Ministère de l'Education Nationale
Université de Montpellier II
Place Eugène Bataillon
34095 Montpellier Cedex 5

TP FMIN105
Algorithmique / Complexité / Calculabilité

Rapport (Décembre 2011)

Travail préparé par :

Thibaut Marmin
Clément Sipieter
William Dyce

Table des matières

1 Partie théorique **5**

1.1 Partie algorithmique 5

1.2 Partie complexité 5

1.2.1 Réduction de SAT vers 3–SAT 5

1.2.2 NP-complétude de 2–SAT 8

1.2.3 2-SAT, un problème polynomial 8

1.3 Partie calculabilité 8

2 Partie pratique **9**

Chapitre 1

Partie théorique

1.1 Partie algorithmique

1.2 Partie complexité

1.2.1 Réduction de SAT vers 3–SAT

(a) **Énoncé de SAT** :

Données :	$\mathcal{V} = \{v_1, v_2 \dots v_n\}$	*Ensemble de n variables*
	$\mathcal{C} = \{c_1, c_2, c_3 \dots c_m\}$	*Ensemble de m clauses*
où	$c_i = (l_{i1} \vee l_{i2} \vee \dots \vee l_{ik})$	*Clauses de k littéraux*
avec	$l_{ij} = v$ ou $\neg v$	*avec $v \in U$*

Problème : existe-il au moins une affectation des variables telle que chaque clause de $\mathcal{C}$ soit vrai.

Énoncé de 3–SAT :

3–SAT est identique au problème SAT avec $k = 3$.

Données :	$\mathcal{V} = \{v_1, v_2, v_3 \dots v_n\}$
	$\mathcal{C} = \{c_1, c_2, c_3 \dots c_m\}$
où	$c_i = (l_{i1} \vee l_{i2} \vee l_{i3})$
avec	$l_{ij} = v$ ou $\neg v$

(b) La réduction du problème SAT peut être définit en montrant que chaque clause c de $\mathcal{C}$ peut-être transformée en un ensemble de clauses $\mathcal{C}'$ tel que pour toute affectation rendant vrai l'ensemble des clauses de $\mathcal{C}$, on peut trouver une affectation rendant vrai chaque clause de $\mathcal{C}'$. Chaque clause de $\mathcal{C}'$ devant être de taille exactement 3. La réciproque doit également être montrée.

Définissons les réductions :

$\boldsymbol{k = 1}$

Soit ci_1 une clause de taille 1, on a $ci_1 = (l)$. Ajoutons deux variables $v_1, v_2 \notin \mathcal{V}$ et transformons la clause c en quatre clauses. On obtient l'ensemble $\mathcal{C}_1 = \{c_1, c_2, c_3, c_4\}$ avec :

$$c_1 = (l \vee v_1 \vee v_2)$$

$$c_2 = (l \vee v_1 \vee \neg v_2)$$

$$c_3 = (l \vee \neg v_1 \vee v_2)$$

$$c_4 = (l \vee \neg v_1 \vee \neg v_2)$$

$\boldsymbol{k = 2}$

Soit ci_2 une clause de taille 2, on a $ci_2 = (l_1 \vee l_2)$. Ajoutons une variable $v \notin \mathcal{V}$ et transformons la clause c en deux clauses. On obtient l'ensemble $\mathcal{C}_2 = \{c_1, c_2\}$ avec :

$$c_1 = (l_1 \vee l_2 \vee v)$$

$$c_2 = (l_1 \vee l_2 \vee \neg v)$$

$\boldsymbol{k = 3}$

La clause ci_3 ne subit pas de transformation.

$$\mathcal{C}_3 = \{ci_3\}$$

$\boldsymbol{k > 3}$

Soit la clause $ci_k = (l_1 \vee l_2 \vee \cdots \vee l_k)$. On ajoute $(k-3)$ nouvelles variables $(v_1, v_2 \ldots v_{k-3})$.

$$\mathcal{C}_k = (l_1 \vee l_2 \vee v_1) \bigwedge_{i=1}^{k-4} (\neg v_i \vee l_{i+2} \vee v_{i+1}) \wedge (\neg v_{k-3} \vee l_{k-1} \vee l_k)$$

Montrons que SAT est vrai si et seulement si 3–SAT est vrai :

SAT → 3–SAT

- Soit une interprétation I_1 qui satisfasse la clause ci_1 :

  $$val(I_1, ci_1) = val(I_1, l) = vrai$$

  Prenons une interprétation I'_1 avec $val(I_1, l) = val(I'_1, l)$, peu importe les affectations de v_1 et v_2, l étant présent dans toutes les clauses de $\mathcal{C}_1$:

  $$val(I'_1, \mathcal{C}) = vrai$$

- Soit une interprétation I_2 qui satisfasse la clause ci_2 :

  $$\exists i, val(I_2, l_i) = vrai$$

  Prenons une interprétation I'_2 avec :

  $$val(I_2, l_1) = val(I'_2, l_1)$$

  $$val(I_2, l_2) = val(I'_2, l_2)$$

  Peu importe l'affectation de v dans I'_2, on a $val(I'_2, \mathcal{C}_2) = vrai$.

– Soit une interprétation I_k qui satisfasse la clause ci_k :

$$\exists i, val(I_k, l_i) = vrai$$

Prenons une interprétation I'_k telle que :

$$\begin{array}{rcl} val(I_k, l_i) & = & val(I'_k, l_i) \\ \forall j \in [1;(i-2)], val(I'_k, v_j) & = & vrai \\ \forall j \in [(i-1);(k-3)], val(I'_k, v_j) & = & faux \end{array}$$

On obtient :

$$val(I'_k, \mathcal{C}_k) = vrai$$

3–SAT → SAT

– Prenons une interprétation I_1 telle que $val(I_1, \mathcal{C}_1) = vrai$.
Sans perte de généralité, on suppose que :

$$val(I_1, v_1) = val(I_1, v_2) = vrai$$

La clause c_4 de $\mathcal{C}_1$ ne peut être satisfaite que si $val(I_1, l) = vrai$.
On a donc :

$$val(I_1, ci_1) = vrai$$

– Prenons une interprétation I_2 telle que $val(I_2, \mathcal{C}_2) = vrai$.
Sans perte de généralité on suppose que :

$$val(I_2, v) = vrai$$

La clause c_2 de $\mathcal{C}_2$ ne peut être satisfaire que si $val(I_2, (l_1 \vee l_2)) = vrai$.
On a donc :

$$val(I_2, ci_2) = vrai$$

(c) Le point (b) définit la réduction de SAT vers 3–SAT. Afin de montrer la NP-Complétude de 3–SAT, montrons que la réduction s'effectue en un temps polynomial.
Soit :
k la taille de la clause initiale,
v_k le nombre de variables à ajouter pour obtenir des clauses de taille 3,
w_k le nombre de clauses de taille 3 obtenues à partir de la clause initiale.

$$\begin{array}{ll} v_3 = 0 & w_3 = 1 \\ v_4 = 1 & w_4 = 2 \\ v_5 = 2 & w_5 = 3 \\ \vdots & \vdots \end{array}$$

Pour tout $k > 3$:

$$v_k = v_{\lceil \frac{k}{2} \rceil + 1} + v_{\lfloor \frac{k}{2} \rfloor + 1} + 1$$
$$w_k = w_{\lceil \frac{k}{2} \rceil + 1} + w_{\lfloor \frac{k}{2} \rfloor + 1}$$

$v_k = \theta(k)$, donc borné par la taille de F. La réduction s'effectue donc en un temps polynomial.
Il est possible de réduire le problème SAT à 3–SAT en un temps polynomial, SAT étant NP-complet, 3–SAT l'est aussi.

1.2.2 NP-complétude de 2–SAT

1. coucou
2. coucou

1.2.3 2-SAT, un problème polynomial

1. coucou
2. coucou

1.3 Partie calculabilité

Chapitre 2

Partie pratique