

Subset Sum

Soient $x_1, x_2, \dots, x_n, B \in \mathbb{N}$

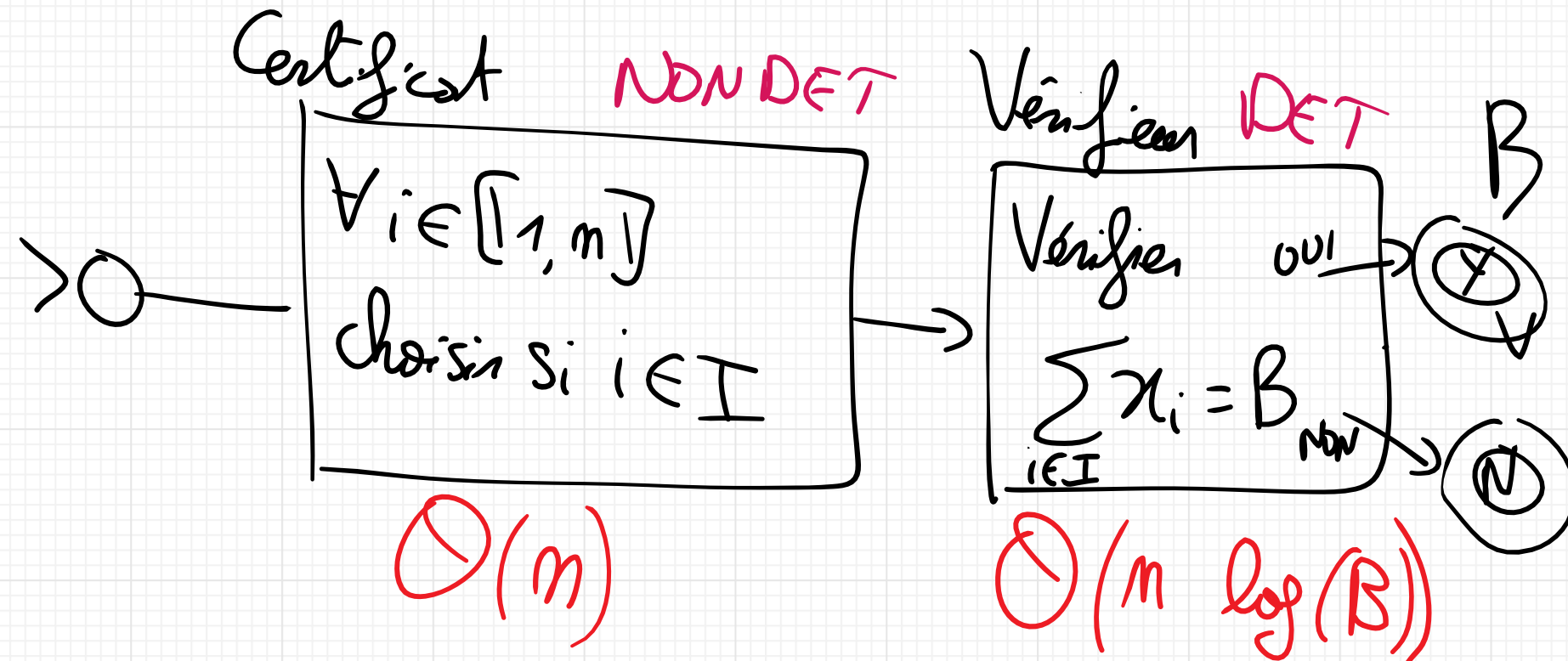
$$\exists? I \subseteq [1, n] / \sum_{i \in I} x_i = B$$

Encodage

(x_1)	0 0 0 ... 1 0 0 1	↑ n+1 ↓
(x_2)	0 0 1 ... 1 0 1 0	
(x_3)	0 0 1 ... 0 1 1 0	
$\vdots$		
(x_n)	0 0 0 ... 1 1 0	
(B)	1 1 0 1 ... 0 0 1	← log(B) →

Non! $|I| \neq n+1 \times \log(B)$
si $\forall i, x_i \leq B$

Ex: $x = [1, 7, 10, 20, 32, 37]$ $B = 25$



$B = 49$ oui $7 + 10 + 32$
 $I = \{2, 3, 5\}$

Preuve $3SAT \leq SETCOVER$

Soit $(x_1 x_2 \dots x_m, C_1 C_2 \dots C_m)$ une instance de 3SAT
 $\varphi = C_1 \wedge C_2 \wedge C_3 \dots \wedge C_m$

Réduction R : on construit l'instance $J = (X, S, k)$ de SET COVER :

$O(m)$ • $\forall i \in [1, m]$ rajouter un élément ~~x_i~~ dans X et 2 ensembles $x_i, \overline{x_i}$ dans S

$O(m)$ • $\forall j \in [1, m]$ rajouter un élément C_j dans X

$O(m) + O(3m) = O(m+m)$
• x_i contient e_i et tout C_j tq la clause C_j contient le littéral x_i dans φ
• $\overline{x_i}$ contient e_i et tout C_j tq la clause C_j contient le littéral $\overline{x_i}$ dans φ

$O(\log(m))$ • $k = m$

La réduction se fait en temps $O(m+m)$ polynomial en $|\varphi| = m+m$

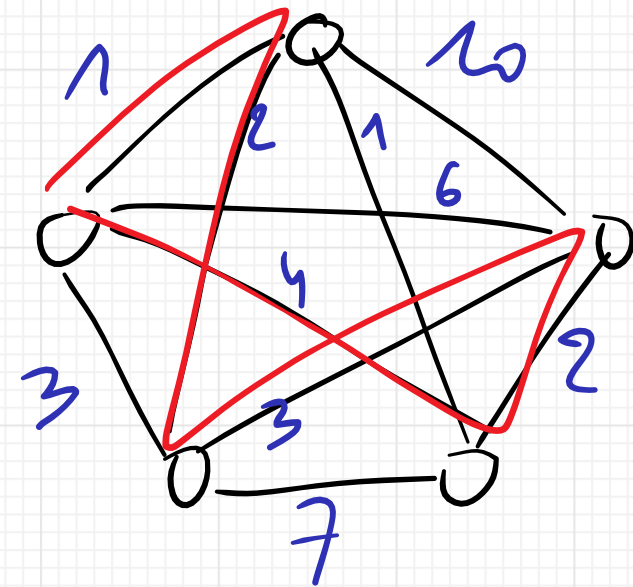
OPT1 $\leq$ "TP Reductions et complétude"
par la Réduction $3SAT \leq SUBSET SUM$

M₉ TSP NP-Comple

(TSP): Soit $G = (V, E)$ un graphe non orienté complet
et $w: E \rightarrow \mathbb{N}$, $k \in \mathbb{N}$

$\exists$? un cycle hamiltonien de G de poids $\leq k$?

M₉ (TSP) $\in$ NP et $\left(\begin{smallmatrix} \text{cycle} \\ \text{Hamiltonien} \end{smallmatrix} \right) \leq (TSP)$ (Cycle Hamiltonien)



$\text{My } \phi \text{ satisfiable } \Leftrightarrow \exists \overbrace{s_1 s_2 \dots s_j}^{\epsilon}, j \leq k / \bigcup_{i=1}^j s_i = X$
 (J instance positive)

$\Rightarrow \exists a_1 a_2 \dots a_n \in \{T, \perp\}$

tq si on affecte a_i à α_i
 $\phi \text{ VRAI} \Leftrightarrow \forall j, C_j \text{ VRAI}$

On construit une solution C ob J

ainsi : $\alpha_i \in C$ si $a_i = T$
 $\overline{\alpha_i} \in C$ sinon

• on a n ensembles dans C : $|C| = n = k$

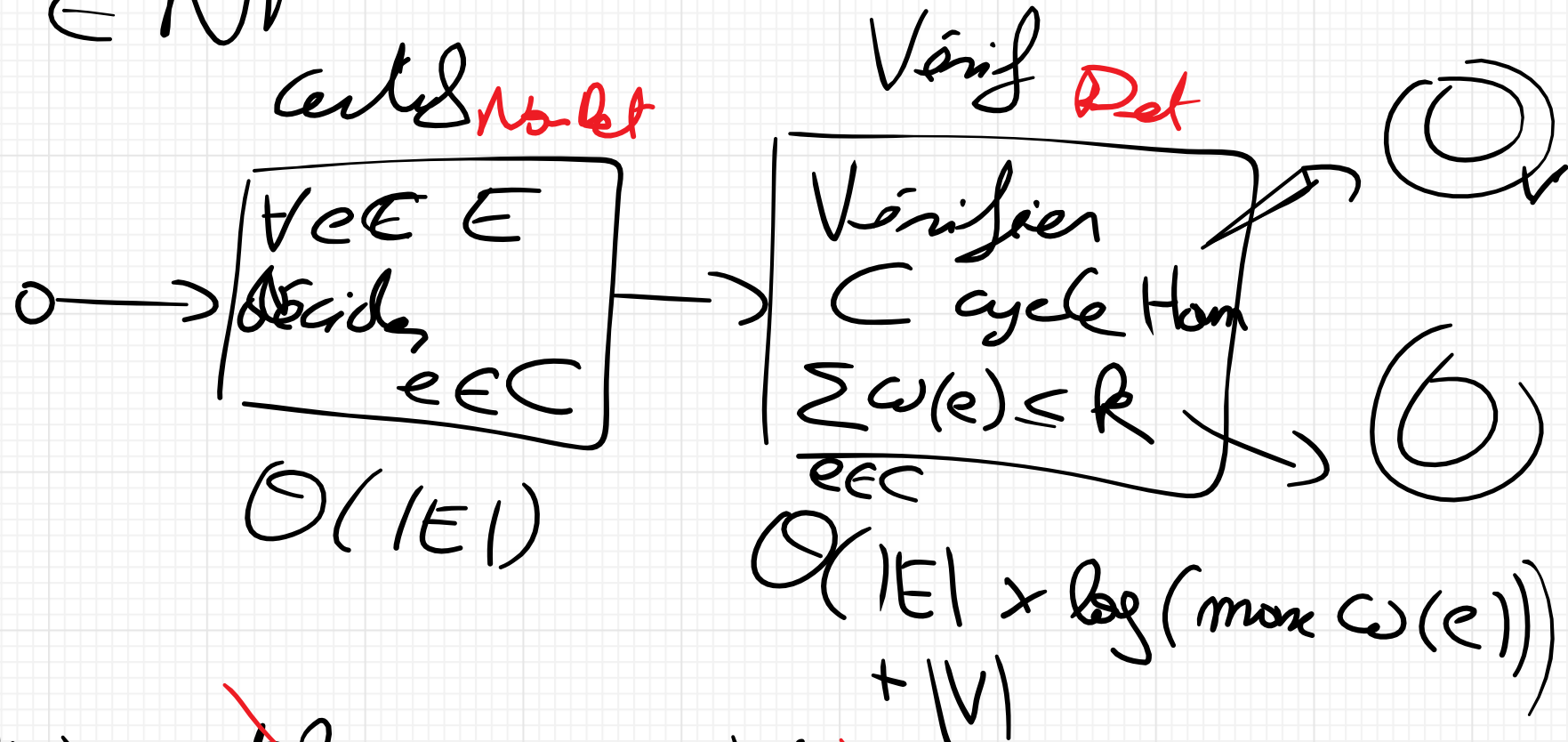
• $\forall i \in [1, n] \quad \alpha_i \in \overline{\alpha_i} \text{ donc } \alpha_i \in C$

• $\forall j \in [1, n] \quad C_j \in C$ car

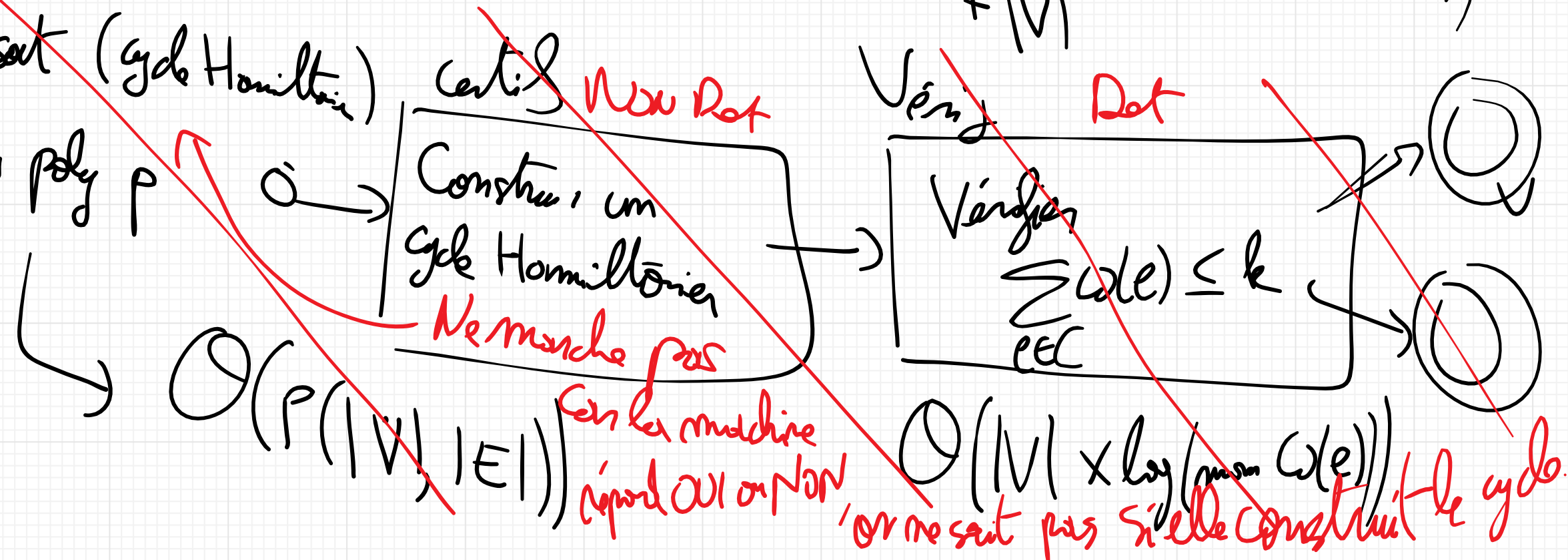
la clause $C_j = (l_1 \vee l_2 \vee l_3)$ est VRAI
 donc $l_1 \text{ VRAI}$ ou $l_2 \text{ VRAI}$ ou $l_3 \text{ VRAI}$

Supposons que $l_1 \text{ VRAI}$ et $l_1 = \overline{\alpha_i}$ (sans perte de généralité)
 $\Rightarrow a_i = \perp \Rightarrow \overline{\alpha_i} \in C$ or $\overline{\alpha_i} \in \overline{\alpha_i} \Rightarrow C_j \in C$

TSP $\in$ NP



~~Une machine ND
qui résout (cycle Hamiltonien)
en temps poly P~~



$\Leftarrow$ Soit C une solution de J
 $\forall i \in [1, n] \text{ } e_i \text{ couvert par } C \Rightarrow |C| \geq n$
 donc $x_i \in C$ ou $\bar{x}_i \in C$
 or $k = n \quad |C| \leq k$
 $\Rightarrow |C| = n$ et x_i et $\bar{x}_i$ ne peuvent appartenir tous les deux à C

Construit une affectation $a_1 a_2 \dots a_n$ des variables de Φ
 ainsi : $a_i = T \Leftrightarrow x_i \in C$

La clause C_j est vérifiée par cette affectation car :

$$C_j = (l_1 \vee l_2 \vee l_3)$$

C_j est couvert par C

or, les seuls ensembles contenant C_j

Sont l_1 l_2 et l_3

donc $l_1 \in C$ ou $l_2 \in C$ ou $l_3 \in C$

donc le littéral l_1 est VRAI

ou l_2 est VRAI

ou l_3 est VRAI

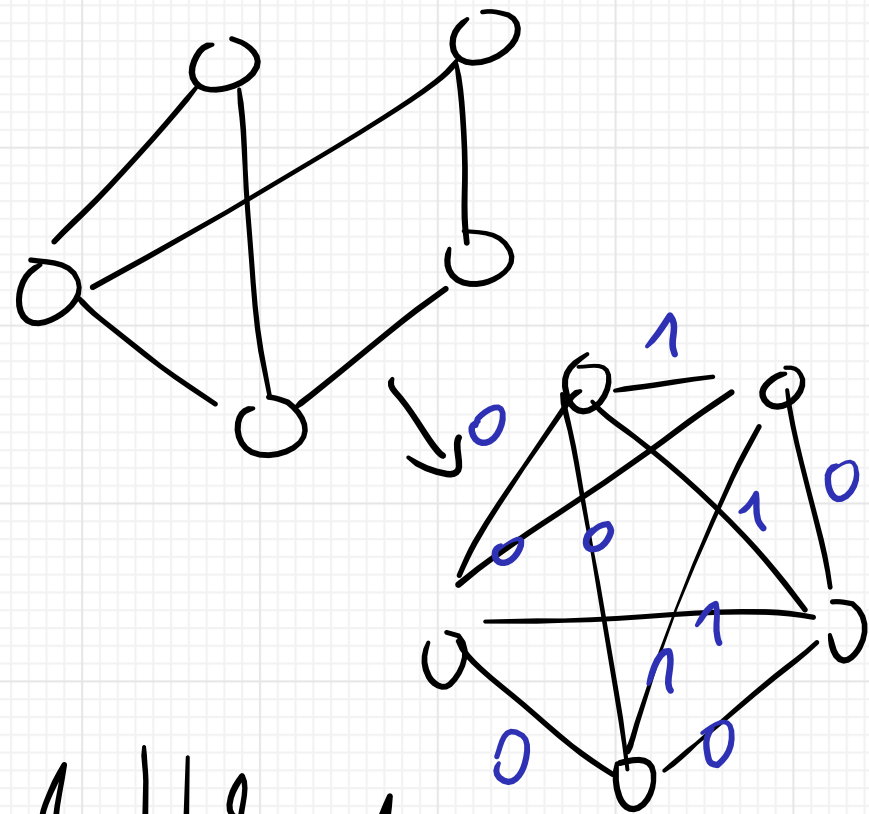
$\Rightarrow C_j \text{ VRAI}$
 $\Rightarrow \Phi \text{ VRAI}$

Soit $G = (V, E)$ un graphe, instance de (CH)
on construit une instance (H, ω, k) de (TSP):

$O(n^2)$ $H = (V, E_H)$ complet

$O(n^2)$ $\omega(e) = 0$ si $e \in E$ et 1 sinon

$O(1)$ $k = 0$



$\exists$ un cycle C hamiltonien de G $\iff \exists$ Cycle hamiltonien de H de poids 0
(n'empruntant aucune arête de $H \setminus G$)

$\Rightarrow$ est un red poly de Karp
 $\Rightarrow CH \leq TSP$
 $\Rightarrow CH \text{ NP-complet}$
 $\Rightarrow TSP \text{ NP-diff}$
 $\Rightarrow TSP \in NP$
 $\Rightarrow TSP \text{ NP-complet}$

$\iff \exists$ cycle hamiltonien de H de poids $\leq k$

Donc R est une réduction
polynomiale de Karp

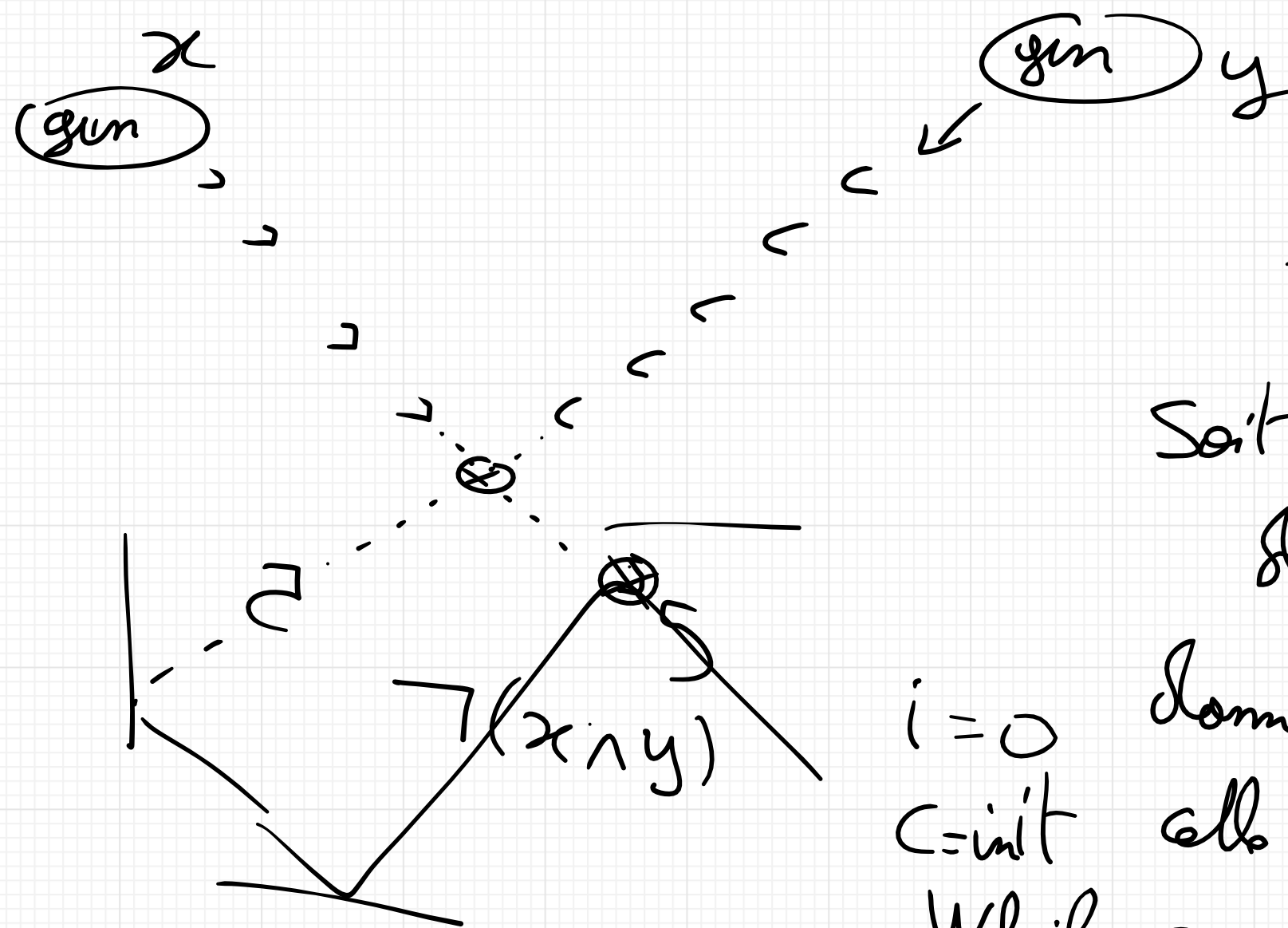
Donc $3SAT \leq SETCOVER$

ou
 $3SAT$ NP-Complet

donc $SETCOVER$ NP Difficile

ou $SETCOVER \in NP$

$\Rightarrow SETCOVER$
NP-Complet



Jeu de la vie :

Soit une config de départ et
d'arrivée, la config de départ

$i = 0$ donne t-elle à un moment donné
 $C = \text{init}$ celle d'arrivée, en k itérations?

While $C \neq \text{target}$ et $i < k$

$C \leftarrow \text{next}(C)$

$i++$