1. Variants de 3-SAT

i) Demostreu que la versió restringida de 3SAT que anomenem 3SAT-Twice at most i que tot seguit definim, és NP-complet:

> Donada una fórmula booleana F en forma normal conjuntiva, amb exactament 3 literals per clàusula i en la que cada literal apareix com a màxim 2 vegades, decidir si F és satisfactible.

Per demostrar que *3SAT-Twice at most* és NP-complet reduirem 3SAT (que sabem que és NP-complet) a *3SAT-Twice at most*. A continuació expliquem la reducció:

<u>Transformació d'una instància de 3SAT a *3SAT-Twice at most*</u>

Considerem una expressió 3SAT general en la qual la variable x apareix k vegades. Reemplaçarem la primera aparició de x per x_1, la segona per x_2, i així successivament, on x_1, x_2, ..., x_k són k noves variables. D'aquesta manera només apareixerà un literal de cada variable x_i, sent un literal positiu o negatiu depenent del signe de la variable x en l'aparició i.

Afegirem $(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge \cdots \wedge (\neg x_k \vee x_1)$ a F, això és equivalent a $x_1 \Rightarrow x_2 \Rightarrow \cdots \Rightarrow x_k \Rightarrow x_1$.

A l'afegir aquestes clàusules incrementarem una vegada el literal positiu x_i i una vegada el literal negatiu $\neg x_i$. Juntament amb la substitució de x per x_i, com a màxim tindrem dues aparicions per literal.

Com aquestes noves clàusules contenen només dos literals, afegirem una nova variable a la clàusula i mitjançant una variable auxiliar forçarem a que sigui falsa, per satisfer la condició que cada clàusula ha de tenir 3 literals:

$(\neg x_1 \vee x_2 \vee y_1) \wedge (\neg x_2 \vee x_3 \vee y_2) \wedge \cdots \wedge (\neg x_k \vee x_1 \vee y_k) \wedge (\neg y_1 \vee aux_1 \vee aux_1) \wedge (\neg y_1 \vee \neg aux_1 \vee \neg aux_1) \wedge \cdots \wedge (\neg y_k \vee aux_k \vee aux_k) \wedge (\neg y_k \vee \neg aux_k \vee \neg aux_k)$.

La fórmula booleana resultant satisfà la condició per la variable x, si l'apliquem a cadascuna de les variables de F obtindrem una fórmula booleana en forma *3SAT-Twice at most*. Veiem com aquest procediment es pot fer en temps lineal respecte el nombre de variables.

<u>Demostrar que *3SAT-Twice at most* és NP</u>

En efecte, podem verificar en temps polinòmic que un certificat de mida polinòmica al problema de *3SAT-Twice at most* és solució.

En aquest cas el certificat de *3SAT-Twice at most* té la mateixa estructura que un certificat per 3SAT: per cadascuna de les variables a F assignem un valor de true o false. Aquest certificat té mida Θ(n) on n és el nombre de variables, i podem verificar la solució en temps polinòmic tal com es pot verificar en el cas de 3SAT o SAT.

<u>Conclusió</u>

Com hem reduït 3SAT a *3SAT-Twice at most* i hem demostrat que *3SAT-Twice at most*, podem concloure que *3SAT-Twice at most* és un problema NP-complet.

ii) En canvi, quan restringim les fórmules d'entrada de manera que cada literal aparegui com a màxim una vegada, el problema es pot resoldre en temps polinòmic. Formalment, definim *3SAT-once at most* de la manera següent:

> Donada una fórmula booleana F en forma normal conjuntiva, amb exactament 3 literals per clàusula i en la que cada literal apareix com a màxim una vegada, decidir si F és satisfactible.

Demostreu que *3SAT-once at most* $\in$ P.

Si cada literal de *3SAT-once at most* apareix com a màxim una vegada, podem aplicar el següent procediment per donar una assignació a qualsevol variable *A* i anar ignorant clàusules:

- Si *A* només apareix a una clàusula *c* com *a*, assignem *A* = true i ignorem *c*.
- Si *A* només apareix a una clàusula *c* com $\neg a$, assignem *A* = false i ignorem *c*.
- Si *A* apareix com *a* i $\neg a$ en una mateixa clàusula *c*, llavors podem assignar *A* tant true com false, i ignorem *c*.

Ara utilitzem la regla de resolució per anar eliminant les variables una per una. Com que cada variable apareix exactament una vegada de forma positiva i una vegada de forma negativa, aquest procés és determinista.

Si en qualsevol moment obtenim la clàusula buida, el conjunt de clàusules no és satisfactible i per tant cal retornar fals. En cas contrari F és satisfactible i podem assignar qualsevol valor a les variables eliminades.

- Si *A* apareix en dues clàusules diferents, per exemple ($a \cup b \cup c$) i ($\neg a \cup d \cup e$), apliquem la regla de resolució per transformar aquestes dues clàusules en una única clàusula ($b \cup c \cup d \cup e$):

$$\frac{a_1 \vee a_2 \vee \cdots \vee c, \quad b_1 \vee b_2 \vee \cdots \vee \neg c}{a_1 \vee a_2 \vee \cdots \vee b_1 \vee b_2 \vee \cdots}$$

Aquest algorisme requerirà temps polinòmic ja que cada variable es resol exactament una vegada. En particular, cada aplicació de la resolució reduirà el nombre total de clàusules en una, de manera que el nombre total de passos de resolució és O(n), on *n* és el nombre de variables.