

Calculabilité et décidabilité



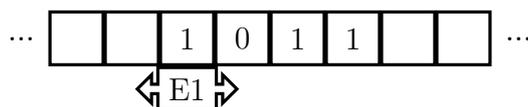
Exercice 1

Compléter la table de transition ci-dessous, afin que la machine effectue une multiplication par 2 d'un nombre binaire écrit sur le ruban à l'état initial.

| état | lu | écrit | déplacement | état suivant |
|------|-------|-------|-------------|--------------|
| E1 | 0 | ... | ... | ... |
| E1 | 1 | ... | ... | ... |
| E1 | blanc | ... | ... | ... |

Dans l'état initial, la tête est sur le bit de poids fort.

Ainsi, par exemple l'état initial est le suivant pour le nombre $11 = (1011)_2$:

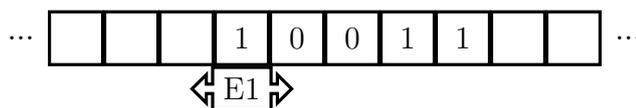


Exercice 2

La table de transition d'une machine de Turing est la suivante :

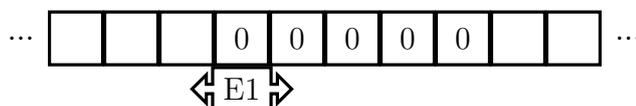
| état | lu | écrit | déplacement | état suivant |
|------|-------|-------|-------------|--------------|
| E1 | 0 | 1 | → | E1 |
| E1 | 1 | 0 | → | E1 |
| E1 | blanc | blanc | ← | E2 |
| E2 | 0 | 1 | ← | E3 |
| E2 | 1 | 0 | ← | E2 |
| E2 | blanc | blanc | → | F |
| E3 | 0 | 0 | ← | E3 |
| E3 | 1 | 1 | ← | E3 |
| E3 | blanc | blanc | → | F |

1. Sachant que la machine est dans l'état initial suivant :



Quelle est la situation à la fin ?

2. Même question dans la situation initiale suivante :



3. Que fait cette machine de Turing ?


```

    if j == i:
        lt = 4 if v is None else 3+len(str(v))
        s1=(lt-len(e))//2
        print(s1*" ",e,sep="")
        break
    else:
        lt = 4 if v is None else 3+len(str(v))
        print(lt*" ",end="")

def affiche_machine(tete,ruban):
    '''affiche la machine, c'est à dire le ruban et la tête'''
    affiche_ruban(ruban)
    affiche_tete(tete,ruban)

def ecrit(ruban,i,w):
    '''écrit sur le ruban la valeur w à l'indice i'''
    pass

def deplace(i,m,ruban):
    '''retourne le nouvel indice de la tête
    et le ruban éventuellement rallongé'''
    if m==">": # déplacement à droite
        if i==len(ruban)-1:
            return ...,...
        else:
            return ...,...
    elif m=="<": # déplacement à gauche
        if i==0:
            return ...,...
        else:
            return ...,...
    else: # aucun déplacement
        return i,ruban

def execute(ruban,tete,etatsFinaux,table):
    '''ruban est une liste non vide
    tete est un couple (indice,état)
    etatsFinaux est une liste d'états finaux
    table est un dictionnaire de dictionnaires de triplets
    dont un exemple est donné plus haut.
    Cette fonction exécute la machine selon les règles
    données dans la table.
    Après chaque étape elle fait afficher l'état de la machine'''
    print("état initial :")
    affiche_machine(tete,ruban)
    while tete[1] not in etatsFinaux:
        i,e=tete
        lu=... # récupère la valeur lue par la tête sur le ruban
        w,m,s=... # récupère l'action à effectuer selon la table
        ecrit(ruban,i,w)
        i,ruban=deplace(i,m,ruban)

```

```
tete=... # nouvelle situation de la tête
affiche_machine(tete,ruban)
```

```
execute([None,None,None,1,0,0,1,1,None,None],(3,"E1"),["F"],tableEx2)
execute([None,None,None,0,0,0,0,0,None,None],(3,"E1"),["F"],tableEx2)
# On peut aussi exécuter cela :
# execute([1,0,0,1,1],(0,"E1"),["F"],tableEx2)
# execute([0,0,0,0,0],(0,"E1"),["F"],tableEx2)
```

Compléter le code et l'exécuter pour vérifier son bon fonctionnement.
On pourra également définir la table de l'exercice 3 et la tester.

Exercice 5 (Facultatif)

Définir la table de transition d'une machine de Turing qui décale son entrée d'une case vers la droite. La position initiale est sur le bit de poids fort.

Exercice 6 (Facultatif)

Définir la table de transition d'une machine de Turing qui, si l'entrée contient un 0, affiche un 1 en tête de l'entrée, et sinon affiche un 0 en tête de l'entrée.

Exercice 7 (Facultatif)

Définir la table de transition d'une machine de Turing qui, si l'entrée contient deux 0 d'affilée, affiche un 1 en tête de l'entrée, et sinon affiche un 0 en tête de l'entrée.

Exercice 8 (Facultatif, difficile)

On considère la table de transition suivante :

| état | lu | écrit | déplacement | état suivant |
|------|-------|-------|-------------|--------------|
| P | 0 | blanc | → | D0 |
| P | 1 | blanc | → | D1 |
| P | blanc | 1 | · | F |
| D0 | 0 | 0 | → | D0 |
| D0 | 1 | 1 | → | D0 |
| D0 | blanc | blanc | ← | V0 |
| D1 | 0 | 0 | → | D1 |
| D1 | 1 | 1 | → | D1 |
| D1 | blanc | blanc | ← | V1 |
| V0 | 0 | blanc | ← | R |
| V0 | 1 | blanc | ← | E |
| V0 | blanc | 1 | · | F |
| V1 | 0 | blanc | ← | E |
| V1 | 1 | blanc | ← | R |
| V1 | blanc | 1 | · | F |
| R | 0 | 0 | ← | R |
| R | 1 | 1 | ← | R |
| R | blanc | blanc | → | P |
| E | 0 | blanc | ← | E |
| E | 1 | blanc | ← | E |
| E | blanc | 0 | · | F |

L'état initial est P. L'état terminal est F.

On considère qu'on initialise la machine en plaçant la tête sur le bit de poids fort.

Que fait-elle ?

On pourra dans un premier temps se demander ce qu'effectue la machine dans chacun de ses états.