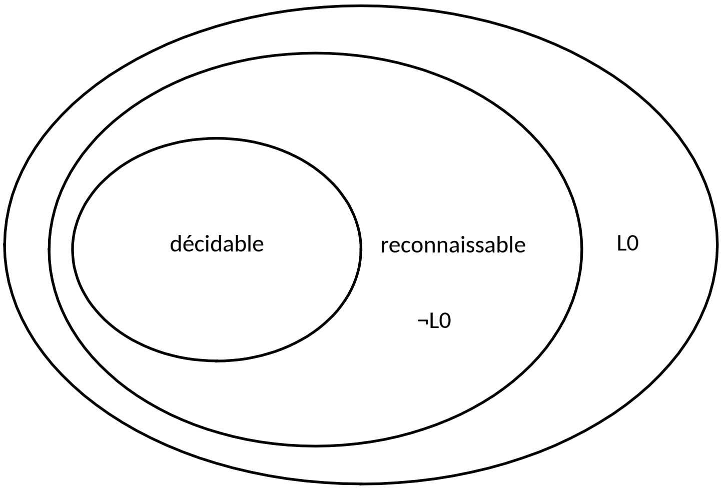
On a vu qu'on pouvait faire une énumération des mots finis sur {0,1} notée $(W_i)_{i\in\mathbb{N}}$ ainsi qu'une énumération des machines de Turing sur l'alphabet d'entrée {0,1} notée $(M_i)_{i\in\mathbb{N}}$

On pose $L_0 = \{ w \mid w = W_i \text{ et } A[M_i, W_i]=N \}$, un langage indécidable et même non reconnaissable

Théorème : L_0 n'est pas reconnaissable par une machine de Turing

<u>Schéma :</u>



On a donc que $\neg L_0 = \{ w \mid w = W_i \text{ et } A[M_i, W_i] = O \}$ est reconnaissable mais indécidable

Théorème : $\neg L_0$ est reconnaissable par une machine de Turing

En effet, on peut décrire cette machine M ainsi :

- Énumérer les W_j jusqu'à W pour déterminer i tel que $W_i = W$
- Énumérer les M_j jusqu'à M_i (où i correspond à celui trouvé à l'étape précédente). On a donc $W=W_i$ et on connaît M_i
- M_i reconnaît W_i ?

On utilisera une machine de Turing universelle pour simuler M_i sur W_i. Alors M accepte W si et seulement si M_i accepte W_i.

Attention : on autorise que la machine ne s'arrête jamais, sinon on est dans le cas décidable !

Remarque : on simule une machine si elle est écrite sur le ruban, on l'exécute sinon

La réduction :

Soit L_1 un langage indécidable. On peut montrer que L_2 est aussi indécidable de la façon suivante : on suppose qu'il existe un algorithme pour décider L_2 et

- On fait un algorithme M_1 qui décide L_1 en utilisant un algorithme qui décide L_2
- On conclut qu'il n'existe pas de machine de Turing qui décide L_2.

Exercice : Soit L_A = { <M>#W | M accepte W } . Montrer que L_A est indécidable par réduction.

(<M> correspond à la description binaire de la machine de Turing M et en particulier <M>#W correspond à la description binaire de la juxtaposition de M et W, qu'on sépare par #. On peut aussi la noter <M,W>)

On suppose qu'il existe M_A qui décide de L_A et on construit M qui décide de $\neg L_0$:

- Enumérer les W_j pour déterminer i tel que W_i = W
- Énumérer les M_j jusqu'à M_i (où i est celui trouvé à l'étape précédente). On a donc W=W_i et on connaît M_i
- Construire le mot <M_i>#W_i et exécuter M_A sur <M_i>#W_i
- Si M_A accepte <M_i>#W_i alors M accepte W = W_i
  Si M_A rejette <M_i>#W_i alors M rejette W = W_i

On aurait alors $\neg L_0$ décidable, ce qui est une contradiction. Donc L_A est indécidable.

Exercice : Soit H = { <M, W> | M s'arrête sur W }. Montrer que H est indécidable par réduction à L_A

On suppose qu'il existe M_H qui décide de L_H et on construit M' qui décide de L_A :

- On prend un mot de la forme <M>#W en entrée
- On exécute M_H avec <M>#W en entrée
- Si M_H renvoie OUI :
  On simule M sur W avec une machine de Turing universelle.
  - Si M renvoie OUI : on retourne OUI et M' accepte <M>#W
  - Si M renvoie NON : on retourne NON et M' rejette < M>#W

(remarque : on est dans le cas où M_H s'arrête donc on peut utiliser MdT universelle)

Si M_H renvoie NON : on retourne NON et M' rejette <M>#W

On aurait alors L_A décidable, ce qui est une contradiction. Donc L_H est indécidable.

Exercice : Soit Hε = { <M> | M s'arrête sur ε }. Montrer que Hε est indécidable par réduction de H.

On suppose qu'il existe $M_{H\varepsilon}$ qui décide Hε et on construit une machine M' qui décide H :

- M' prend en entrée un mot de la forme <M,W>
- M' écrit sur son ruban la description <M''> d'une machine M'' qui :
  - Efface le contenu de son ruban
  - Écrit W au début de son ruban
  - Exécute M à partir de ce qui est sur le ruban
- M' exécute $M_{H\varepsilon}$ sur <M''>
- M' accepte <M,W> si et seulement si $M_{H\varepsilon}$ accepte <M''>

(remarque : M'' s'arrête sur ε si et seulement si M s'arrête sur W)

On aurait alors H décidable, ce qui est une contradiction. Donc Hε est indécidable.

Autre preuve : avec le paradoxe de Russel

On reprend L_A = { <M,W> | M accepte W }

Théorème : L_A est reconnaissable par une machine de Turing

Preuve : Par une machine de Turing universelle on simule M sur W :

- SI M accepte W, on accepte W

- S M s’arrête et répond NON, on rejette W

Théorème : L_A n’est pas décidable
Preuve : On le montre par l’absurde.

- On suppose qu’il existe M_A qui décide de L.
- Grâce à M_A, on construit la machine D qui prend en entrée un mot W ∈ {0,1}* et qui vérifie que W est un codage valide d’une machine de Turing (i.e. : il existe une machine de Turing M telle que W = <M>) :
    - o Si W n’est pas un codage un valide de machine de Turing, alors D rejette W
    - o SI W est le codage valide d’une machine de Turing, alors elle exécute M_A sur <W,W> :
        - Si M_A accepte <W,W> alors D rejette W
        - Si M_A rejette <W,W> alors D accepte W

On fait ici une analogie avec le paradoxe du barbier de Russel. En effet, on peut voir D comme le barbier (qui rase exactement les W qui ne se rasent pas eux-mêmes).
Le cas où M_A accepte <W,W> correspond au cas où W se rase lui-même et le cas M_A rejette <W,W> correspond au cas où W ne se rase pas lui-même.

On se demande si D accepte <D> :

- Si D accepte <D> : alors M_A rejette < <D>, <D> > (par définition de D) et donc D rejette <D> (par définition de M_A) => contradiction
- Si D rejette <D> : alors M_A accepte < <D>, <D> > (par définition de D) et donc D accepte <D> (par définition de M_A) => contradiction

Conclusion : il n’existe pas de machine M_A qui décide de L_A, donc L_A est indécidable.