INFO-F-302, Cours d'Informatique Fondamentale

Emmanuel Filiot
Département d'Informatique
Faculté des Sciences
Université Libre de Bruxelles

Année académique 2020-2021

Informatique Fondamentale

Informatique fondamentale et modélisation

L'informatique fondamentale est une branche de l'informatique dont le principal but est la conception de **modèles** pour des problèmes, concepts ou applications informatiques, et le développement **d'algorithmes** pour analyser ces modèles.

Exemples de modèles

- graphes : modèles de réseaux – sociaux, routiers, web, etc. –, de dépendances (entre les variables d'un programme, entre les classes en programmation objet, etc.), ...

Exemples de modèles

- graphes : modèles de réseaux – sociaux, routiers, web, etc. –, de dépendances (entre les variables d'un programme, entre les classes en programmation objet, etc.), ...
- la logique : modélise les énoncés mathématiques, les propriétés d'un système informatique, d'un programme, etc.

Exemples de modèles

- graphes : modèles de réseaux – sociaux, routiers, web, etc. –, de dépendances (entre les variables d'un programme, entre les classes en programmation objet, etc.), ...
- la logique : modélise les énoncés mathématiques, les propriétés d'un système informatique, d'un programme, etc.
- les systèmes de déduction : modélisent le raisonnement mathématique via la notion de preuve

Exemples de modèles

- graphes : modèles de réseaux – sociaux, routiers, web, etc. –, de dépendances (entre les variables d'un programme, entre les classes en programmation objet, etc.), ...
- la logique : modélise les énoncés mathématiques, les propriétés d'un système informatique, d'un programme, etc.
- les systèmes de déduction : modélisent le raisonnement mathématique via la notion de preuve
- les machines de Turing : modèle d'ordinateur

Exemples de modèles

- graphes : modèles de réseaux – sociaux, routiers, web, etc. –, de dépendances (entre les variables d'un programme, entre les classes en programmation objet, etc.), ...
- la logique : modélise les énoncés mathématiques, les propriétés d'un système informatique, d'un programme, etc.
- les systèmes de déduction : modélisent le raisonnement mathématique via la notion de preuve
- les machines de Turing : modèle d'ordinateur
- les automates : modélisent des programmes informatiques "simples"

Exemples de modèles

- graphes : modèles de réseaux – sociaux, routiers, web, etc. –, de dépendances (entre les variables d'un programme, entre les classes en programmation objet, etc.), ...
- la logique : modélise les énoncés mathématiques, les propriétés d'un système informatique, d'un programme, etc.
- les systèmes de déduction : modélisent le raisonnement mathématique via la notion de preuve
- les machines de Turing : modèle d'ordinateur
- les automates : modélisent des programmes informatiques "simples"
- chaînes de Markov : modélisent des systèmes probabilistes

Exemples de modèles

- graphes : modèles de réseaux – sociaux, routiers, web, etc. –, de dépendances (entre les variables d'un programme, entre les classes en programmation objet, etc.), ...
- la logique : modélise les énoncés mathématiques, les propriétés d'un système informatique, d'un programme, etc.
- les systèmes de déduction : modélisent le raisonnement mathématique via la notion de preuve
- les machines de Turing : modèle d'ordinateur
- les automates : modélisent des programmes informatiques "simples"
- chaînes de Markov : modélisent des systèmes probabilistes
- etc.

Nous verrons de nombreux exemples dans le cours.

Réductions entre problèmes

- une notion essentielle en informatique fondamentale
- modélisation d'un problème par un autre

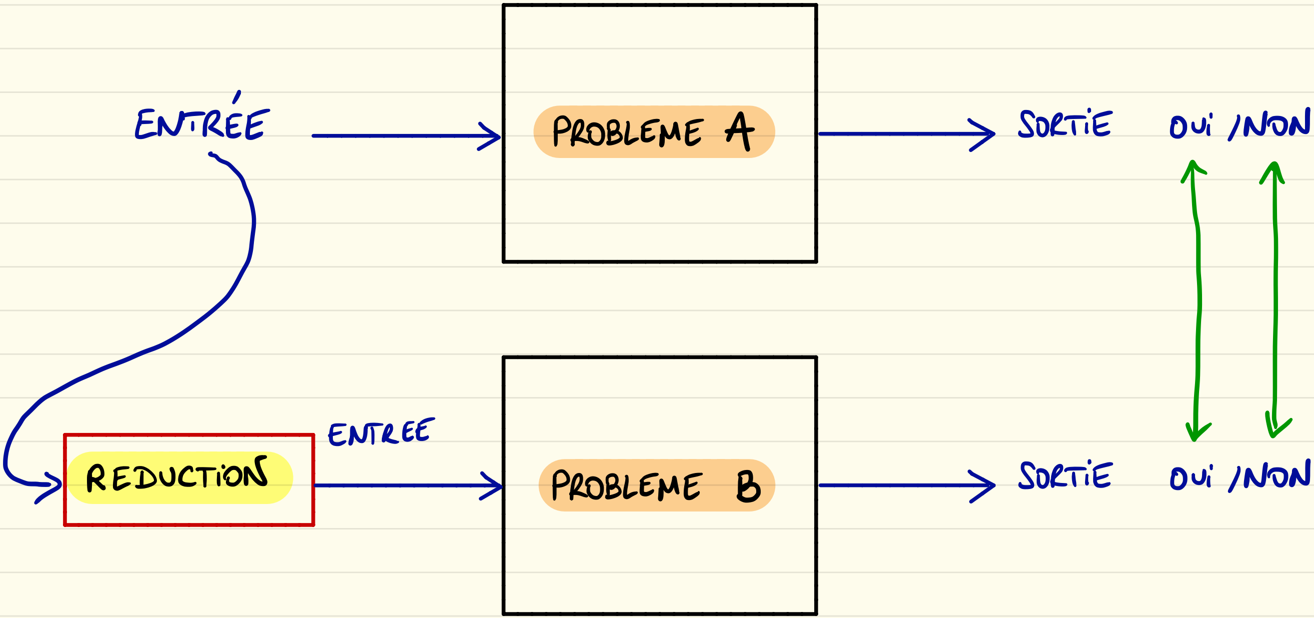
Pour les problèmes de décision (réponse oui/non)

Réduire un problème A vers un problème B, c'est trouver une méthode permettant d'encoder toute entrée I_A du problème A comme une entrée I_B du problème B, telle que

I_A a une solution (la réponse est oui) si et seulement si I_B a une solution

- parfois on demandera que cette méthode soit un algorithme
- pour les problèmes qui ne sont pas des problèmes de décision (trier un tableau par exemple), on demandera que toute solution de I_A s'encode en une solution de I_B et réciproquement toute solution de I_B se décode en une solution de I_A

REDUCTION DE A VERS B



Exemple – Problème B : Bin Packing

Problème Bin Packing (B) :

- **INPUT :** n objets de poids $p_1, \ldots, p_n$, k sacs de capacité C
- **OUTPUT :** oui ssi on peut ranger tous les objets dans au plus k sacs sans dépasser la capacité de chaque sac

Exemple – Problème B : Bin Packing

Problème Bin Packing (B) :

- **INPUT :** n objets de poids $p_1, \ldots, p_n$, k sacs de capacité C
- **OUTPUT :** oui ssi on peut ranger tous les objets dans au plus k sacs sans dépasser la capacité de chaque sac

On veut ranger les objets suivants dans des sacs de capacité 10kg.

Objets	1	2	3	4	5	6	7
Poids	3	4	4	3	3	2	1

Peut-on réussir avec 3 sacs ?

Exemple – Problème B : Bin Packing

Problème Bin Packing (B) :

- **INPUT :** n objets de poids $p_1, \ldots, p_n$, k sacs de capacité C
- **OUTPUT :** oui ssi on peut ranger tous les objets dans au plus k sacs sans dépasser la capacité de chaque sac

On veut ranger les objets suivants dans des sacs de capacité 10kg.

Objets	1	2	3	4	5	6	7
Poids	3	4	4	3	3	2	1

Peut-on réussir avec 3 sacs ? oui, on forme les sacs d'objets suivants : $\{1, 2\}, \{3, 4, 5\}, \{6, 7\}$.

Exemple – Problème B : Bin Packing

Problème Bin Packing (B) :

- **INPUT :** n objets de poids $p_1, \ldots, p_n$, k sacs de capacité C
- **OUTPUT :** oui ssi on peut ranger tous les objets dans au plus k sacs sans dépasser la capacité de chaque sac

On veut ranger les objets suivants dans des sacs de capacité 10kg.

Objets	1	2	3	4	5	6	7
Poids	3	4	4	3	3	2	1

Peut-on réussir avec 3 sacs ? oui, on forme les sacs d'objets suivants : $\{1,2\}, \{3,4,5\}, \{6,7\}$.
et avec 2 sacs ?

Exemple – Problème B : Bin Packing

Problème Bin Packing (B) :

- **INPUT :** n objets de poids $p_1, \ldots, p_n$, k sacs de capacité C
- **OUTPUT :** oui ssi on peut ranger tous les objets dans au plus k sacs sans dépasser la capacité de chaque sac

On veut ranger les objets suivants dans des sacs de capacité 10kg.

Objets	1	2	3	4	5	6	7
Poids	3	4	4	3	3	2	1

Peut-on réussir avec 3 sacs ? oui, on forme les sacs d'objets suivants : $\{1,2\}, \{3,4,5\}, \{6,7\}$.
et avec 2 sacs ?
oui : $\{1,2,4\}, \{3,5,6,7\}$. Cette solution est optimale (on ne peut pas réussir avec un seul sac puisque la somme totale de tous les poids dépasse 10).

Problème A : 2-partition

Problème 2-partition (A) :

- **INPUT :** m entiers naturels $c_1, \ldots, c_m$ tels que $S := \sum_{i=1}^{m} c_i$ est paire,
- **OUTPUT :** oui ssi il existe $J \subseteq \{1, \ldots, m\}$ tel que $\sum_{i \in J} c_i = \sum_{i \notin J} c_i$.

Problème A : 2-partition

Problème 2-partition (A) :

- **INPUT :** m entiers naturels $c_1, \ldots, c_m$ tels que $S := \sum_{i=1}^{m} c_i$ est paire,
- **OUTPUT :** oui ssi il existe $J \subseteq \{1, \ldots, m\}$ tel que $\sum_{i \in J} c_i = \sum_{i \notin J} c_i$.

Exemple avec $1, 4, 2, 3, 2$: $1 + 2 + 3 = 2 + 4$.

Problème A : 2-partition

Problème 2-partition (A) :

- **INPUT :** m entiers naturels $c_1, \ldots, c_m$ tels que $S := \sum_{i=1}^{m} c_i$ est paire,
- **OUTPUT :** oui ssi il existe $J \subseteq \{1, \ldots, m\}$ tel que $\sum_{i \in J} c_i = \sum_{i \notin J} c_i$.

Exemple avec $1, 4, 2, 3, 2$: $1 + 2 + 3 = 2 + 4$.

Réduction vers BinPacking ?

Problème A : 2-partition

Problème 2-partition (A) :

- **INPUT :** m entiers naturels $c_1, \ldots, c_m$ tels que $S := \sum_{i=1}^{m} c_i$ est paire,
- **OUTPUT :** oui ssi il existe $J \subseteq \{1, \ldots, m\}$ tel que $\sum_{i \in J} c_i = \sum_{i \notin J} c_i$.

Exemple avec $1, 4, 2, 3, 2$: $1 + 2 + 3 = 2 + 4$.
Réduction vers BinPacking ? On prend $C = S/2$ et les objets $1, \ldots, m$ de poids $c_1, \ldots, c_m$, et $k = 2$. Il y a une solution à toute instance de 2-partition si et seulement si il y a en une à l'instance de BinPacking ainsi construite.

7 Objectifs du cours

- savoir modéliser des problèmes de manière précise et rigoureuse, en particulier en utilisant des langages logiques
- maîtriser la notion de réduction entre problèmes
- acquérir des notions de base sur les concepts et modèles fondamentaux de l'informatique

8 Plan

1. ~~rappels de concepts mathématiques importants en informatique~~
2. modèles pour les énoncés mathématiques et le raisonnement : la logique et la déduction naturelle
3. modélisation de problèmes en logique propositionnelle : les solveurs SAT
4. introduction aux notions de complexité de problèmes et de réduction entre problèmes
5. introduction à la calculabilité
6. modèles de programmes : les automates finis
7. s'il reste du temps : preuves de programmes, machines de Turing ...

9 Organisation pratique du cours

Annonces Par email et reprises sur l'université virtuelle

Matériel Les slides constituent le syllabus. Ils seront disponibles sur l'UV.

Epreuve intermédiaire Adrien Boiret. **Pas de seconde session !**

Travaux pratiques Assistants : Adrien Boiret, Emmanuel Filiot

Evaluation Examen écrit (3/4) et Epreuve intermédiaire (1/4)

Contact

- **email** : `efiliot@ulb.ac.be`