alg2

ALG2 Assignment B - Quiz

1. Convert the following Boolean formulas into a CNF:

a. $((a \wedge b) \vee (e \wedge c \wedge d))$

$$(a \wedge b) \vee (e \wedge c \wedge d)$$

$$((a \wedge b) \vee e) \wedge ((a \wedge b) \vee c) \wedge ((a \wedge b) \vee d)$$

$$(a \vee e) \wedge (b \vee e) \wedge (a \vee c) \wedge (b \vee c) \wedge (a \vee d) \wedge (b \vee d)$$

Result:

$$(a \vee e) \wedge (b \vee e) \wedge (a \vee c) \wedge (b \vee c) \wedge (a \vee d) \wedge (b \vee d)$$

b. $(((a \vee x) \wedge (b \vee y)) \vee (c \wedge d))$

$$(((a \vee x) \wedge (b \vee y)) \vee (c \wedge d))$$

$$(((a \vee x) \vee (c \wedge d)) \wedge ((b \vee y) \vee (c \wedge d)))$$

$$((a \vee x \vee c) \wedge (a \vee x \vee d)) \wedge ((b \vee y \vee c) \wedge (b \vee y \vee d))$$

Result:

$$(a \vee x \vee c) \wedge (a \vee x \vee d) \wedge (b \vee y \vee c) \wedge (b \vee y \vee d)$$

c. $(a \vee (b \wedge (c \vee (d \wedge e))))$

$$a \vee (b \wedge (c \vee (d \wedge e)))$$

$$a \vee (b \wedge ((c \vee d) \wedge (c \vee e)))$$

$$a \vee ((b \wedge (c \vee d)) \wedge (b \wedge (c \vee e)))$$

$$(a \vee (b \wedge (c \vee d))) \wedge (a \vee (b \wedge (c \vee e)))$$

$$(a \vee b) \wedge (a \vee c) \wedge (a \vee d) \wedge (a \vee b) \wedge (a \vee c) \wedge (a \vee e)$$

Result:

$$(a \vee b) \wedge (a \vee c) \wedge (a \vee d) \wedge (a \vee e)$$

d. $(\neg(a \vee \neg b) \vee (\neg c \Rightarrow d))$

$$(\neg(a \vee \neg b) \vee (\neg c \Rightarrow d))$$

$$(\neg a \wedge b) \vee (c \vee d)$$

$$(\neg a \vee \neg c \vee d) \wedge (b \vee \neg c \vee d)$$

Result:

$$(\neg a \lor \neg c \lor d) \land (b \lor \neg c \lor d)$$

e. $(a \Leftrightarrow ((b \land c) \Rightarrow \neg d))$

$$a \Leftrightarrow ((b \land c) \Rightarrow \neg d)$$

$$a \Leftrightarrow (\neg(b \land c) \lor \neg d)$$

$$a \Leftrightarrow (\neg b \lor \neg c \lor \neg d)$$

$$(a \land (\neg b \lor \neg c \lor \neg d)) \lor (\neg a \land \neg(\neg b \lor \neg c \lor \neg d))$$

$$(a \land (\neg b \lor \neg c \lor \neg d)) \lor (\neg a \land (b \land c \land d))$$

Result:

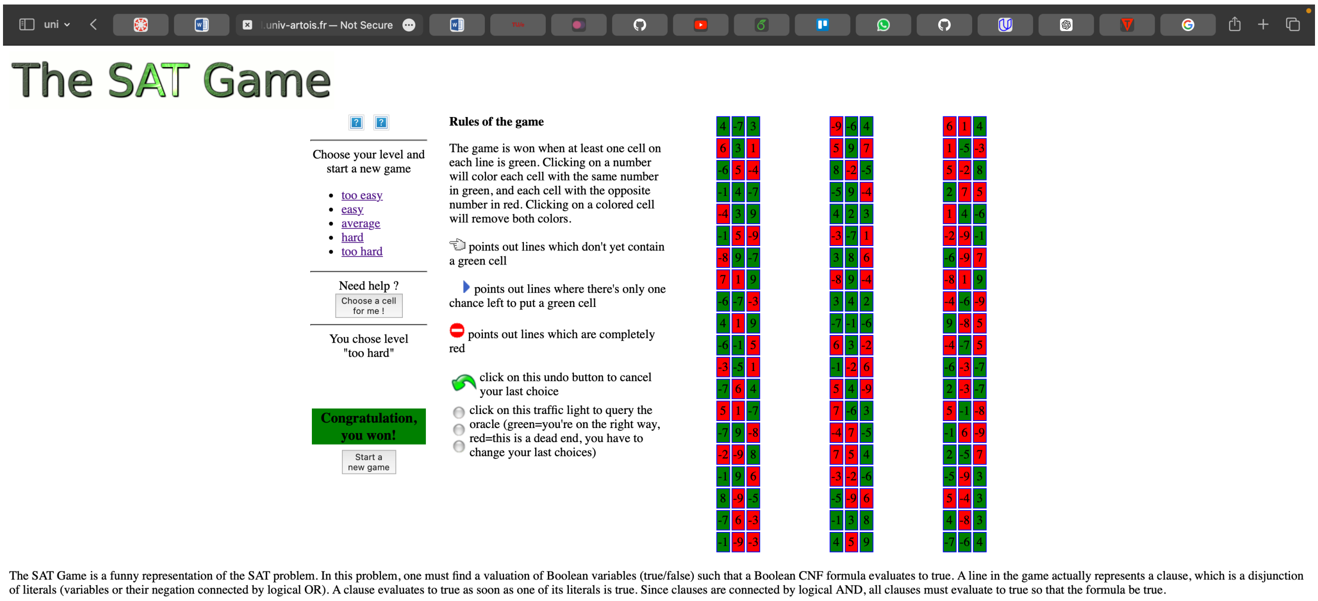
$$(a \lor \neg b \lor \neg c \lor \neg d) \land (\neg a \lor b) \land (\neg a \lor c) \land (\neg a \lor d)$$

2. Convert the following Boolean formula into 3CNF:

$$\neg a \land (a \lor \neg e \lor f) \land (b \lor \neg c)$$

The given formula already meets the requirement of 3 literals or fewer per clause. Thus, it is already in 3-CNF.

3. Solve a 3CNF from [SAT Game]



4. Reducing a problem X to a problem Y in polynomial time implies:

b. Y is at least as hard to solve as X

d. If Y can be solved in polynomial time, then so can X

5. Which statement(s) are true:

b. If there is a polynomial time algorithm for NP-complete problem(s) then there are polynomial time algorithms for all problems in NP

c. If Vertex cover is an NP-complete problem, then so is Clique

6. Now that we know that SAT is an NP-complete problem how could we show that vertex cover, independent set and clique problems are NP-complete?

a. Show that any input for SAT can be transformed into an input for one of these problems

7. Is the following Boolean formula satisfied:

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

From this formula second clause, it is clear that x_1 should be definitely true, otherwise the formula is not satisfied. Then any x_2 or x_3 should be true, then formula is satisfied

Answer: The formula can be satisfied. For example, if $x1 = true$, $x2 = true$, and $x3 = false$, all clauses will be true.

8. When we talk about running time of a SAT problem which of the following parameters should be taken into account:

a. the length of the (input) formula

9. We are considering the following Boolean formula with 6 variables:

b. It has several satisfying assignments

c. it can only be satisfied if exactly one variable is assigned to true

10. When an algorithm is executed on a non-deterministic RAM a configuration (a snapshot) of the RAM is determined (there might be more correct answers):

e. by the input, the current step of the algorithm and the content of the read/write memory

11. A polynomial time algorithm is run on a non-deterministic RAM on an input with length n. How much read/write memory is needed:

d. constant

12. A polynomial time algorithm is run on a non-deterministic RAM on an input with length n. How much algorithms’ steps are executed in total before it terminates:

b. polynomial in n

13. A polynomial time algorithm is run on a non-deterministic RAM on an input with length n. At every single moment of execution the non-deterministic RAM is in a certain configuration (snapshot). Which of the following is true (more answers may be correct):

a. each configuration is in size polynomial in n

b. there can be exponential numbers of configurations

14. The execution of a polynomial time algorithm on a non-deterministic RAM on input of size n is represented as a Boolean formula according to the method used in the proof of the Cook-Levin theorem. Which properties does the Boolean formula have (there might be more correct answers):

a. Its size is polynomial in n

c. every satisfying assignment presents a potential configuration of the non-deterministic RAM during the execution

15. If you can show that SAT is solvable in polynomial time on a deterministic RAM then you:

a. have proved P=NP

16. If we can show that one NP-complete problem is solvable in polynomial time on a deterministic RAM then we (there might be more correct answers):

a. Some problems in NP can be solved in polynomial time

b. All NP-complete problems are solvable in polynomial time

c. All problems in NP are solvable in polynomial time

Actually C implies all other answers

17. What happens if you show that one NP-complete problem is only solvable in exponential time (there might be more correct answers):

a. Some problems in NP are only solvable in exponential time

b. All NP-complete problems are solvable in exponential time

18. Write the proof of the following claim: For a graph $G = (V, E)$ and a set $S \subseteq V$, S is a clique of G if and only if S is an independent set of $\overline{G}$.

There exist Polynomial time algorithm reduction From Clique(X) to Independent Set(Y), Such that it transform Input X to input Y.
Solving problem Y on transformed result will have the same answer as Solving problem X on original Input.

So S is a clique of G if and only if S is an independent set of $\overline{G}$.