Les présentations

En gros mes recherches

Professeur au laboratoire Informatique Signal et Image de la Côte d'Opale (LISIC)

> Conception et étude d'algorithmes d'optimisation
> (inspirés de la biologie)

- **Algorithmes** : méthode univoque ("recette de cuisine")
- **Optimisation** : trouver les meilleures solutions possibles à un problème (transport, emploi du temps, design, réglage de prothèses, modélisation cognitive,...)
- **Bio-inspiré** : extraire les principes actifs de système biologique : théorie de l'évolution, fouragement des fourmis, déplacement d'oiseaux, etc. (systèmes complexes)
- **Conception** : créer et tester de nouveaux algorithmes
- **Etude** : comprendre et prédire pourquoi cela marche, ou mieux, pourquoi ça rate.

Aspects théoriques de l'Informatique (ATI)

- Fondement de l'informatique, bases avant compilation

- Volume :
  9h CM et 15h TD
- Evaluation :
  20 % Interrogation 1 (30min),
  30 % Interrogation 2 (30min),
  50 % Examen final le 11/01/2018 (2h)

Web page de l'enseignement (supports et informations) :
`http://www-lisic.univ-littoral.fr/~verel`

Les systèmes formels

Utilisations pratiques

- Spécification des langages de programmation,
- Compilation, analyseur syntaxique,
- Recherche de motifs (texte, BD, web, etc.),
- les TALN
- Systèmes d'exploitation,
- Compression de texte,
- Cryptographie,
- Automatique,
- etc.

Aspects théoriques de l'Informatique

Questions fondamentales – et pratiques !

- **Spécification de programmes** :
  Sémantique d'un programme
- **Vérification de programmes** :
  Correction du programme
- **Complexité** :
  Temps, espace pour exécuter le calcul
- **Calculabilité** :
  Que peut-on calculer à l'aide d'un algorithme ?
  Quels problèmes peut-on résoudre avec un ordinateur ?

Aspects théoriques de l'Informatique

Questions fondamentales – et pratiques !

- **Spécification de programmes** :
  Sémantique d'un programme
- **Vérification de programmes** :
  Correction du programme
- **Complexité** :
  Temps, espace pour exécuter le calcul
- **Calculabilité** :
  Que peut-on calculer à l'aide d'un algorithme ?
  Quels problèmes peut-on résoudre avec un ordinateur ?

- Mot : réponse possible à un problème
- Langage : ensemble des réponses positives au problème
- Modèle de calcul : programme qui "calcule" le langage
  automate, machine de Turing, automate cellulaire,
  lambda-calcul, fonction récursive, etc.

Dénombrabilité, mot et langage

Aspects Théoriques de l'Informatique (ATI)
Licence 3 informatique

Sébastien Verel
verel@lisic.univ-littoral.fr
http://www-lisic.univ-littoral.fr/~verel

Université du Littoral Côte d'Opale
Laboratoire LISIC
Equipe OSMOSE

Objectifs de la séance 01

- Savoir définir mot et langage.
- Savoir écrire une expression régulière simple
- Savoir définir un ensemble dénombrable
- Savoir définir une bijection entre deux ensembles dénombrables
- Savoir montrer qu'un ensemble est non dénombrable
- Connaitre le cardinal de l'ensemble des partis d'un ensemble fini

Objectifs de la séance 01

- Savoir définir mot et langage.
- Savoir écrire une expression régulière simple
- Savoir définir un ensemble dénombrable
- Savoir définir une bijection entre deux ensembles dénombrables
- Savoir montrer qu'un ensemble est non dénombrable
- Connaitre le cardinal de l'ensemble des partis d'un ensemble fini

Question principale du jour :

Quelle langue parlons-nous ?

Références

Sources principales

- Denis Robilliard, Université du Littoral Côte d'Opale, `http://www-lisic.univ-littoral.fr/~robillia`
- Sandrine Julia, Université Nice Sophia Antipolis, `deptinfo.unice.fr/~julia/IT/`

Mots clés

langage rationnel, automate, automate fini déterministe, Kleene, expression régulière, détermination, lemme de l'étoile, machine de Turing, etc.

Plan

Quelques exemples de mots...

- ulysse
- toison
- heureux
- beau
- celui-là
- voyage

Quelques exemples de mots...

- ulysse
- toison
- heureux
- beau
- celui-là
- voyage

Joachim DU BELLAY (1522-1560)

Quelques exemples de mots...

Ou encore :

- 0605547781
- 0492942724
- 0675389509
- 0492946666

Quelques exemples de mots...

Ou encore :

- 0605547781
- 0492942724
- 0675389509
- 0492946666

Ou encore :

- as1sce
- as2ce
- as3scel

Définition de mot

Alphabet

Un **alphabet** Σ est un ensemble *dénombrable*.

Les éléments de Σ sont appelés les **lettres** ou **symboles** de l'alphabet.

Remarque : très souvent un alphabet sera même un ensemble fini.

Au fait

Les notions de fini et dénombrable
interviennent beaucoup dans ce cours,
et plus généralement en informatique

Au fait

Les notions de fini et dénombrable
interviennent beaucoup dans ce cours,
et plus généralement en informatique

Petit crochet pour l'étudiant
Grand crochet pour l'informatique
(mais pas trop grand)

Petite réflexion venue de loin

2 enclos contigus enferment beaucoup de moutons de deux bergers.

Comment savoir s'ils ont le même nombre de moutons ?

Précisions : les moutons peuvent bouger énormément mais il est possible de les faire passer par une porte...

Petite réflexion venue de loin

2 enclos contigus enferment beaucoup de moutons de deux bergers.

Comment savoir s'ils ont le même nombre de moutons ?

Précisions : les moutons peuvent bouger énormément mais il est possible de les faire passer par une porte...

Cardinalité

Comparer les ensembles au moyen des bijections (et injections)

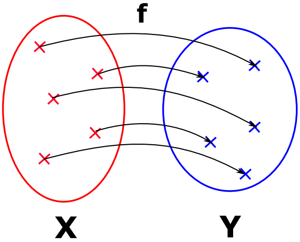
Cardinalité

Bijection (correspondance 1 – 1)

$f : X \to Y$ est une fonction **bijective** ssi :

- **injective :**
  pour tout $x, x' \in X$ tel que $f(x) = f(x')$ alors $x = x'$
- **surjective :**
  pour tout $y \in Y$, il existe $x \in X$ tel que $y = f(x)$

$\Leftrightarrow$ pour tout $y \in Y$, il existe un unique $x \in X$ tel que $y = f(x)$



Autant de moutons de chaque côté, autant de "trucs" de chaque côté, réécriture/codage de Y par X, clé unique, ...

Cardinalité

Comparer les ensembles au moyen des bijections (et injections)

Equipotent

Les ensembles X et Y ont le **même cardinal** (équipotent) s'il existe une bijection de X dans Y

"A chaque fois que X présente un nouvel élément, Y présente un nouvel élément, et réciproquement."

Subpotent

Le cardinal de X est plus petit ou égale au cardinal de Y (subpotent) s'il existe une injection de X dans Y

"A chaque fois que X présente un nouvel élément, Y présente un nouvel élément."

Exercices fiche TD 01.

Ensemble dénombrable

Ensemble fini

Un ensemble X est fini ssi il existe un entier naturel n et une bijection de l'ensemble des entiers compris entre 0 et $n-1$ dans X. Le nombre d'éléments (cardinal) de X est alors n.

Remarque : l'ensemble vide $\emptyset$ est fini.

Ensemble dénombrable

Un ensemble dénombrable est soit un ensemble fini, soit un ensemble en bijection avec l'ensemble des entiers naturel $\mathbb{N}$.

Remarque : $\mathbb{N}$ est un ensemble dénombrable.

Ensemble dénombrable

Ensemble fini

Un ensemble X est fini ssi il existe un entier naturel n et une bijection de l'ensemble des entiers compris entre 0 et $n-1$ dans X. Le nombre d'éléments (cardinal) de X est alors n.

Remarque : l'ensemble vide $\emptyset$ est fini.

Ensemble dénombrable

Un ensemble dénombrable est soit un ensemble fini, soit un ensemble en bijection avec l'ensemble des entiers naturel $\mathbb{N}$.

Remarque : $\mathbb{N}$ est un ensemble dénombrable.

Remarque intuitive

Un ensemble est dénombrable lorsqu'on peut numéroter ses éléments avec les entiers (cf. exercices fiche TD 01) – définit la bijection.

Exemple fondamental : cardinalité de $[0, 1]$

Théorème

L'ensemble des nombres réels entre 0 et 1 n'est pas dénombrable.

Conséquence : on ne peut pas coder l'ensemble des nombres réels de $[0, 1]$ avec un ordinateur !

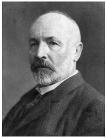
Exemple fondamental : cardinalité de $[0, 1]$

Théorème

L'ensemble des nombres réels entre 0 et 1 n'est pas dénombrable.

Conséquence : on ne peut pas coder l'ensemble des nombres réels de $[0, 1]$ avec un ordinateur !

Preuve : Procédé diagonal de Cantor, voir au tableau.



Georg Cantor, mathématicien allemand, 1845 - 1918.
Théorie des ensembles
Ensemble bien ordonné
"infinité d'infinis"

Fonctions non programmables

Théorème

L'ensemble des fonctions de $\mathbb{N}$ dans $\mathbb{N}$ n'est pas dénombrable.

Fonctions non programmables

Théorème

L'ensemble des fonctions de $\mathbb{N}$ dans $\mathbb{N}$ n'est pas dénombrable.

Conséquence : il existe des fonctions que l'on ne peut pas programmer !
(pourquoi au fait ? cf. TD 01)

Fonctions non programmables

Théorème

L'ensemble des fonctions de $\mathbb{N}$ dans $\mathbb{N}$ n'est pas dénombrable.

Conséquence : il existe des fonctions que l'on ne peut pas programmer !
(pourquoi au fait ? cf. TD 01)

Preuve : Procédé diagonal de Cantor, voir au tableau.

Définition de mot

Alphabet

Un **alphabet** Σ est un ensemble *dénombrable*.

Les éléments de Σ sont appelés les **lettres** ou **symboles** de l'alphabet.

Remarque : très souvent un alphabet sera même un ensemble fini.

Définition de mot

Mot

On appelle **mot** sur Σ toute suite *finie* de Σ :

$$u = (a_1, a_2, \ldots, a_n)$$

où $n \geq 0$, et pour tout $i \in [1..n]$, $a_i \in \Sigma$.
Lorsque $n = 0$, le mot est appelé le **mot vide**, noté ϵ.

Exemples et notation

Notation

Les mots sont notés sans parenthèse, ni virgule.

Par exemple, avec l'alphabet $\Sigma = \{t, a, o, i\}$

$$u = (t, a, t, o)$$

est noté :

$$u = tato.$$

u est bien une suite finie où $u_1 = t$, $u_2 = a$, $u_3 = t$ et $u_4 = o$.

Exemples et notation

Notation

Les mots sont notés sans parenthèse, ni virgule.

Par exemple, avec l'alphabet $\Sigma = \{t, a, o, i\}$

$$u = (t, a, t, o)$$

est noté :

$$u = tato.$$

u est bien une suite finie où $u_1 = t$, $u_2 = a$, $u_3 = t$ et $u_4 = o$.

Quelques exemples supplémentaires

$aota$, $toto$, $taitoi$, $ttaaittooti$, $ootitaota$, ϵ, ...

Longueur et occurrence

Ensemble des mots

L'ensemble des mots sur Σ est noté Σ^*.

Longueur et occurrence

Ensemble des mots

L'ensemble des mots sur Σ est noté Σ^*.

Longueur

La longueur du mot $u = a_1 a_2 \dots a_n$ est l'unique entier n correspondant au cardinal du domaine de la suite u.
La longueur du mot u est notée $|u|$.

Longueur et occurrence

Ensemble des mots

L'ensemble des mots sur Σ est noté Σ^*.

Longueur

La longueur du mot $u = a_1 a_2 \dots a_n$ est l'unique entier n correspondant au cardinal du domaine de la suite u.
La longueur du mot u est notée $|u|$.

Nombre d'occurrences

Le **nombre d'occurrences** d'une lettre $\alpha \in \Sigma$ dans un mot u sur Σ se note $|u|_\alpha$.

Applications

Les mots interviennent dans beaucoups de domaines :

- langages naturelles (linguistique, TALN)
- langages artificiels (programmation)
- compression de texte
- protocoles d'échange
- biologie : génome
- cryptographie
- machines automatiques
- téléphone : numéro, tweet,...
- ...

Produit de concaténation

Produit de concaténation

Soient $u = a_1 a_2 \dots a_n$ et $v = b_1 b_2 \dots b_p$ deux mots sur Σ.
La **concaténation** de u et v est le mot :

$$a_1 a_2 \dots a_n b_1 b_2 \dots b_p$$

On note le concaténé de u et v par uv.

Propriétés algébriques

Produit de concaténation

- $|uv| = |u| + |v|$
- Le produit de concaténation est associatif :
$$u(vw) = (uv)w$$
l'ordre est indifférent, donc on note sans parenthèse, $(uv)w = uvw$
- Le mot vide ϵ est neutre :
$$\epsilon u = u\epsilon = u$$
- Le produit de concaténation n'est pas commutatif :
$$aaaabb \neq bbaaaa$$
- On note comme la répétition d'un mot u par :
$$u^n = uuu \dots u$$

Remarque : on dit que $(\Sigma^*, .)$ est un monoïde

Préfixe, suffixe et facteur

Préfixe, suffixe

Soient u, v et w des mots sur Σ tel que $w = uv$.
u est un **préfixe** et v est un **suffixe** de w.

Facteur

$u \in \Sigma^*$ est un **facteur** de $w \in \Sigma^*$ s'il existe v et $v^{'}$ tels que

$$w = v^{'}uv$$

Sous-mot

Soit $a_1, a_2, \ldots, a_n \in \Sigma$ et $u_0, u_1, \ldots u_n \in \Sigma^*$
Alors le mot $a_1a_2a_3\ldots a_n$ est un **sous-mot** du mot $u_0a_1u_1a_2u_3a_3\ldots a_nu_n$

Remarque : La relation de préfixe défini un ordre sur l'ensemble Σ^*

Exemples

Soit $\Sigma = \{p, o, a\}$ et $u = popoa$

- *oa* est un de u
- *po* est un de u
- *oo* est un de u
- *opo* est un de u

ϵ est un de tout les mots de Σ^*.

Exemples

Soit $\Sigma = \{p, o, a\}$ et $u = popoa$

- *oa* est un suffixe de *u*
- *po* est un préfixe et un facteur de *u*
- *oo* est un sous-mot de *u*
- *opo* est un facteur de *u*

ϵ est un de tout les mots de Σ^*.

Exemples

Soit $\Sigma = \{p, o, a\}$ et $u = popoa$

- oa est un suffixe de u
- po est un préfixe et un facteur de u
- oo est un sous-mot de u
- opo est un facteur de u

ϵ est un préfixe et un suffixe de tout les mots de Σ^*.

Ordre lexicographique

Supposons qu'il existe un ordre sur l'ensemble des lettres Σ noté $\leq$

Ordre lexicographique

L'ordre lexicographique sur Σ^*, noté $\prec$, est défini par :

- *base :*
  - $\epsilon \prec \epsilon$
  - pour tout $(a, a^{'}) \in \Sigma^2$ tel que $a \leq a^{'}$, $a \prec a^{'}$.
- *induction :*
  - si $u \prec v$ et $|u| < |v|$ alors $\forall a \in \Sigma$, $ua \prec va$ et $u \prec va$
  - si $u \prec v$ et $|v| \leq |u|$ alors $\forall a \in \Sigma$, $ua \prec v$

Cardinalité de l'ensemble des mots

Cardinalité de l'ensemble des mots

L'ensemble des mots sur un alphabet Σ est dénombrable.

Cardinalité de l'ensemble des mots

Cardinalité de l'ensemble des mots

L'ensemble des mots sur un alphabet Σ est dénombrable.

Démonstration :
Pour démontrer qu'un ensemble est dénombrable, il suffit de démontrer une bijection avec l'ensemble des entiers naturels $\mathbb{N}$ (numérotation des mots).

- On trie les mots de Σ^* par ordre lexicographique
- On peut alors numéroter chaque mots par un entier naturel.

$$\Sigma^* = (u_0, u_1, u_2, \ldots)$$

- Ce qui prouve l'existence d'une bijection entre $\mathbb{N}$ et Σ^*. C.Q.F.D.

cf. TD 01 exo 3.d

Langage

Langage

Un langage est un sous-ensemble (une partie) de l'ensemble des mots Σ^*.

Remarque : tous les langages sont dénombrables.

Exemples basiques

Soit $\Sigma = \{p, l, d\}$

L'ensemble $\{pl, dl, pdpl, ldl\} \subset \Sigma^*$ est un langage.

Exemples de langage (1)

Ecriture des nombres (1)

$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ., -\}$
Langage des mots qui représentent des nombres à virgule flottante à 15 décimales.

Ecriture des nombres (2)

$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, ., -, e\}$
Langage des mots qui représentent des nombres à virgule flottante selon la norme IEEE 754.

JAVA

$\Sigma = \{a, \ldots, z, A, \ldots, Z, \ldots\}$
Langages des mots clés JAVA

Exemples de langage (2)

Sport

$\Sigma = \{\ldots\}$
Le langage est des scores au cours d'un jeu au tennis.

Automatique

$\Sigma = \{\ldots\}$
Le langage est des séquences possibles d'action d'une machine à café.

Web

$\Sigma = \{\ldots\}$
Le langage est des fichiers valides au format xml.

Expressions régulières (Rappel)

Rappel

- Les expressions régulières sont les expressions que l'on peut construire à partir de $+$, . et $*$ (cf. grep)

Conséquence

Une expression régulière définit un langage.

Par exemple

- $ab(a+b+c)$ est l'ensemble $\{aba, abb, abc\}$
- $(ab)^3$ est l'ensemble $\{ababab\}$
- $(ab)^*$: tous les mots qui répétent un nombre fini de fois ab (voir nul) $\{\epsilon, ab, abab, ababab, \ldots\}$
- $ab(a+b+c)^*$: tous les mots qui comment par ab