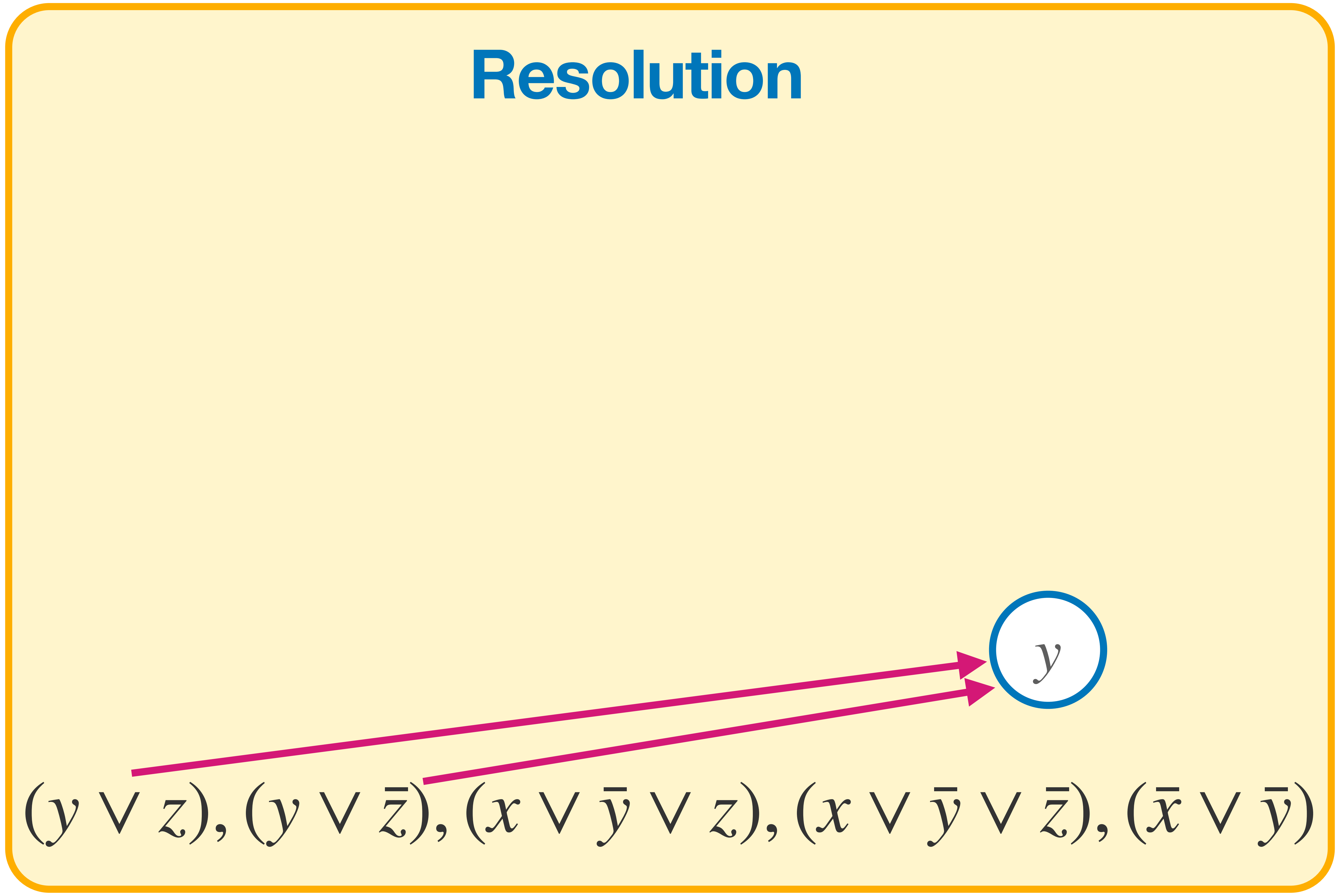
Resolution

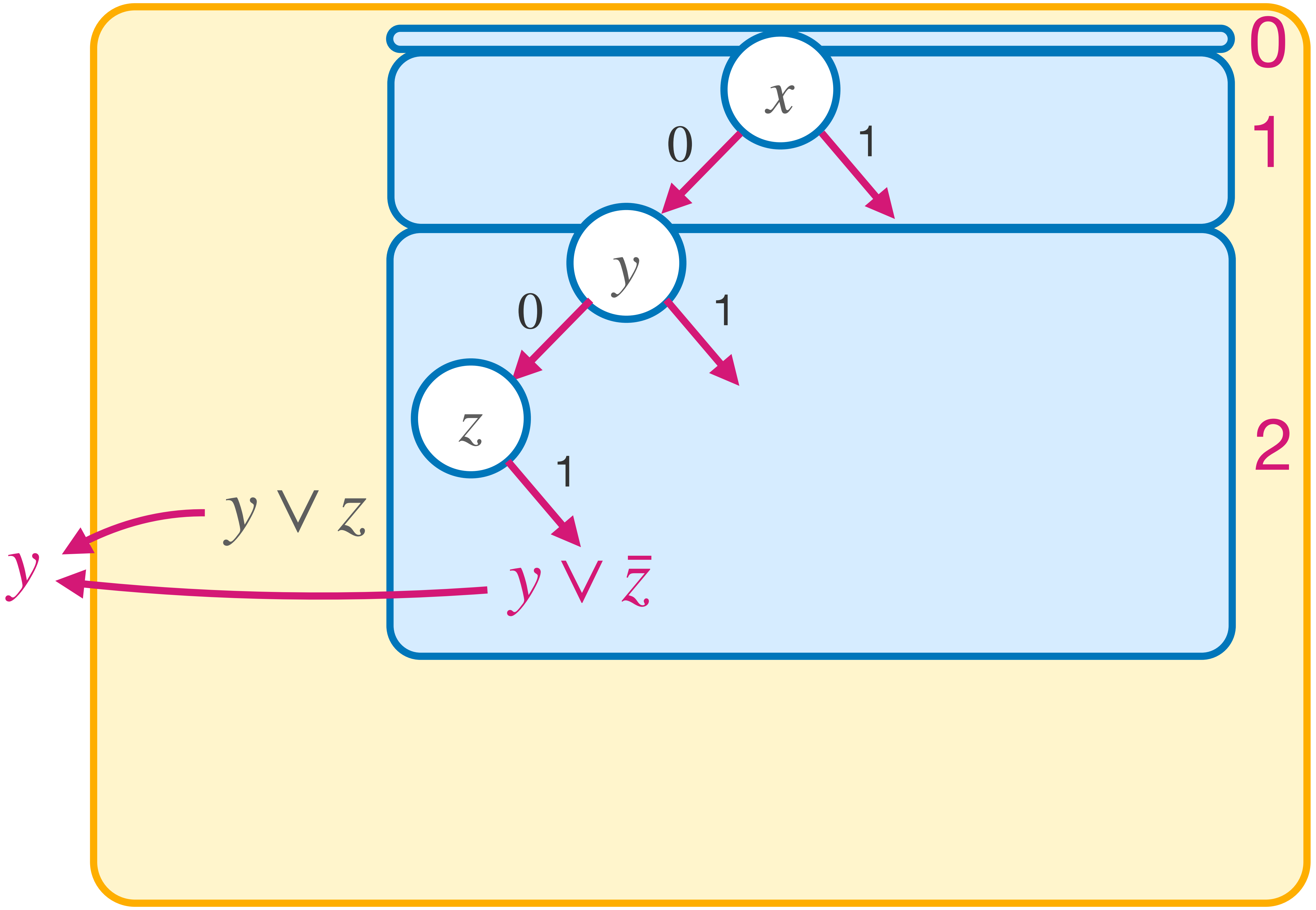
$(y \vee z), (y \vee \bar{z}), (x \vee \bar{y} \vee z), (x \vee \bar{y} \vee \bar{z}), (\bar{x} \vee \bar{y})$



Analyzing CDCL

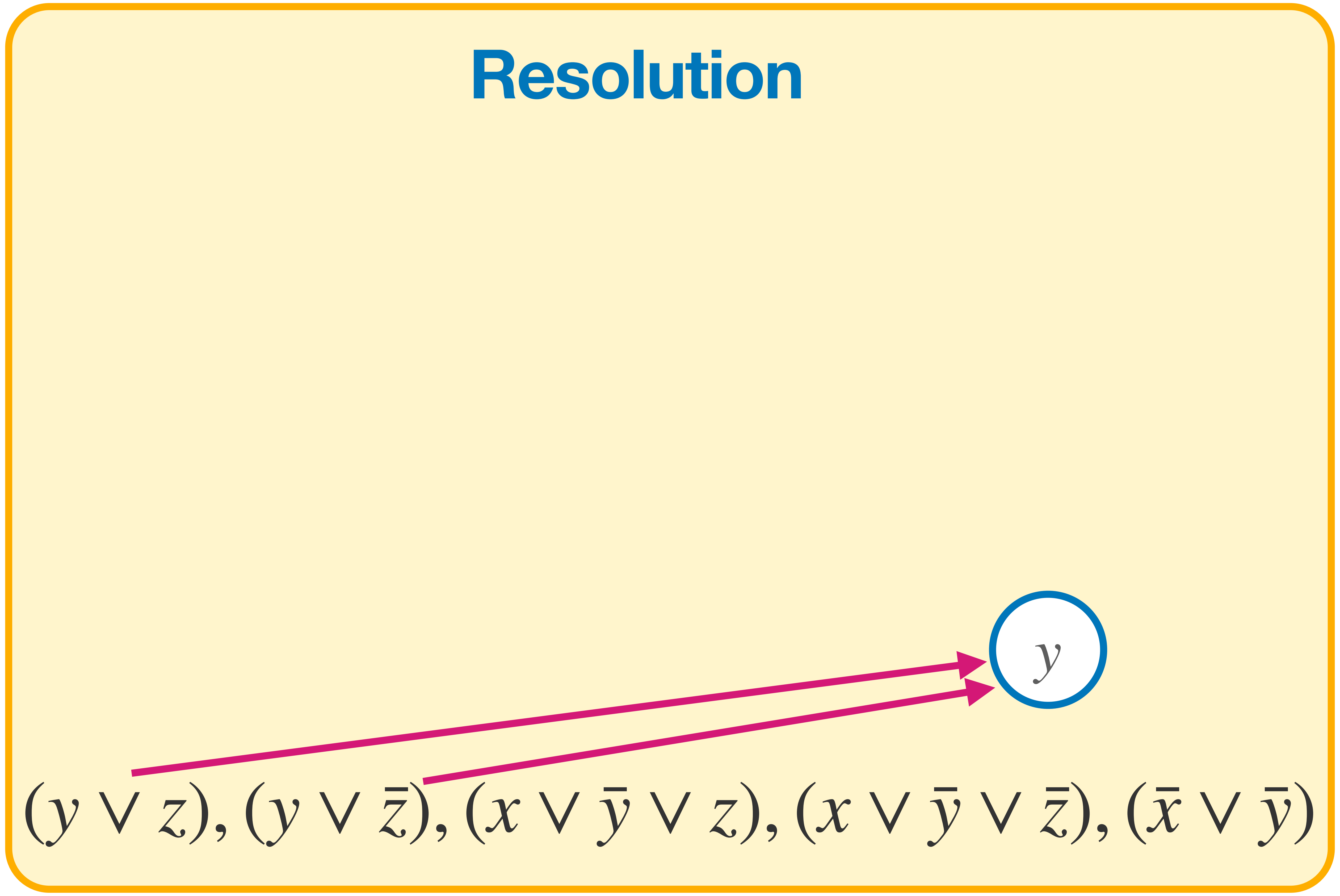
$$(y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})$$

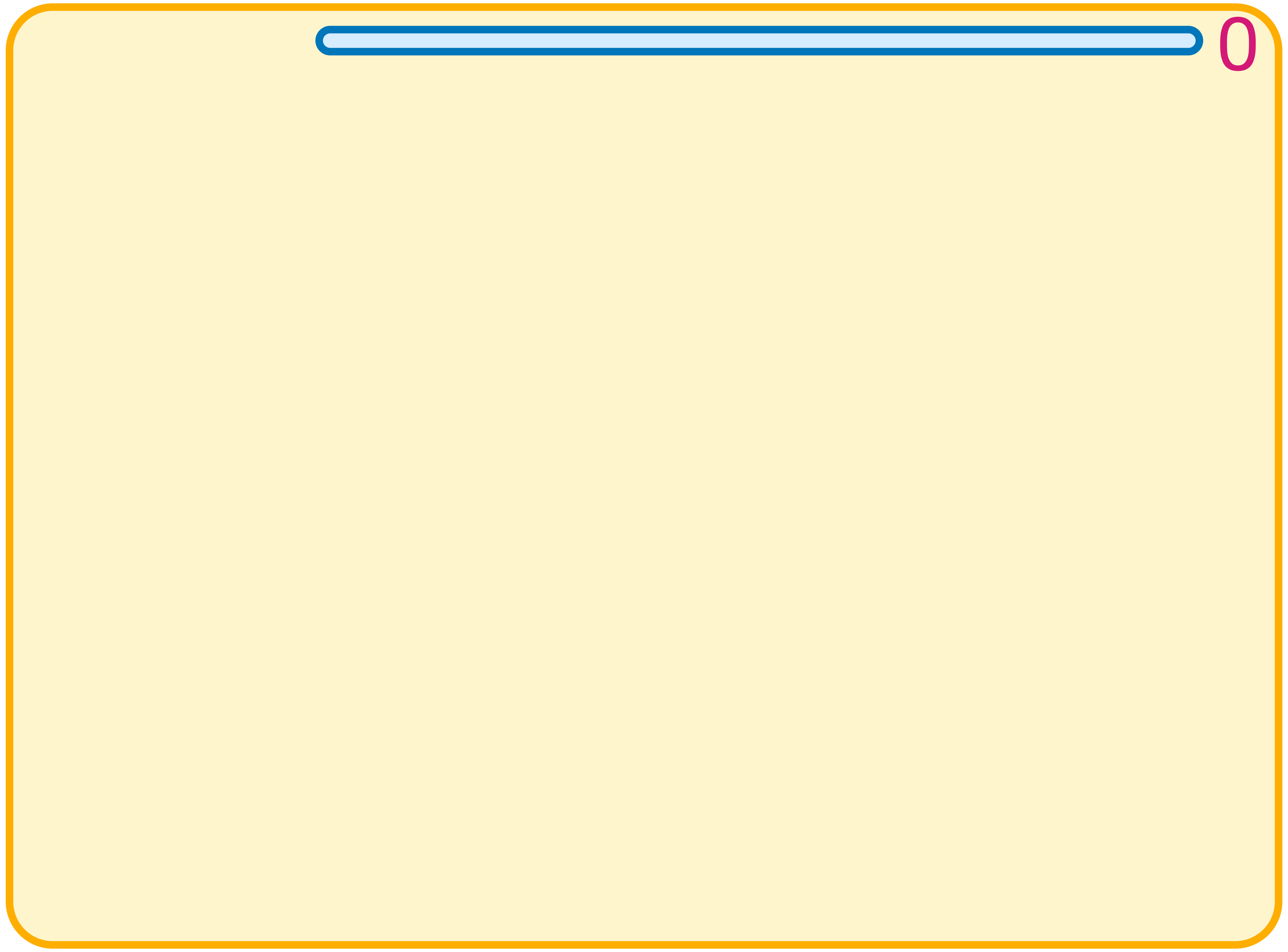




Analyzing CDCL

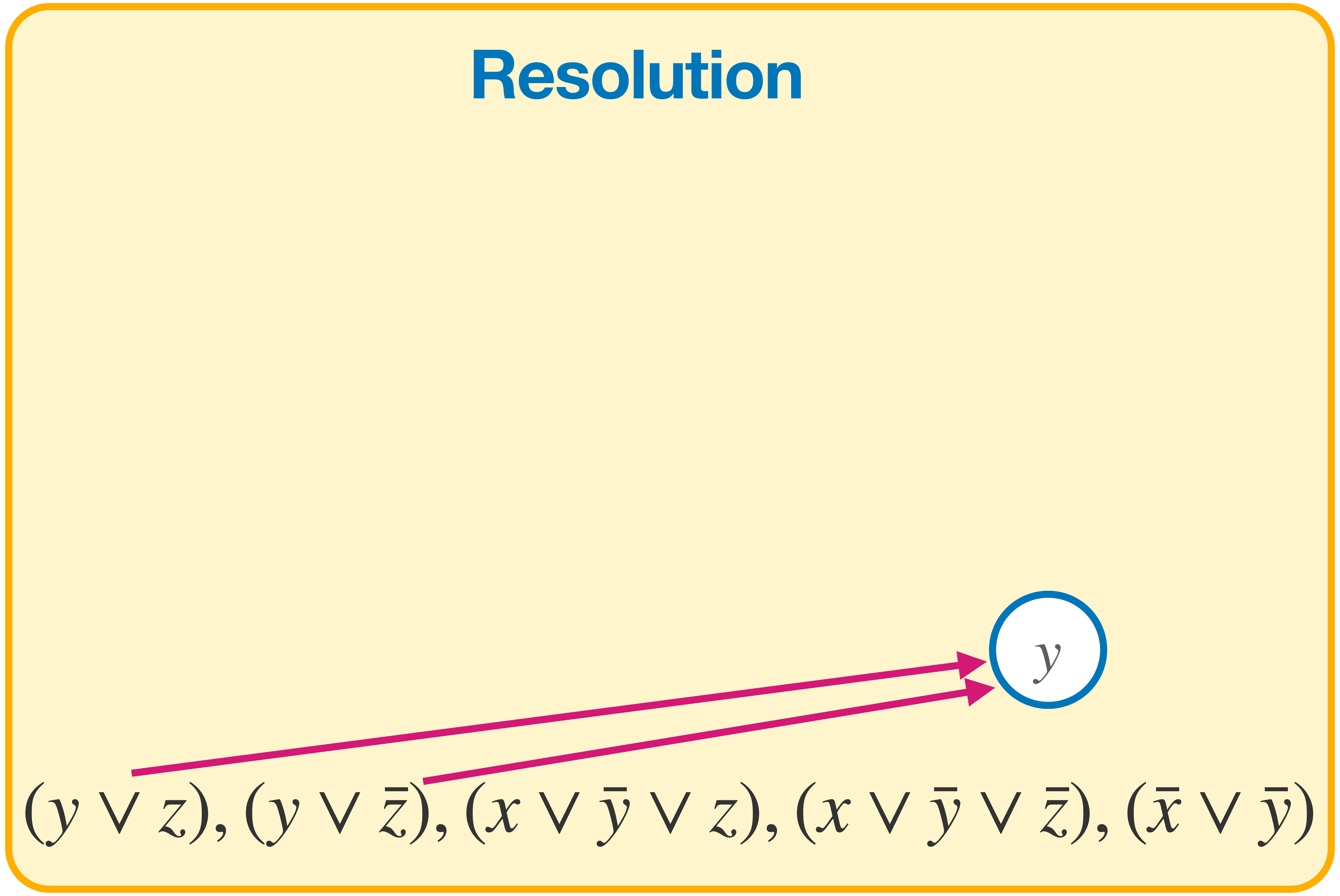
$$(y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})$$

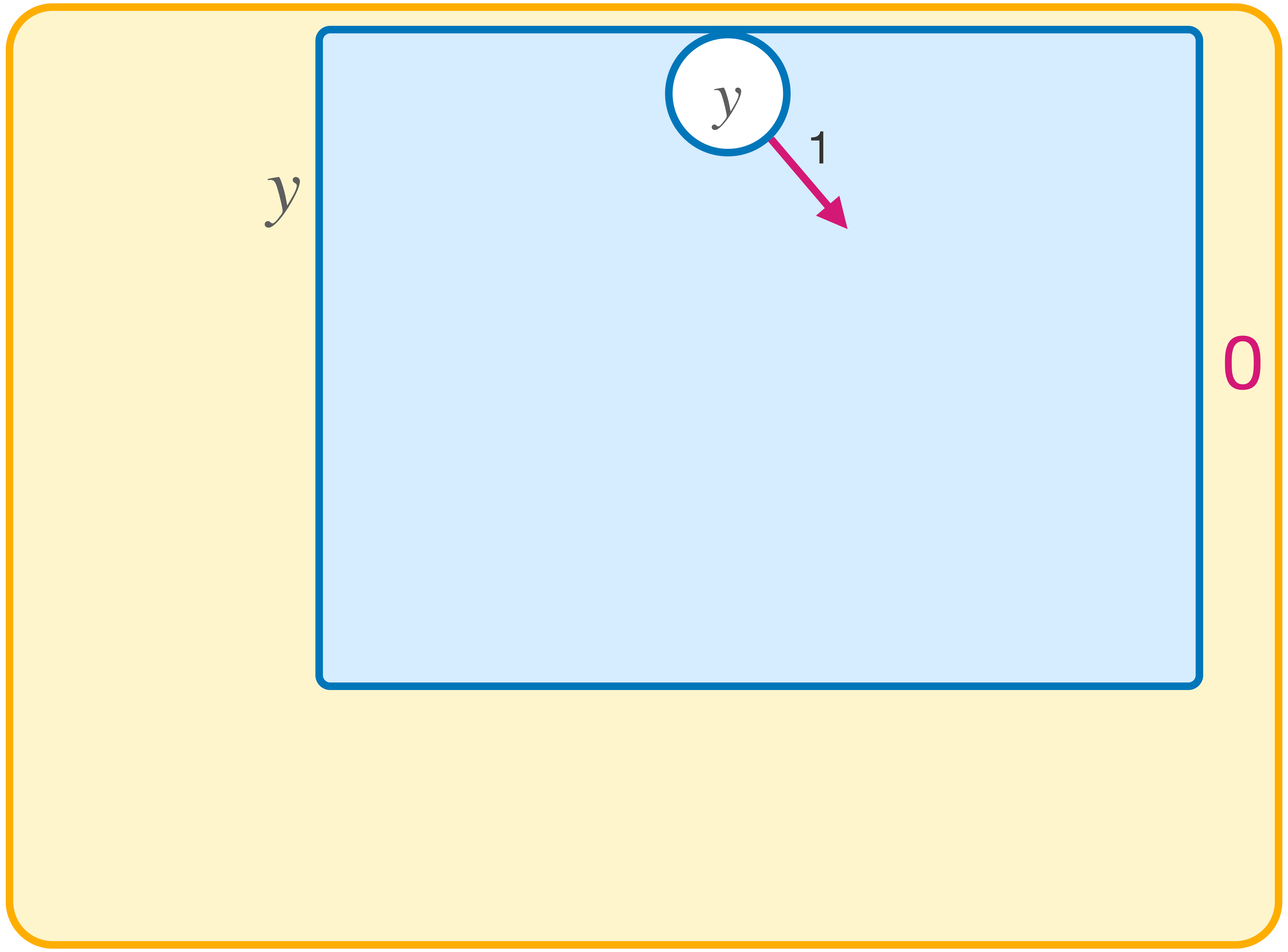




Analyzing CDCL

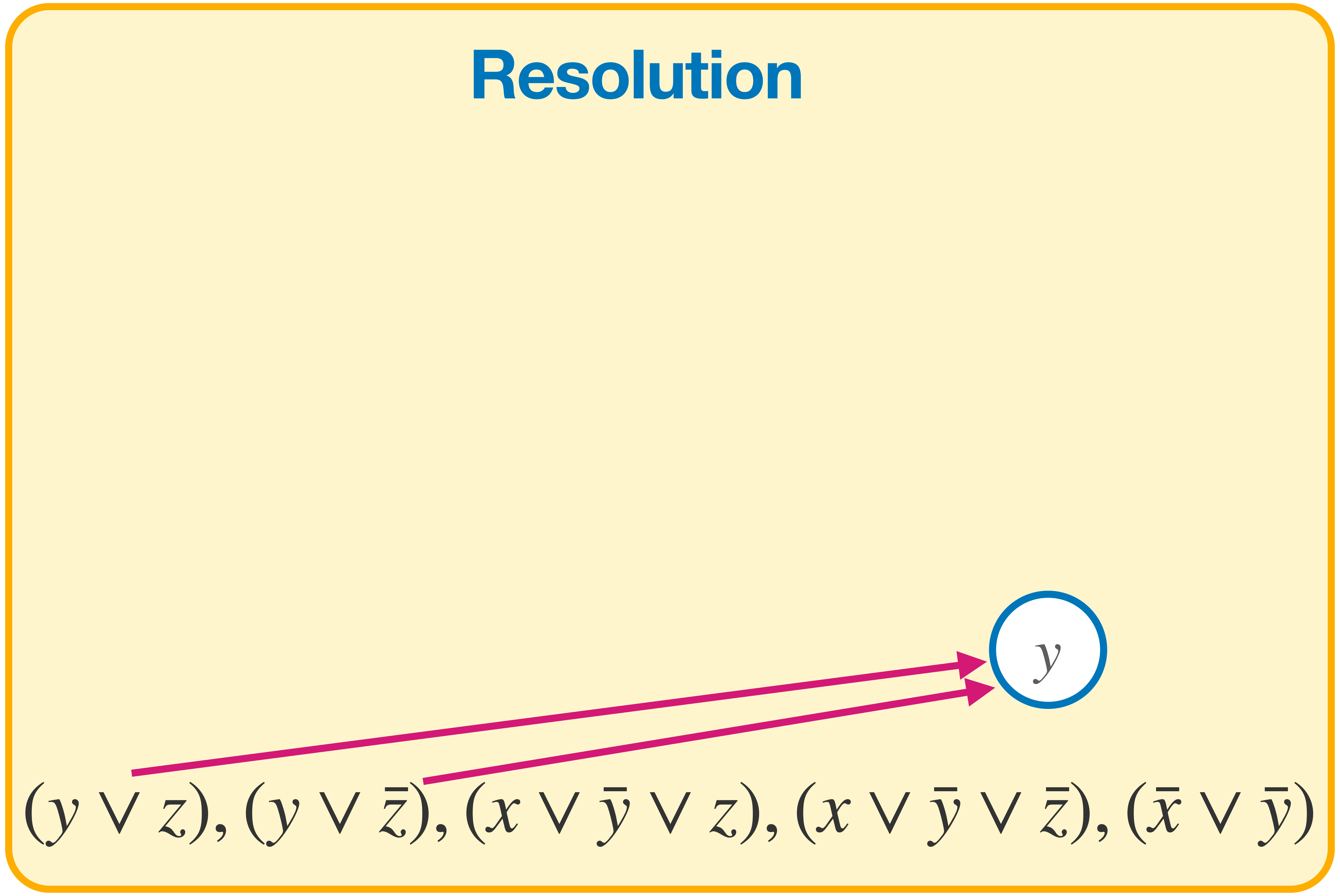
$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$

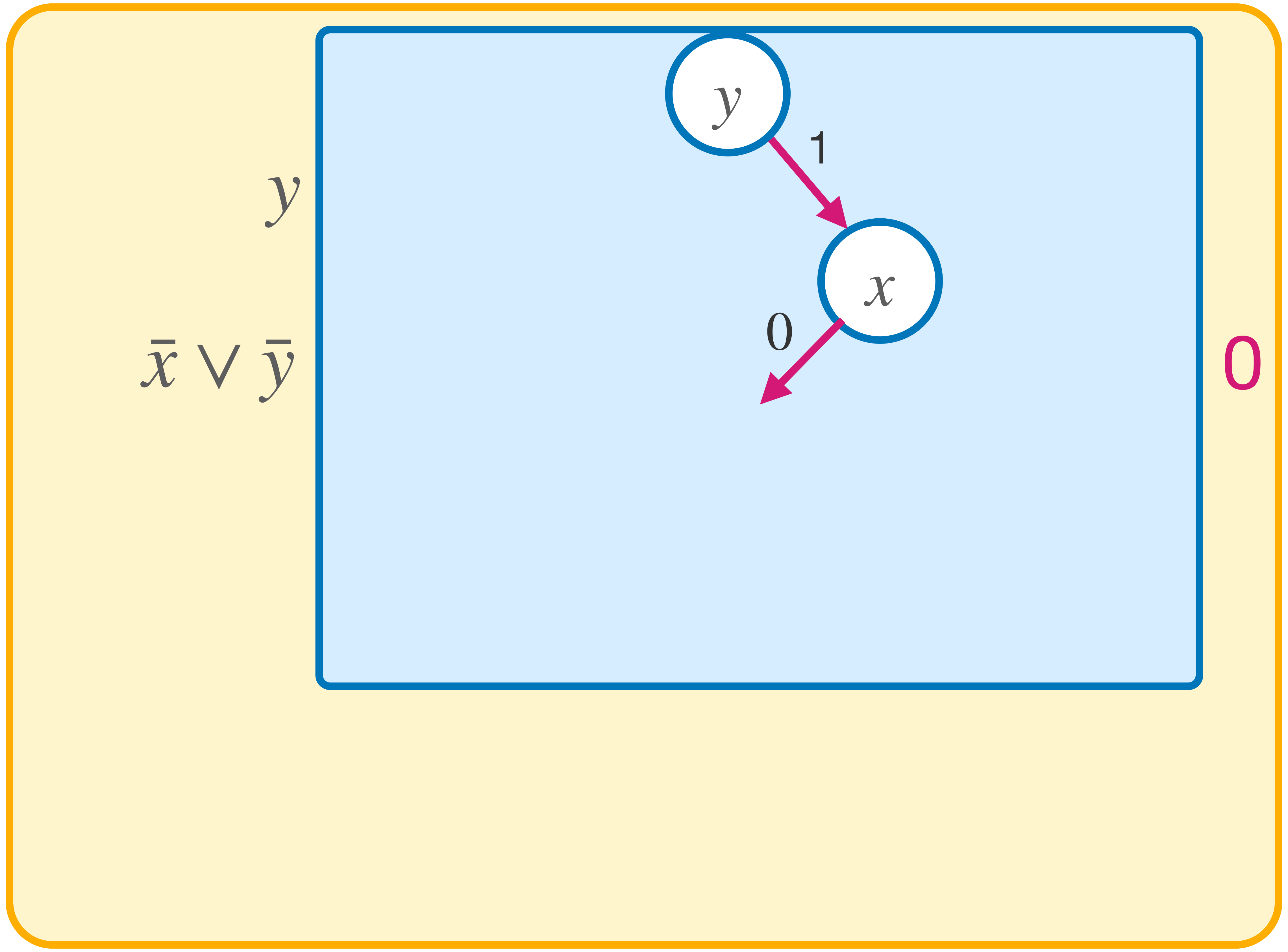




Analyzing CDCL

$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$

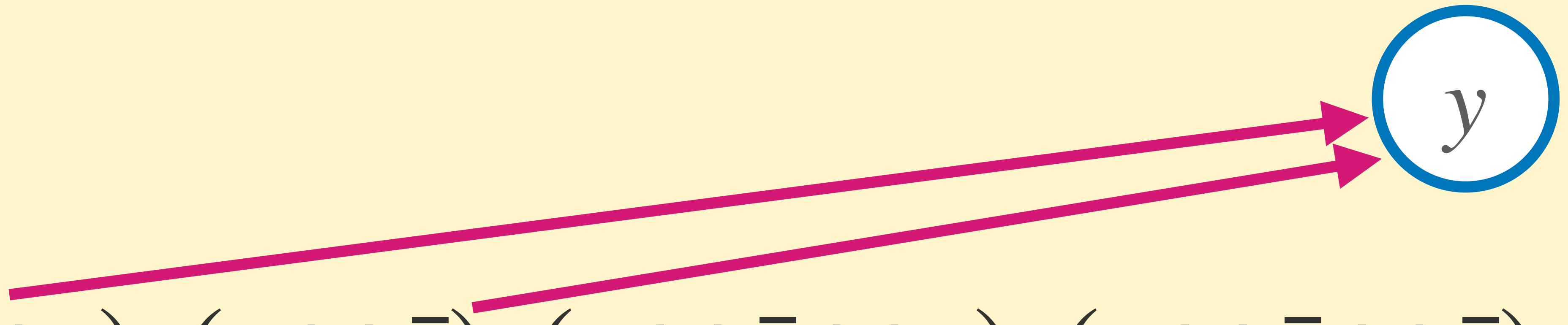




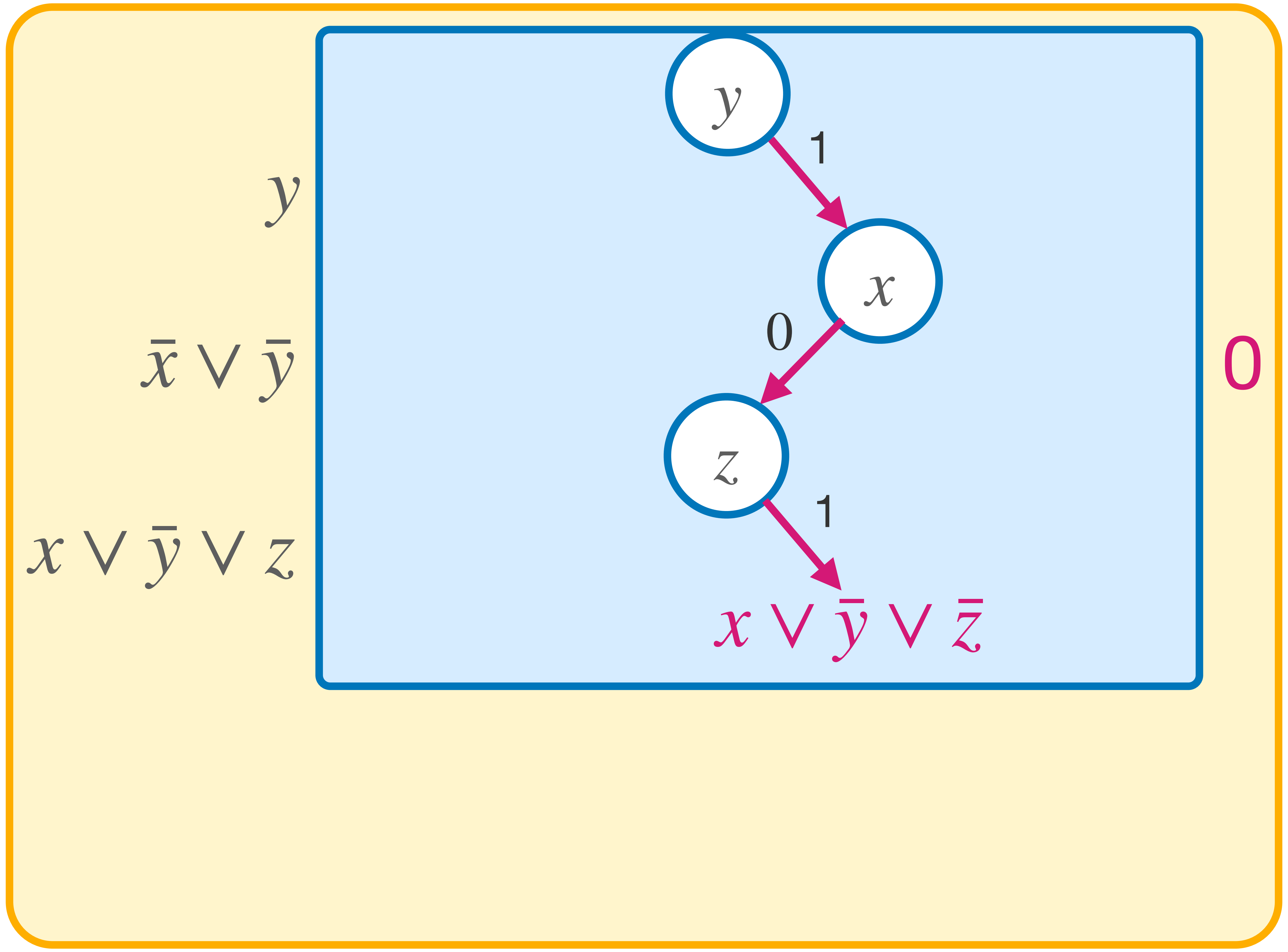
Analyzing CDCL

$$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$

Resolution

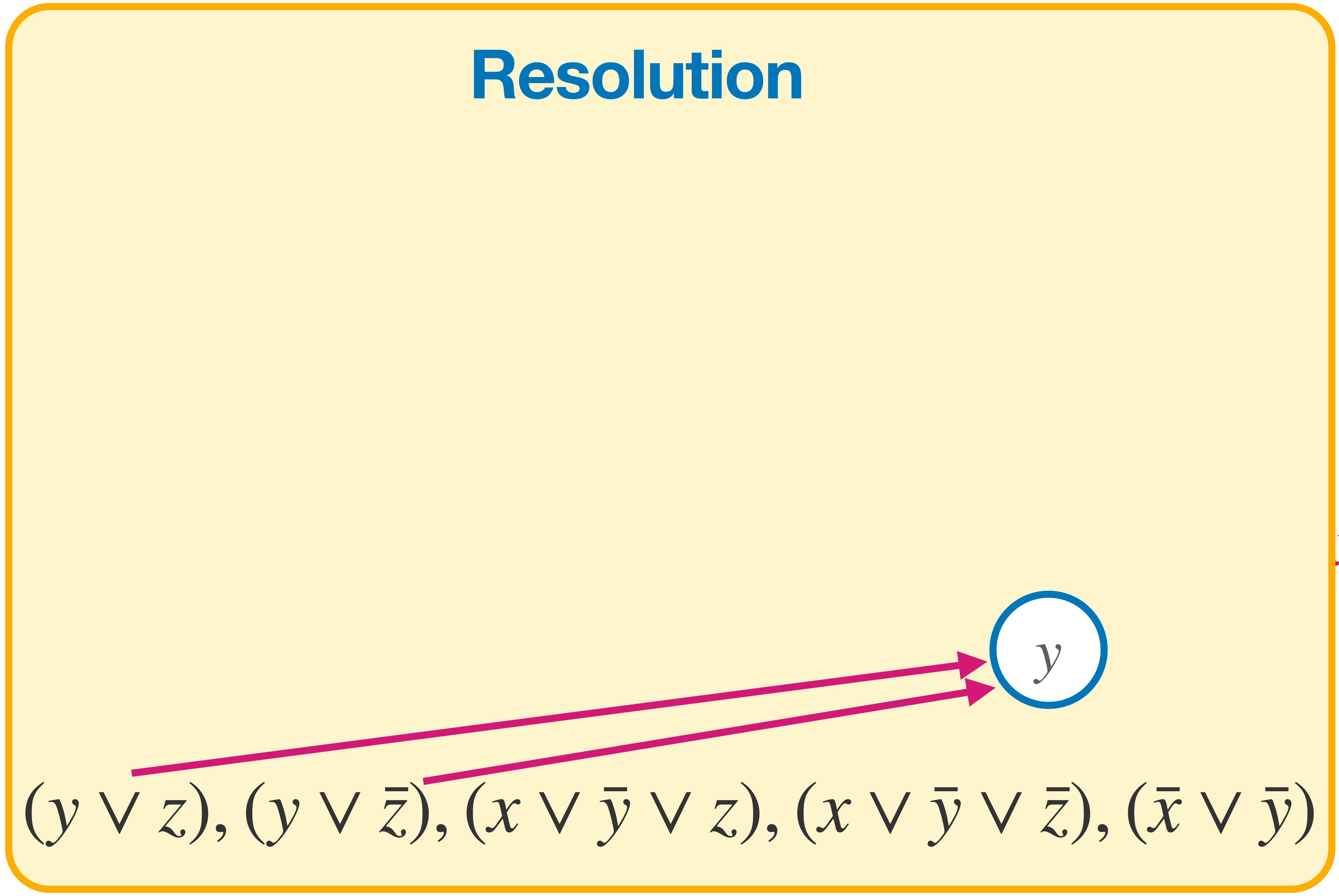


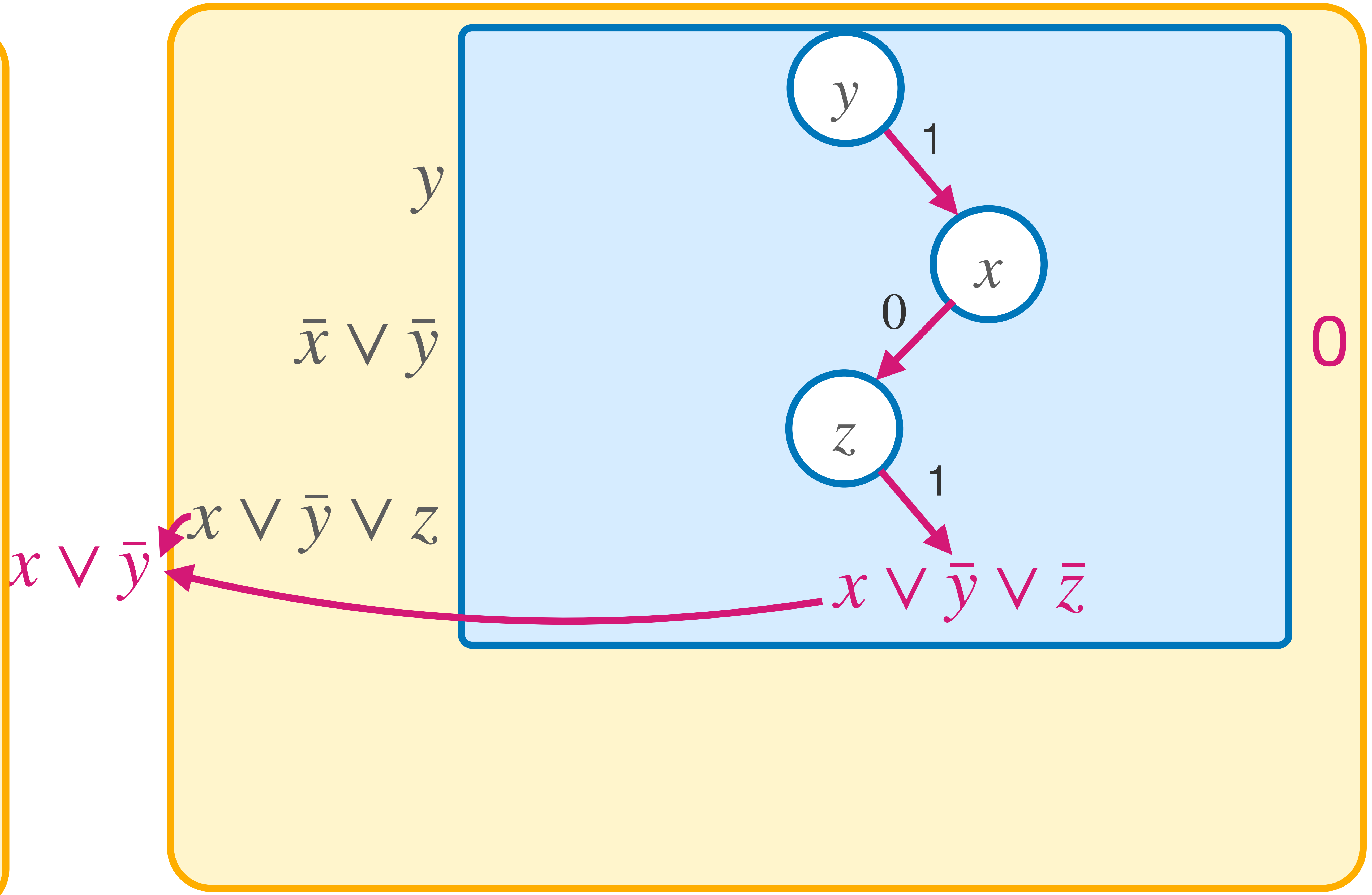
$(y \vee z), (y \vee \bar{z}), (x \vee \bar{y} \vee z), (x \vee \bar{y} \vee \bar{z}), (\bar{x} \vee \bar{y})$



Analyzing CDCL

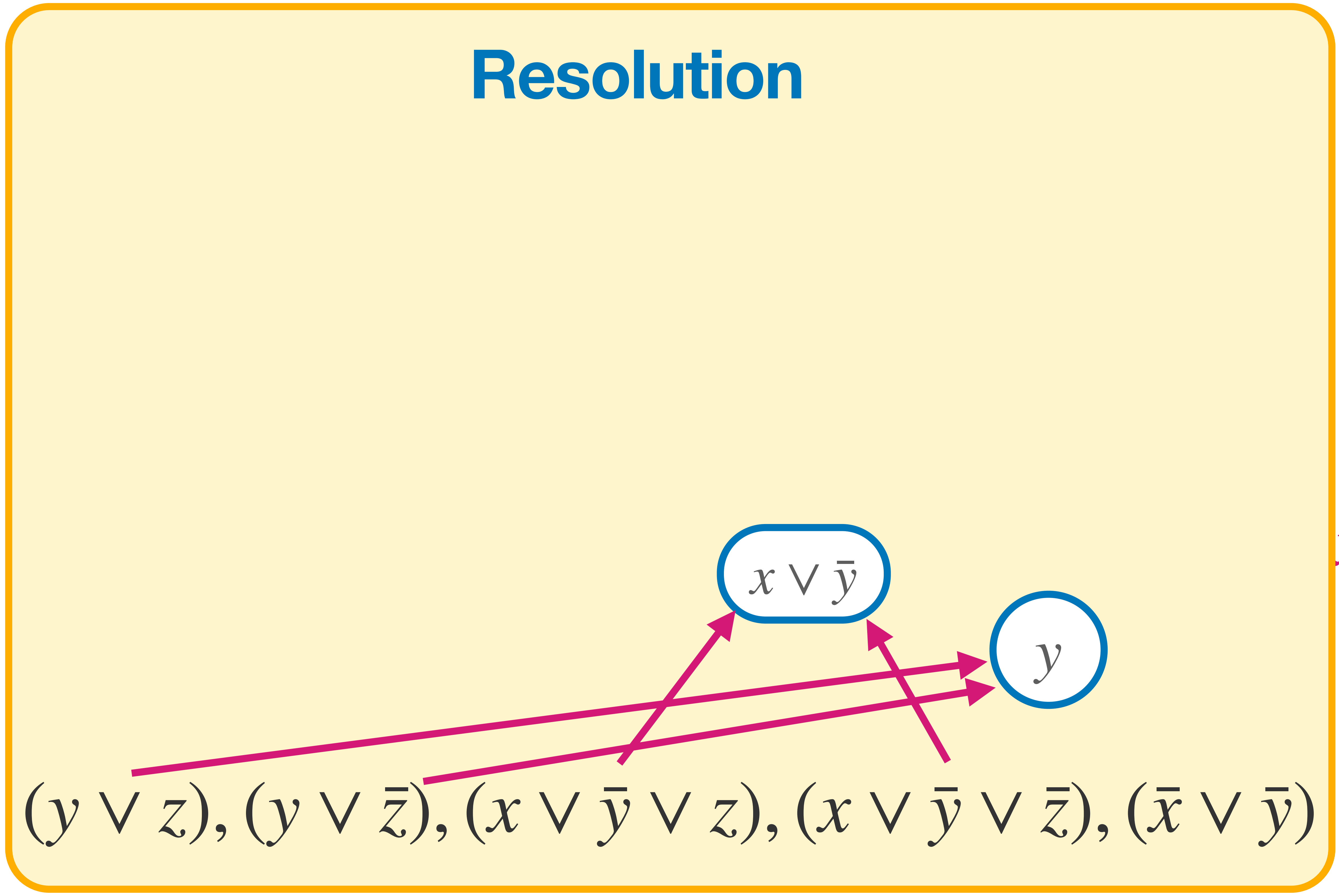
$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$

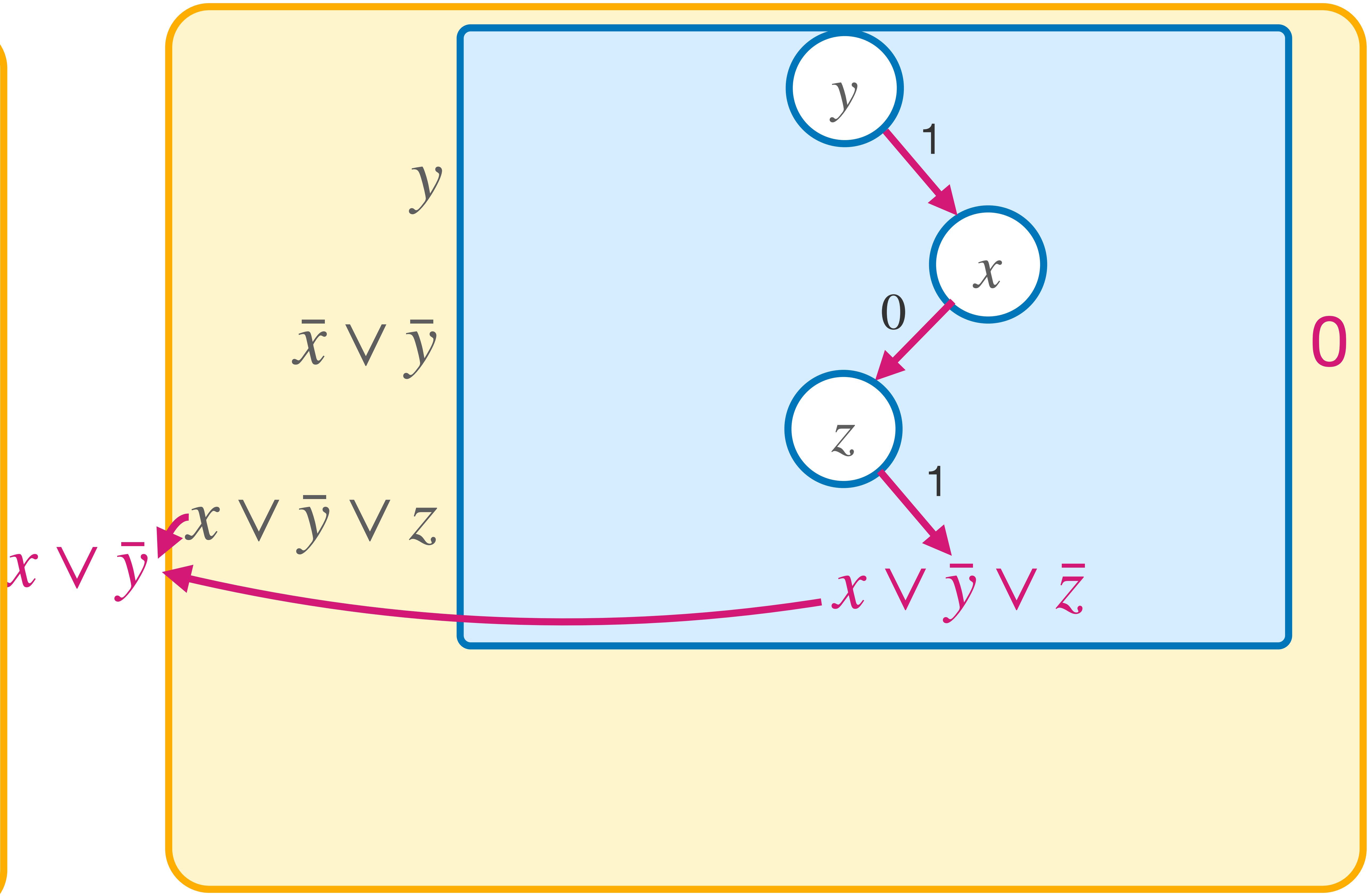




Analyzing CDCL

$(y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$





Analyzing CDCL

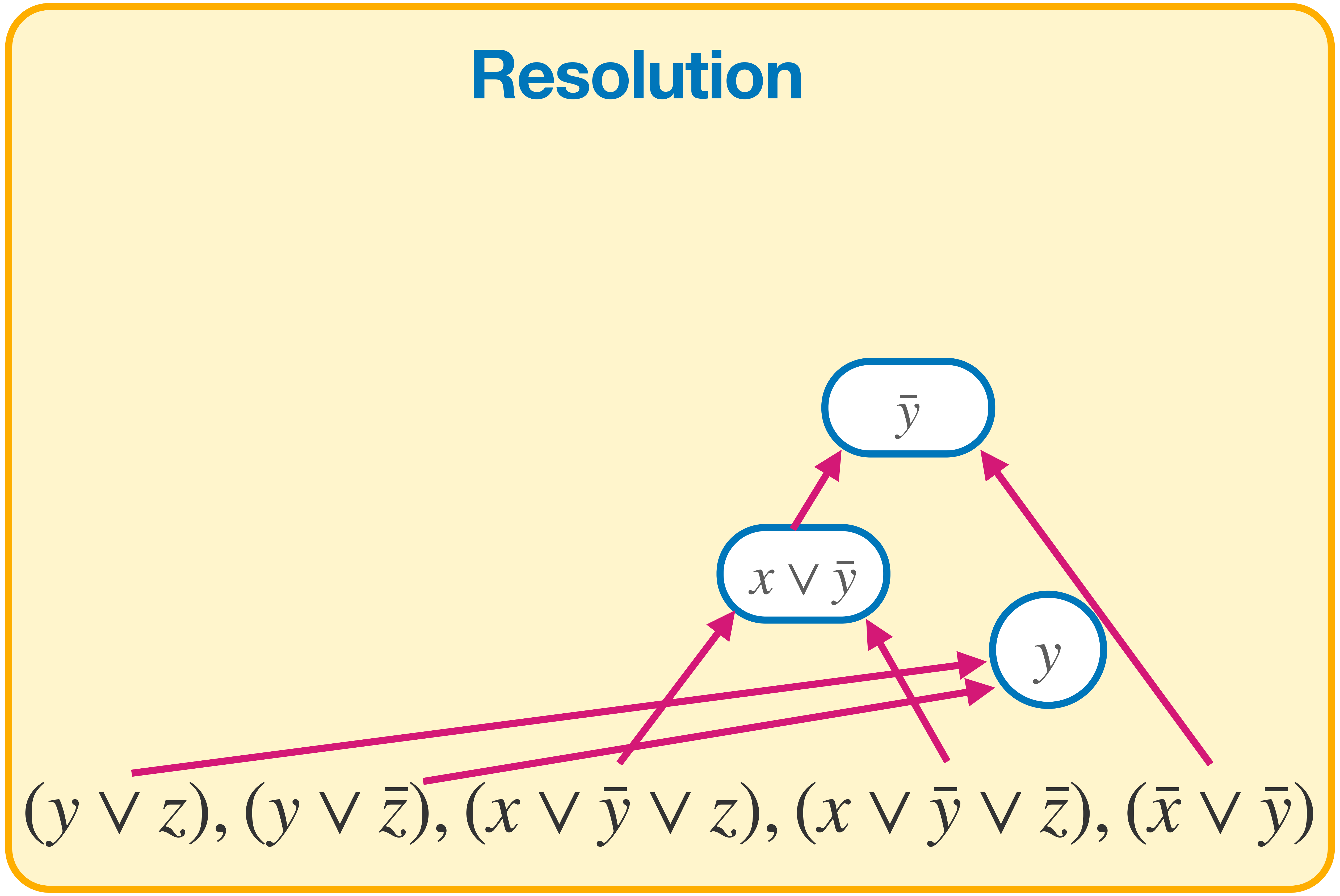
$(\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$

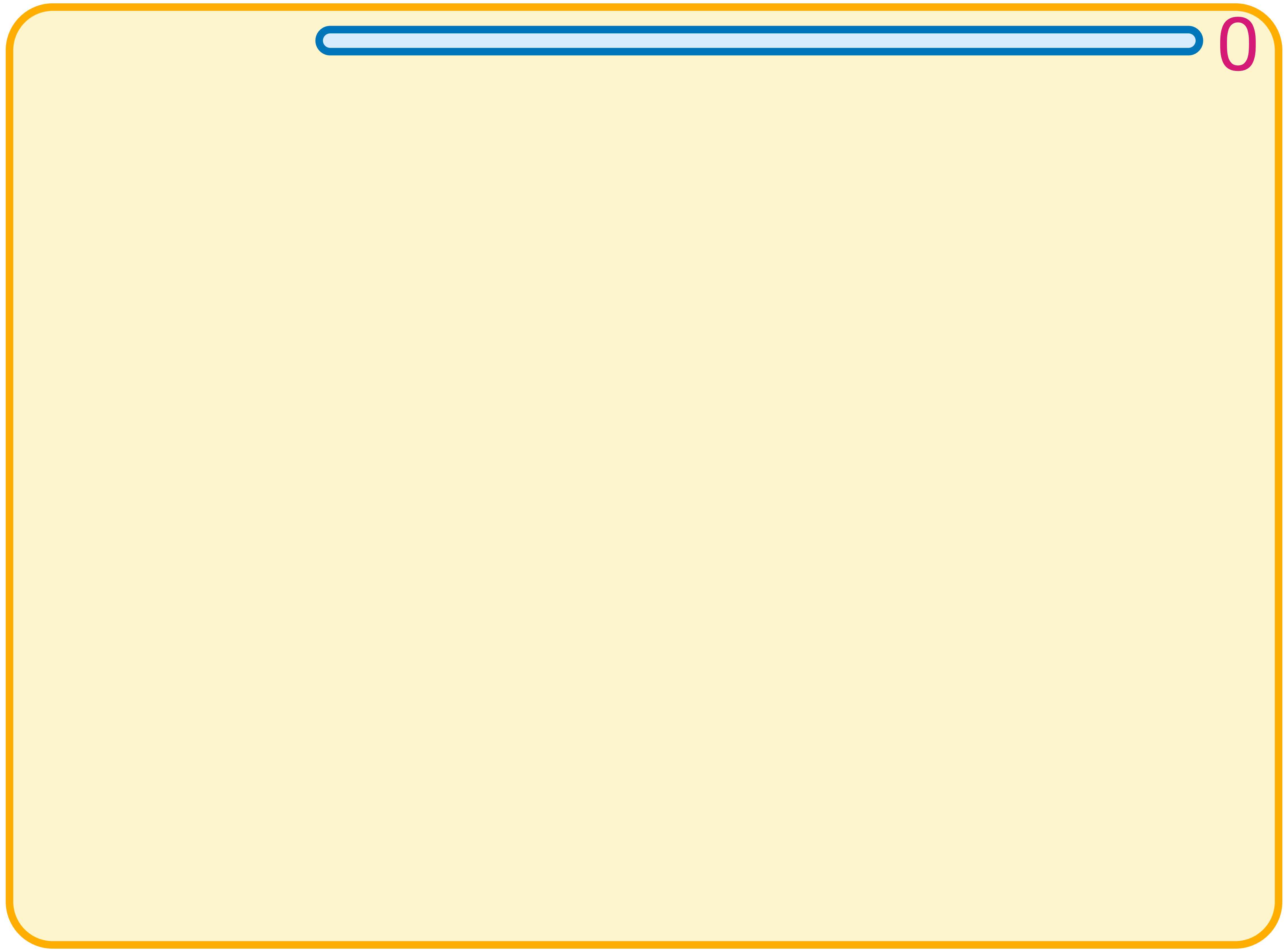
Analyzing CDCL

$$(\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$

Analyzing CDCL

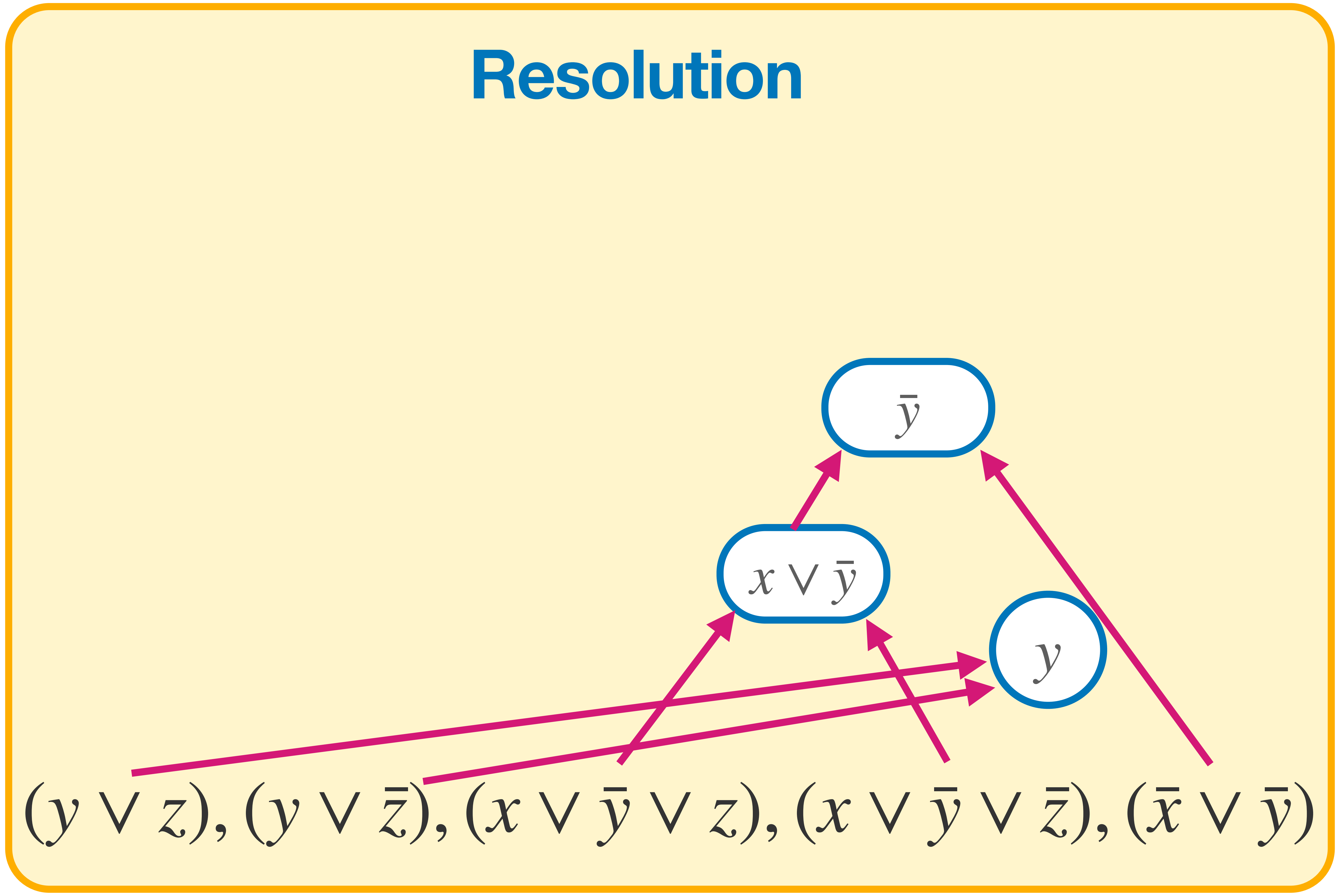
$$(\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$

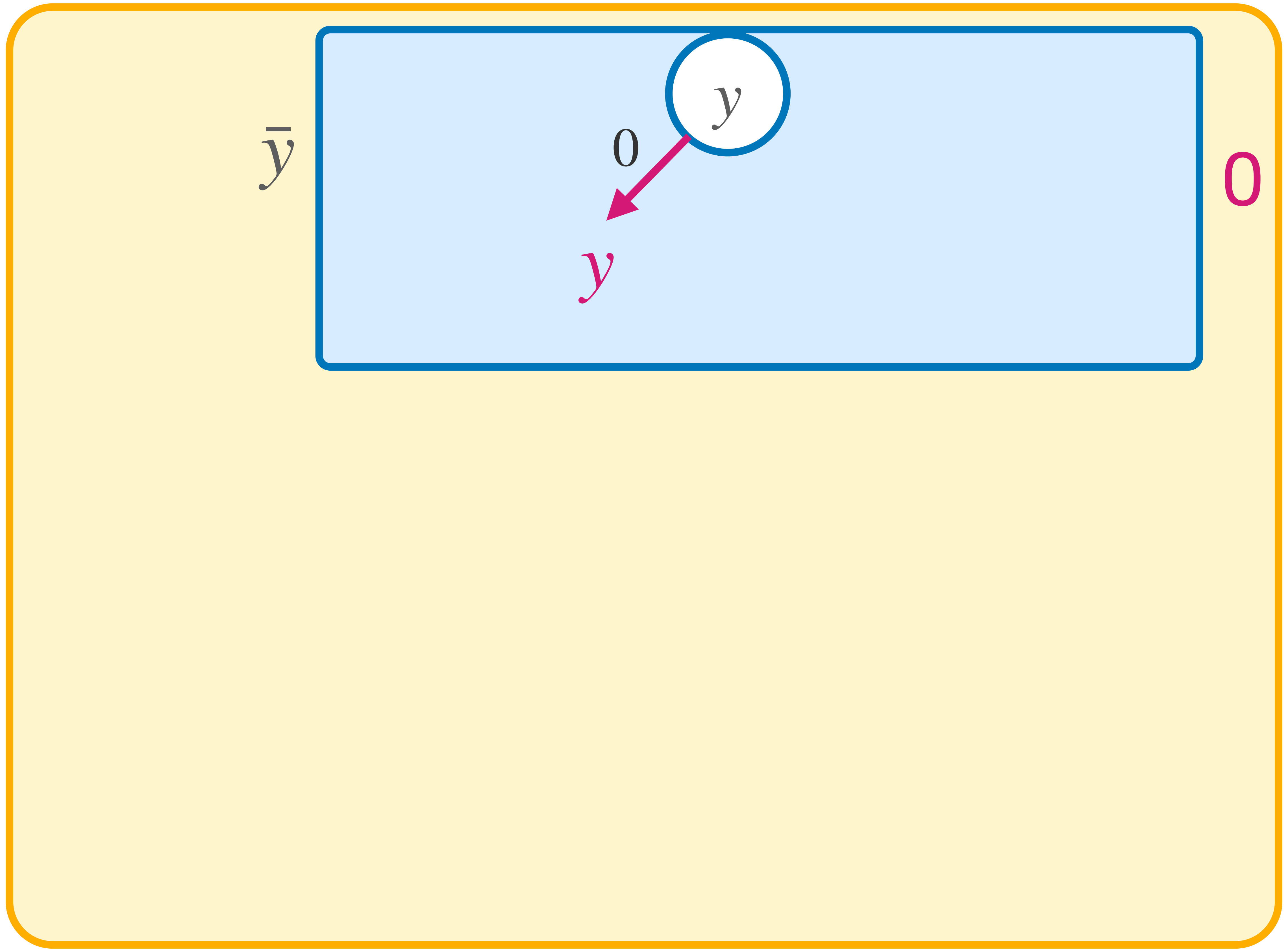




Analyzing CDCL

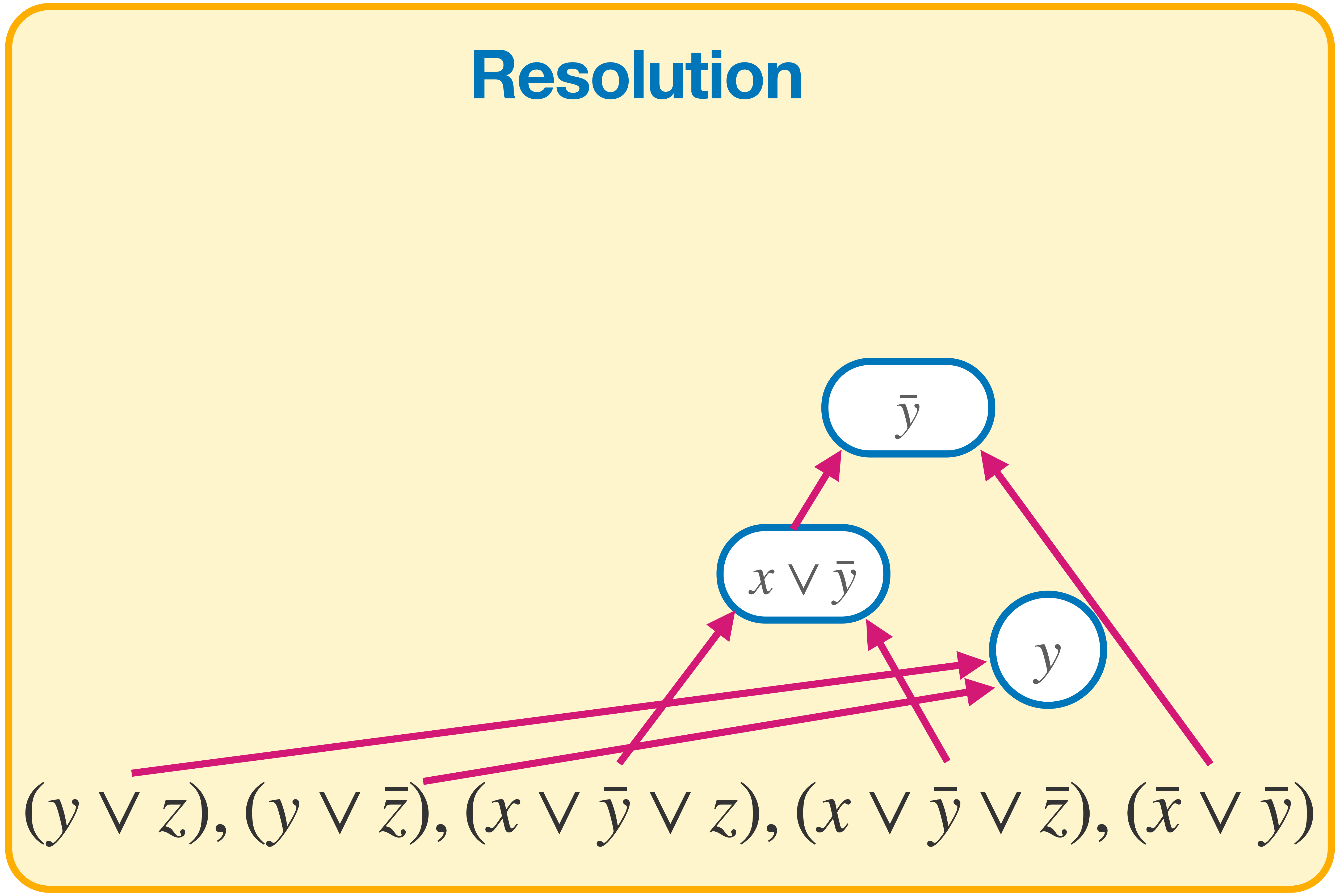
$(\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$

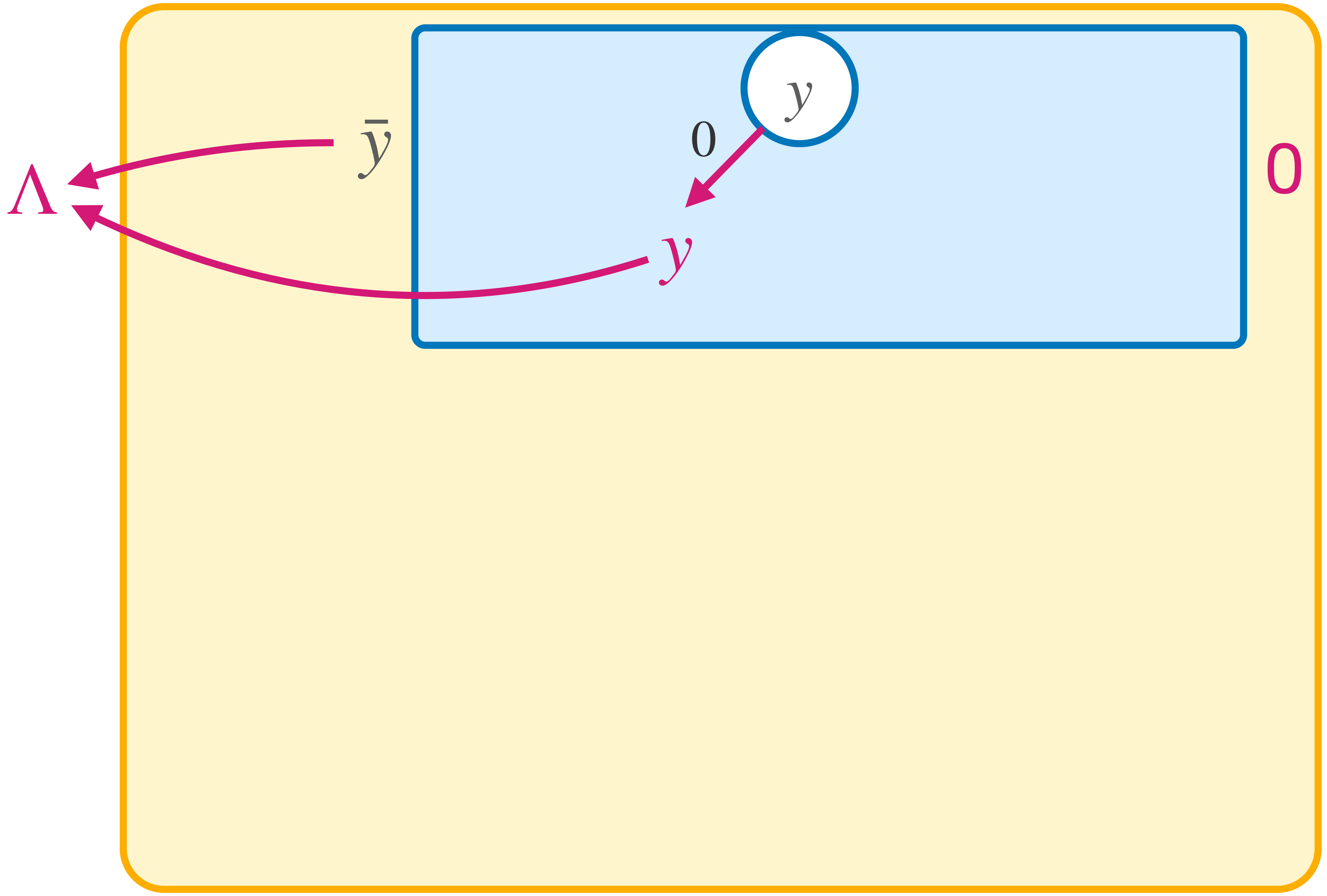




Analyzing CDCL

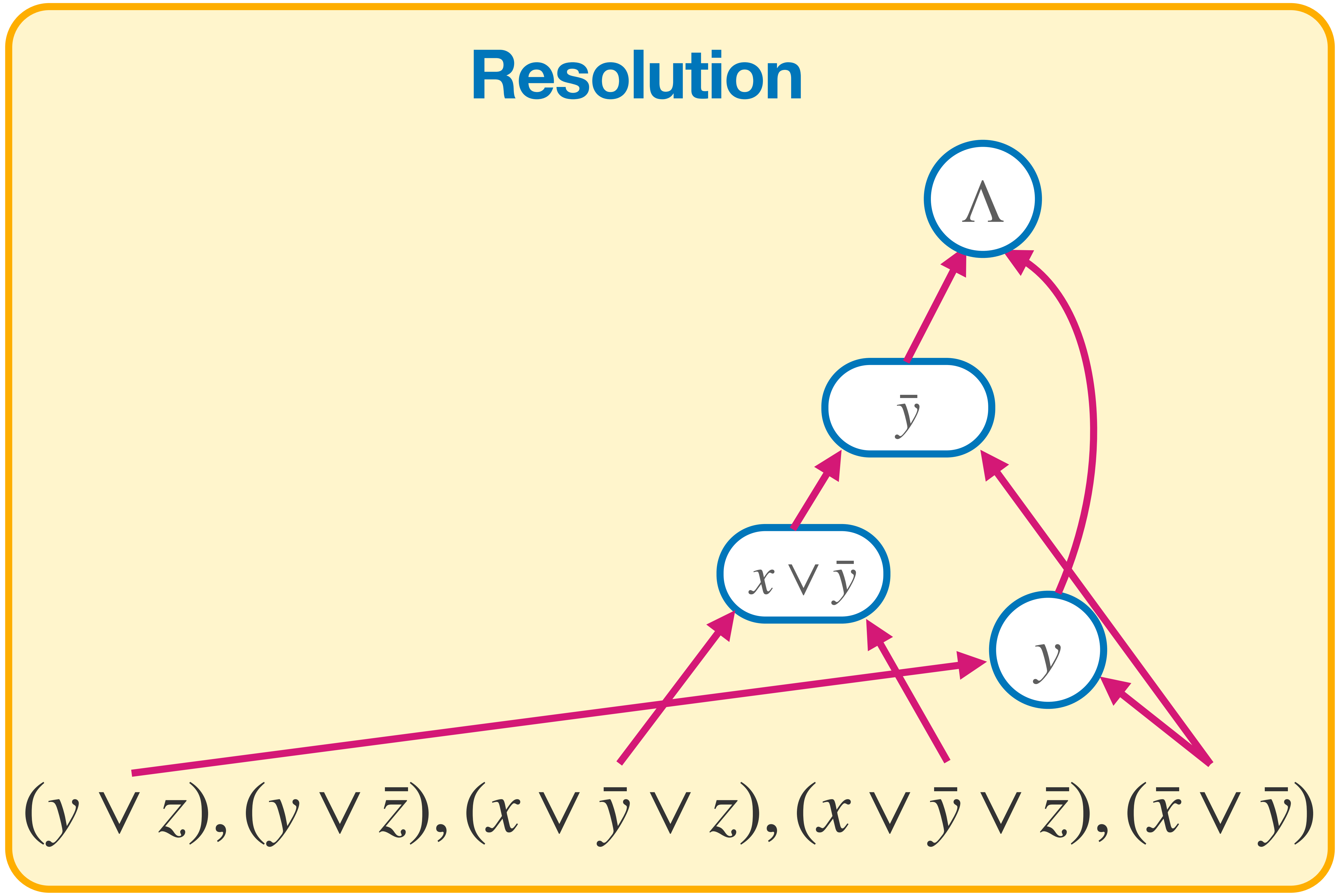
$$\Lambda \wedge (\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$

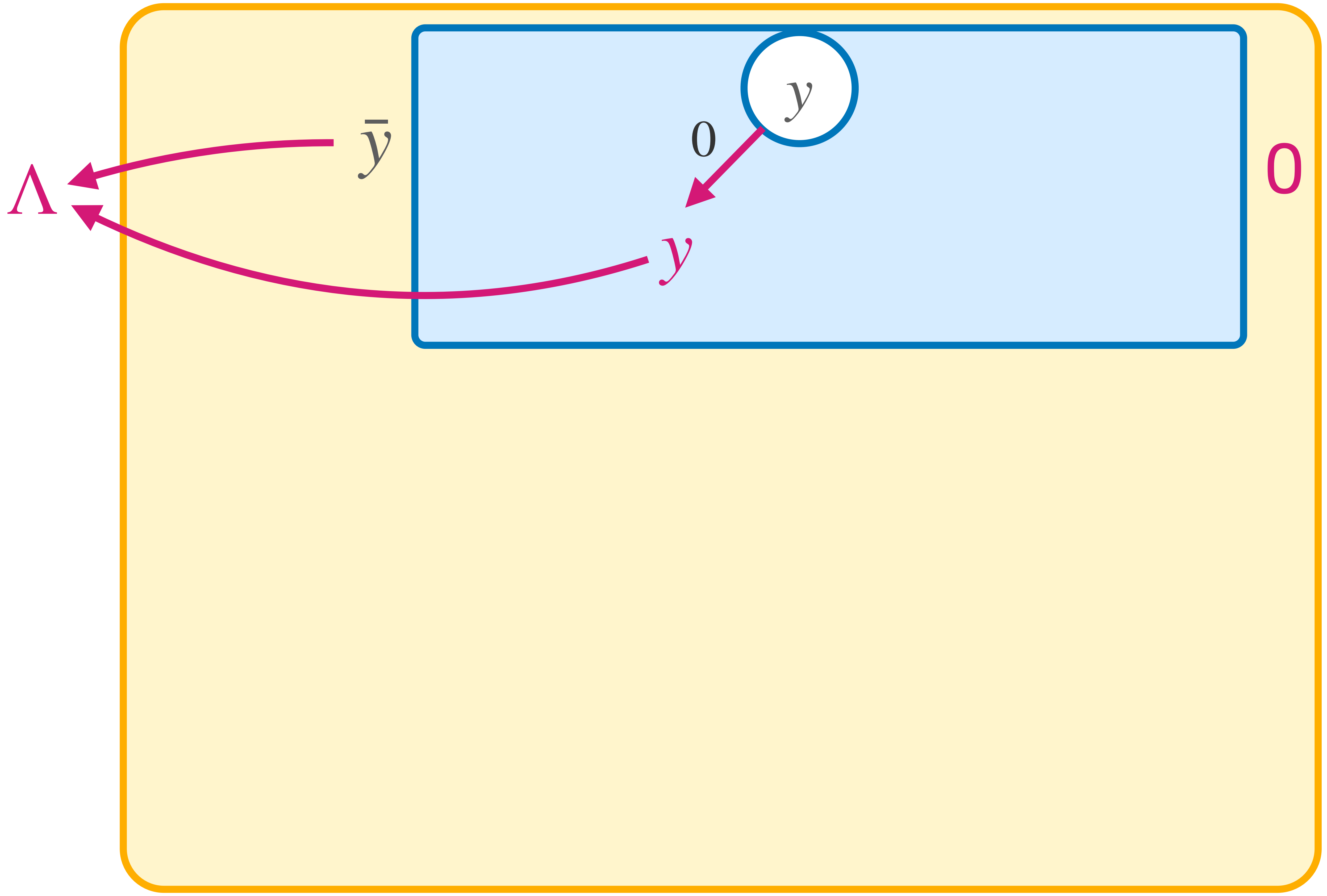




Analyzing CDCL

$$\Lambda \wedge (\bar{y}) \wedge (y) \wedge (y \vee z) \wedge (y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y})$$





Analyzing CDCL

Theorem: Let F be an unsatisfiable CNF formula. If CDCL takes time s to solve F, then there is a size-s Resolution proof of F

In order to prove bounds on the runtime of CDCL it suffices to analyze Resolution proof size

Resolution Lower Bounds — Some History

First lower bound proved by Armin Haken in `85

Technique: Bottleneck Counting

Resolution Lower Bounds — Some History

First lower bound proved by Armin Haken in `85

Technique: Bottleneck Counting

In every Resolution proof of F,

1. Every $x \in \{0,1\}^n$ falsifies a wide clause in the proof

Resolution Lower Bounds — Some History

First lower bound proved by Armin Haken in `85

Technique: Bottleneck Counting

In every Resolution proof of F,

1. Every $x \in \{0,1\}^n$ falsifies a wide clause in the proof
2. Every wide clause is falsified by only a small number of $x \in \{0,1\}^n$

Resolution Lower Bounds — Some History

First lower bound proved by Armin Haken in `85

Technique: Bottleneck Counting

In every Resolution proof of F,

1. Every $x \in \{0,1\}^n$ falsifies a wide clause in the proof
2. Every wide clause is falsified by only a small number of $x \in \{0,1\}^n$

$\Longrightarrow$ Proof must have many wide clauses! (Size lower bound!)

Resolution Lower Bounds — Some History

Connection between width and size formalized by Ben-Sasson Wigderson `99

Resolution Lower Bounds — Some History

Connection between width and size formalized by Ben-Sasson Wigderson `99

Theorem:

For any unsatisfiable CNF formula F on n variables with clauses of width $\leq w$,

$$size_R(F) \geq \exp \Omega((width_R(F) - w)^2/n)$$

Resolution Lower Bounds — Some History

Connection between width and size formalized by Ben-Sasson Wigderson `99

Theorem:

For any unsatisfiable CNF formula F on n variables with clauses of width $\leq w$,

$$size_R(F) \geq \exp \Omega((width_R(F) - w)^2/n)$$

Takeaway: If $w = O(1)$ and $width_R(F) = \omega(\sqrt{n})$ then we get size lower bounds!

Resolution Lower Bounds — Some History

Connection between width and size formalized by Ben-Sasson Wigderson `99

Theorem:

For any unsatisfiable CNF formula F on n variables with clauses of width $\leq w$,

$$size_R(F) \geq \exp \Omega((width_R(F) - w)^2/n)$$

Takeaway: If $w = O(1)$ and $width_R(F) = \omega(\sqrt{n})$ then we get size lower bounds!

We can prove a similar theorem, with a **much** simpler proof by composition!

Resolution Lower Bounds — Some History

Connection between width and size formalized by Ben-Sasson Wigderson `99

Theorem:

For any unsatisfiable CNF formula F on n variables with clauses of width $\leq w$,

$$size_R(F) \geq \exp \Omega((width_R(F) - w)^2/n)$$

Takeaway: If $w = O(1)$ and $width_R(F) = \omega(\sqrt{n})$ then we get size lower bounds!

We can prove a similar theorem, with a **much** simpler proof by composition!

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

Size to Width

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

$F(x_1, \ldots, x_n) \circ XOR$ obtained by substituting $x_i \leftarrow y_i \oplus z_i$

Size to Width

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

$F(x_1, \ldots, x_n) \circ XOR$ obtained by substituting $x_i \leftarrow y_i \oplus z_i = (\bar{y}_i \vee \bar{z}_i) \wedge (y_i \vee z_i)$

Size to Width

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

$F(x_1, \ldots, x_n) \circ XOR$ obtained by substituting $x_i \leftarrow y_i \oplus z_i = (\bar{y}_i \vee \bar{z}_i) \wedge (y_i \vee z_i)$

e.g. $(x_1 \vee \bar{x}_2) \circ XOR$

Size to Width

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

$F(x_1, \ldots, x_n) \circ XOR$ obtained by substituting $x_i \leftarrow y_i \oplus z_i = (\bar{y}_i \vee \bar{z}_i) \wedge (y_i \vee z_i)$

e.g. $(x_1 \vee \bar{x}_2) \circ XOR = (y_1 \oplus z_1) \vee \neg(y_2 \oplus z_2)$

Size to Width

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

$F(x_1, \ldots, x_n) \circ XOR$ obtained by substituting $x_i \leftarrow y_i \oplus z_i = (\bar{y}_i \vee \bar{z}_i) \wedge (y_i \vee z_i)$

e.g. $(x_1 \vee \bar{x}_2) \circ XOR = (y_1 \oplus z_1) \vee \neg(y_2 \oplus z_2)$

$$= ((\bar{y}_1 \vee \bar{z}_1) \wedge (y_1 \vee z_1)) \vee ((y_2 \wedge z_2) \vee (\bar{y}_2 \wedge z_2))$$

Size to Width

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

$F(x_1, \ldots, x_n) \circ XOR$ obtained by substituting $x_i \leftarrow y_i \oplus z_i = (\bar{y}_i \vee \bar{z}_i) \wedge (y_i \vee z_i)$

e.g. $(x_1 \vee \bar{x}_2) \circ XOR = (y_1 \oplus z_1) \vee \neg(y_2 \oplus z_2)$

$= ((\bar{y}_1 \vee \bar{z}_1) \wedge (y_1 \vee z_1)) \vee ((y_2 \wedge z_2) \vee (\bar{y}_2 \wedge z_2))$

$=\ldots$ expand as CNF

Reminder of Today

Prove a lower bound on Resolution and therefore CDCL runtime!

Reminder of Today

Prove a lower bound on Resolution and therefore CDCL runtime!

Step 1. Prove the theorem:

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

Reminder of Today

Prove a lower bound on Resolution and therefore CDCL runtime!

Step 1. Prove the theorem:

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

Step 2. Prove that some formula F (Pigeonhole formula) requires large width

Size to Width

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

Proof: $w := Width_R(F)$, let Π be any Resolution proof of $F \circ XOR$

Size to Width

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

Proof: $w := Width_R(F)$, let Π be any Resolution proof of $F \circ XOR$

Size to Width

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

Proof: $w := Width_R(F)$, let Π be any Resolution proof of $F \circ XOR$

Idea: construct a partial assignment $\rho \in \{0,1,*\}^n$ so that

Size to Width

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

Proof: $w := Width_R(F)$, let Π be any Resolution proof of $F \circ XOR$

Idea: construct a partial assignment $\rho \in \{0,1,*\}^n$ so that

1. $\Pi \restriction \rho$ is a proof of F

Size to Width

Theorem:

For any unsatisfiable CNF formula F,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

Proof: $w := Width_R(F)$, let Π be any Resolution proof of $F \circ XOR$

Idea: construct a partial assignment $\rho \in \{0,1,*\}^n$ so that

1. $\Pi \restriction \rho$ is a proof of F
2. If Π is small $\Longrightarrow \Pi \restriction \rho$ has width $< w$

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

Claim: $\Pi \restriction \rho$ is a proof of $F \restriction \rho = F$

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

Claim: $\Pi \upharpoonright \rho$ is a proof of $F \upharpoonright \rho = F$

For all $i \in [n]$, ρ fixes exactly one of y_i or z_i in $(y_i \oplus z_i)$; suppose it's z_i

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

$\rightarrow$ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

$\rightarrow$ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

Claim: $\Pi \restriction \rho$ is a proof of $F \restriction \rho = F$

For all $i \in [n]$, ρ fixes exactly one of y_i or z_i in $(y_i \oplus z_i)$; suppose it's z_i

- If $z_i = 1$ then $(y_i \oplus z_i) = \bar{z}_i$
- If $z_i = 0$ then $(y_i \oplus z_i) = z_i$

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

$\rightarrow$ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

$\rightarrow$ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

Claim: $\Pi \upharpoonright \rho$ is a proof of $F \upharpoonright \rho = F$

For all $i \in [n]$, ρ fixes exactly one of y_i or z_i in $(y_i \oplus z_i)$; suppose it's z_i

- If $z_i = 1$ then $(y_i \oplus z_i) = \bar{z}_i$
- If $z_i = 0$ then $(y_i \oplus z_i) = z_i$

Thus, $F \upharpoonright \rho = F$ (up to a renaming of the variables, and a flipping of their sign)

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

$\rightarrow$ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

$\rightarrow$ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

Claim: $\Pi \restriction \rho$ is a proof of $F \restriction \rho = F$

For all $i \in [n]$, ρ fixes exactly one of y_i or z_i in $(y_i \oplus z_i)$; suppose it's z_i

- If $z_i = 1$ then $(y_i \oplus z_i) = \bar{z}_i$
- If $z_i = 0$ then $(y_i \oplus z_i) = z_i$

Thus, $F \restriction \rho = F$ (up to a renaming of the variables, and a flipping of their sign)

$\implies \Pi \restriction \rho$ is a Resolution proof of F

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

Claim: If $|\Pi|$ is small, then there is ρ such that $\Pi \restriction \rho$ has width $< w$

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

Claim: If $|\Pi|$ is small, then there is ρ such that $\Pi \upharpoonright \rho$ has width $< w$

Want to show: every wide clause in Π is satisfied by ρ with probability > 0

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

Claim: If $|\Pi|$ is small, then there is ρ such that $\Pi \restriction \rho$ has width $< w$

Want to show: every wide clause in Π is satisfied by ρ with probability > 0

Let C have width w

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

Claim: If $|\Pi|$ is small, then there is ρ such that $\Pi \restriction \rho$ has width $< w$

Want to show: every wide clause in Π is satisfied by ρ with probability > 0

Let C have width w

→ Each literal in C is set to 1 w.p. $1/4 \implies Pr[C(\rho) \neq 1] \leq (3/4)^w$

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

$\rightarrow$ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

$\rightarrow$ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

Claim: If $|\Pi|$ is small, then there is ρ such that $\Pi \restriction \rho$ has width $< w$

Want to show: every wide clause in Π is satisfied by ρ with probability > 0

Let C have width w

$\rightarrow$ Each literal in C is set to 1 w.p. $1/4 \implies Pr[C(\rho) \neq 1] \leq (3/4)^w$

By a union bound over the wide clauses in Π

$$\Pr[\Pi \restriction \rho \text{ has width} \geq w] \leq (3/4)^w |\Pi|$$

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

Claim: If $|\Pi|$ is small, then there is ρ such that $\Pi \restriction \rho$ has width $< w$

Want to show: every wide clause in Π is satisfied by ρ with probability > 0

Let C have width w

→ Each literal in C is set to 1 w.p. $1/4 \implies Pr[C(\rho) \neq 1] \leq (3/4)^w$

By a union bound over the wide clauses in Π

$$\Pr[\Pi \restriction \rho \text{ has width } \geq w] \leq (3/4)^w |\Pi|$$

If $|\Pi| \leq (4/3)^w \implies$ exists ρ such that $\Pi \restriction \rho$ has width $< w$

Size to Width

Generate ρ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

Claim: If $|\Pi|$ is small, then there is ρ such that $\Pi \restriction \rho$ has width $< w$

Want to show: every wide clause in Π is satisfied by ρ with probability > 0

Let C have width w

→ Each literal in C is set to 1 w.p. $1/4 \implies Pr[C(\rho) \neq 1] \leq (3/4)^w$

By a union bound over the wide clauses in Π

$$\Pr[\Pi \restriction \rho \text{ has width} \geq w] \leq (3/4)^w |\Pi|$$

If $|\Pi| \leq (4/3)^w \implies$ exists ρ such that $\Pi \restriction \rho$ has width $< w$ Contradiction!