

# Hiérarchie polynomiale.

▮ **Définition 1.** Étant donnée une classe de langages  $\mathcal{C}$ , on définit

$$\text{co}\mathcal{C} := \{ A \subseteq \Sigma^* \mid \Sigma^* \setminus A \in \mathcal{C} \}.$$

▮ **Définition 2.** Les classes  $\Sigma_i^P$ , pour  $i \geq 0$ , sont définies par induction :

- ▷  $\Sigma_0^P := P$  ;
- ▷  $\Sigma_{i+1}^P := NP^{\Sigma_i^P}$ .

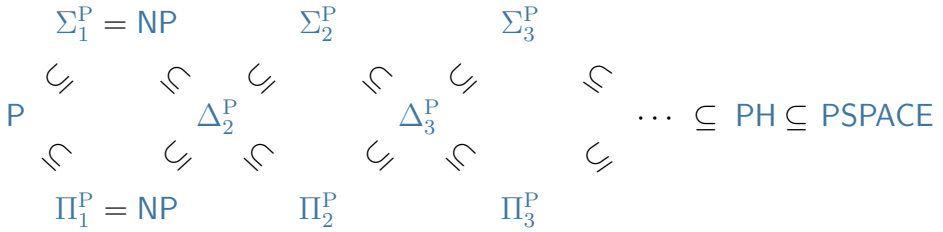
On pose  $PH := \bigcup_{i \geq 0} \Sigma_i^P$ .

On définit aussi  $\Pi_i^P = \text{co}\Sigma_i^P$  et  $\Delta_i^P := P^{\Sigma_{i-1}^P}$ .

▮ **Exemple 1.** On a

- ▷  $\Sigma_1^P = NP^P = NP$ ,
- ▷  $\Pi_1^P = \text{co}NP$ ,
- ▷  $\Delta_2^P = P^{NP}$ ,
- ▷  $\Sigma_2^P = NP^{NP}$ ,
- ▷ *etc.*

En général, on a les inclusions suivantes :



**Remarque 1.** On a que  $\Delta_i^P = P^{\Sigma_{i-1}^P} = P^{\Pi_{i-1}^P}$ . Par exemple, on a que  $\Delta_2^P = P^{NP} = P^{\text{coNP}}$ .

Les classes  $\Delta_i^P$  sont closes par complément. Par exemple, l'inclusion  $\Delta_i^P \subseteq \Pi_i^P$  découle de la clôture par complément et de l'inclusion  $\Delta_i^P \subseteq \Sigma_i^P$ .

On omet parfois l'exposant « P », mais attention, il existe une hiérarchie similaire (avec  $\Sigma_i$ ,  $\Pi_i$  et  $\Delta_i$ ) en calculabilité.

Il ne reste que l'inclusion  $PH \subseteq PSPACE$  à démontrer.

**Proposition 1.** On a  $PH \subseteq PSPACE$ .

**Preuve.** On montre par récurrence sur  $i$  que  $\Sigma_i^P \subseteq PSPACE$ .

- ▷ On a  $\Sigma_0^P = P \subseteq PSPACE$ .
- ▷ Supposons  $\Sigma_{i-1}^P \subseteq PSPACE$ . On a

$$\Sigma_i^P = NP^{\Sigma_{i-1}^P} \subseteq NP^{PSPACE} = PSPACE,$$

$$\text{car } NP^{\text{QBF}} = PSPACE.$$

□

▮ **Proposition 2.** Si  $P = NP$  alors  $PH = P$ .

Plus généralement, pour tout  $i \geq 0$ , si

$$\Sigma_i^P = \Sigma_{i+1}^P,$$

alors  $PH = \Sigma_i^P$ . On dit alors que « la hiérarchie polynomiale s'effondre au  $i$ -ème niveau ».

▮ **Preuve.** Supposons  $\Sigma_i^P = \Sigma_{i+1}^P$ .

On montre par récurrence que  $\Sigma_j^P = \Sigma_i^P$  pour tout  $j \geq i + 1$ . L'initialisation est vraie par hypothèse. L'étape de récurrence est : supposons  $\Sigma_{j-1}^P = \Sigma_i^P$  alors

$$\Sigma_j^P = NP^{\Sigma_{j-1}^P} = NP^{\Sigma_i^P} = \Sigma_{i+1}^P = \Sigma_i^P.$$

□

## 1 Caractérisation par quantificateurs.

▮ **Théorème 1.** Un langage  $A$  est dans  $\Sigma_i^P$  si, et seulement si, il existe  $B \in P$  et un polynôme  $p$  tel que, pour tout  $x \in \{0, 1\}^*$ ,

$$x \in A \iff \left( \begin{array}{l} \exists y_1 \in \{0, 1\}^{p(n)} \\ \forall y_2 \in \{0, 1\}^{p(n)} \\ \exists y_3 \in \{0, 1\}^{p(n)} \\ \vdots \\ Q_i y_i \in \{0, 1\}^{p(n)} \\ \langle x, y_1, y_2, \dots, y_i \rangle \in B \end{array} \right),$$

avec une alternance de  $\forall$  et de  $\exists$ .<sup>1</sup>

□

▮ **Remarque 2.**

1. On note ici  $Q_i := \exists$  si  $i$  est impair et  $Q_i := \forall$  si  $i$  est pair.

1. Cette caractérisation est similaire (c'est une généralisation) à la caractérisation de **NP** avec des certificats.
2. On peut quantifier sur des blocs de taille variables (des chaînes de tailles  $p_1(n), p_2(n), \dots, p_i(n)$ ). On peut aussi enchaîner plusieurs blocs existentiels sans augmenter le  $i$  (il suffit de concaténer les chaînes).
3. On pourrait aussi quantifier sur  $y_k \in \{0, 1\}^{\leq p(n)}$ .
4. On a une caractérisation similaire pour la classe  $\Pi_i^P$  où on commence par «  $\forall y_1 \in \{0, 1\}^{p(n)}$  ».

☞ **Proposition 3.** Si  $\Sigma_i^P = \Pi_i^P$  alors on a que  $\Sigma_i^P = \text{PH}$ .

☞ **Preuve.** Montrons  $\Sigma_i^P = \Sigma_{i+1}^P$ . Soit  $A \in \Sigma_{i+1}^P$ . On a

$$x \in A \iff \exists y_1 \forall y_2 \dots Q_{i+1} y_{i+1} \langle x, y_1, \dots, y_{i+1} \in B \rangle,$$

avec  $B \in \text{P}$ . Et, le langage

$$C := \{ \langle x, y_1 \rangle \mid \forall y_2 \dots Q_{i+1} y_{i+1} \langle x, y_1, \dots, y_{i+1} \in B \rangle \}$$

est dans  $\Pi_i^P$ , donc dans  $\Sigma_i^P$ . D'où, par caractérisation,

$$x \in A \iff \exists y_1 \underbrace{\exists z_1 \forall z_2 \dots Q_i z_i \langle x, y_1, z_1, \dots, z_i \rangle \in D}_{\langle x, y_1 \rangle \in C},$$

avec  $D \in \text{P}$ . Et ainsi  $A$  est un problème de  $\Sigma_i^P$  en combinant les deux «  $\exists$  » (avec la remarque précédente).  $\square$

☞ **Remarque 3 (Propriétés supplémentaires).**

1. La classe  $\Sigma_i^P$  est *close par réduction polynomiale*, c'est-à-dire si  $B \in \Sigma_i^P$  et  $A \leq_P B$  alors  $A \in \Sigma_i^P$ .
2. Le problème de décision **QBF**- $\Sigma_i^P$  est  $\Sigma_i^P$ -complet, où

**QBF- $\Sigma_i^P$**  | **Entrée.** Une formule booléenne quantifiée  $F$  avec  $i$  quantificateurs et commençant par un bloc existentiel  
**Sortie.** Est-ce que  $F$  est vraie ?

3. De même, le problème de décision **QBF- $\Pi_i^P$**  est  $\Pi_i^P$ -complet, où

**QBF- $\Pi_i^P$**  | **Entrée.** Une formule booléenne quantifiée  $F$  avec  $i$  quantificateurs et commençant par un bloc universel  
**Sortie.** Est-ce que  $F$  est vraie ?

## 2 Théorème de Karp-Lipton.

▮ **Théorème 2** (Karp-Lipton). Si  $\mathbf{NP} \subseteq \mathbf{P}/\text{poly}$ , alors  $\Sigma_2^P = \Pi_2^P$ .

▮ **Définition 3.** Un circuit booléen à  $s$  entrées *décide SAT* si, étant donnée une formule booléenne  $F$  de taille  $s$ , le circuit  $C$  décide si  $F$  est satisfiable.

La preuve de ce théorème repose sur deux lemmes.

▮ **Lemme 1.** L'ensemble des (codages de) circuits qui décident SAT est dans  $\text{coNP}$ .

▮ **Preuve.** On utilise le fait que SAT est *auto-réductible*<sup>2</sup> : une formule booléenne  $F(v_1, \dots, v_n)$  est satisfiable si et seulement si l'une des deux formules booléennes

$$F(v_1, \dots, v_{n-1}, \theta) \quad \text{ou} \quad F(v_1, \dots, v_{n-1}, 1)$$

est satisfiable.

Un circuit  $C$  décide SAT ssi pour toute formule  $F$  de taille  $s$

1. si  $F$  n'a pas de variable, alors  $C(F) = 1$  ssi  $F \equiv 1$  ;

2. si  $F$  dépend de  $n \geq 1$  variables  $v_1, \dots, v_n$  alors  $C(F) = 1$  ssi

$$C(F[v_n := 0]) = 1 \quad \text{ou} \quad C(F[v_n := 1]) = 1.$$

Étant donnée  $F$ , les conditions ci-dessous peuvent être vérifiées en temps polynomial (car VALCIRC est dans **P**).

Cette caractérisation commence par un « pour toute formule » et on considère ensuite un problème dans **P**, d'où le langage est bien dans **coNP**.  $\square$

☞ **Lemme 2.** Si **NP**  $\subseteq$  **P/poly**, alors SAT peut être décidé par une famille de circuits booléens de taille polynomiale.  $\square$

☞ **Preuve (du théorème de Karp-Lipton).** On suppose avoir l'inclusion des classes **NP**  $\subseteq$  **P/poly**. Il suffit de montrer que  $\Pi_2^P \subseteq \Sigma_2^P$ . En effet, avec ça on a que

$$\Sigma_2^P = \text{co}\Pi_2^P \subseteq \text{co}\Sigma_2^P = \Pi_2^P.$$

Il suffit de montrer que le problème de décision **QBF- $\Pi_2^P$**  est dans  $\Sigma_2^P$ . Soit  $F$  une formule booléenne de taille  $s$ , alors

$$\forall u \exists v \quad F(u, v),$$

est équivalente à

$$\exists C \forall u \quad C(F(u, \cdot)) = 1 \quad \text{et} \quad C \text{ décide SAT},$$

où  $C$  est un circuit booléen avec  $s$  entrées. Il suffit de quantifier sur des circuits de taille polynomiale d'après le lemme 2. Ceci

---

2. *self-reducible* en anglais.

est équivalent à

$$\exists C \forall u \quad C(F(u, \cdot)) = 1 \quad \text{et} \quad \forall y \in \{\emptyset, 1\}^{p(s)} \langle C, y \rangle \in A,$$

avec  $A \in \mathbf{P}$  d'après le lemme 1 et la caractérisation de  $\mathbf{coNP}$ . On en déduit que ceci est équivalent à

$$\exists C \forall u \forall y \quad \underbrace{C(F(u, \cdot)) = 1 \quad \text{et} \quad \langle C, y \rangle \in A}_{\text{vérifiable en temps polynomial}},$$

et est donc vérifiable dans  $\Sigma_2^{\mathbf{P}}$  grâce à la caractérisation par quantificateurs.  $\square$