TD1 : Machines de Turing et problèmes NP-complets

Exercice 1

Soit la MT suivante :

	$\sqcup$	0	1
z_0		(z_1, 1, D)	
z_1	(z_h, $\sqcup$, D)		(z_0, 0, D)

Quel est le langage reconnu ?

Exercice 2

- Concevoir une MT acceptant le langage $\{0^k 1^k \mid k \geq 1\}$
- Concevoir une MT reconnaissant les chaînes comportant autant de "0" que de "1"
- Concevoir une MT acceptant le langage $\{u\#\bar{u} \mid u \in \{0,1\}^*\}$

Exercice 3

Un problème B est *NP-complet* s'il satisfait deux conditions :
- B est dans NP et
- Tout problème A dans NP est *réductible en temps polynomial* à B.

Par ailleurs, on montre que si B est NP-complet et que B est réductible en temps polynomial à C dans NP alors C est NP-complet. La classe des problèmes NP-complet est importante car si l'on trouve pour l'un d'eux un algorithme polynomial alors N = NP.

Le problème SAT est NP-complet :
- Soit x_i une variable propositionnelle prenant les valeurs *vrai* ou *faux*
- Une clause est une disjonction de variables propositionnelles ou leurs négations :

$$(x_1 \vee x_3 \vee \bar{x}_4)$$

- Une formule propositonnelle est en Forme Normale Conjonctive (CNF) si elle est représentée sous la forme d'une conjonction de clauses :

$$(x_1 \vee x_3 \vee \bar{x}_4) \wedge (x_4) \wedge (x_2 \vee \bar{x}_3)$$

- Le problème SAT est un problème de décision visant à savoir s'il existe une assignation des variables propositionnelles rendant une formule CNF *vrai*

Sur une grille $n \times n$, on dispose des jetons bleus ou rouges. Certaines cases peuvent rester vides. Le jeu consiste à retirer des jetons de la grille de telle sorte que toutes les colonnes ne contiennent que des jetons de la même couleur et que chaque ligne comporte au moins un jeton. On gagne si on atteint cet objectif. Il n'existe pas de solutions pour toutes les configurations de la grille. SOLITAIRE est le problème de décision consistant à dire si pour une configuration donnée il existe une stratégie de jeu gagnante.

Montrez que SOLITAIRE est NP-complet.

Corrections

Exercice 1 Le langage reconnu correspond à l'expression régulière **(01)*0**

Exercice 2 Voir les figures 1 à 3

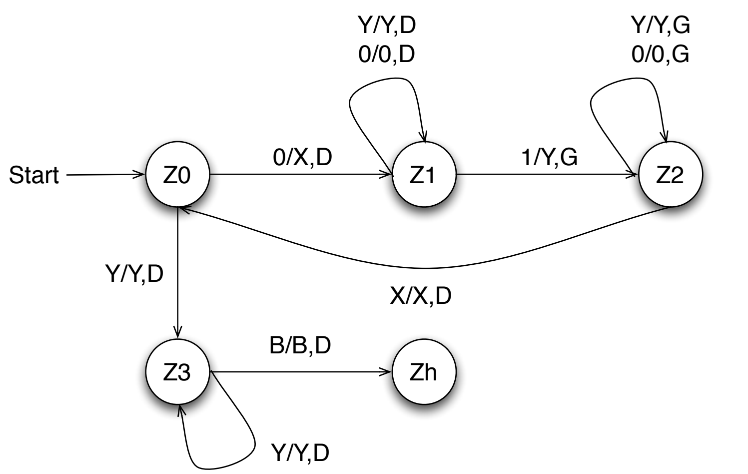

FIG. 1 – MT acceptant le langage $\{0^k1^k \mid k \geq 1\}$

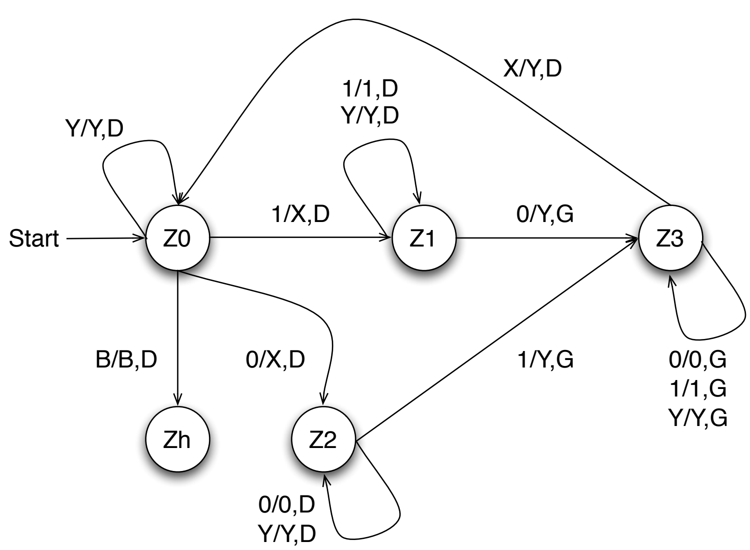

FIG. 2 – "Autant de 0 que de 1"

Exercice 3 Tout d'abord, SOLITAIRE est dans NP car il est possible de vérifier si une stratégie est gagnante en temps polynomial. On montre ensuite que 3SAT est réductible en temps polynomial à SOLITAIRE. Etant donné une formule ϕ avec m variables $x_1, \cdots, x_m$ et k clauses $c_1, \cdots, c_k$, on construit une grille $k \times m$ de la manière suivante : si x_i est dans la clause c_j, mettre un jeton bleu dans la rangée c_j, colonne x_i ; si $\bar{x_i}$ est dans la clause c_j, mettre un jeton rouge dans la rangée c_j, colonne x_i. Il est toujours possible de rendre la grille carrée en recopiant une rangée ou en ajoutant une

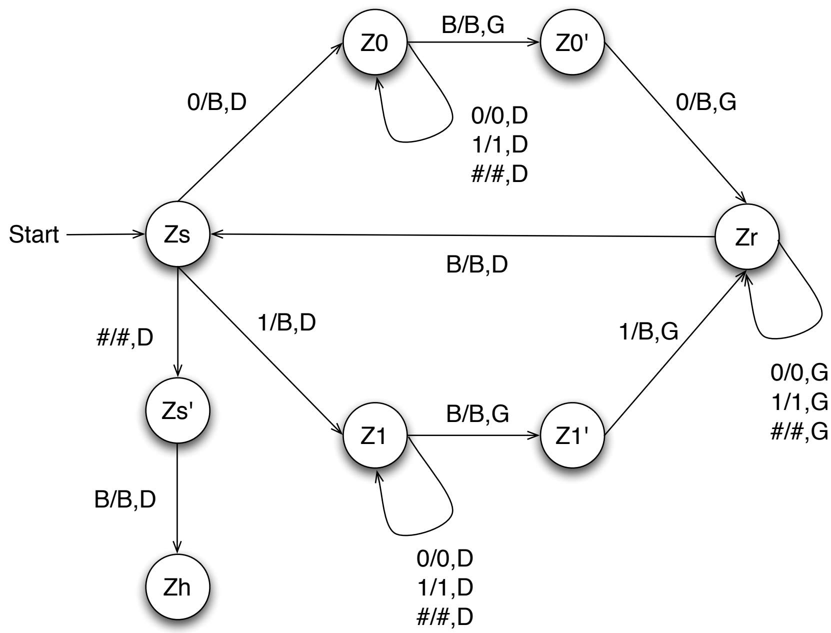

FIG. 3 – MT acceptant le langage $\{u\#\bar{u} \mid u \in \{0,1\}^*\}$

colonne vide sans que cela n'affecte le jeu.

On montre pour finir que ϕ est satisfiable si et seulement si il existe une stratégie gagnante :

($\rightarrow$) Soit une assignation satisfiant ϕ. Si x_i est vrai (faux), retirer les jetons rouges (bleus) dans la colonne correspondant. Par conséquent, les jetons correspondant aux litéraux vrai restent en place. Comme chaque clause a un litéral vrai, chaque rangée a un jeton.

($\leftarrow$) Soit une stratégie gagnante. Si les jetons rouges (bleus) sont retirés d'une colonne, assigner à la variable correspondante la valeur vrai (faux). Chaque rangée a un jeton, donc chaque clause a un litéral vrai. Par conséquent, ϕ est satisfaite.