On poursuit notre route vers notre résultat d'indécidabilité selon lequel on ne peut pas décider de tous les langages.

Or, au vu de la première partie du cours, ce fait semble quelque peu trivial. En effet :

1) L'ensemble des machines de Turing est dénombrable.

Il convient de noter qu'une machine de Turing est codable sur un alphabet fini (puisqu'on les décrit par un nombre fini de transitions) et ce par un mot. Or, avec l'alphabet $\Sigma = \{0,1\}$, on est capable de tout coder. En particulier, une machine de Turing est codable par un mot binaire.
On considère donc un ensemble de mots finis définis sur un alphabet fini, ce qui est nécessairement dénombrable.

2) L'ensemble des langages sur $\Sigma = \{0,1\}$ n'est pas dénombrable.
3) L'ensemble des mot Σ^* est dénombrable. On pose :

	Σ^* :	w0	w1	w2	w3	w4	w5	w6	....	wi	....
ML		0, 0	0	1	0	0	1	0	....	0 ou 1	...
ML'		0, 0	0	1	0	0	0	0	....		

Si ML $\neq$ ML' , alors L $\neq$ L'

Autrement dit, par un résultat de dénombrabilité, on se rend vite compte que décrire un ensemble non dénombrable (= ensemble des langages sur un alphabet) à l'aide d'un ensemble dénombrable (= ensemble des machines de Turing) est impossible…

En revanche, l'ensemble des mots issus d'un alphabet est dénombrable. On peut alors transformer la question ainsi : est-il possible de tout coder sur un alphabet ?

Les Machines de Turing universelles :

Remarque : on peut tout coder sur l'alphabet $\Sigma = \{0,1\}$

Soit $\Sigma' = \{ a1, a2, \ldots\ldots\ldots, a6, \ldots\ldots\ldots\ldots\ldots, ak \}$ On va coder les ai par le codage binaire de i
Exemple : a13 est codé par 1101

Comment coder les mots sur Σ' ?

Idée 1 :

Le mot w1 = a6 a4 a3 sera codé 11010011 (on concatène codage de chaque ai)
Mais on a un problème puisqu'en l'espèce :
Le mot w2 = a52 a1 a1 se code de la même manière 11010011

Idée 2 :

On utilise un séparateur. On pose : 1 → 01 , 0 → 00 , sep → 11 ou 10
On aura alors par exemple :
w1 = 010100110100000110101
w2 = 010100010000011011101

Cette fois ci on a bien une différence, due aux séparateurs.

Question : Est-il possible d'encoder un mot fini sur un alphabet fini quelconque en utilisant un seul caractère 'q' ?

Idée : Codage de Turing par un mot fini sur un alphabet fini

$(q_i, a_j, q_k, a_l$, L ou R) (q, a, …, …, …) ………. $\#q_a q_b \ldots q_s$ $\#q_0$

On peut vérifier par une machine de Turing qu'un mot donné est le codage (ou non) d'une machine de Turing dans le format précédent.

Pour ça, il suffit de :

1) Vérifier que le mot ait une juxtaposition de parenthèses qui contiennent exactement 4 ','
2) Vérifier qu'il y ait exactement 2 '#'
3) Vérifier qu'il n'y ait pas de '(' , ')' , ',' après le premier # (il n'y a que des 0 et des 1 après)
4) Vérifier qu'entre ')' et la ',' précédente il y ait seulement L ou R

Maintenant on peut faire une machine de Turing universelle qui prend en entrée un mot w'#w (avec w qui ne contient pas '#') et qui :

1) Vérifie que w' ait le codage valide, dans le format précédent, d'une machine de Turing qu'on note M
2) Vérifie que w ne contienne que des 0 et des 1
3) Simule le comportement

Comment simuler M sur w :

$\underbrace{\boxed{\text{(qi, aj, qk, al , L ou R) (…. , …. , …, …, R) (…, ……….)} \quad \text{\#qaqb}}\ \text{\#q0}}_{w'} \quad \underbrace{\text{\#011010}}_{w}$

Codage d'une configuration d'une machine de Turing : q_{cour} # w1 ▲ w2 …… wk

Pour simuler :

1) On commence avec la configuration inutile de M

2) A chaque étape, quand on est dans une configuration courante, a_c # m1 ▲ a m2 , on cherche une transition qui commence par (a_c, a, …..)

3) S'il en existe une (a_c, a, q'_c, b, Dir) alors:
- On change q_c en q'_c dans la configuration courante
- On change a en b dans la configuration courante
- On bouge ▲ dans la direction Dir dans la configuration courante

4) Sinon (i.e. s'il n'existe pas de transition) on s'arrête.