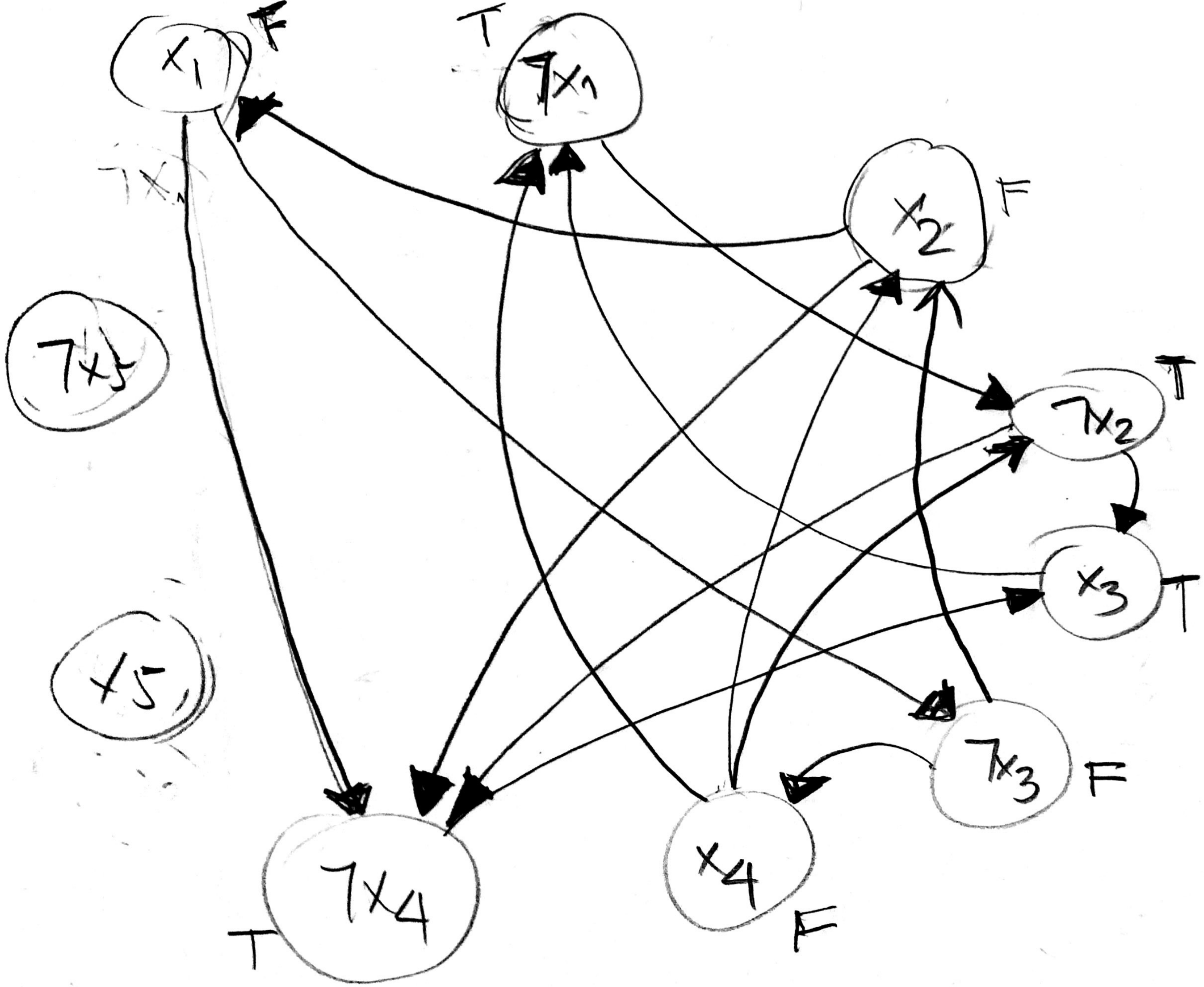


$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_3) \wedge (x_3 \vee x_4) \wedge (\neg x_2 \vee \neg x_4) \wedge (x_2 \vee \neg x_4)$$

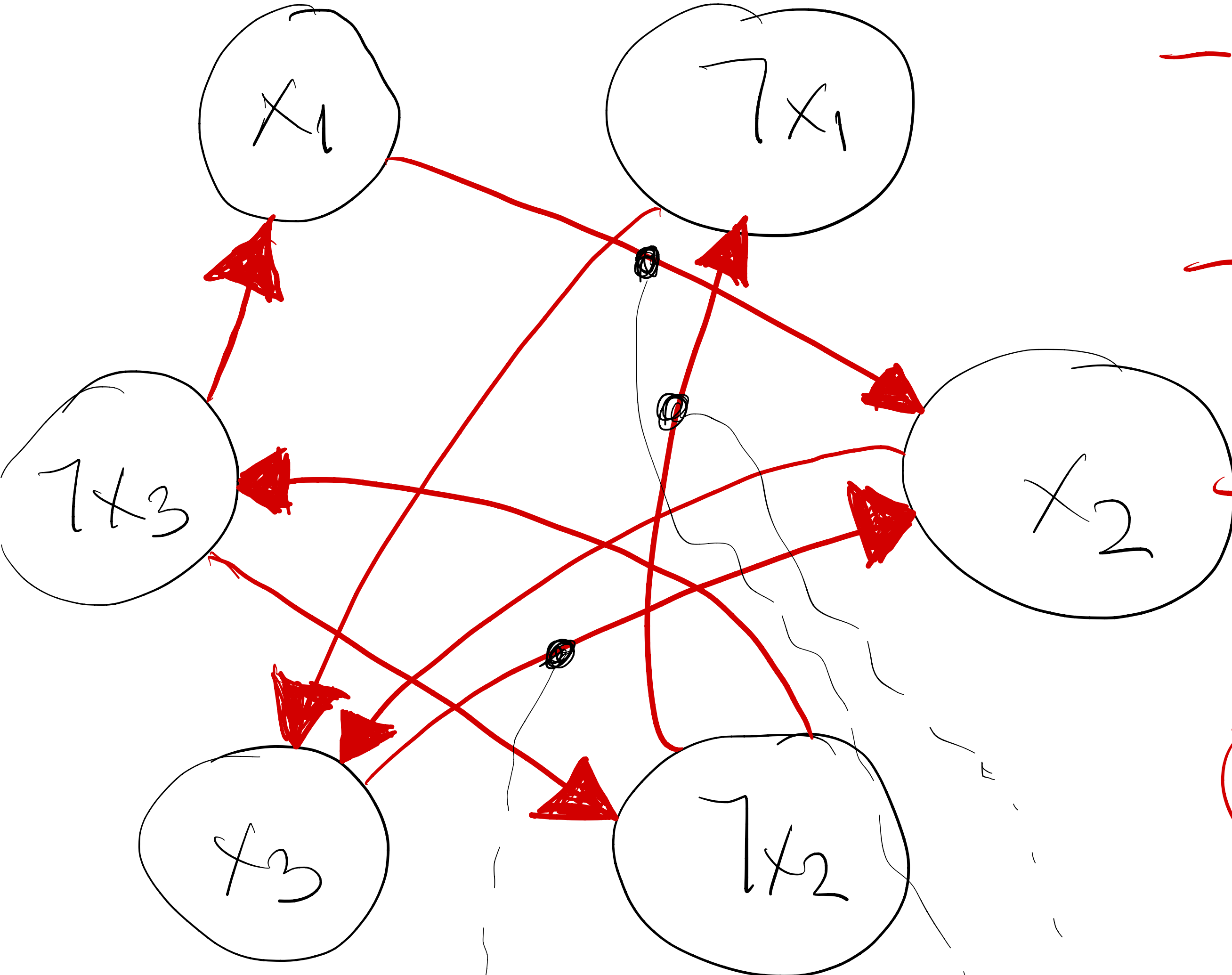
$x_2 \Rightarrow x_1$ | $x_1 \Rightarrow \neg x_3$ | $\neg x_3 \Rightarrow x_4$ | $x_2 \Rightarrow \neg x_4$ | $\neg x_2 \Rightarrow \neg x_4$

$\neg x_1 \Rightarrow \neg x_2$ | $x_3 \Rightarrow \neg x_1$ | $\neg x_4 \Rightarrow x_3$ | $x_4 \Rightarrow \neg x_2$ | $x_4 \Rightarrow x_2$

$$(\neg x_1 \vee \neg x_4) \wedge (x_2 \vee x_3)$$

$x_1 \Rightarrow \neg x_4$ | $\neg x_2 \Rightarrow x_3$

$x_4 \Rightarrow \neg x_1$ | $\neg x_3 \Rightarrow x_2$



1) input formula
each clause $(X \vee Y)$

2) transform each clause into 2 implications

3) Graph: nodes: all literals and all $\neg$literals

4) Make sure all implications are true.

$$(x_2 \vee \neg x_3) \wedge (x_3 \vee x_1) \wedge (x_2 \vee \neg x_1) \wedge (x_3 \vee \neg x_2)$$

$x_3 \Rightarrow x_2$	$\neg x_3 \Rightarrow x_1$	$\neg x_2 \Rightarrow \neg x_1$	$\neg x_3 \Rightarrow \neg x_2$
$\neg x_2 \Rightarrow \neg x_3$	$\neg x_1 \Rightarrow x_3$	$x_1 \Rightarrow x_2$	$x_2 \Rightarrow \cancel{\neg} x_3$

$$\boxed{X \vee Y} \equiv \boxed{\neg X \Rightarrow Y} \equiv \boxed{\neg Y \Rightarrow X}$$

X	Y	$\neg X$	$\neg Y$	$X \vee Y$	$\neg X \Rightarrow Y$	$\neg Y \Rightarrow X$
0	0	1	1	0	0	0
0	1	1	0	1	1	1
1	0	0	1	1	1	1
1	1	0	0	1	1	1

part A, Satisfiability Intro [easy]. A boolean formula is satisfiable if there exists some variable assignment that makes the formula evaluate to true. Namely, a boolean formula is satisfiable if there is some row of the truth table that comes out true. Determining whether an arbitrary boolean formula is satisfiable is called the *Satisfiability Problem.* There is no known efficient solution to this problem, in fact, an efficient solution would earn you a million dollar prize. While this is hard problem in computer science, not all instances of the problem are hard, in fact, determining satisfiability for some types of boolean formulae is easy.

i. First, let's consider why this would be hard. If you knew nothing about a given boolean formula other than that it had n variables, how large is the truth table you would need to construct? Please indicate the number of columns and rows as a function of n

ii. Now consider the following 100 variable formula.

BiG TT

$$x_1 \wedge \boxed{(\neg x_1 \vee x_2)} \wedge (\neg x_2 \vee x_3) \wedge \boxed{(\neg x_3 \vee x_4)} \wedge \ldots \wedge \boxed{(\neg x_{99} \vee x_{100})}$$

Without constructing a truth table, how many satisfying assignments does this formula have, explain your answer.

every clause 3 literals, ex. $(x_1 \wedge x_5 \wedge \neg x_{72})$

not CNF

iii. Now consider an arbitrary 3-DNF formula with 100 variables and 200 clauses. 3-DNF means that the formula is in disjunctive normal form and each clause has three literals. (A literal is the instantiation of the variable in the formula, so for x, $\neg x$ or x.) An example might be something like:

$$\boxed{(\neg x_1 \wedge x_3 \wedge x_{10})} \vee \boxed{(\neg x_3 \wedge x_{15} \wedge \neg x_{84})} \vee \boxed{(x_{17} \wedge \neg x_{37} \wedge x_{48})} \vee \ldots \vee \boxed{(\neg x_{87} \wedge \neg x_{95} \wedge x_{100})}$$

What is the largest size truth table needed to solve this problem. What is the maximum number of such truth tables needed to determine satisfiabilty.

TRICKY

x_1 x_3 x_{10} | $\vee$

1) $\exists \Rightarrow$ true

2) $\forall$ assig $\Rightarrow$ false

2 options on 8 first clause

8

x_{17} x_{37} x_{48}

8

8

8 rows per TT, max 200 TTs $\Rightarrow$ correct?

- overlap? x_2 or $\neg x_2$ in multiple clauses.

satisfiability = "make it true" DNF $\Longleftrightarrow$ $\boxed{\text{one}}$ clause has to be true