

M1if09 – Calculabilité & complexité

*Sylvain Brandel*

2021 – 2022

[sylvain.brandel@univ-lyon1.fr](mailto:sylvain.brandel@univ-lyon1.fr)

# COMPLEXITÉ

# NP-COMPLÉTUDE

# NP-complétude

- « Etalons » ou références
  - Calculabilité : problème de l'arrêt
  - Complexité ?
    - problèmes NP-complets (les plus compliqués de la classe NP)
- Problèmes NP-complets
  - Les problèmes NP-complets  $\in$  NP.
  - Les problèmes de la classe P  $\in$  NP.
  - Il existe des problèmes NP qui n'ont pas été montrés  $\in$  P et qui n'ont pas été montrés NP-complets
- Théorème

*S'il existe un problème NP-complet décidé par un algorithme polynomial, alors tous les problèmes de NP sont décidables en temps polynomial*

(c-à-d **P = NP**)

# NP-complétude

- Réduction
  - Décidabilité : fonction récursive
  - Complexité : fonction polynomiale
- Une fonction  $\Sigma^* \rightarrow \Sigma^*$  est dite **calculable en temps polynomial** ssi il existe une MT (déterministe) polynomialement bornée qui la calcule.
- Soient  $L_1$  et  $L_2 \subseteq \Sigma^*$ .  
Une **réduction polynomiale** (ou **transformation polynomiale**) de  $L_1$  à  $L_2$  est une fonction  $\tau : \Sigma^* \rightarrow \Sigma^*$ , calculable en **temps polynomial**, telle que  
 $x \in L_1$  ssi  $\tau(x) \in L_2$

# NP-complétude

- Un langage  $L$  est NP-complet si
  - $L \in \text{NP}$ ,
  - Pour tout langage  $L' \in \text{NP}$ , il existe une réduction polynomiale de  $L'$  dans  $L$ .
- Le premier point est facile à établir
  - Algorithme non déterministe polynomial.
- Le second point est plus délicat
  - Mais si on connaît un langage  $L''$  **NP-complet**, il suffit de démontrer qu'il existe une réduction polynomiale de  $L''$  dans  $L$ .

# NP-complétude

- Exemples de problèmes
  - Planification de 2 machines (*2-machine scheduling*) :  
n tâches de durées respectives  $a_1, a_2, \dots, a_n$  à répartir sur 2 machines de sorte qu'un deadline  $D$  soit respecté.
  - Problème du sac-à-dos (*Knapsack*) :  
Soit un ensemble d'objets avec des poids  $S = \{a_1, a_2, \dots, a_n\}$ , et un entier  $K$ , le tout donné en binaire.  
Trouver un sous-ensemble  $P \subseteq S$  tel que  $\sum_{a_i \in P} a_i = K$ .
  - Problème de la *partition* :  
Soit un ensemble de  $n$  entiers positifs  $S = \{a_1, a_2, \dots, a_n\}$  représentés en binaire.  
Existe-t-il  $P \subseteq \{1, 2, \dots, n\}$  tel que  $\sum_{a_i \in P} a_i = \sum_{a_i \notin P} a_i$  ?
- Ces trois problèmes sont équivalents du point de vue de la complexité polynomiale :
  - On peut réduire polynomialement chacun d'eux dans les autres. <sup>5</sup>

# NP-complétude

- **HAM** : Problème du cycle hamiltonien  
soit  $G$  un graphe.  
Existe-t-il dans  $G$  un cycle passant par chaque **sommet** une fois et une seule ?
- Réduction HAM vers SAT
  - $n^2$  variables  $x_{ij}$  :  $x_{ij}$  vraie si le sommet  $i$  de  $G$  est à la  $j^{\text{ème}}$  position d'un cycle ham.
  - FNC  $F$  exprimant :
    - 1) À la  $j^{\text{ème}}$  position doit apparaître exactement un sommet
      - a) À la  $j^{\text{ème}}$  position doit apparaître au moins un sommet
      - b) À la  $j^{\text{ème}}$  position doit apparaître au plus un sommet
    - 2) Chaque sommet apparaît dans le cycle exactement une fois
      - a) Chaque sommet apparaît au moins une fois
      - b) Chaque sommet apparaît au plus une fois
    - 3) On ne peut pas passer dans  $G$  du sommet  $j$  au sommet  $j+1$  s'il n'y a pas d'arête
  - $G$  hamiltonien ssi  $F$  est satisfiable
- Si HAM est NP-complet, alors SAT est NP-complet

# NP-complétude

- **HAM** : Problème du cycle hamiltonien  
soit  $G$  un graphe.  
Existe-t-il dans  $G$  un cycle passant par chaque **sommet** une fois et une seule ?
- Réduction 3-SAT vers HAM
  - Construire un graphe  $G$  à partir d'une FNC  $F$
  - $G$  hamiltonien ssi  $F$  est satisfiable
- Si 3-SAT est NP-complet, alors HAM est NP-complet

# NP-complétude

- Lemme

*Si  $\tau_1$  est une réduction polynomiale de  $L_1$  vers  $L_2$ ,  
et  $\tau_2$  est une réduction polynomiale de  $L_2$  vers  $L_3$ ,  
alors  $\tau_1 \circ \tau_2$  est une réduction polynomiale de  $L_1$  vers  $L_3$ .*



# Théorème de Cook

- SAT

Soit  $F$  une formule propositionnelle en forme normale conjonctive.

Existe-t-il une interprétation qui rend  $F$  vraie ?

(Satisfiabilité d'une FNC)

- Théorème

*Le problème SAT est NP-complet*

- Preuve (Stephen A. Cook, Canada, 1939 - )
  - Premier problème NP-complet prouvé (1971)

# Théorème de Cook

## Preuve

- SAT NP-complet :
  - SAT  $\in$  NP
  - Tout problème NP peut être réduit à SAT
- Soit  $L \in$  NP.  $L$  décidé par  $M$  non dét. polynomialement bornée par  $p$
- $\tau : M, w \rightarrow$  une instance de SAT positive ssi  $M$  accepte  $w$
- $M$  accepte  $w$  ssi il existe une **exécution** de  $M$  sur  $w$  d'au plus  $p(n)$  étapes ( $n = |w|$ )
- **Exécution** : suite d'au plus  $p(n)+1$  configurations successives de  $M$ 
  - Tableau  $R [p(n)+1] [p(n)+1]$  de symboles du ruban de  $M$   
 $R(i,j) =$  symbole dans la  $j^{\text{ème}}$  case du ruban de  $M$  à la  $i^{\text{ème}}$  configuration (étape)
  - Vecteur  $P [p(n)+1]$  d'entiers  $1 \dots p(n)+1$   
 $P(i) =$  position de la tête de lecture de  $M$  à la  $i^{\text{ème}}$  étape
  - Vecteur  $Q [p(n)+1]$  d'états de  $M$   
 $Q(i) =$  état courant de  $M$  à la  $i^{\text{ème}}$  étape
  - Vecteur  $C [r]$  d'entiers  $1 \dots r$  (degré du non déterminisme)  
 $C(i) =$  choix non dét. effectué par  $M$  à la  $i^{\text{ème}}$  étape

# Théorème de Cook

## Preuve

- FNC  $F$  exprimant :
  - 1)  $R(i,j)$  contient exactement une valeur
    - a)  $R(i,j)$  contient au moins une valeur
    - b)  $R(i,j)$  contient au plus une valeur
  - 2)  $P(i)$  contient une et une seule valeur
  - 3)  $Q(i)$  contient une et une seule valeur
  - 4)  $C(i)$  contient une et une seule valeur
  - 5) La première configuration est initiale
    - a)  $R(1,1) = B, R(1,2) = \sigma_1, R(1,3) = \sigma_2 \dots, R(1,n+1) = \sigma_n$
    - b)  $P(1) = 1$
    - c)  $Q(1) = s$
    - d)  $C(1) = \text{indifférent}$
  - 6) Chaque configuration est obtenue à partir de la précédente
    - a) Les cases ne se trouvant pas sous la tête ne sont pas modifiées
    - b) La case sous la tête est modifiée **et** déplacement de la tête
  - 7) A la fin de l'exécution on arrive dans un état acceptant
- $M$  accepte  $w$  ssi  $F$  est satisfiable

# Théorème de Cook

- SAT est NP-complet (on l'a prouvé)
  - On dispose à présent d'un problème NP-complet
  - On va pouvoir s'en servir pour démontrer que d'autres problèmes sont NP-complets par réduction polynomiale.
- **3-SAT** : satisfiabilité de FNC comportant exactement 3 littéraux par clause.
- **MAX-SAT** : Etant donné un ensemble de clauses et un entier  $K$ , existe-t-il une interprétation satisfaisant au moins  $K$  clauses ?
- Théorème  
*3-SAT est NP-complet.*
- Théorème  
*MAX-SAT est NP-complet.*

# Pour finir

- Considérations sur la NP-complétude

On a un problème, on cherche un algorithme pour le résoudre, mais ce problème est établi NP-complet.

Comme  $P \neq NP$  (hypothèse), ce problème n'a pas de solution polynomiale.

Faut-il pour autant renoncer à résoudre ce problème ? pas forcément.

La mesure de la complexité est pour le pire des cas :

- Pas d'algo polynomial  $\rightarrow$  pas d'algo pour **toutes** les instances du problème.
- Mais il peut très bien exister un algo polynomial pour certains cas, **voire presque tous**.

Il n'est obligé d'explorer tous les cas (nombre exponentiel de cas) :

- On peut utiliser des **heuristiques** pour limiter le nombre de cas à explorer.
- $\rightarrow$  critères approximatifs pour découvrir rapidement la solution recherchée (efficace des fois.)

Plutôt que de chercher **la** solution optimale, on peut chercher **une** solution s'en approchant.

On peut résoudre un ou plusieurs cas particuliers d'un problème NP-complet, en utilisant des algorithmes polynomiaux.