Calculabilité-Complexité Cours1

Hadjila Fethallah

Maître de Conférences au Département d'Informatique

F_hadjila@mail.univ-tlemcen.dz

Introduction

- Y a-t-il des limites à ce que nous pouvons faire avec un ordinateur ?
- des limites à ce que nous pouvons calculer (dans l'absolu) ?
- Quelle sont les ressources requises pour une solution algorithmique (temps processeur, espace mémoire)

introduction

- **Problème de décision**

C’est une fonction acceptant des arguments (codés avec des entiers) et fournissant un résultat binaire {oui ,non}

- **Notion de procedure effective**

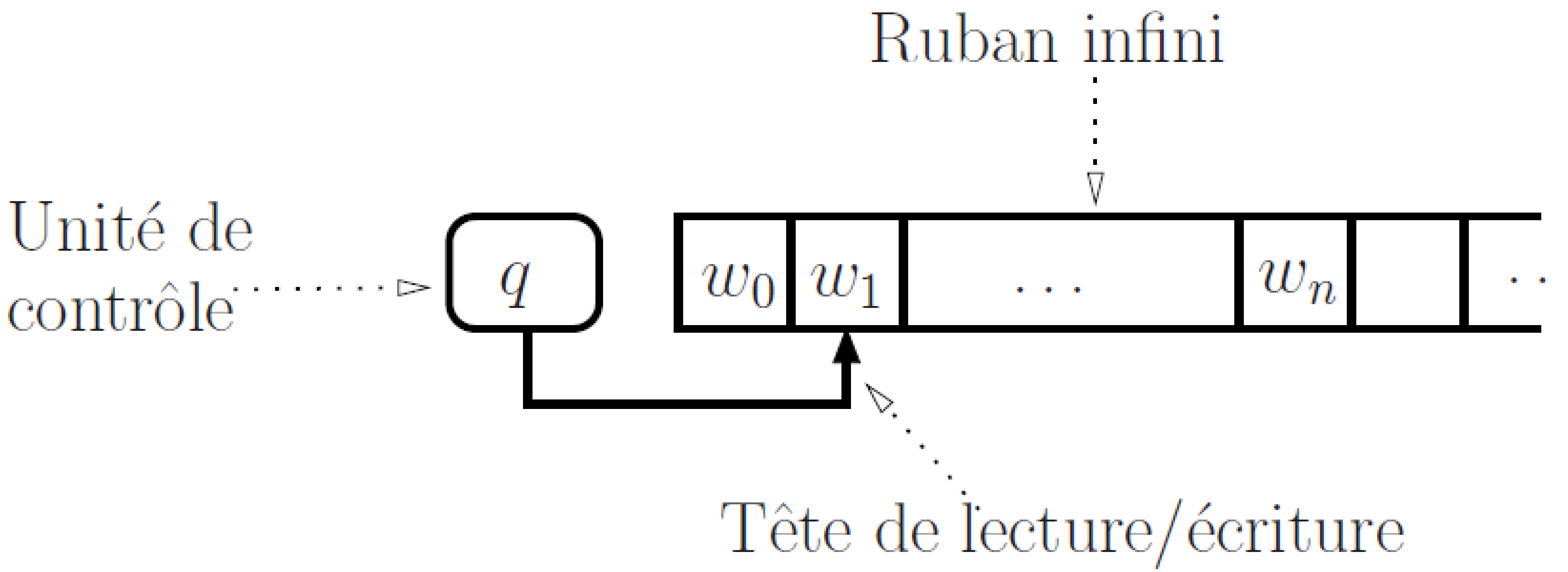
C’est une suite finie d’instructions, qui permet pour chaque entrée x, d'obtenir son image en un nombre fini d'étapes.

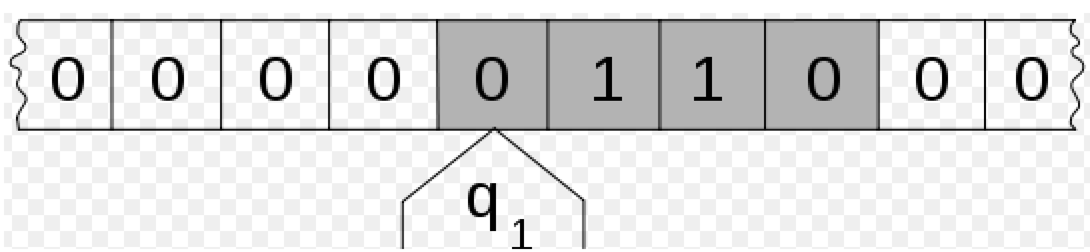
Tentatives de formalisation

Définition formelle de la notion d'algorithme:

- **Turing**: Machine de Turing (1936) (UK)
- **Post**: Machine de Post (USA)
- **Godel:** fonctions récursives(GR)
- **Church**: λ-calcul(USA)
- **Markov**: processus de markov(RUS)

Machine de Turing





Machine de turing

La MT est 7-tuplet $(Q,\Sigma,\Gamma,\delta,q0,B,F)$, avec:

I. Q : ensemble fini d'états.
II. Γ : l'alphabet du ruban,
III. $\Sigma \subseteq \Gamma$ est l'alphabet d'entrée
IV. $B \in \Gamma - \Sigma$ est le symbole blanc
V. $q0 \in Q$ est l'etat initial
VI. $F \subseteq Q$ l'ensemble des etats finaux
VII. $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$ est une fonction de transition.

la valeur de δ (q,X) est soit indefinie ou un triplet (nouvel etat, symbole de replacement, direction (L/R) de la tête).

Hypothèses implicites

- La sequence d'entrée est contigue dans le ruban
- toutes les cellulues restantes contiennent le symbole blanc 'B'
- la tête est positionnée sur le 1er symbole de la chaine d'entrée.
- Il y a un seul état initial
- Les machines de turing indetreministes, à plusieurs ruban, ayant 02 cotés infinis sont equivalentes

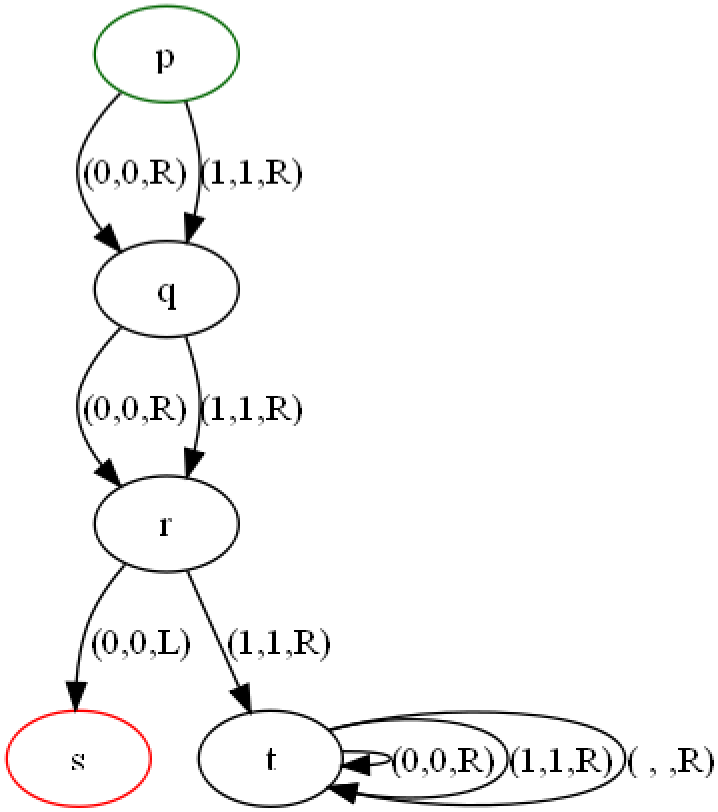
Machine de Turing

- Objectifs:
  - Reconnaissance de langages
  - Calcul de fonctions
  - Etude de complexité des problèmes/algorithmes

exemple

- On cherche une MT qui verifie que le 3eme symbole (d’un chaine binaire) est un 0, dans cas, on accepte la chaine, sinon, on fait une boucle infinie.
- $M=(\{p,q,r,s,t\},\{0,1,\},\{0,1,B\}, \delta,p,B,\{s\})$
- $\delta(p,X) = (q,X,R)$ pour $X=0,1$
- $\delta(q,X) = (r,X,R)$ pour $X=0,1$
- $\delta(r,0) = (s,0,L)$
- $\delta(r,1) = (t,1,R)$
- $\delta(t,X) = (t,X,R)$ pour $X=0,1,B$

exemple

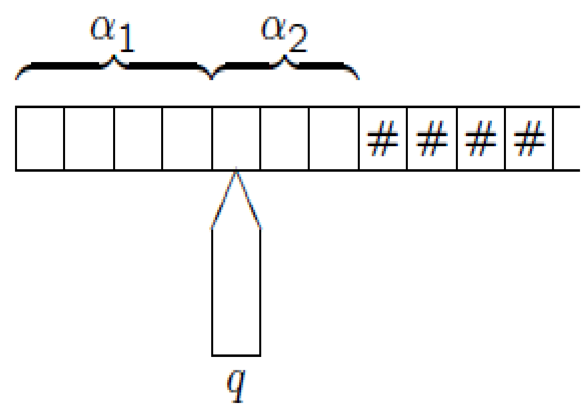


Configuration d’une MT

- configuration : triplet contenant:

1. L’état de la machine,
2. le sous mot à gauche de la tête de lecture
3. le sous mot à droite de la tête de lecture, y compris la position de la tête.

Configurations (q, α_1, α_2)



Problèmes indécidables

- Certains problèmes n'admettent aucun algorithme.
- ex. Problème d'arrêt
- ex. vérifier si un polynôme avec des coefficients entiers possède des racines entières
- ex. vérifier si une formule de la logique des prédicats est valide ou non.
- La majorité des problèmes n’est pas décidable

These de turing-church

- Tout problème possédant une solution algorithmique, peut être calculé sur un machine de turing
- Tout ce qui ne peut pas être calculable sur une machine de turing n'a aucun algorithme.

Thèse de Church-Turing

- La classe des problèmes décidables coïncide avec celle des fonctions implémentables sur machine de turing/ partiellement récursives

Notion intuitive d'algorithme

=

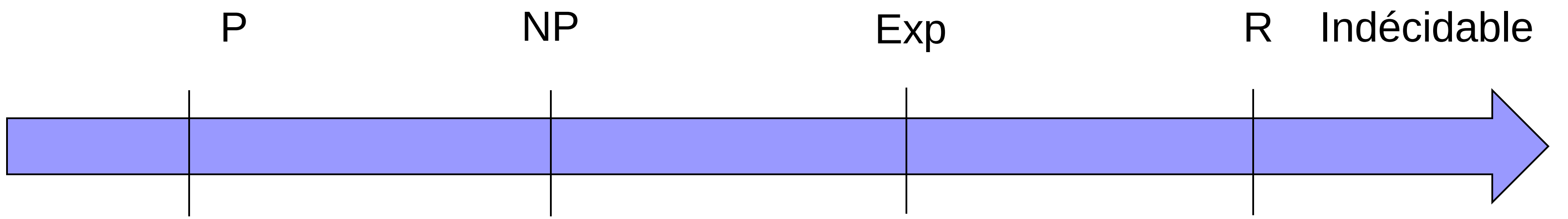
machine de Turing

=

Fonction partiellement recursive

Complexité des problèmes

- Augmentation de la difficulté



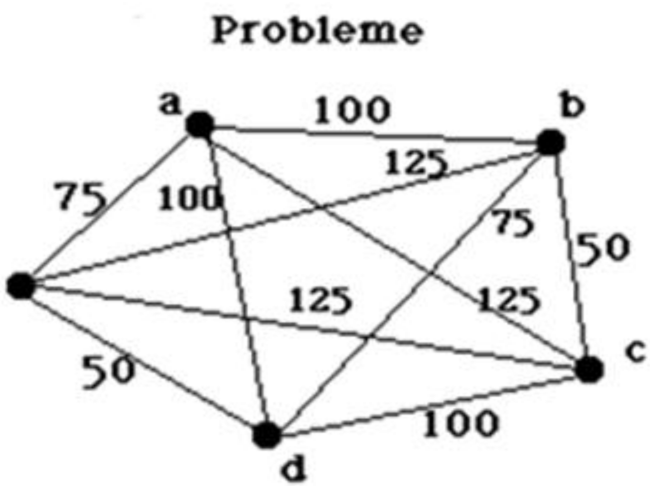
Problèmes polynomiaux (P)

- Problème résolu en faisant au maximum n^k étapes (instructions), avec k une constante
- Exemples:
- Trier un tableau d'entiers de taille n
- La recherche du plus court chemin entre deux points (nœuds) d'un graphe
- Produit scalaire….

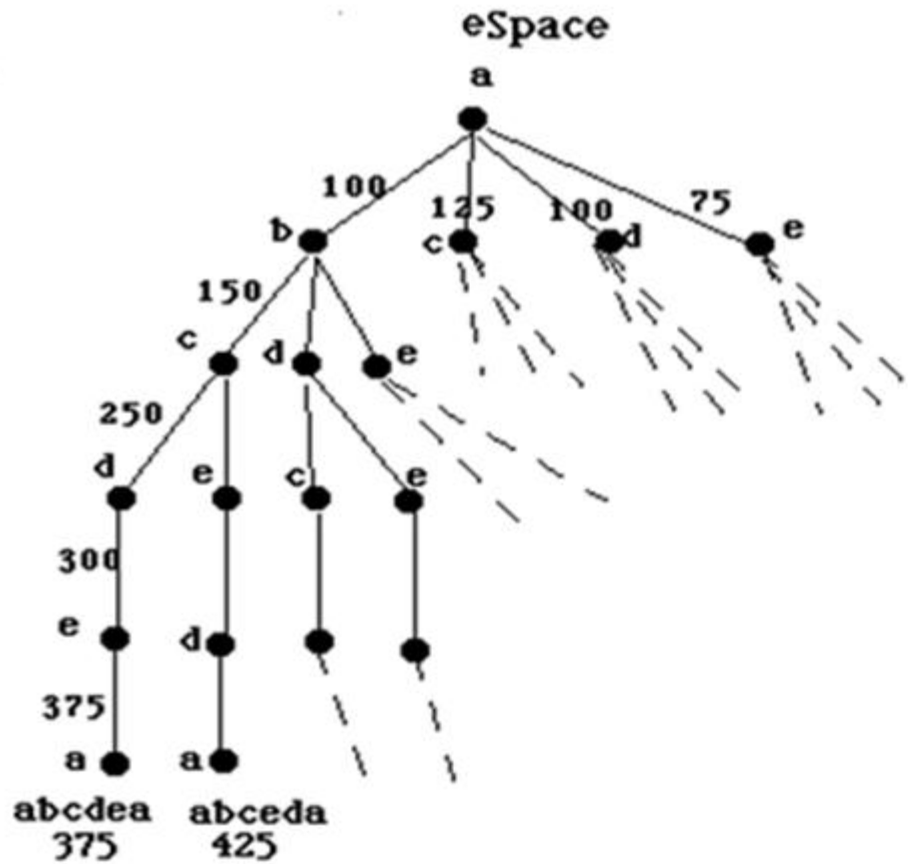
Problèmes Non déterministes polynomiaux (NP)

- Un problème p' $\in$NP si la vérification d'une solution potentielle nécessite un temps polynomial
- Exemples:
- 3 SAT, 4 SAT,…
- La 2-partition, la 3-partition
- Probleme de voyageur de commerce
- Sudoku, Knapsack (sac à dos)
- Factorisation d'un nombre entier

Voyageur de commerce/ Sudoku



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



P vs NP

- Est-ce qu’on peut résoudre un problème de recherche sans faire la recherche ?
- La majorité des chercheurs pense que les 02 classes sont différentes

Problèmes exponentiels (EXP)

- Problème résolu en faisant au maximum 2^{n^k} étapes (instructions), avec k une constante
- Exemple:
- Jeu d'échec : Décider si le joueur A réussit ou non une épreuve d'échec, en partant d'une configuration particulière de l'échiquier

Problèmes R

- Problème résolu en faisant un nombre fini d'Operations (finite execution time problem)
- R dénote tous les problèmes calculables (partiellement récursives)

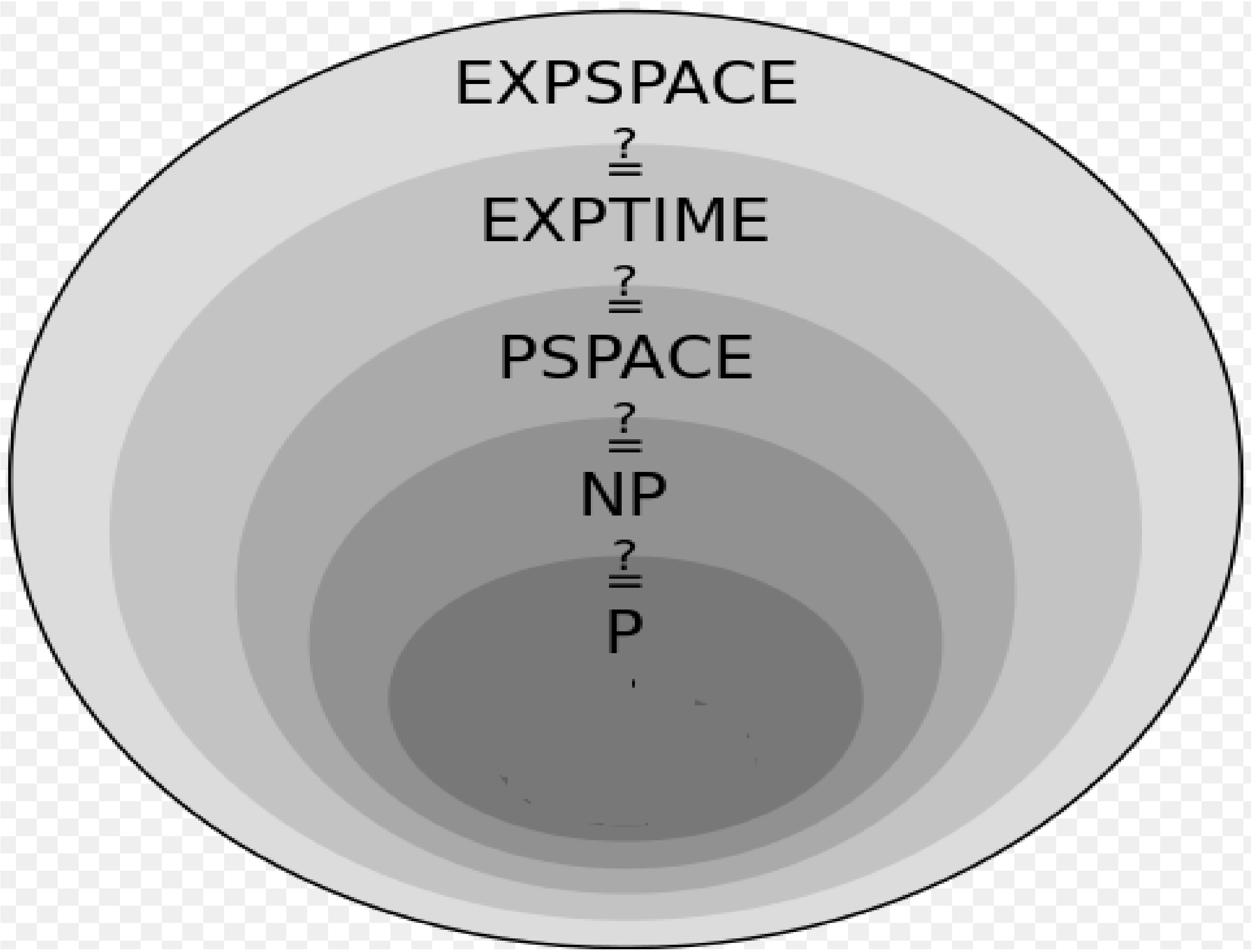
Notion de réduction

- Objectif : réutiliser une solution algorithmique associée à un problème B, et l'adapter pour un autre problème A
- *notation: A ≤ B*
- formellement, une **réduction** est un algorithme 'f' transformant un problème en un autre
- L'algorithme (ou la fonction) f, initialise les entrées de B avec ceux de A (ou avec des astuces), on s'assure aussi que, si f(x) est une solution de B alors x est une solution de A, et inversement

exemples

- Réduire un problème de recherche de plus court chemin d'un graphe non pondéré, en un probleme de recherche de plus court chemin d'un graphe pondéré
- Réduire un problème de recherche de plus long chemin d'un graphe pondéré, en un problème de recherche de plus court chemin d'un graphe pondéré
- Réduire un problème de recherche de chemin (d'un graphe)maximisant le produit de ses pondérations, en un problème de recherche de plus court chemin d'un graphe pondéré.

Récapitulation



FIN