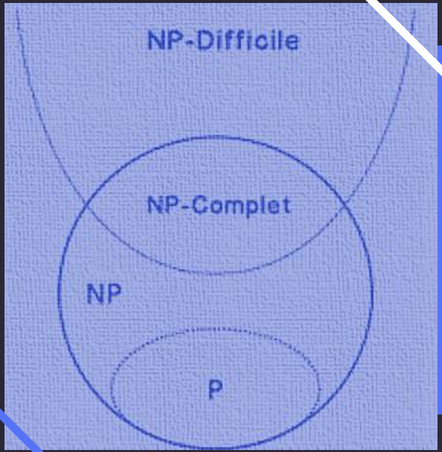
PRÉSENTATION

NP-complétude

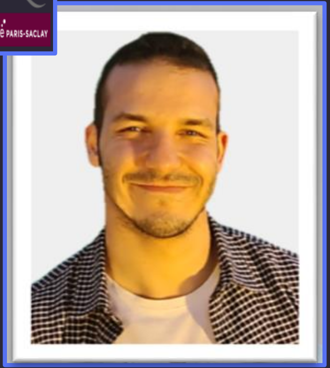


M2 AMIS | CALCULABILITÉ | UVSQ

PROFESSEUR : M. YANN STROZECKY

ÉQUIPE

Sarah Ouhocine



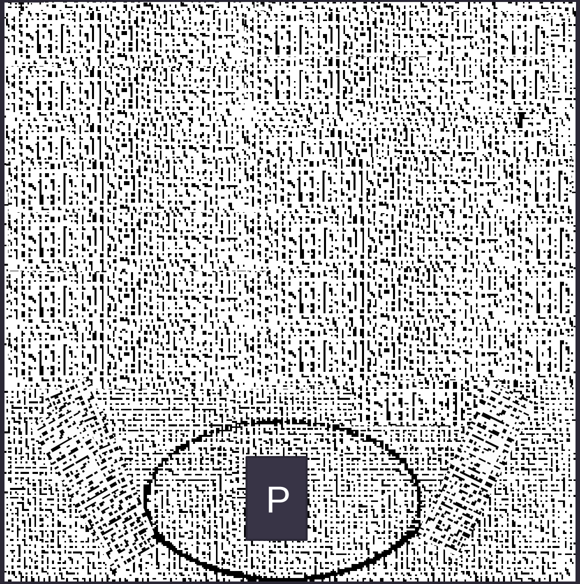
Djillali Boutouili

SOMMAIRE

1

INTRODUCTION

LES CLASSES DE PROBLÈMES

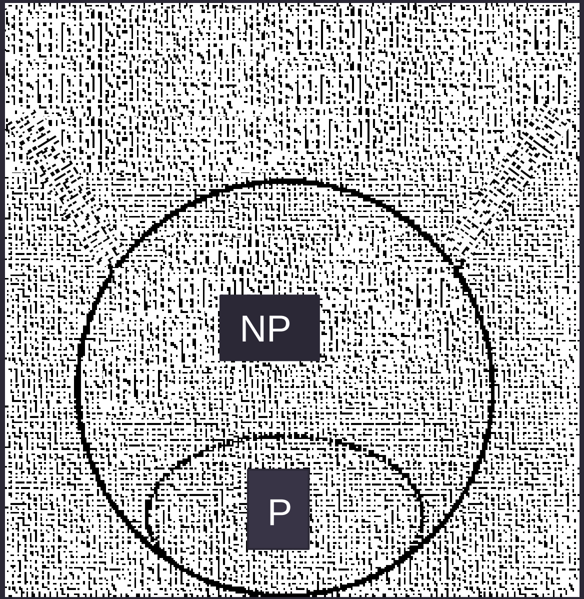


1. La classe des problèmes P

- Problèmes décidables en temps polynomial

[problèmes résolvables en temps polynomial par une machine de Turing déterministe]

LES CLASSES DE PROBLÈMES



1. La classe des problèmes P

- Problèmes décidables en temps polynomial

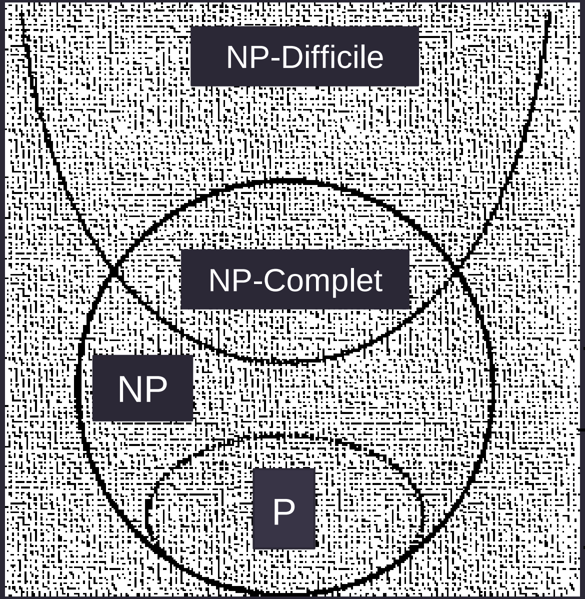
[problèmes résolvables en temps polynomial par une machine de Turing déterministe]

2. La classe des problèmes NP

- Problèmes non décidables en temps polynomial

[problèmes résolvables en temps polynomial par une machine de Turing non déterministe]

LA NP-COMPLÉTUDE



Concept en théorie de l'informatique qui concerne la complexité d'un problème

Les problèmes NP-Complets

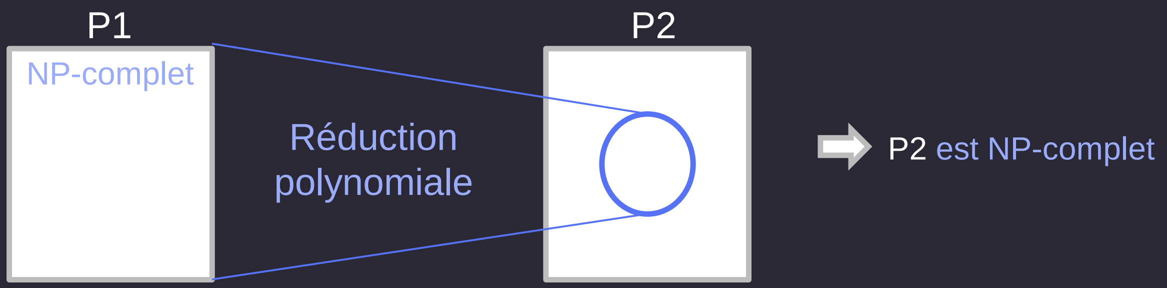
Un Problème est NP-complet s’il est :

1. **NP** (vérifiable en temps polynomial par une machine de Turing non déterministe)

2. **Réductible** à un autre problème NP-complet connu en temps polynomial.

PROUVER LA NP-COMPLÉTUDE

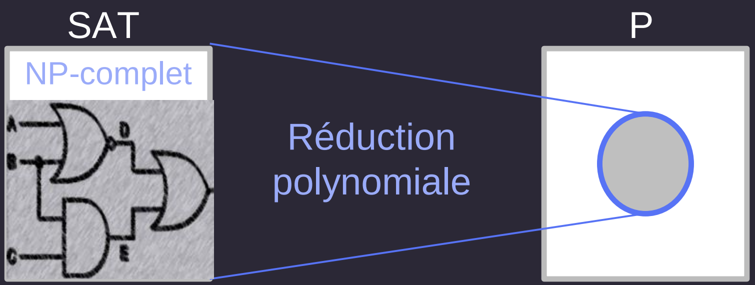
Tout problème NP-complet P1 qui se réduit à un autre problème P2, ce dernier est forcément un problème NP-complet



PROUVER LA NP-COMPLÉTUDE

Le problème NP-complet le plus connu et le plus couramment utilisé pour prouver la NP-Complétude c'est :

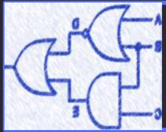
LE PROBLÈME DE SATISFIABILITÉ (SAT)



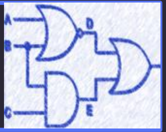
⇨ P est NP-complet

SAT

2

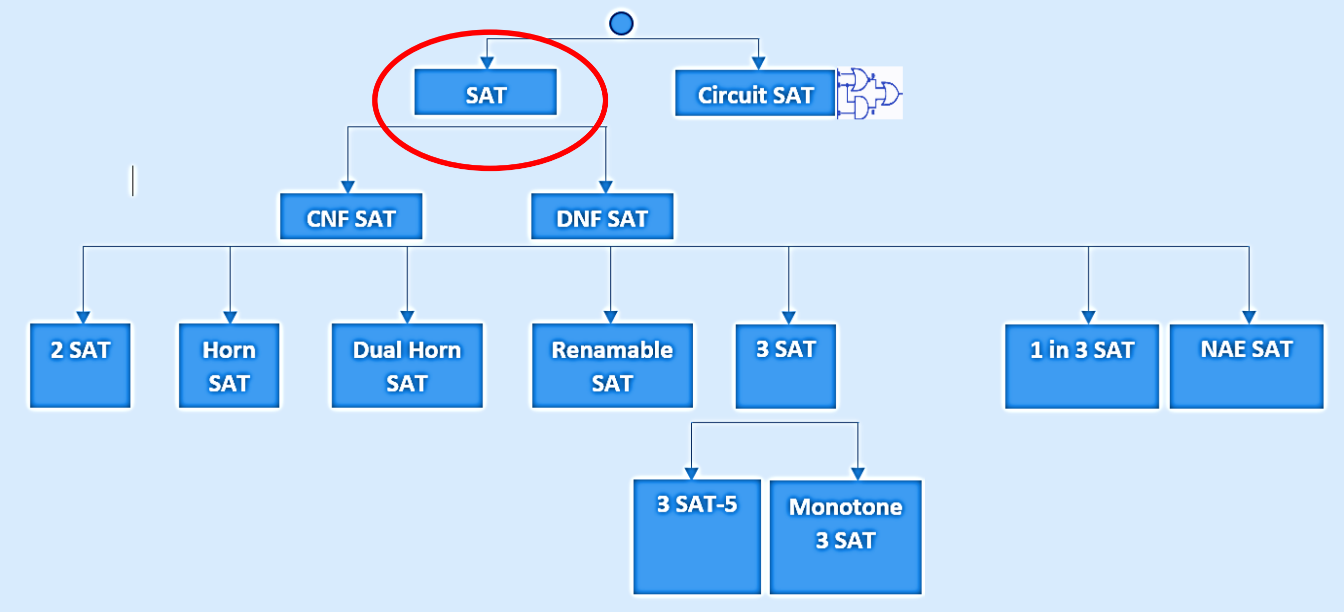


Variantes de SAT



SAT

[Variantes de SAT]



SAT

[algébrique]

SAT est un problème de décision en informatique qui consiste à déterminer s'il existe une affectation de valeurs (vrai ou faux) aux littéraux d'une formule booléenne de sorte que la formule soit évaluée "vrai".

Une formule F booléenne est composée de plusieurs clauses reliées par des opérateurs logiques « ET" (∧) ou « OU" (∨).

Chaque clause C est composée d'un littéral ou de plusieurs littéraux reliés par des opérateurs logiques "ET" (∧) ou "OU" (∨).

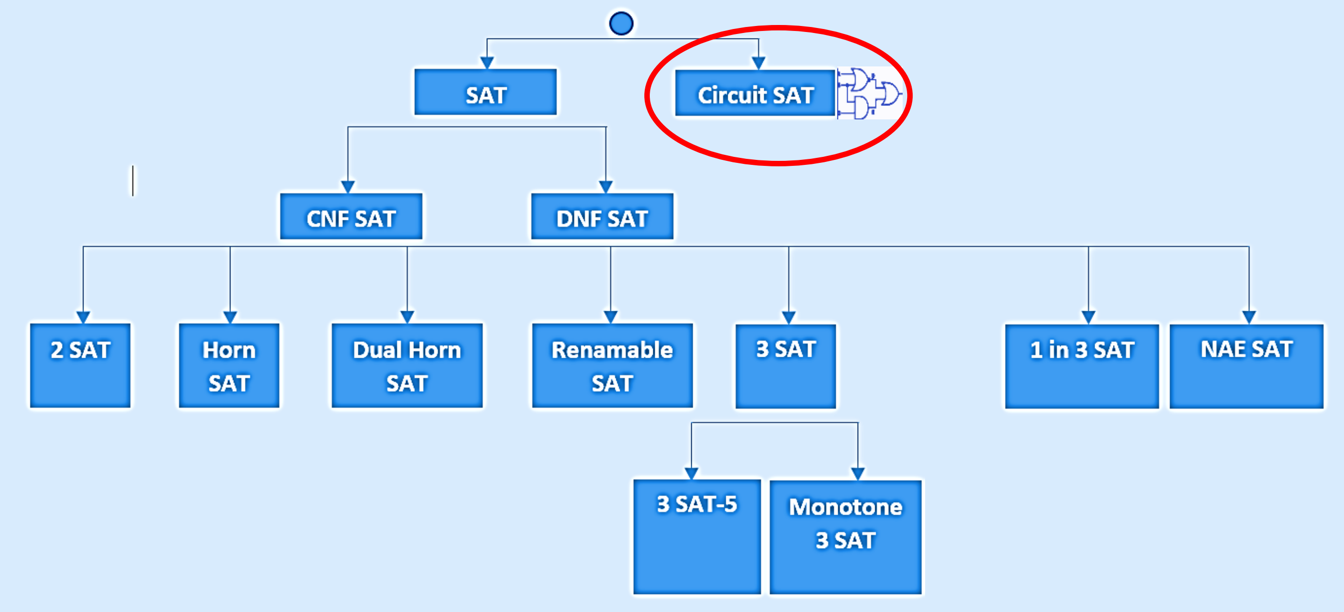
Un littéral l est une variable booléenne X ou sa négation ¬X (¬ est un opérateur logique qui inverse la valeur booléenne d'un littéral).

$$F = \underbrace{(l_{11} \wedge/\vee\ l_{21} \wedge/\vee \ \dots\ l_{n1})}_{C_1} \wedge/\vee \underbrace{(l_{12} \wedge/\vee\ l_{22} \wedge/\vee \ \dots\ l_{n2})}_{C_2} \wedge/\vee \ \dots\dots\ \underbrace{(l_{1n} \wedge/\vee\ l_{2n} \wedge/\vee \ \dots\ l_{nn})}_{C_n}$$

SAT est NP-complet

SAT

[Variantes de SAT]



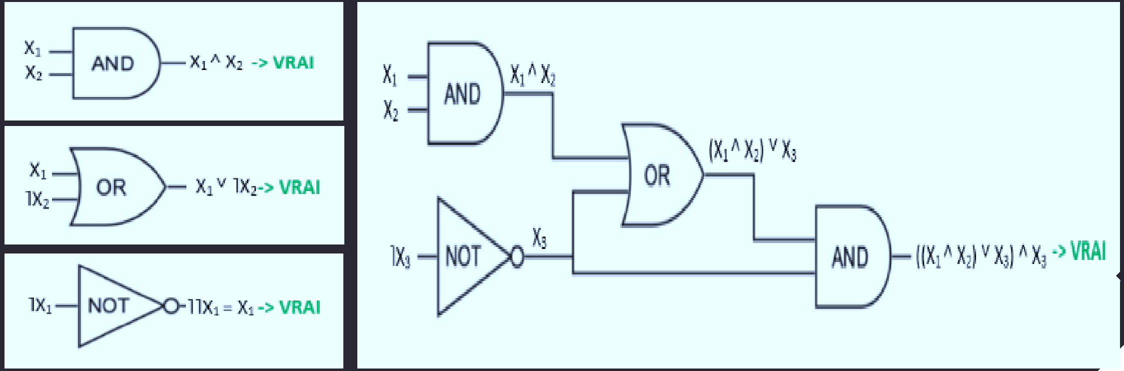
CIRCUIT SAT

[graphique]

CIRCUIT SAT est une variante du problème SAT, où la formule du SAT est remplacée par un circuit électrique acyclique.

Un circuit électrique est constitué de littéraux reliés par des portes logiques "ET", "OU" et "NEGATION" et les clauses sont similaires à celles du SAT.

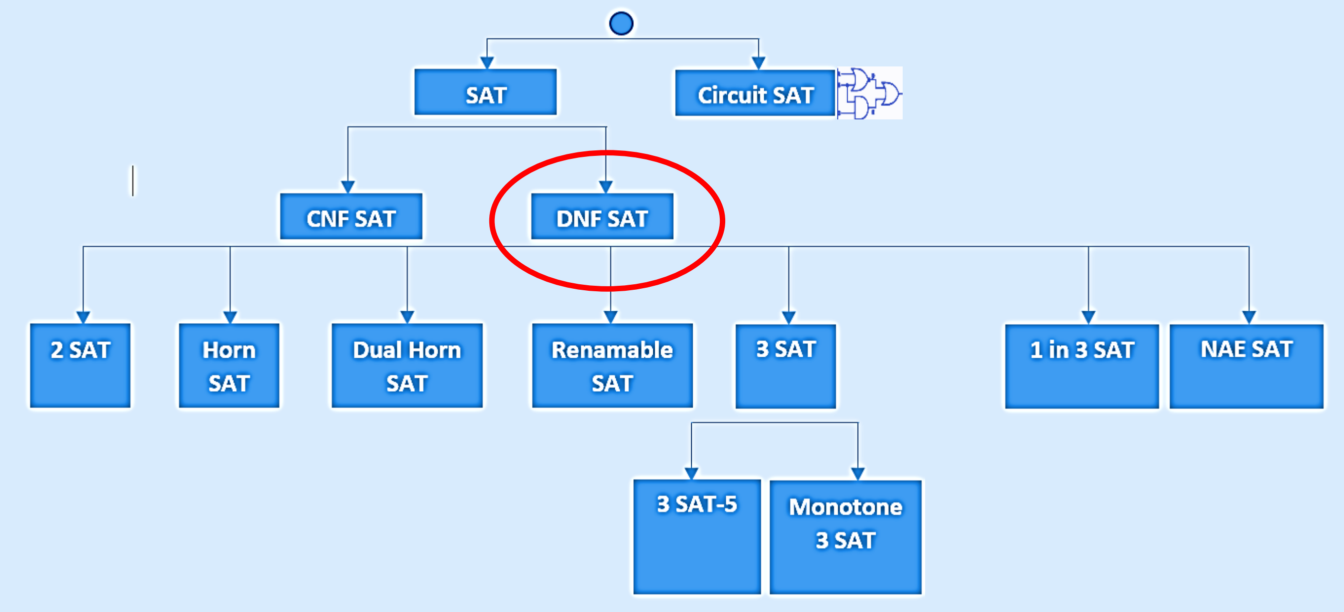
Le CIRCUIT SAT est un problème de décision qui consiste à déterminer si un circuit électrique donné peut être satisfaisant i.e. s'il existe une affectation de valeurs (vrai ou faux) aux littéraux de manière à ce que toutes les clauses du circuit soient vraies.



CIRCUIT SAT est NP-complet

SAT

[Variantes de SAT]



DNF SAT

DNF SAT (Forme Normale Disjonctive) est une variante du problème SAT, où la formule du SAT est écrite sous la forme d'une disjonction de conjonctions de littéraux.

$$F = \underbrace{(l_{11} \wedge l_{21} \wedge \ldots\, l_{n1})}_{C_1} \vee \underbrace{(l_{12} \wedge l_{22} \wedge \ldots\, l_{n2})}_{C_2} \vee \ldots\ldots\ldots \underbrace{(l_{1n} \wedge l_{2n} \wedge \ldots\, l_{nn})}_{C_n}$$

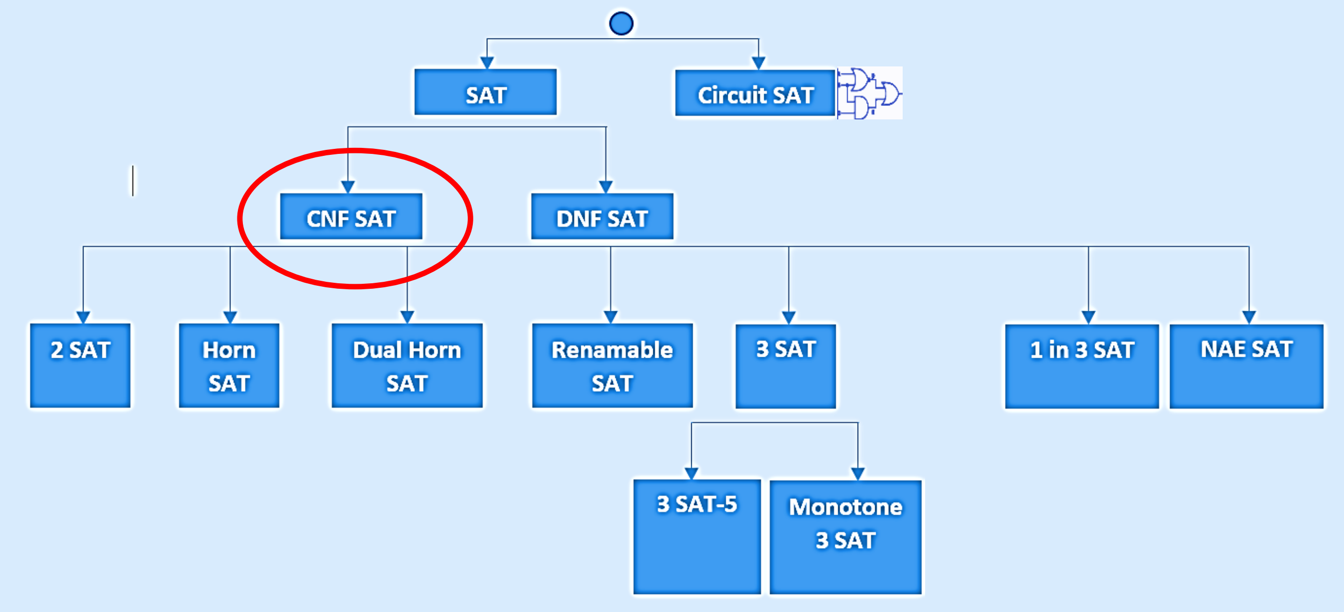
Exemple : la formule (X1 ∧ X2) ∨ (X3 ∧ X4) est satisfaisable si X1 et X2 sont vrais, ou si X3 et X4 sont vrais.

DNF SAT Є P (problème qui peut être résolu en temps polynomial par une machine de Turing) car pour que la formule soit satisfiable, il suffit de vérifier si une seule clause est vraie en prenant en compte les contradictions éventuelles entre les littéraux de ces clauses (par exemple entre X1 et ¬X1)

DNF SAT est Є P

SAT

[Variantes de SAT]



CNF SAT

CNF SAT (Forme Normale Conjonctive) est une variante du problème SAT, où la formule du SAT est écrite sous la forme d'une conjonctions de disjonction de littéraux.

$$F = \underbrace{(l_{11} \vee l_{21} \vee \dots\, l_{n1})}_{C_1} \wedge \underbrace{(l_{12} \vee l_{22} \vee \dots\, l_{n2})}_{C_2} \wedge \dots\dots\dots \underbrace{(l_{1n} \vee l_{2n} \vee \dots\, l_{nn})}_{C_n}$$

Exemple : la formule (X1 ∨ X2) ∧ (X3 ∨ X4) est satisfaisable si X1 ou X2 sont vrais, et X3 ou X4 sont vrais.

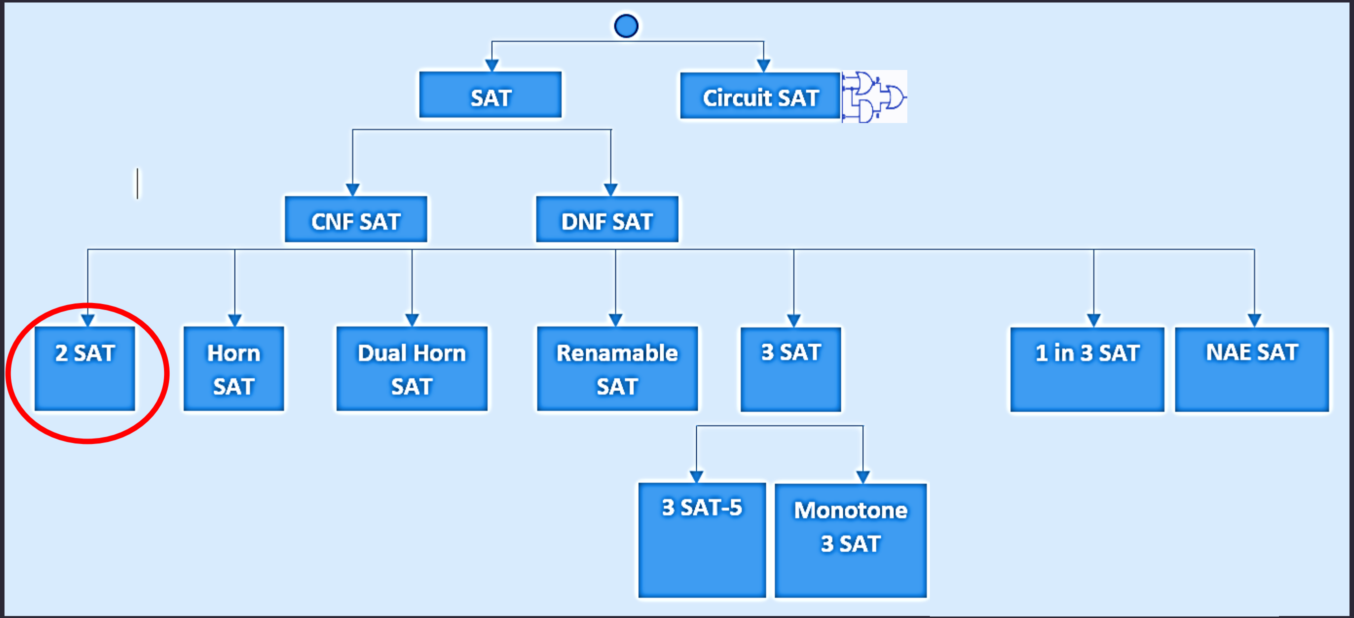
Pour qu'une formule CNF SAT soit satisfiable, il faut vérifier si toutes ses clauses sont vraies en prenant en compte les contradictions éventuelles entre les littéraux de ces clauses (par exemple entre X1 et ꓶX1)

CNF SAT est un problème NP-complet et peut être utilisée pour résoudre de nombreux problèmes de décision difficiles

CNF SAT est NP-Complet

SAT

[Variantes de SAT]



2 SAT

2 SAT est une variante du problème SAT, où la formule du SAT se présente sous la forme d'une CNF avec deux littéraux par clause

$$F = \underbrace{(l_{11} \vee l_{21})}_{C_1} \wedge \underbrace{(l_{12} \vee l_{22})}_{C_2} \wedge \ldots\ldots\ldots \underbrace{(l_{1n} \vee l_{2n})}_{C_n}$$

2 SAT Є P (problème qui peut être résolu en temps polynomial par une machine de Turing) car pour chaque clause on peut utiliser l'équivalence :

$a \rightarrow b \equiv \neg a \vee b$

Exemple : pour la clause (**X1** v **X2)**

$X1 \vee X2 \equiv \neg X1 \rightarrow X2$

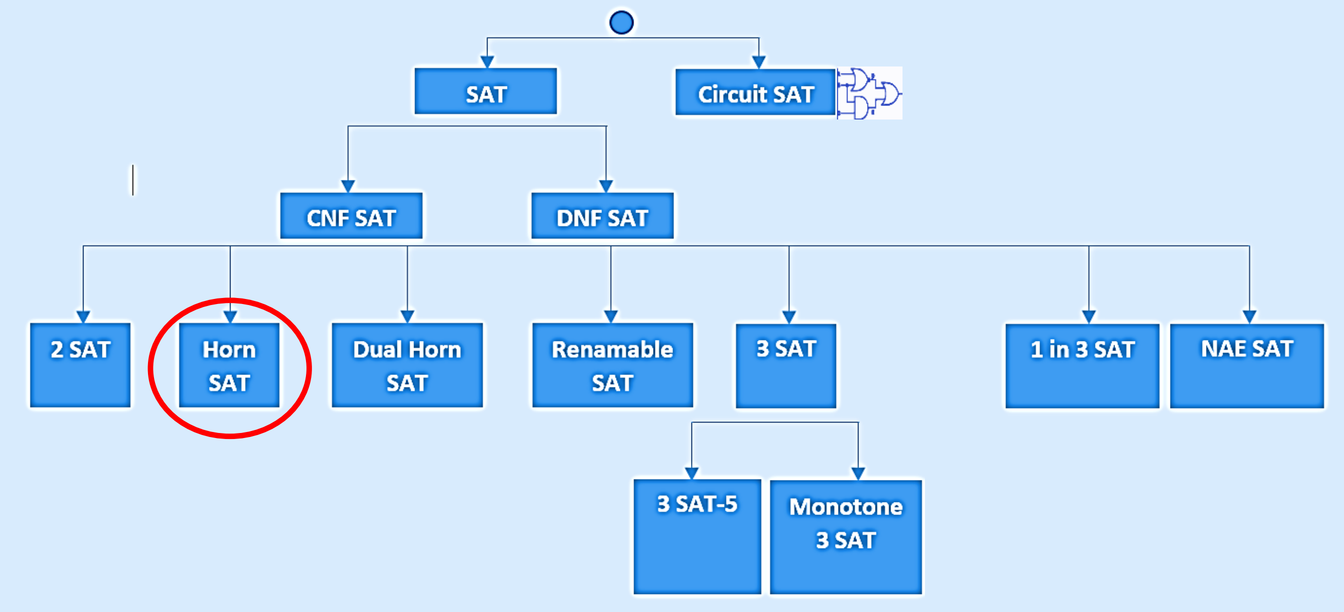
i.e.

Si X1 est faux, alors X2 doit être vrai car vrai → vrai

2 SAT est Є P

SAT

[Variantes de SAT]



HORN SAT

HORN SAT est une variante du problème SAT, où la formule du SAT se présente sous la forme d'une CNF avec au plus 1 littéral positif par clause

HORN SAT Є P (problème qui peut être résolu en temps polynomial par une machine de Turing) car pour chaque clause on peut utiliser l'équivalence :

$a \rightarrow b \equiv \neg a \vee b$

Exemple : pour la clause (**⌉A v ⌉B v ⌉C v D) tels que D est le littéral positif**

On revient à 2-SAT en considérant tous les littéraux négatifs comme étant un seul littéral (1er)

$\neg(A \vee B \vee C) \vee D \equiv (A \vee B \vee C) \rightarrow D$

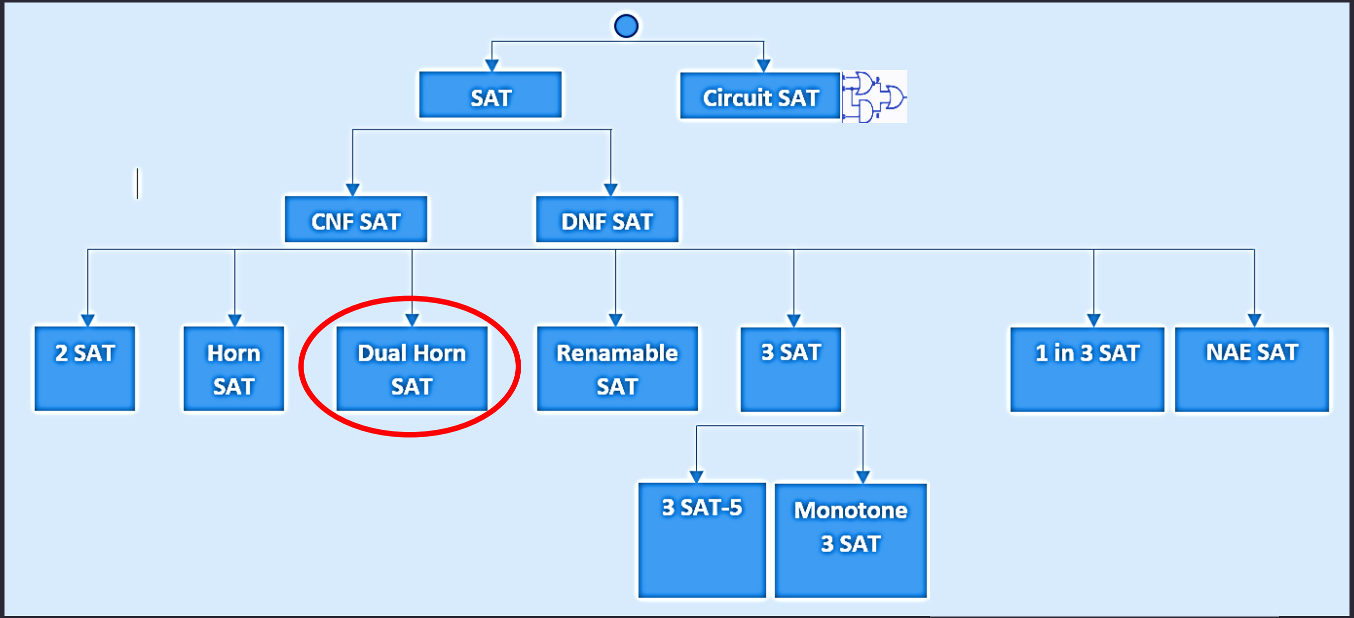
i.e.

Si A, B et C sont vrais, alors D doit être vrai car vrai → vrai

HORN SAT est Є P

SAT

[Variantes de SAT]



DUAL HORN SAT

DUAL HORN SAT est une variante du problème SAT, où la formule du SAT se présente sous la forme d'une CNF avec au plus 1 littéral négatif par clause

DUAL HORN SAT est la version symétrique de HORN SAT

DUAL HORN SAT Є P (problème qui peut être résolu en temps polynomial par une machine de Turing) car chaque formule DUAL HORN SAT peut être transformée en une formule HORN SAT en inversant tous les littéraux des clauses, puis en ré-inversant à nouveau les littéraux du résultat final pour avoir la solution du problème original

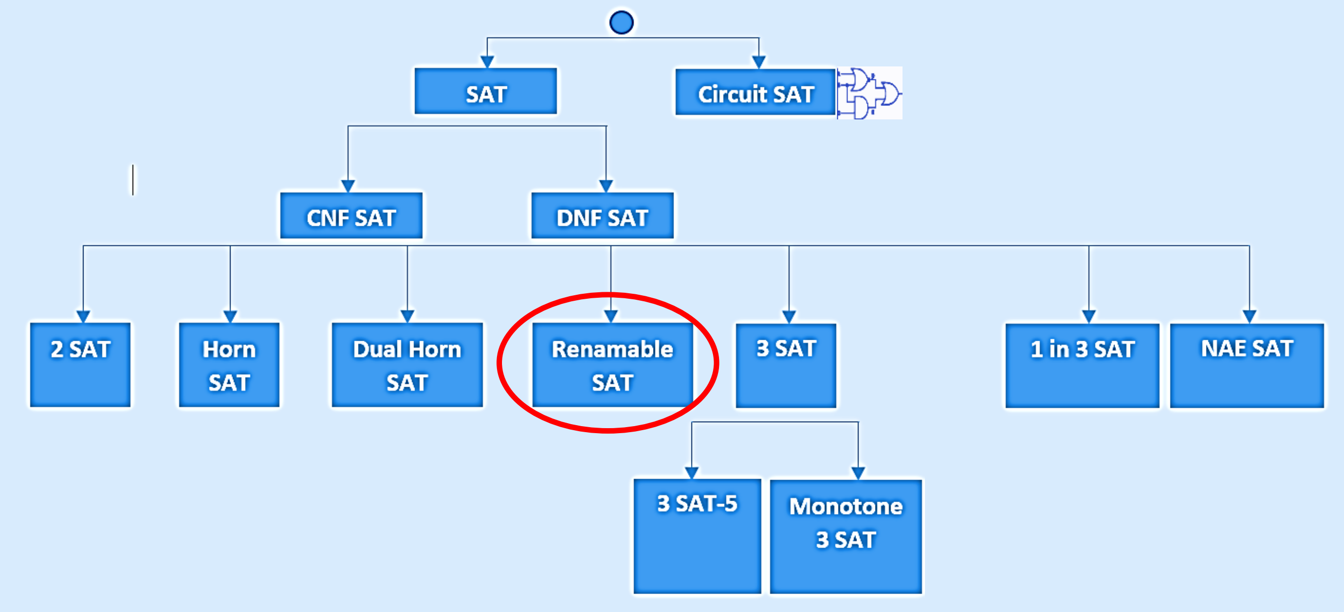
Exemple : clause (**A ∨ B ∨ C ∨ ¬D**) **, ¬D est le littéral négatif**

$$\overline{A} \lor \overline{B} \lor \overline{C} \lor \overline{\neg D} \xRightarrow{[\text{ HORN SAT }]} \neg A \lor \neg B \lor \neg C \lor D$$

DUAL HORN SAT est Є P

SAT

[Variantes de SAT]



RENAMABLE HORN SAT

RENAMABLE HORN SAT est une variante du problème SAT, où la formule du SAT se présente sous la forme d'une CNF où chaque clause a plus d’un littéral positif et où il est possible de renommer (inverser) les variables de telle sorte que toutes les clauses aient exactement un seul littéral positif (HORN SAT)

RENAMABLE HORN SAT Є P (problème qui peut être résolu en temps polynomial par une machine de Turing) car chaque formule RENAMABLE HORN SAT peut être transformée en une formule HORN SAT

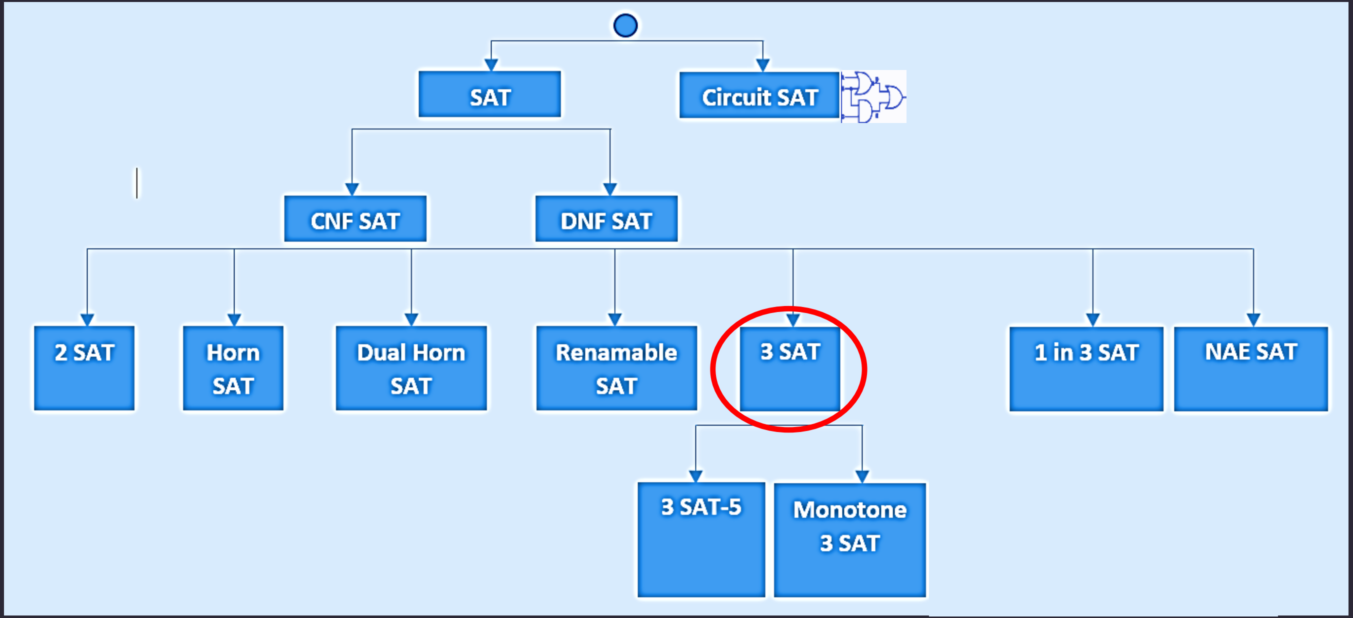
Exemple : clause (˥A v **B** v ˥C v **D**) **, X2 et X4 sont des littéraux positifs**

˥A v $\overline{\text{B}}$ v ˥C v D $\xrightarrow{\text{[HORN SAT]}}$ ˥A v ˥B v ˥C v D

RENAMABLE HORN SAT est Є P

SAT

[Variantes de SAT]



3 SAT

3 SAT est une variante du problème SAT, où la formule du SAT se présente sous la forme d'une CNF avec trois littéraux par clause

$$F = \underbrace{(l_{11} \vee l_{21} \vee l_{31})}_{C_1} \wedge \underbrace{(l_{12} \vee l_{22} \vee l_{32})}_{C_2} \wedge \ldots\ldots\ldots \underbrace{(l_{1n} \vee l_{2n} \vee l_{3n})}_{C_n}$$

Exemple 3 SAT : (A V B V C) ∧ (A V ¬B V D) ∧ (¬ A V ¬B V D) ∧ (¬A V ¬C V ¬D) …

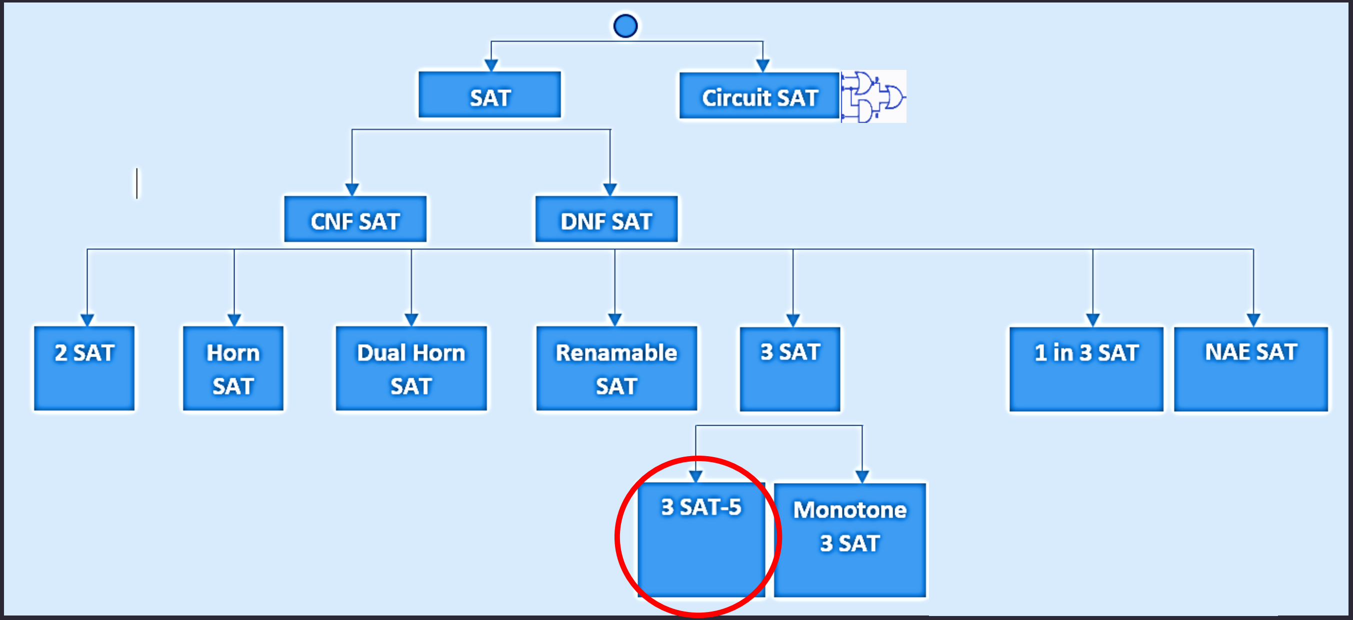
3 SAT consiste à trouver une affectation de valeurs aux variables qui rende la formule vraie

3 SAT est le problème NP-complet le plus connu et le plus couramment utilisé pour prouver la NP-Complétude

3-SAT est NP-complet

SAT

[Variantes de SAT]



3 SAT-5

3 SAT-5 est une variante du problème 3-SAT, où chaque variable apparait au plus dans 5 clauses.

$(A \vee B \vee C) \wedge (A \vee D \vee E) \wedge (A \vee F \vee G) \wedge (A \vee H \vee I) \wedge (A \vee J \vee K) \wedge (A \vee L \vee M) \ldots$

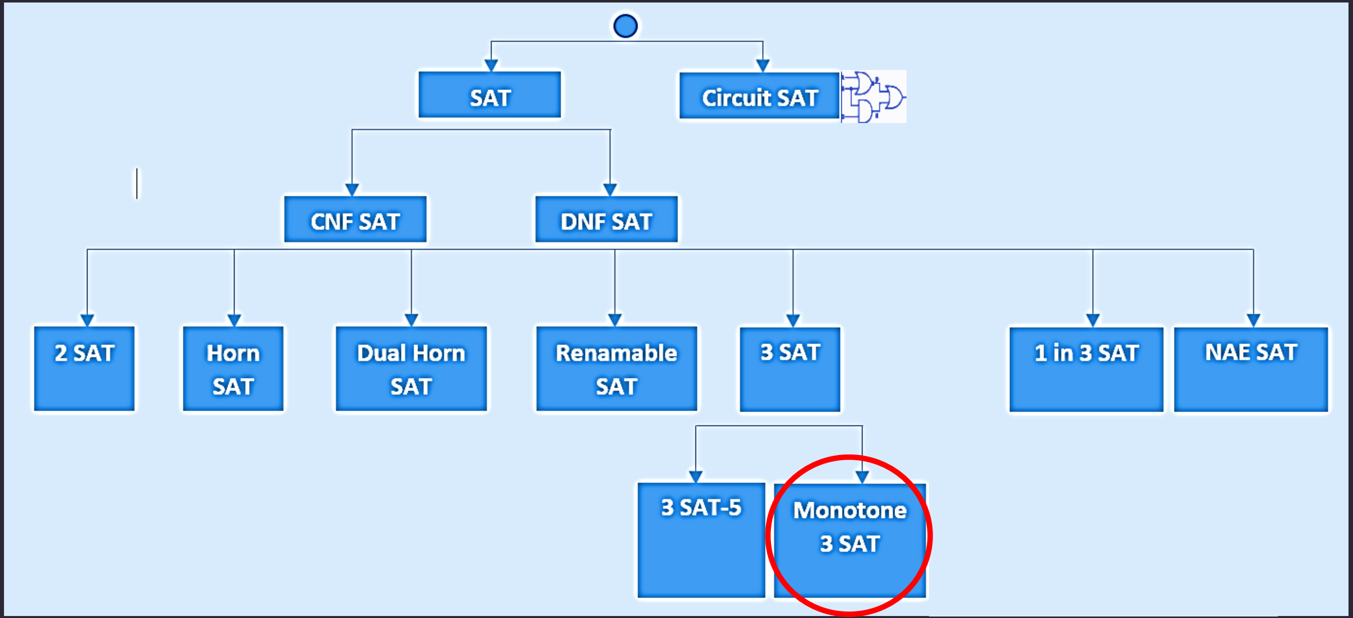
3 SAT-5 consiste à trouver une affectation de valeurs aux variables qui rende la formule vraie

3 SAT-5 est aussi un problème NP-complet

3 SAT-5 est NP-complet

SAT

[Variantes de SAT]



Monotone 3 SAT

Monotone 3 SAT est une variante du problème 3 SAT, où les littéraux de la formule sont soit tous positifs (vrais) ou soit tous négatifs (faux).

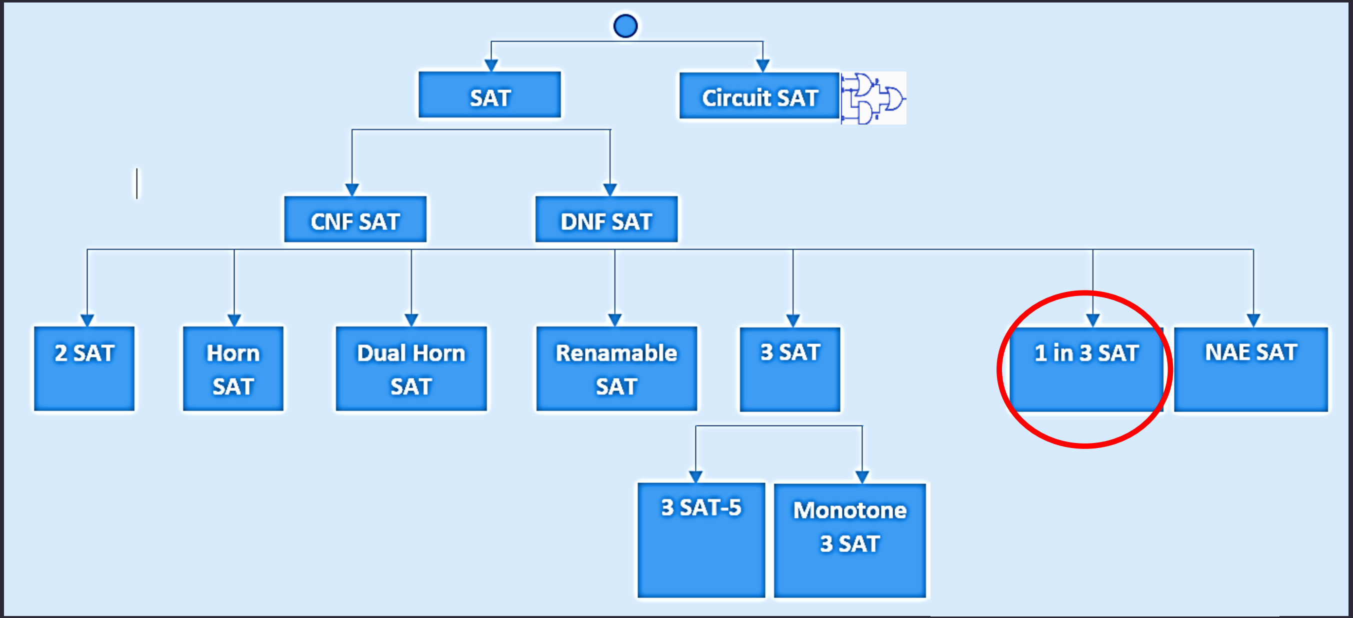
$$(A \lor B \lor C) \land (\neg A \lor \neg B \lor \neg C) \land \dots$$

Monotone 3 SAT consiste à trouver une affectation de valeurs aux variables qui rende la formule vraie

Monotone 3 SAT est aussi un problème NP-complet

SAT

[Variantes de SAT]



1 in 3 SAT

1 in 3 SAT est une variante du problème 3 SAT, où chaque clause a exactement 1 seul littéral positif parmi 3.

$$(A \vee \neg B \vee \neg C) \wedge (\neg D \vee E \vee \neg F) \wedge (\neg G \vee \neg H \vee I) \ldots$$

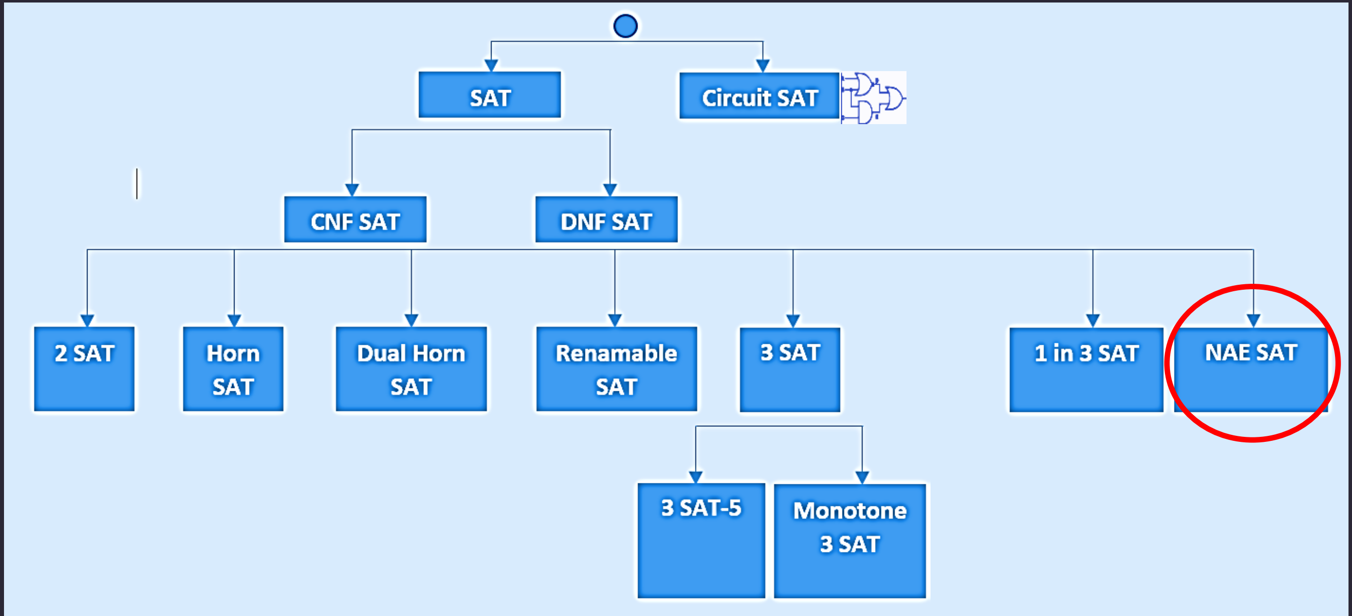
1 in 3 SAT consiste à trouver une affectation de valeurs aux variables qui rende la formule vraie

1 in 3 SAT est un problème NP-complet

1 in 3 SAT est NP-complet

SAT

[Variantes de SAT]



NAE 3 SAT

NAE 3 SAT est une variante du problème 3 SAT, où les littéraux d'une clause ne sont pas tous égaux

(A V B V ¬C) ∧ (A V ¬B V C) ∧ (A V B V ¬C) ∧ (A V B V C) ∧ (¬A V ¬B V ¬C) …

NAE 3 SAT consiste à trouver une affectation de valeurs aux variables qui rende la formule vraie

NAE 3 SAT est un problème NP-complet

NAE 3 SAT est NP-complet

SAT

3

Théorème de Dichotomie de Schaefer

Théorème de Dichotomie de Schaefer

[Théorème Universel]

Énoncer quelles variantes de SAT sont polynômiales et quelles variantes sont NP-complet.

Théorème a été utilisé pour prouver la NP-complétude de 3 SAT et ses deux variantes de 3 SAT les plus importantes :

1. 1 in 3 SAT
2. NAE SAT

Théorème de Dichotomie de Schaefer

[Énoncé du Théorème]

La variante de SAT est **Polynomiale** si :

1 – La mise de tous les littéraux à vrai ou à faux satisfait la formule

2 – Les clauses sont toutes HORN ou toutes DUAL HORN

3 – Les clauses sont toutes 2-SAT

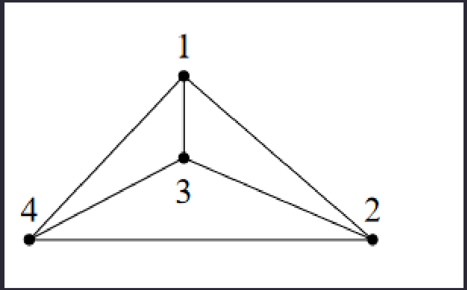
Sinon, la variante de SAT est **NP-complet**

4

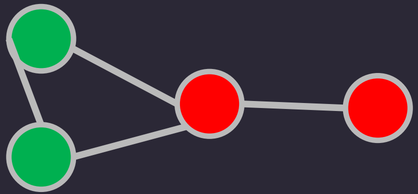
2-Colorable Perfect Matching

2-Colorable Perfect Matching

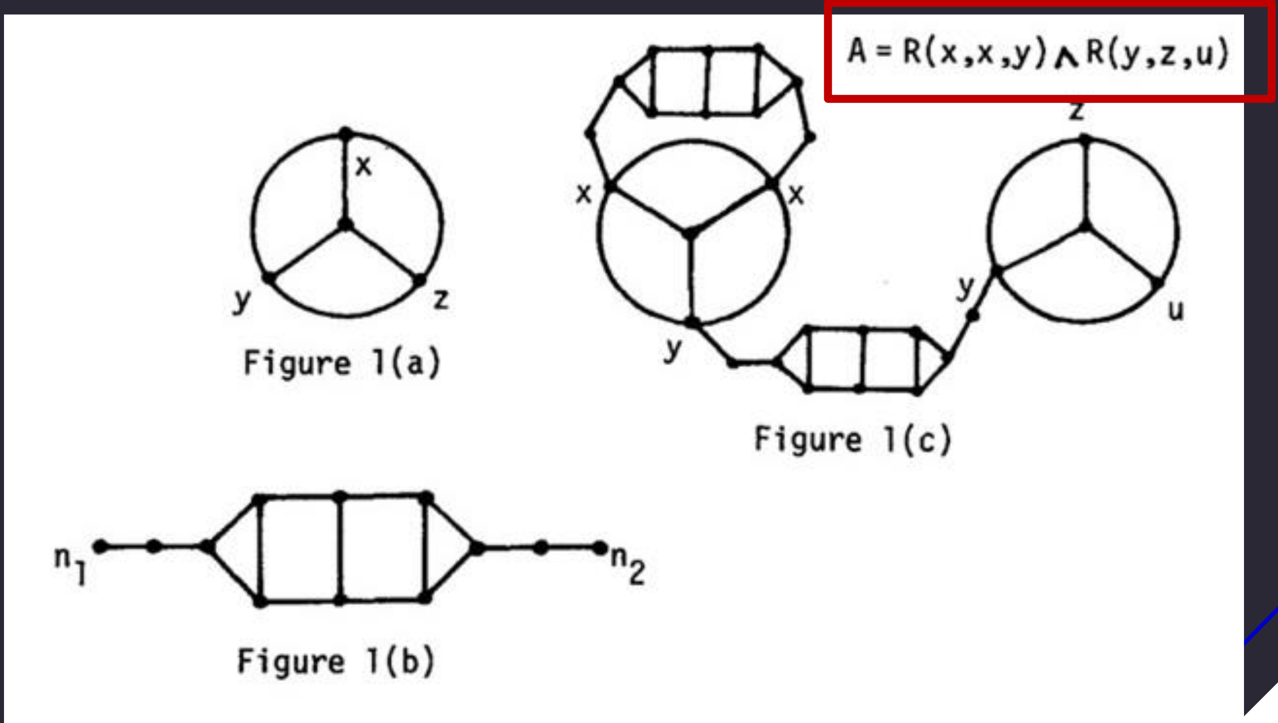
Entrées : G Un graphe planaire 3-régulier



Question : Peut on colorer les sommets de G avec 2 couleurs en faisant en sorte que chaque sommet ai exactement 1 seul voison ayant la meme couleur ?

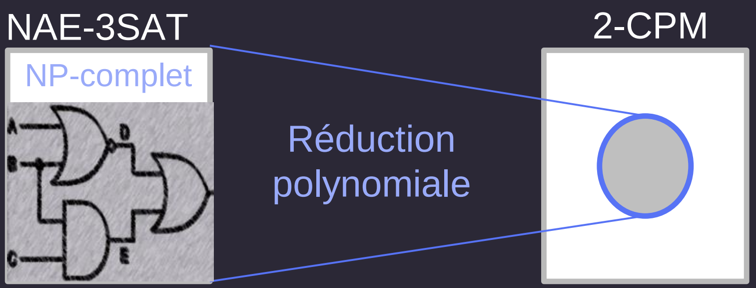


2-Colorable Perfect Matching



Réduction polynomiale

2-Colorable Perfect Matching



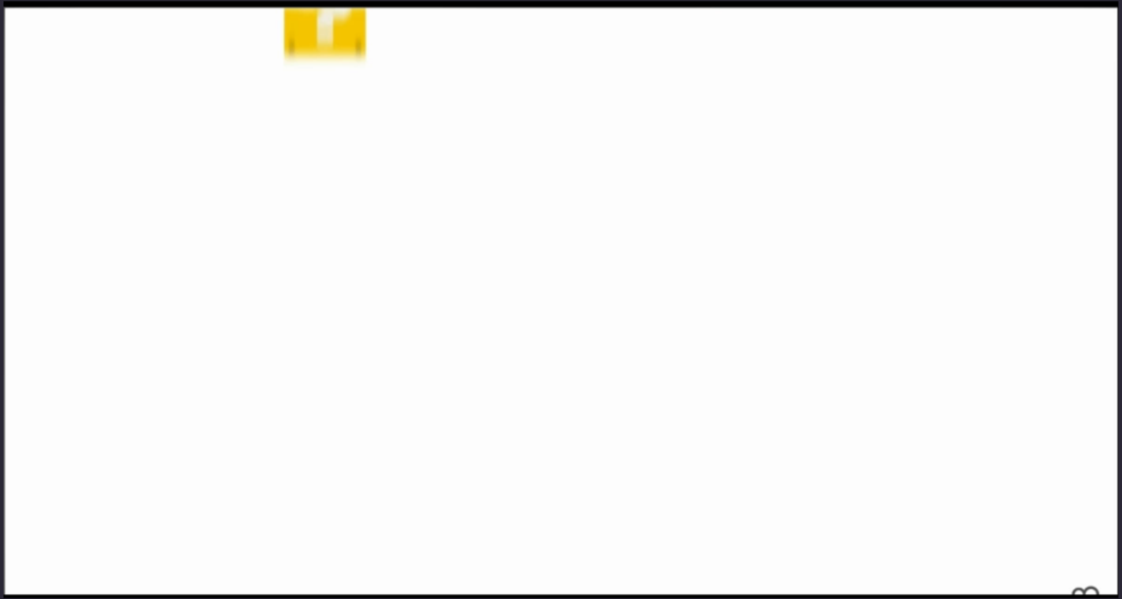
⇒ 2-CPM est NP-complet

5

JEU

JEU

Règles du Jeu

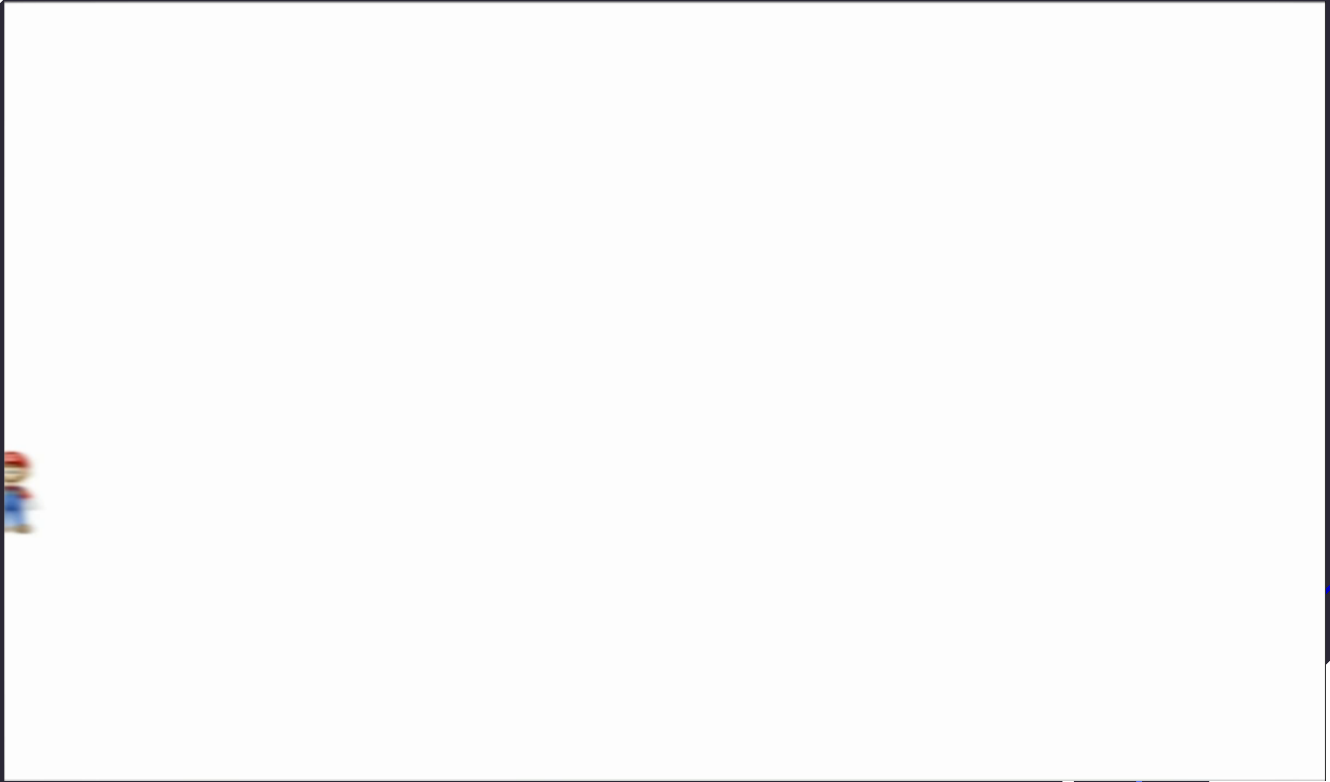


Réduction Polynomiale de 3SAT vers Mario

3-SAT

(x **OR** y) **AND** (¬x **OR** ¬y) **AND** (¬x **OR** y)

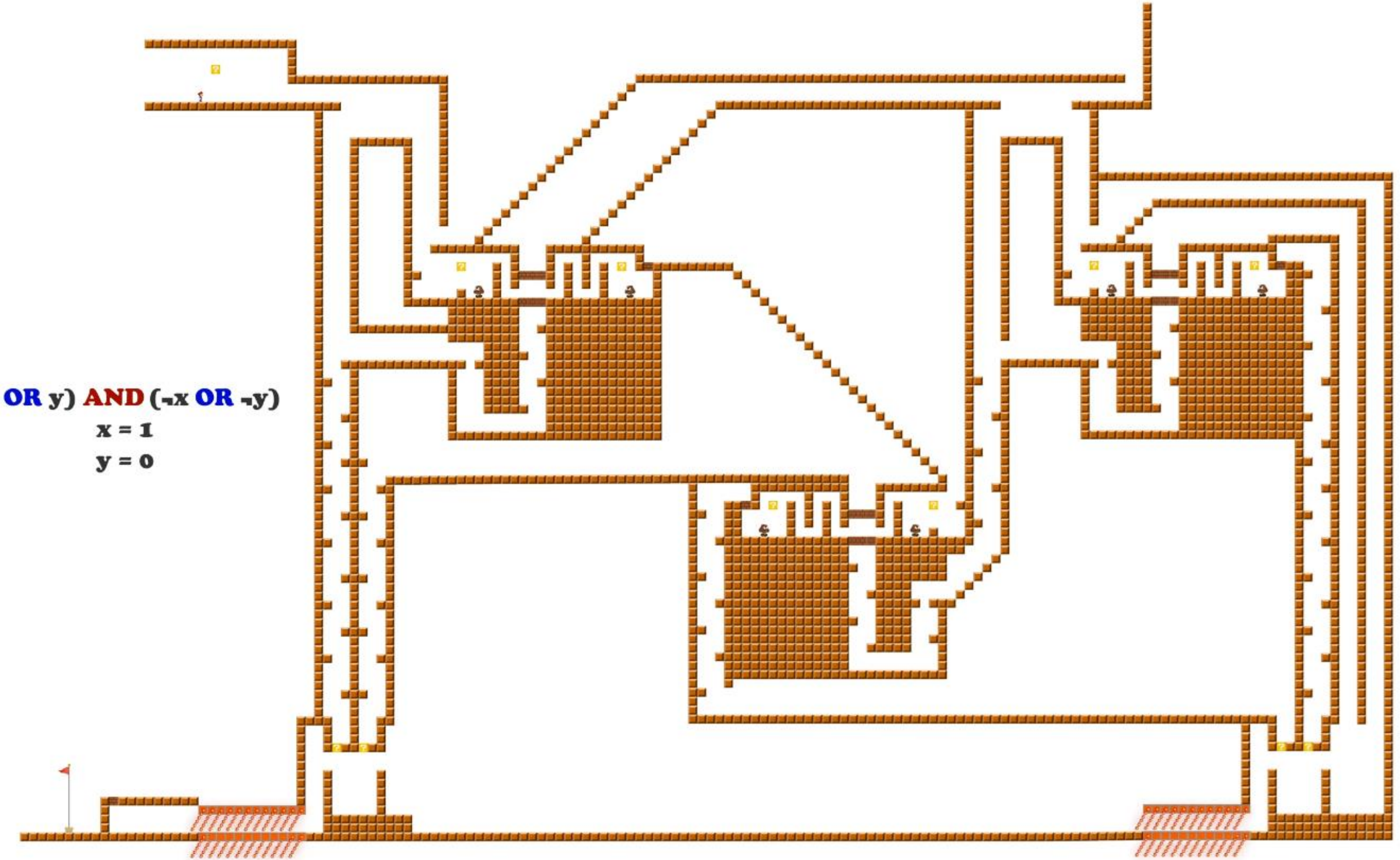
Fonctionnement



(x **OR** y) **AND** (¬x **OR** ¬y)

x = 1

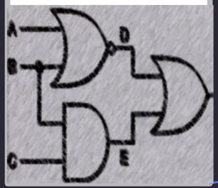
y = 0



RÉSULTAT

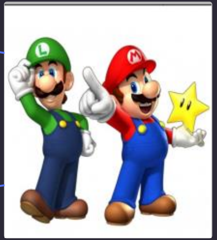
3SAT

NP-complet



Réduction polynomiale

Super Mario Brothers



Super Mario BROS est au moins NP-complet

6

CONCLUSION

RESSOURCES

LE SITE : HTTPS://COURSES.CSAIL.MIT.EDU/

- Site Web de la bibliothèque de cours en ligne de l'Institut de Technologie du Massachusetts (MIT), un état situé dans la région Nord-Est des États-Unis.

- Offre un accès en ligne à de nombreux cours de l'université, y compris des cours de sciences informatiques et de technologie de l'information.

- Les cours sont accessibles gratuitement et sont souvent accompagnés de ressources de cours, comme des feuilles de travail, des projets et des examens.

MERCI DE VOTRE ATTENTION