Homework 9, Solutions CPSC 421/501 2024

(1) Joel Friedman

(2) If $f(\vec{x}) = T \iff \vec{x} = (T, F, T)$ or (F, F, F),

(a) then f is expressed by the formula

$$(x_1 = T \text{ and } x_2 = F \text{ and } x_3 = T)$$

$$\text{or } (x_1 = F \text{ and } x_2 = F \text{ and } x_3 = F)$$

i.e.

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

(b) $\neg f$ is obtained by exchanging $\wedge$ with $\vee$ and negating all literals in

hence

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

(3) Note that $x_1 \vee x_2 \vee x_3$ holds whenever

$x_1 = x_2 = x_3 = T$ and $x_4, \dots, x_n$ are any values

in $\{F, T\}$, giving 2^{n-3} possible values of $\vec{x}$

where $x_1 \vee x_2 \vee x_3 = F$. Similarly for

$y_i \vee y_j \vee y_k$ where $y_i = x_i$ or $\neg x_i$

$y_j = x_j$ or $\neg x_j$

$y_k = x_k$ or $\neg x_k$

and i, j, k are distinct. If a 3CNF clause contains a literal and its negation, (e.g. $\neg x_1 \vee x_1 \vee x_2$) then it is always true. Otherwise we have a literal repeated twice (e.g. $x_1 \vee x_1 \vee x_2$) or three times (e.g. $\neg x_1 \vee \neg x_1 \vee \neg x_1$)

in which case this is false for 2^{n-2} or 2^{n-1} values of $\vec{x} \in \{F,T\}^n$. Hence any 3CNF formula represents a function $f: \{F,T\}^n \to \{F,T\}$ that is either

(1) never false, or

(2) false for at least 2^{n-3} values in $\{F,T\}^n$

Hence $f(x_1,\ldots,x_4) = x_1 \vee x_2 \vee x_3 \vee x_4$, which is false on only 1 value (and $1 < 2^{n-3} = 2$) cannot be expressed as a 3CNF

(4) Say that $a_1 \vee a_2 \vee \dots \vee a_n = T$.

Then at least one of $a_1, \dots, a_n = T$.

If $a_i = T$ and $i = 1, 2$, we can set

$z_1 = z_2 = \dots = z_{n-3} = F$ to make

$a_1 \vee a_2 \vee z_1 = T$, and all other clauses

are true since they contain one of

$\neg z_1, \neg z_2, \dots, \neg z_{n-3}$ all of which are T.

Similarly if $i = n-1, n$, setting

$z_1 = z_2 = \dots = z_{n-3} = T$.

If $3 \le i \le n-2$, the clause

$$\neg z_{i-2} \vee a_i \vee z_{i-1} = T$$

if $z_{i-2} = T$ and $z_{i-1} = F$, and so

taking $z_1 = z_2 = \dots = z_{i-2} = T$ and

$z_{i-1} = \cdots = z_{n-3} = F$ makes all the other clauses contain at least one T.

Hence

$$a_1 \vee a_2 \vee \cdots \vee a_n = T \Rightarrow \varphi(z_1, \ldots, z_{n-3}) \text{ is satisfiable}$$

Now say that
$a_1 \vee a_2 \cdots \vee a_n \neq T$.
Then $a_1 = \cdots = a_n = F$, and

$$\varphi(z_1, \ldots, z_{n-3})$$

$$= (F \vee F \vee z_1) \wedge (\neg z_1 \vee F \vee z_2) \wedge \cdots \wedge (\neg z_{n-3} \vee F \vee F)$$

which is equivalent to

$$z_1 \wedge (\neg z_1 \vee z_2) \wedge \cdots \wedge (\neg z_{n-4} \vee z_{n-3}) \wedge (\neg z_{n-3})$$

We claim that is always F, for if not, then $z_1 = T$, and then $\neg z_1 \vee z_2 = T$

So $z_2 = T$, and similarly $z_3 = \dots = z_{n-3} = T$.

But the last clause is $\neg z_{n-3}$, which is false. Hence $\varphi \notin SAT$. Hence

$$a_1 \vee a_2 \vee \dots \vee a_n = T \Rightarrow \varphi \in SAT, \text{ and}$$

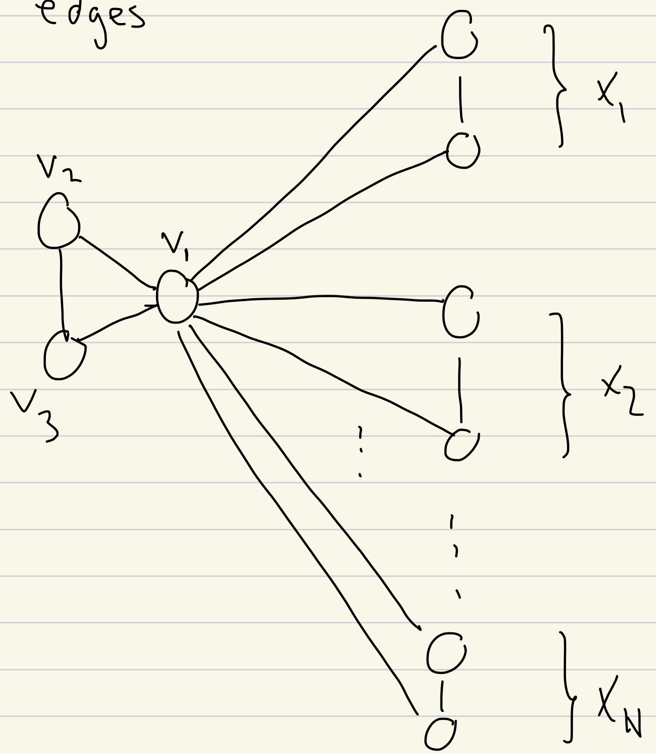
$$a_1 \vee a_2 \vee \dots \vee a_n = F \Rightarrow \varphi \notin SAT.$$

Hence

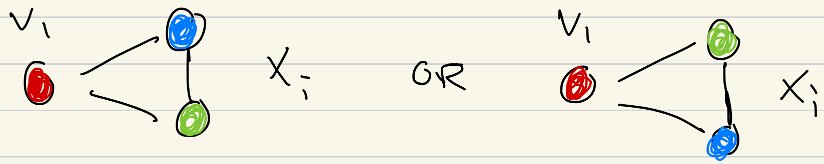
$$a_1 \vee a_2 \vee \dots \vee a_n = T \Leftrightarrow \varphi \in SAT.$$

(5)

Given a 3CNF in variables $x_1, \dots, x_N$, for each variable introduce vertices and edges

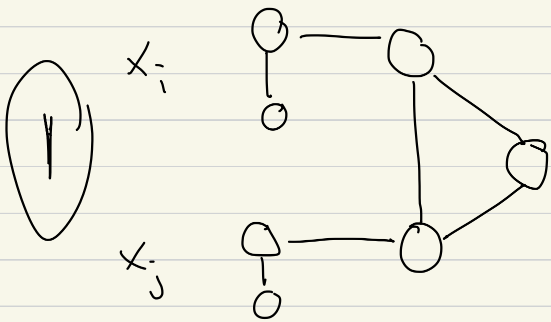


Hence if v_1 has the colour red (say the colours are R = red, G = green, B = blue) then each variable must be coloured G, B or B, G:



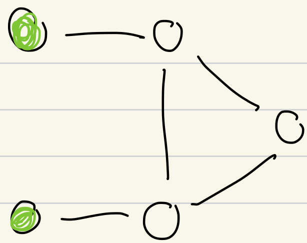
Now we view the top x_1 vertex as representing T (true), the bottom as F (false).

For each clause $C_1, \ldots, C_m$ of the 3CNF, we add the following: for a clause x_i OR x_j OR x_k we add:

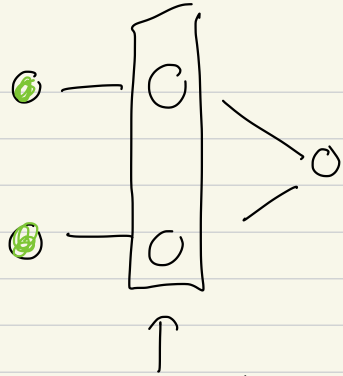


this can be blue iff one of the top x_i or top x_j vertices is blue

Since

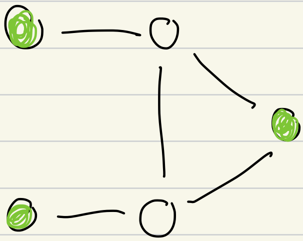


implies

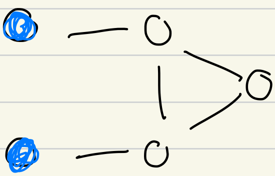


one is blue,
one red

implies

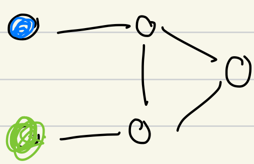


but, similarly

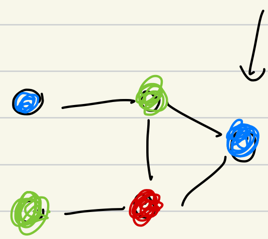


implies this
must be blue,

and

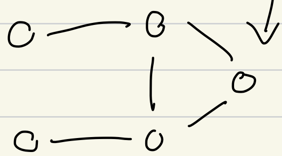


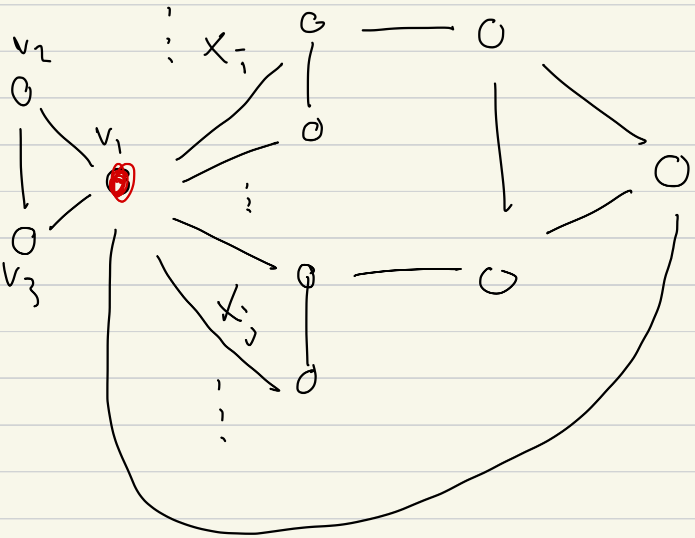
allows
for



can be
blue

To enforce that is blue or green, we can add an edge to "red"

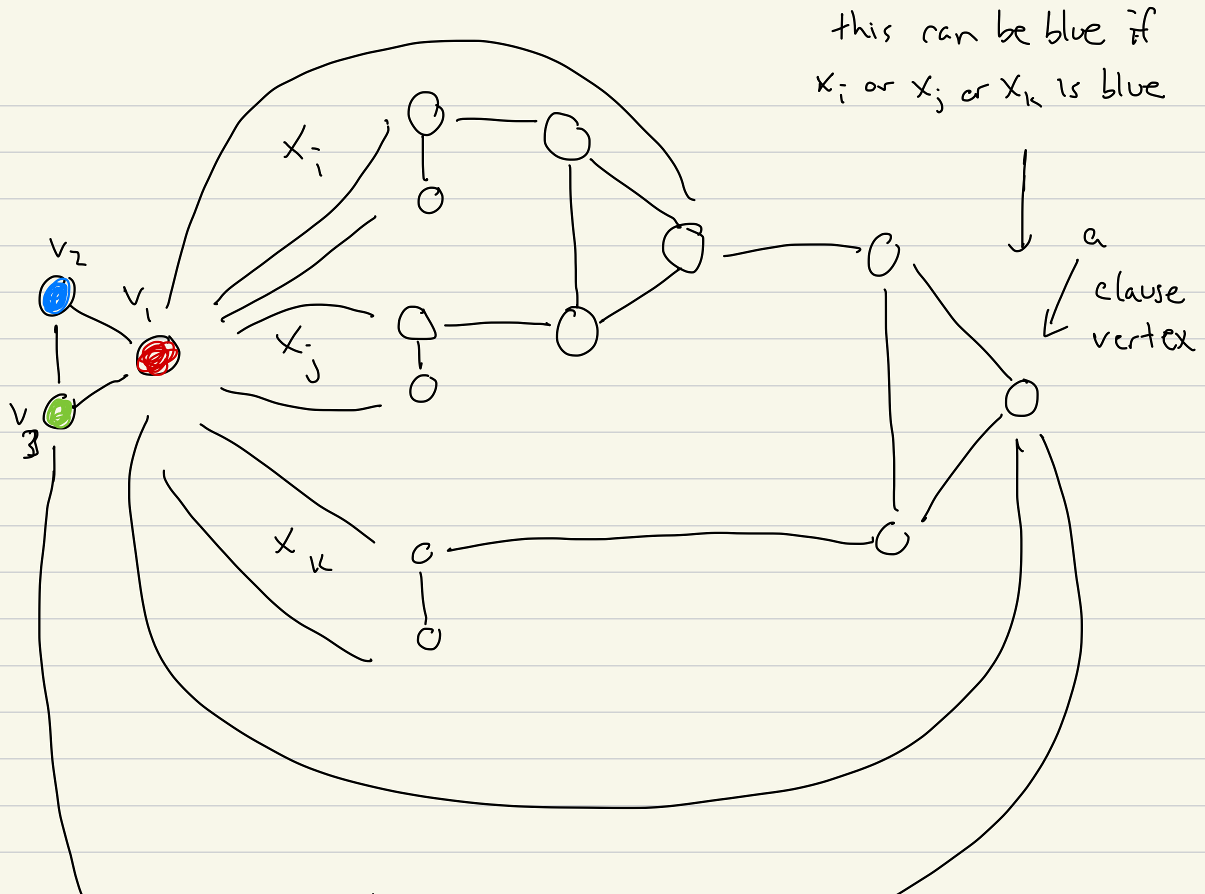




"x_i OR x_j"

enforce green or blue

② We now add a similar "OR" gadget between "x_i OR x_j" and "x_k"



now we add an edge to insist

hence the rightmost vertex must be blue, i.e. the colour at v_2 on the left. We do the same for every other clause, but connecting to the bottom vertex of x_i if $\neg x_i$ appears, and similarly for $x_j, \neg x_j$ and $x_k, \neg x_k$.

Hence we can satisfy the 3CNF iff

each clause vertex is colourable (with the color at V_2, here shown in blue).

Hence the 3CNF in question is satisfiable iff this graph can be 3-coloured.