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
Williman l'Australien

Table des matières

1 Partie théorique **5**

1.1 Partie algorithmique 5

1.2 Partie complexité 5

1.2.1 Réduction de SAT à 3–SAT 5

1.2.2 NP-complétude de 2–SAT 6

1.2.3 2-SAT, un problème polynomial 7

1.3 Partie calculabilité 7

2 Partie pratique **9**

Chapitre 1

Partie théorique

1.1 Partie algorithmique

1.2 Partie complexité

1.2.1 Réduction de SAT à 3–SAT

(a) **Énoncé de SAT** :

Données :	$U = \{v_1, v_2 \dots v_n\}$	*Ensemble de n variables*
	$F = \{c_1, c_2, c_3 \dots c_m\}$	*Ensemble de m clauses*
où	$C_i = l_{i1} \vee l_{i2} \cdots l_{ik}$	*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 F soit vrai.

Énoncé de 3–SAT :

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

Données :	$U = \{v_1, v_2, v_3 \dots v_n\}$
	$F = \{c_1, c_2, c_3 \dots c_m\}$
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 F peut-être transformée en un ensemble de clauses $\{c_1, c_2, \dots, c_i\}$ tel que pour toute affectation rendant vraie C, on peut trouver une affectation rendant vrai chaque c_i. Chaque c_i devant être de taille exactement 3. La réciproque doit également être montrée.

$k = 4$

Soit C une clause de taille 4 : $C = l_1 \vee l_2 \vee l_3 \vee l_4$. Ajoutons une variable $v \notin U$ et transformons la clause C en deux clauses :

$$C_1 = l_1 \vee l_2 \vee v$$

$$C_2 = l_3 \vee l_4 \vee \neg v$$

4 ⇒ 3 Soit une affectation qui rend C vrai. Il existe un i tel que l_i est vrai.

Sans perte de généralité, prenons $i = 1$. Conservons les mêmes affectations pour les l_i avec $v = faux$.

On a $C_1 \wedge C_2 = vrai$.

3 ⇒ 4 Soit une affectation telle que $C_1 \wedge C_2$ est vrai.

Sans perte de généralité, on suppose que v est vrai. Cela implique que $l_3 \vee l_4$ est vrai, et donc que C est vrai.

$\boldsymbol{k = 5}$

Soit la clause $C = l_1 \vee l_2 \vee l_3 \vee l_4 \vee l_5$, sur le même principe que pour $k = 4$:

$$C \rightarrow (l_1 \vee l_2 \vee l_3 \vee v) \wedge (l_4 \vee l_5 \vee \neg v)$$
$$\rightarrow (l_1 \vee l_2 \vee w) \wedge (l_3 \vee v \vee \neg w) \wedge (l_4 \vee l_5 \vee \neg v)$$

La clause C est transformée en 3 clauses de taille 3, en ajoutant 2 nouvelles variables.

Généralisation :

Une clause de taille $k > 3$ est transformée en 2 clauses C_1 et C_2 en ajoutant une variable :
- Taille de C_1 : $\lceil \frac{k}{2} \rceil + 1$
- Taille de C_2 : $\lfloor \frac{k}{2} \rfloor + 1$

Avec $k > 3$, chaque réduction diminue stictement la taille des clauses, car on a :

$$k > \left\lceil \frac{k}{2} \right\rceil + 1 \geq \left\lfloor \frac{k}{2} \right\rfloor + 1$$

Tout problème SAT peut dont être réduit à un problème 3–SAT.

(c) Le point (b) définit la réduction de SAT à 3–SAT.

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 $k > 3$:

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

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