CT.1 (10) Examine whether the following restricted versions of SAT are tractable.

- **Not-All-Equal SAT, NAESAT**: The input formula F is a CNF, and the question is whether F can be satisfied by some assignment σ to the variables such that in no clause of F all literals are true.
  Example: $F = (x_1 \vee x_2) \wedge (x_2 \vee \neg x_3)$; take e.g. $\sigma(x_1) = 1$, $\sigma(x_2) = 0$, $\sigma(x_3) = 0$.

- **Read-**2 **SAT**: The input formula F is a CNF, and each variable occurs only at most 2 times in F.
- **All-Resolve-SAT**: The input formula F is a CNF, and each pair of distinct clauses c and c' of F resolves, i.e., there is a variable x such that $c = x \vee \alpha$ and $c' = \neg x \vee \beta$ or the other way round ($c = \neg x \vee \alpha$ and $c' = x \vee \beta$)
  Examples: $F = x \wedge \neg x$; $G = (x \vee y) \wedge (\neg x \vee z) \wedge (\neg z \vee \neg y)$

Bonus: determine whether **Read-**k **SAT**, where $k > 1$ is fixed, is tractable, for each k.

Solution:

- **Not-All-Equal SAT, NAESAT**: The input formula F is a CNF, and the question is whether F can be satisfied by some assignment σ to the variables such that in no clause of F all literals are true.
  Example: $F = (x_1 \vee x_2) \wedge (x_2 \vee \neg x_3)$; take e.g. $\sigma(x_1) = 1$, $\sigma(x_2) = 0$, $\sigma(x_3) = 0$.

  This problem is not-tractable, in fact it is $\mathbf{NP}$-hard. To support this claim a polynomial reduction from **CNF** to **NAESAT** will be given. Let φ be a **CNF** instance, i.e. a formula in CNF. Consider the following translation.
  Let t and f be two new propositional variables, let $\varphi := (p_{11} \vee \cdots \vee p_{1n_1}) \wedge \cdots \wedge (p_{m1} \vee \cdots \vee p_{mn_m})$ be a formula in CNF with p_{ij} being literals. Let $l_{i1}, \ldots, l_{in_i}, l'_{i1}, \ldots, l'_{in_i}$ be literals for $i \in \{1, \ldots m\}$ and let τ be the following transformation

  $$\tau(\varphi) := (t \vee f) \wedge \bigwedge_{i=1}^{m} (l'_{i1} \vee \cdots \vee l'_{in_i} \vee t) \wedge \bigwedge_{i=1}^{m} (l_{i1} \vee \cdots \vee l_{in_i}, \vee f)$$

  such that $l_{ij} := p_{ij}[x_k/y_k]$ and $l'_{ij} := p_{ij}[x_k/y'_k]$, with $x_k \in Var(\varphi)$ and y_k, y'_k being fresh and distinct new variables, i.e. $y_k, y'_k \in Var(\tau(\varphi))$. That is, using φ we simply build two copies φ_t and φ_f of φ where each copy has its own set of variables and each clause in the prior is extended by the variable t and each in the latter by f.
  What remains to show is that $\chi_{\mathbf{CNF}}(\varphi) = 1$ if and only if $\chi_{\mathbf{NAESAT}}(\tau(\varphi)) = 1$.

  ”⇒:” Assume that $\chi_{\mathbf{CNF}}(\varphi) = 1$, meaning that φ is in CNF and has an satisfying assignment σ. Let σ_τ be the truth assignment for $\tau(\varphi)$, which shall be constructed based on the assignment σ. Firstly, φ_f is merely a copy of φ, which clauses are extended by f. Hence, let $\sigma_\tau(y_k) = \sigma(x_k)$. Since, σ is a satisfying assignment at least one literal in each clause is satisfied. Moreover, let $\sigma_\tau(f) = 0$, thus it follows that every clause in φ_f has one literal that evaluates to 1 and one that evaluates to 0 under σ_τ. Now, since $(t \vee f)$ and $\sigma_\tau(f) = 0$

it follows that $\sigma_\tau(t) = 1$. Hence, every clause in φ_t is now satisfied, thus for all y'_k let $\sigma_\tau(y'_k) = 0$. Therefore, in each clause of φ_t there exists one literal that evaluates to 1 and one that evaluates to 0. A satisfying assignment of $\tau(\varphi)$ for the problem **NAESAT** is found, and thus $\chi_{\mathbf{CNF}}(\tau(\varphi)) = 1$.
"$\Leftarrow$": Assume that $\chi_{\mathbf{CNF}}(\tau(\varphi)) = 1$, meaning that there exists an satisfying assignment σ_τ such that not all literals are true. That is, either $\sigma_\tau(t) = 1$ or $\sigma_\tau(f) = 1$. Without loss of generality assume the prior. Implying that $\sigma_\tau(f) = 0$. However, φ_f is satisfied under σ_τ, thus for each clause C_i in φ_f there must be a literal $l_ij \neq f$ that is satisfied under σ_τ. Now, given the fact that φ_f is a copy of φ with every clause being extended by f and the fact that $\sigma_\tau(f) = 0$ it follows that $\sigma(x_k) = \sigma_\tau(y_k)$ is a satisfying assignment of φ. Resulting in $\chi_{\mathbf{CNF}}(\varphi) = 1$.
Clearly, if this is the ψ_f is satisfied, without assigning f to true. Meaning that for ψ_f there exists an assignment that satisfies every clause, which can only be the case if in each clause, there exists at least one literal that evaluates to true.

- **Read-**2 **SAT**: The input formula F is a CNF, and each variable occurs only at most 2 times in F.

  This variant is tractable. To show this statement, let φ be a CNF-formula, where every variable occurs at most twice. Now consider the following algorithm.

```
Read-2-SAT(φ):
    Γ := Cl(φ)
    while Γ ≠ ∅:
      if ∃C,C' ∈ Γ C = (x) ∧ C' = (¬x):   // (0)
         return False
      Γ' := ∅:
      while Γ ≠ Γ':  // (1)
         Γ := Γ'
         Γ' := remove_tautological_clauses(Γ) // (1.1)
         Γ' := remove_clauses_with_monotone_variables(Γ') // (1.2)
      C := choose_random_clause(Γ)
      for C' ∈ Γ:  // (2)
         if C = (l ∨ α) and C' = (l̄ ∨ β) // (2.1)
            C_n := (α ∨ β) // α or β could be empty (not both -> (0))
            Γ := (Γ \ {C,C'}) ∪ {C_n}
            break // exit for-loop
    return True
```

  (0) If there are two complementary unit clauses, there can not be an assignment satisfying the formula.
  (1.1) A variable is monotone if it only occurs either positively or negatively. Hence, one can safely satisfy the all clauses in which this variable occurs by simply choosing an assignment corresponding to the polarity of the occurrence, i.e. if x is negative monotone then choose σ such that $\sigma(x) = 0$ and vice verse. This choice can not lead to another clause being no longer satisfiable, i.e. no satisfying assignment relies on the fact that the such a monotone literal is not satisfied. Hence, this step preserves satisfiability.

(1.2) A clause is tautological if it has the form $(x \vee \neg x \vee \alpha)$. Because any assignment satisfies this clause,they can be removed safely as well, i.e. removing this clause preserves satisfiability.

Hence, after (1) Γ exhibits the following properties.

- *No clause occurs twice.*
  If they would the variables within must be monotone and can contain a variable twice, thus those clauses were removed in step (1.1).
- *Every variable occurs once as positive literal and once as negative literal in two different clauses.*
  Otherwise those variables are monotone and were removed in step (1.1) or they would be part of a tautological clause.

The whole step (1) can be done in polynomial time. Firstly, the number of iterations in the while loop is clearly bounded by the number of clauses in Γ, i.e. clauses are removed until it is no longer possible. Secondly, (1.1) runs in polynomial time as well, i.e. for each variable run through every clause, if it is present in said clause save the polarity of its occurrences and compare the two values for monotonicity. Thirdly, running through all clauses and checking if they contain complementary literals is also polynomial wrt. to the input. Lastly, with every step preserving satisfiability Γ after (1) is satisfiable if and only if Γ before (1) is satisfiable.

Observation 1: Let $C = (x \vee \alpha)$ and $C' = (\neg x \vee \beta)$ where α and β are some disjunction of literals not containing x or $\neg x$. Then $\{C, C'\}$ is satisfiable if and only if $\{(\alpha \vee \beta)\}$ is satisfiable.

" $\Rightarrow$ ": $\{C, C'\}$ is satisfiable. Therefore, either α or β has to be satisfied, i.e. x and $\neg x$ can not both be true. Hence, any assignment σ satisfying $\{C, C'\}$ must also satisfy $\{(\alpha \vee \beta)\}$. " $\Leftarrow$ ": $\{(\alpha \vee \beta)\}$ is satisfiable. Hence, for every assignment σ either α or β is satisfied. If α is satisfied under σ, then C is satisfied, without relying on the truth assignment of x. Hence, σ can be extended by the assignment $\{x \mapsto 0\}$ in order to satisfy C'. Analogue for the other case.

As for step (2). Choosing a clause $C = (l \vee \alpha)$ at random, due to (0) and (1) one knows that there must be a clause $C' = (\bar{l} \vee \beta)$, different from C in Γ containing the complement of the literal l, i.e. $\bar{l}$. W.l.o.g. let $l = x$. After the respective clause is found, the next step (2.1) is basically a resolution step. From Observation 1 it follows that Γ is satisfiable if and only if $(\Gamma \setminus \{C, C'\}) \cup \{(\alpha \vee \beta)\}$ is satisfiable. Moreover, since each variable occurs twice, step (2.1) eliminates all occurrences of x. Furthermore, with the promise from (0) that there are no two complementary unit clauses, this step can not result in the empty clause. Hence, resulting set is an instance of the problem **Read-**2 **SAT** that is satisfiable, if and only of the original instance was satisfiable. However, the only difference is this instance is shorter by at least one variable. Therefore, the whole proceeder is linearly bounded by the number of variables in φ. Therefore, after at most $|Var(\varphi)|$ steps, this algorithm will terminate, if it does not return false during its run time, the resulting empty set of clauses is equi-satisfiable to the original set of clauses.

- **All-Resolve-SAT**: The input formula F is a CNF, and each pair of distinct clauses c and c' of F resolves, i.e., there is a variable x such that $c = x \vee \alpha$ and $c' = \neg x \vee \beta$ or the other

way round ($c = \neg x \vee \alpha$ and $c' = x \vee \beta$)
Examples: $F = x \wedge \neg x$; $G = (x \vee y) \wedge (\neg x \vee z) \wedge (\neg z \vee \neg y)$

Observation 1: Consider a set of clauses Γ over m variables, such that each $C \in \Gamma$ contains m unique variables, i.e. $|Var(C)| = m$. Γ is satisfiable if and only if $|\Gamma| < 2^m$, i.e. if there are strictly less than 2^m unique clauses in Γ.

Firstly, there can be at most 2^m unique clauses. That is, with every clause having m unique variables it follows that $Var(\Gamma) = Var(C)$ for each C. Now since every variable can occur either positively or negatively there are 2^m combinations, and thus 2^m clauses. Secondly, clearly if $|\Gamma| = 2^m$ Γ is unsatisfiable. That is, for each clause $C \in \Gamma$ there exists a clause $\overline{C} \in \Gamma$ where for every literal in C, there exists its complement in $\overline{C}$.

Thirdly, consider the case $|\Gamma| < 2^m$. Hence, there exists a clause $C \in \Gamma$ which does not have a complement $\overline{C} \in \Gamma$. Consider the assignment σ_C induced by C, i.e. $\sigma_C(x) = 1$ iff x is a positive literal in C. Assume that there exist a clause $C' \in \Gamma$ such that C' is not satisfied under σ_C. This would imply that for every positive literal of C, C' contains its negative counterpart. Analogue for the negative literals in C. However, this would mean that C' is actually $\overline{C}$, which clearly is impossible.

Observation 2: Let C be a clause and let x be a new variable not contained in C. C is equivalent to $\{C \vee x, C \vee \neg x\}$.

Clearly, any model satisfying C satisfies $C \vee x$ and $C \vee \neg x$. On the other hand any model satisfying $C \vee x$ and $C \vee \neg x$ must satisfy C, as no model can satisfy x and $\neg x$ at the same time.

Let $\mathcal{E}_x(\Gamma) = \bigcup_{C \in \Gamma}\{C \vee x, C \vee \neg x\}$ for some set of clauses Γ. As syntactic sugar let $\mathcal{E}_x(C) = \mathcal{E}_x(\{C\})$. Moreover, let $\mathcal{E}_V(C)$ be the set of clauses computed by applying the above extension presented in Observation 2 until every clause in the resulting set contains all variables in V. That is,

$$\begin{aligned} &\mathcal{E}_\emptyset(\Gamma) = \Gamma \\ &\mathcal{E}_V(\Gamma) = \mathcal{E}_x(\mathcal{E}_{V \setminus \{x\}}(\Gamma)) \quad \text{for some } x \notin Var(\Gamma) \text{ and } x \in V \end{aligned}$$

Observation 3: Let V be a set of variables, s.t. $|V| = m$. Let Γ be a set of clauses over V and let $C \in \Gamma$ such that $|Var(C)| = n \leq m$ (unique variables) then $|\mathcal{E}_V(C)| = 2^{m-n}$.

For a clause with n distinct variables applying $\mathcal{E}_V$ for $|V| = m$, will result $m - n$ applications of $\mathcal{E}_x$. At each iteration the input set is doubled, i.e. every clause in the input set is replaced by two clauses. Therefore, since starting with a clause set of size one, one obtains 2^{m-n} clauses in the final set.

Observation 4: Let Γ be a set of unique clauses without duplicate variables over the set of variables V, where $\forall C, C' \in \Gamma$ C and C' resolve. Then $\forall C, C' \in \Gamma$ $C \neq C' \Rightarrow \mathcal{E}_V(C) \cap \mathcal{E}_V(C') = \emptyset$

Assume that there are two distinct clauses C and C' in Γ such that $\mathcal{E}_V(C) \cap \mathcal{E}_V(C') \neq \emptyset$. Hence, $D \in \mathcal{E}_V(C) \cap \mathcal{E}_V(C')$ exists. It is known that there exists a variable $x \in Var(C) \cap Var(C')$ such that, w.l.o.g., $x \in Lit(C)$ and $\neg x \in Lit(C')$. The operator $\mathcal{E}_V$

produces only clauses that are extensions of the original clauses without adding already existing variables. Hence, $\forall C_e \in Var(C)\ x \in Lit(C_e)$ and $\forall C'_e \in Var(C')\ \neg x \in Lit(C'_e)$. Thus, forcing the conclusion that D can not exist.

Observation 5: Let Γ be a set of unique clauses without duplicate variables over the set of variables V, where $\forall C, C' \in \Gamma$ C and C' resolve. Then Γ is satisfiable if and only if $\sum_{C\in\Gamma} |\mathcal{E}_V(C)| < 2^{|V|}$.

By assumption, all clauses in Γ are unique and resolve pairwise, while having no duplicate variables. Hence, by Observation 4, it follows that $|\bigcup_{C\in\Gamma} \mathcal{E}_V(C)| = \sum_{C\in\Gamma} |\mathcal{E}_V(C)|$. Now, by Observation 1, one knows that $\mathcal{E}_V(\Gamma)$ is satisfiable if and only if $|\mathcal{E}_V(\Gamma)| < 2^{|V|}$. Lastly, from Observation 2 it is known that $\mathcal{E}_V$ preserves validity. Therefore, $\mathcal{E}_V(\Gamma)$ is satisfiable iff Γ is satisfiable.

Observation 6: Let Γ be a set of clauses over the set of variables V, where $\forall C, C' \in \Gamma$ C and C' resolve. Then Γ testing satisfiability can be done in quadratic (probably linear) time wrt. to the combined length of the clauses in Γ.

Firstly, iterate over all clauses and check if they contain duplicate variables, If so remove all duplicates. This operation is clearly save. Secondly, iterate over all clauses and check for duplicates. Both operations can be done in linear time (or at least in quadratic time, if one is not smart about it). The resulting Γ' is now a pairwise resolving set of unique clauses without duplicate variables over the set of variables V By Observation 5 it is known that Γ' is satisfiable if and only if $\sum_{C\in\Gamma} |\mathcal{E}_V(C)| < 2^{|V|}$. By Observation 3, one has the equality $|\mathcal{E}_V(C)| = 2^{|V|-|Var(C)|}$. Hence, one obtains $\sum_{C\in\Gamma} |\mathcal{E}_V(C)| = \sum_{C\in\Gamma} |2^{|V|-|Var(C)|}| < 2^{|V|}$. And since this sum only requires the length of the original clause and the exponent can also be computed in linear time one only has to compute the given some to decide satisfiability of Γ.

- Bonus: determine whether **Read-k SAT**, where $k > 1$ is fixed, is tractable, for each k.

  For $k = 2$ see the actual exercise. Hence, consider an arbitrary $k > 2$. Any of those **Read-k SAT** is not tractable, which can be shown through the reductions **CNF** $\leq_m^P$ **Read-3 SAT** and **Read-3 SAT** $\leq_m^P$ **Read-k SAT**. Starting with the prior. Let φ be some CNF-formula, where $Var(\varphi) := \{x_1, \dots, x_n\}$ and where $Cl(\varphi) := \{C_1, \dots C_m\}$ such that no variable occurs twice in the same clause. This assumption is sound, because

  - if a variable occurs multiple times in the same clause with the same polarity, i.e. either only positively or only negatively, all but one of those occurrences can be removed;
  - if a variable occurs both positively and negatively in a clause this clause can be removed;

  and checking this can clearly be done in polynomial time. Moreover, let x_{ij} for $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$ be fresh variables. Now consider τ

$$\begin{aligned}\tau(\varphi) :=& \psi = \psi_c \wedge \psi_e \\ =& \bigwedge_{j\in\{1,\dots,m\}} C[x_1/x_{1j}, \dots, x_n/x_{nj}] \wedge \\ & \bigwedge_{i\in\{1,\dots,n\}} \big((\neg x_i \vee x_{i1}) \wedge \bigwedge_{j\in\{1,\dots,m-1\}} (x_{ij} \vee \neg x_{i(j+1)}) \wedge (\neg x_{im} \vee x_i)\big)\end{aligned}$$

Firstly, the ψ_e forces for each variable x_i that the following circle of implications is uphold $x_i \Rightarrow x_{i1} \Rightarrow x_{i2} \Rightarrow \ldots x_{im} \Rightarrow x_i$, i.e. it expresses equality among all x_{ij} for all $j \in \{1, \ldots, m\}$ and with x_i. Secondly, each x_{ij} occurs at most once in $Var(\psi)$, since x_i occurs at most once in the clause C_j. Moreover, x_{ij} occurs clearly exactly twice in ψ_e. Therefore, each variable occurs at most three times in ψ. Thirdly, ψ_c is the same size as φ and ψ_e is linear with respect to the number of clauses times the number of variables in φ. Hence, the transformation can be done in polynomial time.

Lastly, it has to be demonstrated that $\varphi\ sat. \iff \psi\ sat.$.
$" \Rightarrow "$. If φ is satisfiable, then there exists an assignment σ under which φ is satisfiable. Let σ_τ be an extension of σ such that $\sigma_\tau := \sigma \cup \{x_{ij} \mapsto \sigma(x_i) \mid \forall i \in \{1, \ldots, n\} \forall j \in \{1, \ldots, m\}\}$. Meaning that under this interpretation x_i is replaced in ψ_c with an object of the same truth value. Hence, ψ_c is satisfied under σ_τ. Moreover, with ψ_e encoding a series of equivalences between x_{ij} and x_i for some fixed i, this assignment, ensuring equal evaluation of exactly those variables, must therefore satisfy ψ_e as well.
$" \Leftarrow "$. if ψ is satisfiable, then there exists an assignment σ under which ψ is satisfiable. Under this assignment every variable in the set $\{x_i\} \cup \{x_{ij} \mid \forall j \in \{1, \ldots, m\}\}$ for any fixed i must be evaluated the same. Otherwise, ψ_e would not have been satisfied. Hence, ψ_c is satisfied by an assignment where every instance of x_i has the same truth value. Therefore, by restricting σ to the variables in φ results in an satisfying assignment for φ.
The reduction **Read-**3 **SAT** $\leq_m^P$ **Read-**k **SAT** is trivial. Meaning, any instance of **Read-**3 **SAT** of **Read-**k **SAT**. Hence, if **Read-**k **SAT** would be tractable, then **Read-**3 **SAT** must be as well, which by transitivity would mean that **CNF** would can be solved in deterministic polynomial time.

CT.5 (10) Consider the following variant of **SAT**, called **Weighted SAT**: the instance consists of a (propositional) CNF $F = \bigwedge_{i=1}^{m} c_i$, integer weights $w(c_i) > 0$ for all clauses $i = 1, \ldots, m$.

A *maximum weight assignment* is a truth assignment σ to the variables in F such that the total weight of clauses satisfied by σ, i.e., $w(\sigma) = \sum_{i=1}^{k} w(c_i) * \sigma(c_i)$, is maximum.

(a) Show that, given an integer $k \geq 0$, deciding whether $w(\sigma) \geq k$ for some maximum weight assignment σ is **NP**-complete.

(b) Show that, deciding whether for any maximum weight assignment σ, the number $w(\sigma)$ is even is Δ_2^P-complete.

Bonus: Given an integer $k \geq 0$, what is the complexity of deciding whether $w(\sigma) \leq k$ for every maximum weight assignment σ ?

Solution:

(a) Show that, given an integer $k \geq 0$, deciding whether $w(\sigma) \geq k$ for some maximum weight assignment σ is **NP**-complete.

Let the problem described above be called k-**WSAT** First hardness will be shown by giving a polynomial time reduction from **CNF** to k-**WSAT**. Hence, by giving such a reduction it follows that k-**WSAT** has to be at least **NP**-hard. Otherwise, **CNF** would not be **NP**-hard. To establish **CNF** $\leq_m^P$ k-**WSAT**, it has to be shown that there exists function τ that is computable in polynomial time wrt. to its input size such that for all inputs x, $\chi_{\textbf{CNF}}(x) = \chi_{k\text{-}\textbf{WSAT}}(\tau(x))$ holds.

Firstly, let $Cl(\varphi)$ be the set of all clauses of an CNF-formula φ. Now, consider an input formula φ for **CNF** and the following transformation:

$$\tau(\varphi) := (\varphi, w, k)$$

where

- $w(c) = 1$ for all clauses $c \in Cl(\varphi)$;
- $k = |Cl(\varphi)|$.

Assume that φ is satisfiable, i.e. $\chi_{\textbf{CNF}}(\varphi) = 1$. Hence, all of its clauses are satisfied by the truth assignment σ. Considering the weight function w obtained through τ one obtains $w(\sigma) = \sum_{c \in Cl(\varphi)} w(c) * \sigma(c) = \sum_{c \in Cl(\varphi)} 1 * \sigma(c) = |Cl(\varphi)|$. That is, all clauses are satisfied and all clauses have the weight 1. Clearly, σ is a maximum weight assignment, in fact by construction any other truth assignment satisfying φ has the same weight as σ. Lastly, since $k = |Cl(\varphi)|$ it follows that $\chi_{k\text{-}\textbf{WSAT}}(\tau(\varphi)) = 1$.

Assume that φ is not satisfiable, i.e. $\chi_{\textbf{CNF}}(\varphi) = 0$. Meaning that there does not exist a truth assignment which satisfies all clauses in φ. That is, for all truth assignments σ it follows that $w(\sigma) = \sum_{c \in Cl(\varphi)} w(c) * \sigma(c) = \sum_{c \in Cl(\varphi)} 1 * \sigma(c) < |Cl(\varphi)|$. Hence, no maximum weight assignment of weight greater or equal to k exists, thus one obtains $\chi_{k\text{-}\textbf{WSAT}}(\tau(\varphi)) = 0$.

Lastly, it has to be established that τ can be computed in polynomial time. Since, w is constant it can be constructed on constant time. Hence, the only computationally demanding task in the transformation is to count the number of clauses in φ. However, this can clearly be accomplished on linear time wrt. to the number of clauses and since those are bound by the size of the formula, one can conclude that τ can be computed in polynomial time. Hence, completing the reduction.

Having established $\mathbf{NP}$-hardness, it remains to show $\mathbf{NP}$-membership. To do so a simple guess and check algorithm is sketched.

```
guess(σ); /* guess a truth assignment for φ*/
if σ satisfies φ and w(σ) ≥ k then:
     succeed;
else:
     fail;
```

This algorithm returns the required results due to the fact that as soon as one has found an assignment with weight greater or equal k, it is clear that any maximal assignment σ_{max} has to have a weight greater or equal to k as well. Moreover, this algorithm employs the guess and check method introduced in the lecture. That is, first one guesses an assignment and then check whether this assignment satisfies the formula φ. As established in the lecture, this checking can be done in polynomial time. Similarly, summing up the weights can also be done in polynomial time. Hence, one can conclude that this algorithm establishes $\mathbf{NP}$-membership. Together with the previous result $\mathbf{NP}$-completeness follows in kind.

(b) Show that, deciding whether for any maximum weight assignment σ, the number $w(\sigma)$ is even is Δ_2^P-complete.

Let the problem described above be called **e-WSAT** Similarly, as before Δ_2^P-hardness will be established first. To that end a reduction from **MSA** to **e-WSAT** , i.e. **MSA** $\leq_m^P$ **e-WSAT** is presented. Hence, consider the following transformation τ where

$$\tau(\pi(\chi), (x_1, \ldots, x_n)) := (\psi, w)$$

where $\varphi := \pi(\chi)$ is an equi-satisfiable CNF-version of the input formula χ, i.e. π is some standard CNF transformation (e.g. Tseytin transformation), and $(x_1, \ldots, x_n)$ is a vector indicating an ordering over the set of variables in χ, i.e. $\mathit{Var}(\chi)$. Moreover, for $n := |\mathit{Var}(\chi)|$

$$\begin{aligned}
\psi :=& \psi_c \wedge \psi_x \wedge \psi_p = \Big(\bigwedge_{C \in \mathit{Cl}(\varphi)} (\neg p \vee C) \Big) \wedge \Big(\bigwedge_{i \in \{1,\ldots,n\}} x_i \Big) \wedge p \\
w :=& \{(p) \mapsto 2^{n+1} + 1\} \cup \{(\neg p \vee C) \mapsto 2^{n+2} \mid \forall C \in \mathit{Cl}(\varphi)\} \\
& \cup \{(x_i) \mapsto 2^{(n-i)} \mid \forall i \in \{1, \ldots, n\}\}
\end{aligned}$$

Since π is known to be polynomial and the number of additional clauses is linear with respect to the number of variables in χ, the whole transformation can be done in polynomial time. (Note: In an actual implementation one does not need to compute and store the exponents 2^i, that is it suffices to store and compare the number i).

Fact 1: $2^n > \sum_{i=0}^{n-1} 2^i$.

Observation 2: If there exists a satisfying assignment of the formula φ, then p is satisfied under every weight maximal assignment.
Consider an weight maximal assignment σ where $\sigma(p) = 0$. This assignment clearly satisfies every clause in ψ_c without relying on the variables $x_1, \ldots, x_n$. Hence, in order for this assignment to be maximal all clauses in ψ_x are satisfied. That is, the only clause not satisfied is (p). However, $w(p) > \sum_{C \in Cl(\psi_x)} w(x)$. Meaning that as long as the same clauses in ψ_c are satisfied, any assignment σ' such that $\sigma'(p) = 1$ has a greater weight that σ. Now, given the assumption that φ is satisfiable, all clauses in ψ_c can be satisfied, without fixing the assignment of p. Hence, if φ is satisfiable, there exists at least one of such assignment, which then must be chosen by maximality.

Observation 3: For two satisfying assignment of φ, σ_1 and σ_2 , $\sigma_1 \prec_{lex} \sigma_2$ with respect to the vector $(x_i)_{i\in\{1,\ldots n\}}$, if and only if $w(\sigma_1) < w(\sigma_2)$.
Firstly, since both assignments satisfy φ, p is always satisfied. Therefore, the order induced by the weight does only depend on which "variable" clause in ψ_x is satisfied. Hence, it suffices to focus solely on that part. Assume $\sigma_1 \prec_{lex} \sigma_2$ with respect to the vector $(x_i)_{i\in\{1,\ldots n\}}$ then there exists one k such that for all $1 \leq i < k$ it holds that $\sigma_1(x_i) = \sigma_2(x_i)$ and that $\sigma_1(x_k) < \sigma_2(x_k)$, with the remaining variables being arbitrary. Now since $2^{k-1} > \sum_{i=0}^{k-2} 2^i$ the assignment of variables x_i for $i > k$ is irrelevant, i.e. even if all variable clauses greater k are satisfied they can not outweigh $w(x_k)$. Therefore the inequality $w(\sigma_2) > w(\sigma_1)$ must hold. Assuming that $w(\sigma_1) < w(\sigma_2)$. For the subformula ψ_x, those have the form $\omega_1 := \sum_{i\in\{1,\ldots,n\}} \sigma_1(x_i) \cdot 2^{n-i}$ and $\omega_2 := \sum_{i\in\{1,\ldots,n\}} \sigma_2(x_i) \cdot 2^{n-i}$. Since $\omega_1 < \omega_2$, σ_1 and σ_2 must differ at least on one position. Let k be the first of those differences. Similar as before, since satisfying the position k outweighs every possible assignment of the variable clauses after (x_k), i.e. (x_i) for $k < i$, it must be the case that $\sigma_2(x_k) = 1$ and $\sigma_1(x_k) = 0$, thus $\sigma_1 \prec_{lex} \sigma_2$.

Observation 4: If p is satisfied under a weight maximal assignment σ, then φ is satisfied under σ.
Since $\sigma(p) = 1$, it follows that (p) is satisfied. Additionally, since $w(C) > w(p)$ for any $C \in Cl(\psi_c)$, if any of the clauses in ψ_c would not be satisfied, flipping the assignment of p would increase the total weight of the assignment. However, with σ being weight maximal, one can thus conclude that all clauses in ψ_c are satisfied. Hence, σ is also a satisfying assignment of φ.

Finally, those observations culminate in the following.

" $\Rightarrow$ ": Assume that the instance $(\chi, (x_1, \ldots, x_n))$ is a satisfying instance of the problem **MSA**. Hence, χ is satisfiable and the maximal lexicographical assignment σ evaluates x_n to 1. Since, π preserves satisfiability, in such a way that restricting an assignment for $\varphi := \pi(\chi)$ to the variables only occurring in χ provides a model for χ. Meaning that every clause in $Cl(\varphi)$ is satisfiable. Hence, p will be satisfied under any maximal weight assignment. Moreover, by assumption x_n will be satisfied, thus the total weight of the assignment will be even. Furthermore, since x_n is satisfied under the lexicographical maximal assignment, by Observation 3, it must be satisfied under the weight maximal assignment as well. Thus it is a positive instance of **e-WSAT**.

" $\Leftarrow$ ": Assume that the instance ψ, is a satisfying instance of the problem **e-WSAT**. Meaning for some weight maximal assignment σ, $w(\sigma)$ is even. Recall $\varphi = \bigwedge_{(\neg x \vee C) \in Cl(\psi_c)} C$. There are two cases.

- Case 1: $\sigma(p) = 0$ and $\sigma(x_n) = 0$
  By Observation 2 φ, and therefore χ, can not be satisfied. Since, $\sigma(p) = 0$ all clauses in ψ_c are satisfied. Hence, the variables $x_1, \dots, x_n$ can be chosen freely without interfering with the truth value of the clauses clauses in ψ_c. Thus, there exists an assignment σ' that is the same σ, except $\sigma'(x_n) = 1$. Clearly, $w(\sigma') > w(\sigma)$ and $w(\sigma')$ odd. Implying the impossibility of this case.
- Case 2: $\sigma(p) = 1$ and $\sigma(x_n) = 1$
  Since, σ is weight maximal, it follows form Observation 4, that φ is satisfied under σ such that $\sigma(x_n) = 1$. Moreover, from this and from Observation 3, it follows that the weight maximal assignment is also the lexicographical maximal assignment. Thus by restricting σ to the variables in χ one obtains a lexicographical maximal assignment in which x_n evaluates to 1. Hence, the original problem is a positive instance of **MSA**.

In a last step membership has to be shown. In order to establish membership, it suffices to give an appropriate algorithm. The idea of which is fairly simple:

- Use an algorithm $\mathcal{A}_{WSAT}(\varphi, k)$ that solves for k-**WSAT** as an oracle.
- Check if there exists a satisfying assignment, if there does not return false.
- Otherwise, run binary search to find the greatest k, such that $\forall n > k$ the input (φ, n) is no longer an accepted instance of k-**WSAT**.
- if k is even return true; else return false

If there exists a solution, this program will return a smallest satisfying assignment. Moreover, since the complexity of binary search is $\mathcal{O}(logn)$, the oracle will be called upon at most a logarithmic amount of times, wrt. to the total weight Moreover, since those weights are some natural number their sum can be bounded by some exponent of $2^{f(|Cl(\varphi)|)}$ for some polynomial function f. Hence, making it a polynomial number of calls to the oracle, wrt. to the number of clauses.

```
e-SAT(φ,w):
   u := Σ_{c∈Cl(φ)} w(c)
   l := 0
   while u ≠ l:
          n := ⌈(u+l)/2⌉
          if A_WSAT(φ,n) = true:
              l := n
          else:
              u := n − 1
   if u is even:
          return true
   else:
          return false
```

(c) Bonus: Given an integer $k \geq 0$, what is the complexity of deciding whether $w(\sigma) \leq k$ for every maximum weight assignment σ ?

As established above given an integer $k \geq 0$, deciding whether $w(\sigma) \geq k$ for some maximum weight assignment σ is **NP**-complete. Written more concise

$$\exists \sigma \ (\sigma \textit{ maximum weight assignment } \wedge w(\sigma) \geq k)$$

Hence, the negation of this problem is, given an integer $k \geq 0$

$$\forall \sigma \ (\neg \sigma \textit{ maximum weight assignment } \vee \neg (w(\sigma) \geq k))$$

which is the same as

$$\forall \sigma \ (\sigma \textit{ maximum weight assignment } \implies w(\sigma) < k)$$

Therefore, the problem: Given an integer $k \geq 0$, what is the complexity of deciding whether $w(\sigma) \leq k$ for every maximum weight assignment σ. Is merely the negation of the problem $(k+1)$-**WSAT**. That is,

$$\exists \sigma \ (\sigma \textit{ maximum weight assignment } \wedge w(\sigma) \geq k+1)$$

is equal to

$$\forall \sigma \ (\sigma \textit{ maximum weight assignment } \implies w(\sigma) < k+1)$$

which is equal to

$$\forall \sigma \ (\sigma \textit{ maximum weight assignment } \implies w(\sigma) \leq k)$$

Now, given the fact that $(k+1)$-**WSAT** is **NP**-complete, its complement is co-**NP**-complete.

CT.6 (10) The class $\mathbf{FP^{NP}}[log, wit]$ (cf. Unit 1) contains the search problems that can be solved in polynomial time with a witness oracle, which loosely speaking returns some solution to a problem in $\mathbf{NP}$ (e.g., for **SAT**, a satisfying assignment).

A search problem Π can be solved in $\mathbf{FP^{NP}}[log, wit]$, if for every instance I of Π some solution of I can be computed in polynomial time with a witness oracle for $\mathbf{NP}$, which can be consulted at most $O(\log n)$ times, where $n = |I|$ is the length of the input.

(a) show that computing some smallest (w.r.t. cardinality) model of a Boolean formula is in $\mathbf{FP^{NP}}[log, wit]$;

(b) show that computing some minimal (w.r.t. $\subseteq$) model of a Boolean formula is in $\mathbf{FP^{NP}}[log, wit]$;

(c) Technically, $\mathbf{FP^{NP}}[log, wit]$ requires that each input has some output, i.e., it consists of *total (multi-valued) functions* (thus, for computing a maximum model of a SAT instance E, a special value like “unsat” is output in order to flag unsatisfiability); and each run produes output.

Show that $\mathbf{FP^{NP}}[log, wit] \subseteq \mathbf{FP^{NP}}$, where $\mathbf{FP^{NP}}$ are the search problems Π such that for every instance I of Π some solution of I can be computed in polynomial time with an “ordinary” oracle for $\mathbf{NP}$.

Bonus: consider whether this inclusion would hold if problem instances may have no solutions, and in this case computations do not generate output.

Solution:

WARNING: This whole exercise relies on the fact that $\mathbf{FSAT}$ is an $\mathbf{FNP}$-complete problem.

(a) show that computing some smallest (w.r.t. cardinality) model of a Boolean formula is in $\mathbf{FP^{NP}}[log, wit]$;

*Observation 1: The problem **FSAT**$_{\leq k}$ taking an $k \in \mathbb{N}$ and an propositional formula φ as input, and returning a model of size smaller or equal to k is **FNP**-complete.*

Membership: Consider

```
FSAT≤k(φ, k):
       M := guess(φ) // guess an assignment for φ
       if |M| ≤ k and M ⊨ φ :
          return M
       else :
          return false
```

This algorithm is a guess and check algorithm running in polynomial time and returning, if possible, a result for the actual problem. Thereby establishing membership

Hardness: $\mathbf{FSAT}$ returns a model of any size, if φ is satisfiable. Moreover, those model are at most of size $|\mathit{Var}(\varphi)|$. Therefore, it is clear to see that $\mathbf{FSAT}$ is merely an special case of $\mathbf{FSAT}_{\leq k}$, namely $\mathbf{FSAT}_{|\mathit{Var}(\varphi)|}$. Given the fact that $\mathbf{FSAT}$ is $\mathbf{FNP}$-complete, one obtains $\mathbf{FNP}$-hardness.

In order to establish membership, it suffices to give an appropriate algorithm. The idea of which is fairly simple:

- Use $\mathbf{FSAT}_{\leq k}$ as an oracle.
- Check if there exists a satisfying assignment, if there does not return false.
- Otherwise, run binary search to find the smallest k, such that $\forall i < k$ the input (φ, i) is no longer an accepted instance of $\mathbf{FSAT}_{\leq i}$.
- return the last satisfying model.

If there exists a solution, this program will return a smallest satisfying assignment. Moreover, since the complexity of binary search is $\mathcal{O}(logn)$, the oracle will be called upon at most a logarithmic amount of times, wrt. the number of variables in φ. Moreover, only the last call of $\mathbf{FSAT}_{\leq k}$ is important, all other calls could can also be done with a normal decision oracle. To be more specific.

```
Min-Card-SAT(φ):
    u := |Var(φ)|
    l := 0
    if FSAT≤k(φ, u) = false:
        return false
    while u ≠ l:
        n := ⌊(u+l)/2⌋
        M := FSAT≤k(φ, n)
        if M = false:
            l := n + 1
        else:
            u := n
    return M
```

(b) show that computing some minimal (w.r.t. $\subseteq$) model of a Boolean formula is in $\mathbf{FP}^{\mathbf{NP}}[log, wit]$;
In order to establish membership, it suffices to give an appropriate algorithm. However, in this particular case membership directly follows form the following observation.

Observation 2: Any minimal model with respect to cardinality is a subset minimal model.
This observation is trivial, as otherwise, there would be another model with less elements than a model with smallest number of elements. Which is clearly a contradiction.

Since, the problem only requires the computation of *some* subset minimal model, it is clear that this can be done by computing a smallest model with respect to cardinality. Therefore, establishing $\mathbf{FP}^{\mathbf{NP}}[log, wit]$ membership, through the runtime of `Min-Card-SAT(φ)`.

(c) Show that $\mathbf{FP}^{\mathbf{NP}}[log, wit] \subseteq \mathbf{FP}^{\mathbf{NP}}$, where $\mathbf{FP}^{\mathbf{NP}}$ are the search problems Π such that for every instance I of Π some solution of I can be computed in polynomial time with an "ordinary" oracle for $\mathbf{NP}$.

The main idea required for showing that result is to use the concept of self-reducibility to show that it is possible to replace a single call to the $\mathbf{NP}$-complete witness oracle by a polynomial number of calls to a $\mathbf{NP}$-complete decision oracle.

*Observation 3: **SAT** is self-reducible*

That is,

- Given an oracle of **SAT**, solve **FSAT** in polynomial time.
- Given an oracle of **FSAT**, solve **SAT** in polynomial time.

The latter is trivial, i.e. ask **FSAT** once for a model, if a model is returned the formula in question is clearly satisfiable. As for the prior, consider the following algorithm.

```
FSAT(φ):
  If SAT(φ) = FALSE:
     return FALSE
  σ := {} // Empty variable assignment
  for x ∈ Var(φ):
     if SAT(φ ∧ x) = TRUE:
        σ := σ ∪ {x ↦ 1}
        φ := φ ∧ x
     else:
        σ := σ ∪ {x ↦ 0}
        φ := φ ∧ ¬x
  return σ
```

The first observation is that the algorithm above runs in polynomial time wrt. the number of variables. In fact, the number of calls to the oracle are linear. The second one, is that it produces a valid assignment. Thirdly, a formula has no model if and only if it is unsatisfiable. Hence, the algorithm above detects all negative cases. Lastly, having weeded out the negative cases, it has to be established that the algorithm always finds a model. To that end consider a partial assignment $\sigma_{(x_1,\ldots,x_i)}$ of the variables $x_1, \ldots, x_i \in \mathit{Var}(\varphi)$ such that there exists an extension $\sigma_{(x_1,\ldots,x_n)}$ under which φ is satisfiable. Furthermore, consider the variable x_{i+1}. Due to the guarantee that $\sigma_{(x_1,\ldots,x_i)}$ can be extended, it follows that if $\sigma_{(x_1,\ldots,x_i)} \cup \{x_{i+1} \mapsto 1\}$ can no longer be extended to a satisfying assignment, then $\sigma_{(x_1,\ldots,x_i)} \cup \{x_{i+1} \mapsto 0\}$ must be, and vice versa. This is precisely, what happens in the `for`-loop. Meaning that at each step, the assignment is extended such that satisfiability is maintained. However, as the usual $\mathbf{SAT}$ problem does not take partial assignments as input, unit clauses are added, forcing a particular assignment of variables.

Now using this observation and the fact that $\mathbf{FSAT}$ is $\mathbf{FNP}$-complete, one can use the following polynomial time transformation.

- $\mathbf{FSAT}$ is $\mathbf{FNP}$-complete.
- Any other $\mathbf{FNP}$-complete $\mathcal{P}_F$ can be reduced to $\mathbf{FSAT}$.
- $\mathbf{FSAT}$ can be solved by a polynomial number of calls to a $\mathbf{SAT}$ oracle.
- However, $\mathbf{SAT}$ can be reduced to any other $\mathbf{NP}$-complete problem.

The other direction is trivial, i.e. similar as in the $\mathbf{SAT}$ case, the decision problem can be solved by solving the search problem.

Hence, for each $\mathbf{FNP}$-complete problem, there exists an algorithm that can transform said problem in its corresponding decision problem.

The last remaining step is to show $\mathbf{FP}^{\mathbf{NP}}[log, wit] \subseteq \mathbf{FP}^{\mathbf{NP}}$. Consider an arbitrary problem $\mathcal{P}$ in $\mathbf{FP}^{\mathbf{NP}}[log, wit]$. Hence, there exists an algorithm $\mathcal{A}[\mathcal{Q}_F]$, that given an instance for $\mathcal{P}$, always returns a solution in polynomial time with a logarithmic number of calls to the witness oracle $\mathcal{Q}_F$. Meaning that $\mathcal{Q}_F$ is be some **FNP**-complete problem. Given the observation above, this oracle can be exchanged by its corresponding $\mathbf{NP}$-complete decision problem $\mathcal{Q}$ with just a polynomial time overhead. Let this algorithm be called $\mathcal{B}$ and thus $\mathcal{Q}_F = \mathcal{B}(\mathcal{Q})$. Hence, there resulting algorithm $\mathcal{A}' = \mathcal{A}[\mathcal{B}(\mathcal{Q})]$ is in $\mathbf{FP}^{\mathbf{NP}}$.

CW.1 (10) Give a formal proof that for a given propositional theory T, CWA(T) is consistent if and only if T has a single $\leq$-minimal model M.

Solution:

Note: Since consistency is a syntactic notion and minimal models are a semantic one, the following observations rely implicitly on the soundness and completeness of propositional logic.

Observation 1: Let T be a consistent propositional theory, $CWA(T) = T \cup \{\neg p \mid p \in Var(T),\, \exists \mathcal{M} \in MMod(T)\; \mathcal{M} \models \neg p\}$

Consider the definition of the CWA, i.e. $CWA(T) = T \cup \{\neg p \mid p \in Var(T),\ T \not\models p\}$. By semantics of entailment one obtains the equality

$$\{\neg p \mid p \in Var(T),\ T \not\models p\} = \{\neg p \mid p \in Var(T),\, \exists \mathcal{M} \in Mod(T)\; \mathcal{M} \models \neg p\}.$$

Now, consider the definition of subset minimality. That is, for the two models $\mathcal{I}$ and $\mathcal{I}'$ of a formula χ, $\mathcal{I}' \leq \mathcal{I}$ iff $\mathcal{I}' \subseteq \mathcal{I}$ iff $\forall p\, Var(\chi)\; \mathcal{I}' \models p \Rightarrow \mathcal{I} \models p$. Hence, if there exists a model $\mathcal{M}$ such that $\mathcal{M} \models \neg p$ then there must be a minimal model $\mathcal{M}'$ also satisfying $\neg p$. Therefore,

$$\{\neg p \mid p \in Var(T),\, \exists \mathcal{M} \in Mod(T)\; \mathcal{M} \models \neg p\} = \{\neg p \mid p \in Var(T),\, \exists \mathcal{M} \in MMod(T)\; \mathcal{M} \models \neg p\}$$

Hence, arriving at

$$CWA(T) = T \cup \{\neg p \mid p \in Var(T),\, \exists \mathcal{M} \in MMod(T)\; \mathcal{M} \models \neg p\}$$

Observation 2: If T is a satisfiable propositional theory such that $MMod(T) := \{\mathcal{M}_m\}$, then $CWA(T)$ is consistent.

From Observation 1, one obtains that $CWA(T) = T \cup \{\neg p \mid p \in Var(T),\, \exists \mathcal{M} \in MMod(T)\; \mathcal{M} \models \neg p\}$, which given the assumption about T gives the equality

$$CWA(T) = T \cup \{\neg p \mid p \in Var(T),\ \mathcal{M}_m \models \neg p\}$$

Clearly, $\mathcal{M}_m$ satisfies both T and $\{\neg p \mid p \in Var(T),\ \mathcal{M}_m \models \neg p\}$. Hence, by semantics it $\mathcal{M}_m$ satisfies the union of those sets. Therefore, $\mathcal{M}_m$ is a model of $CWA(T)$, thus establishing consistency.

Observation 3: If T is a satisfiable propositional theory such that $|MMod(T)| > 1$, then $CWA(T)$ is inconsistent.

From Observation 1, one obtains that $CWA(T) = T \cup \{\neg p \mid p \in Var(T),\, \exists \mathcal{M} \in MMod(T)\; \mathcal{M} \models \neg p\}$. Hence, for any model $\mathcal{M}_{CWA}$ of $CWA(T)$ it must be that

$$\mathcal{M}_{CWA} \cap \{p \mid p \in Var(T),\, \exists \mathcal{M} \in MMod(T)\; \mathcal{M} \models \neg p\} = \emptyset$$

Thus implying that

$$\mathcal{M}_{CWA} \subseteq Var(T) \setminus \{p \mid p \in Var(T),\, \exists \mathcal{M} \in MMod(T)\; \mathcal{M} \models \neg p\}$$

which is equivalent to

$$\mathcal{M}_{CWA} \subseteq \{p \mid p \in \mathit{Var}(T),\, \forall \mathcal{M} \in \mathit{MMod}(T)\; \mathcal{M} \models p\}$$

Now consider the set $\{p \in \mathit{Var}(T) \mid \forall \mathcal{M} \in \mathit{MMod}(T)\; \mathcal{M} \models p\}$ this is the same as considering the intersection $\mathcal{M}_\cap := \bigcap_{\mathcal{M} \in \mathit{MMod}(T)} \mathcal{M}$. However, there are at least two distinct minimal models $\mathcal{M}_1$ and $\mathcal{M}_2$. Clearly, $\mathcal{M}_\cap \subset \mathcal{M}_1$ and $\mathcal{M}_\cap \subset \mathcal{M}_2$. Hence, if $\mathcal{M}_\cap$ would be a model of T, minimality of $\mathcal{M}_1$ and $\mathcal{M}_2$ would be violated. Hence, $\mathcal{M}_\cap$ as well as any subset of it, can not be a model of T. Thus only the interpretations $\mathcal{M}' \subseteq \mathcal{M}_\cap$ can model $\{\neg p \mid p \in \mathit{Var}(T),\, \exists \mathcal{M} \in \mathit{MMod}(T)\; \mathcal{M} \models \neg p\}$, while at the same time they can not be models of T. Hence, no model can satisfy $\mathit{CWA}(T)$, thereby rendering it inconsistent.

By negating Observation 3 one obtains "For T being a satisfiable propositional theory. If $\mathit{CWA}(T)$ is consistent, then $|\mathit{MMod}(T)| = 1$.". Together with Observation 2 one therefore obtains that, given a satisfiable propositional theory T, $\mathit{CWA}(T)$ is consistent if and only if T has a single $\leq$-minimal model $\mathcal{M}$. Lastly, assume that T is not satisfiable, then trivially $\mathit{CWA}(T)$ can not be consistent and $|\mathit{MMod}(T)| = |\emptyset| \neq 1$. Hence, allowing for arbitrary propositional theories, and thus the required claim is established.

TCR.7 (10) Show that $\mathbf{CF}_G$ is Π_2^P-hard by a reduction from Closed World Reasoning under the EGCWA, i.e., from EGCWA$(T) \models F$ (which has the same complexity). Provide also a reduction in the other direction, i.e., from deciding EGCWA$(T) \models F$ to $\mathbf{CF}_G$.

Bonus: Consider the use of ECWA$(T; P; Z)$ in place of EGCWA(T).

Solution:

Firstly, $\mathbf{CF}_G$ is the problem of deciding whether for some theory T and two formulas φ and ψ, $T \models \psi >_G \varphi$ holds. Which again is merely a shorthand for $T \circ_G \psi \models \varphi$. This leading to the definition of the two problems at hand.

$T \circ_G \psi \models \varphi$ is equivalent to $\{T' \cup \{\psi\} \mid T' \in W(\psi, T)\} \models \varphi$. Leading the completely unwrapped definition of

$$\{T' \cup \{\psi\} \mid T' \in \max_{\subseteq}\{T'' \subseteq T : T'' \not\models \neg\psi\}\} \models \varphi$$

That is, $W(\psi, T)$ is the set of maximal sub-theories of T that are consistent with ψ. Hence, $T \circ_G \psi \models \varphi$ expresses that all maximal ψ-consistent sub-theories of T entail φ.

By contrast, $EGCWA(T) \models \varphi$ is equivalent to

$$T \cup \{\neg C \mid C \ conjunction \ of \ a \in Var(T), \ MMod(T) \models \neg C\} \models \varphi$$

with $MMod(T) \models \psi$ expressing that all minimal models wrt. positive information ($M \in MMod(T) \wedge M \subseteq M' \implies (\sigma_M(x) = 1 \Rightarrow \sigma_{M'}(x) = 1)$) entail ψ.

- Starting with the reduction $EGCWA(T) \models \varphi \leq_m^P \mathbf{CF}_G$ for input $\tau(T, \varphi)$, where τ is defined as

$$\tau(T, \varphi) = (KB, T_\wedge, \varphi) = (\{\neg x \mid \forall x \in Var(T)\}, \bigwedge_{\psi \in T} \psi, \varphi)$$

  meaning that one obtains the problem $KB \circ_G T_\wedge \models \varphi$.

  Observation 1: If T is satisfiable, then for $A \in W(KB, T_\wedge)$, there exists exactly one model $\mathcal{M}$, such that $\mathcal{M} \models A \cup T$.

  Any model satisfying A must evaluate all variables in A as 0. Therefore, any model of $\mathcal{M} \models A \cup T$ must build on the partial assignment $\sigma_A := \{x \mapsto 0 \mid \forall x \in Var(A)\}$, i.e. $\sigma_A \subseteq \sigma_{\mathcal{M}}$. It is known that there exists a model $\mathcal{M}_A$ of A such that $\mathcal{M}_A \models T_\wedge$. Claim, this model has the form $\mathcal{M}_A := \{x \mid \forall x \in Var(T) \setminus Var(A)\}$. Assume it does not. That is, consider $\mathcal{M} \supset \mathcal{M}_A$. Thus, it would be possible to add $\mathcal{M}_A \setminus \mathcal{M}$ to A, while retaining the condition $A \not\models \neg\psi$. Thereby, violating maximality. Hence, if there exists a smaller model of A, then A is not maximal. Hence, $\mathcal{M}_A$ is the only model of A. Now, since there must exist a model of A that satisfies $T_\wedge$ it follows that $\mathcal{M}_A$ is a model of $T_\wedge$ and therefore the only model of $A \cup T$.

Observation 2: For a given theory T,

$$MMod(T) = \bigcup_{A \in W(KB, T_\wedge)} \{\mathcal{M} \mid \forall \mathcal{M}\ \mathcal{M} \models A \cup T\}$$

If T is not satisfiable, clearly

$$MMod(T) = \emptyset = \bigcup_{A \in W(KB, T_\wedge)} \{\mathcal{M} \mid \forall \mathcal{M}\ \mathcal{M} \models A \cup T\}$$

" $\Rightarrow$ ": Take $\mathcal{M} \in MMod(T)$. Clearly, $\mathcal{M}$ models T and thus $\mathcal{M} \models T_\wedge$. Using the induced assignment $\sigma_\mathcal{M}$ construct $A_\mathcal{M} := \{\neg x \mid \forall x \in Var(T)\ \sigma_\mathcal{M}(x) = 0\}$, which is satisfied by $\mathcal{M}$ by construction. Hence, it remains to show that $A_\mathcal{M}$ is maximal. Assume that there exists a $A_{\mathcal{M}'} \supset A_\mathcal{M}$, such that $\mathcal{M}'$ satisfies T . Hence, there exists x, such that $\sigma_{\mathcal{M}'}(x) = 0$, while $\sigma_\mathcal{M}(x) = 1$. However, this contradict the minimality of $\mathcal{M}$.
" $\Leftarrow$ ": Take $\mathcal{M}_A \in \bigcup_{A \in W(KB, T_\wedge)} \{\mathcal{M} \mid \forall \mathcal{M}\ \mathcal{M} \models A \cup T\}$. By construction $\mathcal{M}_A$ models T. It remains to show that $\mathcal{M}_A$ is minimal. Since, A contains the maximal amount of negative literals such that T is not contradicted (see observation 1), any model $\mathcal{M} \subset \mathcal{M}_A$ can not be a model of T. Therefore, $\mathcal{M}_A$ is a minimal model of T.
Since $\mathcal{M} \models A \cup T$ is unique one could rewrite the statement to: for a given theory T,

$$MMod(T) = \{\mathcal{M}_A \mid \forall A \in W(KB, T_\wedge)\ \mathcal{M}_A \models A \cup T\}$$

Using those observation one obtains the following.
Assume that $EGCWA(T) \models \varphi$. Hence, $\forall \mathcal{M} \in Mod(EGCWA(T))\ \mathcal{M} \models \varphi$, which given the theorem in the lecture slides, is equivalent to $\forall \mathcal{M} \in MMod(T)\ \mathcal{M} \models \varphi$. Now using the observation 2 one obtains the equivalent statement $\forall \mathcal{M} \in \{\mathcal{M}_A \mid \forall A \in W(KB, T_\wedge)\ \mathcal{M}_A \models A \cup T\}\ \mathcal{M} \models \varphi$, which is equivalent to $KB \circ_G T_\wedge \models \varphi$.

- Secondly, the reduction $KB \circ_G \psi \models \varphi \leq_m^P EGCWA(T) \models \varphi$, where τ is defined as $\tau(KB, \psi, \varphi) = (T, \varphi)$ with T being

$$\{\psi\} \cup \{x_i \not\equiv y_i \mid \forall x_i \in Var(KB \cup \{\psi\})\} \cup \{d_i \vee \chi_i \mid \forall \chi_i \in KB\}$$

where y_i, d_i are some fresh propositional variables (including the variables in φ). As a shorthand let $T_y := \{x_i \not\equiv y_i \mid \forall x_i \in Var(KB \cup \{\psi\})\}$ and let $T_d := \{d_i \vee \chi_i \mid \forall \chi_i \in KB\}$

First of all, this transformation can be done in polynomial time with respect to the amount of clauses (for constructing T_d) and variables (for construction T_y).

Observation 1: $MMod(T_y) = Mod(T_y)$.
Clearly $MMod(T_y) \subseteq Mod(T_y)$. Consider $\mathcal{M} \in Mod(T_y)$ assume that there exists a $\mathcal{M}' \subset \mathcal{M}$. Hence, $\exists z \in \mathcal{M}\ z \notin \mathcal{M}'$. However, w.l.o.g. assume that $z = x_i$ then $\sigma_{\mathcal{M}'}(x_i) = 0$, this requires through $x_i \neq y_i$ that $\sigma_{\mathcal{M}'}(y_i) = 1$ and thus $y_i \in \mathcal{M}'$. Clearly,

$y_i \notin \mathcal{M}$. Hence, $\mathcal{M}' \not\subset \mathcal{M}$. Thus, $\mathcal{M}$ is minimal, i.e. $\mathcal{M} \in MMod(T_y)$

Observation 2: There exists a bijection π from $\bigcup_{KB' \in W(\psi,KB)} Mod(KB' \cup \{\varphi\})$ to $MMod(T)$. Where

$$\pi(\mathcal{M}_K) := \mathcal{M}_T = \mathcal{M}_K \cup \{y_i \mid \forall (x_i \not\equiv y_i) \in T_y \; x_i \notin \mathcal{M}_K\} \cup \{d_i \mid \forall (d_i \vee \chi_i) \in T_d \; \mathcal{M}_K \not\models \chi_i\}$$

where its inverse is simply the removal of the added elements, i.e. $\{y_i \mid \forall (x_i \not\equiv y_i) \in T_y \; x_i \notin \mathcal{M}_K\} \cup \{d_i \mid \forall (d_i \vee \chi_i) \in T_d \; \mathcal{M}_K \not\models \chi_i\}$.
First, it has to be demonstrated that $\mathcal{M}_T$ is actually a minimal model. Let $\mathcal{M}_K$ be an arbitrary model of $KB' \cup \{\varphi\}$ for some $KB' \in W(\psi, KB)$. Clearly, $\mathcal{M}_K \cup \{y_i \mid \forall (x_i \not\equiv y_i) \in T_y \; x_i \notin \mathcal{M}_K\}$ satisfies T_y. By Observation 1, it follows that it is minimal. Hence, it is from now on only relevant, if a χ_i is satisfied, and not how it is satisfied. Speaking of which, since KB' is a maximal subset of KB that is still consistent with ψ, $\mathcal{M}_K$ must satisfy a maximal amount of χ_i's, i.e. adding any additional χ_i leads to inconsistency with ψ. Therefore, the set $\{\chi_i \mid \forall \chi_i \in KB \mathcal{M}_K \models \chi_i\}$ is maximal and thereby the set $\{\chi_i \mid \forall \chi_i \in KB \mathcal{M}_K \not\models \chi_i\}$ is minimal. Hence, only a minimal amount of d_i's are satisfied, i.e. $\{d_{ij} \mid \forall (d_i \vee \chi_i) \in T_d \; \mathcal{M}_K \not\models \chi_i\}$ is minimal. Notice whether a model is minimal, depends solely on how many d_i's it satisfies. Since, d_i is only satisfied, if χ_i is not modelled by $\mathcal{M}_K$ and vice versa, $\mathcal{M}_T$ is minimal model of T.

Secondly, one has to demonstrate that π is surjective. Assume that there exists a model $\mathcal{M}_T \in MMod(T)$ and that there exists no $\mathcal{M}_K \in \bigcup_{KB' \in W(\psi,KB)} Mod(KB' \cup \{\varphi\})$ such that $\pi(\mathcal{M}_K) = \mathcal{M}_T$. Clearly, $\mathcal{M}_T$ is of the form $\mathcal{M}_X \cup \{y_i \mid \forall (x_i \not\equiv y_i) \in T_y \; x_i \notin \mathcal{M}_K\} \cup \{d_i \mid \forall (d_i \vee \chi_i) \in T_d \; \mathcal{M}_K \not\models \chi_i\}$ for some partial model $\mathcal{M}_X$. Moreover, $\mathcal{M}_X$ induces an assignment on the variables x_i. Since, $\mathcal{M}_T$ is minimal, either d_i or χ_i is satisfied. Since, d_i is more expensive it follows that, a maximal amount of χ_i will be satisfied. Therefore, $\mathcal{M}_X$ satisfies a maximal amount of χ_i, as their truth value only depends on the values of x_i's. Thus, it is a model of the set, $\{\chi_i \mid \forall \chi_i \in KB \; \mathcal{M}_K \models \chi_i\} \in W(\psi, KB)$ for some model $\mathcal{M}_K$.

Thirdly, π is injective due to the fact that the assignments of the variables y_i's and d_i's are uniquely determined by the assignment of x_i's.

Finally, one can tackle the reduction. Let π be the bijection from Observation 2.
" $\Rightarrow$ ": Assume that $KB \circ_G \psi \models \varphi$. Meaning that $\forall KB' \in W(\psi, KB)$ and $\forall \mathcal{M}_K$ such that $\mathcal{M}_K \models KB' \cup \{\varphi\}$, it holds that $\mathcal{M}_K \models \varphi$. Which is precisely the domain of π. Since, $\mathcal{M}_T = \pi(\mathcal{M}_K)$ is an extension of $\mathcal{M}_K$ by variables that are not present in φ one obtains $\mathcal{M}_T \models \varphi$. As established above $\mathcal{M}_T \in MMod(T)$, thus through the fact that π is bijective one obtains $MMod(T) \models \varphi$.
" $\Leftarrow$ ": Assume that $MMod(T) \models \varphi$. Take an arbitrary model $\mathcal{M}_T \in MMod(T)$, since the truth value of φ does not depend on the variables y_i and d_i in the sentences T_y and T_d respectively, one can safely restrict the model to $\mathcal{M}_X$ in $\mathcal{M}_X \cup \{y_i \mid \forall (x_i \not\equiv y_i) \in T_y \; x_i \notin \mathcal{M}_K\} \cup \{d_i \mid \forall (d_i \vee \chi_i) \in T_d \; \mathcal{M}_K \not\models \chi_i\}$, which is incidentally $\pi^{-1}(\mathcal{M}_T)$. Thus, $\pi^{-1}(\mathcal{M}_T) \models \varphi$. With $\mathcal{M}_T$ arbitrary and π bijective.

- Bonus: Consider the use of ECWA$(T; P; Z)$ in place of EGCWA(T).

  As a reminder

  - $P \ldots$ atoms to be minimized;
  - $Q \ldots$ atoms that are fixed;
  - $Z \ldots$ atoms that may take arbitrary value when minimizing P;

  The reduction $KB \circ_G \psi \models \varphi \leq_m^P ECWA(T; P; Z) \models \varphi$, is the same as the one on the main exercise, i.e. $KB \circ_G \psi \models \varphi \leq_m^P EGCWA(T) \models \varphi$. This results from the fact that EGCWA = ECWA where $Q = Z = \emptyset$.

  Moving on to the reduction, $ECWA(T; P; Z) \models \varphi \leq_m^P$ $\mathbf{CF}_G$ for input $\tau(T, \varphi)$, where τ is defined as

  $$\tau(T; P; Z, \varphi) = (KB, T_\wedge, \varphi) = (\{\neg x \mid \forall x \in P\}, \bigwedge_{\psi \in T} \psi, \varphi)$$

  where

  $$KB := \{\neg p \mid \forall p \in P\} \cup \{\neg q \mid \forall q \in Q\} \cup \{\neg q \mid \forall q \in Q\}$$
  $$T_\wedge := \bigwedge_{\psi \in T} \psi$$

  Meaning that one obtains the problem $KB \circ_G T_\wedge \models \varphi$.

  Firstly, notice due to the fact that no atom $z \in Z$ occurs in KB, z is neither minimised nor fixed and can therefore be chosen in an arbitrary manner to satisfy $T_\wedge$.

  Observation 1: Two subsets KB' and KB'' of KB that are consistent with $T_\wedge$ are only comparable if $Q \cap KB' = Q \cap KB''$.

  This is, due to the fact that the subset has to be consistent with $T_\wedge$. Hence, the subset itself must be consistent. However, with z and $\neg z$ being complementary, it can never be that both are present in such a subset of KB. That is, $q \in KB' \wedge q \notin KB''$ implies $\neg q \notin KB' \wedge \neg q \in KB''$.

  Observation 2: For a given theory T,

  $$MMod(T; P; Z) = \bigcup_{A \in W(KB, T_\wedge)} \{\mathcal{M} \mid \forall \mathcal{M}\ \mathcal{M} \models A \cup \{T_\wedge\}\}$$

  If T is not satisfiable, clearly

  $$MMod(T; P; Z) = \emptyset = \bigcup_{A \in W(KB, T_\wedge)} \{\mathcal{M} \mid \forall \mathcal{M}\ \mathcal{M} \models A \cup \{T_\wedge\}\}$$

  "$\subseteq$": Take $\mathcal{M} \in MMod(T; P; Z)$. Hence, $\mathcal{M} \models T_\wedge$. Using this the a subset $A := \{\neg z \mid \forall x \in KB\ \mathcal{M}_K \not\models z\}$ of KB is constructed. $\mathcal{M}$ is a $\leq_{P;Z}$-minimal model. Meaning that,

for the class of models with a particular subset Q' of Q, $\mathcal{M}$ is subset minimal with respect to the atoms in P. Through Observation 1, one obtains that two subsets of KB consistent with $T_\wedge$ are only comparable if they share the same subset of Q. Therefore, with $\mathcal{M}$ being subset minimal with respect to elements in P for a fixed subset of Q, it follows that A contains a maximal amount of $\neg p$ for $p \in P$ for a given subset of Q. Hence, A is a maximal consistent subset of KB that is consistent with $T_\wedge$ and that is by construction satisfied by $\mathcal{M}$. Thus showing the first inclusion.
" $\supseteq$ ": Take $\mathcal{M} \in \bigcup_{A \in W(KB,T_\wedge)} \{\mathcal{M} \mid \forall \mathcal{M}\ \mathcal{M} \models A \cup \{T_\wedge\}\}$. Let $Q_\mathcal{M} = \mathcal{M} \cap Q$, $P_\mathcal{M} = \mathcal{M} \cap P$, $Z_\mathcal{M} = \mathcal{M} \cap Z$. For this particular model, there exists at least one $A \in W(KB, T_\wedge)$ such that $\mathcal{M} \models A$. By Observation 1, if for another $A' \in W(KB, T_\wedge)$, $A \subseteq A' \vee A' \subseteq A$ holds then $Q \cap A = Q \cap A'$. Furthermore, since A forces a particular assignment of the atoms in Q, i.e. either $z \in A$ or $\neg z \in A$ and by consistency never both, one obtains $Q_\mathcal{M} = A \cap Q$. Now assume that there exists an model $\mathcal{M}' \leq_{P;Z} \mathcal{M}$. That is, $\exists \mathcal{M}'\ \mathcal{M}' \leq_{P;Z} \mathcal{M} \wedge \mathcal{M} \not\leq_{P;Z} \mathcal{M}'$, since by assumption the first part of the conjunction holds, it follows that the second part can only hold if $\mathcal{M}' \cap P \subset \mathcal{M} \cap P$. Hence, $\mathcal{M} \cap Q = \mathcal{M} \cap Q$ and $\mathcal{M}' \cap P \subset \mathcal{M} \cap P$. However, as seen in Observation 1 of the first reduction of the whole exercise. This can not be the case. Otherwise, A does not contain the maximal number of negative atoms of P. (Explanation: That is, if there would be a $p \in \mathcal{M}$ that is not in $\mathcal{M}'$, then $\neg p \in A$ by maximality, and under the assumption that $\mathcal{M}'$ is a model of A would imply that $\mathcal{M}$ is not.). Hence, $\mathcal{M} \in MMod(T; P; Z)$.

Lastly, by the same argumentation as above (i.e. first reduction of this exercise) one can conclude that $ECWA(T; P; Z) \models \varphi$ if and only if $KB \circ_G \psi \models \varphi$ such that $\tau(T; P; Z, \varphi) = KB, T_\wedge, \varphi)$.

TCR.8 (10) The problem **ODDSAT** is the instance of the problem on slide 62 for $k = 1$, i.e.,

INSTANCE: SAT instances $E_1, \dots, E_n$, $n \geq 1$.

QUESTION: Is the number of satisfiable formulas among $E_1, \dots, E_n$ (i.e., $|\{E_i \mid E_i$ is satisfiable, $1 \leq i \leq n\}|$) an odd number?

(a) Show that **ODDSAT** $\leq_m^p$ **EVENSAT**, which has the same instances but the question is negated (i.e., the number of satisfiable formulas E_i is even)

(b) Show that **EVENSAT** $\leq_m^p$ **CF**$_D$ by providing a polynomial-time transformation; without loss of generality, you may assume that satisfiability of E_i implies satisfiability of E_{i+1}, for all $1 \leq i < n$.

Bonus: show that the "without loss of generality" assumption in item (b) holds.

Solution:

(a) Show that **ODDSAT** $\leq_m^p$ **EVENSAT**, which has the same instances but the question is negated (i.e., the number of satisfiable formulas E_i is even)

Consider the following transformation τ

$$\tau(E_1, \dots, E_k) = E_1, \dots, E_k, (p \vee \neg p)$$

where p is a fresh propositional variable.
" $\Rightarrow$ ": Assume that an odd number, let it be n, of formulas among $E_1, \dots, E_k$ is satisfied. Since, $(p \vee \neg p)$ is a tautology, it follows that $n + 1$ is even. Hence, among $\tau(E_1, \dots, E_k)$ an even number of formulas is satisfied. Therefore, one can conclude that $\tau(E_1, \dots, E_k)$ is an accepted instance of **EVENSAT**.
" $\Leftarrow$ ": Assume that an even number, let it be n, of formulas among $\tau(E_1, \dots, E_k)$ is satisfied. Since, $(p \vee \neg p)$ is a tautology, there is at least one formula satisfied. Hence, $n > 0$, and due to the fact that n even it follows that $n > 1$. Meaning among $E_1, \dots, E_k$ there is at least one formula satisfied. With the corner case covered, the general argument is that since n one obtains through $(p \vee \neg p)$ being a tautology, that $n = m + 1$ with m odd. Therefore, one can conclude that an odd number of formulas among $E_1, \dots, E_k$ is satisfied, i.e. they are an accepting instance of **ODDSAT**.

(b) Show that **EVENSAT** $\leq_m^p$ **CF**$_D$ by providing a polynomial-time transformation; without loss of generality, you may assume that satisfiability of E_i implies satisfiability of E_{i+1}, for all $1 \leq i < n$.
Firstly, with out loss of generality, assume that k is even. This can be done since one can simply add an unsatisfiable formula at the beginning of all formulas and shift the index, e.g. $E_1, \dots, E_k$ with k odd, to $E'_1, \dots, E'_{k+1} = (p \wedge \neg p), E_1, \dots, E_k$.
Secondly, without loss of generality, assume that the variables of each formula are pairwise disjoint, i.e. $\forall i, j \in \{1, \dots, k\}\ i \neq j \Rightarrow \mathit{Var}(E_i) \cap \mathit{Var}(E_j) = \emptyset$.

Consider the following transformation $\tau(E_1, \dots, E_k) = (T, \psi, \varphi)$ where (T, ψ, φ) is an input instance for $T \circ_D \psi \models \varphi$ and where

$$T := \{c_1, \dots, c_k\}$$
$$\psi := \bigwedge_{i \in \{1,\dots,k\}} c_i \to E_i$$
$$\varphi := c_1 \vee \Big(\bigvee_{i \in \{1,\dots,\frac{k}{2}-1\}} (\neg c_{2i} \wedge c_{2i+1}) \Big) \vee \neg c_k$$

Firstly, this transformation only increases the input in a linear manner with respect to the number of clauses. Thus this transformation can be done in polynomial time.

" $\Rightarrow$ ": Assume that an even number, let it be n, of formulas among $E_1, \dots, E_k$ is satisfied. *Case 1*, all formulas are satisfied. As established in the lecture all models in $Mod(T \circ_D \psi)$ satisfy the maximum number of c_i's. In this case all of them must be satisfied. Hence, under all models c_1 and thus φ is satisfied. *Case 2*, no formula is satisfied. In this case no c_i is satisfied, i.e. the only model of ψ is the one that is empty. Hence, said model is the only one of $T \circ_D \psi$, indecently this satisfies $\neg c_k$ and thus also φ. *Otherwise*, there exists at least one formula that is not satisfied and at least one that is. Hence, given the input restriction, there exists an m such that E_m is not satisfied and E_{m+1} is satisfied. Every model $\mathcal{M} \in Mod(T \circ_D \psi)$ $\mathcal{M}$, deviates as little as possible from the single model $\{c_1, \dots, c_k\}$ of T. Hence, it maximises the number of c_i's it contains. However, c_i can only be satisfied if E_i is. Now, knowing that all E_i with $i \leq m$ are not satisfiable, the corresponding c_i can not be part of the model. Therefore, it follows for every model $\mathcal{M} \in Mod(T \circ_D \psi)$ $\mathcal{M}$ $\mathcal{M} \models \neg c_i$ for $i \leq m$ and $\mathcal{M} \models c_i$ for $i > m$. Meaning that m is the same in all models and must be even, i.e. $m = 2j$. Thus $\neg c_{2i} \wedge c_{2i+1}$ is satisfied in all models. Thus one obtains $T \circ_D \psi \models \varphi$.

" $\Leftarrow$ ": Assume that an odd number, let it be n, of formulas among $E_1, \dots, E_k$ is satisfied. Given the assumptions made above, neither the case where all formulas nor the case were no formula is satisfied apply have to be considered. Hence, there exists at least one formula that is not satisfied and at least one that is. Hence, given the input restriction, there exists an m such that E_m is not satisfied and E_{m+1} is satisfied. Clearly, this m is odd, i.e. $m = 2j + 1$ for some j. Moreover, given the input restriction it is known that for all $i \leq m$ E_i is unsatisfiable and that for all $m < i \leq k$ E_i is satisfiable. Since, every model $Mod(T \circ_D \psi)$ maximises the number of c_i's, it follows that for all $i \leq m$ c_i evaluates to 0 and that for all $m < i \leq k$ c_i evaluates to 1 in every model. Hence, in every model of $Mod(T \circ_D \psi)$, the transition from unsatisfied formulas to satisfied formulas occurs on c_{2j+1} and $c_{2\cdot(j+1)}$. Hence, no model can satisfy $c_1 \vee \big(\bigvee_{i \in \{1,\dots,\frac{k}{2}-1\}} (\neg c_{2i} \wedge c_{2i+1}) \big) \vee \neg c_k$. Therefore, $T \circ_D \psi \not\models \varphi$.

(c) Bonus: show that the "without loss of generality" assumption in item (b) holds.

Here unfortunately an somewhat indirect answer. Meaning that by giving an algorithm that establishes $\Delta_2^P[O(log)]$-membership of the unrestricted case, then using the fact that the restricted case is $\Delta_2^P[O(log)]$-complete to show completeness of the general case and thereby establish that those problems can be used interchangeably. Hence, consider the following algorithm

```
EVENSAT((E_1, ..., E_k)):
    Q := (λx_1 ... λx_k(SAT(x_i))_{i∈{1,...,k}}) // (1)
    Q_a := Q (E_1, ... E_k) // (2)
    if ∑_{a_i∈Q_a} a_i is even:
        return True
    else:
        return False
```

In step (1) k independent queries to a **SAT**-oracle, returning 1 if the instance is satisfiable and 0 otherwise, are set up. Then, those prepared queries are executed in parallel on the formulas $(E_1, \dots, E_k)$. With a linear amount of parallel oracle calls and with all remaining steps being done in linear time as well, this algorithm establishes $\mathbf{P}_{||}^{\mathbf{NP}}$-membership. However, since this class coincides with the class $\Delta_2^P[O(log)]$ the membership can be transferred. Now knowing that the restricted case is $\Delta_2^P[O(log)]$-complete, completeness of the general case follows in suit. Hence, there must exist a reduction of the form $\mathbf{EVENSAT}_G \leq_m^p \mathbf{EVENSAT}_R$, i.e. general to restricted case. Hence, the reduction in the previous exercise suffices.