# Lecture 7 : Clause Learning

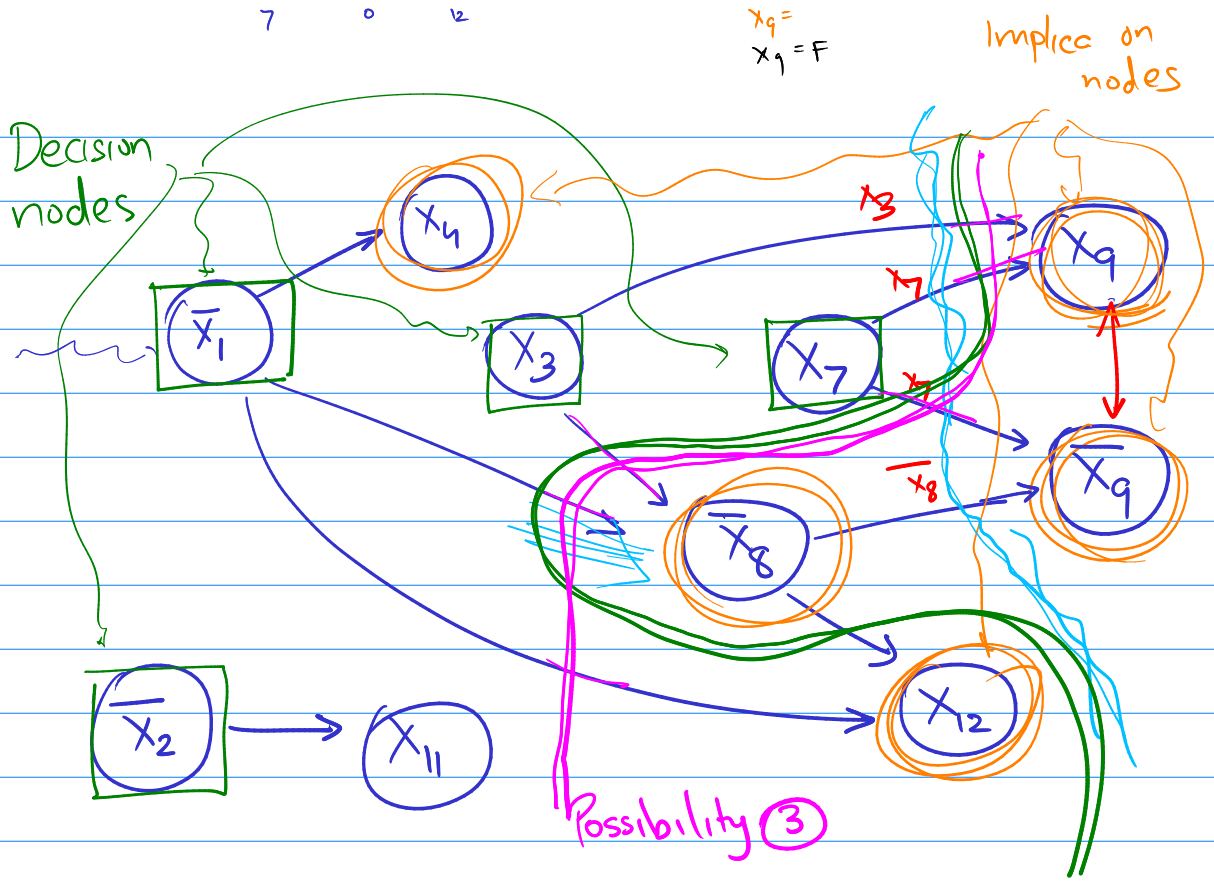
- By soundness, I mean that
  if the solver says unsat
  then the formula is actually unsat.
- Given a formula $\varphi$,
  another clause $c$ is a sound conclusion
  if whenever $\varphi$ is sat, so is $\varphi \wedge c$
  whenever $\varphi \wedge c$ is unsat, so is $\varphi$.
- In other words, does $\varphi \Rightarrow c$ ?

- Example

① Does $\varphi \Rightarrow (x_1 \vee x_2 \vee \overline{x_3} \vee \overline{x_7})$ Poss ②

② Does $\varphi \Rightarrow (x_1 \vee \overline{x_3} \vee \overline{x_7})$ Poss ③

If you claim that Poss ③ is sound doesn't it follow that Poss ② is also sound?



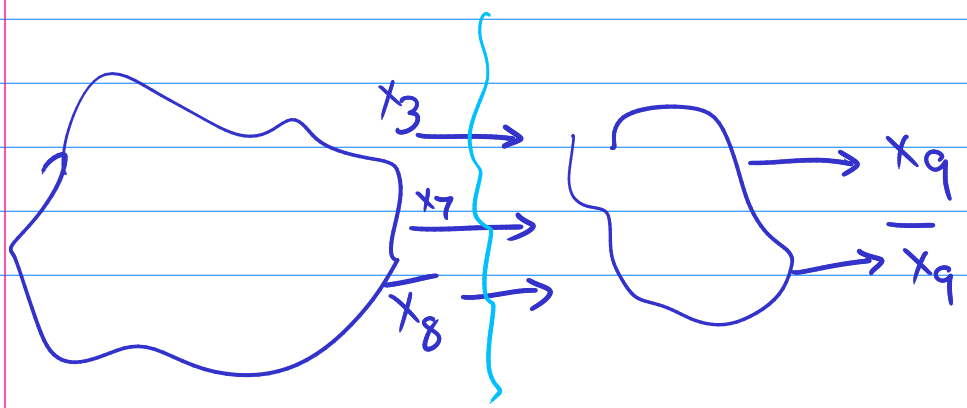

Requirements of blue cut:

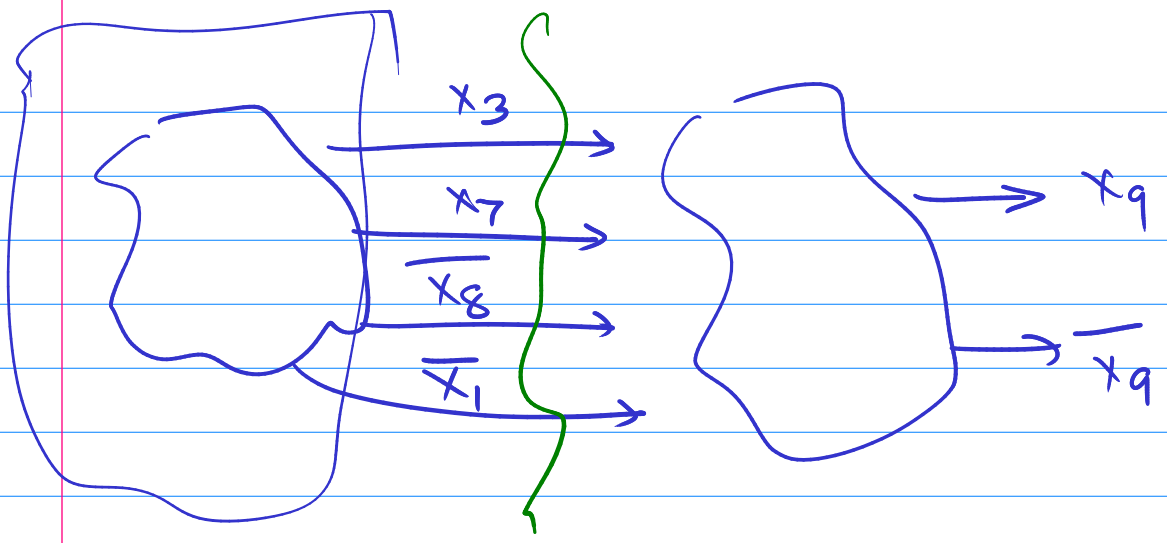
(1) All decision nodes on left side

(2) Both conflict nodes on the right side.

Log all variable assignments crossing the cut

Claim: If this situation repeats, a conflict will occur.



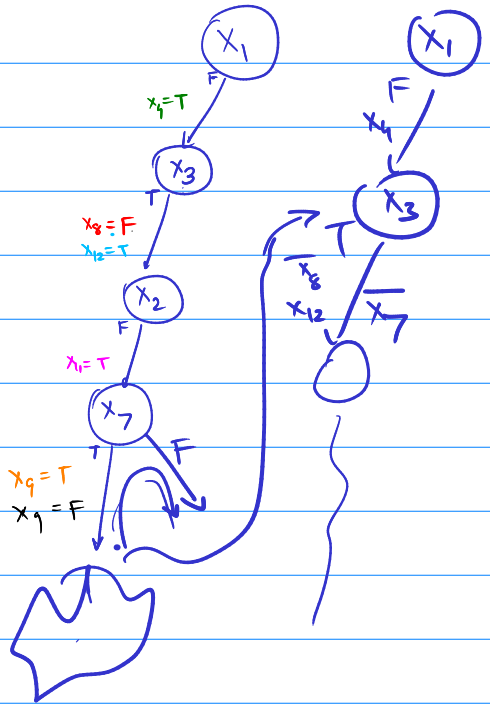



One possibility for learning conflict clauses:

Min cut in implication graph.

Examples of sound lemmas:

1. $x_1 \vee x_2 \vee \overline{x_3} \vee \overline{x_7}$
2. $x_1 \vee \overline{x_3} \vee \overline{x_7}$
3. $\overline{x_3} \vee \overline{x_7} \vee x_8$




Non-chronological backtracking

Backtrack to the earliest point where up would have precluded this partial assignment.

false ∨ (a ∨ b ∨ c)

CDCL ( φ  partial assignment S

implication graph G. )

- If empty clause exists: return unsat
- Perform up on S. (S' G')
- If conflict occurs: analyze conflict

add learned clause

"watched literals"

backtrack to earliest point allowing new up

- If formula is empty, then return sat
- Pick variable x
- Check CDCL ( φ  S' ∧ x  G' )

CDCL ( φ  S' ∧ $\bar{x}$  G' )

---

## Restarts  Gomes et al. AAAI 1998

Boosting combinatorial search through randomization

Restart after: 1 2 4 8 16 ... conflicts (geometric)

Luby sequence: 1 1 2 1 1 2 ... conflicts

Uniform sequence: 128 128 128 ... conflicts

# Decision heuristics

- Dynamic Largest Individual Sum

  Choose variable + assignment which satisfies most presently unsatisfied clauses.

- MAPLE sat from U Waterloo

  Machine learning guided decision heuristics

- Variable State Independent Decaying Sum

  (VSIDS) CHAFF

- For each variable $v$, $C_v$ = # of clauses where $v$ occurs positively

  $C_{\bar{v}}$ = # of clauses where $v$ occurs negatively

- Pick variable $v$ & polarity ($v$ / $\bar{v}$) such that $C_{v/\bar{v}}$ is maximized

- Periodically, divide all counters by 2.

# – Is propositional logic sufficient?

## Or, Satisfiability Modulo Theories (SMT)

```
x := input()
y := x + 3
if y < 0, then crash!
```

$\exists x.\ x + 3 < 0$

```
x := input()
y := x^2 + 3
If y < 0, then crash!
```

$\nexists x.\ x^2 + 3 < 0$

Bitblasting: One approach: x is ultimately a 32 bit number

Explicitly construct 32 bit arithmetic circuits.

Example: $x_1 \neq x_2$ and $x_2 = x_3$ and $x_3 = x_1$

not $x_1 = x_2$

not (a) and (b) and (c).

Example: $x_1 = x_2 + 8$ and ($x_3 = x_1 - 6$ or $x_3 = x_1 + 2$)

(a) (b) (c)

and ($x_3 = x_2 + 2$)

(d)

a and (b or c) and d.

- Common theories:

Equalities (E)

Equality with Uninterpreted Functions (EUF)

(LIA) Linear Integer Arithmetic / Difference logic

(LRA) Linear Real Arithmetic

(A) Arrays

(BV) Bitvectors

(S) Strings

AUFLIA

```
for (i=0; i < |a|; i++)
    for (j=0; j < |a| - i; j++)
        a[i+j] = 8
```

$i+j < |a|$

Is this array access safe?

DPLL / CDCL ⟶ DPLL(T)

Theory

Theory solvers

EUF

LRA / Difference logic ⟵ LIA

Bitvectors

Strings

Theory combination

$x = 0 \quad x < y < z \quad z = 1$