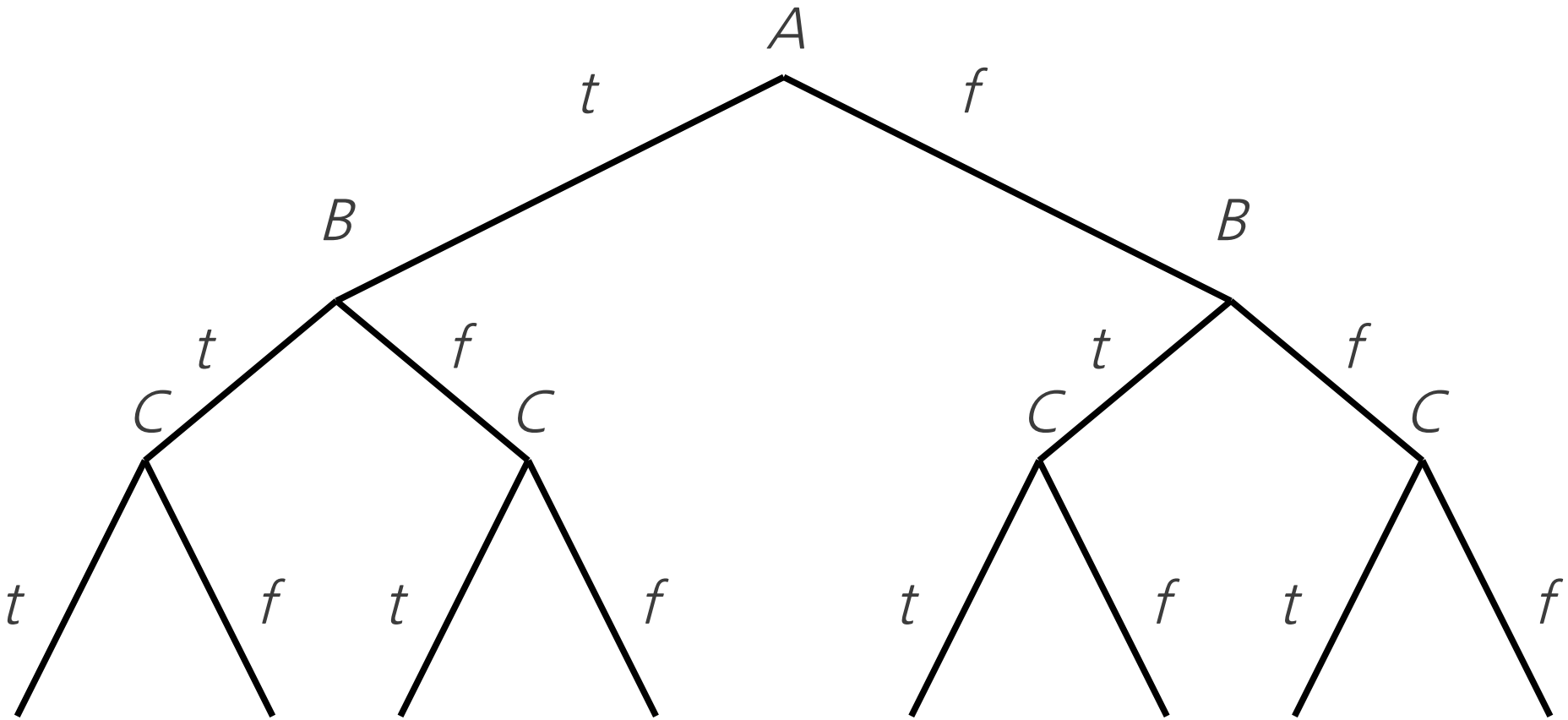
SAT Algorithms
DPLL and CDCL

서울대학교 프로그래밍 연구실
조성근

$$(\neg A \vee B) \wedge (\neg B \vee C)$$

이 식을 참으로 만드는 A, B, C가 있나?

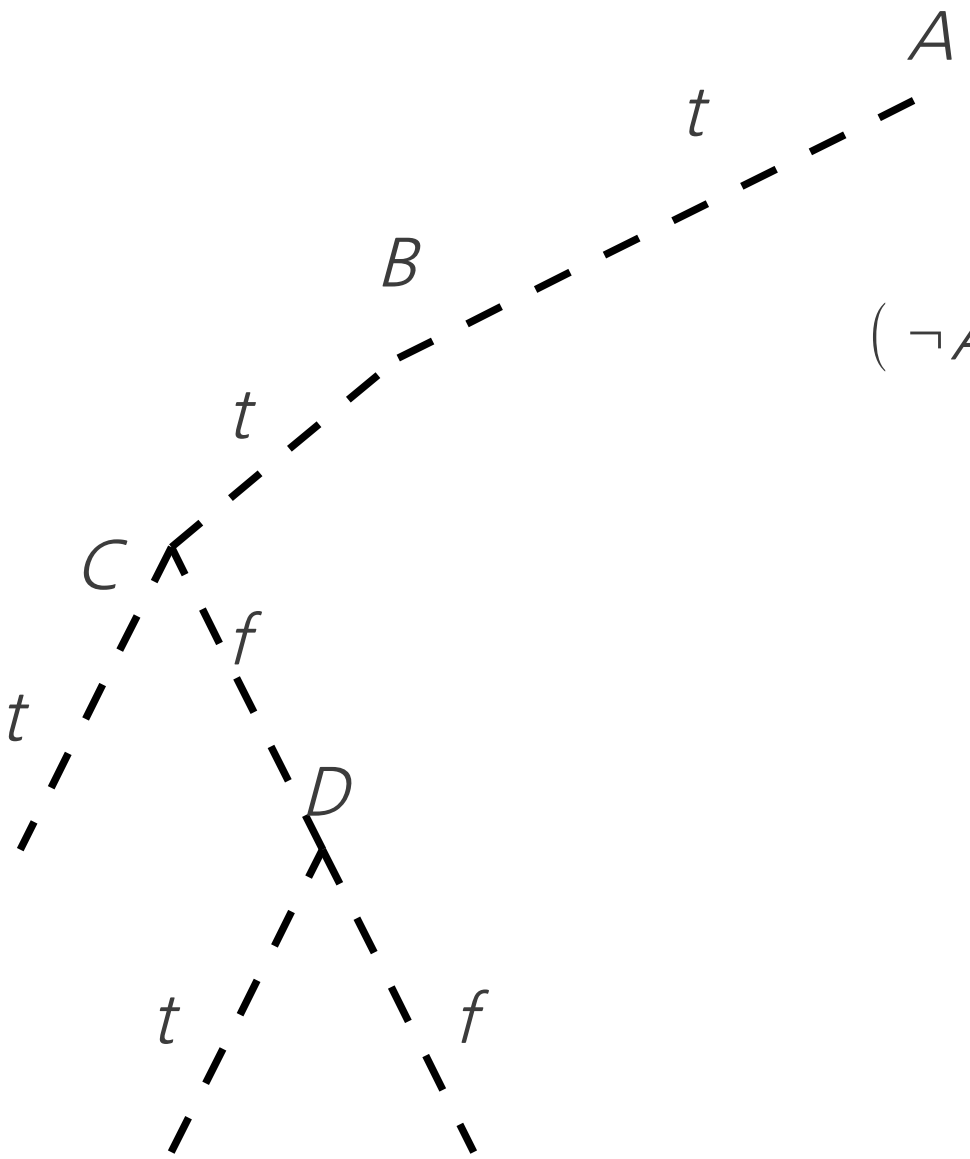
$$(\neg A \vee B) \wedge (\neg B \vee C)$$



DPLL 알고리즘

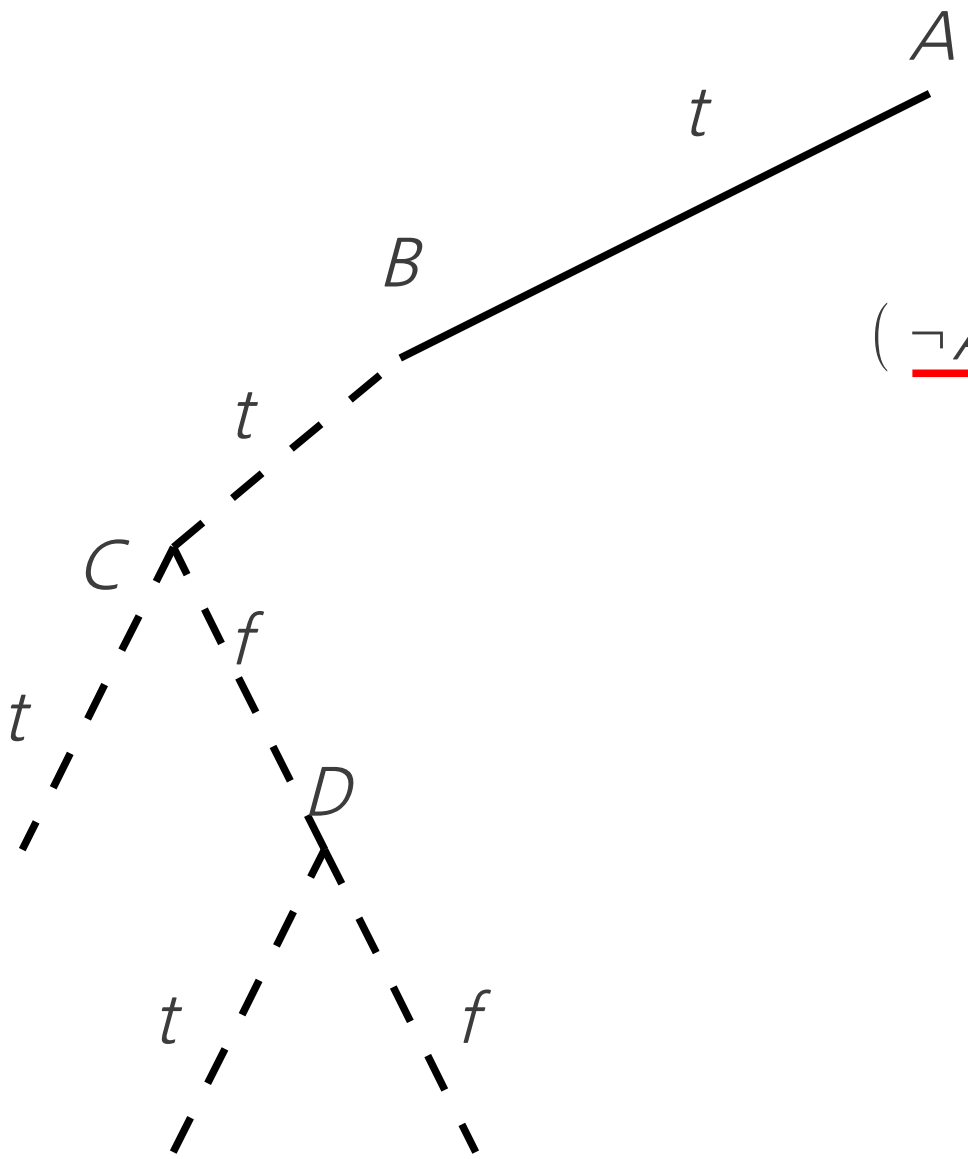
```
(정의 (dpll 제약조건)
  (조건
    [(다풀었니? 제약조건)                (공집합)]
    [(충돌이있니? 제약조건)              (못풀어)]
    [(다르니? (dpll (x를참으로  제약조건)) (못풀어))
                                  (추가 (결과) (x=참))]
    [(다르니? (dpll (x를거짓으로  제약조건)) (못풀어))
                                  (추가 (결과) (x=거짓))]
    [(참) (못풀어)]
  )
)
```

DPLL



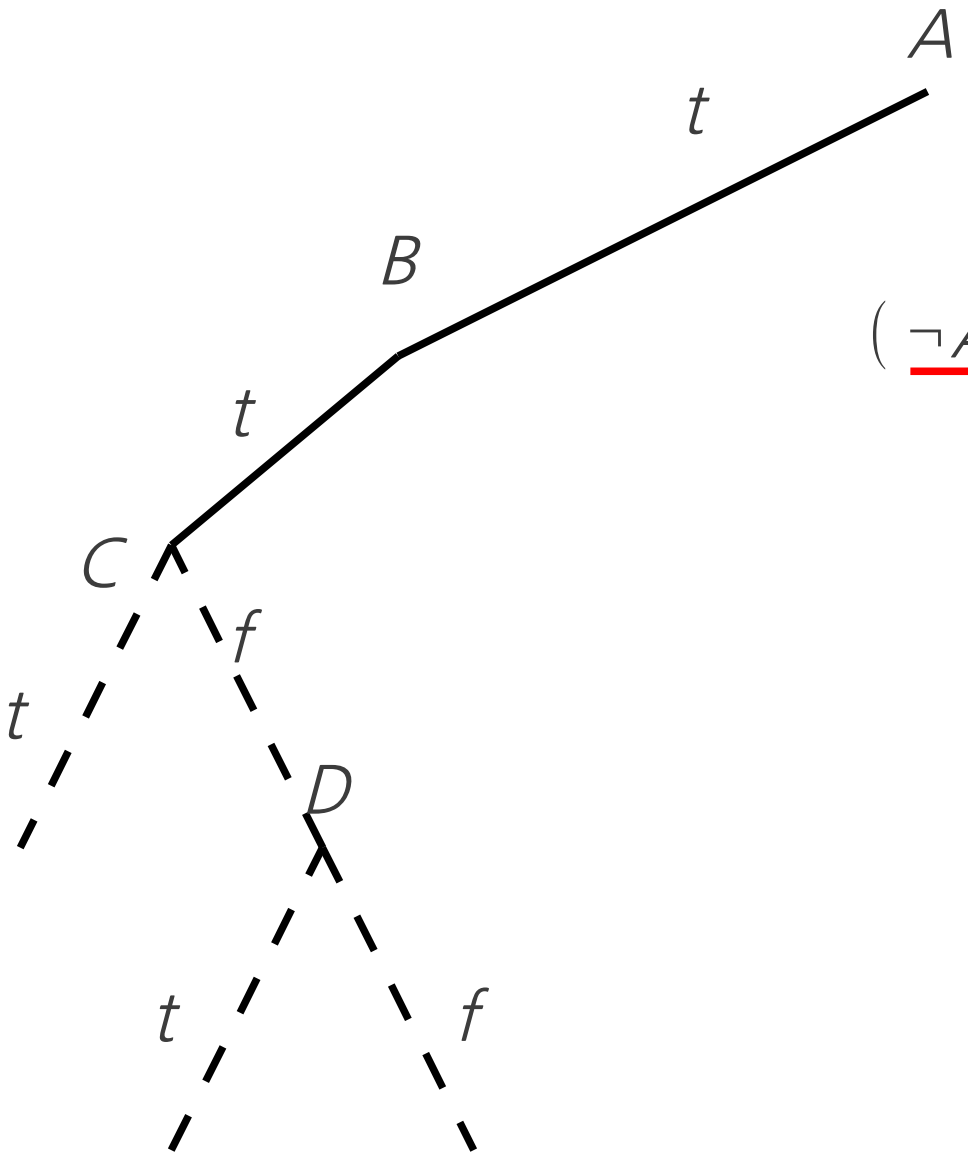
$$(\neg A \vee B) \wedge (\neg B \vee \neg C) \wedge (C \vee \neg D)$$

DPLL



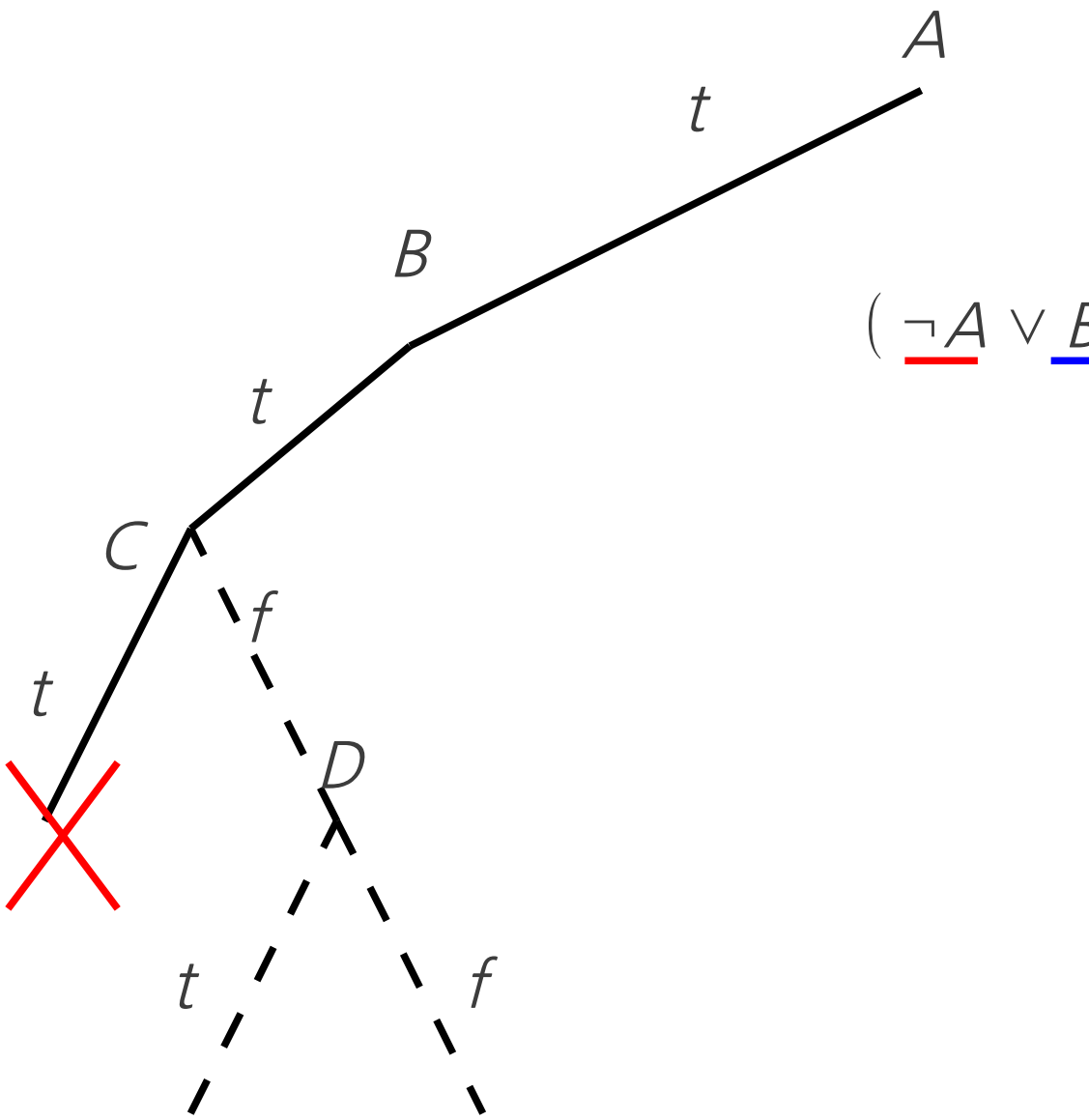
$$(\underline{\neg A} \vee B) \wedge (\neg B \vee \neg C) \wedge (C \vee \neg D)$$

DPLL



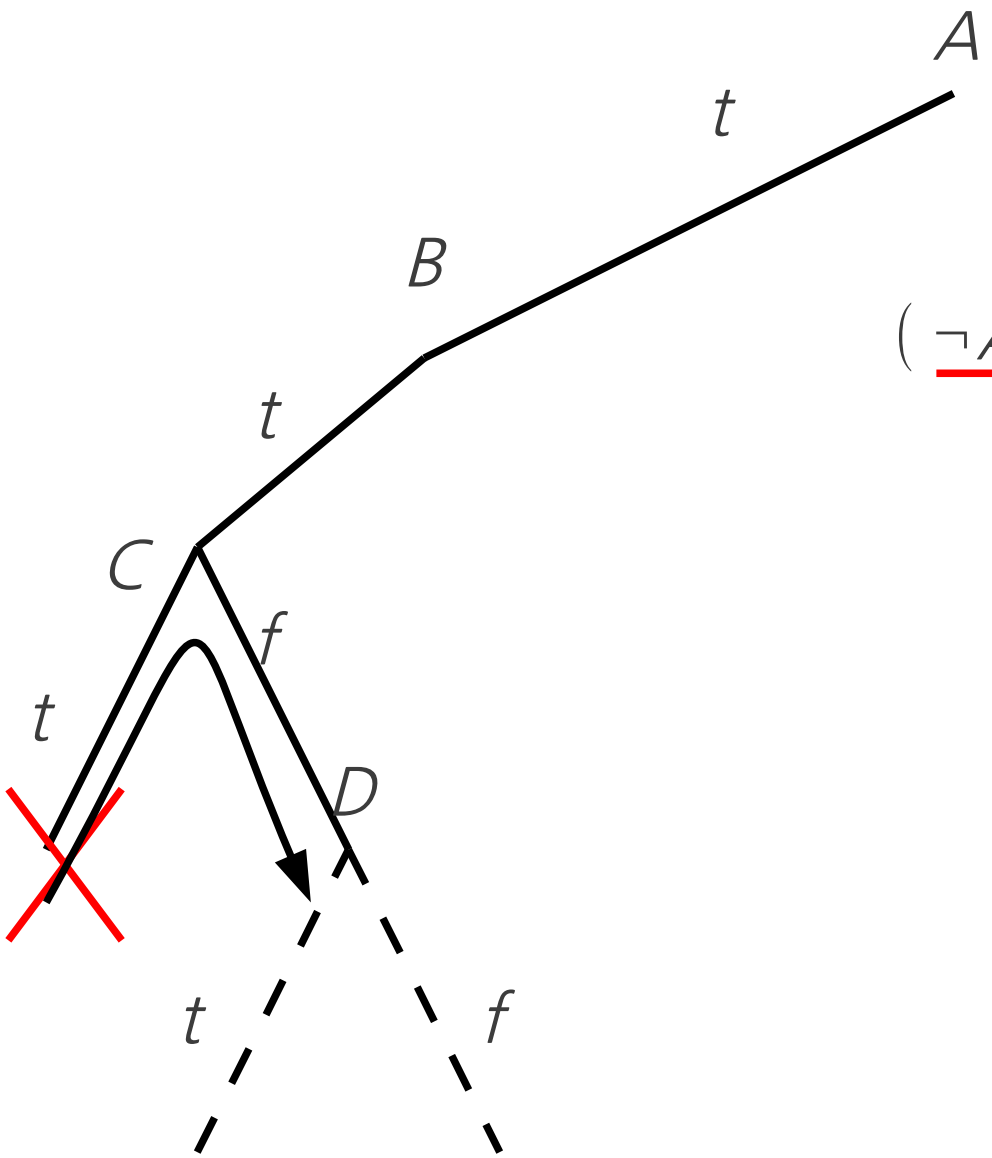
$$(\neg A \vee B) \wedge (\neg B \vee \neg C) \wedge (C \vee \neg D)$$

DPLL



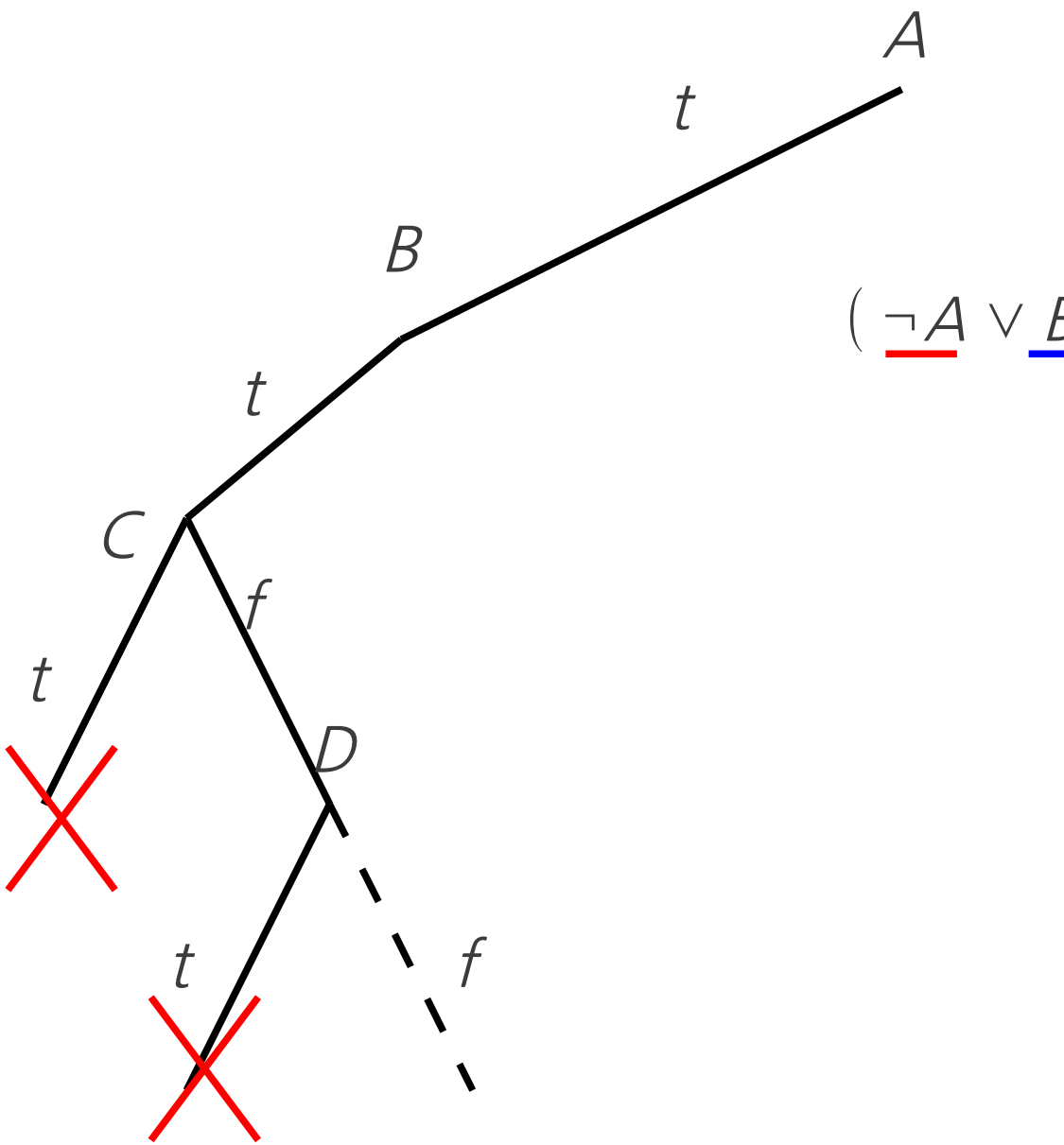
$$(\underline{\neg A} \vee \underline{B}) \wedge (\underline{\neg B} \vee \underline{\neg C}) \wedge (\underline{C} \vee \neg D)$$

DPLL



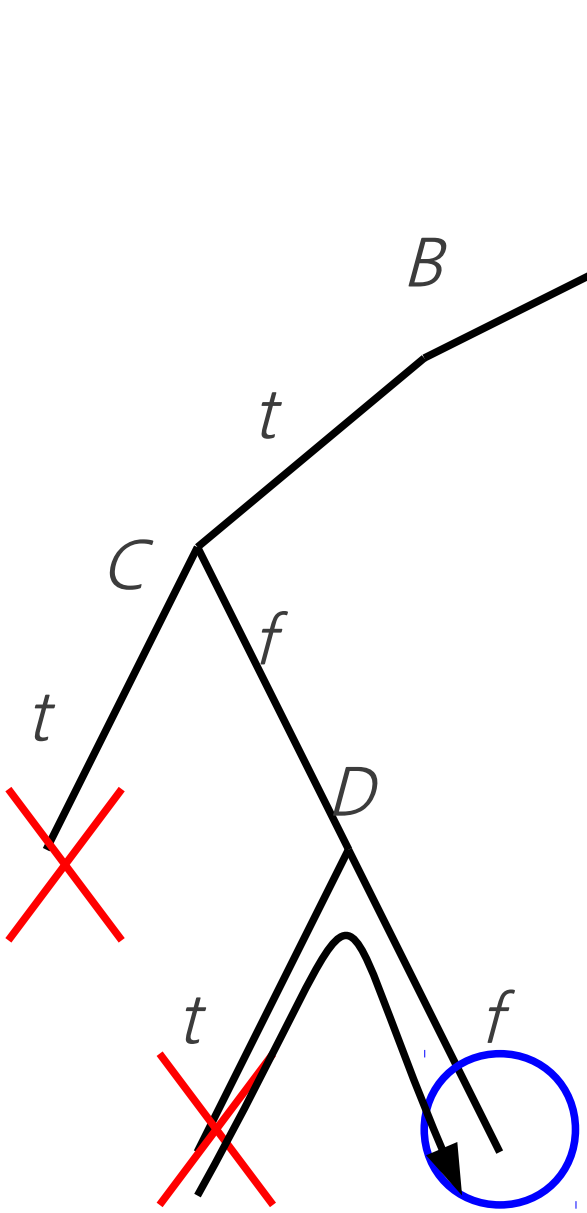
$$(\underline{\neg A} \vee \underline{B}) \wedge (\underline{\neg B} \vee \underline{\neg C}) \wedge (\underline{C} \vee \underline{\neg D})$$

DPLL



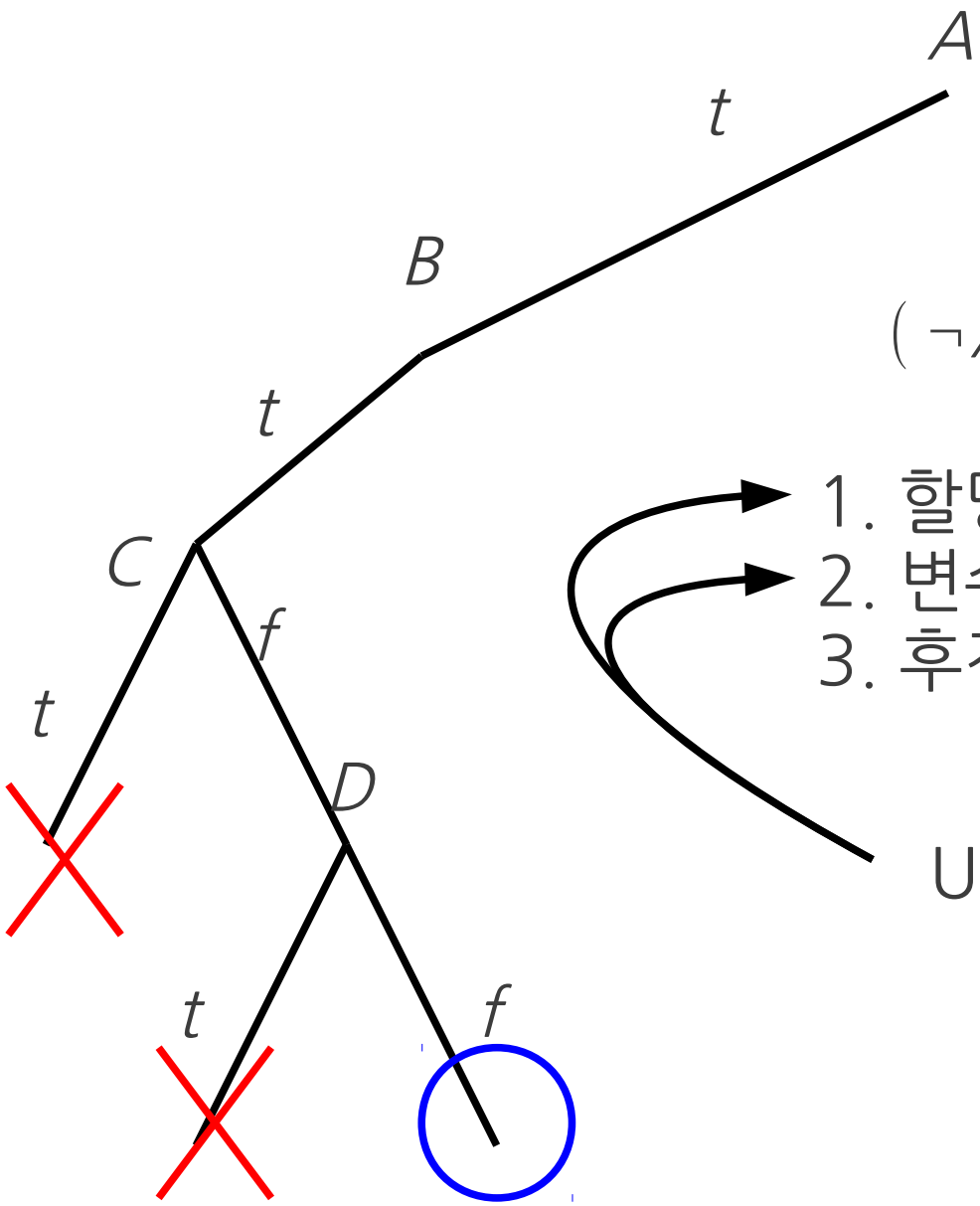
$$(\underline{\neg A} \vee \underline{B}) \wedge (\underline{\neg B} \vee \underline{\neg C}) \wedge (\underline{C} \vee \underline{\neg D})$$

DPLL



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

DPLL



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

1. 할당하는 변수의 순서
2. 변수에 좋은 값 주기
3. 후진(backtracking) 잘 하기

Unit Resolution

Unit Resolution

- literal이 하나 밖에 없는 clause가 있을 때
- 예
  - $(\underline{B}) \wedge (\neg B \vee \neg C) \wedge (C \wedge \neg D)$
  - $(\neg A \vee B) \wedge (\underline{\neg C}) \wedge (C \wedge \neg D)$
- 그 literal이 참이 되도록 값을 할당한다.

CDCL

- **C**onflict-**D**riven **C**lause **L**earning
- DPLL로부터 몇가지 기능이 추가된 버전
  - 충돌로부터 새로운 clause를 배움
  - 더 효율적인 데이터 구조(lazy data structure)
  - 주기적으로 처음부터 다시 시작
  - 새롭게 배운 clause가 너무 많아지면 조금 지우기

CDCL

- **C**onflict-**D**riven **C**lause **L**earning
- DPLL로부터 몇가지 기능이 추가된 버전
  - **충돌로부터 새로운 clause를 배움**
  - 더 효율적인 데이터 구조(lazy data structure)
  - 주기적으로 처음부터 다시 시작
  - 새롭게 배운 clause가 너무 많아지면 조금 지우기

Implication Graph

어떤 변수에게 참/거짓을 할당하였더니
다른 변수가 참/거짓이어야 하더라.

Implication Graph

$$\varphi = \omega_1 \wedge \omega_2 \wedge \omega_3 \wedge \omega_4 \wedge \omega_5 \wedge \omega_6$$
$$= (x_1 \vee x_8 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge$$
$$(\neg x_4 \vee \neg x_5) \wedge (x_7 \vee \neg x_4 \vee \neg x_6) \wedge (x_5 \vee x_6)$$

값을 할당하는 변수의 순서는

x_7 , x_8 , x_1 , ⋯

Implication Graph

$$\varphi = \omega_1 \wedge \omega_2 \wedge \omega_3 \wedge \omega_4 \wedge \omega_5 \wedge \omega_6$$
$$= (x_1 \vee x_8 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge$$
$$(\neg x_4 \vee \neg x_5) \wedge (x_7 \vee \neg x_4 \vee \neg x_6) \wedge (x_5 \vee x_6)$$

$x_7 = 0$

Implication Graph

$$\varphi = \omega_1 \wedge \omega_2 \wedge \omega_3 \wedge \omega_4 \wedge \omega_5 \wedge \omega_6$$
$$= (x_1 \vee x_8 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge$$
$$(\neg x_4 \vee \neg x_5) \wedge (x_7 \vee \neg x_4 \vee \neg x_6) \wedge (x_5 \vee x_6)$$

$x_8=0$

$x_7=0$

Implication Graph

$$\varphi = \omega_1 \wedge \omega_2 \wedge \omega_3 \wedge \omega_4 \wedge \omega_5 \wedge \omega_6$$
$$= (x_1 \vee x_8 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge$$
$$(\neg x_4 \vee \neg x_5) \wedge (x_7 \vee \neg x_4 \vee \neg x_6) \wedge (x_5 \vee x_6)$$

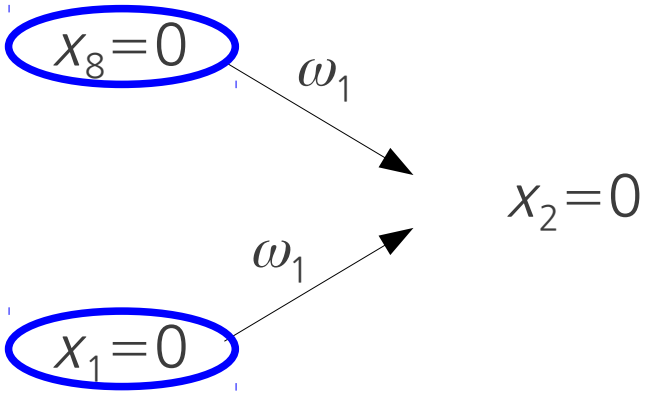
$x_8=0$

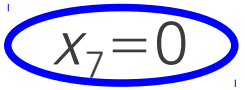
$x_1=0$

$x_7=0$

Implication Graph

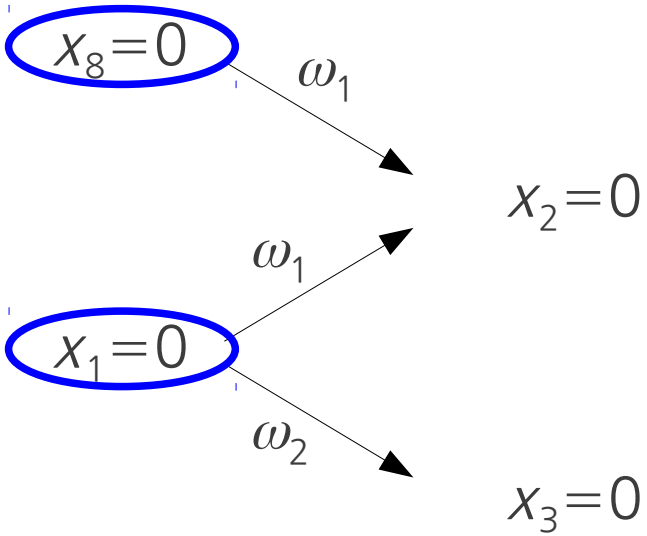
$$\varphi = \omega_1 \wedge \omega_2 \wedge \omega_3 \wedge \omega_4 \wedge \omega_5 \wedge \omega_6$$
$$= (x_1 \vee x_8 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge$$
$$(\neg x_4 \vee \neg x_5) \wedge (x_7 \vee \neg x_4 \vee \neg x_6) \wedge (x_5 \vee x_6)$$

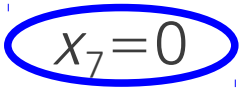




Implication Graph

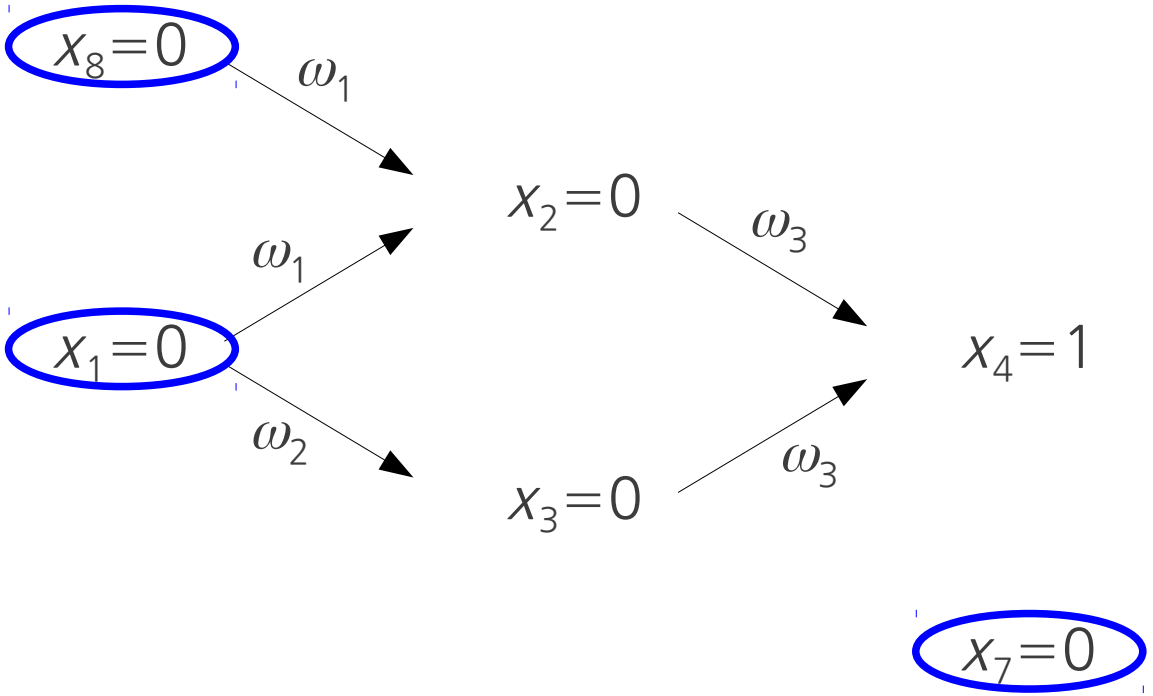
$$\varphi = \omega_1 \wedge \omega_2 \wedge \omega_3 \wedge \omega_4 \wedge \omega_5 \wedge \omega_6$$
$$= (x_1 \vee x_8 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge$$
$$(\neg x_4 \vee \neg x_5) \wedge (x_7 \vee \neg x_4 \vee \neg x_6) \wedge (x_5 \vee x_6)$$





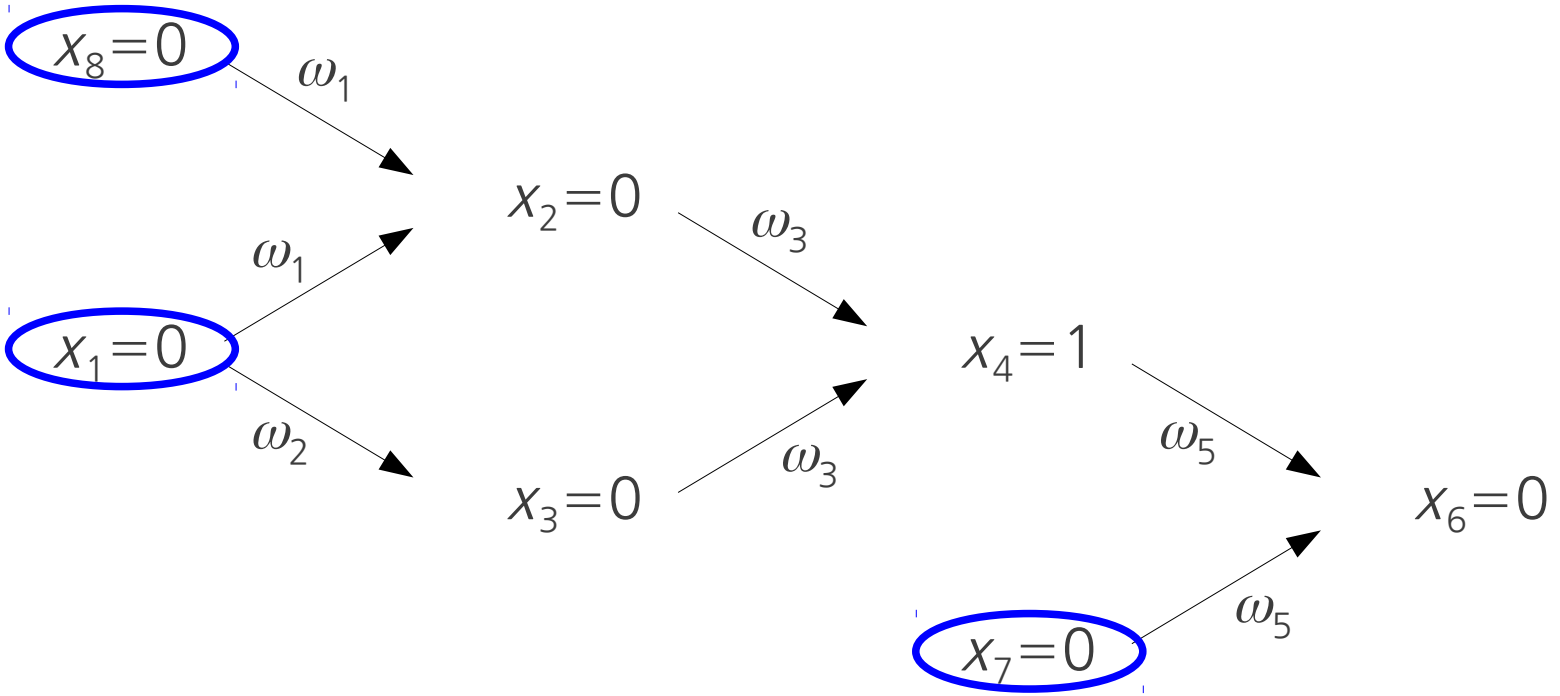
Implication Graph

$$\varphi = \omega_1 \wedge \omega_2 \wedge \omega_3 \wedge \omega_4 \wedge \omega_5 \wedge \omega_6$$
$$= (x_1 \vee x_8 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge$$
$$(\neg x_4 \vee \neg x_5) \wedge (x_7 \vee \neg x_4 \vee \neg x_6) \wedge (x_5 \vee x_6)$$



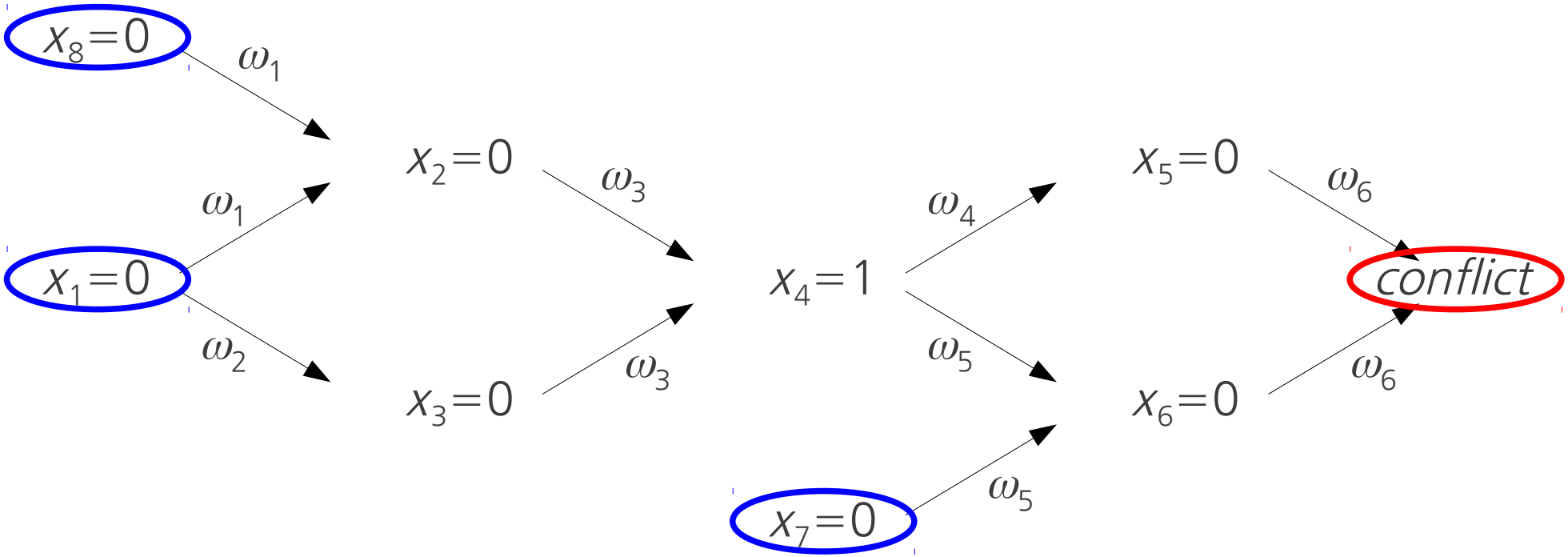
Implication Graph

$$\varphi = \omega_1 \wedge \omega_2 \wedge \omega_3 \wedge \omega_4 \wedge \omega_5 \wedge \omega_6$$
$$= (x_1 \vee x_8 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge$$
$$(\neg x_4 \vee \neg x_5) \wedge (x_7 \vee \neg x_4 \vee \neg x_6) \wedge (x_5 \vee x_6)$$



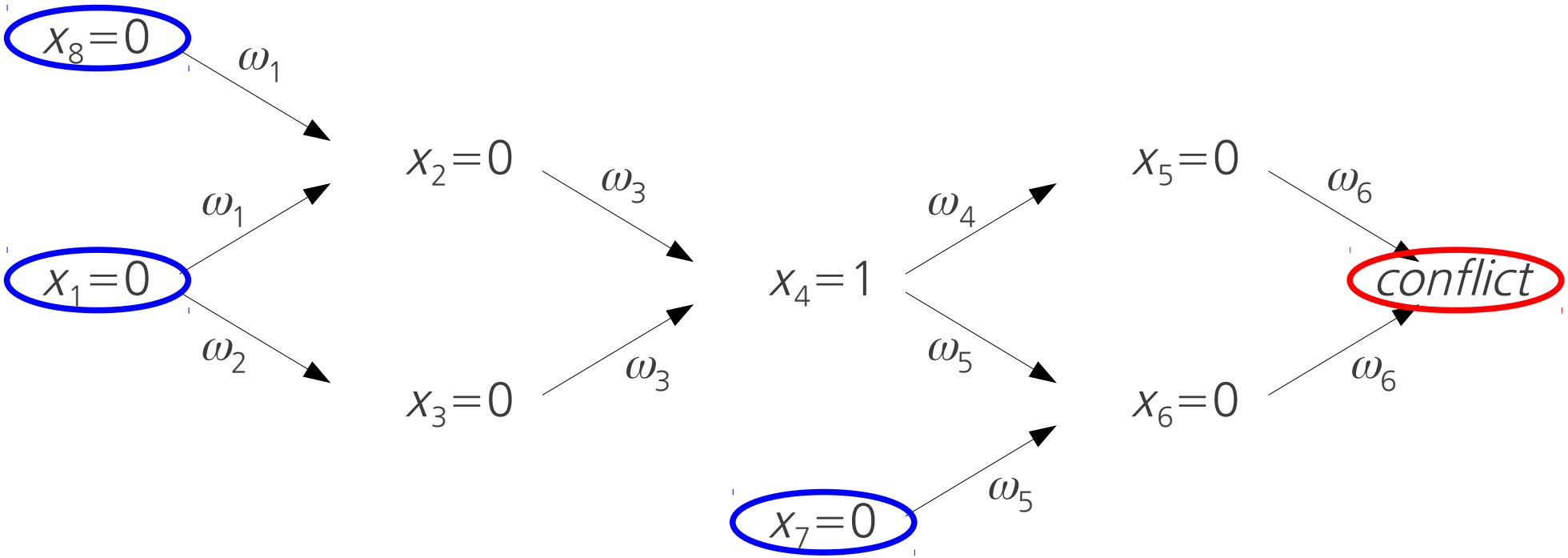
Implication Graph

$$\varphi = \omega_1 \wedge \omega_2 \wedge \omega_3 \wedge \omega_4 \wedge \omega_5 \wedge \omega_6$$
$$= (x_1 \vee x_8 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge$$
$$(\neg x_4 \vee \neg x_5) \wedge (x_7 \vee \neg x_4 \vee \neg x_6) \wedge (x_5 \vee x_6)$$



Clause 배우기

- 충돌을 일으키는데 기여하는 clause들로부터 새로운 clause를 만들어 배우자.



Clause 배우기

$\omega_6 = (x_5 \vee x_6)$

$\omega_4 = (\neg x_4 \vee \neg x_5)$

$\omega_5 = (x_7 \vee \neg x_4 \vee \neg x_6)$

$\omega_3 = (x_2 \vee x_3 \vee x_4)$

$\omega_1 = (x_1 \vee x_8 \vee \neg x_2)$

$\omega_2 = (x_1 \vee \neg x_3)$

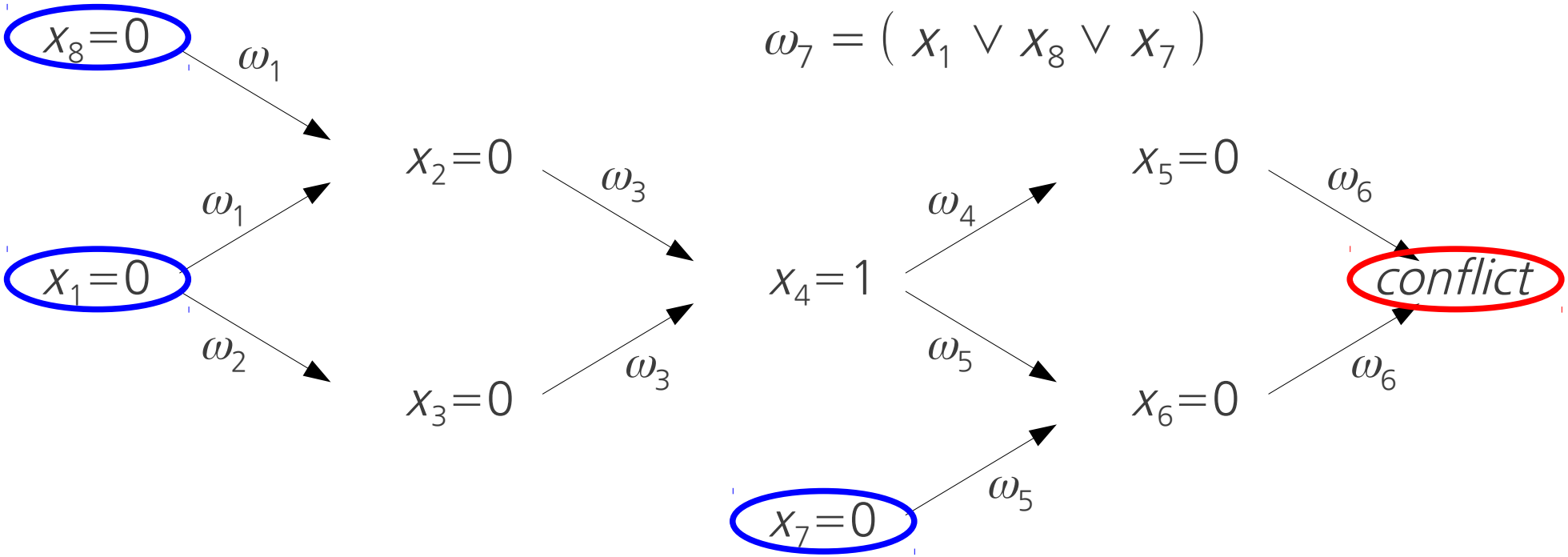
$(\neg x_4 \vee x_6)$

$(\neg x_4 \vee x_7)$

$(x_2 \vee x_3 \vee x_7)$

$(x_1 \vee x_8 \vee x_3 \vee x_7)$

$\omega_7 = (x_1 \vee x_8 \vee x_7)$



Clause 배웠다

$$\varphi = \omega_1 \wedge \omega_2 \wedge \omega_3 \wedge \omega_4 \wedge \omega_5 \wedge \omega_6$$
$$= (x_1 \vee x_8 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (x_7 \vee \neg x_4 \vee \neg x_6) \wedge (x_5 \vee x_6)$$

$$\omega_7 = (x_1 \vee x_8 \vee x_7)$$

$$\varphi = \omega_1 \wedge \omega_2 \wedge \omega_3 \wedge \omega_4 \wedge \omega_5 \wedge \omega_6 \wedge \omega_7$$
$$= (x_1 \vee x_8 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (x_7 \vee \neg x_4 \vee \neg x_6) \wedge (x_5 \vee x_6) \wedge (x_1 \vee x_8 \vee x_7)$$

결론

- 간단한 DPLL
  - 후진(backtracking)에 기초한 SAT 알고리즘
- CDCL
  - DPLL의 업그레이드 버전
  - 충돌로부터 제약조건을 배우는 SAT 알고리즘

감사합니다.