

SAE S1.02

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

○ *Formule* =  $[c_1, c_2, c_3, c_4, c_5, c_6]$  avec par exemple :

▪  $c_2 = [-1, 3, -4]$  et  $c_3 = [-2, 4]$

○ Si on rajoute une contrainte fixée par ailleurs comme le fait que la variable *b* doive prendre la valeur *False* on va devoir considérer l'ensemble des valuations issues de  $list\_var = [None, False, None, None]$  où la valeur *None* désigne le fait que la valeur est indéterminée et qu'elle peut donc prendre les deux valeurs logiques (attention les opérateurs logiques de python ne sont pas toujours compatibles avec cette idée et la valeur *None* de python)

# *evaluer\_clause* et *evaluer\_cnf*

- *evaluer\_clause*( $[-1, 3, -4]$ ,  $[True, False, True, None]$ ) =
- *evaluer\_clause*( $[-1, 3, -4]$ ,  $[True, False, None, None]$ ) =
- *evaluer\_clause*( $[-1, 3, -4]$ ,  $[True, False, False, True]$ ) =
- *evaluer\_clause*( $[\ ]$ ,  $[True, False, True, True]$ ) =
- *evaluer\_cnf* appelle les différents résultats renvoyés par *evaluer\_clause* pour les différentes clauses. Par exemple pour une formule à trois clauses on a :

$$\circ \quad \overbrace{True}^{ev\_cl(c\_1, li\_va)} \quad \wedge \quad \overbrace{None}^{ev\_cl(c\_2, li\_va)} \quad \wedge \quad \overbrace{True}^{ev\_cl(c\_3, li\_va)} \quad \xrightarrow{ev\_cnf([c\_1, c\_2, c\_3], li\_va)}$$

$$\circ \quad \overbrace{True}^{ev\_cl(c\_1, li\_va)} \quad \wedge \quad \overbrace{None}^{ev\_cl(c\_2, li\_va)} \quad \wedge \quad \overbrace{False}^{ev\_cl(c\_3, li\_va)} \quad \xrightarrow{ev\_cnf([c\_1, c\_2, c\_3], li\_va)}$$

# *evaluer\_clause* et *evaluer\_cnf*

- $evaluer\_clause([-1, 3, -4], [True, False, True, None]) = True$
- $evaluer\_clause([-1, 3, -4], [True, False, None, None]) = None$
- $evaluer\_clause([-1, 3, -4], [True, False, False, True]) = False$
- $evaluer\_clause([], [True, False, True, True]) = False$
- *evaluer\_cnf* appelle les différents résultats renvoyés par *evaluer\_clause* pour les différentes clauses. Par exemple pour une formule à trois clauses on a :

$$\circ \quad \overbrace{ev\_cl(c\_1, li\_va)}^{True} \quad \wedge \quad \overbrace{ev\_cl(c\_2, li\_va)}^{None} \quad \wedge \quad \overbrace{ev\_cl(c\_3, li\_va)}^{True} \quad \xrightarrow{ev\_cnf([c\_1, c\_2, c\_3], li\_va)} None$$

$$\circ \quad \overbrace{ev\_cl(c\_1, li\_va)}^{True} \quad \wedge \quad \overbrace{ev\_cl(c\_2, li\_va)}^{None} \quad \wedge \quad \overbrace{ev\_cl(c\_3, li\_va)}^{False} \quad \xrightarrow{ev\_cnf([c\_1, c\_2, c\_3], li\_va)} False$$

# *determine\_valuations(list\_var)*

- *determine\_valuations([None, False, None, None])*
- Init.  $[[None, False, None, None]]$  puis boucle sur différentes var
- $i = 0$  :  $[[True, False, None, None]] + [[False, False, None, None]] = [[True, False, None, None], [False, False, None, None]]$
- $i = 1$  : Aucun changement
- $i = 2$  :  $[[True, False, True, None], [False, False, True, None]] + [[True, False, False, None], [False, False, False, None]] = [[True, False, True, None], [False, False, True, None], [True, False, False, None], [False, False, False, None]]$
- $i = 3$  : ...

*resol(for, [None, False, None, None], [ ])*

*progress*

*resol(for, [True, False, None, None], [[0, True]])*

*progress*

*resol(for, [True, False, True, None],  
[[0, True], [2, True]])*

*retour*

*resol(for, [True, False, False, None],  
[[0, True], [2, False]])*

*progress*

*retour*

*resol(for,  
[True, False,  
True, True],  
[[0, True],  
[2, True], [3, True]])*

*resol(for,  
[True, False,  
True, False],  
[[0, True],  
[2, True], [3, False]])*

# Exemple numérotation variables pour une grille taille 2

Variables 1 à 4	Variables 5 à 8	Variables 9 à 12	Variables 13 à 16
Variables 17 à 20	Variables 21 à 24	Variables 25 à 28	Variables 29 à 32
Variables 33 à 36	Variables 37 à 40	Variables 41 à 44	Variables 45 à 48
Variables 49 à 52	Variables 53 à 56	Variables 57 à 60	Variables 61 à 64