AIG to CNF

- **Most modern SAT solvers accept input formulae in conjunctive normal form (CNF)**
  - Literals: variables & their complements, e.g. a, b, ¬ c
  - Clauses: disjunction of literals, e.g. (a ∨ b ∨ ¬ c)
  - Cubes: conjunction of literals, e.g. (a ∧ b ∧ ¬ c)
  - CNF formula: conjunction of clauses, product-of-sums

    e.g. (a ∨ b ∨ ¬ c) ∧ (¬ a ∨ ¬ b ∨ d)
  - DNF formula: disjunction of cubes, sum-of-products

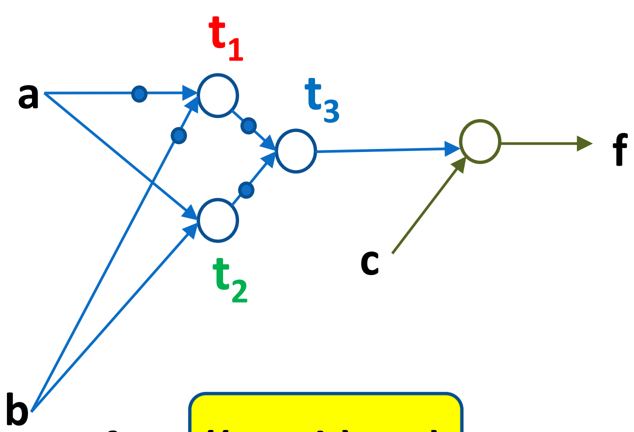
    e.g. (a ∧ c ∧ ¬ d) ∨ (¬ b ∧ d ∧ c)
- **Given AIG for f, generating f in CNF can blow up badly**
  - Start from DNF formula f (size: |f|)
  - Build AIG for f (size: O(|f|))
  - Convert AIG to CNF
    - Effectively convert DNF to CNF: known worst-case exponential blow-up

AIG to CNF

- **Solution: Given AIG for f, generate equisatisfiable g in CNF**
- **Equisatisfiablity**
  - f and g equisatisfiable iff both f and g are satisfiable or both are unsatisfiable
  - $(a \vee b)$ equisatisfiable with $(c \wedge d)$, not equisatisfiable with $d \wedge (\neg d \vee c) \wedge \neg c$
  - Checking satisfiability of g tells us if f is satisfiable
- **Tseitin encoding**
  - Given a Boolean circuit (AIG is a special case) for f, generate equisatisfiable g
    - g is in CNF
    - Every satisfying assignment for f gives unique satisfying assignment for g and vice-versa
    - Size of g linear in size of circuit (AIG) for f
  - Widely used in SAT solving context

AIG to CNF: Tseitin Encoding

- Given Boolean circuit (or AIG) for f
  - Associate new variable with output of each gate
  - For each Boolean gate, generate formula expressing output in terms of inputs
  - Convert each formula generated above to CNF
    - Boolean gate with fixed # inputs $\Rightarrow$ CNF formula for gate is of fixed size
  - Conjoin CNF formula for each gate and output variable of circuit



$f = ((a \oplus b) \wedge c)$

$t_1 \leftrightarrow (\neg a \wedge \neg b)$

$t_2 \leftrightarrow (a \wedge b)$

$t_3 \leftrightarrow (\neg t_1 \wedge \neg t_2)$

$f \leftrightarrow (t_3 \wedge c)$

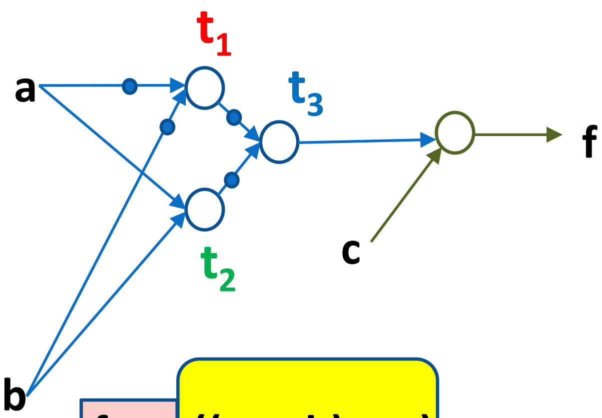
$(\neg t_1 \vee \neg a) \wedge (\neg t_1 \vee \neg b) \wedge (a \vee b \vee t_1) \wedge$

$(\neg t_2 \vee a) \wedge (\neg t_2 \vee b) \wedge (\neg a \vee \neg b \vee t_2) \wedge$

$(\neg t_3 \vee \neg t_1) \wedge (\neg t_3 \vee \neg t_2) \wedge (t_1 \vee t_2 \vee t_3) \wedge$

$(\neg f \vee t_3) \wedge (\neg f \vee c) \wedge (\neg t_3 \vee \neg c \vee f) \wedge f$

AIG To CNF: Tseitin Encoding



$f = ((a \oplus b) \wedge c)$

$t_1 \leftrightarrow (\neg a \wedge \neg b)$ | $(\neg t_1 \vee \neg a) \wedge (\neg t_1 \vee \neg b) \wedge (a \vee b \vee t_1) \wedge$

$t_2 \leftrightarrow (a \wedge b)$ | $(\neg t_2 \vee a) \wedge (\neg t_2 \vee b) \wedge (\neg a \vee \neg b \vee t_2) \wedge$

$t_3 \leftrightarrow (\neg t_1 \wedge \neg t_2)$ | $(\neg t_3 \vee \neg t_1) \wedge (\neg t_3 \vee \neg t_2) \wedge (t_1 \vee t_2 \vee t_3) \wedge$

$f \leftrightarrow (t_3 \wedge c)$ | $(\neg f \vee t_3) \wedge (\neg f \vee c) \wedge (\neg t_3 \vee \neg c \vee f) \wedge f$

$= g$

» Bijection between SAT assignments of f and g

For every SAT assignment of f

> Evaluate all gate outputs for given assignment of inputs

> a = 1, b = 0, c = 1 gives a = 1, b = 0, c = 1, $t_1 = 0$, $t_2 = 0$, $t_3 = 1$, f = 1

SAT assignment of g

For every SAT assignment of g

> Project assignment on variables of f

> a = 1, b = 0, c = 1, t_1 = 0, t_2 = 0, t_3 = 1, f = 1 gives a = 1, b = 0, c = 1

SAT assignment of f