Chapitre 9 : Logique propositionnelle

(le problème SAT)

slides disponibles sur `http://cahier-de-prepa/mp2i-fermat`

Plan

Plan

Plan

Plan

Plan

Le problème

Dans cette section on s'intéresse à la satisfiabilité des formules (sur un ensemble de variables fini, *i.e.* $\mathrm{Card}(\mathcal{Q}) \in \mathbb{N}$.)

Pourquoi ? Une formule propositionnelle peut modéliser un problème concret dont une solution serait donnée par un environnement satisfaisant la formule.

exemple du sudoku : la valeur de vérité d'une variable $p_{i,j,k}$ indique si la case i, j contient la valeur k, l'environnement complet décrit une solution, et s'il satisfait la formule, alors c'est une solution valide.

Mais on s'intéresse déjà au problème de décision (plus simple) de savoir si une formule est satisfiable, sans demander par quel environnement.

SAT ‖ entrée : $A \in \mathbb{F}_p(\mathcal{Q})$
sortie : oui si A est satisfiable, non sinon

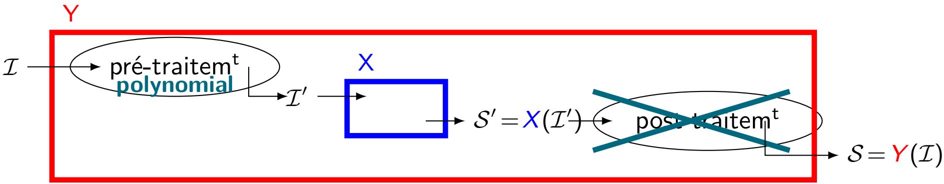
Remarque : puisque la satisfiabilité d'une formule ne dépend que de sa classe, on peut résoudre le problème sur une formule ayant une forme particulière, mais attention au coût de transformation.

Variantes du problème SAT

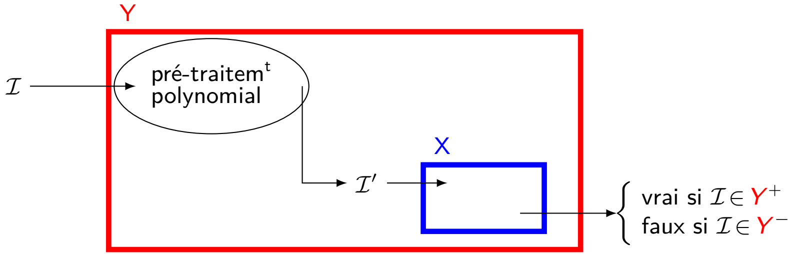
SAT	entrée : $A \in \mathbb{F}_p(\mathcal{Q})$ sortie : oui si A est satisfiable, non sinon
FND-SAT	entrée : $A \in \mathbb{F}_p(\mathcal{Q})$ sous FND sortie : oui si A est satisfiable, non sinon
FNC-SAT	entrée : $A \in \mathbb{F}_p(\mathcal{Q})$ sous FNC sortie : oui si A est satisfiable, non sinon
3-SAT	entrée : $A \in \mathbb{F}_p(\mathcal{Q})$ sous FNC ac des clauses de 3 littéraux seulem^t sortie : oui si A est satisfiable, non sinon
2-SAT	entrée : $A \in \mathbb{F}_p(\mathcal{Q})$ sous FNC ac des clauses de 2 littéraux seulem^t sortie : oui si A est satisfiable, non sinon

Quels liens entre ces variantes ?

Rappel : réduction **polynomiale** entre problèmes



Réduction **polynomiale** entre problèmes de décision



Définition

Soient X et Y deux problèmes de décisions.
On note X^+ (resp. X^-) les instances positives (resp. négatives) de X, i.e. celles pour lesquelles la sol. est vrai (resp. faux). On définit de m̂ Y^+ et Y^-.

*On dit que Y **se réduit** à X **en temps polynomial***
s'il existe une une transformation φ, calculable en temps polynomial, qui transforme toute instance de Y en une instance de X de sorte que

$$\forall \mathcal{I},\ \varphi(\mathcal{I}) \in X^+ \Leftrightarrow \mathcal{I} \in Y^+$$

Réductions entre les variantes de SAT - 1/2

- **SAT** est plus dur que **FND-SAT** et **FNC-SAT**
  $\hookrightarrow$ **FNC-SAT** se réduit à **SAT** (pour $\varphi = id$)
  $\hookrightarrow$ **FND-SAT** se réduit à **SAT** (pour $\varphi = id$)
- **FNC-SAT** est plus dur que **2-SAT** et **3-SAT**
  $\hookrightarrow$ **2-SAT** se réduit à **FNC-SAT** (pour $\varphi = id$)
  $\hookrightarrow$ **3-SAT** se réduit à **FNC-SAT** (pour $\varphi = id$)
- **FNC-SAT** se réduit à **3-SAT**
  $\hookrightarrow$ on transforme chaque clause de taille < 3 en clause de taille 3

$$a \equiv (a \vee a \vee a) \qquad (a \vee b) \equiv (a \vee a \vee b)$$

  $\hookrightarrow$ on transforme chaque clause de taille > 3 en conjonction de clauses de taille 3 en ajoutant des nouvelles variables

$$(a \vee b \vee c \vee d) \rightsquigarrow (a \vee b \vee z) \wedge (\neg z \vee c \vee d)$$

$$(a \vee b \vee c \vee d \vee e) \rightsquigarrow (a \vee b \vee z_1) \wedge (\neg z_1 \vee c \vee z_2) \wedge (\neg z_2 \vee d \vee e)$$

$\hookrightarrow$ transformations en temps poly. qui préservent la satisfiablilité

Exercice - pour la réduction de **FNC-SAT** à **3-SAT**

Soit $C = \ell_1 \vee \ell_2 \vee \ell_3 \vee \dots \ell_k$ une clause de $\mathbb{F}_p(\mathcal{Q})$ de taille $k > 3$.
Soit z une nouvelle variable (*i.e.* $z \notin \mathcal{Q}$).
On pose alors $C_1 = (\ell_1 \vee \ell_2 \vee z)$ et $C_2 = (\neg z \vee \ell_3 \vee \dots \ell_k)$.

Montrer les deux propositions suivantes :

→ $\forall \rho \in \mathbb{B}^{\mathcal{Q}}, [C]^{\rho} = V \Rightarrow \left(\exists \tilde{\rho} \in \mathbb{B}^{\mathcal{Q} \cup \{z\}}, \tilde{\rho}|_{\mathcal{Q}} = \rho \text{ et } [C_1 \wedge C_2]^{\tilde{\rho}} = V \right)$

→ $\forall \tilde{\rho} \in \mathbb{B}^{\mathcal{Q} \cup \{z\}}, [C_1 \wedge C_2]^{\tilde{\rho}} = V \Rightarrow [C]^{\tilde{\rho}} = V$

La même stratégie permet-elle de réduire **FNC-SAT** *à* **2-SAT** *?*

Réductions entre les variantes de SAT - 2/2

- **FNC-SAT** ne se réduit pas à **2-SAT**
  $\hookrightarrow$ on verra que **2-SAT** se résout en temps poly. (cf chap. graphes),
  $\hookrightarrow$ vous verrez que **3-SAT** est NP-complet (th. de Cook, 2eme année)
  $\hookrightarrow$ à moins que P=NP, ces deux problèmes ne sont pas équivalents
- **FNC-SAT** ne se réduit pas à **FND-SAT**
  $\hookrightarrow$ la transfo. d'une forme à l'autre se fait en temps exponentiel et peut aussi augmenter la taille de l'entrée de manière exponentielle
  $\hookrightarrow$ on verra que **FND-SAT** se résout en temps polynomial (juste après),
  $\hookrightarrow$ à moins que P=NP, ces deux problèmes ne sont pas équivalents

À retenir : importance de la forme sous laquelle sont données les entrées d'un problème !

Modéliser une FND

FND = disjonction de conjonctions de littéraux.
D'un point de vue sémantique (*i.e.* dans $\mathbb{F}_p(\mathcal{Q})/\equiv$),

→ l'ordre des littéraux au sein d'une conjonction est-il important ? et leur multiplicité ? **non**, une conjonction est satisfaite par un env. prop. ssi il satisfait l'ensemble de ses termes.

→ quel objet mathématique est adapté pour modéliser les conjonctions de littéraux ? un **ensemble de littéraux** ou un **couple d'ensembles de variables** : d'une part celles apparaissant dans les littéraux positifs, d'autre part celles apparaissant dans les littéraux négatifs

→ l'ordre des conjonctions au sein de la disjonction est-il important ? et leur multiplicité ? **non**, une disjonction est satisfaite par un env. prop. ssi il satisfait l'un de ses termes.

→ quel objet mathématique est adapté pour modéliser une FND ? un ensemble de conjonction

→ quelle structure de données est adaptée pour modéliser une FND ? une liste de liste de littéraux convient

FND-SAT (FNC-SAT)

On suppose que $\mathcal{Q}=\{q_1, q_2, \dots q_N\}$.

FND-SAT

entrée : $\left((\ell_{i,j})_{i\in[1..n_j]}\right)_{j\in[1..n]}$ une famille de littéraux de $\mathcal{Q}$

sortie : oui si $A = \bigvee_{j=1}^{n} \bigwedge_{i=1}^{n_j} \ell_{i,j}$ est satisfiable,

non sinon

FNC-SAT

entrée : $\left((\ell_{i,j})_{i\in[1..n_j]}\right)_{j\in[1..n]}$ une famille de littéraux de $\mathcal{Q}$

sortie : oui si $A = \bigwedge_{j=1}^{n} \bigvee_{i=1}^{n_j} \ell_{i,j}$ est satisfiable,

non sinon

FND-SAT - exemple 1/2

la formule initiale :
$(a \wedge b \wedge \neg c) \vee (\neg b \wedge \neg c) \vee (a \wedge c)$
sous forme d'ensemble d'ensembles de littéraux :
$\{\{a, b, \neg c\}, \{\neg b, \neg c\}, \{a, c\}\}$
sous forme d'ensemble de couples :
$\{(\{a, b\}, \{c\}), (\emptyset, \{b, c\},), (\{a, c\}, \emptyset)\}$

la formule initiale :
$(a \wedge b \wedge \neg c) \vee (a \wedge c \wedge a) \vee (\neg b \wedge \neg c)$
sous forme d'ensemble d'ensembles de littéraux :
$\{\{a, b, \neg c\}, \{\neg b, \neg c\}, \{a, c\}\}$
sous forme d'ensemble de couples :
$\{(\{a, b\}, \{c\}), (\emptyset, \{b, c\},), (\{a, c\}, \emptyset)\}$

FND-SAT - exemple 2/2

la formule initiale :
$(a \wedge b \wedge \neg a) \vee (a \wedge c \wedge \neg a) \vee (\neg b \wedge \neg c)$
sous forme d'ensemble d'ensembles de littéraux :
$\{\{a, b, \neg a\}, \{a, \neg a, \neg c\}, \{\neg b, \neg c\}\}$
sous forme d'ensemble de couples :
$\{(\{a, b\}, \{a\}), (\{a\}, \{a, c\}), (\emptyset, \{b, c\},), \}$

Cette formule est-elle satisfiable ? Justifier.
Par quelle procédure décider ?

Algorithme pour FND-SAT

Algorithme FND-SAT sur $\mathcal{Q}=\{q_1, q_2, \dots q_N\}$

entrée : $\left(\left(\ell_{i,j}\right)_{i\in[1..n_j]}\right)_{j\in[1..n]}$ une famille de littéraux de $\mathcal{Q}$

sortie : vrai si $A = \bigvee_{j=1}^{n} \bigwedge_{i=1}^{n_j} \ell_{i,j}$, faux sinon

Pour j allant de 1 à n :
    Créer T un tableau indicé par $[1..N]$ initialisé à -1
    Essayer :
        Pour i allant de 1 à n_j :
            Si $\ell_{i,j} = q_k$
            alors si $T[k] = -1$, alors $T[k] \leftarrow 1$
                sinon si $T[k] = 0$ alors déclencher "conj. non sat."
            Si $\ell_{i,j} = \neg q_k$
            alors si $T[k] = -1$, alors $T[k] \leftarrow 0$
                sinon si $T[k] = 1$ alors déclencher l'exception "conj. non sat."
        Retourner **vrai**
    Rattraper "conj. non sat."
Retourner **faux**

Modélisation - exemple du solitaire 1/3

État initial du jeu : Sur chaque case d'un damier carré de $m \times m$ cases, il y a une pierre bleue (b), ou bien une pierre rouge (r) ou rien.

But du jeu : enlever des pierres de manière à ce que :

- → sur chaque colonne toutes les pierres sont de la même couleur
- → sur chaque ligne il y a au moins une pierre

Exercice :

1. Formaliser le problème de décision consistant à savoir si une partie peut être gagnée.
2. Réduire ce problème à **3-SAT**
3. Réduire **3-SAT** à ce problème (bonus)

À retenir : puissance d'encodage de **3-SAT**

Modélisation - exemple du solitaire 2/3

1. **JEU**
   entrée : $m \in \mathbb{N}^*$ la taille du damier
   $B \subseteq [1..m]^2$ l'ensemble des cases init. bleues.
   $R \subseteq [1..m]^2 \setminus B$ l'ensemble des cases init. rouges.
   sortie : oui s'il existe $B' \subseteq B$ et $R' \subseteq R$ tel que :
   $\rightarrow \forall j \in [1..m],\ B' \cap [1..n] \times \{j\} = \emptyset$ ou $R' \cap [1..m] \times \{j\} = \emptyset$
   $\rightarrow \forall i \in [1..m],\ (R' \cup B') \cap \{i\} \times [1..m] \neq \emptyset$

2. Soit (m, B, R) une instance de **JEU**.
On cherche à construire, en temps polynomial, une FNC A telle que $A \in$ **3-SAT**$^+$ ssi $(m, B, R) \in$ **JEU**$^+$, c'est-à-dire telle que A est satisfiable ssi il existe $B' \subseteq B$ et $R' \subseteq R$ tel que ...

Pour chaque $(i, j) \in [1..m]^2$, on introduit 2 variables :
- $b'_{i,j}$ modélisant $(i, j) \in B'$
- $r'_{i,j}$ modélisant $(i, j) \in R'$

Modélisation - exemple du solitaire 3/3

On A doit modéliser les 4 contraintes ci-dessous :

→ $B' \subseteq B$

→ $R' \subseteq R$

→ $\forall j \in [1..m],\ B' \cap [1..n] \times \{j\} = \emptyset$ ou $R' \cap [1..m] \times \{j\} = \emptyset$

→ $\forall i \in [1..m],\ (R' \cup B') \cap \{i\} \times [1..m] \neq \emptyset$

On pose alors A la conjonction des 4 formules ci-dessous :

→ $\displaystyle\bigwedge_{(i,j) \in [1..m] \setminus B} \neg b_{i,j}$

→ $\displaystyle\bigwedge_{(i,j) \in \in [1..m] \setminus R} \neg r_{i,j}$

→ $\displaystyle\bigwedge_{j \in [1..m]} \bigwedge_{(i,i') \in [1..m]^2} \neg b_{i,j} \vee \neg r_{i',j}$

→ $\displaystyle\bigwedge_{i \in [1..m]} \bigvee_{j \in [1..m]} r_{i,j} \vee b_{i,j}$