1. a) $\{\}$ no clauses
b) $\{\epsilon\}$ the empty clause

2. $(x \wedge y) \to z$ $\quad \neg(x \wedge y) \vee z = \neg x \vee \neg y \vee z$

h = healthy
l = lazy
d = dancer
p = happy

$h \wedge \bar{l} \to d$	$\bar{h} l d$	✓
$l \wedge \bar{d} \to p$	$\bar{l} d p$	✓
$h \wedge d \to p$	$\bar{h} \bar{d} p$	✓
$p \wedge \bar{d} \to h$	$\bar{p} d h$	✓
$l \wedge h \to \bar{p}$	$\bar{l} \bar{h} \bar{p}$	✓
$\bar{p} \wedge \bar{h} \to l$	$p h l$	✓
$l \wedge d \to h$	$\bar{l} \bar{d} h$	✓

$x = \; d \bar{l} p$ $\quad$ dance, happy, not lazy

3. $$\sum_{d=1}^{n-1} n-(j-1)d + \sum_{d=1}^{n-1} n-(k-1)d$$ with $n-(j-1)d \geq 0$

$$n-(j-1)d \geq 0$$
$$d \leq \left\lfloor \frac{n}{j-1} \right\rfloor$$

$$\sum_{d=1}^{\lfloor \frac{n}{j-1} \rfloor} n-(j-1)d = n\left\lfloor \frac{n}{j-1} \right\rfloor - (j-1) \frac{\lfloor \frac{n}{j-1} \rfloor \left(\lfloor \frac{n}{j-1} \rfloor + 1 \right)}{2}$$

$$|\mathrm{waerden}(j,k;n)| = n\left\lfloor \frac{n}{j-1} \right\rfloor - (j-1)\left\lfloor \frac{n}{j-1} \right\rfloor \left(\left\lfloor \frac{n}{j-1} \right\rfloor + 1 \right)/2 + n\left\lfloor \frac{n}{k-1} \right\rfloor - (k-1)\left\lfloor \frac{n}{k-1} \right\rfloor \times \ldots$$
$$\ldots \left(\left\lfloor \frac{n}{k-1} \right\rfloor + 1 \right)/2$$

7. $$\{ (x_i \vee x_{i+2^d} \vee x_{i+2^{d+1}}) \mid 1 \leq i \leq n-2^{d+1}, d \geq 0 \} \cup$$
$$\{ (\bar{x}_i \vee \bar{x}_{i+2^d} \vee \bar{x}_{i+2^{d+1}}) \mid 1 \leq i \leq n-2^{d+1}, d \geq 0 \}$$

no 3 of same color are equally spaced by powers of 2

the string 001001001... is always a solution
- 1s are always a multiple of 3 apart
- 0s in positions i and j, such that $i \equiv j \pmod 3$ are always a multiple of 3 apart.

10. m clauses, n variables

$\Downarrow$

$m+n$ clauses, $2n$ variables
first m clauses have only negative literals
last n clauses have only positive literals, binary

x_{i0} - the original x_i, used positively
x_{i1} - the original x_i, used as its negation

Then, $\{(\bar{x}_{i0} \vee \bar{x}_{i1}) \mid 1 \le i \le n\} \cup \{$the original clauses with x_i replaced with x_{i0} and $\bar{x}_i$ replaced with $x_{i1}\}$.

12. $$S_1(y_1, \dots, y_p) = (y_1 \vee \dots \vee y_p) \wedge \bigwedge_{1 \le j < k \le p} (\bar{y}_j \vee \bar{y}_k)$$

$$S_{\le 1}(y_1, \dots y_p) = \exists t (S_{\le 1}(y_1, \dots, y_j, t) \wedge S_{\le 1}(\bar{t}, y_{j+1}, \dots, y_p))$$

$$S_1(y_1, \dots, y_p) = (y_1 \vee \dots \vee y_p) \wedge S_{\le 1}(y_1, \dots, y_p)$$

add new variables every 2 original variables, taking care of the edges with base cases.

$$\underbrace{y_1\, y_2\, y_3\, t_1}_{S_{\le 1}}\ \underbrace{\bar{t}_1\, y_4\, y_5\, t_2}_{S_{\le 1}}\ \underbrace{\bar{t}_2\, y_6\, y_7}_{S_{\le 1}} \dots \quad \dots\, t_K\, y_p$$

$\dots\, t_K\, y_{p-1}\, y_p$

$\dots\, t_K\, y_{p-2}\, y_{p-1}\, y_p$

13. a) ${}^{1}d_1 s_1 s_3$, ${}^{2}d_1 s_2 s_4$, ${}^{3}d_1 s_3 s_5$, ${}^{4}d_1 s_4 s_6$, ${}^{5}d_1 s_5 s_7$, ${}^{6}d_1 s_6 s_8$
${}^{7}d_2 s_1 s_4$, ${}^{8}d_2 s_2 s_5$, ${}^{9}d_2 s_3 s_6$, ${}^{10}d_2 s_4 s_7$, ${}^{11}d_2 s_5 s_8$
${}^{12}d_3 s_1 s_5$, ${}^{13}d_3 s_2 s_6$, ${}^{14}d_3 s_3 s_7$, ${}^{15}d_3 s_4 s_8$ $\quad$ 1 2 3 4 5 6 7 8
${}^{16}d_4 s_1 s_6$, ${}^{17}d_4 s_2 s_7$, ${}^{18}d_4 s_3 s_8$

$$S_1(x_4, x_6, x_9, x_{13}, x_{16}) \wedge S_1(x_5, x_{10}, x_{14}, x_{17}) \wedge S_1(x_6, x_{11}, x_{15}, x_{18})$$
$$\wedge S_1(x_1, x_2, x_3, x_4, x_5, x_6) \wedge S_1(x_7, x_8, x_9, x_{10}, x_{11}) \wedge S_1(x_{12}, x_{13}, x_{14}, x_{15})$$
$$\wedge S_1(x_{16}, x_{17}, x_{18}) \wedge S_1(x_1, x_7, x_{12}, x_{16}) \wedge S_1(x_2, x_8, x_{13}, x_{17})$$
$$\wedge S_1(x_3, x_1, x_9, x_{14}, x_{18}) \wedge S_1(x_2, x_4, x_7, x_{10}, x_{15}) \wedge S_1(x_3, x_5, x_8, x_{11}, x_{12})$$

b) If the instance is represented by a disjunction of S_1 clauses, we can add each binary clause $(\bar{y}_i \vee \bar{y}_j)$ once iff. y_i and y_j appear both in at least one of the S_1 clauses.

c) See print from Kdlimsat.

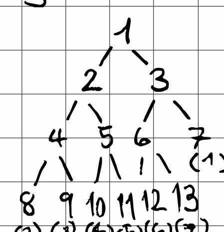
14. They make the instance more constrained, so that solvers might consider fewer nodes in the search tree. The additional clauses are also binary, which cascades unit propagation.

23. $n=7 \quad r=4$

$$
\begin{array}{ll|l}
 & & K \\
 & (\bar{s}_1^1 \vee s_2^1) \wedge (\bar{s}_2^1 \vee s_3^1) & 1 \\
\wedge & (\bar{s}_1^2 \vee s_2^2) \wedge (\bar{s}_2^2 \vee s_3^2) & 2 \\
\wedge & (\bar{s}_1^3 \vee s_2^3) \wedge (\bar{s}_2^3 \vee s_3^3) & 3 \\
\wedge & (\bar{s}_1^4 \vee s_2^4) \wedge (\bar{s}_2^4 \vee s_3^4) & 4 \\
\wedge & (\bar{x}_1 \vee s_1^1) \wedge (\bar{x}_2 \vee s_2^1) \wedge (\bar{x}_3 \vee s_3^1) & 0 \\
\wedge & (\bar{x}_2 \vee \bar{s}_1^1 \vee s_1^2) \wedge (\bar{x}_3 \vee \bar{s}_2^1 \vee s_2^2) \wedge (\bar{x}_4 \vee \bar{s}_3^1 \vee s_3^2) & 1 \\
\wedge & (\bar{x}_3 \vee \bar{s}_1^2 \vee s_1^3) \wedge (\bar{x}_4 \vee \bar{s}_2^2 \vee s_2^3) \wedge (\bar{x}_5 \vee \bar{s}_3^2 \vee s_3^3) & 2 \\
\wedge & (\bar{x}_4 \vee \bar{s}_1^3 \vee s_1^4) \wedge (\bar{x}_5 \vee \bar{s}_2^3 \vee s_2^4) \wedge (\bar{x}_6 \vee \bar{s}_3^3 \vee s_3^4) & 3 \\
\wedge & (\bar{x}_5 \vee \bar{s}_1^4) \wedge (\bar{x}_6 \vee \bar{s}_2^4) \wedge (\bar{x}_7 \vee \bar{s}_3^4) & 4
\end{array}
$$

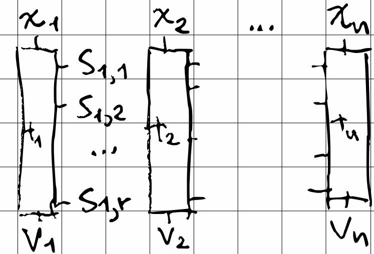
+12 variables
+23 clauses

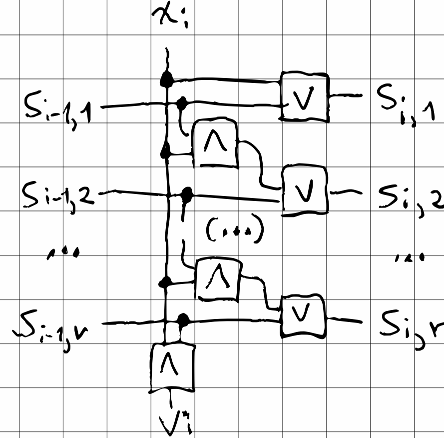
$$
\begin{array}{ll}
 & (\bar{x}_2 \vee b_1^4) \wedge (\bar{x}_3 \vee b_1^4) \wedge (\bar{x}_2 \vee \bar{x}_3 \vee b_2^4) \\
\wedge & (\bar{x}_4 \vee b_1^5) \wedge (\bar{x}_5 \vee b_1^5) \wedge (\bar{x}_4 \vee \bar{x}_5 \vee b_2^5) \\
\wedge & (\bar{x}_6 \vee b_1^6) \wedge (\bar{x}_7 \vee b_1^6) \wedge (\bar{x}_6 \vee \bar{x}_7 \vee b_2^6) \\
\wedge & (\bar{x}_1 \vee b_1^3) \wedge (\bar{b}_1^6 \vee b_1^3) \wedge (\bar{x}_1 \vee \bar{b}_1^6 \vee b_2^3) \wedge (\bar{b}_2^6 \vee b_2^3) \\
\wedge & (\bar{x}_1 \vee \bar{b}_2^6 \vee b_3^3) \\
\wedge & (\bar{b}_1^4 \vee b_1^2) \wedge (\bar{b}_1^5 \vee b_1^2) \wedge (\bar{b}_2^4 \vee b_2^2) \wedge (\bar{b}_2^5 \vee b_2^2) \wedge (\bar{b}_1^4 \vee \bar{b}_1^5 \vee b_2^2) \\
\wedge & (\bar{b}_2^4 \vee \bar{b}_1^5 \vee b_3^2) \wedge (\bar{b}_1^4 \vee \bar{b}_2^5 \vee b_3^2) \wedge (\bar{b}_2^4 \vee \bar{b}_2^5 \vee b_4^2) \\
\wedge & (\bar{b}_4^2 \vee \bar{b}_1^3) \wedge (\bar{b}_3^2 \vee \bar{b}_2^3) \wedge (b_2^2 \vee \bar{b}_3^3)
\end{array}
$$



+5 variables
+25 clauses

26.





$S_{i,j} \rightarrow S_{i+1,j}$
$x_{i+1} \land S_{i,j} \rightarrow S_{i+1,j+1}$

After shifting the rows:

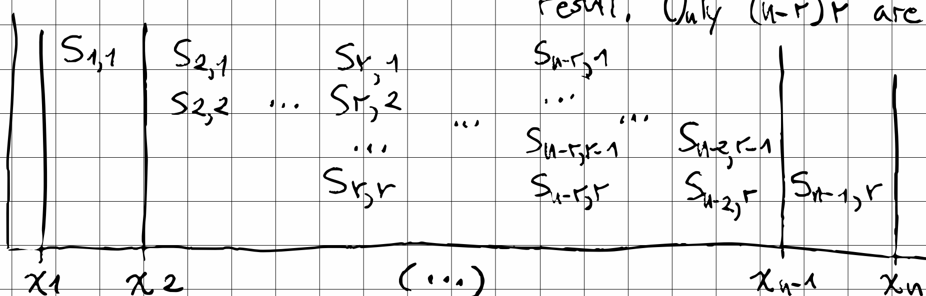
$S_{i,j} \rightarrow S_{i+1,j}$ $(1 \leq i < n-r,\ 1 \leq j \leq r)$ $(n-1)r$ variables naively

$x_{i+j} \land S_{i,j} \rightarrow S_{i,j+1}$
$(1 \leq i \leq n-r,\ 0 \leq j \leq r)$

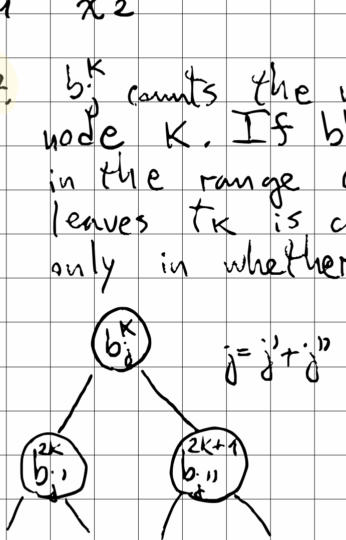
$S_{i,0} = 1$ (first bit in each row)

$S_{i,r+1} = 0$ (overflow bits)

However, note that certain variables are not really needed because they can't affect the result. Only $(n-r)r$ are needed.



27. b_j^k counts the number of active variables under node k. If b_j^k is set, it means there are j variables in the range covered by node k. The number of leaves t_k is clamped to r since we are interested only in whether $r+1$ are set. So $b_j^k \rightarrow [x \text{ in range of } k = j]$.



The boundary conditions are set by asserting that the nodes 2 and 3 do not have $r+1$ leaves set between the two.

28. $S_{\geq 1}(x_1, \dots, x_n) = S_{\leq n-1}(x_1, \dots, x_n)$

Then, if we have a K-SAT set of clauses, we can convert each clause $C_i = (x_{i1}, x_{i2}, \dots, x_{ik})$ into a set of equivalent 3-SAT clauses with $S_{\geq 1}(C_i)$. This works for any function $S_{\geq 1}$ that uses clauses of length at most 3.

217. False: consider $A = B = \{l, \bar{l}\}$. A and B are simultaneously satisfiable by setting l to any value. But $C = (A \cup B) \setminus \{l, \bar{l}\} = \epsilon$ is not satisfiable.

218. $x\,?\,B : A$

224. Add x to every leaf containing a clause originally containing x and propagate x along the tree via resolution on other variables. The final node will be either ϵ as before, or $\{x\}$.

109. We need to find lexicographically smallest solution $x_1 \dots x_n$. We can assume that $x_0 = 1$ for simplicity. Initially, we set $K=0$ and find the largest j such that $K < j$ and there is a solution with $x_{K+1} \dots x_j = 0$. Applying this process repeatedly obtains the required solution. In the worst case, $O(n^2)$ calls to the SAT solver are required.

16. Yes. For example, in the McGregor graph of order 3, the nodes 01, 02, 12 and 32 form a 4-clique. This corner is a 4-clique in any order as well. (eg. 08, 09, 19 and a9 in the order 10 graph) Thus any McGregor graph requires 4 colors.

40. It is unsatisfiable whenever z is prime and the largest index i s.t. $z_i = 1$ is larger than $\max(m,n)$. Otherwise, $x = z$ and $y = 1$ satisfy the instance.

$7 = (111)_2$ is thus the largest prime which is still satisfiable.

42. $x \leftarrow t \oplus u \oplus v$

t	u	v	x		
0	0	0	0	$\bar t \wedge \bar u \wedge \bar v \to \bar x$	$(t \vee u \vee v \vee \bar x)$
0	0	1	1	$\bar t \wedge \bar u \wedge v \to x$	$(t \vee u \vee \bar v \vee x)$
0	1	0	1	$\bar t \wedge u \wedge \bar v \to x$	$(t \vee \bar u \vee v \vee x)$
0	1	1	0	$\bar t \wedge u \wedge v \to \bar x$	$(t \vee \bar u \vee \bar v \vee \bar x)$
1	0	0	1	$t \wedge \bar u \wedge \bar v \to x$	$(\bar t \vee u \vee v \vee x)$
1	0	1	0	$t \wedge \bar u \wedge v \to \bar x$	$(\bar t \vee u \vee \bar v \vee \bar x)$
1	1	0	0	$t \wedge u \wedge \bar v \to \bar x$	$(\bar t \vee \bar u \vee v \vee \bar x)$
1	1	1	1	$t \wedge u \wedge v \to x$	$(\bar t \vee \bar u \vee \bar v \vee x)$

$y \leftarrow \langle tuv \rangle$

t	u	v	y		
0	0	0	0	$\bar t \wedge \bar u \wedge \bar v \to \bar y$	$(t \vee u \vee v \vee \bar y)$
0	0	1	0	$\bar t \wedge \bar u \wedge v \to \bar y$	$(t \vee u \vee \bar v \vee \bar y)$
0	1	0	0	$\bar t \wedge u \wedge \bar v \to \bar y$	$(t \vee \bar u \vee v \vee \bar y)$
0	1	1	1	$\bar t \wedge u \wedge v \to y$	$(t \vee \bar u \vee \bar v \vee y)$
1	0	0	0	$t \wedge \bar u \wedge \bar v \to \bar y$	$(\bar t \vee u \vee v \vee \bar y)$
1	0	1	1	$t \wedge \bar u \wedge v \to y$	$(\bar t \vee u \vee \bar v \vee y)$
1	1	0	1	$t \wedge u \wedge \bar v \to y$	$(\bar t \vee \bar u \vee v \vee y)$
1	1	1	1	$t \wedge u \wedge v \to y$	$(\bar t \vee \bar u \vee \bar v \vee y)$

$t \wedge u \to y$ $\quad (\bar t \vee \bar u \vee y)$ $\quad \bar t \wedge \bar u \to \bar y$ $\quad (t \vee u \vee \bar y)$

$t \wedge v \to y$ $\quad (\bar t \vee \bar v \vee y)$ $\quad \bar t \wedge \bar v \to \bar y$ $\quad (t \vee v \vee \bar y)$

$u \wedge v \to y$ $\quad (\bar u \vee \bar v \vee y)$ $\quad \bar u \wedge \bar v \to \bar y$ $\quad (u \vee v \vee \bar y)$

41. $2 \le m \le n$ $\qquad$ $n+m$ bins

	$\wedge$	$\vee$	$\oplus$		C
half-adder	1	0	1	=	$2 = h$
full-adder	2	1	2	=	$5 = f$

mn $\wedge$s for products $x_i y_j$

$$\text{cost}(b) = \left\lfloor \frac{b-1}{2} \right\rfloor f + ((b+1) \bmod 2)\, h$$

$$\overbrace{0, 1, 2, 3, \dots, \underbrace{m, \dots, m}_{n-m+1}, \dots, 3, 2, 1}^{n+m}$$

$$\text{total} = \sum_{i=2}^{m} \text{cost}(2(i-1)) + (n-m)\,\text{cost}(2m-1) + \text{cost}(2(m-1))$$
$$+ \sum_{i=2}^{m-1} \text{cost}(m-i+m-i+1) + mn$$
$$= (m-2)(m-1)f + (m-1)h + (n-m)(m-1)f + (m-2)f + h + mn$$
$$= f\big((m-2)m + (n-m)(m-1)\big) + hm + mn$$
$$= f(mn - m - n) + hm + mn$$

i			
2	0	=	2
3	1	=	4
4	2	=	6
5	3	=	8
(...)			

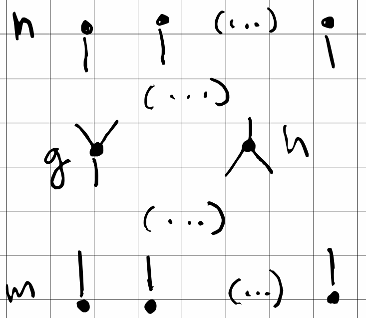
m	$m-2$	=	$2(m-1)$
m	$m-1$	=	$2m-1$
m	$m-1$	=	$2m-1$
m	$m-1$	=	$2m-1$
(...)			

$m-1$	$m-1$	=	$2(m-1)$
$m-2$	$m-1$	=	$2(m-2)+1$
$m-3$	$m-2$	=	$2(m-3)+1$
...			
1	2	=	3

$\text{cost}(2(i-1))$ $\qquad$ $\text{cost}(2m-1)$ $\qquad$ $\text{cost}(2(m-1))$ $\qquad$ $\text{cost}(2(m-i)+1)$

$\sum_{i=2}^{m} (i-2)f + h$ $\qquad$ $(m-1)f$ $\qquad$ $(m-2)f + h$

47. m outputs, n inputs
g 2-to-1 gates
h 1-to-2 gates (fanout)

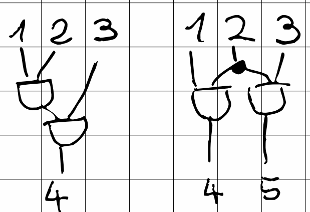


$$\sum_{v \in V} \deg(v) = 2|E|$$

$$|V| = m + n + g + h$$

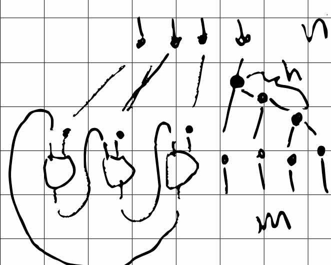
$$n + m + (g+h)3 = 2w \qquad (1)$$

regardless of how gates are interconnected, the number of wires remain constant.



$$g + m - n = h$$

$$w = 2g + m + h \qquad (2)$$



$$n + m + 3(g + g + m - n) = 2w$$
$$n + m + 6g + 3m - 3n = 2w$$
$$4m + 6g - 2n = 2w$$

$$w = 2m - n + 3g \qquad (3)$$

119. $\{123, 234, 345, 456, 567, 678, 789, 135, 246, 357, 468, 579,$
$147, 258, 369, 159,$
$\bar{1}\bar{2}\bar{3}, \bar{2}\bar{3}\bar{4}, \bar{3}\bar{4}\bar{5}, \bar{4}\bar{5}\bar{6}, \bar{5}\bar{6}\bar{7}, \bar{6}\bar{7}\bar{8}, \bar{7}\bar{8}\bar{9}, \bar{1}\bar{3}\bar{5}, \bar{2}\bar{4}\bar{6}, \bar{3}\bar{5}\bar{7}, \bar{4}\bar{6}\bar{8}, \bar{5}\bar{7}\bar{9},$
$\bar{1}\bar{4}\bar{7}, \bar{2}\bar{5}\bar{8}, \bar{3}\bar{6}\bar{9}, \bar{1}\bar{5}\bar{9}\}$

$x=5$ appears 16 times

$F|5 = \{123, 234, 678, 789, 246, 468, 147, 369,$
$\bar{1}\bar{2}\bar{3}, \bar{2}\bar{3}\bar{4}, \bar{3}\bar{4}, \bar{4}\bar{6}, 67, \bar{6}\bar{7}\bar{8}, \bar{7}\bar{8}\bar{9}, \bar{1}\bar{3}, \bar{2}\bar{4}\bar{6}, \bar{3}\bar{7}, \bar{4}\bar{6}\bar{8}, \bar{7}\bar{9},$
$\bar{1}\bar{4}\bar{7}, \bar{2}\bar{8}, \bar{3}\bar{6}\bar{9}, \bar{1}\bar{9}\}$

24 clauses
− 8 clauses
− 8 literals

121. see implementation

122. A2: if $d = n+1$, process current assignment and go to step A6.

A2: do not use move codes for pure literals, since some solutions might involve a literal whose value is redundant.

A2: do not check for $C(l) = a$ and instead always continue to step A3.

123.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
L =	7	9	3	5	7	8	3	5	6	8	3	4	6	8	2	4	6	9	2	4	7

	1	2	3	4	5	6	7
START =	19	16	13	10	7	4	1
LINK =	0	0	0	0	0	0	0

124. see implementation

125. B2: if $d > n$, process current assignment and go to step B6.

126.

312	315	714	915	$\bar1\bar2\bar3$	$\bar1\bar3\bar5$	$\bar1\bar4\bar7$	$\bar1\bar5\bar9$
324	624	825		$\bar2\bar3\bar4$	$\bar2\bar4\bar6$	$\bar2\bar5\bar8$	
345	357	369		$\bar5\bar4\bar3$	$\bar5\bar3\bar7$	$\bar9\bar3\bar6$	
456	648			$\bar5\bar4\bar6$	$\bar6\bar4\bar8$		
657	759			$\bar5\bar6\bar7$	$\bar5\bar7\bar9$		
678				$\bar8\bar6\bar7$			
789				$\bar8\bar7\bar9$			

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	Units	Choice	Change
(6978)	0	0	1	1	0	–	–	–	–	9	9	$\bar6\bar3\bar9$

127. 1 1 4 1 4 5 4 5 $\longrightarrow$ 1 1 4 2

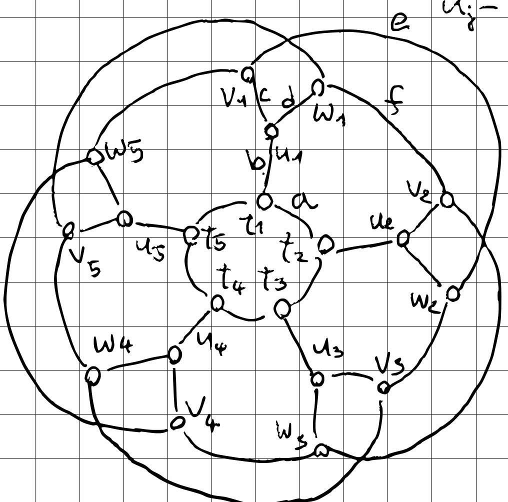
130. See implementation (just select a new watchee for all clauses watching $l = 2k + x_k$, also taking care of adding the new watchees to the active ring if they were not already in it)

190. $(x_1 \vee x_2 \vee x_3 \vee x_4)$

any clause $(x_1' \vee x_2' \vee x_3')$ will be falsified by some satisfying assignment

176. J_q 4q vertices t_j, u_j, v_j, w_j

6q edges $t_j - t_{j+1}$, $t_j - u_j$, $u_j - v_j$, $u_j - w_j$, $v_j - w_{j+1}$, $w_j - v_{j+1}$



a)

$a_j - a_{j+1}$
$a_j - b_j$
$a_j - b_{j+1}$
$b_j - c_j$
$b_j - d_j$
$c_j - d_j$
$c_j - e_j$
$d_j - f_j$
$e_j - d_{j+1}$
$f_j - c_{j+1}$
$e_j - f_{j+1}$
$f_j - e_{j+1}$

b) q even

$\chi(J_q) = 2$

Consider the following 2-coloring, valid for even q:

$c(t_j) = j \bmod 2$
$c(u_j) = (j+1) \bmod 2$
$c(v_j) = j \bmod 2$
$c(w_j) = j \bmod 2$

$\Longrightarrow$

$c(t_j) \neq c(t_{j+1})$
$c(t_j) \neq c(u_j)$
$c(u_j) \neq c(v_j)$
$c(u_j) \neq c(w_j)$

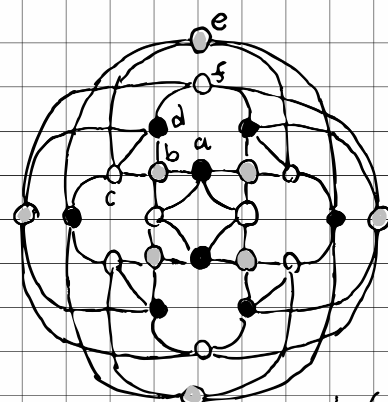
$c(v_j) \neq c(w_{j+1})$
$c(w_j) \neq c(v_{j+1})$

$\chi(L(J_q)) \geq 3$ because $L(J_q)$ has a 3-clique (for example $\{a_{j+1}, a_j, b_j\}$)

Explicit 3-coloring for $L(J_q)$

$c(a_j) = j \bmod 2$
$c(b_j) = 2$
$c(c_j) = j \bmod 2$
$c(d_j) = (j+1) \bmod 2$
$c(e_j) = 2$
$c(f_j) = j \bmod 2$

valid for q even.



$L(J_4)$

c) For J_q, we can modify the coloring from b) by setting $c(t_q) = c(v_q) = c(w_q) = 2$. Thus, $\chi(J_q) = 3$. A 2-coloring is not possible because J_q always contains an odd-length cycle (e.g. $\{t_1, t_2, \dots, t_q, t_1\}$)

- $L(J_q)$ can't be 2-colored because it has an odd-length cycle $(a_1, a_2, \dots, a_q, a_1)$.

- $L(J_q)$ can't be 3-colored because (b_j, c_j, d_j) must be color-wise distinct to (a_j, e_j, f_j), because (b, c, d) is a triangle with each vertex adjacent to (a, e, f), respectively. So, each of these groups can be colored with one of the 3 dearrangements of 3 elements $\{(1,2,3), (2,3,1), (3,1,2)\}$. Additionally, because $a_j - b_j$, the dearrangement used for (a_j, e_j, f_j) must be different to the one used for $(a_{j+1}, e_{j+1}, f_{j+1})$, and since there are q of them and q is odd, this can't be done.

- Finally, $L(J_q)$ can be 4-colored because of Brook's theorem.

219. $C = C' \diamond C''$

(i) $(x \vee A') \diamond (\bar{x} \vee A'') = (A' \vee A'')$
(ii) arbitrary clauses
(iii) $C' \wedge C'' \rightarrow C' \diamond C''$

For arbitrary clauses C' and C'', we can define

$$C = C' \diamond C''$$

where $C = \begin{cases} \exists x,\ x \in C',\ \bar{x} \in C'' & C = (C' \setminus \{x\}) \vee (C'' \setminus \{\bar{x}\}) \\ \wp \text{ otherwise} \end{cases}$

$C' \wedge C'' \rightarrow C' \diamond C''$ because a satisfying assignment for $C' \wedge C''$ satisfies either $(C' \setminus \{x\})$ or $(C'' \setminus \{x\})$, and if there is no such x, then $\wp$ is satisfied by any assignment.