Probabilités, Statistiques, Combinatoire

Philippe Duchon

Université de Bordeaux – Licence Informatique

2018-2019

Administratif

- **Équipe enseignante :** G. Aupy, Ph. Duchon, L. Fredes-Carrasco, J. Herrmann, F. Kardos, A. Lentz, F. Mazoit, Th. Pierron, A. Strock
- Cours (vendredi après-midi), TD, TD Machine (16×1h20 TD, 5×1h20 TDM)
- Évaluation : 1 DS (semaine 12, 22 mars) ; 1 examen (coeff. 1/2 chacun ; coeff. 1 exam si meilleur)
- Documents en DS/Examen : une feuille A4 recto-verso, **manuscrite**, à votre nom.
- En session 2 : même chose (on ne refait pas le DS)

Équipe enseignante

- **Cours :** Ph. Duchon
- **TD :**
  - **INF A1** : Th. Pierron
  - **INF A2** : F. Mazoit
  - **INF A3** : G. Aupy (TDM A. Lentz)
  - **INF A4** : F. Kardos
  - **INF A5** : L. Fredes-Carrasco
  - **MI A1** : Ph. Duchon
  - **MI A2** : A. Strock (TDM F. Kardos)
- **TDM :** les mêmes, plus
  - **TM A1-A2** : Ph. Duchon
  - **TM A3-A4** : J. Herrmann
  - **TM A51 + CMI + MI** : Th. Pierron
  - **TM A52** : L. Fredes-Carrasco

Contenu de l'UE

Probabilités et **statistiques** : disciplines *mathématiques* censées étudier les situations d'*incertitude* ("le hasard", c'est quoi ?)

- **Probabilités :** on définit un modèle pour une expérience, et on *prédit* par le calcul ce qu'il est plus ou moins plausible d'observer

Contenu de l'UE

Probabilités et **statistiques** : disciplines *mathématiques* censées étudier les situations d'*incertitude* ("le hasard", c'est quoi ?)

- **Probabilités :** on définit un modèle pour une expérience, et on *prédit* par le calcul ce qu'il est plus ou moins plausible d'observer
- **Statistiques :** à partir de données concrètes, on essaie de *diagnostiquer*
  - décrire de manière concise les données observées
  - proposer des valeurs plausibles pour les paramètres d'un modèle
  - "est-ce que les données sont raisonnablement compatibles avec l'hypothèse que. . ."

Contenu de l'UE

Probabilités et **statistiques** : disciplines *mathématiques* censées étudier les situations d'*incertitude* ("le hasard", c'est quoi ?)

- **Probabilités :** on définit un modèle pour une expérience, et on *prédit* par le calcul ce qu'il est plus ou moins plausible d'observer
- **Statistiques :** à partir de données concrètes, on essaie de *diagnostiquer*
  - décrire de manière concise les données observées
  - proposer des valeurs plausibles pour les paramètres d'un modèle
  - "est-ce que les données sont raisonnablement compatibles avec l'hypothèse que. . ."
- **Combinatoire :** discipline (mathématique ? informatique ?) qui étudie (et *énumère*, compte. . .) les objets discrets (combinaisons, mots, chemins, arbres. . .)
  - parfois un préalable indispensable à certains calculs de probabilités
  - utilisée en particulier pour *analyser la complexité* (en moyenne, le plus souvent) des algorithmes

Pourquoi un cours de probas-stats en informatique ?

Probabilités et statistiques sont présentes dans de nombreux sous-domaines de l'informatique, aussi bien au niveau de la théorie que des applications. . . difficile de les éviter !

- **Conception d'algorithmes :** énormément d'algorithmes "probabilistes" (faisant appel au hasard dans la définition de l'algorithme) existent qui ont des performances bien meilleures que ce qu'atteignent des algorithmes "déterministes" d'un degré comparable de simplicité.

Pourquoi un cours de probas-stats en informatique ?

Probabilités et statistiques sont présentes dans de nombreux sous-domaines de l'informatique, aussi bien au niveau de la théorie que des applications. . . difficile de les éviter !

- **Conception d'algorithmes :** énormément d'algorithmes "probabilistes" (faisant appel au hasard dans la définition de l'algorithme) existent qui ont des performances bien meilleures que ce qu'atteignent des algorithmes "déterministes" d'un degré comparable de simplicité.
- **Analyse d'algorithmes :** les techniques probabilistes permettent de faire des analyses assez précises de la complexité d'algorithmes (**QuickSort** n'est pas juste un bon algorithme parce que, *en moyenne*, il est rapide ; on montre que *dans l'immense majorité des cas*, il est rapide)

Pourquoi un cours de probas-stats en informatique ?

Probabilités et statistiques sont présentes dans de nombreux sous-domaines de l'informatique, aussi bien au niveau de la théorie que des applications. . . difficile de les éviter !

- **Conception d'algorithmes :** énormément d'algorithmes "probabilistes" (faisant appel au hasard dans la définition de l'algorithme) existent qui ont des performances bien meilleures que ce qu'atteignent des algorithmes "déterministes" d'un degré comparable de simplicité.
- **Analyse d'algorithmes :** les techniques probabilistes permettent de faire des analyses assez précises de la complexité d'algorithmes (**QuickSort** n'est pas juste un bon algorithme parce que, *en moyenne*, il est rapide ; on montre que *dans l'immense majorité des cas*, il est rapide)
- **Analyse statistique des données :** classification, apprentissage. . . font beaucoup appel aux probabilités et aux statistiques

Pourquoi un cours de probas-stats en informatique ?

Probabilités et statistiques sont présentes dans de nombreux sous-domaines de l'informatique, aussi bien au niveau de la théorie que des applications... difficile de les éviter !

- **Conception d'algorithmes :** énormément d'algorithmes "probabilistes" (faisant appel au hasard dans la définition de l'algorithme) existent qui ont des performances bien meilleures que ce qu'atteignent des algorithmes "déterministes" d'un degré comparable de simplicité.
- **Analyse d'algorithmes :** les techniques probabilistes permettent de faire des analyses assez précises de la complexité d'algorithmes (**QuickSort** n'est pas juste un bon algorithme parce que, *en moyenne*, il est rapide ; on montre que *dans l'immense majorité des cas*, il est rapide)
- **Analyse statistique des données :** classification, apprentissage... font beaucoup appel aux probabilités et aux statistiques
- **PageRank** (Google) est au départ basé sur l'application de la théorie des chaînes de Markov (*i.e.* des probabilités)

Partie I : Combinatoire

Vocabulaire et notations : ensembles

Les notions et notations suivantes sont supposées connues :

- **description d'ensemble :** soit en listant ses éléments ($E = \{1, 2, 3, 4, 5, 6\}$), soit par une description des éléments (E est l'ensemble des entiers compris entre 1 et 6)
- **notation :** $x \in E$ ("x appartient à E" ; "E contient x") pour dire que x est un élément de l'ensemble E
- **notation :** $A \subset B$ ("A est inclus dans B", "B inclut A", ou "A est un sous-ensemble de B") pour dire que tout élément de A est aussi un élément de B (y compris si $A = B$)
- **description d'un sous-ensemble :** "ensemble des éléments de A qui satisfont la propriété P", $\{x \in A : P(x)\}$
- **ensembles particuliers :** $\emptyset$, $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$... (en notations françaises, $\mathbb{N}$ contient 0 ; ce n'est pas l'usage en anglais)

Vocabulaire et notations : opérations ensemblistes

- **opérations ensemblistes :** union, intersection, produit cartésien : $A \cup B$, $A \cap B$, $A \times B$; A^n.
- **différence :** $A - B = \{x \in A : x \notin B\}$ (attention, ça ne suppose pas que B soit inclus dans A) ; peut aussi être noté $A \backslash B$.
- $\mathcal{P}(A) = \{B : B \subset A\}$: "powerset" de A, l'ensemble de tous les sous-ensembles de A (c'est bien un *ensemble d'ensembles* : un ensemble dont les éléments sont eux-mêmes des ensembles).
- **notations d'unions ou d'intersections itérées :** si pour chaque $i \in I$, A_i est un ensemble,
  - $\bigcup_{i \in I} A_i$ est l'ensemble de tous les éléments qui sont dans **au moins un** des ensembles A_i ;
  - $\bigcap_{i \in I} A_i$ est l'ensemble de tous les éléments qui sont dans **tous** les ensembles A_i.
- l'ensemble de **toutes les fonctions (totales) de A vers B** est parfois noté B^A

Ensembles finis, cardinal

- notion d'ensemble **fini** ou **infini** ; pour un ensemble fini A, son **cardinal** est son nombre d'éléments, noté $\#A$.
- deux ensembles sont **disjoints** si leur intersection est l'ensemble vide ; quand on parle de plus de deux ensembles, il faut distinguer deux notions :
  - des ensembles $(A_i)_{i\in I}$ sont **deux à deux disjoints** si, quelques soient les deux ensembles A_i et A_j avec $i \neq j$, ces deux ensembles sont disjoints (aucun élément n'appartient à plus d'un des ensembles) ;
  - des ensembles $(A_i)_{i\in I}$ sont **globalement disjoints** si leur intersection à tous est vide (aucun élément n'appartient à chacun des ensembles).

Vocabulaire sur les fonctions

E et F deux ensembles, finis ou infinis ; f une fonction de E vers F

- si $x \in E$, $f(x) \in F$ est son **image** par f ; et si $f(x) = y$, x est un **antécédent** par f (il peut *a priori* y en avoir plus d'un).
- **Notations :** si $A \subset E$, $f(A) = \{y \in F : \exists x \in A, y = f(x)\}$; si $B \subset F$, $f^{-1}(B) = \{x \in E : f(x) \in B\}$
- f est **surjective** (une surjection) si chaque $y \in F$ a au moins un antécédent.
- f est **injective** (une injection) si chaque $y \in F$ a au plus un antécédent ; autrement dit, si $f(x) = f(x')$, alors $x = x'$.
- f est **bijective** (une bijection) si elle est à la fois injective et surjective ; autrement dit, chaque $y \in F$ a **exactement** un antécédent.

(Une bijection entre deux ensembles établit une correspondance exacte entre leurs éléments)

Injections, bijections, et cardinaux

L'existence d'injections ou de surjections de E et F se traduit sur la finitude des ensembles, et sur leurs cardinaux :

- S'il existe une surjection de E vers F, et que E est fini, alors F est fini, et $\#E \geq \#F$.
- S'il existe une injection de E vers F, et que F est fini, alors E est fini, et $\#E \leq \#F$.
- Par conséquent, s'il existe une bijection de E vers F, et que l'un des deux ensembles est fini, alors l'autre l'est également, et $\#E = \#F$.

Séquences et mots

- $A \times B$ est l'ensemble des **couples** formés d'un élément de A, puis d'un élément de B : $A \times B = \{(a, b) : a \in A, b \in B\}$

Séquences et mots

- $A \times B$ est l'ensemble des **couples** formés d'un élément de A, puis d'un élément de B : $A \times B = \{(a, b) : a \in A, b \in B\}$
- Pour $A = B$, on note aussi A^2 pour $A \times A$.

Séquences et mots

- $A \times B$ est l'ensemble des **couples** formés d'un élément de A, puis d'un élément de B : $A \times B = \{(a, b) : a \in A, b \in B\}$
- Pour $A = B$, on note aussi A^2 pour $A \times A$.
- Les couples d'éléments de A sont aussi, en fait, les **suites de longueur 2** d'éléments de A.

Séquences et mots

- $A \times B$ est l'ensemble des **couples** formés d'un élément de A, puis d'un élément de B : $A \times B = \{(a, b) : a \in A, b \in B\}$
- Pour $A = B$, on note aussi A^2 pour $A \times A$.
- Les couples d'éléments de A sont aussi, en fait, les **suites de longueur 2** d'éléments de A.
- Plus généralement, A^n (pour un entier $n > 0$) est l'ensemble des **séquences (ou suites) de longueur n, d'éléments de A**.

Séquences et mots

- $A \times B$ est l'ensemble des **couples** formés d'un élément de A, puis d'un élément de B : $A \times B = \{(a, b) : a \in A, b \in B\}$
- Pour $A = B$, on note aussi A^2 pour $A \times A$.
- Les couples d'éléments de A sont aussi, en fait, les **suites de longueur 2** d'éléments de A.
- Plus généralement, A^n (pour un entier $n > 0$) est l'ensemble des **séquences (ou suites) de longueur n, d'éléments de A**.
- Au lieu de **séquences**, on peut aussi parler de **mots** : en considérant A comme un "alphabet" dont les éléments sont appelés "lettres", on appelle également "mots de longueur n sur l'alphabet A" les éléments de A^n. Par convention, A^0 contient un unique élément, noté ε ("le mot vide").

Séquences et mots

- $A \times B$ est l'ensemble des **couples** formés d'un élément de A, puis d'un élément de B : $A \times B = \{(a, b) : a \in A, b \in B\}$
- Pour $A = B$, on note aussi A^2 pour $A \times A$.
- Les couples d'éléments de A sont aussi, en fait, les **suites de longueur 2** d'éléments de A.
- Plus généralement, A^n (pour un entier $n > 0$) est l'ensemble des **séquences (ou suites) de longueur n, d'éléments de A**.
- Au lieu de **séquences**, on peut aussi parler de **mots** : en considérant A comme un "alphabet" dont les éléments sont appelés "lettres", on appelle également "mots de longueur n sur l'alphabet A" les éléments de A^n. Par convention, A^0 contient un unique élément, noté ε ("le mot vide").
- **Notations sur les mots :** $|w|$ pour la longueur du mot w ; w_i pour la i-ème lettre de w $(1 \leq i \leq |w|)$; et si a est une lettre, $|w|_a$ pour le *nombre d'occurrences de a dans w*, soit $|w|_a = \#\{i \in [[1, |w|]] : w_i = a\}$

Mots (suite)

- On introduit la notation A^*, correspondant à la définition

$$A^* = \bigcup_{n \in \mathbb{N}} A^n.$$

(ensemble de **toutes** les suites finies d'éléments de A)

- Sur les mots, on a une nouvelle opération : la **concaténation** : si $w \in A^n$ et $w' \in A^m$, $w \cdot w' \in A^{n+m}$ est défini ainsi : $w \cdot w'$ est le mot w'' défini par

$$w''_i = \begin{cases} w_i & \text{si } 1 \leq i \leq n \\ w'_{i-n} & \text{si } n+1 \leq i \leq n+m \end{cases}$$

- On note souvent les mots sans parenthèses quand cela ne prête pas à confusion : (a, a, b) devient aab ; ainsi, on a $abba \cdot abc = abbaabc$.

Combinatoire

- En **combinatoire énumérative**, on va définir des ensembles d'objets d'intérêt, et chercher à les *énumérer* : dire combien ils ont d'éléments (souvent, il y aura des paramètres).
- En **combinatoire bijective**, on va chercher à "expliquer" des égalités du type $\#A = \#B$ (les ensembles A et B ont le même nombre d'éléments) par la description d'une *bijection* (la plus simple possible) entre A et B.
- Les objets seront souvent des mots, des arbres, des chemins...

Deux principes fondamentaux

A, B désignent des ensembles *finis*.

- **Principe additif :** si A et B sont deux ensembles **disjoints**, alors
$$\#(A \cup B) = \#A + \#B.$$
- **Principe multiplicatif :** quelques soient les ensembles finis A et B, on a
$$\#(A \times B) = \#A \, \#B.$$

Conséquences faciles

En itérant les deux principes (additif et multiplicatif), on en obtient des versions à n'importe quel nombre d'ensembles :

- pour n'importe quel n, si les n ensembles $A_1, \ldots, A_n$ sont **deux à deux disjoints**, alors

$$\# \left(\bigcup_{i=1}^{n} A_n \right) = \sum_{i=1}^{n} \# A_i.$$

- pour n'importe quel n,

$$\# \left(A_1 \times A_2 \times \cdots \times A_n \right) = \prod_{i=1}^{n} \# A_i.$$

(Preuves : par récurrence sur n)

Cardinal d'une union, en général

On a une formule plus générale que le principe additif, valable quelques soient les ensembles :

Théorème

Soient A et B deux ensembles finis quelconques. On a

$$\#(A \cup B) = \#A + \#B - \#(A \cap B).$$

En particulier, on a toujours $\#(A \cup B) \leq \#A + \#B$.

Cardinal d'une union, en général

On a une formule plus générale que le principe additif, valable quelques soient les ensembles :

Théorème

Soient A et B deux ensembles finis quelconques. On a

$$\#(A \cup B) = \#A + \#B - \#(A \cap B).$$

En particulier, on a toujours $\#(A \cup B) \leq \#A + \#B$.

Preuve : On commence par écrire $A \cup B$ comme union de trois ensembles deux à deux distincts :
$A \cup B = (A - B) \cup (B - A) \cup (A \cap B)$.
Puis, on remarque que A est l'union de deux d'entre eux :
$A = (A - B) \cup (A \cap B)$; et de manière similaire pour B :
$B = (B - A) \cup (A \cap B)$.
On écrit le principe additif pour chaque union, on triture, et. . .

Cardinal d'une union, en général

On a une formule plus générale que le principe additif, valable quelques soient les ensembles :

Théorème

Soient A et B deux ensembles finis quelconques. On a

$$\#(A \cup B) = \#A + \#B - \#(A \cap B).$$

En particulier, on a toujours $\#(A \cup B) \leq \#A + \#B$.

Preuve : On commence par écrire $A \cup B$ comme union de trois ensembles deux à deux distincts :
$A \cup B = (A - B) \cup (B - A) \cup (A \cap B)$.
Puis, on remarque que A est l'union de deux d'entre eux :
$A = (A - B) \cup (A \cap B)$; et de manière similaire pour B :
$B = (B - A) \cup (A \cap B)$.
On écrit le principe additif pour chaque union, on triture, et...
Conséquence (évidente ?) : Si $A \subset B$, alors $\#A \leq \#B$.

Notion de “classe combinatoire”

En combinatoire, on va souvent chercher à “compter les objets d’un certain type”. Le plus souvent, l’ensemble de tous les objets en question est infini, donc la réponse est en apparence simple : “il y en a une infinité”. Mais on a généralement une notion de **taille** des objets, et la vraie question est plutôt : “combien, en fonction de n, y a-t-il d’objets de taille n ?”

Notion de “classe combinatoire”

En combinatoire, on va souvent chercher à “compter les objets d’un certain type”. Le plus souvent, l’ensemble de tous les objets en question est infini, donc la réponse est en apparence simple : “il y en a une infinité”. Mais on a généralement une notion de **taille** des objets, et la vraie question est plutôt : “combien, en fonction de n, y a-t-il d’objets de taille n ?”

Définition : Classe combinatoire

Une classe combinatoire est la donnée d’un ensemble C, et d’une fonction $t : C \to \mathbb{N}$, telle que *pour tout n, l’ensemble C_n des éléments de C qui ont n par image par t est fini*.

$$C_n = \{x \in C : t(x) = n\}.$$

La *suite de comptage de la classe C* est alors simplement la suite $(c_n)_{n\geq 0}$ définie par : pour tout n, $c_n = \#C_n$.

Notion de “classe combinatoire”

En combinatoire, on va souvent chercher à “compter les objets d’un certain type”. Le plus souvent, l’ensemble de tous les objets en question est infini, donc la réponse est en apparence simple : “il y en a une infinité”. Mais on a généralement une notion de **taille** des objets, et la vraie question est plutôt : “combien, en fonction de n, y a-t-il d’objets de taille n ?”

Définition : Classe combinatoire

Une classe combinatoire est la donnée d’un ensemble C, et d’une fonction $t : C \to \mathbb{N}$, telle que *pour tout n, l’ensemble C_n des éléments de C qui ont n par image par t est fini*.

$$C_n = \{x \in C : t(x) = n\}.$$

La *suite de comptage de la classe C* est alors simplement la suite $(c_n)_{n\geq 0}$ définie par : pour tout n, $c_n = \#C_n$.

La fonction t est appelée fonction de taille pour la classe ; si on change t, on change généralement de suite de comptage (et même, on peut ne plus avoir une classe combinatoire).

Comptage direct d'une classe combinatoire

Réaliser le comptage d'une classe combinatoire, c'est en règle générale trouver une formule pour sa suite de comptage.
La méthode "directe" pour cela, c'est de trouver un *codage* pour les objets d'une taille n fixée : décrire une façon *exhaustive* et *non ambigüe* de définir un objet de taille n, de manière à ce qu'on soit capable d'écrire une formule pour le nombre de codages.

- **exhaustive :** chaque objet doit avoir un codage
- **non ambigüe :** chaque objet ne doit avoir qu'un seul codage

Comptage direct d'une classe combinatoire

Réaliser le comptage d'une classe combinatoire, c'est en règle générale trouver une formule pour sa suite de comptage.
La méthode "directe" pour cela, c'est de trouver un *codage* pour les objets d'une taille n fixée : décrire une façon *exhaustive* et *non ambigüe* de définir un objet de taille n, de manière à ce qu'on soit capable d'écrire une formule pour le nombre de codages.

- **exhaustive :** chaque objet doit avoir un codage
- **non ambigüe :** chaque objet ne doit avoir qu'un seul codage

Parfois, on n'obtient pas une formule pour la suite, mais seulement une récurrence qui permet de calculer facilement les termes de la suite de proche en proche, mais qu'on ne sait pas résoudre en une formule close.

Quelques exemples

On prend un alphabet à 2 lettres, $A = \{a, b\}$, et comme ensemble, $C = A^*$: tous les mots sur l'alphabet A.

- avec comme fonction taille, la longueur du mot, $t(w) = |w|$: on a une classe combinatoire, et la suite de comptage est $a_n = 2^n$ (car $C_n = A^n$) ;

Quelques exemples

On prend un alphabet à 2 lettres, $A = \{a, b\}$, et comme ensemble, $C = A^*$: tous les mots sur l'alphabet A.

- avec comme fonction taille, la longueur du mot, $t(w) = |w|$: on a une classe combinatoire, et la suite de comptage est $a_n = 2^n$ (car $C_n = A^n$) ;
- avec comme fonction taille, le nombre d'occurrences de la lettre a ($t'(w) = |w|_a$), on n'a pas une classe combinatoire : en effet on peut former une infinité de mots de "taille" 0 (mots sans la lettre a).

Quelques exemples

On prend un alphabet à 2 lettres, $A = \{a, b\}$, et comme ensemble, $C = A^*$: tous les mots sur l'alphabet A.

- avec comme fonction taille, la longueur du mot, $t(w) = |w|$: on a une classe combinatoire, et la suite de comptage est $a_n = 2^n$ (car $C_n = A^n$) ;
- avec comme fonction taille, le nombre d'occurrences de la lettre a ($t'(w) = |w|_a$), on n'a pas une classe combinatoire : en effet on peut former une infinité de mots de "taille" 0 (mots sans la lettre a).
- avec comme fonction taille, le nombre d'occurrences de la lettre a plus deux fois le nombre d'occurrences de la lettre b (notation : $t''(w) = |w|_a + 2|w|_b$), on a bien une classe combinatoire (preuve ?), et la suite de comptage commence ainsi : $c_0 = 1$, $c_1 = 1$, $c_2 = 2$, $c_3 = 3$, $c_4 = 5$, $c_5 = 8$...(peut-on deviner la suite ?)

Exemple classique : parties d'un ensemble

Soit E un ensemble fini, $n = \#E$, et k un entier compris entre 0 et n.

Le nombre de sous-ensembles de E est...

Exemple classique : parties d'un ensemble

Soit E un ensemble fini, $n = \#E$, et k un entier compris entre 0 et n.

Le nombre de sous-ensembles de E est... 2^n (preuve : par codage via les éléments de $\{V, F\}^n$)

Exemple classique : parties d'un ensemble

Soit E un ensemble fini, $n = \#E$, et k un entier compris entre 0 et n.

Le nombre de sous-ensembles de E est... 2^n (preuve : par codage via les éléments de $\{V, F\}^n$)

On note $\binom{n}{k}$ le nombre de sous-ensembles de cardinal k de E.

Exemple classique : parties d'un ensemble

Soit E un ensemble fini, $n = \#E$, et k un entier compris entre 0 et n.
Le nombre de sous-ensembles de E est... 2^n (preuve : par codage via les éléments de $\{V, F\}^n$)
On note $\binom{n}{k}$ le nombre de sous-ensembles de cardinal k de E.
Avant même d'écrire une formule pour $\binom{n}{k}$, on peut montrer que $\binom{n}{k} = \binom{n}{n-k}$ (preuve : bijection entre...)

Exemple classique : parties d'un ensemble

Soit E un ensemble fini, $n = \#E$, et k un entier compris entre 0 et n.
Le nombre de sous-ensembles de E est... 2^n (preuve : par codage via les éléments de $\{V, F\}^n$)
On note $\binom{n}{k}$ le nombre de sous-ensembles de cardinal k de E.
Avant même d'écrire une formule pour $\binom{n}{k}$, on peut montrer que $\binom{n}{k} = \binom{n}{n-k}$ (preuve : bijection entre...)
On peut également prouver une relation de récurrence ("triangle de Pascal") : pour $0 < k < n$,

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}.$$

Exemple classique : permutations

Définition : permutation

Une *permutation* sur un ensemble A est une bijection de E vers lui-même.
On note S_n l'ensemble des permutations sur l'ensemble $[[1, n]]$, et $S = \cup_{n\in\mathbb{N}} S_n$ (on prend donc comme "taille" d'une permutation, le nombre d'éléments de l'ensemble).

La suite de comptage des permutations est connue :

Théorème

Pour tout entier $n \geq 1$, le nombre de permutations d'un ensemble de cardinal n, est $n! = n.(n-1)\ldots 2.1$ ("factorielle n").

Preuve : On va donner deux preuves, une preuve directe par codage, et une preuve par récurrence sur n.

Comptage des permutations

Preuve par récurrence : on a $n! = n.(n-1)!$; on va montrer que pour tout $n > 1$, il y a n fois plus de permutations sur $[[1, n]]$ que sur $[[1, n-1]]$ (et que le nombre de permutations de $\{1\}$ est bien $1 = 1!$).
Preuve par codage : on choisit un entier n quelconque ; on identifie un ensemble A_n pour lequel il est facile de montrer que son cardinal est $n!$, et on trouve un codage des permutations de $[[1, n]]$ par les éléments de cet ensemble (*i.e.*, une bijection de A_n vers S_n.
Au passage : $n!$ est aussi le nombre de façons d'ordonner n valeurs (séquences de longueur n d'entiers entre 1 et n, tous différents).