NAE-3-SAT

NAE = Not All Equal

Similar cu 3-SAT, dar cu diferența că se adaugă următoarea regulă: o clauză nu are voie să aibă toate valorile egale între ele (i.e.: trebuie să aibă minim o valoare **True** și minim o valoare **False**).

Pentru simplitate, vom presupune în explicațiile de mai jos că k-SAT are *EXACT* k literali, nu *cel mult* k.

La fel ca 3-SAT, NAE-3-SAT este **NP-Complete**!

Notație: **A $\leq_p$ B** înseamnă ca A se poate reduce la B în timp polinomial.

NAE-4-SAT este la fel ca NAE-3-SAT, cu diferența că fiecare clauză are 4 literali în loc de 3 (evident).

Propoziţie: 3-SAT $\leq_p$ NAE-4-SAT

Să reducem 3-SAT la NAE-4-SAT. Considerăm o instanţă a 3-SAT notată **ϕ**. Adăugăm la fiecare clauză o variabilă **S** (aceeași la toate clauzele) și numim această instanţă **ϕ'**. Exemplu:

$$(x_3 \vee \bar{x_5} \vee \bar{x_5}) \rightarrow (x_3, \bar{x_5}, \bar{x_5}, S)$$

Evident, **ϕ'** este o instanţă de 4-SAT. Impunem pentru **ϕ'** condiția să aibă în fiecare clauză minim o valoare **True** și minim una **False**, pentru a face **ϕ'** o instanţă de NAE-4-SAT.

O poză cu Minnie ca să alunge plictiseala :)

Propoziție: **ϕ satisfiabilă (ca 3-SAT) ⇔ ϕ' satisfiabilă (ca NAE-4-SAT).**

Reminder că A⇔B înseamnă că A implică B și B implică A.

Explicație de ce **ϕ satisfiabilă (3-SAT)** implică **ϕ' satisfiabilă (NAE-4-SAT):**

Dacă avem o soluție **X** pentru **ϕ**, atunci **X ∪ (S=False)** este soluție pentru **ϕ'** deoarece:

- dacă formula este satisfăcută indiferent de literalul **S**, nu mai contează ce valoare îi dăm lui **S**, formula va fi satisfăcută oricum
- dacă formula este satisfăcută, fiecare clauză are minim o valoare True. Atribuind **S=False**, garantăm că fiecare clauză are minim o valoare **True** și minim o valoare **False** => satisface NAE-SAT

Explicație de ce **ϕ' satisfiabilă (NAE-4-SAT)** implică **ϕ satisfiabilă (3-SAT)**:

Dacă avem o soluție **(X,S)** - **S** fiind atribuirea pentru **S**, iar **X** atribuirile pentru celelalte variabile - pentru **ϕ'**, atunci:

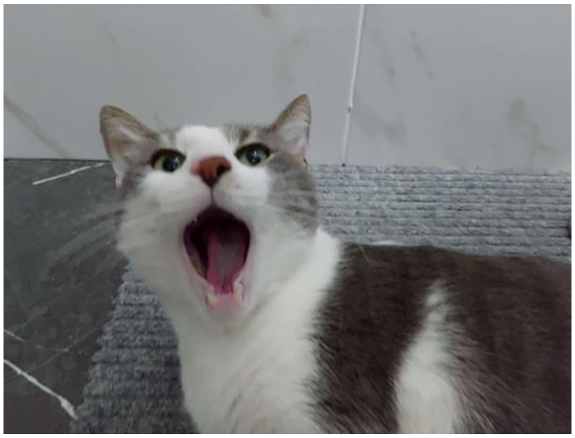
- dacă **S=False**, atunci **X** satisface **ϕ** (la fel ca mai sus, **S** nu schimbă nimic)
- dacă **S=True**, atunci **¬X ∪ (S=False)** satisface **ϕ'** de asemenea: fiind **Not-All-Equal**, fiecare clauză are minim o valoare **False**. Negând toate valorile din soluție, obținem în fiecare clauză minim o valoare **True** (=> formula este satisfăcută). Setând **S=False**, garantăm că avem în toate clauzele și o valoare **False**. Acum că am găsit o soluție **(¬X,S=False)** pentru **ϕ'**, suntem în cazul de la prima liniuță, deci **¬X** satisface **ϕ**

Folosind proprietate că o clauză **(a, b, c, d)** este echivalentă cu **(a,b,x)∧(¬x,c,d)**, se poate demonstra că **NAE-4-SAT $\leq_p$ NAE-3-SAT**.

Așadar, până acum știm:

- **3-SAT $\leq_p$ NAE-4-SAT**
- **NAE-4-SAT $\leq_p$ NAE-3-SAT**

De aici putem deduce că **3-SAT $\leq_p$ NAE-3-SAT** (3-SAT se poate reduce polinomial la NAE-3-SAT). Deoarece știm că 3-SAT este NP-Complete, arătând că aceasta se reduce la NAE-3-SAT demonstrează că NAE-3-SAT este de asemenea NP-Complete!



*Mișu a simțit că nu ai dat importanță ultimelor noțiuni, dar vrea să-ți atragă atenția că e drăguț (și suficient) să reții că **3-SAT se reduce în timp polinomial la NAE-3-SAT via NAE-4-SAT**! :)*

Bonus fact: NAE-2-SAT se reduce la 2-COLORING!