Algorithme de Quine

Motivation → Méthode de résolution simple du problème de satisfiabilité.

Définitions : On définit :
- un littéral par : $\ell ::= p \mid \neg p$, $p \in \mathcal{P}$
- une clause par un ensemble de littéraux (conjonction)
- une formule CNF par un ensemble de clauses (disjonction)

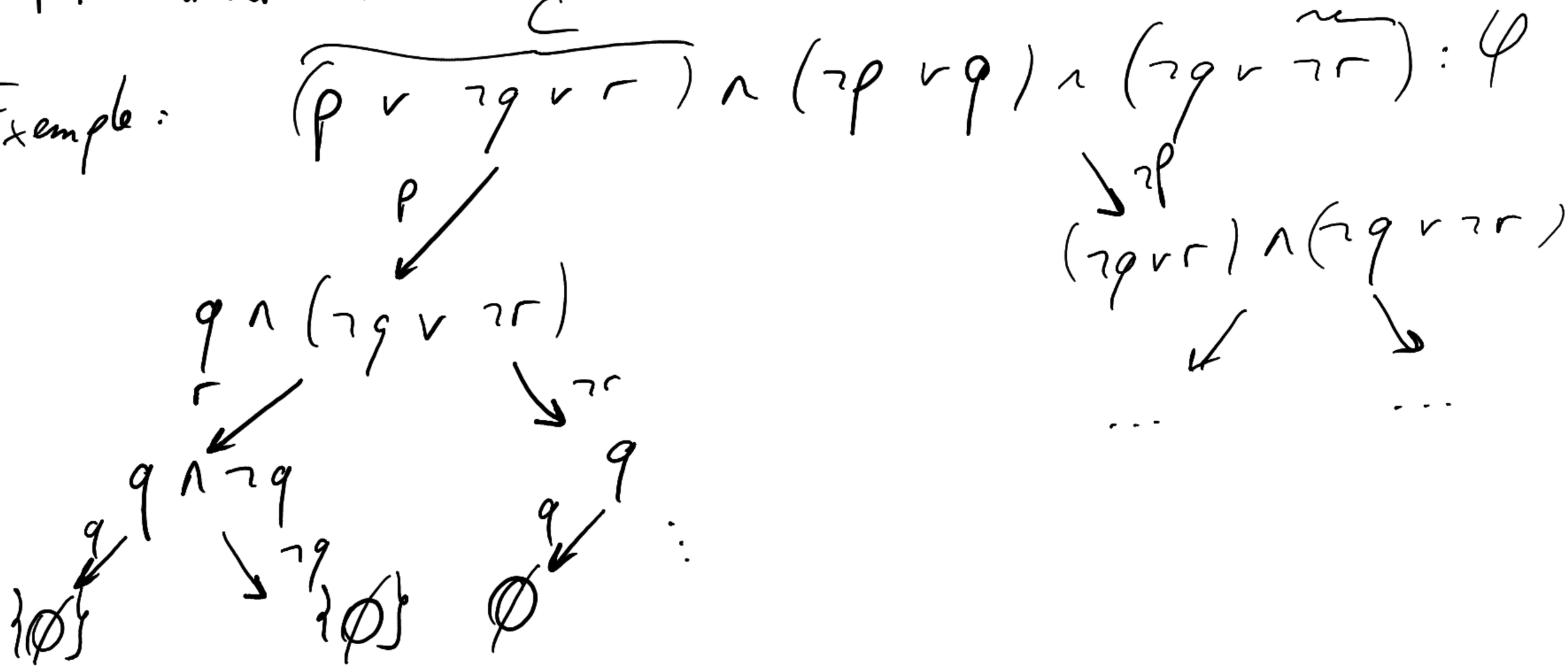
Problème : Satisfiabilité (SAT-CNF)
Entrée : formule CNF φ
Sortie : Valuation $v : \mathcal{P} \to \mathbb{B}$ telle que $v(\varphi) = V$, si elle existe

Idée : Construire un arbre de décision en remplaçant chaque variable successivement par $\top$ ou $\bot$, puis en simplifiant la formule obtenue après chaque affectation.

```
Fonction quine(φ)
    Si φ = ∅ alors Retourner V
    Sinon si ⊥ ∈ φ alors Retourner F
    Sinon
        choisir p ∈ Props(φ)
        Retourner quine(simplifier(φ, p))
                  ∨ quine(simplifier(φ, ¬p))
```

```
Fonction simplifier (Φ, ℓ)
    Ψ := ∅
    Pour C ∈ Φ :
        Si ℓ̄ ∈ C
            Ψ := Ψ ∪ (C \ {ℓ̄})
        Sinon si ℓ ∉ Φ :
            Ψ := Ψ ∪ C
    Retourner Ψ
```

Exemple : $\overbrace{(p \vee \neg q \vee r)}^{C} \wedge (\neg p \vee q) \wedge \overbrace{(\neg q \vee \neg r)}^{\ell} : \varphi$



Terminaison : $|Props(\varphi)|$ ↘↘ pour chaque appel récursif

Complexité : $O(2^n)$ dans le pire cas (SAT-CNF est NP-complet)

Correction : φ satisfiable $\Leftrightarrow$ queine (φ) renvoie V.

On définit $\forall \varphi$ formule CNF, $\forall p \in Props(\varphi)$:

$$\varphi[p \leftarrow \top] = \{ C \setminus \{\bar{p}\},\ C \in \varphi \mid \bar{p} \in C \} \cup \{ C \in \varphi \mid p \notin C \}.$$

On définit la relation $\to_p$ par :

$$\varphi \to_p \varphi[p \leftarrow \top].$$

① Lemme : p n'apparaît pas dans $\varphi[p \leftarrow \top]$ ou $\varphi[\neg p \leftarrow \top]$.

② Lemme : $I \models \varphi[p \leftarrow \top] \iff I[p \leftarrow v] \models \varphi$

(i) On veut mq : $\varphi \to^* \emptyset \iff \varphi$ satisfiable et

(ii) φ satisfiable $\Rightarrow \forall p \in Props(\varphi)$, $\varphi[p \leftarrow \top]$ ou $\varphi[\neg p \leftarrow \top]$ satisfiable.

. $\varphi \to^* \emptyset \Rightarrow \varphi$ sat : On raisonne par récurrence sur $\to^n$:

. si $\varphi \to^0 \emptyset$, alors $\varphi = \emptyset$ et $\emptyset$ sat.

. soit $n \in \mathbb{N}$ tq $\varphi \to^n \emptyset \Rightarrow \varphi$ sat. On a : $\varphi \to^{n+1} \emptyset$:

On a : $\varphi \to \varphi[\ell \leftarrow \top] \to^n \emptyset$.

Donc : HR : $\varphi[\ell \leftarrow \top]$ satisfiable : $\exists I : Props(\varphi) \to \mathbb{B}$ tq $I \models \varphi[\ell \leftarrow \top]$, d'où $I[\ell \leftarrow v] \models \varphi$: φ sat.

. φ sat $\Rightarrow \varphi \to^* \emptyset$: $\exists I : Props(\varphi) \to \mathbb{B}$ tq $I \models \varphi$.

On pose : $Props(\varphi) = \{p_1, \dots, p_n\}$.

On construit : $\varphi = \varphi_0 \to \varphi_1 \to \dots \to \varphi_n$ par :

$$\varphi_{i+1} = \begin{cases} \varphi_i[P_{i+1} \leftarrow \top] & \text{si } [\![P_{i+1}]\!]^I = V \\ \varphi_i[\neg P_{i+1} \leftarrow \top] & \text{si } [\![P_{i+1}]\!]^I = F \end{cases}$$

Par récurrence immédiate avec le lemme 2 : φ_n est satisfiable, et φ_n a 0 variables donc $\varphi_n = \emptyset$.

Donc : $\varphi \to^* \emptyset$. (i)

(ii) Soit φ formule CNF sat et $P \in Props(\varphi)$.

$\varphi \to^* \emptyset$ par (i) donc $\exists I$ val $\not\models$ $I \models \varphi$.

- Si $I(P) = V$, alors $I \models \varphi[P \leftarrow \top]$ par ②
  donc $\varphi[P \leftarrow \top]$ sat.
- Sinon, $I(\neg P) = V$ donc $I \models \varphi[\neg P \leftarrow \top]$ par ②
  donc $\varphi[\neg P \leftarrow \top]$ sat.