# RÉSOLUTION DE PROBLÈMES COMBINATOIRES

## SAT & DPLL

Loïc & Loïc

# Prise de notes collaboratives

minifi.ca/rpc1

# Menu du jour

- SAT
- DPLL
- SAT vs CP
- Sudoku party

# SAT

## Qu’est ce qu’un problème SAT ?

- Nom officiel : Boolean SATisfiability Problem
- Formulation :

> Soit une formule de logique propositionnelle, existe-t-il une assignation des variables propositionnelles qui rend la formule vraie ?

# Terminologie

- Formule de logique propositionnelles :
  - Construites à partir de variables propositionelles et des connecteurs booléens « et » ($\wedge$), « ou » ($\vee$), « non » ($\neg$).
- Clauses et forme normales conjonctives :
  - Un **littéral** $l$ est une variable propositionnelle $v_j$ (littéral positif) ou la négation d'une variable propositionnelle $\neg v_j$ (littéral négatif).
  - Une **clause** est une disjonction de la forme:

$$\bigvee_{i=1}^{n} l_i = (l_1 \vee l_2 \vee \ldots \vee l_n)$$

# Terminologie

Une formule du calcul propositionnel est en forme normale conjonctive (*Conjunctive Normal Form* (CNF) en anglais) si elle est une conjonction de clauses.

Exemple:

$$(v_1 \vee v_2) \wedge (\neg v_1 \vee v_3) \wedge (\neg v_2 \vee v_1)$$

Si la formule propositionnelle n'est pas sous forme normale conjonctive, on peut la transformer en une forme normale conjonctive équivalente de taille au plus exponentielle en la formule initiale.

# Davis–Putnam–Logemann–Loveland (DPLL)

# Historique

- Introduit en 1962 par Martin Davis, Hilary Putnam, George Logemann et Donald Loveland (Davis et al. 1962);
- Extension de l'algorithme de Davis-Putnam (1960) (Davis et Putnam 1960) ;
- Procédure efficace à la base des solveurs modernes.

# Résolution

DPLL étend l'algorithme de backtracking par l'utilisation des deux règles suivantes :

## La propagation unitaire

- Si une clause est unitaire, c'est-à-dire qu'elle contient un et un seul littéral, elle ne peut être satisfaite qu'en affectant l'unique valeur qui la rend vraie à son littéral. Il n'y a par conséquent plus à choisir.
- En pratique, son application entraîne une cascade d'autres clauses unitaires de manière déterministe, et évite donc d'explorer une grande partie de l'espace de recherche.
- Elle peut être vue comme une forme de propagation de contraintes.

# Résolution

## L'élimination des littéraux « purs »

- Si une variable propositionnelle apparaît seulement sous forme positive ou seulement sous forme négative alors ses littéraux sont dits purs.
- Les littéraux purs peuvent être affectés d'une manière qui rend toutes les clauses qui les contiennent vraiess.
- Par conséquent ces clauses ne contraignent plus l'espace de recherche et peuvent être éliminées.

# Algorithme

```
function DPLL(ϕ)
    while il existe une clause unitaire l ∈ ϕ do
        ϕ ← propagation-unitaire(l, ϕ)
    while il existe un littéral pur l ∈ ϕ do
        ϕ ← affecter-littéral-pur(l, ϕ)
    if ϕ est vide then
        return true
    if ϕ contient une clause vide then
        return faux
    l ← choisir-littéral(ϕ)
    return  DPLL(ϕ ∧ l) ∨ DPLL(ϕ ∧ ¬l)
```

# Algorithme

## Améliorations

- Backtracking non-chronologique
- Apprentissage (prochain cours)

# Sudoku Party

## TODO

1. Modéliser la résolution d'un Sudoku avec un modèle SAT.
2. Résoudre le modèle avec un solver SAT.
3. Comparer avec la version CP.

### Formats possibles :

- DIMACS CNF

# Bibliographie

Davis M, Logemann G, Loveland D (1962) A machine program for theorem-proving. Communications of the ACM 5:394-397. https://doi.org/10.1145/368273.368557

Davis M, Putnam H (1960) A Computing Procedure for Quantification Theory. Journal of the ACM 7:201-215. https://doi.org/10.1145/321033.321034