On a donc transformé les 2 propriétés sur G intrinsèques aux instances positives de STABLE, à savoir,

* Il doit exister un sous-ensemble de sommets de S, X, pour lequel tous ses sommets ne sont pas adjacents ;
* X doit contenir au moins k sommets de S ;

en autres 2 propriétés sur G :

* Chaque sommet de S est soit un des k sommets de X, qui n’est aucun autre sommet de S, soit n’est pas dans X ;
* Si un sommet est dans X, aucun de ses adjacents ne l’est.

Ces propriétés peuvent être plus facilement traduite dans un formalisme propositionnel :

On associe à chaque sommet x une variable propositionnelle px, qui indiquera s’il est ou non dans un stable solution.

* De cette façon, si y est un adjacent de x, alors (px => -py && py => -px), ce qui se peut se traduire par une clause, que l’on appellera *clause d’adjacence*,

(-px ˅ -py).

* L’attribution des k sommets de X à des sommets de S peut se voir comme l’attribution d’une position dans le stable : ainsi, si x = , x sera le i-ème sommet de X, et aucun autre sommet de S ne sera à la i-ème position dans X.

On peut ainsi traduire cela selon la logique propositionnelle, en un ensemble de conjonction de clauses, que l’on appellera *conjonction de positions* :

Soient x0 la variable propositionnelle indiquant que x n’est pas dans X, et x1,…,xk les variables propositionnelles indiquant que x est à la i-ème position, et z n’importe quel autre sommet du graphe :

(-x0 ˅ -x1) ˄ (-x0 ˅ -x2) ˄ … ˄ (-x0 ˅ -xk) //Une seule position par sommet

˄ (-x0 ˅ -z0) ˄ (-x1 ˅ -z1) ˄ … ˄ (-xk ˅ -zk) //Un seul sommet par position ˄ (x0 ˅ z0) ˄ (x1 ˅ z1) ˄ … ˄ (xk ˅ zk) //Toutes les positions doivent être //attribuées

En combinant ainsi ces deux types de clauses, on peut donc former une formule normale conjonctive que l’on pourra passer au problème SAT qui formalisera bien le problème STABLE, qui sera de la forme :

F = *clauses d’adjacences* ˄ *conjonction de positions*