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 $evaluer\_cnf$

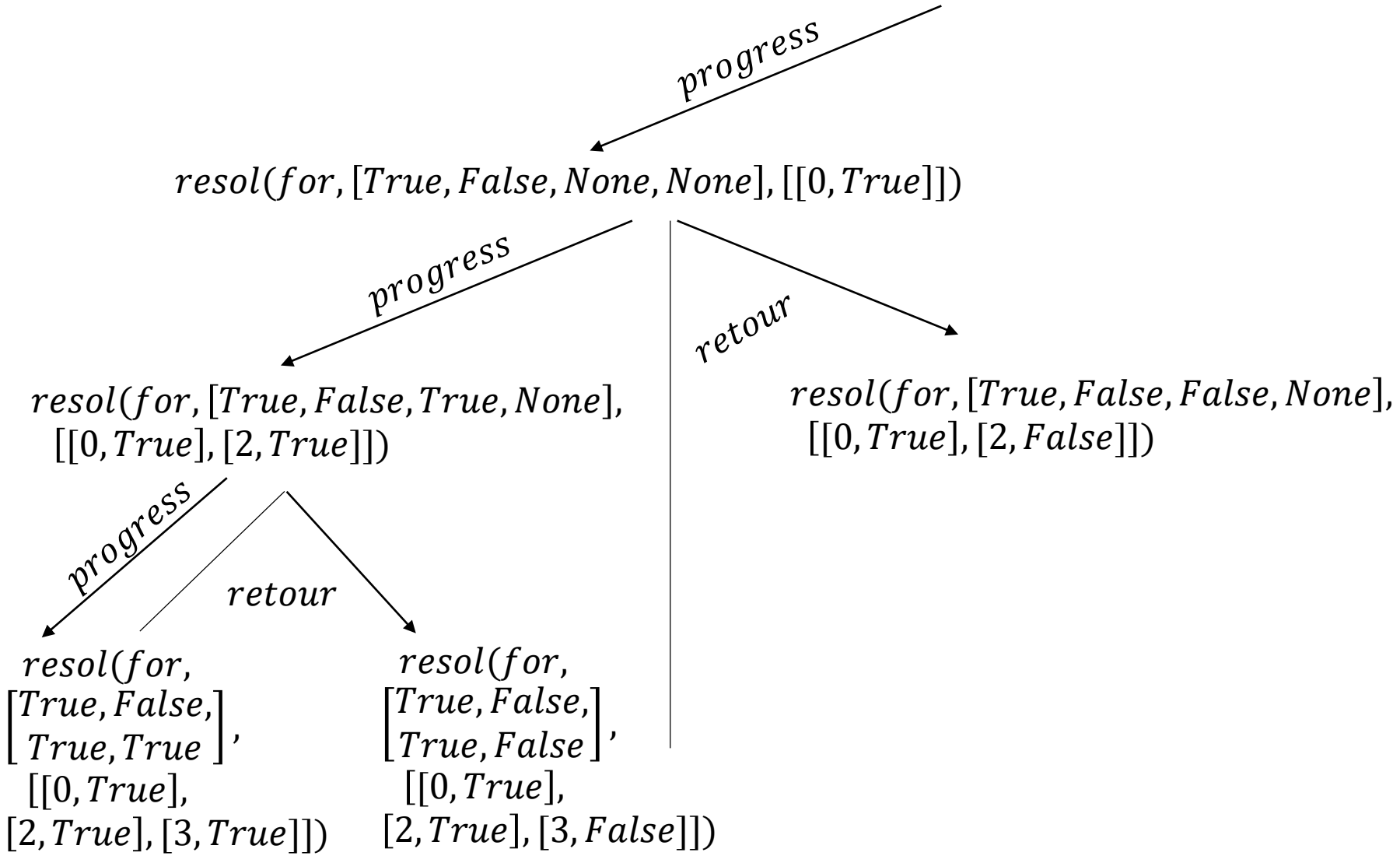
- $evaluer\_clause([-1,3,-4],[True,False,True,None]) =$
- $evaluer\_clause([-1,3,-4],[True,False,None,None]) =$
- $evaluer\_clause([-1,3,-4],[True,False,False,True]) =$
- $evaluer\_clause([\,],[True,False,True,True]) =$
- $evaluer\_cnf$ appelle les différents résultats renvoyés par $evaluer\_clause$ pour les différentes clauses. Par exemple pour une formule à trois clauses on a :
  - $\overbrace{True}^{ev\_cl(c\_1,li\_va)} \wedge \overbrace{None}^{ev\_cl(c\_2,li\_va)} \wedge \overbrace{True}^{ev\_cl(c\_3,li\_va)} \xrightarrow{ev\_cnf([c\_1,c\_2,c\_3],li\_va)}$
  - $\overbrace{True}^{ev\_cl(c\_1,li\_va)} \wedge \overbrace{None}^{ev\_cl(c\_2,li\_va)} \wedge \overbrace{False}^{ev\_cl(c\_3,li\_va)} \xrightarrow{ev\_cnf([c\_1,c\_2,c\_3],li\_va)}$

$evaluer\_clause$ 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 :
  - $\overbrace{True}^{ev\_cl(c\_1,li\_va)} \wedge \overbrace{None}^{ev\_cl(c\_2,li\_va)} \wedge \overbrace{True}^{ev\_cl(c\_3,li\_va)} \xrightarrow{ev\_cnf([c\_1,c\_2,c\_3],li\_va)} None$
  - $\overbrace{True}^{ev\_cl(c\_1,li\_va)} \wedge \overbrace{None}^{ev\_cl(c\_2,li\_va)} \wedge \overbrace{False}^{ev\_cl(c\_3,li\_va)} \xrightarrow{ev\_cnf([c\_1,c\_2,c\_3],li\_va)} \text{False}$

$determine\_valuations(list\_var)$

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


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