Retour sur les machines de Turing universelles :

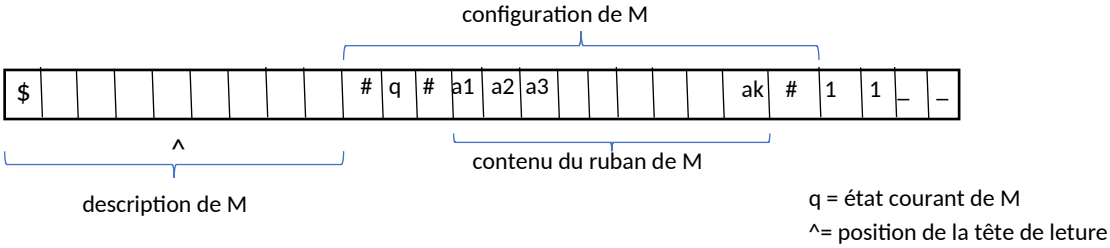
On a vu qu'on pouvait encoder une machine de Turing en donnant toutes ses transitions sous forme de quintuplet (q0, a, q', b, Dir).
De plus, on sait qu'on peut coder n'importe quel mot sur l'alphabet {0,1} avec un système d'encodage et de séparateurs.

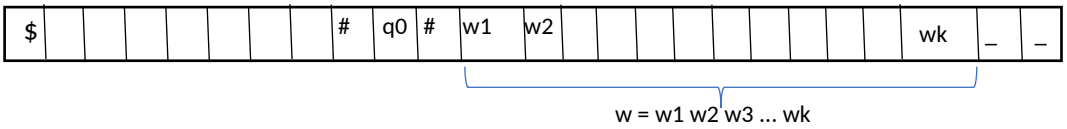
Si M_{uni} est une machine de Turing universelle, elle prendra en entrée < M > # w où :

- <M> correspond à l'encodage de M SUR {0,1} *
- W correspond à un mot sur {0,1} *

Elle va alors simuler M sur w.

M_{uni} : Simulation de M sur w





Proposition :

Soit L un langage.
Si L est décidé par une machine de Turing (i.e. la machine de Turing qui décrit L s'arrête **toujours**), alors L est reconnu par une machine de Turing (i.e. la machine a le droit de **ne pas s'arrêter** lorsque le mot n'est pas dans L).
Le fait de ne pas s'arrêter n'est d'ailleurs pas effectif…

Remarque :
Par contraposée, si L n'est pas reconnu par une machine de Turing, alors L n'est pas décidé par une telle machine.

On va maintenant montrer qu'il existe des langages qui ne sont pas décidables et même plus, qui ne sont pas reconnus.
Pour ce faire, on va de nouveau passer par un processus de diagonalisation.

On peut déjà noter que l'ensemble des mots finis sur un alphabet fini (ici {0,1}) est dénombrable. Il en va de même de l'ensemble des machines de Turing (puisqu'elles sont toutes codables par un mot fini sur {0,1}).

Soient deux énumérations :

- W1 w2 w3 ….. des mots finis sur {0,1}
- M1 M2 M3 ….. des machines de Turing

On pose la matrice A :

	w1	w2	…	wi	…	wk	…
M1	0	N	…	N	…	0 ou N	…
M2	0	N	…	0	…	0 ou N	…
…	…	…	…	…	…	…	…
Mi	0	N	…	0	…	0 ou N	…
…	…	…	…	…	…	…	…
Mk	0 ou N	0 ou N	…	0 ou N	…	0 ou N	…
…	…	…	…	…	…	…	…

A est telle que : A[Mi , Wj] = 0 si Mi accepte Wj (i.e. Mi s'arrête et répond "oui")
N sinon

Exercice :

Soit L0 le langage défini par L0 = { W ∈ {0,1} * | W = Wi et A[Mi, W] = N }
Montrez que L0 n'est pas reconnaissable par une machine de Turing.

Par l'absurde, on montre que L0 n'est pas reconnaissable par une machine de Turing.
Supposons qu'il existe une machine de Turing M qui reconnaît L0.
Alors Wj ∈ L0 implique que M accepte Wj (car M reconnaît L0).
Autrement dit : A[Mj, Wj] = 0
Pourtant, par définition de L0, on est censé avoir A[Mj, Wj] = N si Wj ∈ L0
On a donc une contradiction !
Conclusion : Il n'existe pas de machine de Turing qui reconnaît L0

Autre possibilité : On aurait pu le faire ainsi aussi
Wj ∉ L0 implique que M n'accepte pas Wj (car M reconnaît L0).
Alors on a que A[Mj, Wj] = N (par définition de A) ce qui implique que Wj ∈ L0 (par définition de L0).
On a de nouveau une contradiction qui nous amène à la même conclusion.

Finalement : L0 n'est pas reconnaissable par une machine de Turing donc (par la proposition précédente), L0 n'est pas décidable.

Remarque :
Cette preuve est très proche de celle fait pour la dénombrabilité de $\mathbb{R}$ avec la diagonale de Cantor. On prend le langage complémentaire à celui qu'on forme sur la diagonale de A (si on a un 0 il devient N et inversement, si on a un N, il devient 0).

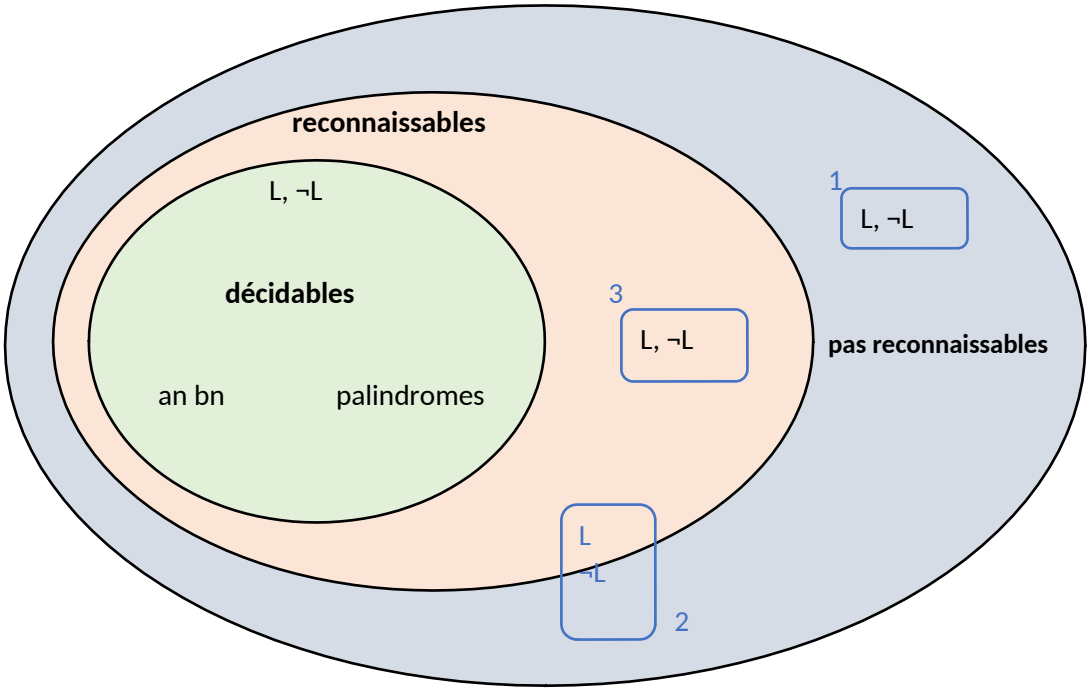
On note alors qu'aucun langage sur les lignes ne correspond à ce langage complémentaire de la diagonale (on n'a pas réellement des langages sur les lignes de A mais des machines qui reconnaissent un langage unique ; on se permet donc cet abus de langage).
L'idée est toujours la même, on prend une machine qui reconnaît L0 (c'est le complémentaire de la diagonale par définition de L0) et on montre qu'elle n'appartient pas aux lignes de A : donc que L0 n'est pas reconnaissable (et par extension pas décidable).

Remarque :
Cette technique de preuve par l'absurde et d'extraction diagonale ressemble également beaucoup au paradoxe de Russel. Ce procédé est proche de la définition des ensembles ne se contenant pas eux-mêmes, et les paradoxes associés se retrouvent donc par analogie.
C'est une autre preuve possible.

Question :
Existe-t-il des langages reconnaissables mais pas décidables ? On va voir que oui et en donner un exemple.

Lemme :
Le complément ¬L d'un langage L qui est décidable est décidable.
Preuve :
On définit ¬L comme : $\{w \mid w \notin L\}$
Soient M' la machine qui décide ¬L et M la machine qui décide L
M' correspond à la machine M dans laquelle on a échangé les états finaux et non finaux. De manière plus formelle : $F' = Q \setminus F$ avec $M = (Q, \Sigma, \#, \Gamma, \delta, q_0, F)$ et
$M' = (Q', \Sigma, \#, \Gamma, \delta, q_0, F')$

Schéma :



Lemme :
Pour tout langage L, si L et ¬L sont reconnaissables alors L est décidable et donc ¬L aussi d'après le lemme précédent.
Dans le schéma ci-dessus, on se rend alors bien compte que la situation 3 ne peut pas arriver.

Exercice :

Soient deux machines M et M' telles que M reconnaît L et M' reconnaît ¬L. Montrez qu'il existe une machine M'' qui décide L.
M'' simule M et M' en faisant une étape de calcul alternativement dans chacune d'entre elles :

- Une étape dans M
- Puis une étape dans M'
- Etc

M'' s'arrête la première fois que l'une de M et M'' s'arrête. Alors M'' répond la même chose que la machine M ou M' qui s'est arrêtée la première.

Exercice :

Soit ¬L0 = { W | W = Wi et A[Mi, Wi] = 0 }
L0 est non décidable. On a donc que ¬L0 est soit reconnaissable non décidable ; soit non reconnaissable.
Montrez que ¬L0 est reconnaissable.

Commençons par dresser une machine de Turing M0 qui reconnaît ¬L0

On a la matrice représentative :

	W1	W2	W3	…	Wi	…
M1	O ou N	O ou N	O ou N	…	O ou N	…
M2	O ou N	O ou N	O ou N	…	O ou N	…
M3	O ou N	O ou N	O ou N	…	O ou N	…
…	…	…	…	…	…	…
Mi	O ou N	O ou N	O ou N	…	O ou N	…
…	…	…	…	…	…	…

On pose :

$M_{enum}W$ = une machine qui énumère les Wi dans l'ordre

$M_{enum}M$ = une machine qui énumère les Mi dans l'ordre

M0 examine les Wi comme $M_{enum}W$. C'est à dire qu'à chaque étape, elle compare Wi et W et elle trouve le i tel que W = Wi.

Ensuite, M0 énumère les Mi comme $M_{enum}M$. Autrement dit, à chaque étape, M0 compare i et j quand j=i. Elle s'arrête et elle a alors W = Wj et Mj sur son ruban.

Enfin, elle simule Mj sur Wj comme une machine de Turing universelle.