Limited Backtracking

2-SAT

Input: A formula ϕ consisting of

- A set of n boolean variables $x_1, \ldots, x_n$
- A conjuction of m clauses $C_1, \ldots, C_m$, i.e., disjunctions of 2 literals $C_j = (c_j^{(1)} \vee c_j^{(2)})$, where a literal is either a variable or its negation.

A truth assignment is a function $\tau : \{x_1, \ldots, x_n\} \rightarrow \{\top, \bot\}$

- A clause $C_j = (c_j^{(1)} \vee c_j^{(2)})$ is *satisfied* by τ according to the rules of boolean algebra.
- ϕ is satisfied iff all m clauses $C_1, \ldots, C_m$ are satisfied.

Question: Is there a truth assigment that satisfies ϕ?

Example

Formula

$$\phi = (x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_4})$$

Example

Formula

$$\phi = (x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_4})$$

Satisfying assignment:

$$x_1 = \bot \quad x_2 = \bot \quad x_3 = \top \quad x_4 = \top$$

A Naive Solution

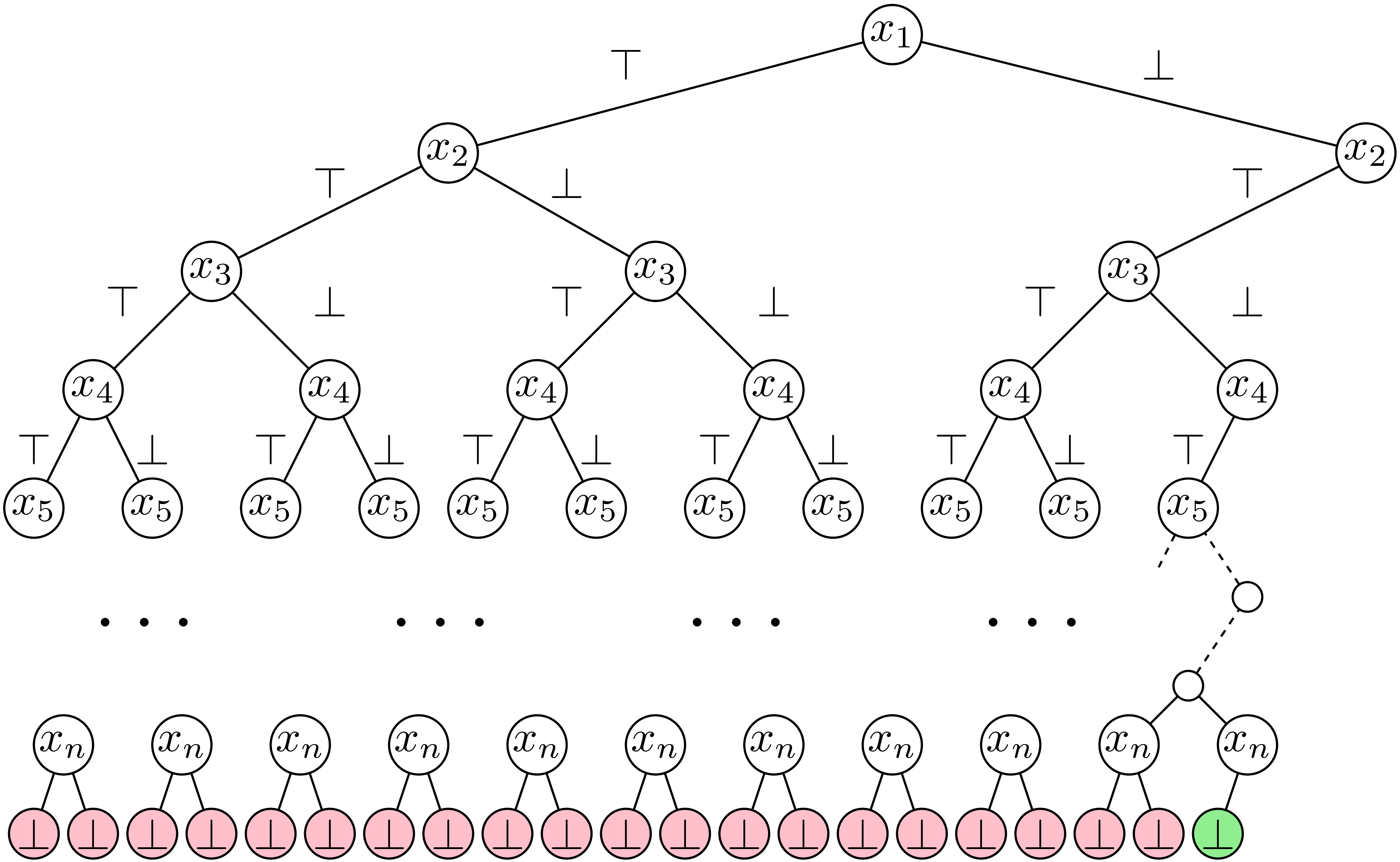
Idea: Recursively try all possible variable assignments

Initially all variables are *unassigned*.

Bruteforce():

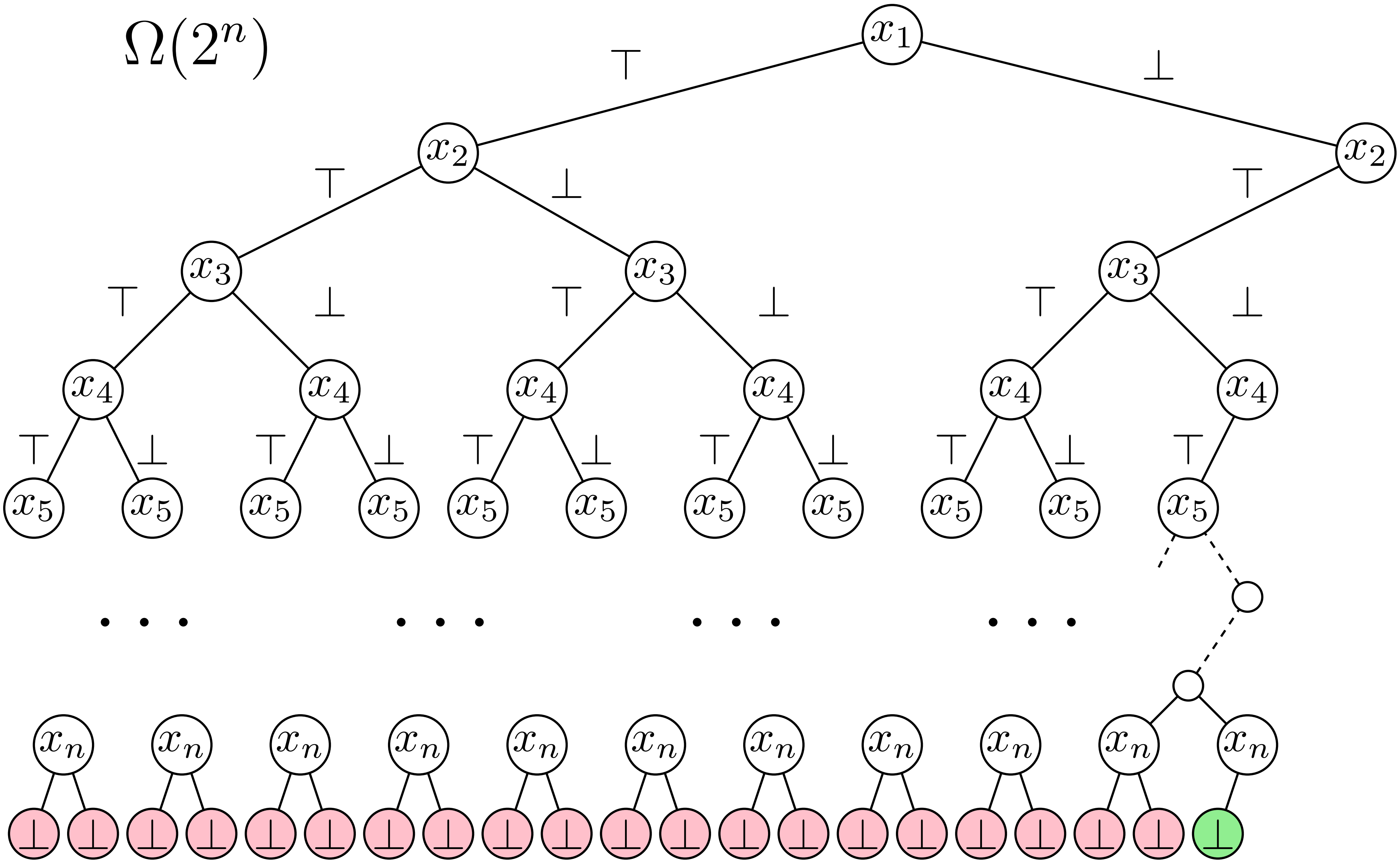
- If $\exists$ unassigned variable x_i
    - Set x_i to $\top$
    - If Bruteforce(): Return `true`
    - Set x_i to $\bot$
    - If Bruteforce(): Return `true`
    - Set x_i to “unassigned” and return `false`
- Else
    - Return `true` $\iff$ ϕ is satisfied.

Time Complexity



Time Complexity

$\Omega(2^n)$



An Observation

- A clause of the form $(\neg x_i \vee x_j)$ corresponds to $x_i \implies x_j$
- If $x_i = \top$ then, in any satisfying assignment, $x_j = \top$
- We say that x_j is **implied**.

- The same holds for any clause $C_j = (c_j^{(1)} \vee c_j^{(2)})$
- If $c_j^{(1)} = \bot$, then $c_j^{(2)} = \top$
- We say that the variable x_k corresponding to $c_j^{(2)}$ **implied**.
- If $c_j^{(2)} = x_k$, then $x_k = \top$. If $c_j^{(2)} = \overline{x_k}$, then $x_k = \bot$.

Example

$$\phi = (x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_4})$$

Example

$$\phi = (x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_4})$$

Assume that $x_1 = \top$

Example

$$\phi = (x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_4) \wedge (\overline{x_1} \vee \underline{\overline{x_3}}) \wedge (x_3 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \underline{\overline{x_4}})$$

Assume that $x_1 = \top$

x_3 and x_4 are implied. $x_3 = \bot$ and $x_4 = \bot$

Example

$$\phi = (x_1 \vee \overline{x_2}) \wedge (\underline{x_2} \vee x_4) \wedge (\overline{x_1} \vee \underline{\overline{x_3}}) \wedge (x_3 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \underline{\overline{x_4}})$$

Assume that $x_1 = \top$

x_3 and x_4 are implied. $x_3 = \bot$ and $x_4 = \bot$

$x_2 = \top$ is implied

Example

$$\phi = (x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_4})$$

Assume that $x_1 = \top$

x_3 and x_4 are implied. $x_3 = \bot$ and $x_4 = \bot$

$x_2 = \top$ is implied

$x_2 = \bot$ is implied, a contradiction!

Example

$$\phi = (x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_4})$$

Assume that $x_1 = \bot$

Example

$$\phi = (x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_4})$$

Assume that $x_1 = \bot$

$x_2 = \bot$ is implied.

Example

$$\phi = (x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_4})$$

Assume that $x_1 = \bot$

$x_2 = \bot$ is implied.

$x_4 = \top$ is implied.

Example

$$\phi = (x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_3 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_4})$$

Assume that $x_1 = \bot$

$x_2 = \bot$ is implied.

$x_4 = \top$ is implied.

We found a satisfying assignment.

A Better Algorithm?

A Better Algorithm?

Idea: Only branch on non-implied variables.

Bruteforce():

- If $\exists$ unassigned variable x_i:
  - Set x_i to $\top$
  - Iteratively set all implied variables
  - If no contradiction is found and Bruteforce():
    - Return `true`
  - Revert all changes
  - Set x_i to $\bot$
  - Iteratively set all implied variables
  - If no contradiction is found and Bruteforce():
    - Return `true`
  - Revert all changes and return `false`
- Else
  - Return `true`

A Better Algorithm?

Idea: Only branch on non-implied variables.

Bruteforce():

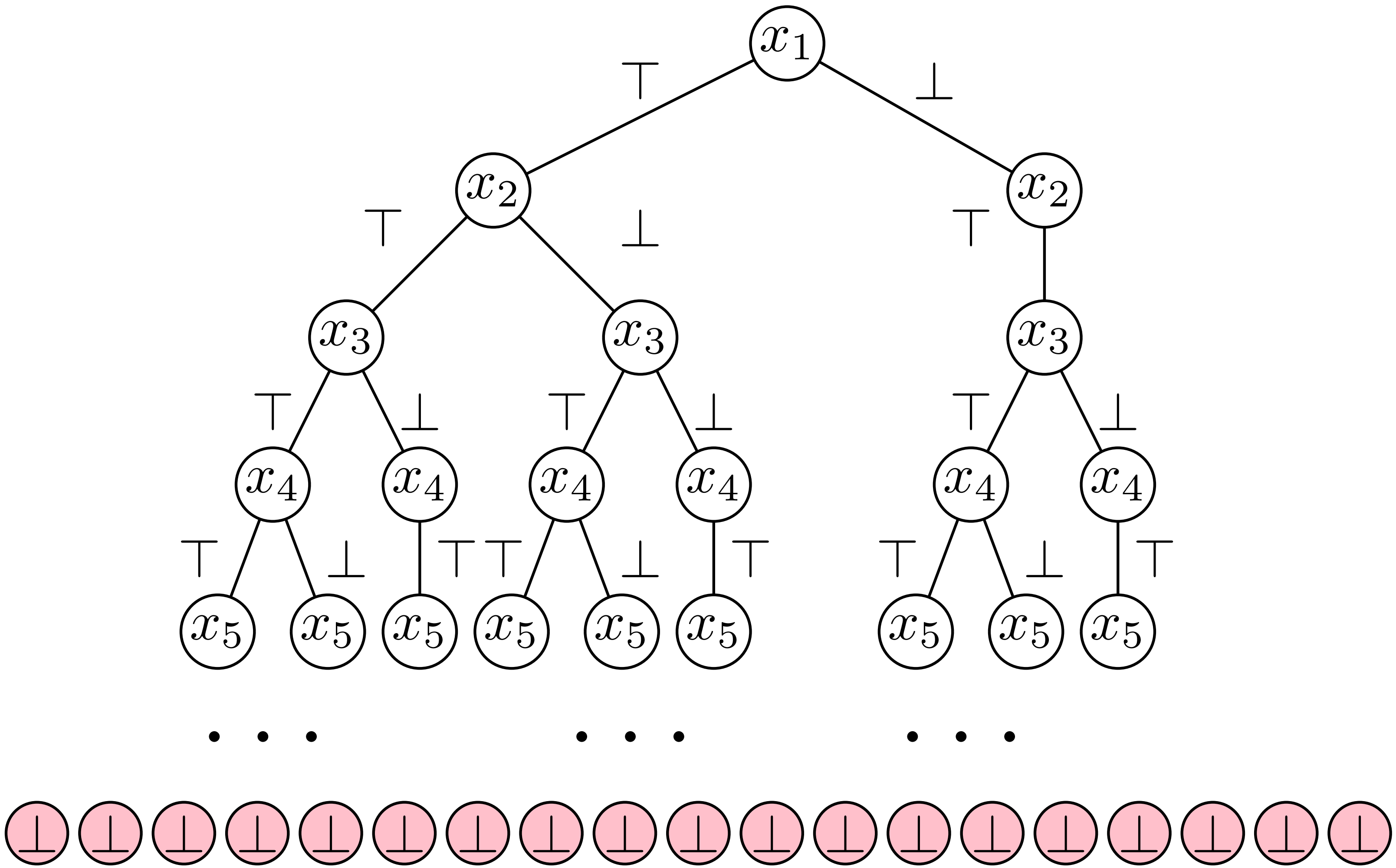
- If $\exists$ unassigned variable x_i: ← x_i is not implied
  - Set x_i to $\top$
  - Iteratively set all implied variables
  - If no contradiction is found and Bruteforce():
    - Return `true`
  - Revert all changes
  - Set x_i to $\bot$
  - Iteratively set all implied variables
  - If no contradiction is found and Bruteforce():
    - Return `true`
  - Revert all changes and return `false`
- Else
  - Return `true` ← No variables left to assign and no contradictions

Time Complexity

$$\phi = \ (x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge \cdots \wedge (x_{n-1}, x_n)$$
$$\wedge (x_{n-1} \vee \overline{x_n}) \wedge (\overline{x_{n-1}} \vee x_n) \wedge (\overline{x_{n-1}} \vee \overline{x_n})$$

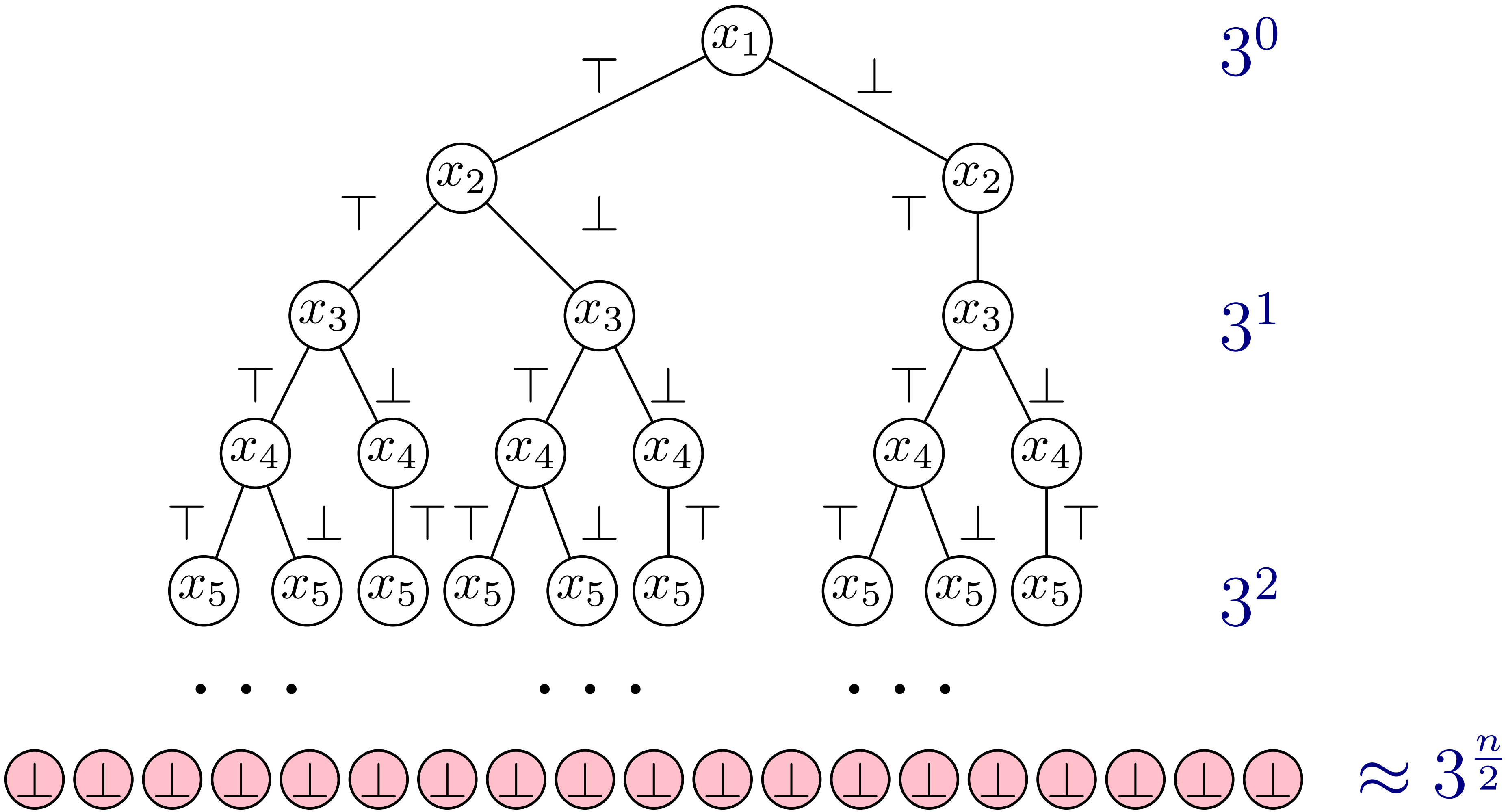
Time Complexity

$$\phi = (x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge \cdots \wedge (x_{n-1}, x_n)$$
$$\wedge (x_{n-1} \vee \overline{x_n}) \wedge (\overline{x_{n-1}} \vee x_n) \wedge (\overline{x_{n-1}} \vee \overline{x_n})$$



Time Complexity

$$\phi = \ (x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge \cdots \wedge (x_{n-1}, x_n)$$
$$\wedge\, (x_{n-1} \vee \overline{x_n}) \wedge (\overline{x_{n-1}} \vee x_n) \wedge (\overline{x_{n-1}} \vee \overline{x_n})$$



Limited Backtracking

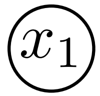
Bruteforce():

- If $\exists$ unassigned variable x_i:
  - Set x_i to $\top$
  - Iteratively set all implied variables
  - If no contradiction is found and Bruteforce():
    - Return `true`
  - Revert all changes
  - Set x_i to $\bot$
  - Iteratively set all implied variables
  - If no contradiction is found and Bruteforce():
    - Return `true`
  - Abort the whole algorithm and report ϕ as not satisfiable
- Else
  - Return `true`

Example

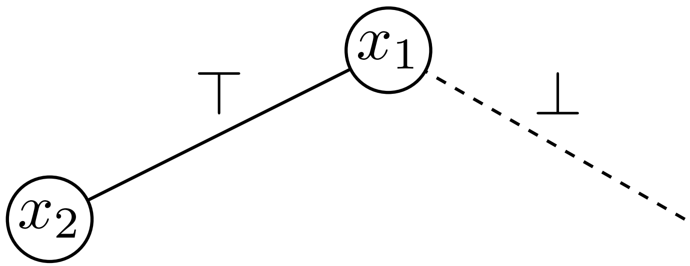
$$\phi = \ (x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge \cdots \wedge (x_{n-1}, x_n)$$

$$\wedge\, (x_{n-1} \vee \overline{x_n}) \wedge (\overline{x_{n-1}} \vee x_n) \wedge (\overline{x_{n-1}} \vee \overline{x_n})$$



Example

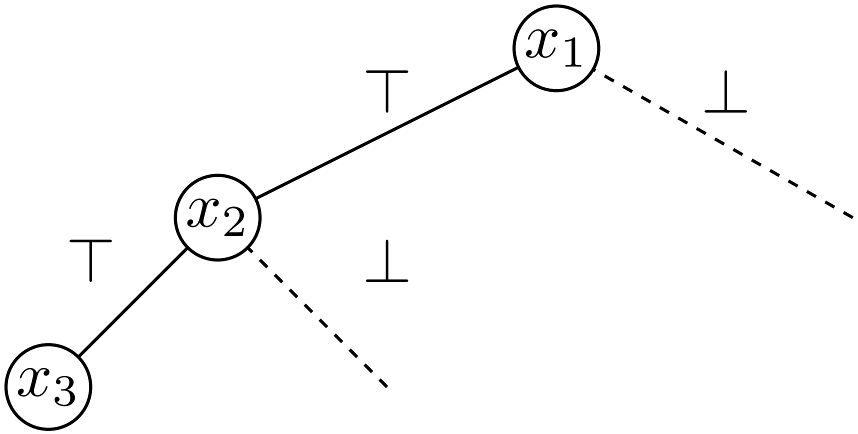
$$\phi = \ (x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge \cdots \wedge (x_{n-1}, x_n)$$
$$\wedge (x_{n-1} \vee \overline{x_n}) \wedge (\overline{x_{n-1}} \vee x_n) \wedge (\overline{x_{n-1}} \vee \overline{x_n})$$



Example

$$\phi = \; (x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge \cdots \wedge (x_{n-1}, x_n)$$

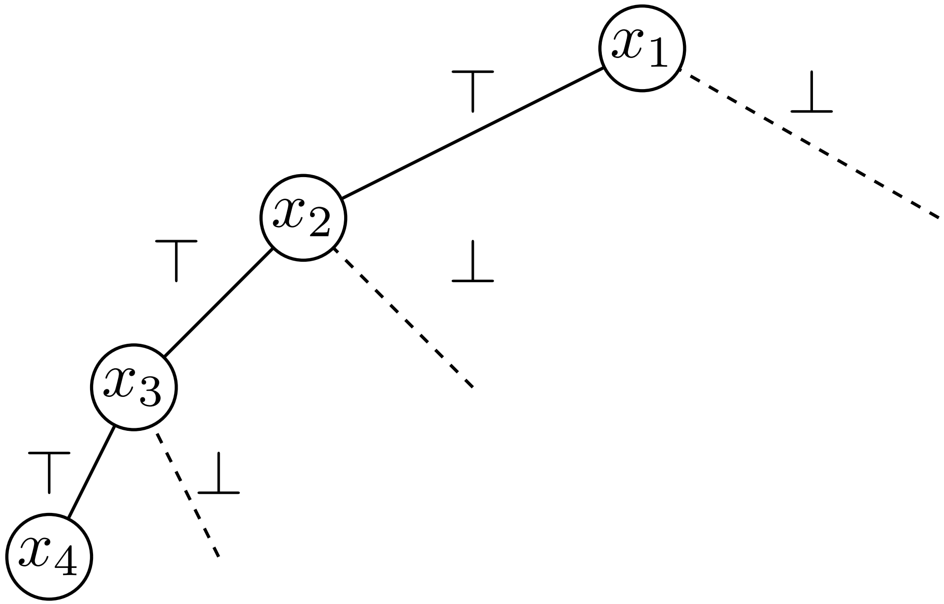
$$\wedge\, (x_{n-1} \vee \overline{x_n}) \wedge (\overline{x_{n-1}} \vee x_n) \wedge (\overline{x_{n-1}} \vee \overline{x_n})$$



Example

$$\phi = \ (x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge \cdots \wedge (x_{n-1}, x_n)$$

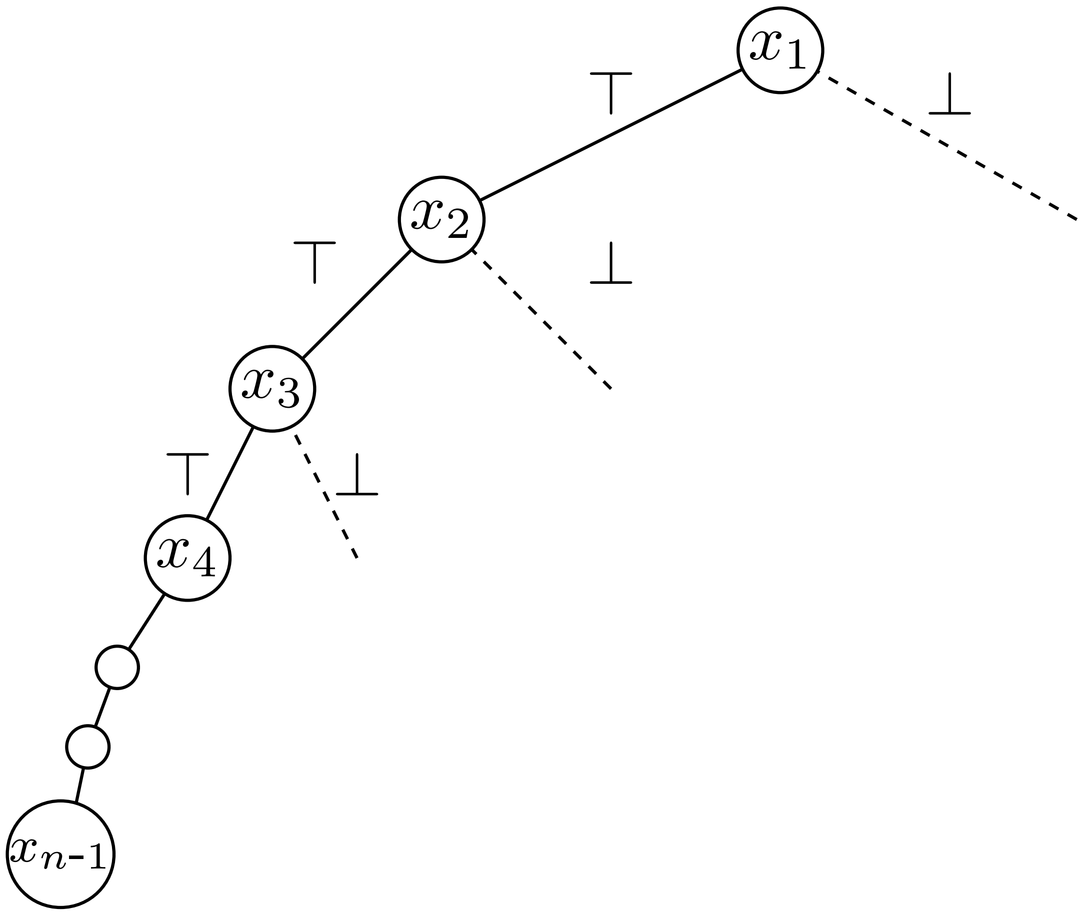
$$\wedge\, (x_{n-1} \vee \overline{x_n}) \wedge (\overline{x_{n-1}} \vee x_n) \wedge (\overline{x_{n-1}} \vee \overline{x_n})$$



Example

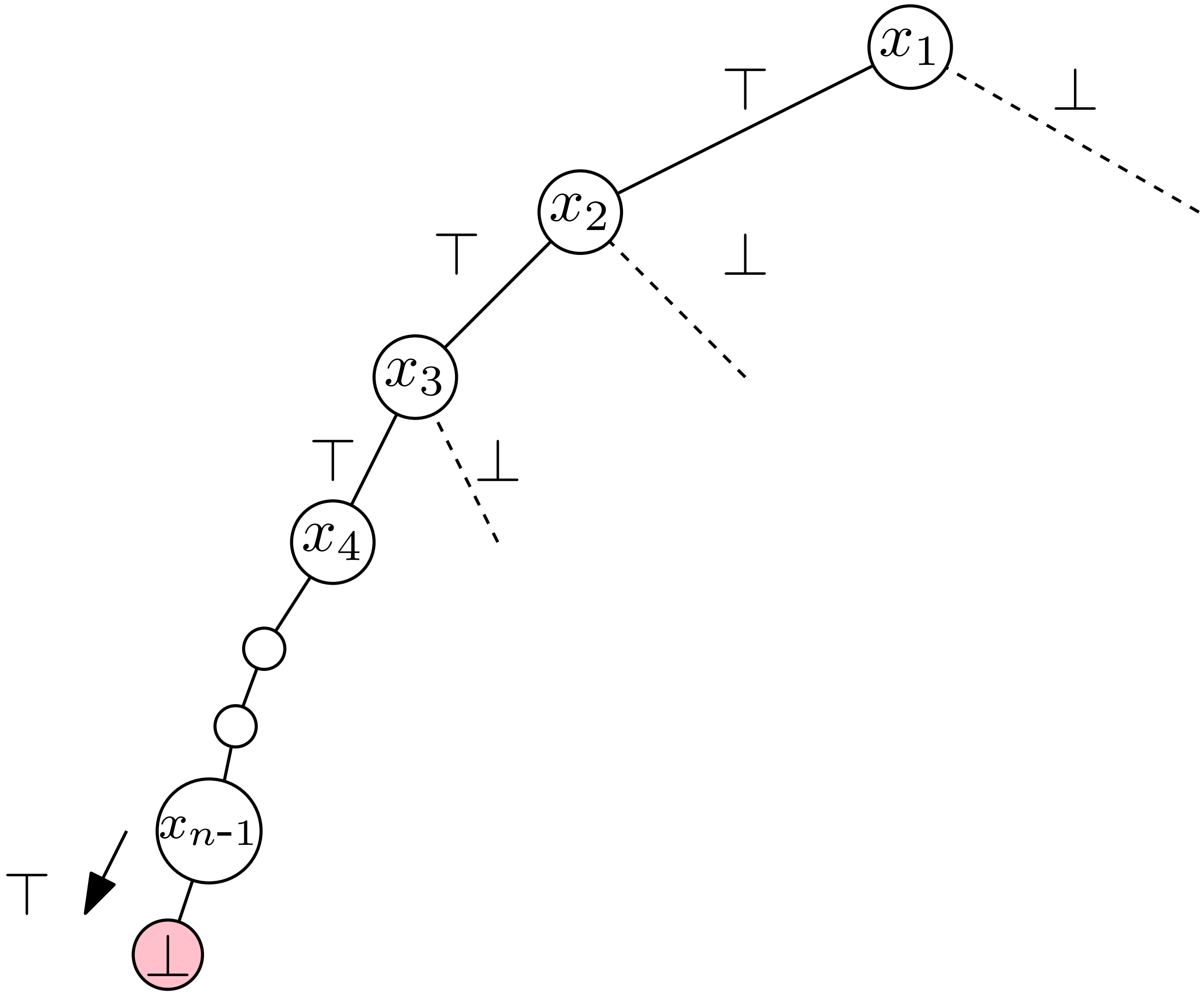
$$\phi = \ (x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge \cdots \wedge (x_{n-1}, x_n)$$

$$\wedge (x_{n-1} \vee \overline{x_n}) \wedge (\overline{x_{n-1}} \vee x_n) \wedge (\overline{x_{n-1}} \vee \overline{x_n})$$



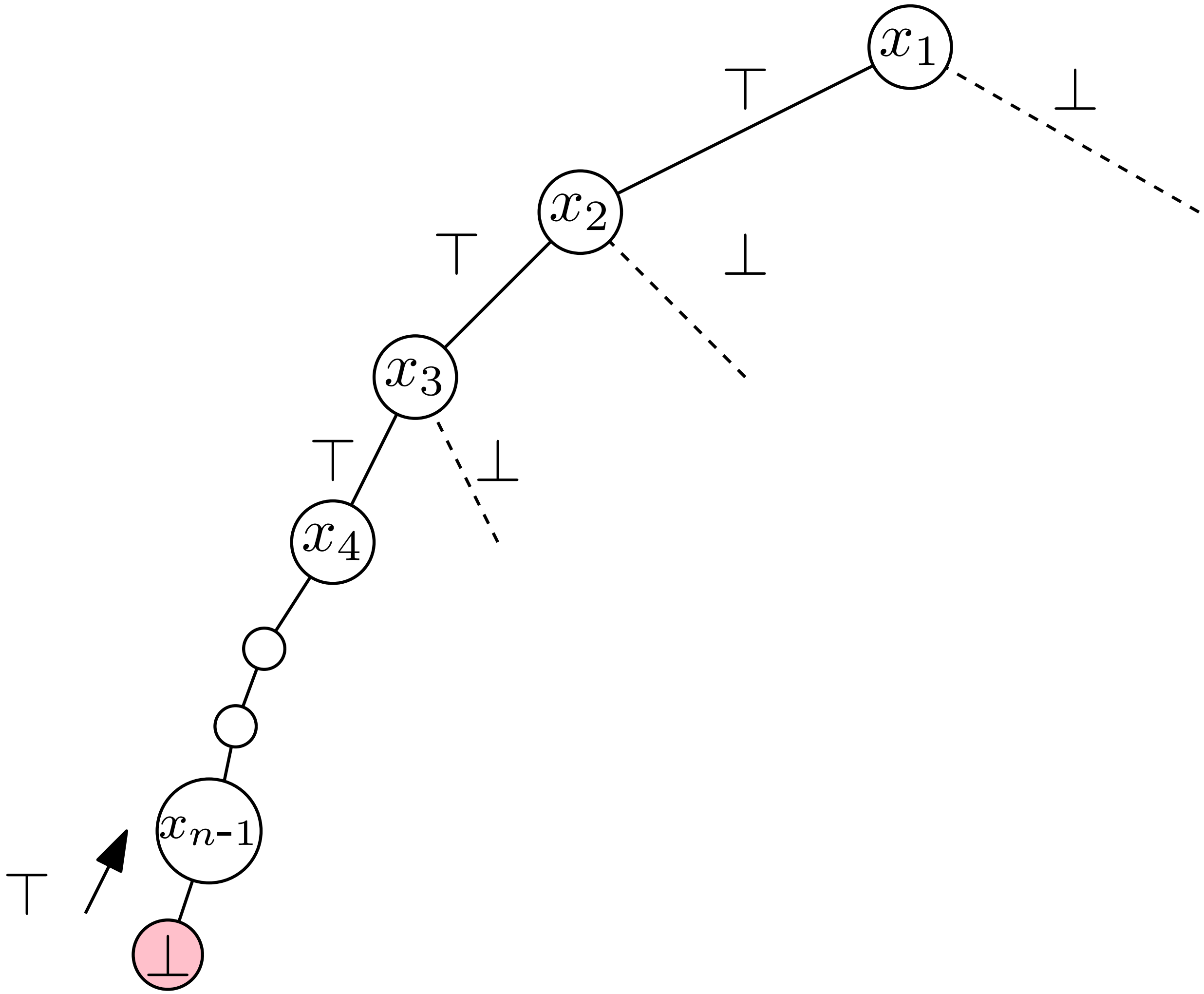
Example

$$\phi = \ (x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge \cdots \wedge (x_{n-1}, x_n)$$
$$\wedge\, (x_{n-1} \vee \overline{x_n}) \wedge (\overline{x_{n-1}} \vee x_n) \wedge (\overline{x_{n-1}} \vee \overline{x_n})$$



Example

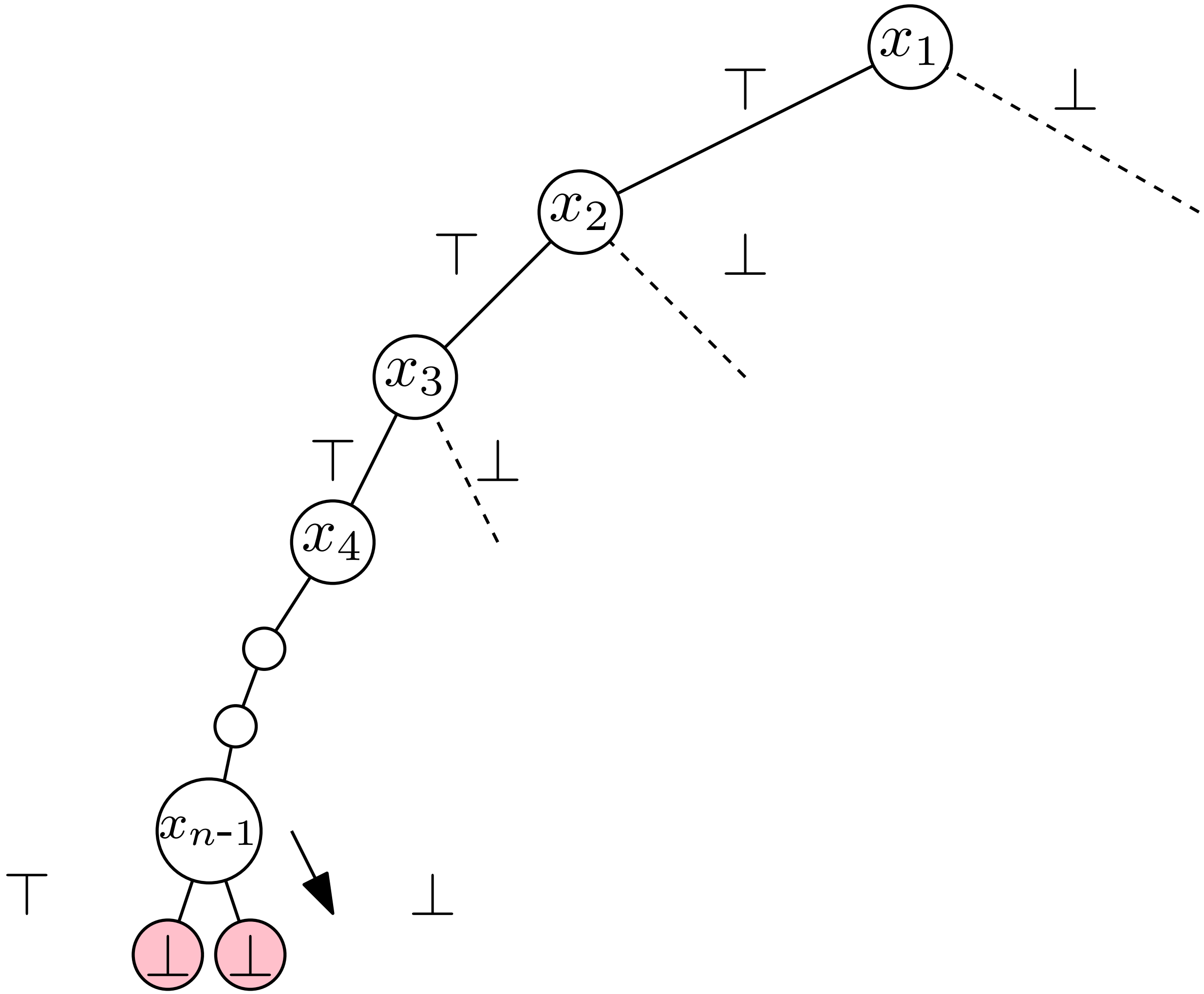
$$\phi = \; (x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge \cdots \wedge (x_{n-1}, x_n)$$
$$\wedge\, (x_{n-1} \vee \overline{x_n}) \wedge (\overline{x_{n-1}} \vee x_n) \wedge (\overline{x_{n-1}} \vee \overline{x_n})$$



Example

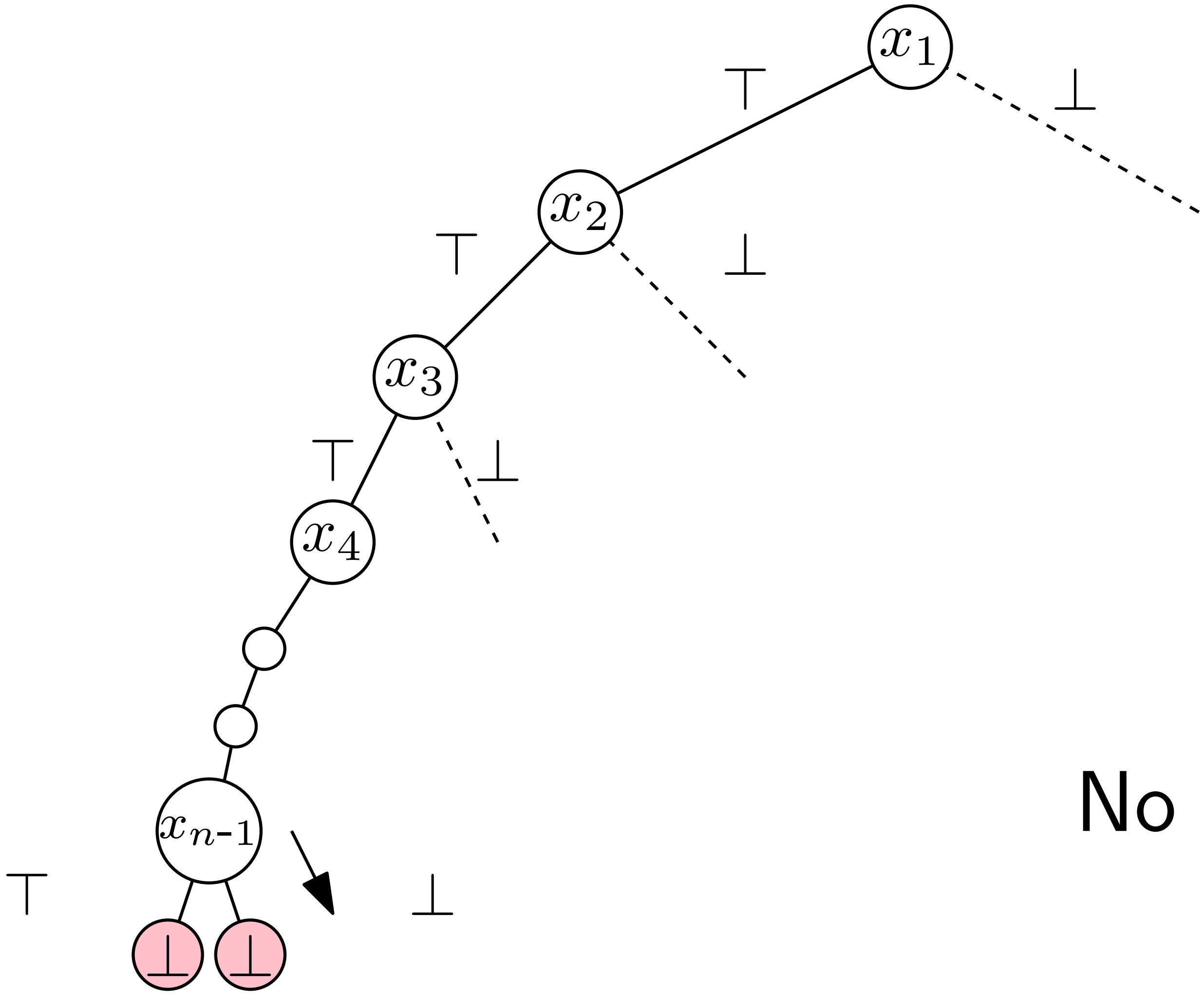
$$\phi = \ (x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge \cdots \wedge (x_{n-1}, x_n)$$

$$\wedge\ (x_{n-1} \vee \overline{x_n}) \wedge (\overline{x_{n-1}} \vee x_n) \wedge (\overline{x_{n-1}} \vee \overline{x_n})$$



Example

$$\phi = \ (x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6) \wedge \cdots \wedge (x_{n-1}, x_n)$$
$$\wedge\, (x_{n-1} \vee \overline{x_n}) \wedge (\overline{x_{n-1}} \vee x_n) \wedge (\overline{x_{n-1}} \vee \overline{x_n})$$



No solution!

Correctness

Claim: If the algorithm returns `true`, then ϕ is satisfiable.

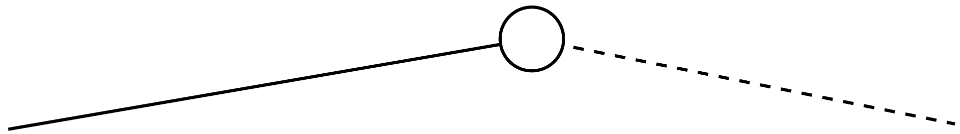
Proof: trivial. The algorithm can only return `true` if a satisfying assignment is constructed.

Claim: If the algorithm aborts, then ϕ is not satisfiable.

- Consder the recursive call that aborted and reported ϕ as unsatisfiable.
- All (previously unassigned) variables that are assigned during this call are not implied by any variable assigned in a previous call.
- Assigning either $\top$ or $\bot$ to x_i leads to a contradiction regardless of the values of the previous variables.

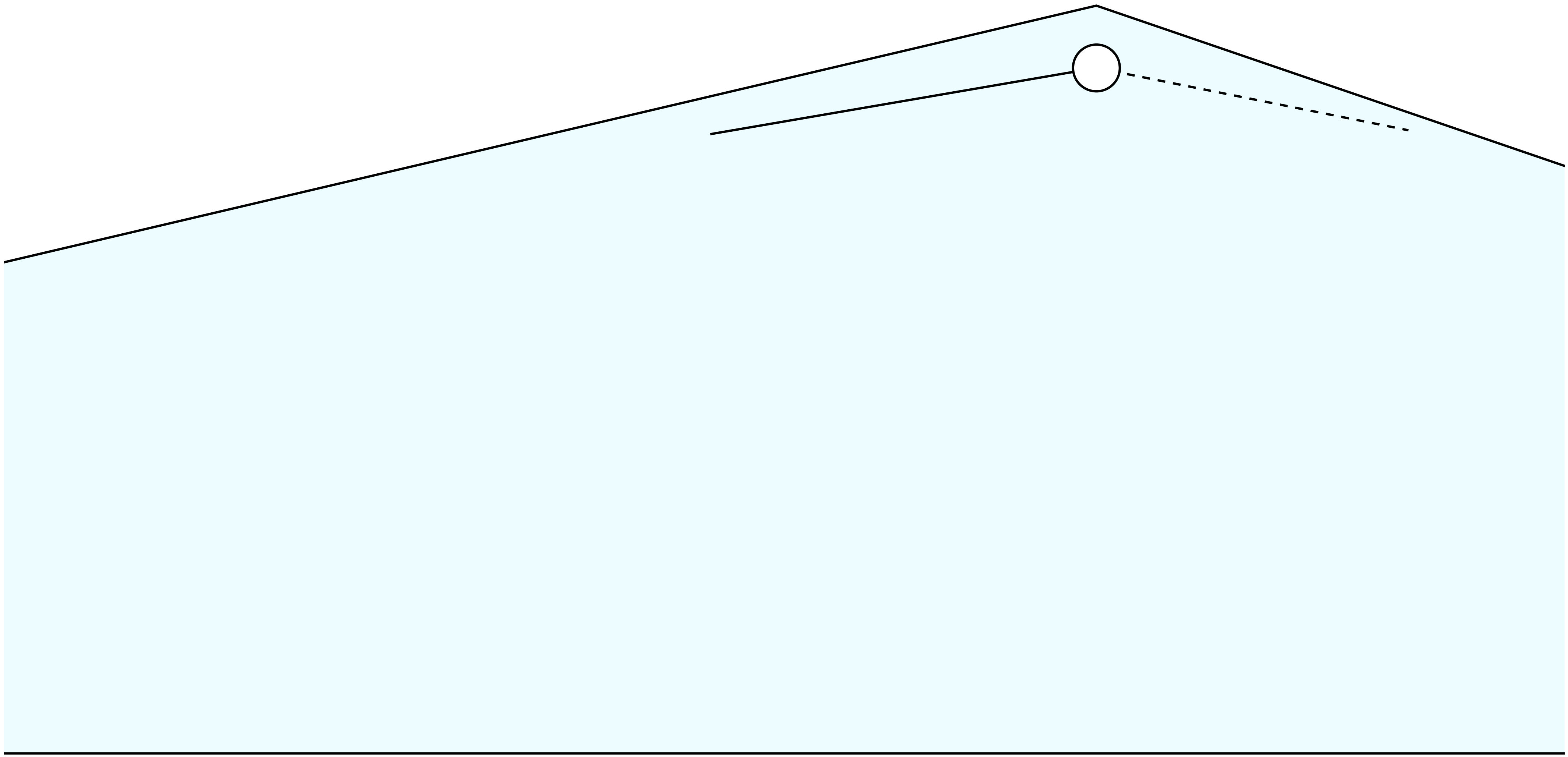
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



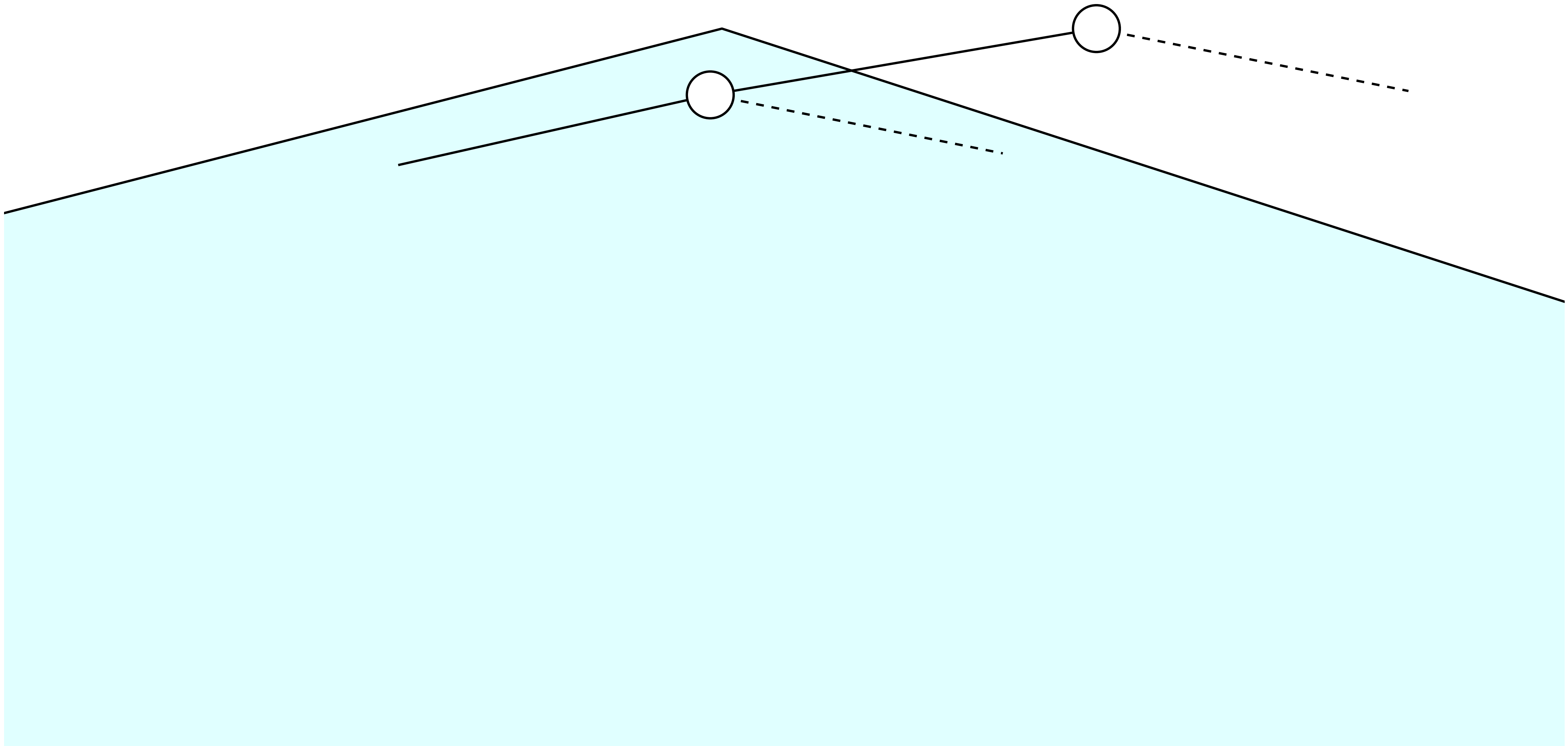
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



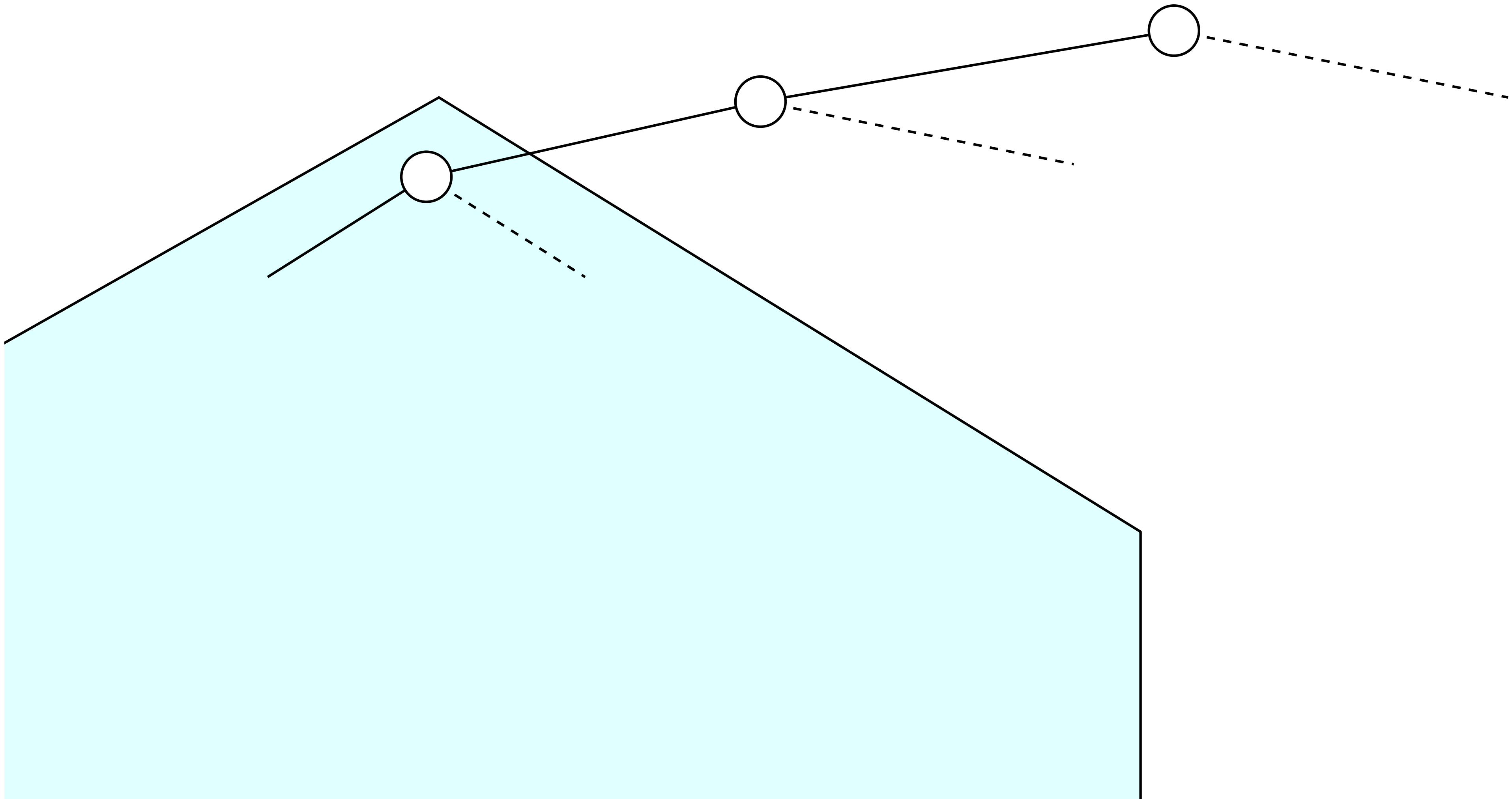
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



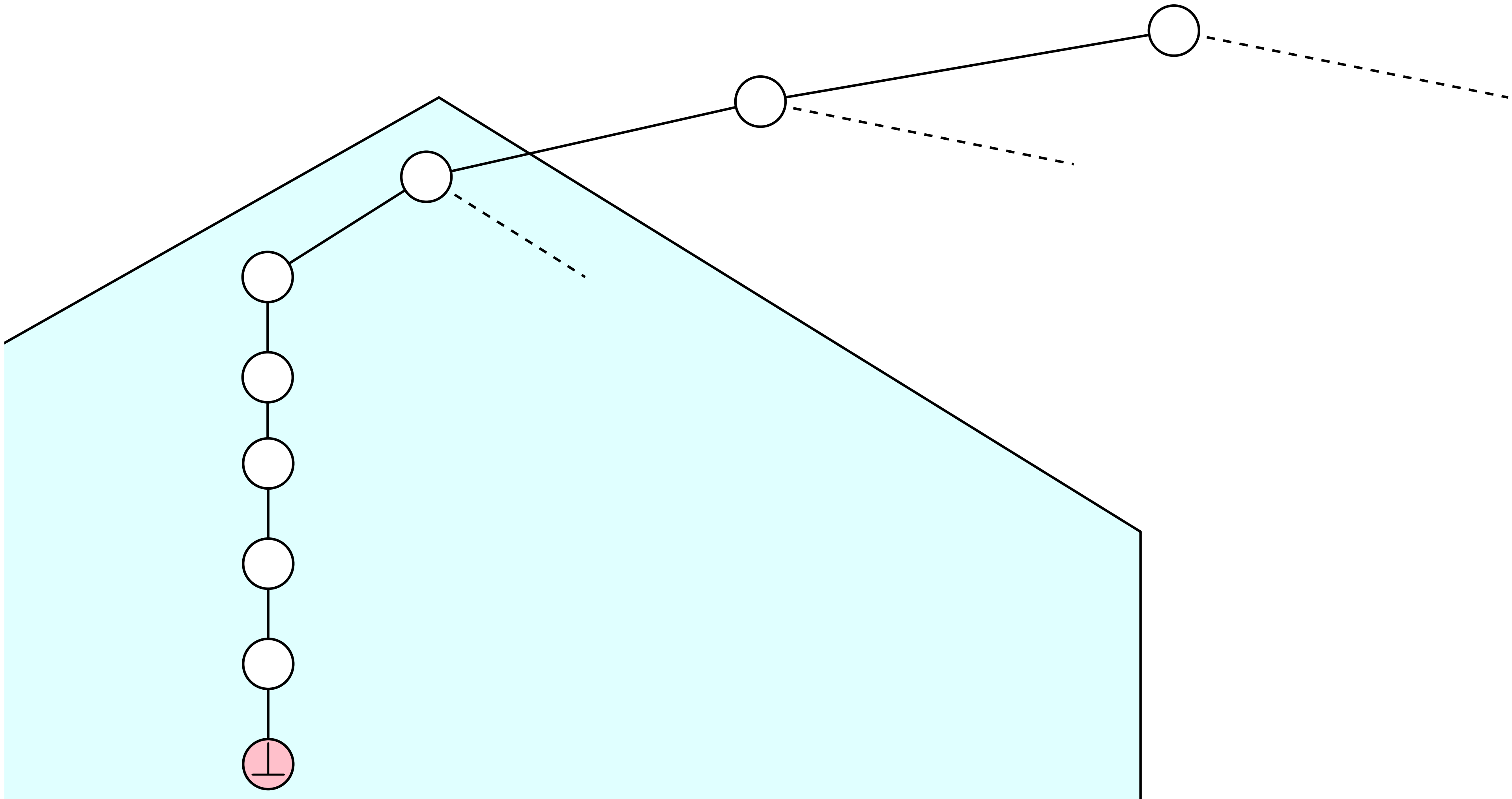
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



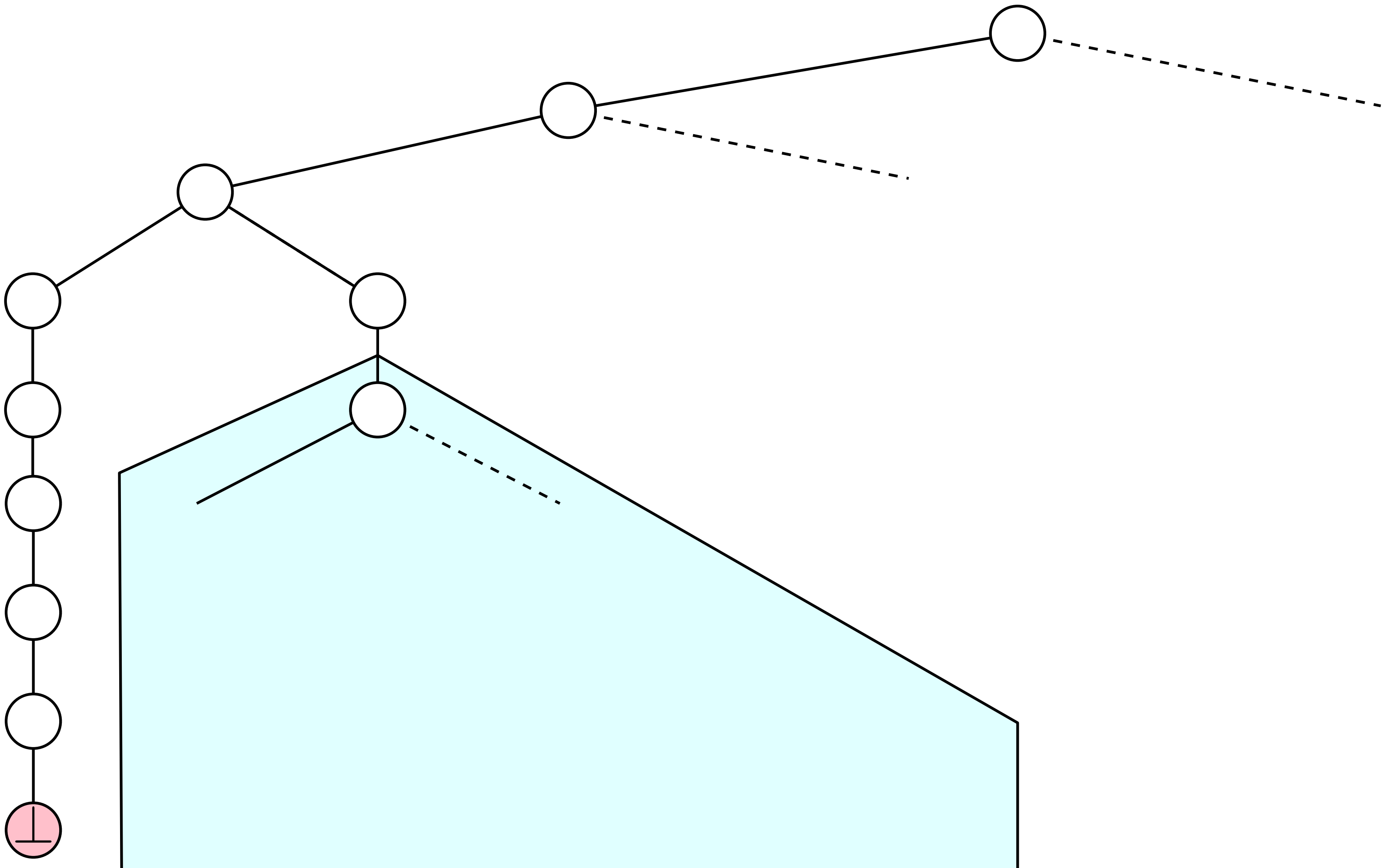
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



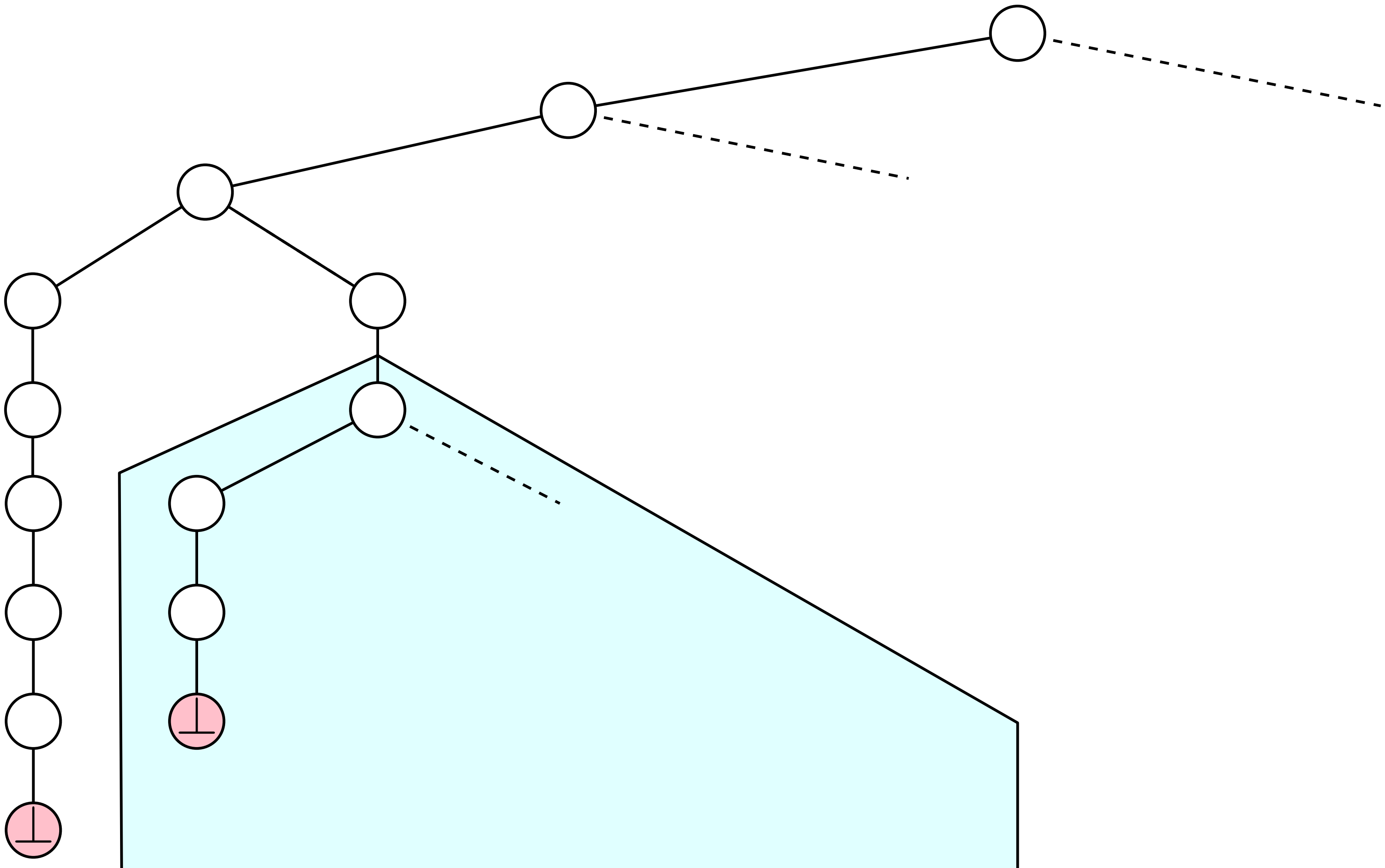
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



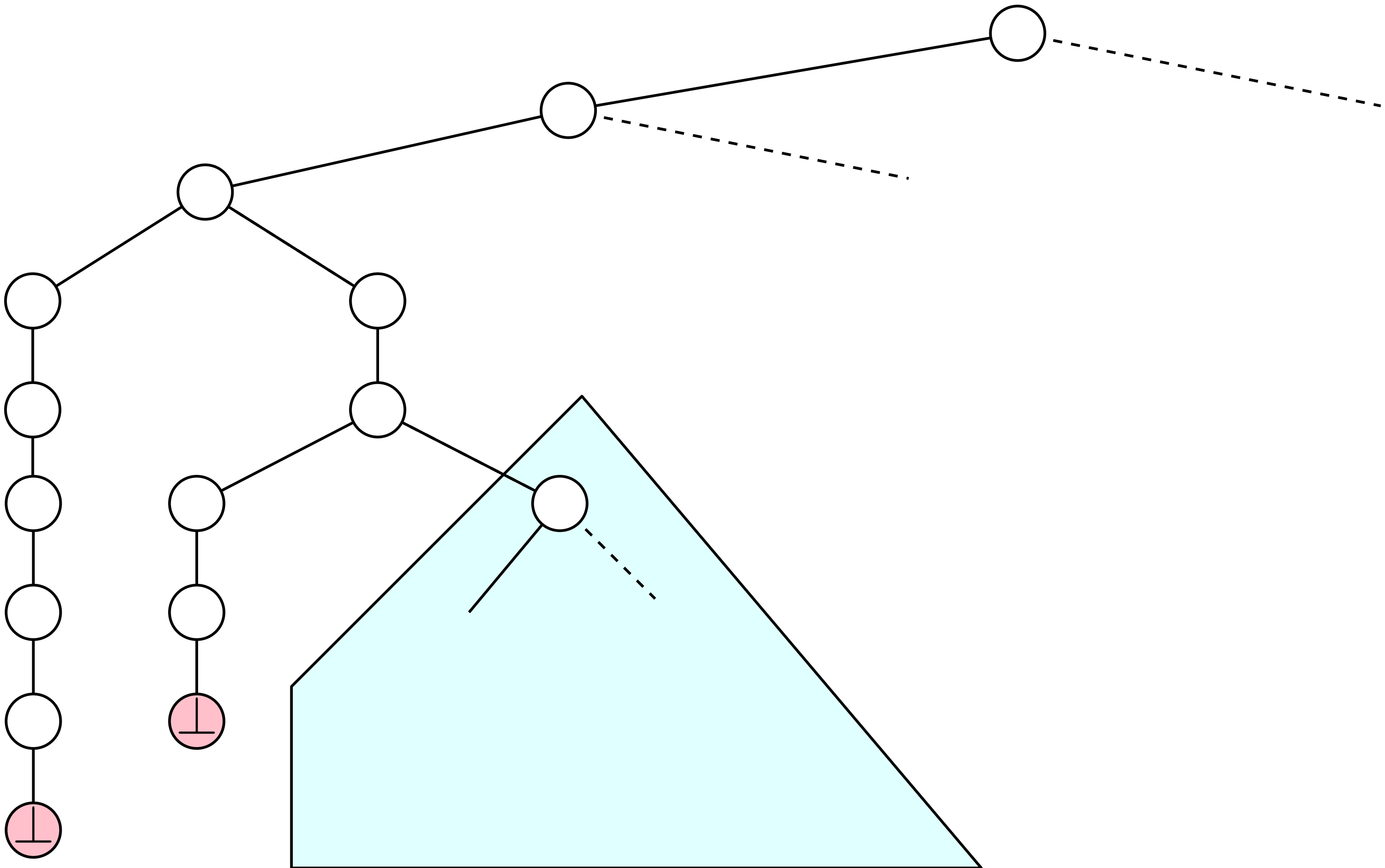
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



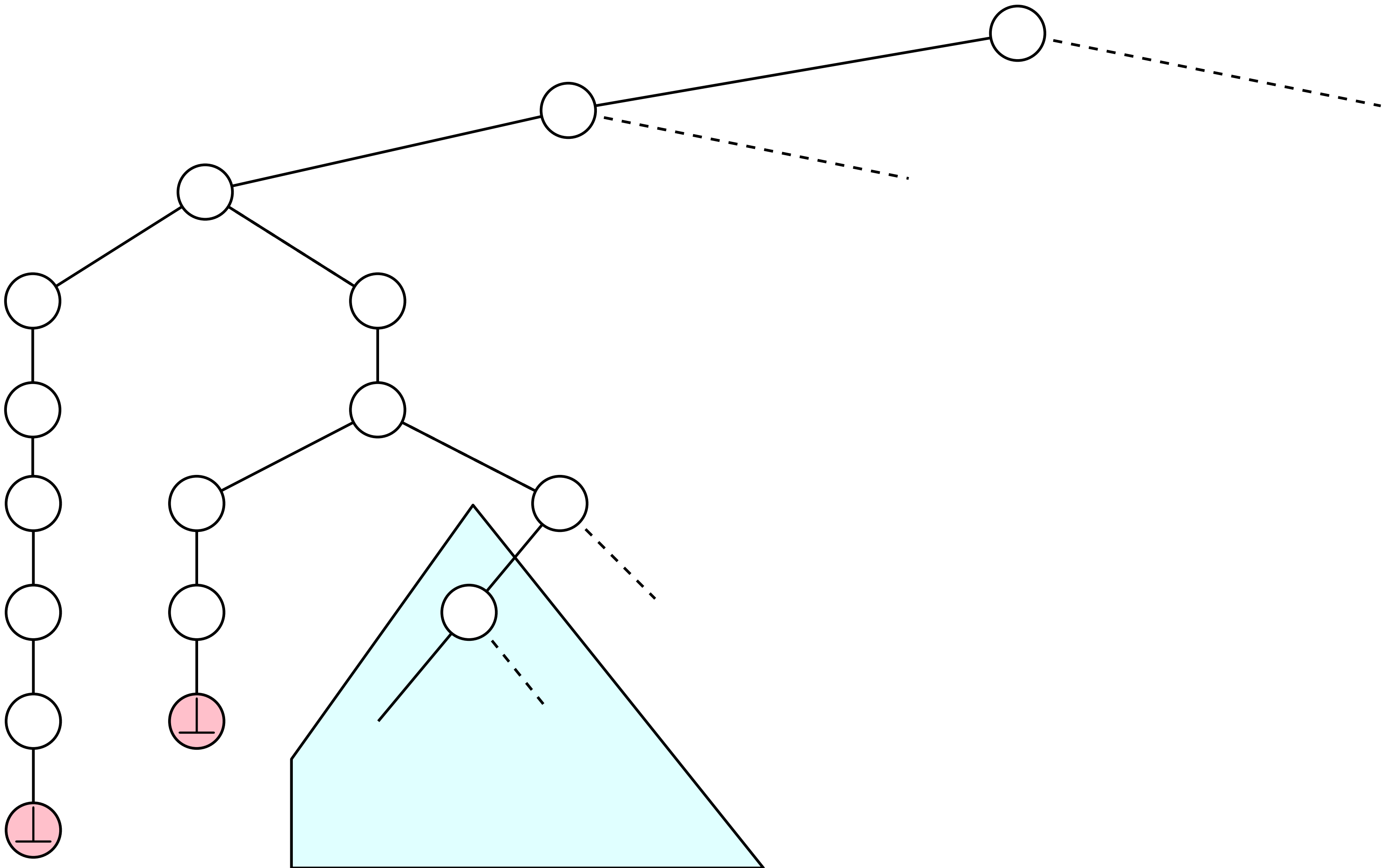
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



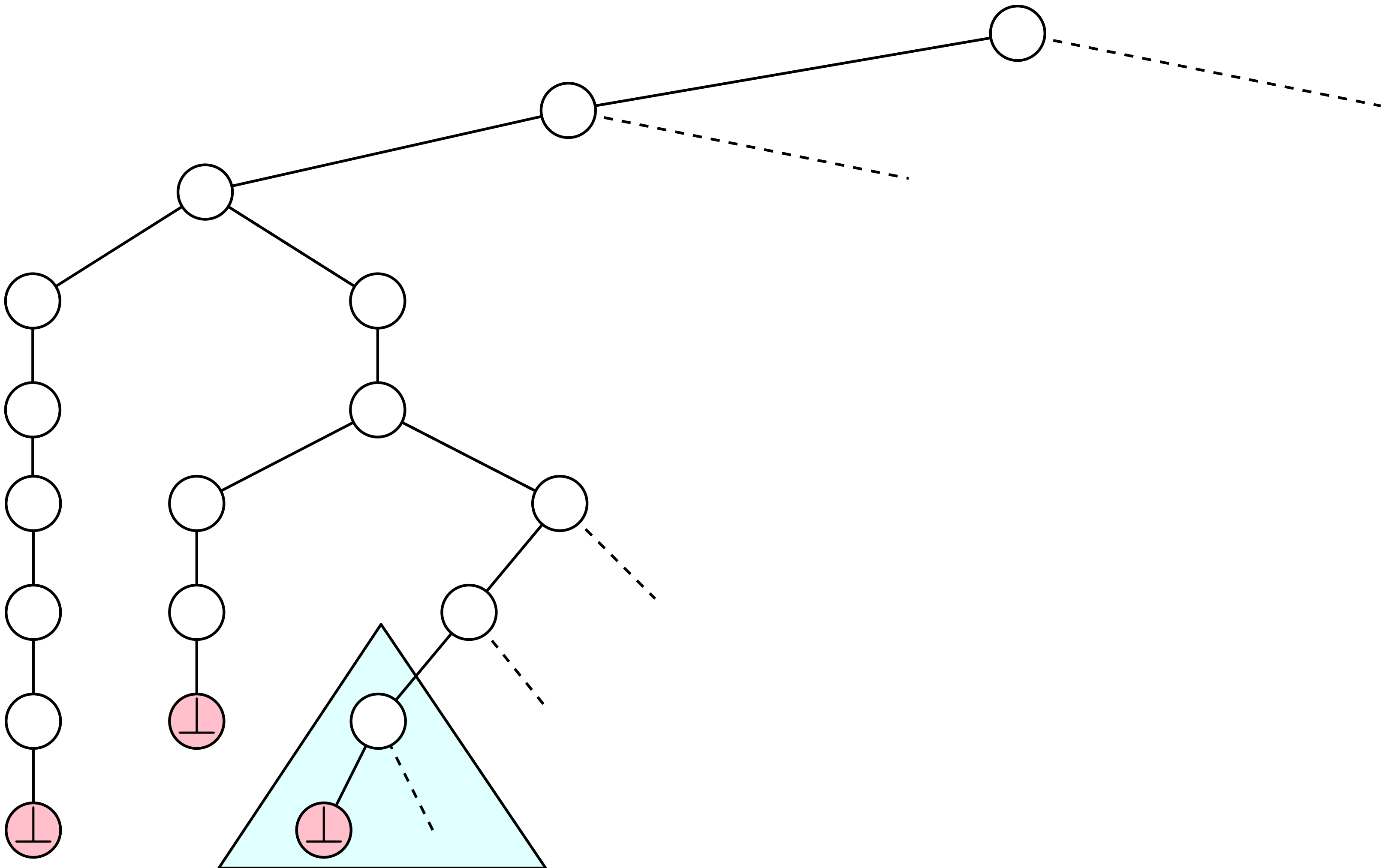
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



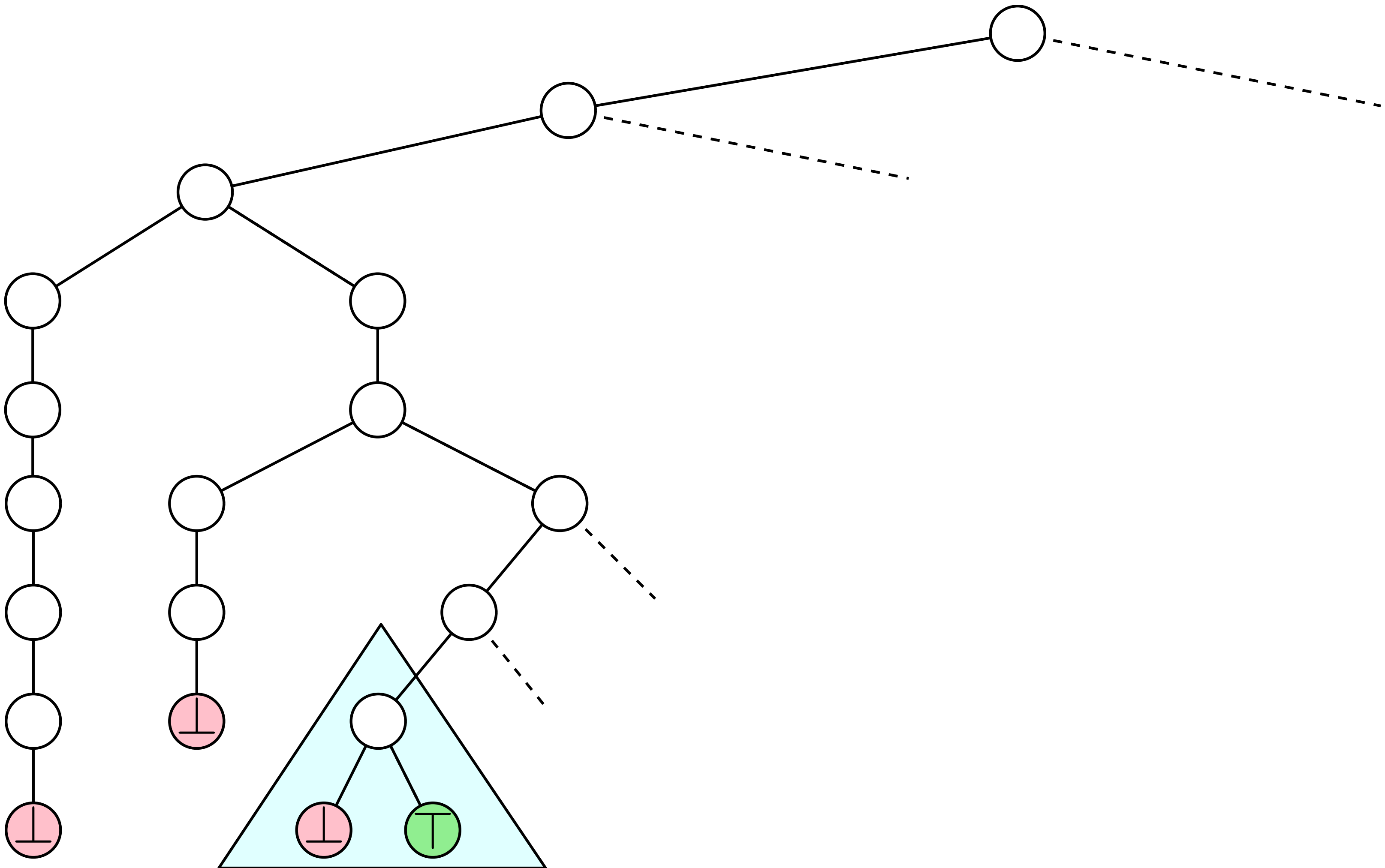
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



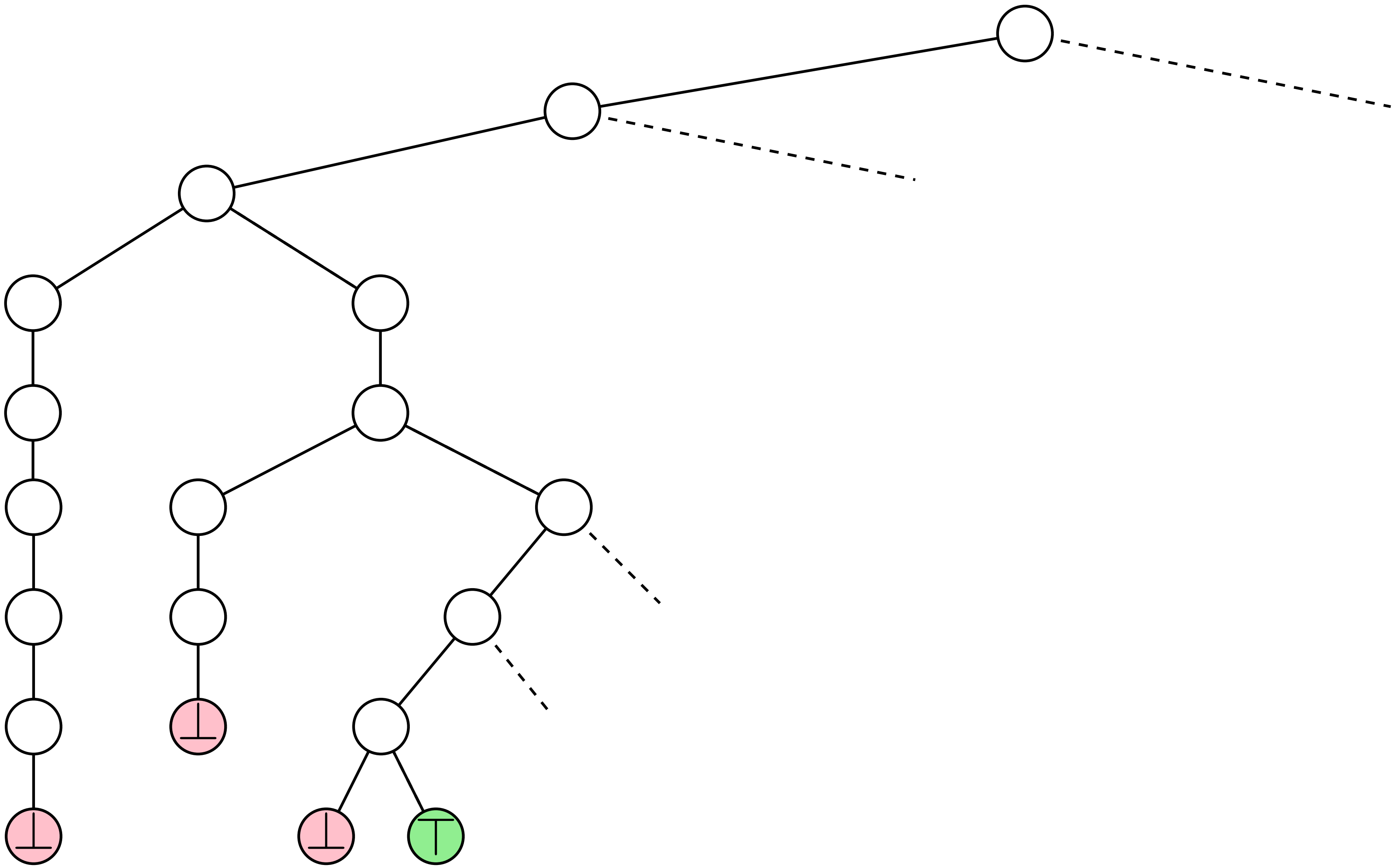
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



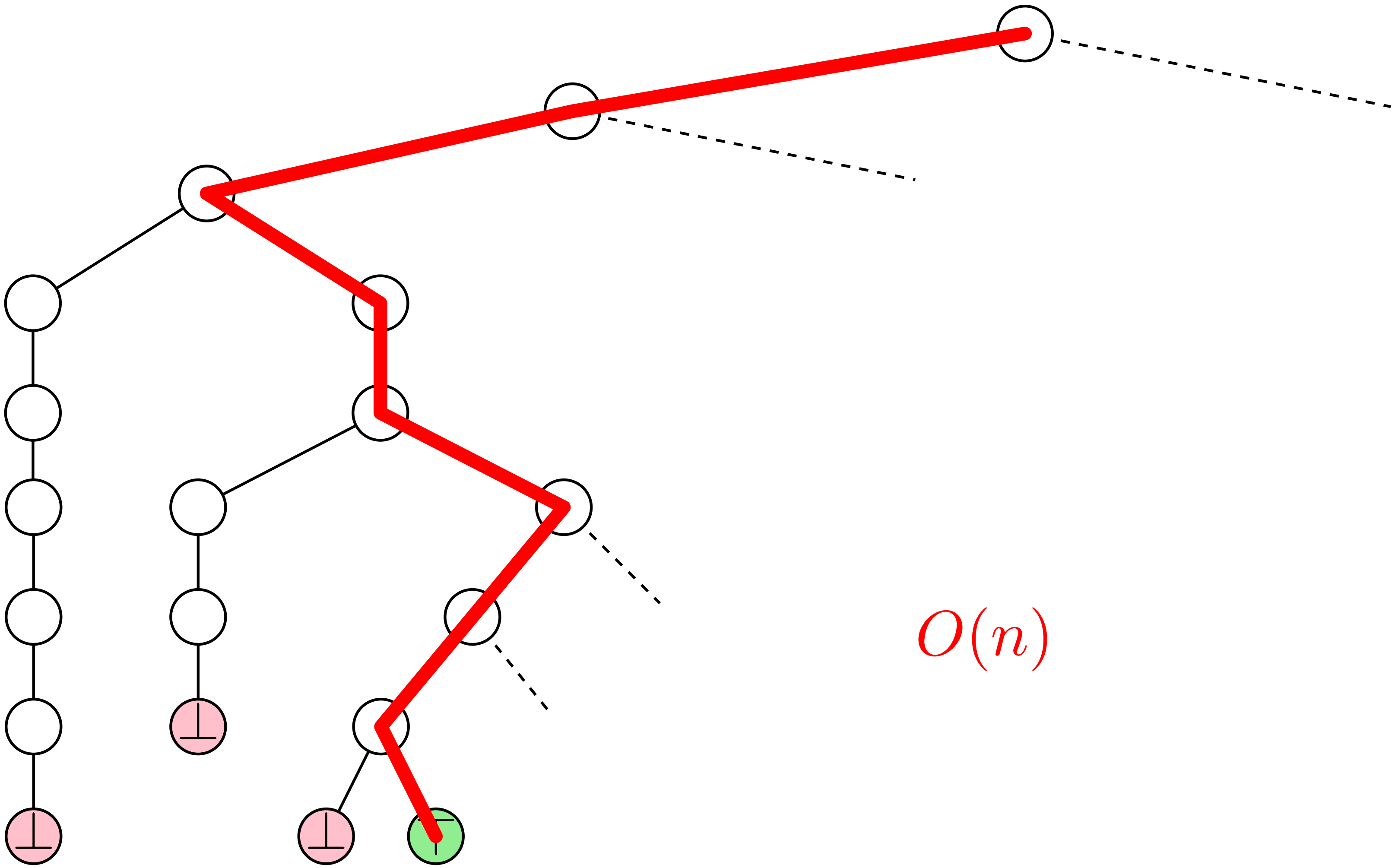
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



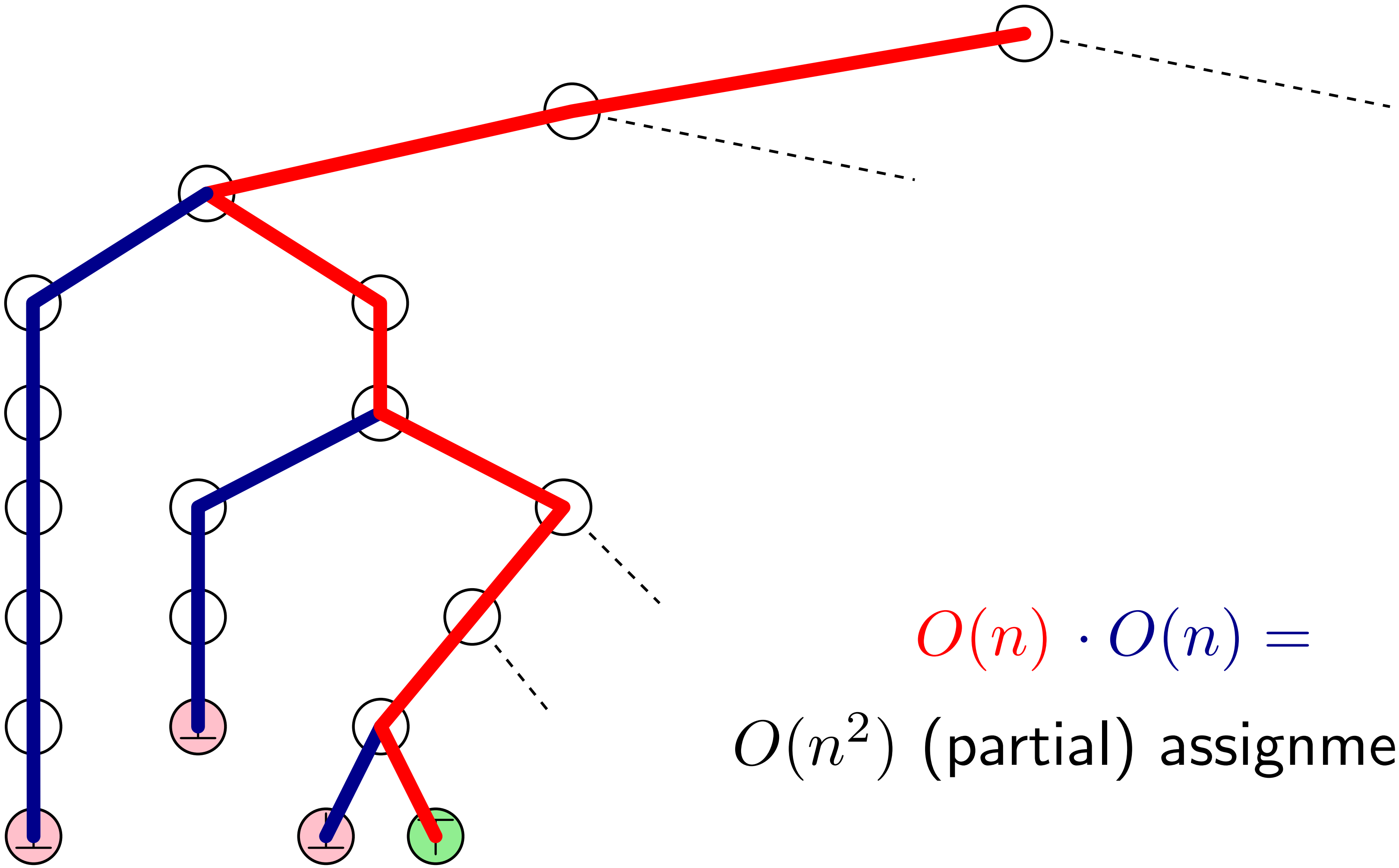
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.

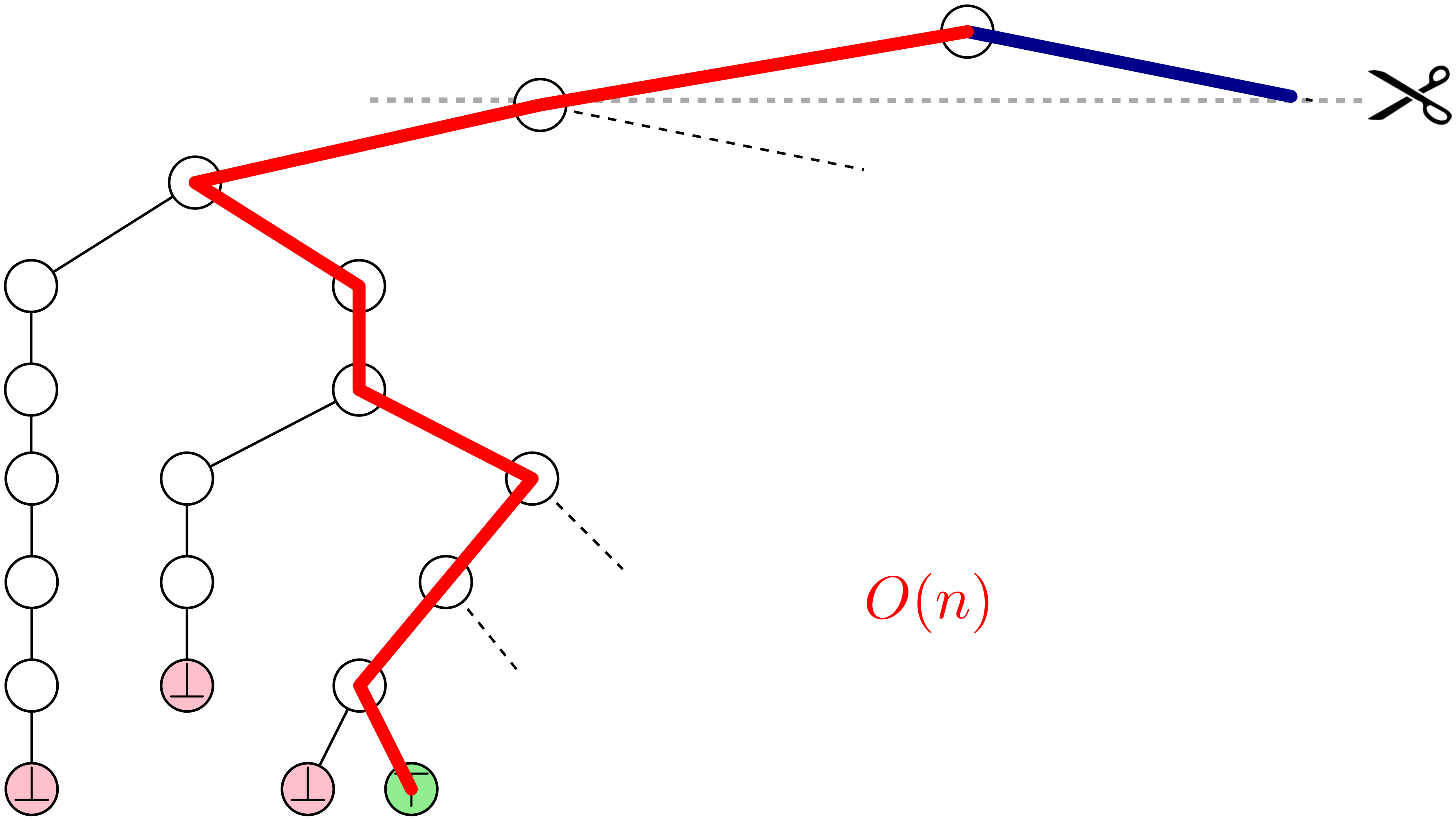


$O(n) \cdot O(n) =$

$O(n^2)$ (partial) assignments

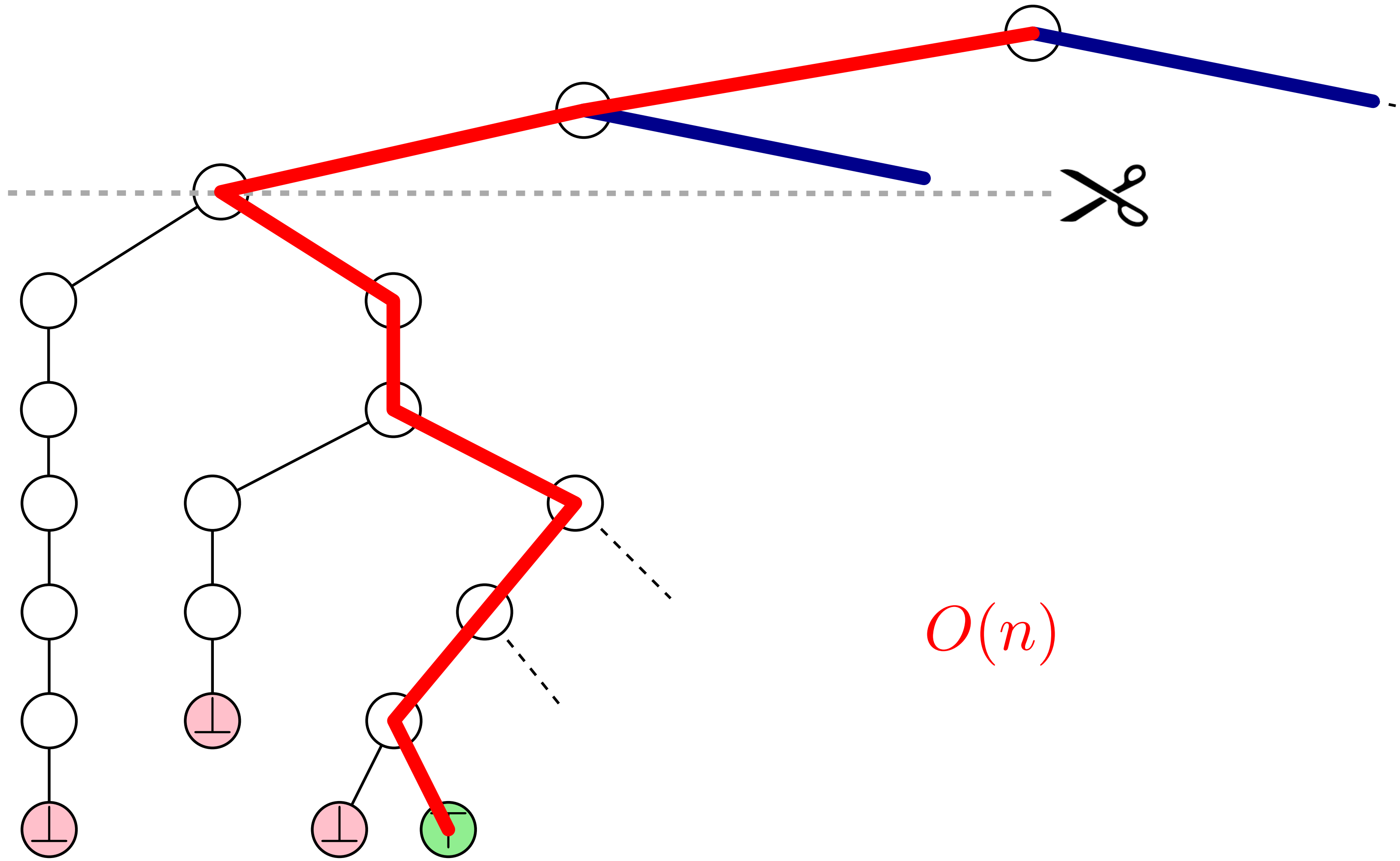
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



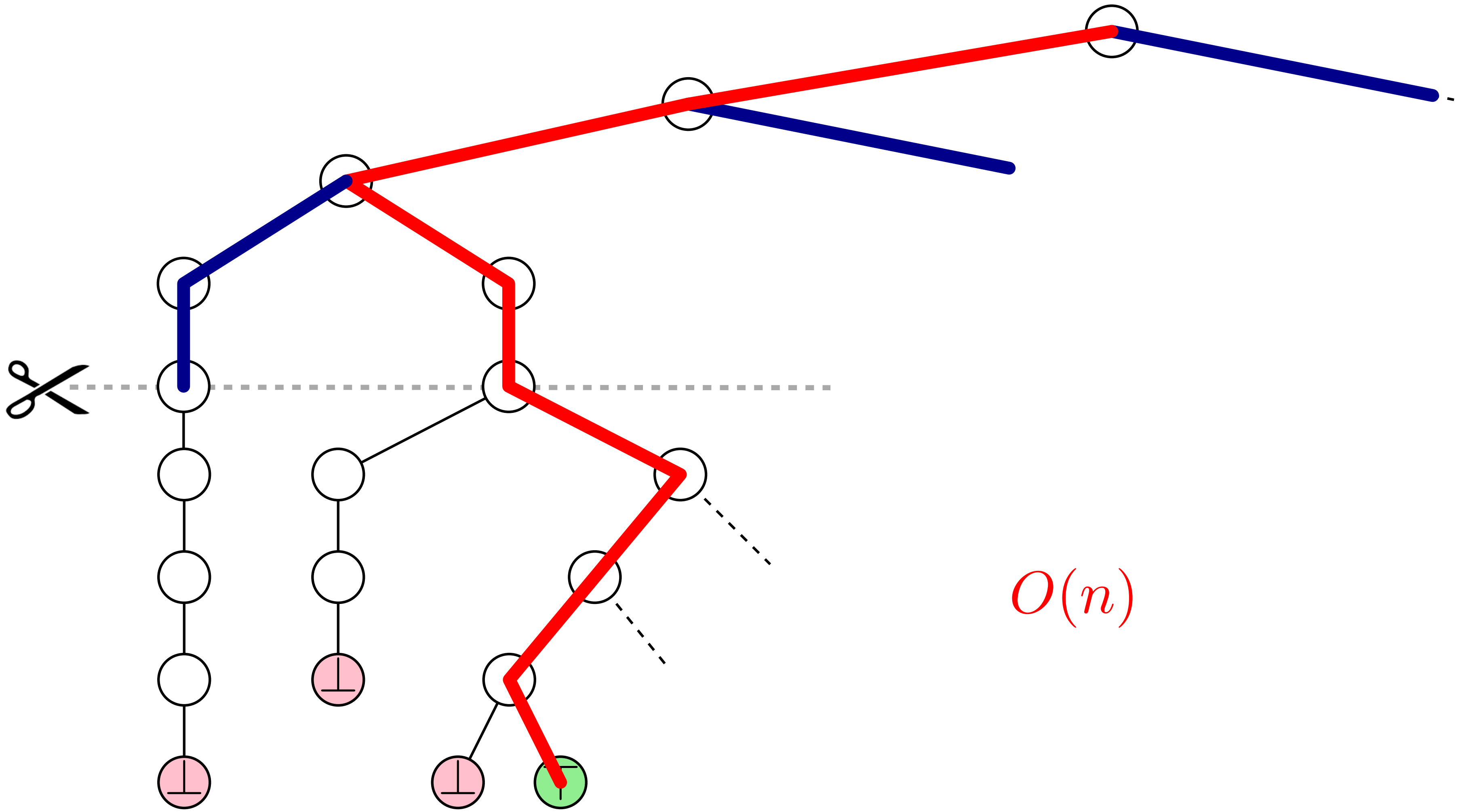
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



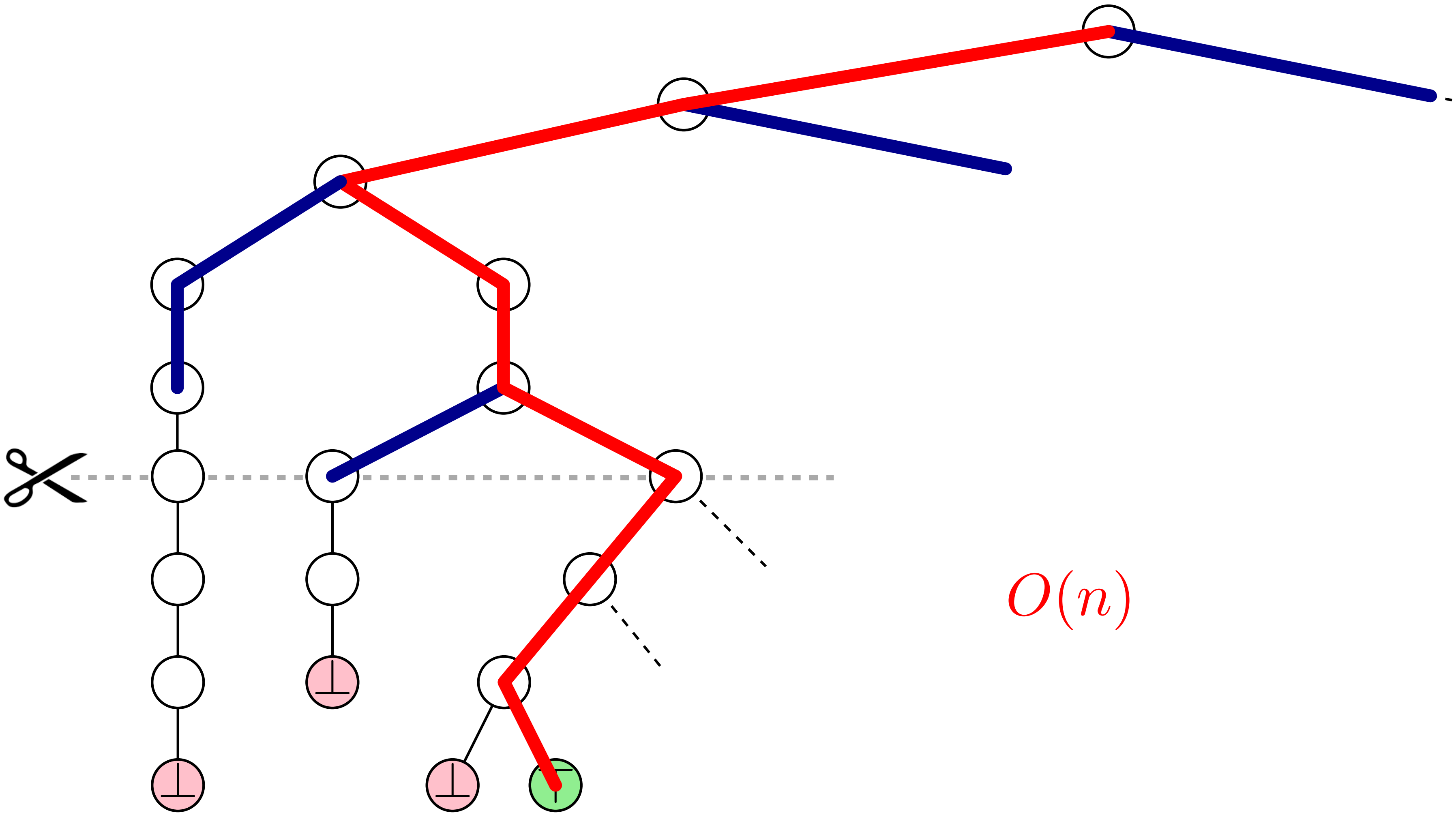
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



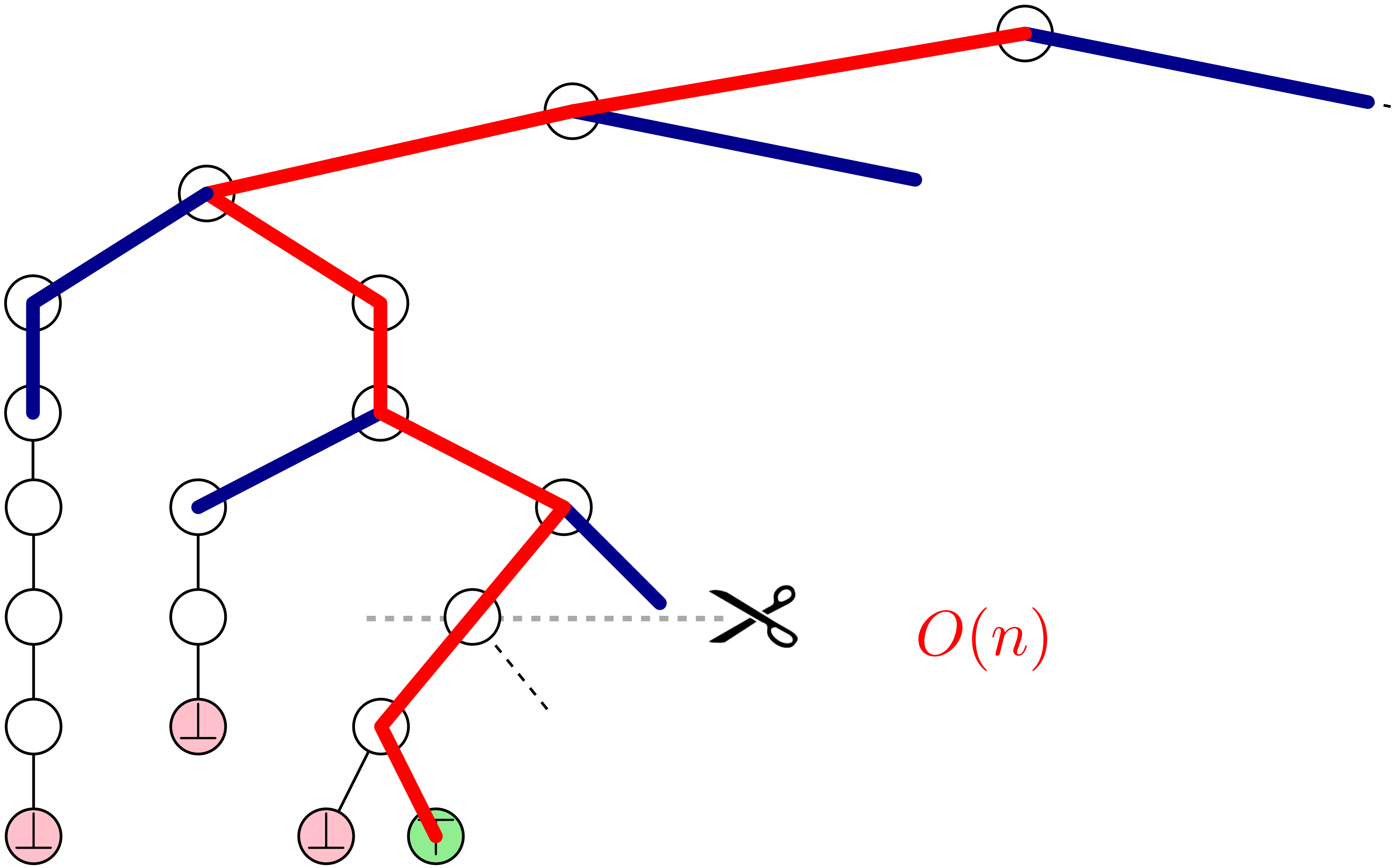
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



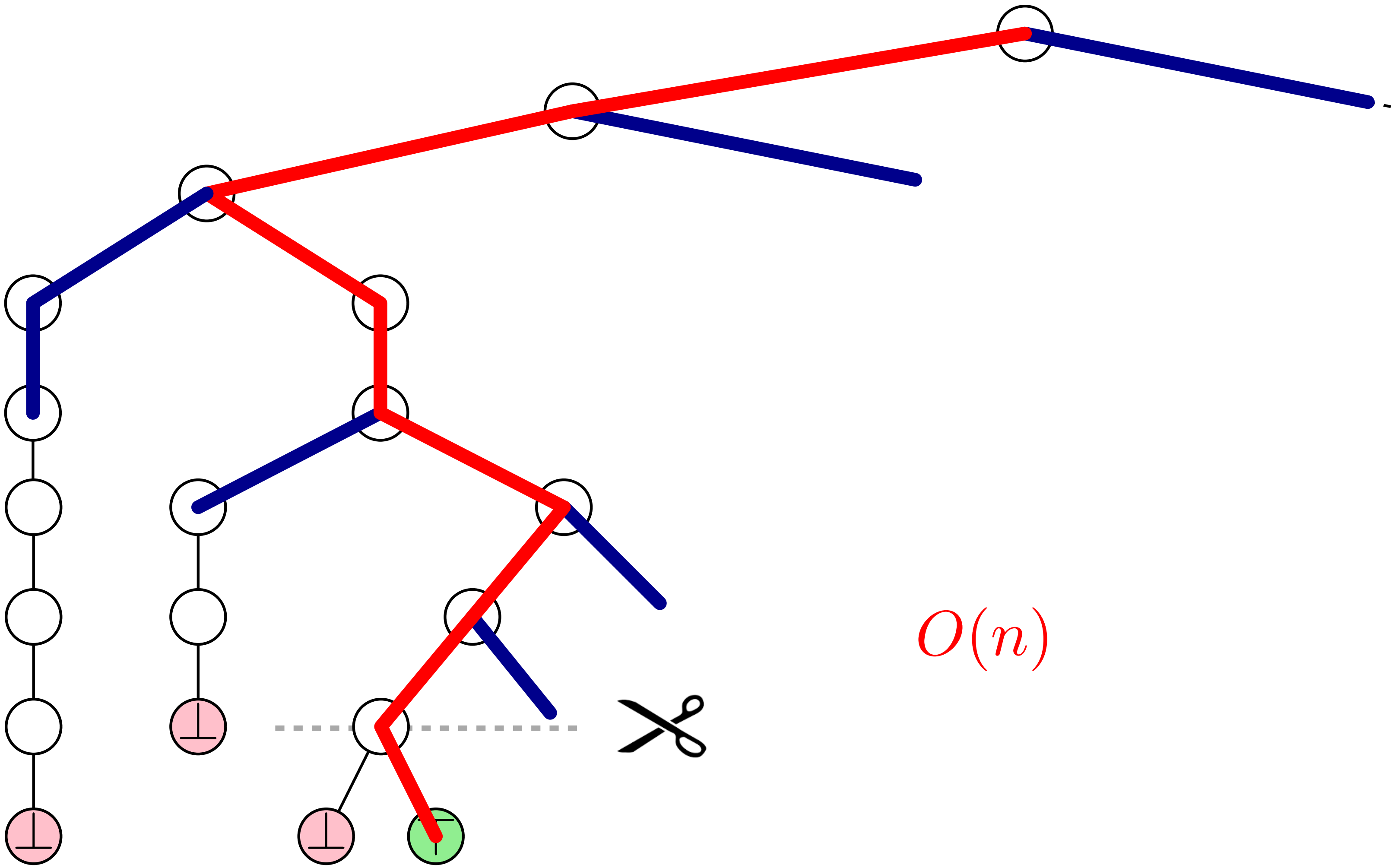
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



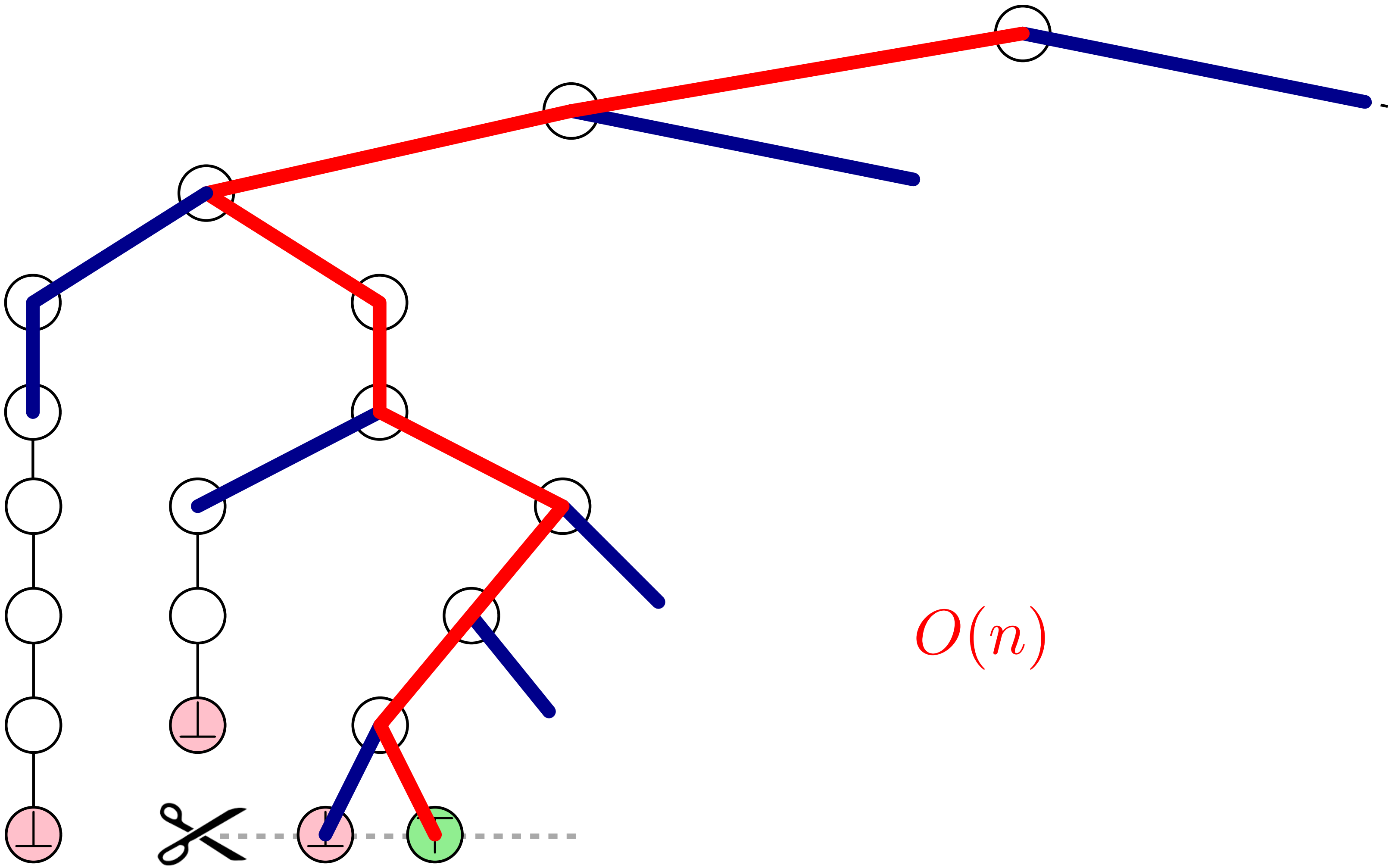
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



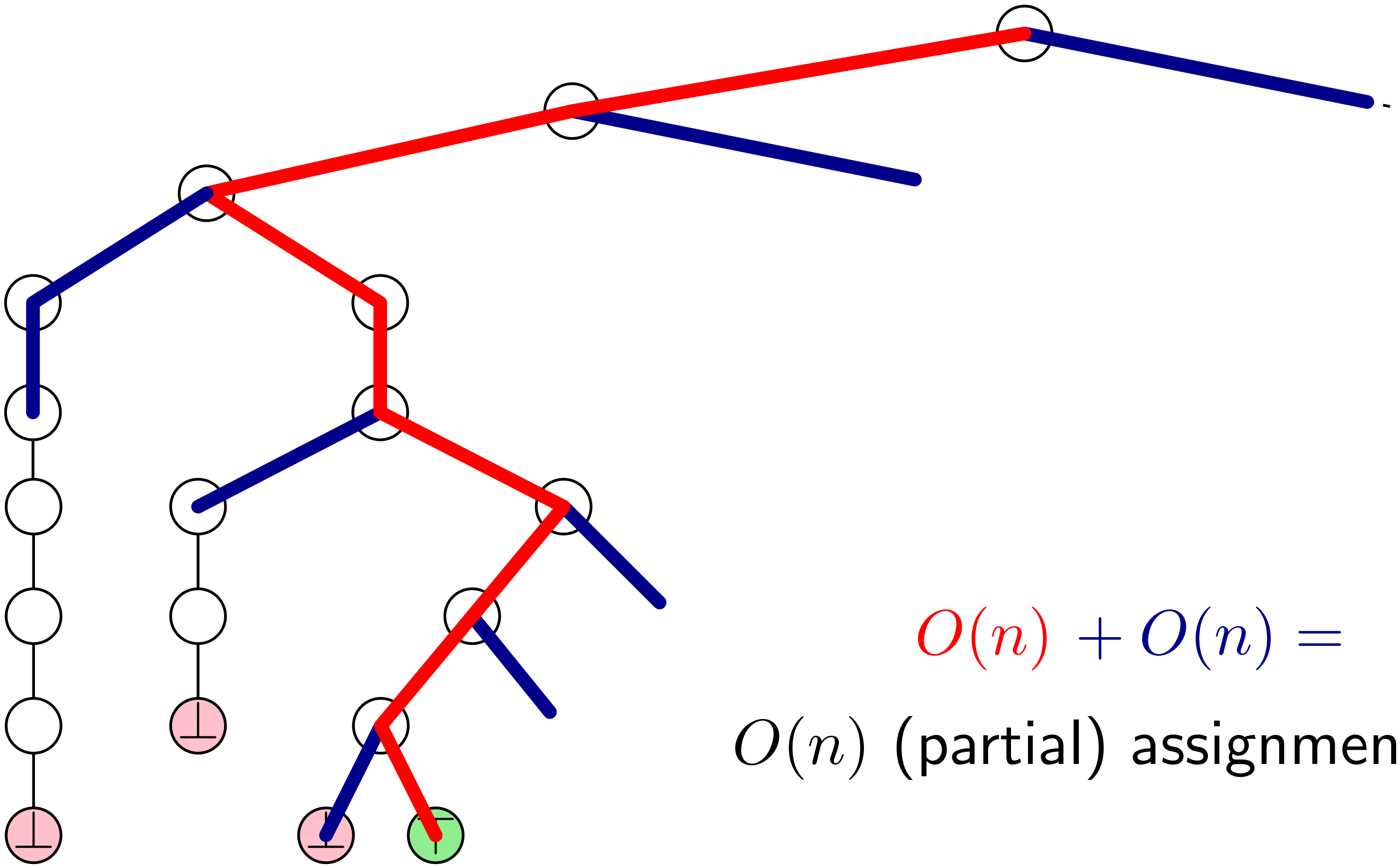
Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



Time Complexity

- Assume that we can find in $O(1)$ time:
  - A contradiction, if one exists.
  - An unassigned implied variable, if any.



$O(n) + O(n) =$

$O(n)$ (partial) assignments