

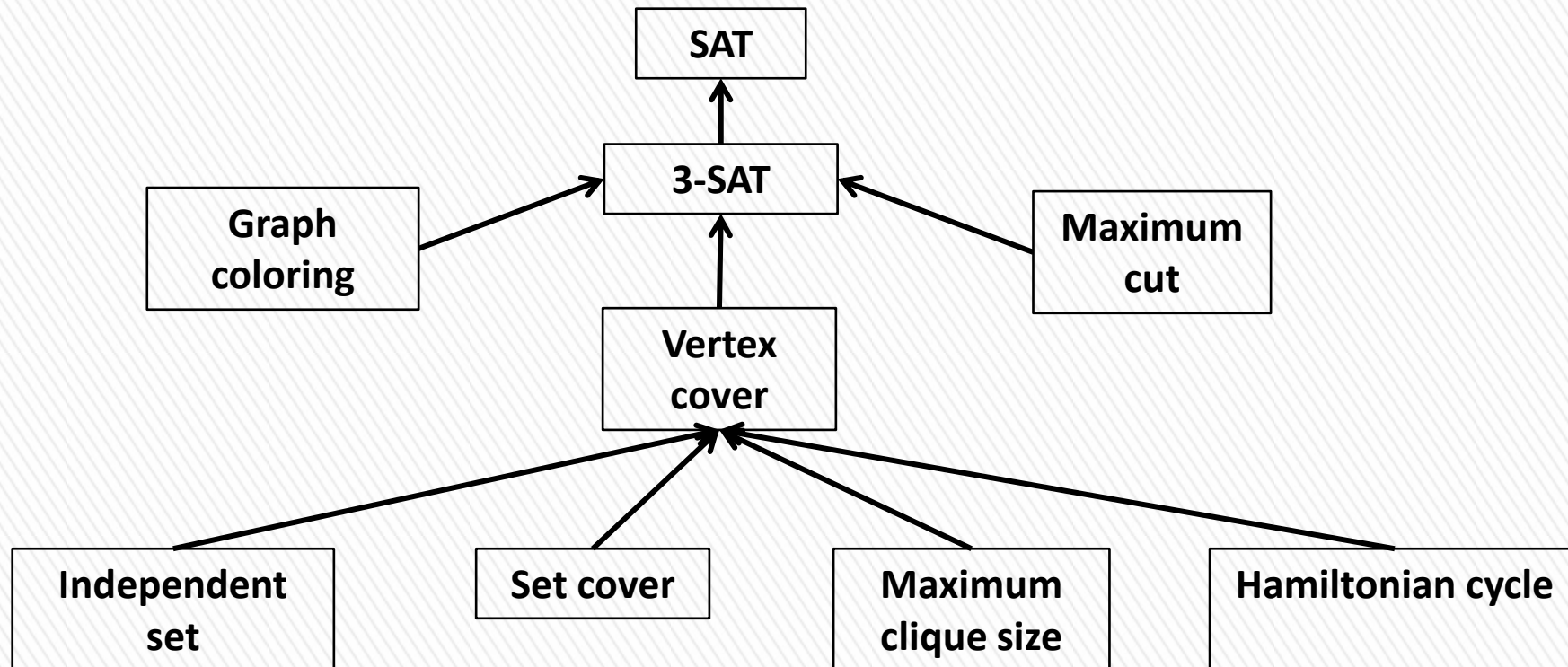


# **Programming assignment:**

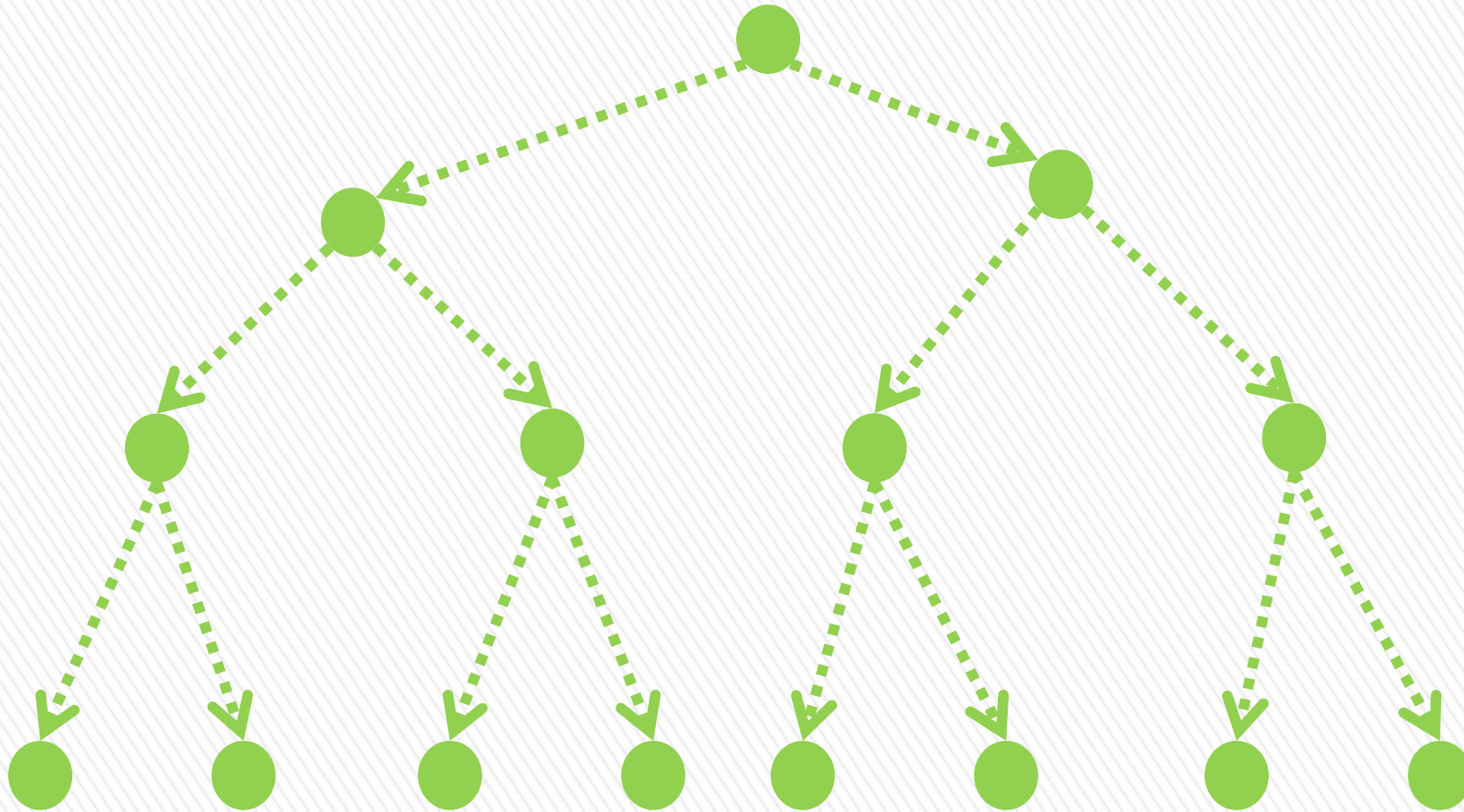
## **A basic SAT solver**

# The Satisfiability Problem

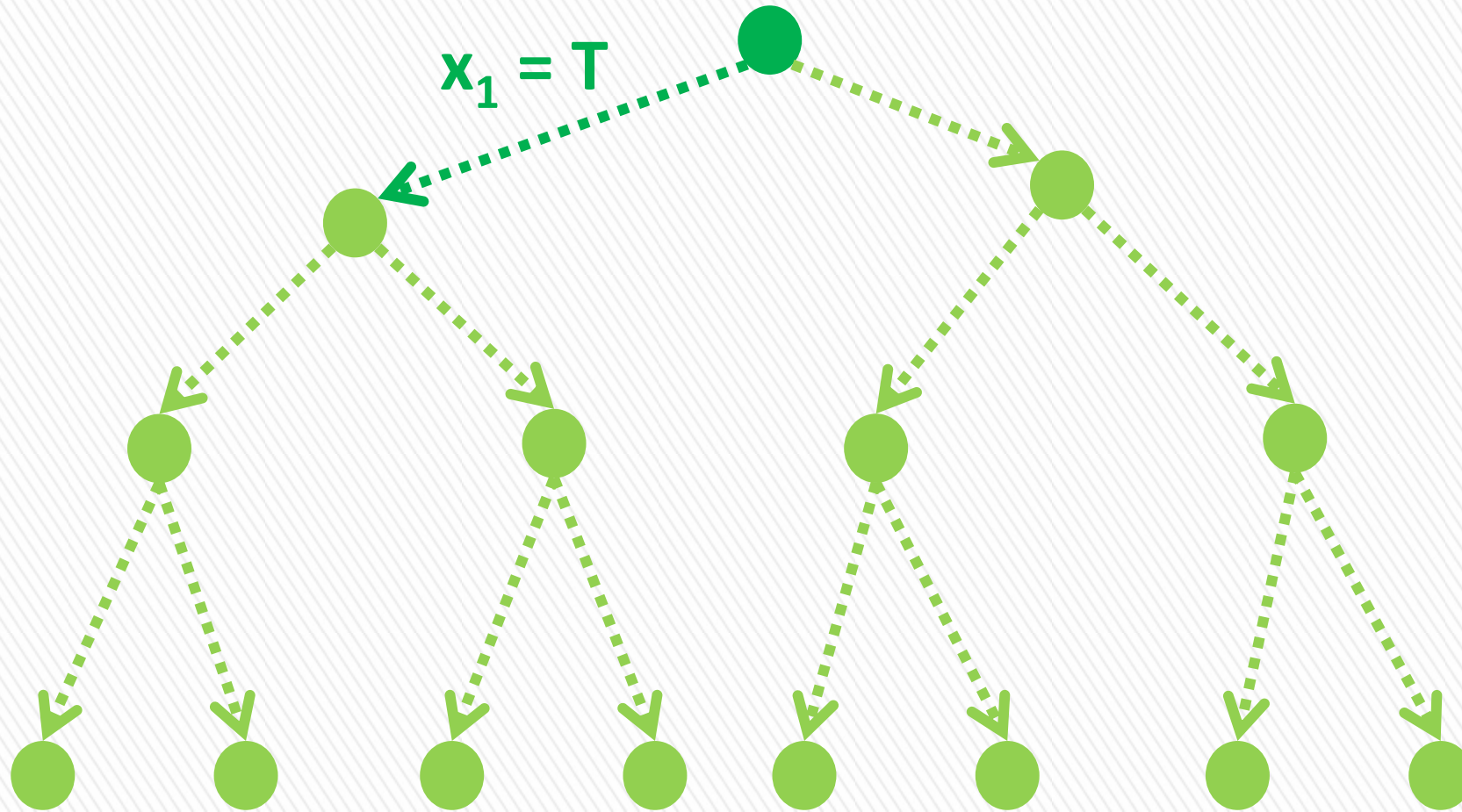
- Given a formula in conjunctive normal form (CNF), determining if there exists an assignment that satisfies the given Boolean formula.



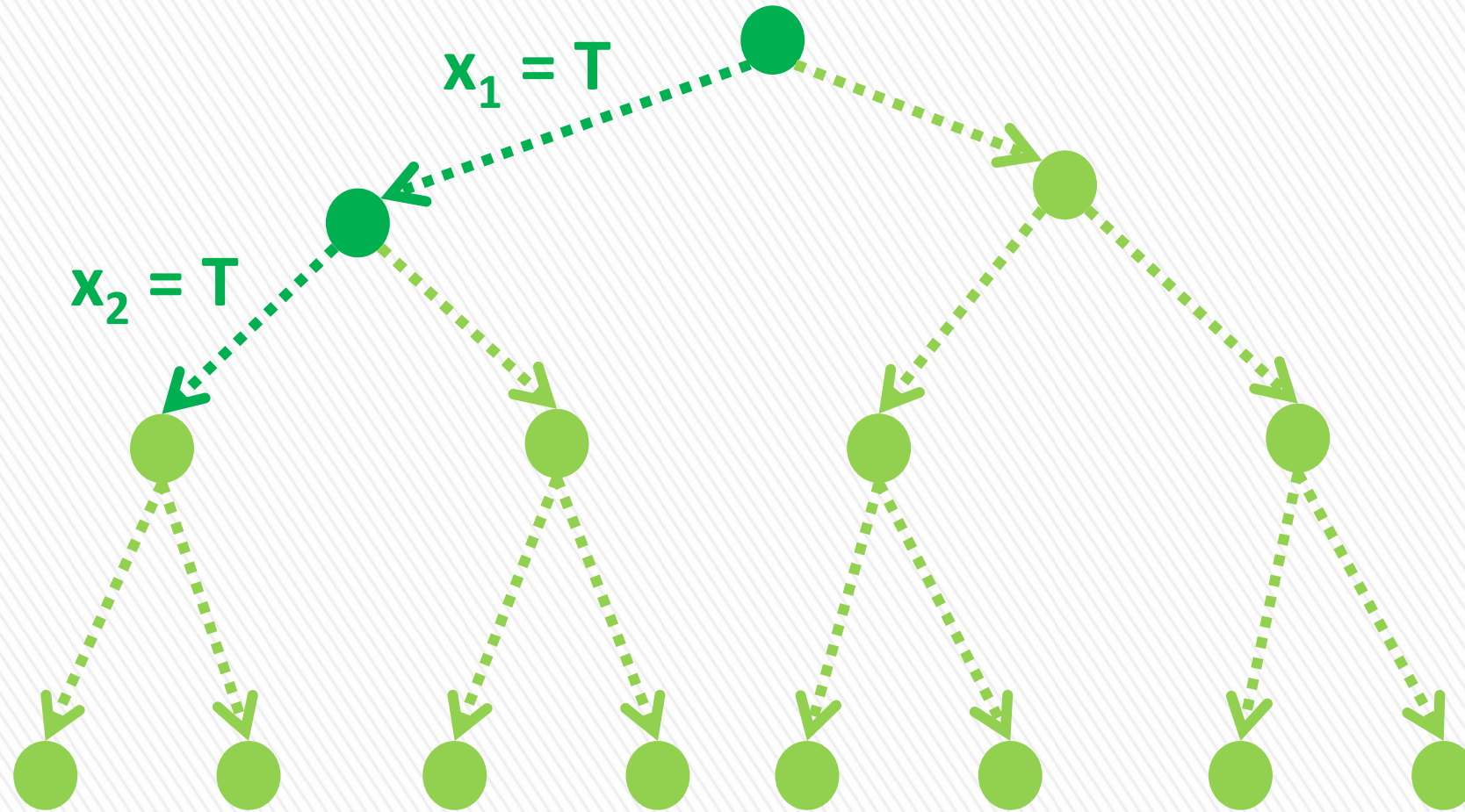
# Basic Backtracking Search



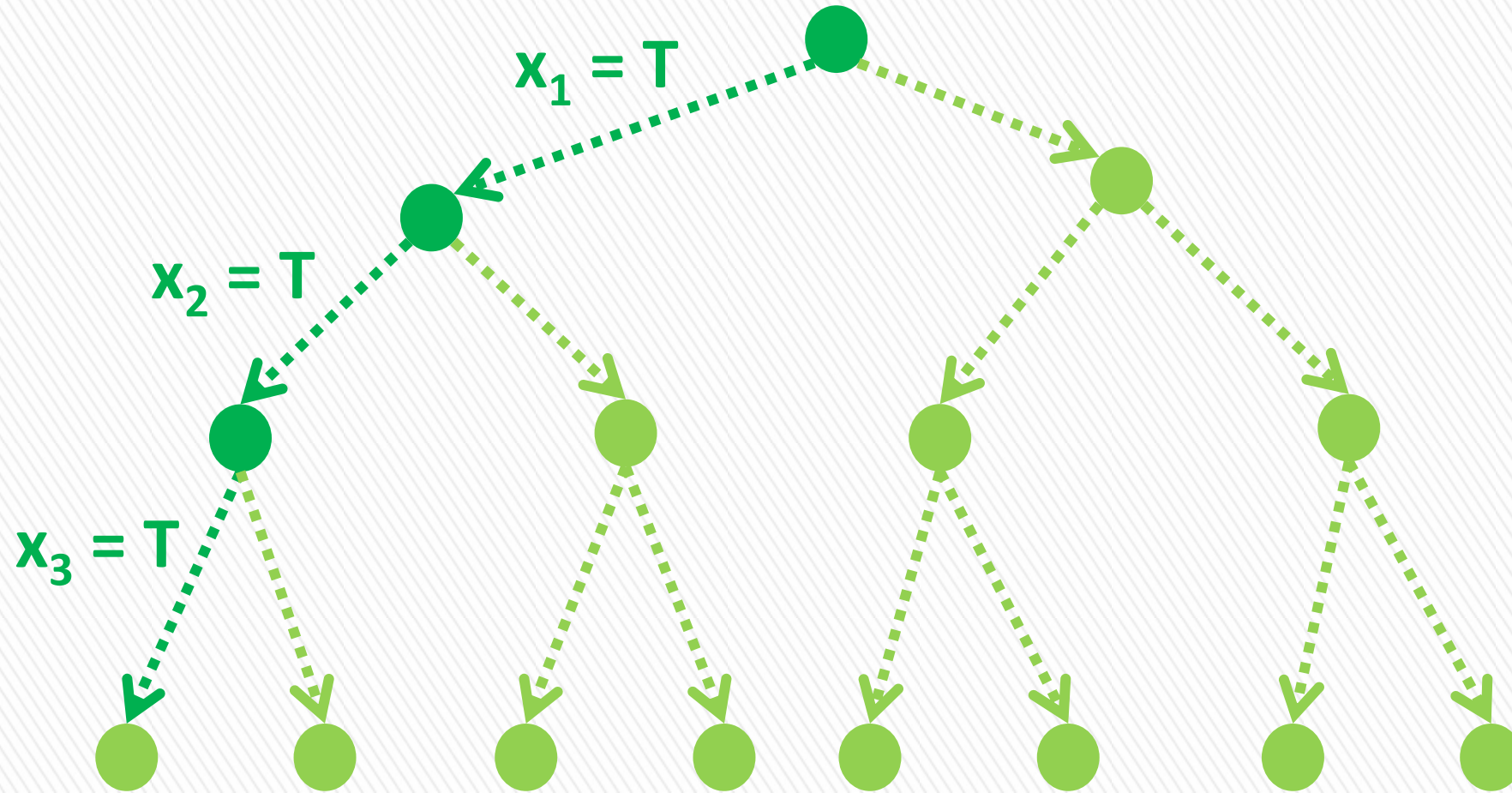
# Basic Backtracking Search



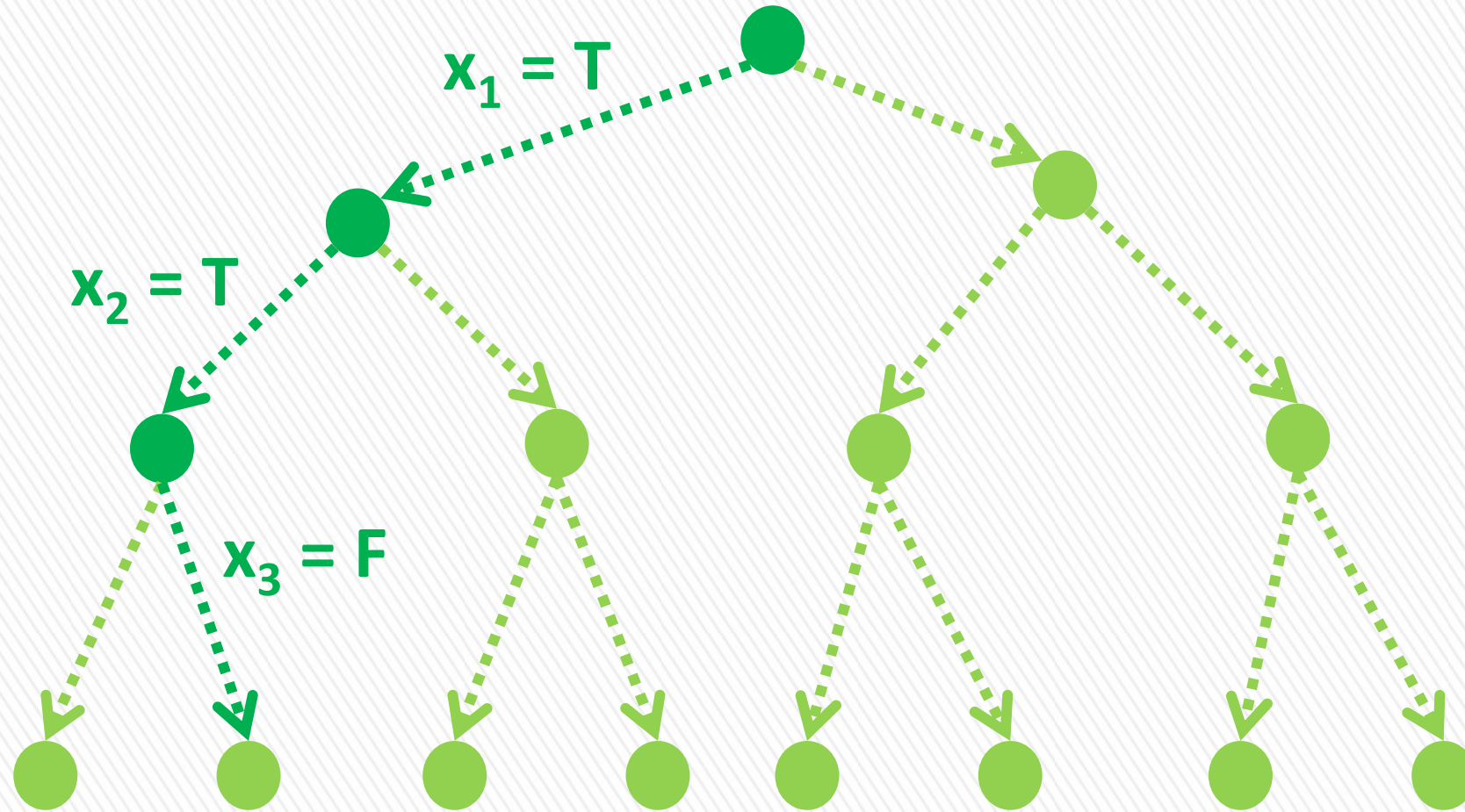
# Basic Backtracking Search



# Basic Backtracking Search



# Basic Backtracking Search

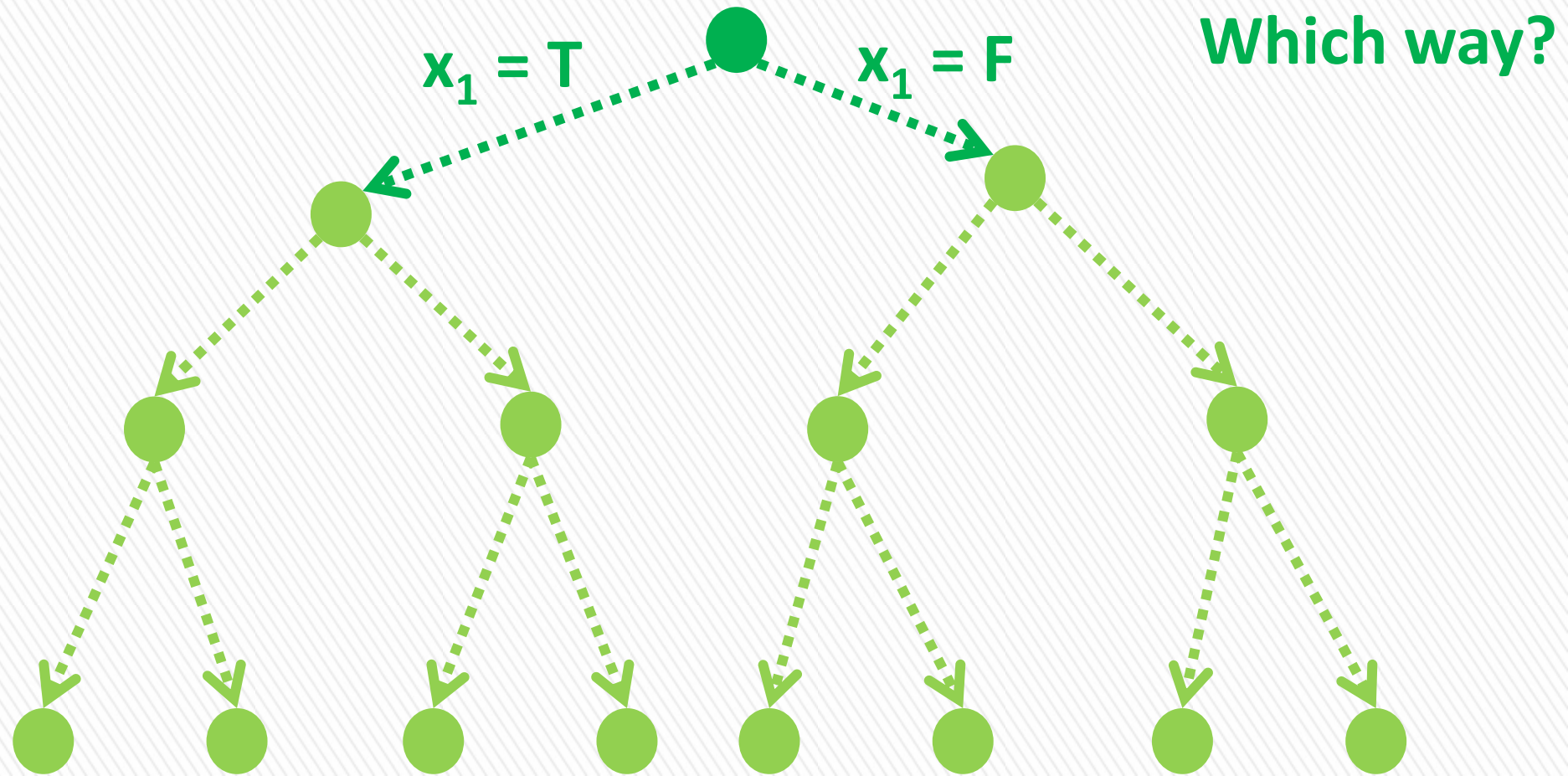




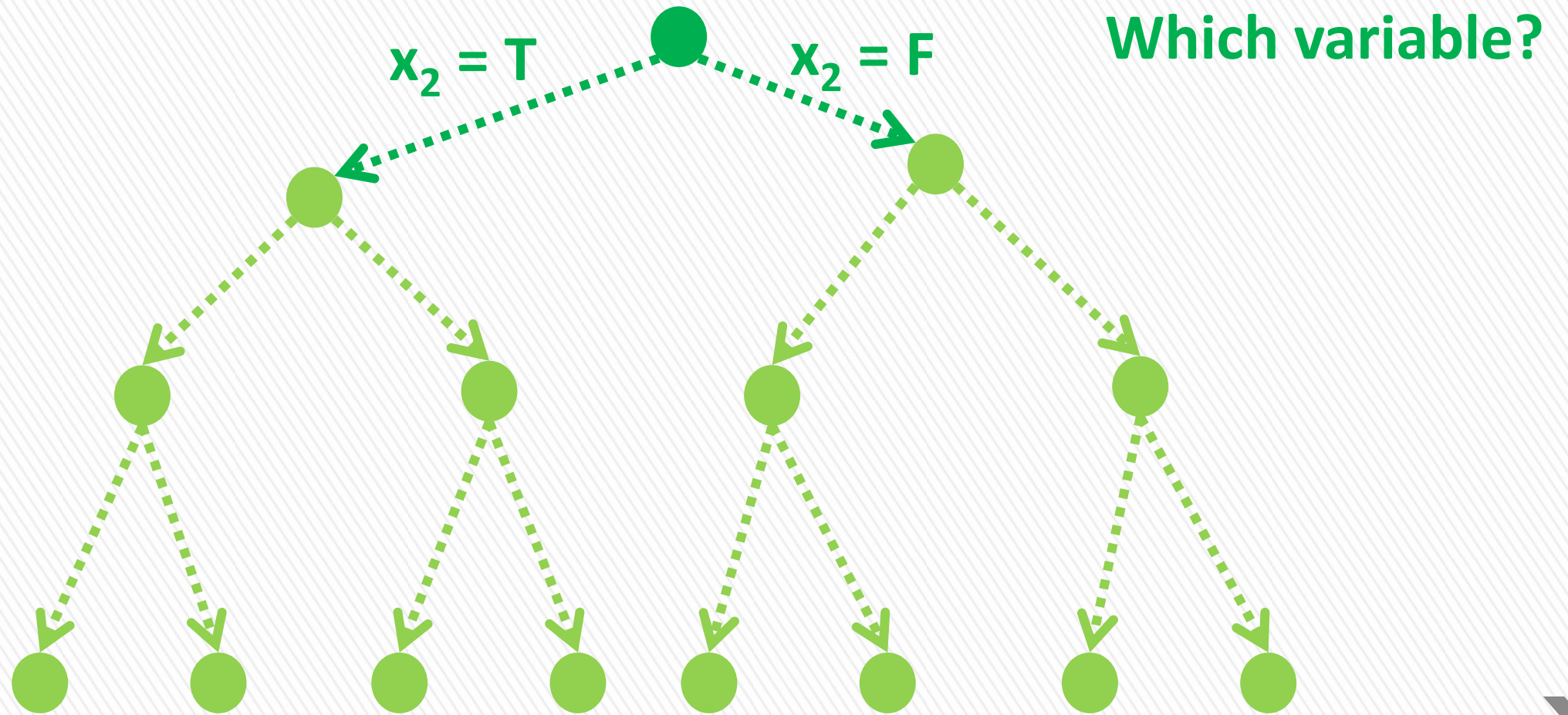




# Basic Backtracking Search



# Basic Backtracking Search



# DPLL Algorithm

## ■ Unit propagation

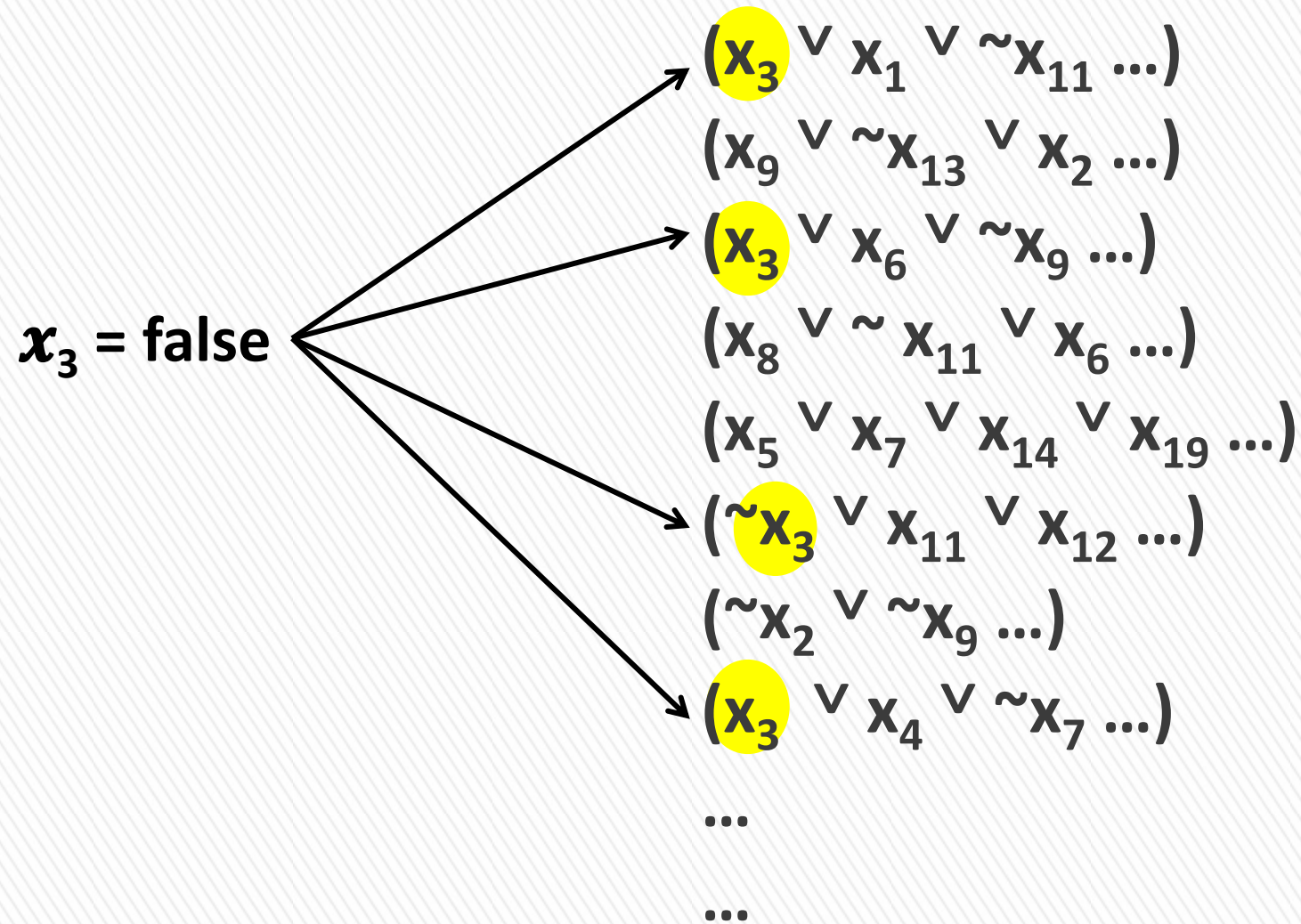
- $x_1 = \text{false}$
- $x_2 = \text{false}$
- $(x_1 \vee x_2 \vee x_3)$
- $x_3 \rightarrow \text{true}$

## ■ Pure literal elimination

- $(x_1 \vee x_3) (x_2 \vee x_3) (x_2 \vee \sim x_4) (\sim x_1 \vee \sim x_2 \vee x_4)$
- $x_3 \rightarrow \text{true}$

■ *Skip unnecessary guesses!*

# Two Watched Literals



# Two Watched Literals


$x_6 = \text{false}$   
 $x_1 = \text{false}$

$\vdots \quad \vdots$   
 $\downarrow \quad \downarrow$   
 $(x_1 \vee x_2 \vee x_3 \vee \sim x_4 \vee \sim x_5 \vee x_6 \dots)$

# Two Watched Literals

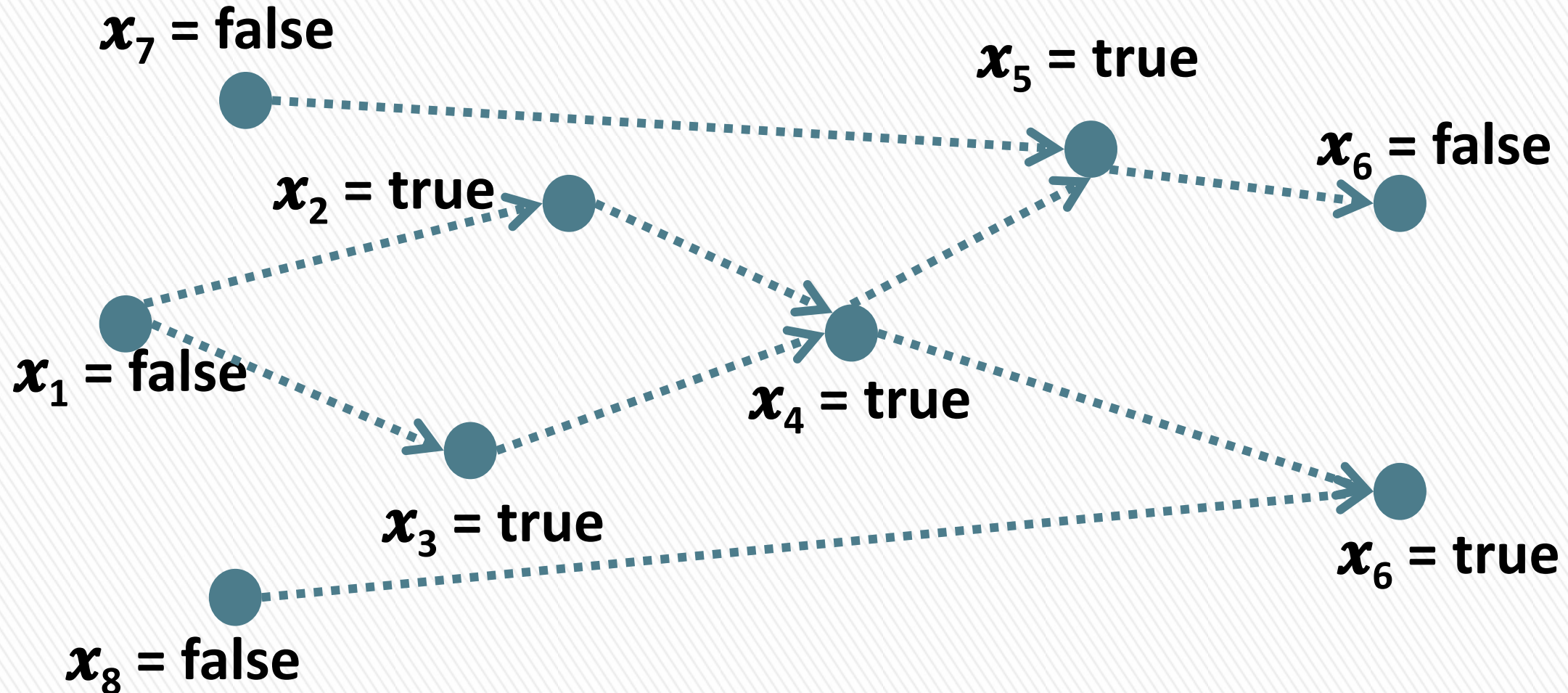
$x_6 = \text{false}$   
 $x_1 = \text{false}$

$(x_1 \vee x_2 \vee x_3 \vee \sim x_4 \vee \sim x_5 \vee x_6 \dots)$



# CDCL Algorithm

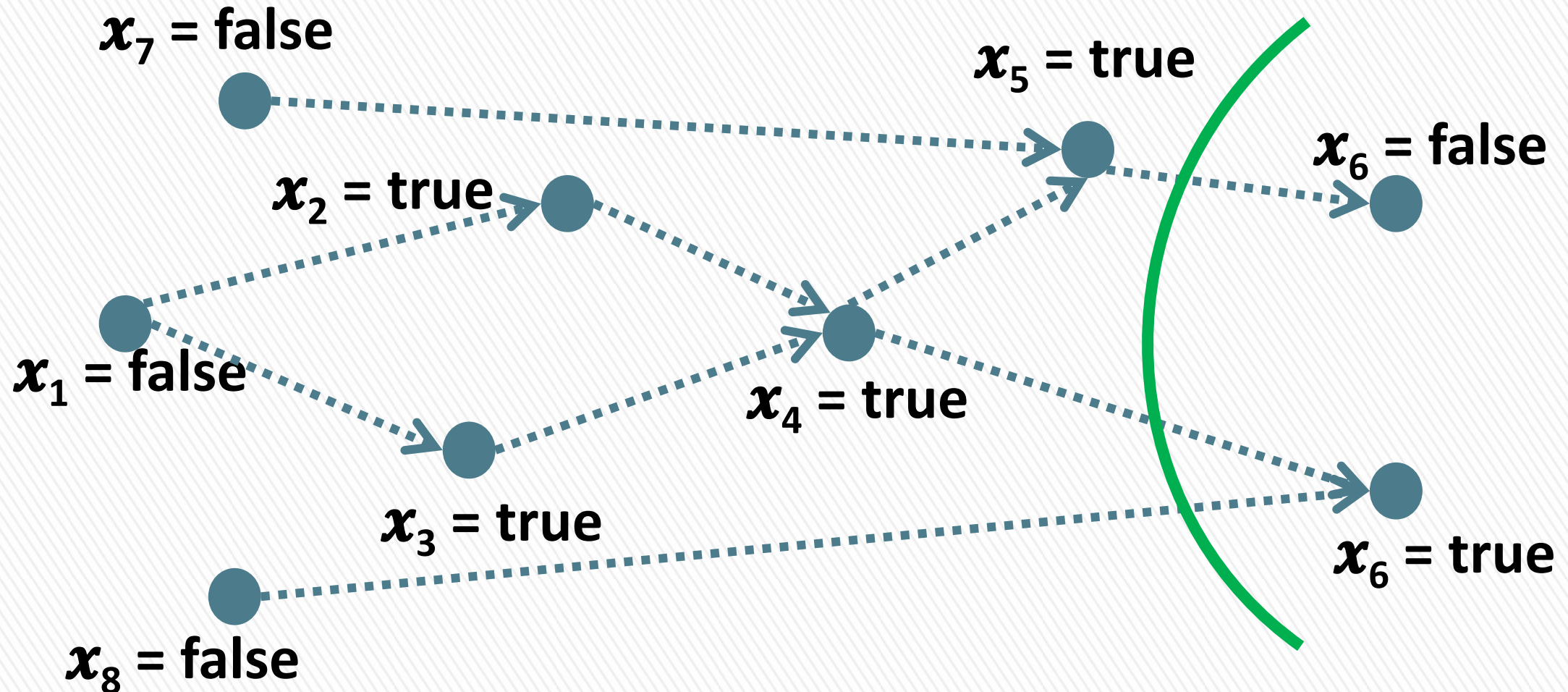
## ■ Conflict-driven clause learning





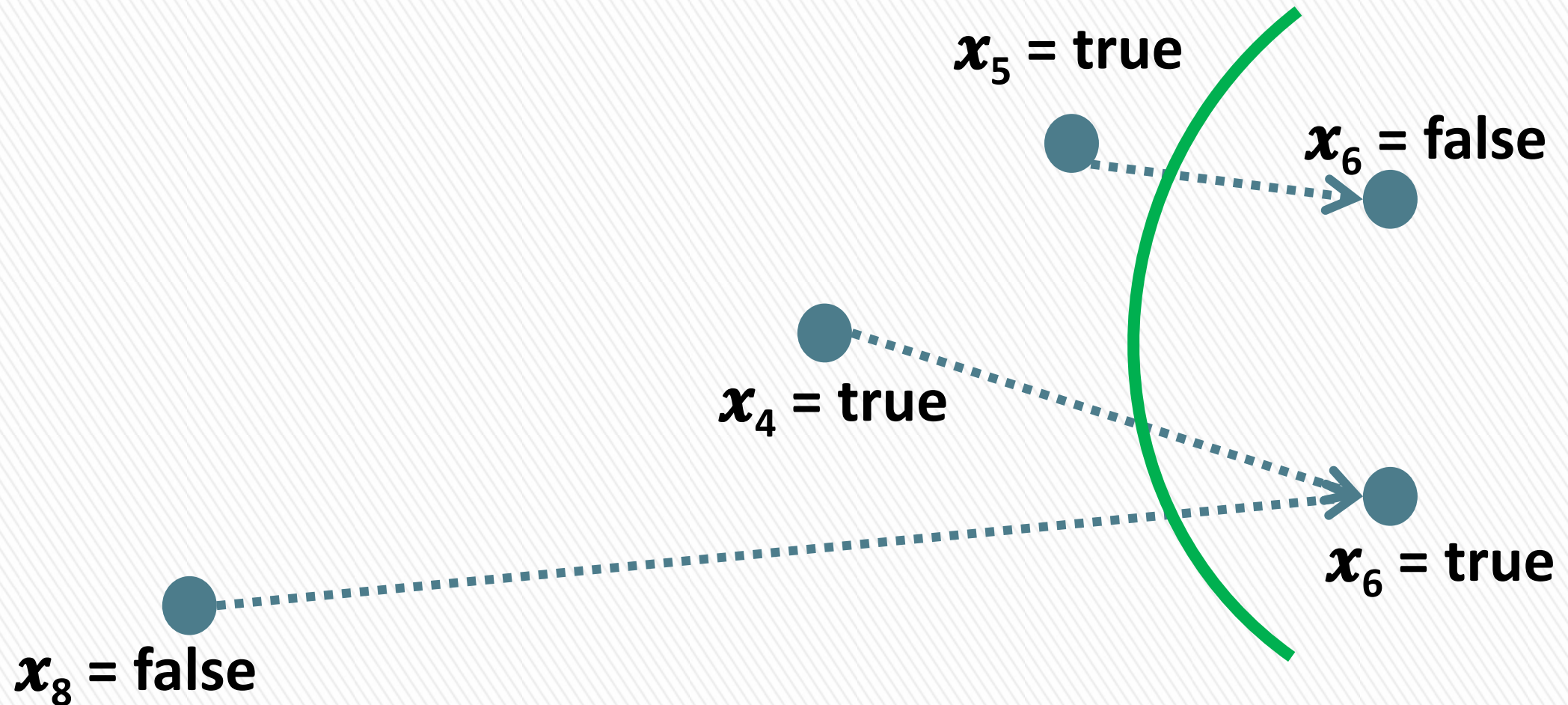
# CDCL Algorithm

## ■ Conflict-driven clause learning



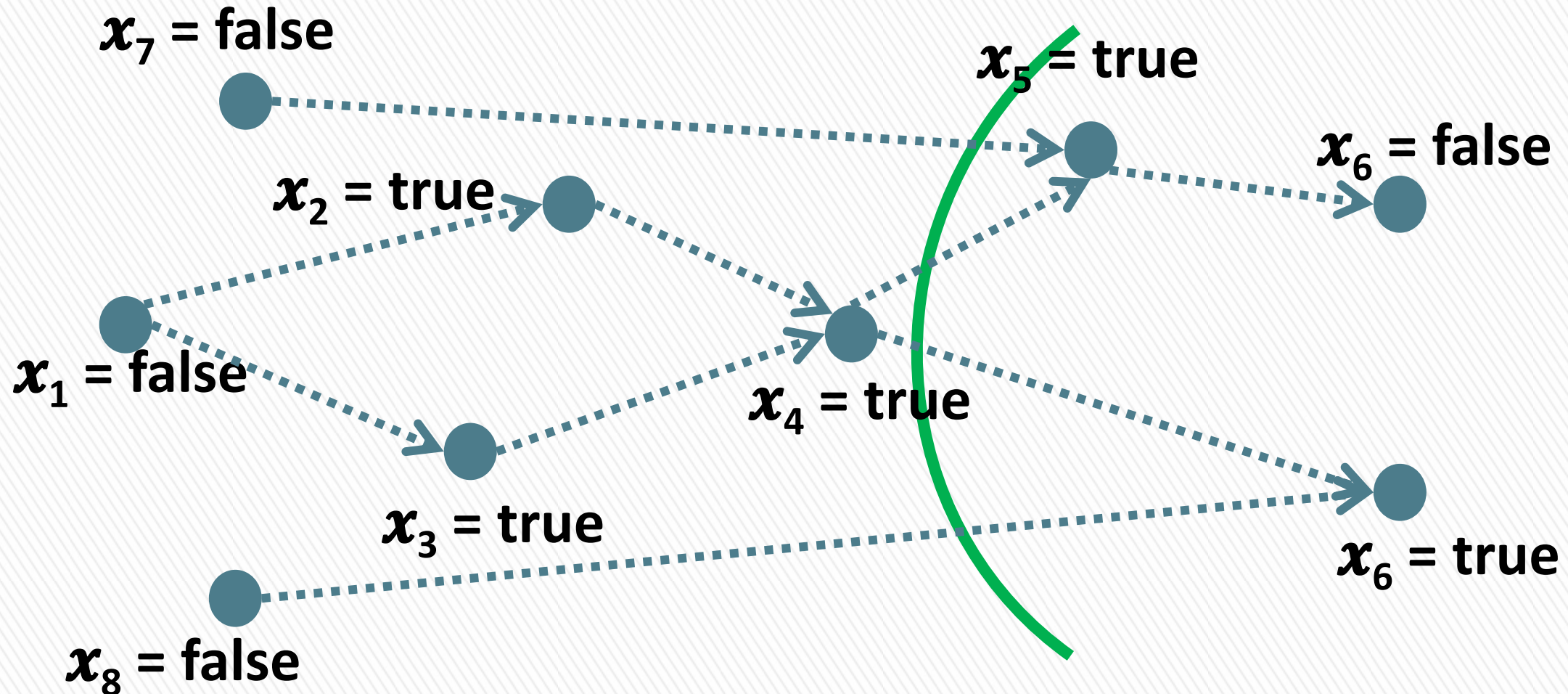
# CDCL Algorithm

## ■ Conflict-driven clause learning



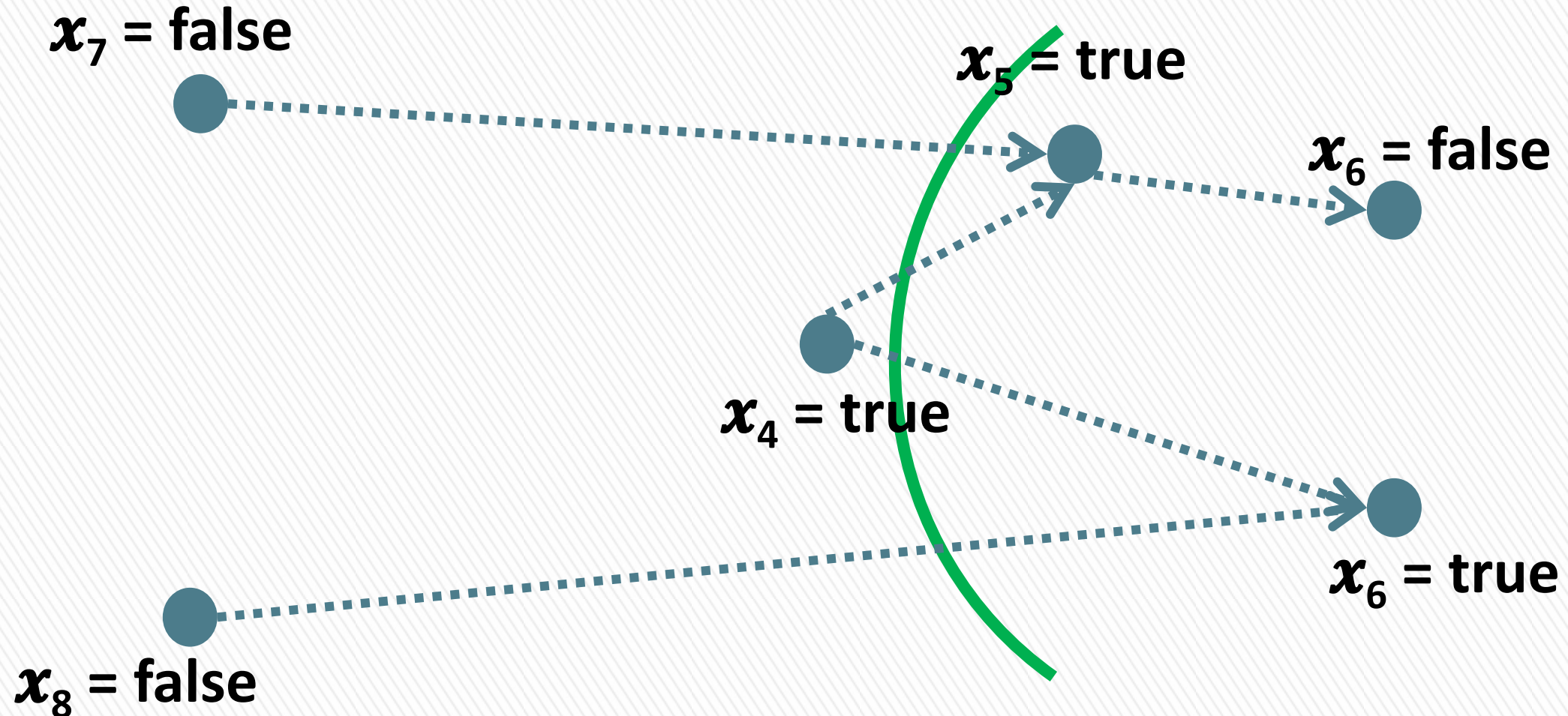
# CDCL Algorithm

## ■ Conflict-driven clause learning



# CDCL Algorithm

## ■ Conflict-driven clause learning



# CDCL Algorithm

## ■ Conflict-driven clause learning

$(x_4 \wedge \sim x_7 \wedge \sim x_8) \rightarrow \text{conflict}$   
                                  ?  
                                   $\leftarrow \text{not conflict}$

*Contrapositive*

$(\sim(x_4 \wedge \sim x_7 \wedge \sim x_8)) \leftarrow \text{not conflict}$   
 $(\sim x_4 \vee x_7 \vee x_8) \leftarrow \text{not conflict}$

# CDCL Algorithm

## ■ Conflict-driven clause learning

## ■ Potential issues

- Clause maintenance
- Clause minimization
- Phase saving

# Input format

```
1 c sat-mutex-lemmas 1 four-bits-ez.lemmas
2 c my attempted simplification of Peterson's 4-bit alg
3 c file created by SAT-TO-DIMACS Wed Jul  8 08:17:39 2015
4 c #56 -> 129
5 c #55 -> 128
6 c #54 -> 127
7 c #53 -> 126
8 c #52 -> 125
9 c #51 -> 124
10 c #50 -> 123
11 c #49 -> 122
12 c #48 -> 121
13 c #47 -> 120
14 c #46 -> 119
```

```
130 c 001A50 -> 3
131 c 001A51 -> 2
132 c 001A52 -> 1
133 p cnf 129 354
134 129 128 127 126 125 124 123 122 121 120 119
    118 117 116 115 114 113 112 111 110 109 108
    107 106 105 104 103 102 101 100 99 98 97 96
    95 94 93 92 91 90 89 88 87 86 85 84 83 82 81
    80 79 78 77 76 75 74 0
135 -71 -129 0
136 17 -129 0
137 42 -69 0
138 -71 -128 0
139 18 -128 0
140 42 -68 0
141 -71 -127 0
142 19 -127 0
143 42 -67 0
144 -71 -126 0
145 21 -126 0
146 42 -66 0
147 -71 -125 0
148 22 -125 0
```



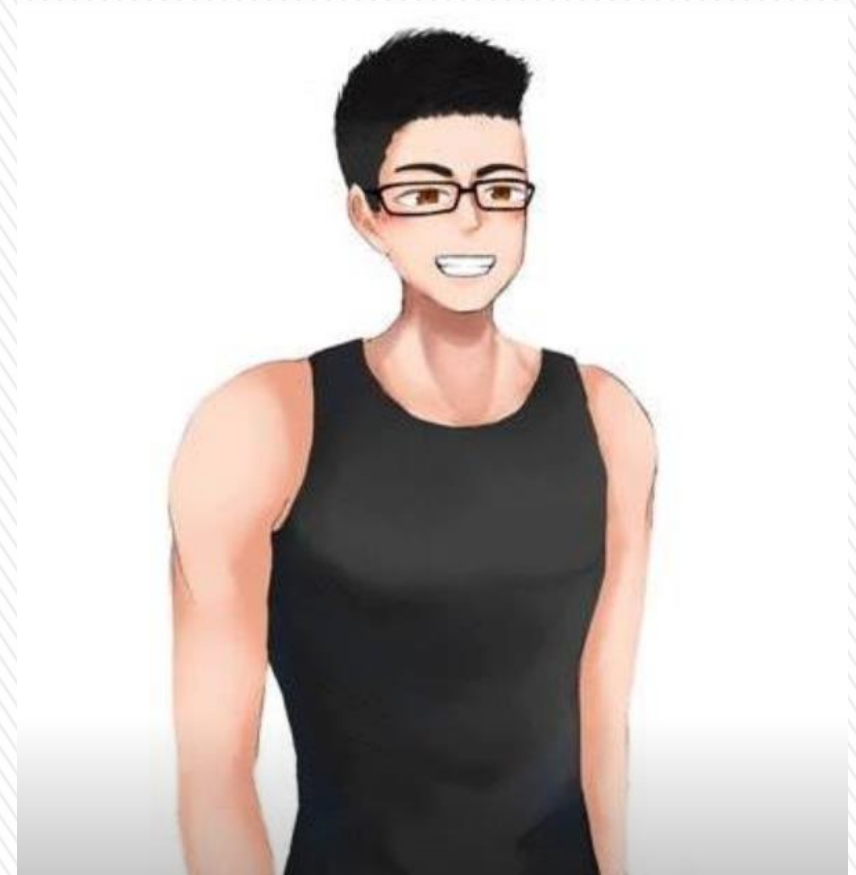
# Environment

## ■ Sky Online Judge

- <https://pc2.tfcis.org/dev/index.php>
- Create an account with your **student ID**.
- C/C++ language
- Problem ID: 100, 101 and 102

### 題目列表

NAME	
100	SAT-easy
101	SAT-normal
102	SAT-hard



Many thanks to 日月卦長!

# Evaluation

## ■ Correctness (20%)

- Only *Accepted*, *Memory Limit Exceed*, and *Time Limit Exceed* are allowed in all the 26 released cases.

## ■ Performance (80%)

- 4% for each *Accepted* case.

## ■ Reference

- The Art of Computer Programming / Donald E. Knuth - satisfiability

- Please upload your source code to ilms before 11:59 p.m. on April 22.

*Do not cheat in the programming assignment!!!*