FONDEMENTS DU RAISONNEMENT ET DU CALCUL

PIERRE GUILLOT ET JULIEN NARBOUX

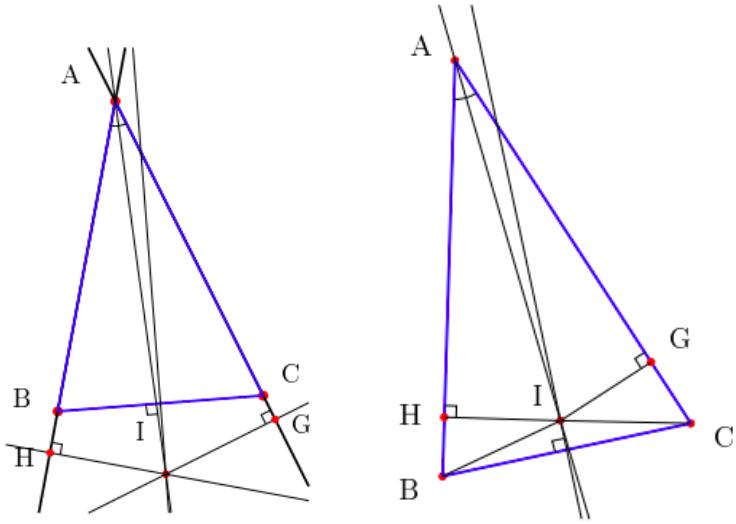
Table des matières

1. Introduction

Attention preuve fausse !

Tous les triangles sont isocèles. Démonstration : Soit ABC un triangle quelconque.
Soit $\mathcal{D}$ la médiatrice de $[BC]$ et $\mathcal{D}'$ la bissectrice intérieure de l'angle $\angle BAC$.
Si $\mathcal{D} \parallel \mathcal{D}'$ alors ABC est isocèle en A.
Sinon, $\mathcal{D}$ et $\mathcal{D}'$ se coupent en un point I. Il y a deux cas :
Si I est dans le triangle ABC. Soit H (resp. G) le pied de la perpendiculaire à la droite (AB) (resp. (AC)) passant par I. Les triangles AIG et AIH sont égaux car ils ont un côté commun et deux angles égaux, donc $AH = AG$ et $IH = IG$. De plus $IB = IC$ car I est sur la médiatrice de $[BC]$. De même les triangle IGC et IHB sont égaux (un angle droit et deux côtés égaux), donc $HB = GC$. Comme on a $AH = AG$ et $HB = GC$ par addition, on a $AB = AC$.
Sinon I est en dehors du triangle ABC. La preuve est similaire, excepté qu'à la fin on réalise une soustraction plutôt qu'une addition.

Date: 2018.



preuve faussepreuve fausse

Remerciements. Ce polycopié mentionne des exercices et exemples tirés des publications et polycopiés de Viviane Durand-Guerrier, Zoe Mesnil, René Cori, T Joly, A. Pierrot, à compléter.

2. Le discours mathématique

2.1. **La structure générale d'un texte mathématique.** Décrivons quelques-uns des éléments de structure que l'on retrouve dans tous les textes mathématiques. Nous commençons par décrire ce qu'est une « définition », en prenant un exemple :

> Définition 2.1 – On dit qu'un entier n est pair lorsqu'il existe un entier k tel que $n = 2k$.

En général dans une définition, il y a un mot ou un groupe de mots mis en valeur, ici « pair » ; c'est le mot que l'on définit. Le principe, très simple, est de donner un nom court à une propriété (ou un un objet) parfois long à décrire. (Le fonctionnement est le même que pour les « macros » en informatique.) Dans la suite, à chaque fois que l'on voit « n est pair », où n est un entier, on le remplace (mentalement...) par la définition donnée.

Par exemple l'expression « 13 est pair » a un sens (c'est d'ailleurs faux), de même que « 24 est pair » (et cette fois c'est vrai). (Si n n'est pas un entier, on va en général considérer que l'expression « n est pair » n'a pas de sens, et en tout cas on va éviter de l'utiliser ; d'autres conventions seraient possibles.)

Plus loin vous pouvez voir :

> Définition 2.2 – Un entier qui n'est pas pair est dit impair.

Le mot « pair » étant défini, on peut l'employer dans la définition suivante.

Autre élément (célèbre !) du discours mathématique, le théorème :

> Théorème 2.3 – *Soit n un entier impair. Alors il existe un entier k tel que $n = 2k + 1$.*

Le mot « théorème » annonce que l'énoncé présenté va découler de ce qui précède – si vous êtes convaincu par le texte jusqu'ici, alors vous devrez accepter aussi le contenu du théorème.

Précisons que lorsqu'un énoncé est supposé vrai à un moment donné, on dit que c'est une hypothèse (dans le contexte considéré). Le fait que « n est un entier impair » est une hypothèse du théorème ; le fait que « $n = 2k+1$ » est une conclusion du théorème. Le théorème lui-même affirme que, quand ses hypothèses sont vraies, alors ses conclusions aussi. (On peut imaginer un théorème sans hypothèses, du genre « $2 + 2 = 4$ », mais dans l'ensemble, c'est plutôt rare ; et de toute façon, il y a toujours les hypothèses implicites, comme « $+$ est le symbole pour l'addition ».)

On ne peut pas vraiment continuer la discussion autour de la nature des théorèmes sans parler de démonstration ; en voici une, pour rester concret :

Démonstration. Soient q et r le quotient et le reste, respectivement, dans la division euclidienne de l'entier impair n par 2. On a donc

$$n = 2q + r$$

et $0 \leq r < 2$. Puisque r est entier, on a $r = 0$ ou $r = 1$. Voyons le cas $r = 0$: on a alors $n = 2q$, ce qui prouve que n est pair par définition ; or c'est absurde, car on a supposé le contraire. Donc le cas $r = 0$ mène à une absurdité, et l'on se tourne vers le cas $r = 1$. Alors $n = 2q + 1$. C'est précisément l'énoncé que l'on souhaite, avec $k = q$. □

(Le symbole □, dans un texte mathématique imprimé, symbolise généralement la fin de la démonstration.)

Dans ce cours, et dans ce polycopié, on va passer beaucoup de temps à décrire ce qu'est une démonstration, mais il semble utile de dire tout de suite quelques mots, en guise de préliminaires. La démonstration est un texte qui doit convaincre le lecteur que l'énoncé du théorème est une conséquence inévitable de ce qui a été accepté jusque-là. Les anglo-saxons disent « proof » pour démonstration, l'anglicisme « preuve » est maintenant très répandu, et la métaphore juridique est, à vrai dire, assez bonne. Il faut apporter des éléments au lecteur pour qu'il n'ait plus le moindre doute sur l'énoncé.

A l'évidence, « convaincre le lecteur », voilà qui est ambigu. Que faire si le lecteur émet toujours plus de doutes quant à nos arguments ? Il faut découper chaque étape de plus en plus finement, bien sûr. Au bout d'un moment, pour clore les débats, si vraiment on ne peut pas faire autrement, le mathématicien et son lecteur doivent tomber d'accord sur des règles de logique qu'ils acceptent tous les deux (et ne vont plus jamais remettre en cause au cours de la discussion).

Dans ce cours, nous décrirons les règles de la logique dite « usuelle ». Ce sont les règles qui paraissent intuitives, et que l'on a bien envie d'utiliser. Par exemple, dans la démonstration ci-dessus, on rencontre une contradiction à un moment donné, et il y a une règle de logique qui indique – conformément à ce qu'on attend, mais avec précision – ce que l'on peut conclure. Tout ceci est détaillé plus loin dans le cours. Par ailleurs, nous n'aurons pas le temps d'en parler, mais cette logique usuelle a une grande cohérence interne : on pourrait réduire drastiquement le nombre de règles, et déduire les autres. On pourrait aussi définir ce qu'est une « formule valide » et prouver que les règles préservent la validité. Tout ceci augmente la confiance que l'on peut avoir dans les règles, mais nous le ferons pas (ce serait l'objet d'un cours de logique).

Cependant, il faut bien admettre qu'il s'agit là d'un ensemble de conventions. Par exemple, pour nous, le « principe du tiers exclu » sera valide : il affirme que toute proposition est soit vraie, soit fausse. Une chose qui va peut-être vous surprendre, c'est qu'il y a des mathématiciens qui travaillent en refusant d'utiliser ce principe ! (Par exemple, pour vérifier s'il est vraiment utile pour arriver à telle ou telle conclusion.) Bien sûr, ce genre de subtilité ne nous concerne pas dans ce cours.

Une démonstration, finalement, est un texte qui convainc le lecteur que, au besoin, une preuve formelle pourrait être produite, c'est-à-dire une rédaction avec chaque règle de logique indiquée.

Une chose est à noter. Pour donner les règles logiques, il va bien falloir définir ce qu'est une « proposition », comment en bâtir de nouvelles, etc. Or, les objets d'étude du mathématicien sont les nombres, les figures géométriques, etc, mais pas les propositions. Celles-ci sont les objets d'étude de la logique. Les mathématiciens font donc appel à la logique quand c'est utile, de même qu'un architecte fait fréquemment appel aux mathématiques, mais pas à chaque fois qu'il accroche un tableau au mur. Dans ce cours on se contentera d'une définition informelle des propositions mathématiques.

L'informatique est aussi une science cousine de la logique, le formalisme logique étant omniprésent de la conception des ordinateurs, à la spécification et preuve de programmes. D'ailleurs, quand on lit la description d'un nouveau langage informatique, on a sous les yeux quelque chose qui est parfois très proche de la description informelle, qui suit, du langage des mathématiques.

Pour que le mathématicien et son lecteur s'entendent, il y a une autre chose à régler : le point de départ. Il faut bien tomber d'accord sur un certain nombre de faits, qui sont acceptés une fois pour toutes. Ces énoncés acceptés sont appelés les axiomes. Les axiomes « usuels », pour les mathématiques de tous les jours, sont extrêmement rébarbatifs à lire, et finalement leur formulation est très abstraite, donc dans ce cours on ne cherchera pas vraiment à vous les indiquer. Sachez qu'il est possible de débattre sur la définition, et les propriétés basiques, des nombres entiers, jusqu'à atteindre une précision totale. Mais nous ne le ferons pas ici.

Il sera tout de même utile de savoir ce qu'est un axiome. Parfois, un professeur ne souhaite pas démontrer un certain résultat, et annoncera « pour l'instant, prenons ça comme un axiome » ; par exemple, au lycée vous avez un peu parlé des limites de fonctions, sans jamais vraiment démontrer leur existence, ni même les définir proprement. Mais ça n'empêche pas de travailler avec les limites, en acceptant leurs propriétés comme axiomes (quitte à démontrer plus tard que ces axiomes sont en fait conséquences de choses plus basiques).

Voici pour le concept de preuve formelle, et pour les axiomes. Retournons aux éléments constitutifs d'un texte mathématique. Vous allez recontrer des « lemmes ». Un lemme, c'est exactement la même chose qu'un théorème, mais on indique au lecteur que l'énoncé n'est pas inoubliable, et que la démonstration sera facile. Il s'agit en général d'une étape intermédiaire avant un théorème plus consistant. L'exemple de théorème ci-dessus serait sans doute un lemme dans la plupart des cours de maths !

Il y a des exemples historiques amusants de lemmes qui se sont avérés d'une importance capitale : le lemme de Zorn, le lemme de Schur, le lemme d'Urysohn, etc. Les auteurs peuvent se tromper...

Très souvent, vous rencontrerez aussi des énoncés intitulés « Proposition » ; c'est intermédiaire entre lemme et théorème, mais attention ! Dans un cours de logique, le terme de « proposition » signifie autre chose (voir ci-dessous), donc nous ne parlerons plus de cet emploi (même si ce cours n'est pas un cours de logique, nous empruntons souvent son vocabulaire).

Autre terme que vous découvrez peut-être : un « corollaire » est encore un théorème, mais on indique au lecteur que c'est une conséquence facile du dernier théorème. Par exemple, après le théorème que nous avons démontré, on pourrait énoncer, même si c'est un peu idiot :

COROLLAIRE 2.4 – *Si n et m sont tous les deux impairs, alors $n+m$ est pair.*

Démonstration. D'après le théorème, on peut écrire $n = 2k+1$ et $m = 2\ell+1$, où k et ℓ sont des entiers. Donc $n+m = 2k+1+2\ell+1 = 2(k+\ell+1)$, et on a bien montré que $n+m$ est pair. □

Enfin, vous entendrez aussi parler de « conjecture ». Une conjecture est une proposition dont on n'a pas encore de démonstration, mais que l'on pense être vraie car on n'a pas trouvé de contre-exemple. Par exemple, la conjecture des nombres premiers jumeaux est que : « Il existe une infinité de nombres premiers p tels que $p+2$ soit aussi premier. »

Et pour en finir avec le vocabulaire, citons le mot « postulat », qui est synonyme de « axiome ».

2.2. **Les variables.** Dans le discours mathématique nous utilisons des variables. Toutes les variables ne jouent pas le même rôle. Quand je dis "n est premier" il s'agit d'un énoncé qui parle d'un entier qui s'appelle n. En revanche, quand je dis "Si n est premier alors n^2 est premier.", cet énoncé ne parle pas d'un entier n particulier, on pourrait renommer n sans changer le sens de l'énoncé : "Si m est premier alors m^2 est premier". On peut même se passer de nommer la variable : "Le carré d'un entier impair est impair".

On appelle les variables que l'on peut renommer en utilisant un nom nouveau (qui n'apparaissait pas avant) sans changer le sens de l'énoncé, des variables muettes ou liées. Les autres sont dites libres ou parlante.

Il y a des signes qui rendent les variables liées, on les appelle des lieurs.

Par exemple, dans le discours mathématique les signes suivants sont des lieurs :

$$x \mapsto x+1 \qquad \sum_{i=0}^{n} f(i) \qquad \prod_{i=0}^{n} f(i)$$
$$\int_0^{10} f(x)\,\mathrm{d}x \qquad \lim_{x\to+\infty} f(x)$$
$$\forall x\ P(x) \qquad \exists x\ P(x)$$

Le lieur $x \mapsto f(x)$ apparaît aussi en programmation sous différentes syntaxes suivant le langage.

Par exemple en Python ces deux programmes ont le même sens :

```
def f(x):
    return (x+1)
```

```
def f(z):
    return (z+1)
```

Le filtrage par motifs présent dans certains language (Ocaml, Haskell, Scala, ...) introduit aussi une forme de lieur. Dans l'exemple en Ocaml ci-dessous, le motif `t::r` permet de lier les variables `t` et verb|r|.

```
let length l =
 match l with
  [] -> 0
 |t::r -> 1 + length r
```

Un programme ne doit pas mentionner de variables libres, sinon on obtient un message d'erreur, par exemple en Ocaml si nous essayons d'utiliser une variable `a` pas définie (c'est à dire pas liée, *unbound*) :

```
# let f x = a;;
Error: Unbound value a
```

Attention, en fait, il faut être un peu plus précis et ne pas parler de variable liée ou libre, mais d'occurrence liée ou libre, en effet une même variable peut à la fois apparaître libre et liée dans un même énoncé.

Par exemple, si on écrit

$$i + \sum_{i=0}^{n} i^2,$$

la première occurrence de i est libre et la deuxième est liée.

On aurait pu écrire :

$$i + \sum_{k=0}^{n} k^2$$

En général, pour plus de clarté on donne un nom différent et distinct des variables libres à chacune des variables liées d'une expression. Dans ce cours on adoptera cette convention et on s'interdira d'écrire la première version.

Une expression mathématique qui comprend une ou plusieurs variables libres est une expression dite ouverte ou contingente, dans le cas contraire, on parlera d'expression close ou fermée.

Les expressions ouvertes n'ont pas de valeur de vérité, par exemple l'expression $x > 4$ n'est ni vrai, ni fausse, ça dépend de la valeur de x.

2.3. **Deux jeux d'assemblage.** *Attention, les points abordés dans cette sous-partie sont un peu difficiles. Dans une première lecture, on peut les ignorer, et passer directement aux connecteurs logiques, ci-dessous.*

Il y a une analogie entre la structure d'un discours mathématique et la programmation en informatique [1]. Les propositions mathématiques sont démontrées en faisant appel aux propositions précédentes jusqu'à arriver à des axiomes. En programmation, les programmes sont obtenu en écrivant des fonctions utilisant les fonctions précédentes, jusqu'à arriver à une interface restreinte avec l'extérieur parfois appelée API (*Application Programming Interface*). Cette interface restreinte avec l'extérieur en informatique permet la portabilité du code et rend celui-ci réutilisable : le programme fonctionnera alors avec n'importe quelle autre implantation de cette interface. De même, en mathématiques un propriété des groupes pourra s'appliquer à n'importe quel groupe. Ces interfaces abstraites sont omniprésente en informatique, elles se situent à différents niveaux : systèmes d'exploitation, microprocesseurs, bibliothèque de fonctions, ...

En informatique, les programmes peuvent être vus comme des transformateurs de propriétés. Apparaît alors une dualité entre le programmeur d'une fonction et l'utilisateur. Le programmeur doit garantir que sa fonction est correcte pour toute entrée vérifiant une certaine propriété. La propriété supposée sur l'entrée d'une fonction est appelée pré-condition. La propriété garantie pour la sortie de la fonction est appelée post-condition. Écrire les pré et post conditions s'appelle spécifier

1. Si vous voulez étudier de manière un peu plus profonde cette analogie, chercher des informations sur la correspondance de Curry-Howard

le comportant d'un programme. Cette notion a donné naissance à la notion de programmation par contrat. Ceux qui continueront en informatique verront cela en détail dans les cours sur la certification du logiciel.

De manière plus générale, le goût pour la généricité est partagé par informaticien et mathématicien et en fait des sciences cousines.

Prenons un exemple, supposons que l'on souhaite écrire un programme calculant la puissance entière n-ième d'un entier x. Pour cela, on pourrait écrire une petite fonction récursive qui renvoie 1 si x vaut 0 et sinon renvoie $x \times x^{n-1}$ (ou une boucle ça serait pareil). Cette fonction utilise alors $n-1$ multiplications pour calculer x^n. On peut aussi écrire un programme un peu plus efficace en utilisant la remarque suivante :

$$x^n = \begin{cases} (x^2)^{\frac{n}{2}} & \text{si } n \text{ est pair} \\ x(x^2)^{\frac{n}{2}} & \text{sinon} \end{cases}$$

Par exemple en Python comme la division euclidienne est noté `//` :

```
def puiss(x,n):
    if (n==0):
        return(1);
    else:
        return(x*puiss(x,n-1));

def puiss2(x,n):
    if (n==0):
        return(1);
    elif (n%2==0):
        return(puiss2(x*x, n//2));
    else:
        return(x*puiss2(x*x, n//2));
```

La théorème indiquant que notre programme est correct est alors le suivant : Si n et x sont des entiers, alors `puiss2(x,n)=puiss(x,n)`.

Supposons maintenant que je veuille maintenant concaténer n fois une chaîne de caractères s, je peux utiliser la même astuce. La concaténation étant notée `+` en Python, on obtient :

```
def puiss3(x,n):
    if (n==0):
        return("");
    elif (n%2==0):
        return(puiss3(x+x, n//2));
    else:
        return(x+puiss3(x+x, n//2));
```

Il devient alors évident que l'on ne souhaite pas écrire une nouvelle fonction à chaque fois que l'on à nouvel exemple similaire. La question est donc quand est-ce que l'on peut utiliser la même technique et comment généraliser ?

La réponse est une structure mathématique bien connue (appelée Monoïde), celle où l'on a une ensemble muni d'une opération interne binaire associative et d'un élément neutre.

On peut alors écrire une fonction générique qui prend en argument une opération interne sur un ensemble `o` et un élément `e`, et notre fonction calculera bien la

puissance n-ième dés lors que `o` est associative et que `e` est un élément neutre pour `o`.

```
def puiss_gen(o,e,x,n):
    if (n==0):
        return(e);
    elif (n%2==0):
        return(puiss_gen(o,e,o(x,x), n//2));
    else:
        return(o(x,puiss_gen(o,e,o(x,x), n//2)));

def mult(x,y):
    return(x*y);

def puiss_ent(x,n):
    return(puiss_gen(mult,1,x,n));

def puiss_string(x,n):
    return(puiss_gen((lambda x,y: x+y),"",x,n));
```

Nous avons renforcé le contrat de notre fonction, en faisant une hypothèse plus faible, i.e. en ayant une pré-condition plus faible sur l'entrée.

Remarquons au passage qu'il est important d'être rigoureux sur les hypothèses réalisées. Par exemple les opérations d'addition et de multiplication sur les nombres en virgule flottante (float) ne sont pas associatives et cela peut créer des erreurs d'arrondis. Des erreurs d'arrondi qui s'accumulent peuvent produire des résultats aberrants (voir par exemple les bugs du système de missiles américain *Patriot*).

Cette dualité entre celui qui construit un programme et celui qui l'utilise, se retrouve de manière analogue en mathématique, entre celui qui démontre une propriété et celui qui utilise la proposition prouvée pour en démontrer une autre.

Les propriétés une fois prouvées peuvent être vues comme des boites noires. Il n'est pas nécessaire de connaître le détail d'une démonstration pour se servir de la propriété démontrée. De même en programmation, en principe, il n'est pas nécessaire de savoir comment une bibliothèque est programmée pour pouvoir s'en servir.

2.4. **Les connecteurs logiques.** Dans cette partie, nous allons étudier les connecteurs logiques principaux. Ils nous permettent de créer de nouvelles propositions en combinant celles qu'on a déjà.

Voici toute de suite les notations usuelles de ces connecteurs en langue naturelle, en logique/informatique et dans les langages de programmation que vous serez sans doute amenés à manipuler :

En français			En Ocaml	En Python	En C
et	$\wedge$	.	`&&`	`and`	`&&`
ou	$\vee$	$+$	`||`	`or`	`||`
ou exclusif	$\oplus$		`<>`	`!=`	`!=`
non	$\neg$	$\bar{}$	`not`	`not`	`!`
implique	$\Rightarrow$				
équivalent	$\Longleftrightarrow$		`=`	`==`	`==`
vrai	$\top$	1	`true`	`True`	
faux	$\bot$	0	`false`	`False`	

Ainsi, si A et B sont des propositions mathématiques (correctement écrites), alors on peut former $A \vee B$ ou $\neg(A)$ ou encore $A \implies B$, qui sont de nouvelles propositions (correctement écrites !). On peut bien sûr continuer, et parler de

$$[(A \wedge C) \vee (\neg(B))] \implies D,$$

etc, etc.

Au-delà de l'écriture, on souhaite pouvoir parler des cas dans lesquels une proposition est vraie ou fausse, évidemment. La valeur de vérité de, disons, $A \wedge B$, ou de toute autre construction obtenue à partir des propositions A et B, va inévitablement dépendre des valeurs de vérité de ces deux-là. On donne alors une table de vérité, qui couvre tous les cas.

A	B	$A \wedge B$	$A \vee B$	$A \Rightarrow B$	$A \iff B$
$Faux$	$Faux$	$Faux$	$Faux$	$Vrai$	$Vrai$
$Faux$	$Vrai$	$Faux$	$Vrai$	$Vrai$	$Faux$
$Vrai$	$Faux$	$Faux$	$Vrai$	$Faux$	$Faux$
$Vrai$	$Vrai$	$Vrai$	$Vrai$	$Vrai$	$Vrai$

A	$\neg A$
$Faux$	$Vrai$
$Vrai$	$Faux$

Par exemple, la table nous informe que, si A est faux et B et vrai, alors $A \wedge B$ est faux. Ça correspond à notre intuition du connecteur « et » en français !

D'une manière générale, ces tables ne sont pas surprenantes. La seule à propos de laquelle on peut se poser des questions est celle de l'implication $\implies$. Il est tout-à-fait normal de voir que, si A est vraie, et si $A \implies B$ est vraie, alors B est également vraie (c'est lié au *modus ponens*, voir ci-dessous). Par contre, si A est fausse, alors cette table affirme que $A \implies B$ est vraie, quelle que soit la valeur de vérité de B (règle du *ex falso quodlibet*, là encore on y reviendra). Nous verrons plus bas que le connecteur $\implies$ est surtout pertinent en combinaison avec le symbole $\forall$: voir l'exemple 2.8 pour des éclaircissements.

On retiendra que le seul cas où A n'implique pas B, c'est quand A est vrai et B est faux. Et on retiendra par cœur que $A \Rightarrow B$ est équivalent à $(\neg A) \vee B$.

Remarquons que si A est la proposition « $x > 4$ », alors sa valeur de vérité dépend de x, et A n'est ni vraie, ni fausse, en l'attente d'informations supplémentaires ; on peut alors dire la même chose de $\neg A$ ou de $A \vee B$ etc, qui ne sont « pas encore » vraies ou fausses. Dans ces situations, on préfère une notation du type $A(x)$ plutôt que A.

Remarque 2.5. (1) Nous n'avons pas listé tous les connecteurs binaires, mais seulement les principaux. Pour ceux qui suivront un cours de logique, vous verrez que l'on peut définir certains connecteurs à partir d'autres, et que l'on peut les définir tous à partir du *nand* (« non et »). Le nand n'étant pas cher à produire, il est très utilisé dans les circuits électroniques.

(2) Remarquons au passage que nous avons supposé que les propositions n'ont que deux valeurs de vérité possibles : vrai ou faux. Il existe d'autres logiques où l'on considère plus de valeurs de vérité. Par exemple en logique tri-valuée les valeurs de vérité sont : Vrai, Faux, JeNeSaisPas, ou alors en logique floue, la valeur de vérité est un réel compris entre 0 et 1.

(3) Finalement, une proposition mathématique, c'est une proposition obtenue à partir des expressions atomiques qu'on s'autorise d'entrée, comme « $x \in E$ » ou « $1 + 1 = 5$ » (mais pas, disons, « $2+ =\} \in$ »), en utilisant de manière

répétée des connecteurs logiques. On va aussi se permettre de glisser des quantificateurs, que l'on va décrire maintenant.

2.5. **Les quantificateurs.** Pour l'instant, nous avons vu ce qu'on appelle la logique propositionnelle, cest-à-dire l'étude des formules que l'on peut construire avec $\neg, \Rightarrow, \vee, \wedge, \iff$. Cette logique est relativement simple dans le sens où elle est décidable : il existe un algorithme permettant de dire en temps fini si une formule est valide [2] ou non. Il suffit de faire la table de vérité.

Maintenant, introduisons un nouvel outil, plus complexe. Les groupes de mots « pour tout » (ou « quel que soit »), et « il existe au moins un » s'appellent des quantificateurs. Le « pour tout » est noté $\forall$ et s'appelle le quantificateur universel, et le « il existe au moins », noté $\exists$, s'appelle le quantificateur existentiel.

La plupart du temps, quand on rencontre la proposition $\forall y\ A$ ou $\exists y\ A$, la variable y a au moins une occurrence libre dans A, et on écrit souvent $A(y)$ pour souligner ce fait.

L'effet des quantificateurs est exactement celui que l'intuition suggère. Mais puisqu'il faut bien rendre les règles du jeu explicites, on va préciser.

Sans surprise, en présence d'un $\forall$, on a une règle qui nous autorise la chose suivante :

Règle d'utilisation du "pour tout"

Si on a $\forall x\ P(x)$, alors on peut en déduire $P(a)$ quel que soit a (du bon type).

Exemple 2.6 – Si on sait que f est une fonction définie sur $\mathbb{R}$, alors $\forall x\ f(x) \neq 0$ signifie que f ne s'annule jamais. (Ici, dans la formule il est sous-entendu que x est dans $\mathbb{R}$, c'est pour ça que l'on indique « du bon type » dans la règle). Autre exemple, la formule $\forall x \forall y \quad x \leq y \implies f(x) \leq f(y)$ signifie que f est croissante.

Et dans l'autre sens, si on nous demande de démontrer une formule avec un $\forall$, voici une façon de faire qui est acceptée :

Règle de construction du "pour tout"

Pour démontrer $\forall x\ P(x)$, il suffit de montrer $P(x)$ sans faire d'hypothèse sur x.

Vous ne devriez pas être étonnés de ces règles. Dans la suite du poly, nous allons systématiquement donner les règles par paires, une « d'utilisation » et une de « construction ». Il s'agit à chaque fois de formaliser ce que l'on a bien envie de considérer comme vrai. Par exemple, les voici pour le $\exists$.

Règle d'utilisation du "existe"

Si on a $\exists x\ P(x)$ on peut en extraire un nouveau a tel que $P(a)$.

Règle de construction du "existe"

Si on a $P(a)$ alors on a prouvé que $\exists a\ P(a)$.

2. Une formule est valide si sa table de vérité ne contient que des « vrais »

Remarque 2.7. (1) Les quantificateurs sont souvent implicites dans le discours mathématique. Le mot « un » peut à la fois désigner une quantification universelle ou existentielle. Par exemple la phrase « Un réel positif possède une racine carrée. » signifie :

$$\forall x\ x \in \mathbb{R}^+ \Rightarrow \exists y\ y^2 = x$$

Dans les exercices, vous allez voir des exemples de « traduction » entre le français et le langage formalisé. Il faut s'y habituer, car vos professeurs vont évidemment s'exprimer en français (pour gagner énormément de temps...).

(2) Les quantificateurs changent le statut des variables : ils transforment les variables libres en variables liées. On dit que ce sont des lieurs ou mutificateurs.

Exemple 2.8 – Voyons la règle d'utilisation du $\forall$ sur l'exemple $\forall x\ x \in \mathbb{R}^+ \Rightarrow \exists y\ y^2 = x$, et surtout, voyons comment elle se combine avec la table de vérité de $\implies$ (qui est une chose troublante, au début). La règle nous dit : soit x un réel quelconque, alors la proposition $x \in \mathbb{R}^+ \implies \exists y\, y^2 = x$ est vraie. Si A est vraie, et $A \implies B$ est vraie, alors dans la table de vérité nous voyons que B est vraie ; et donc, conformément à ce qu'on attend, si $x \in \mathbb{R}^+$, alors on a bien $\exists y\, y^2 = x$.

Mais si x n'est pas dans $\mathbb{R}^+$, autrement dit si $x < 0$, la règle nous dit que $x \in \mathbb{R}^+ \implies \exists y\, y^2 = x$ est vraie tout de même... On voit pourquoi la table de vérité nous dit que, lorsque A est fausse, $A \implies B$ est vraie indépendamment de B.

De manière plus simple : les cas pour lesquels $A(x)$ est fausse ne doivent pas avoir d'incidence sur la véracité de $\forall x\, A(x) \implies B(x)$. Ou encore : si on avait choisi la table de vérité de $A \implies B$ autrement, nous n'aurions pas droit à la règle de logique ci-dessus !

2.5.1. *Quantificateurs et négation.*

$$\begin{aligned} \neg(\forall x\ P(x)) &\equiv \exists x\ \neg P(x) \\ \neg(\exists x\ P(x)) &\equiv \forall x\ \neg P(x) \end{aligned}$$

2.5.2. *Ordre des quantificateurs.*

$$\forall x\ \forall y\ P(x,y) \equiv \forall y\ \forall x\ P(x,y)$$

$$\exists x\ \exists y\ P(x,y) \equiv \exists y\ \exists x\ P(x,y)$$

Attention, on ne peut pas permuter $\forall$ et $\exists$. Par exemple, en parlant d'élements de $\mathbb{N}$, la formule $\exists n \forall m\ n \geq m$ est fausse, mais $\forall m \exists n\ n \geq m$ est vraie.

2.5.3. *Quantificateurs relativisés.* On écrit parfois $\forall x \in \mathbb{N}\ P(x)$ ou $\exists x \in \mathbb{N}; P(x)$. On appelle cette notation un quantificateur relativisé. $\forall x \in \mathbb{N}\ P(x)$ signifie $\forall x\ x \in \mathbb{N} \Rightarrow P(x)$. $\exists x \in \mathbb{N}; P(x)$ signifie $\exists x\ x \in \mathbb{N} \wedge P(x)$. On remarquera que le connecteur implicite n'est pas le même dans les deux cas. Il s'agit d'une implication dans le cas universel et d'une conjonction dans le cas existentiel.

2.5.4. *Il existe un et un seul.* On note $\exists! x$ le quantificateur qui signifie : il existe un et un seul x tel que $P(x)$.

Négation d'un « existe un et un seul » :

3. Les règles de raisonnement

3.1. **Règles de raisonnement atomiques.** Dans cette partie nous allons lister les règles de logique dite classique. Pour chaque connecteur logique il y a une ou plusieurs règles d'utilisation du connecteur [3], et une ou plusieurs règles pour prouver une proposition mentionnant le connecteur [4].

3.1.1. *Implication.* Commençons par l'implication, qui comporte deux règles : une règle pour prouver une implication et une règle pour utiliser une implication.

Règle de construction d'une implication

Pour prouver une implication de la forme $A \Rightarrow B$, il suffit de supposer que l'on a A et de prouver B.

Cette règle est la règle que l'on utilise presque systématiquement au début d'une démonstration.

Règle d'utilisation d'une implication (modus ponens)

Si on sait d'une part que $A \Rightarrow B$ et d'autre part que A, alors on peut en déduire B.

Le modus ponens est la règle que l'on utilise quand on applique un théorème $(A \Rightarrow B)$ à une hypothèse A pour prouver B.

L'implication étant omni-présente en mathématique il existe plusieurs façons de décrire une implication : $A \Rightarrow B$ s'écrit aussi :

- si A alors B
- A seulement si B
- Pour que B il suffit que A
- Pour que A il faut que B
- A est une condition suffisante pour B
- B est une condition nécessaire pour A

Remarque 3.1. Attention ! il ne faut pas confondre implication logique et rapport de cause à effets. Par exemple, l'assertion "Quand les gens mangent beaucoup de glace alors ils sont nombreux à se noyer" est probablement statistiquement valide, mais il ne faut pas en conclure que c'est à cause des glaces qu'on a mangé qu'on se noie. Il se pourrait par exemple que "Quand il fait chaud, les gens mangent beaucoup de glace", et que "Quand il fait chaud, les gens sont nombreux à se baigner", et que "Quand les gens sont nombreux à se baigner, ils sont nombreux à se noyer."

Exemple 3.2 – Voyons une démonstration de $\forall x \quad x \geq 0 \implies \exists y \quad x = y^2$, où x et y sont des réels. On utilise la règle du $\forall$, et donc on commence par écrire « Soit x un réel ». On doit maintenant prouver $x \geq 0 \implies \exists y \quad x = y^2$. On utilise la règle de construction de l'implication, et donc on écrit « Supposons que $x \geq 0$ », et maintenant on cherche à démontrer $\exists y \quad x = y^2$. Finissons, en supposant que l'on connait le fonctionnement du symbole $\sqrt{}$: on peut parler de $\sqrt{x}$ puisque $x \geq 0$, et $x = \sqrt{x}^2$, donc $x = y^2$ est vraie pour $y = \sqrt{x}$; d'après la règle de construction de $\exists$, on a bel et bien démontré $\exists y \quad x = y^2$, donc on a fini.

(Evidemment, il y a un peu de triche ici (voire beaucoup !) : vous connaissez le symbole $\sqrt{}$ et ses propriétés depuis un moment, mais au lycée, on ne vous a rien démontré... Il n'empêche que le schéma de démonstration est correct.)

3. En logique, on les appelle des règles d'élimination.
4. En logique, on les appelle des règles d'introduction

3.1.2. *Conjonction.* On appelle conjonction noté $A \wedge B$ le fait d'avoir à la fois une démonstration de A et une démonstration de B.

Règle de construction d'une conjonction

Pour prouver une conjonction de la forme $A \wedge B$, il suffit de prouver A d'une part et B d'autre part.

Règle d'utilisation d'une conjonction (à gauche)

Quand on sait que $A \wedge B$, on peut en déduire A.

Règle d'utilisation d'une conjonction (à droite)

Quand on sait que $A \wedge B$, on peut en déduire B.

3.1.3. *Disjonction.* La disjonction $A \vee B$ de deux propositions est une façon d'affirmer qu'au moins une de ces deux propositions est vraie. La disjonction est fausse uniquement quand les deux propositions sont fausses.

Pour démontrer une disjonction, nous avons les deux règles suivantes :

Règle de construction d'une disjonction (à gauche)

Pour prouver une disjonction de la forme $A \vee B$, il suffit de prouver A.

Règle de construction d'une disjonction (à gauche)

Pour prouver une disjonction de la forme $A \vee B$, il suffit de prouver B.

La règle d'utilisation d'une disjonction correspond au raisonnement par cas. L'utilisation de cette règle est facile à repérer dans un raisonnement mathématique car elle crée deux branches dans la structure de la preuve.

Règle d'utilisation d'une disjonction (raisonnement par cas)

Quand on sait que $A \vee B$ et que l'on veut prouver P, il suffit de prouver d'une part que en supposant A on peut prouver P et d'autre part en supposant B on peut prouver P.

3.1.4. *Négation.* Pour prouver une négation on peut utiliser la règle suivante :

Règle de construction d'une négation

Pour prouver $\neg A$ il suffit de supposer A et d'arriver à une contradiction.

Cette règle est souvent confondue avec le raisonnement par l'absurde (qui consiste à prouver A en supposant $\neg A$ et en arrivant à une contradiction (voir plus loin) car dans les deux cas on arrive à une contradiction à la fin. Mais il existe des logiques où l'on distingue les deux règles.

Si on pourrait définir la négation $\neg A$ par $A \Rightarrow \bot$, dans ce cas là la règle pour démontrer une négation correspond en fait à la règle pour démontrer une implication.

Inversement, pour utiliser une négation déjà prouvée, on peut la voir comme l'implication $A \Rightarrow \bot$ donc en utilisant la règle du modus-ponens on obtient la règle suivante (le mot contradiction voulant dire faux) :

Règle d'utilisation d'une négation

Pour arriver à une contradiction, il suffit de prouver A et $\neg A$.

3.1.5. *Faux.* Il y a une seule règle concernant le faux, c'est la règle d'utilisation qui dit que du faux on déduit tout. Comme dit précédemment quand on arrive à faux, on dit parfois qu'on arrive à une contradiction. Quand on procède à une raisonnement par cas, et que un des cas permet de prouver « faux », on dit parfois que ce cas est impossible.

On appelle parfois cette règle exfalso quod libet (du faux on déduit tout). Cette règle correspond bien au fait que dans la table de vérité de l'implication $A \Rightarrow B$, celle-ci est vrai quand A est faux, quelque soit la valeur de vérité de B.

Règle D'utilisation du faux

Si on sait que Faux alors on peut déduire P (quelque soit P).

3.1.6. *Équivalence.* Pour l'équivalence, nous avons les règles suivantes :

Règle de construction d'une équivalence

Pour prouver $A \iff B$ il suffit de prouver $A \Rightarrow B$ d'une part et $B \Rightarrow A$ d'autre part.

Attention, les débutants ont parfois tendance à oublier une ou l'autre des deux implications.

Règle d'utilisation de l'équivalence (gauche)

Si on a A et $A \iff B$ on peut en déduire B.

Règle d'utilisation de l'équivalence (droite)

Si on a B et $A \iff B$ on peut en déduire A.

Comme pour la négation, on aurait aussi pu définir l'équivalence à l'aide de la conjonction et de l'implication et en déduire les règles ci-dessus.

$$A \iff B \underset{\text{def}}{\Leftrightarrow} (A \Rightarrow B) \wedge (B \Rightarrow A)$$

$A \iff B$ signifie la même chose que :
— A si et seulement si B.
— A est une condition nécessaire et suffisante pour B.
— Pour que A, il faut et il suffit que B.

3.1.7. *Règle de raisonnement par l'absurde.*

Règle de raisonnement par l'absurde

Pour montrer une proposition A, il suffit en supposant $\neg A$ d'arriver à une contradiction.

Cette règle de raisonnement est équivalente à supposer que $A \vee \neg A$ pour n'importe quelle proposition A.

Cette règle permet aussi quand on distingue deux cas $A \vee B$, de supposer $\neg A$ quand on étudie le cas B.

3.1.8. *Règle de l'axiome.* Pour compléter notre présentation des règles, nous sommes obligés de donner une dernière règle, qui parait évidente, mais qui est tout de même nécessaire pour terminer les démonstrations :

Règle de l'axiome

Pour montrer une proposition A, il n'y a rien à faire si A est une hypothèse locale ou un axiome.

3.2. **Exemple.** Maintenant que nous avons vu les règles atomique, reprenons notre exemple du théorème 2.3. Voici la démonstration reformulée en exprimant les proposition par des formules logiques et en spécifiant à chaque étape du raisonnement la règle utilisée. Nous allons voir que c'est très fastidieux de donner vraiment toutes les étapes, c'est pourquoi nous le feront qu'une fois dans ce cours. Ensuite, nous nous autoriseront d'autres règles logiques permettant des raccourcis.

Commençons par formaliser l'énoncé et les lemmes utilisés dans la preuve.

On suppose que :

$$\forall ab,\ b \neq 0 \Rightarrow \exists qr,\ a = bq + r \ \wedge\ r < b \tag{1}$$

$$\forall x,\ x \in \mathbb{N} \wedge 0 \leq x < 2 \Rightarrow x = 0 \vee x = 1 \tag{2}$$

Nous voulons montrer que :

$$\forall n, Impair(n) \Rightarrow \exists k, n = 2k + 1$$

Soit n, montrons que :

$$Impair(n) \Rightarrow \exists k, n = 2k + 1$$

Supposons que $Impair(n)$ montrons que :

$$\exists k, n = 2k + 1$$

D'après 1 et la règle d'utilisation du « pour tout » :

$$\forall b,\ b \neq 0 \Rightarrow \exists qr,\ n = bq + r \ \wedge\ r < b$$

D'où d'après la règle d'utilisation du « pour tout » :

$$2 \neq 0 \Rightarrow \exists qr,\ n = 2q + r \ \wedge\ 0 \leq r < 2$$

D'où d'après la règle d'utilisation de l'implication (modus ponens) et le fait que $2 \neq 0$:

$$\exists qr,\ n = 2q + r \ \wedge\ 0 \leq r < 2$$

D'où d'après la règle d'utilisation du « existe », on a q tel que :

$$\exists r,\ n = 2q + r \ \wedge\ 0 \leq r < 2$$

D'où d'après la règle d'utilisation du « existe », on a r tel que :

$$n = 2q + r \ \wedge\ 0 \leq r < 2$$

D'où d'après la règle d'utilisation de la conjonction on a :

$$n = 2q + r \tag{3}$$

et

$$0 \leq r < 2 \tag{4}$$

D'après 2 et la règle d'utilisation du « pour tout » on a :

$$x \in \mathbb{N} \wedge 0 \leq r < 2 \Rightarrow r = 0 \vee r = 1 \tag{5}$$

D'après la règle de construction de la conjonction, on a :

$$x \in \mathbb{N} \wedge 0 \leq r < 2 \tag{6}$$

D'où d'après la règle d'utilisation de l'implication on a :

$$r = 0 \vee r = 1 \tag{7}$$

D'où d'après la règle d'utilisation d'un disjonction, il suffit de démontrer $\exists k, n = 2k + 1$ en supposant $r = 0$ d'une part et $r = 1$ d'autre part.

— Supposons $r = 0$. D'après la règle de substitution, et 3 on a :

$$n = 2k$$

Donc d'après la règle de construction du « existe » :

$$\exists k,\ n = 2k$$

D'où $Pair(n)$ d'après la définition de Pair. On a supposé $Impair(n)$ donc par définition d'Impair, on a $\neg Pair(n)$. D'après la règle d'utilisation d'une négation on a faux. D'après la règle d'utilisation du faux on a tout ce qu'on veut, en particulier $\exists k, n = 2k + 1$.

— Supposons $r = 1$.

D'après la règle de substitution, et 3 on a :

$$n = 2k + 1$$

D'après la règle de construction du « existe » on a bien ce qu'on voulait montrer :

$$\exists k, n = 2k + 1$$

3.3. **Règles dérivées.** Dans la pratique mathématique, on ne revient pas toujours aux règles de base données précédemment, on utilise aussi d'autres règles qui peuvent se déduire des règles précédentes, que l'on peut voir comme l'application de plusieurs règles en une seule fois.

3.3.1. *Règles combinées.*

Règle raisonnement vers l'avant

Si on a une formule de la forme :

$$\forall\, x_1 \ldots x_n\ (H_1(x_1, \ldots, x_n) \wedge \ldots \wedge H_m(x_1, \ldots, x_n) \Rightarrow$$
$$\exists\, y_1 \ldots y_k\ P_1(x_1, \ldots, x_n, y_1, \ldots, y_n) \wedge \ldots \wedge P_l(x_1, \ldots, x_n, y_1, \ldots, y_k))$$

et que l'on a $H_1(a_1, \ldots, a_n)$, et $\ldots$ et $H_m(a_1, \ldots, a_n)$ alors on peut en déduire que l'on a des $b_1 \ldots b_k$ tels que :
$P_1(a_1, \ldots, a_n, b_1, \ldots, b_n)$ et $\ldots$ et $P_l(a_1, \ldots, a_n, b_1, \ldots, b_k)$.

Cette règle correspond à l'utilisation combinée des règles d'utilisation des quantificateurs existentiels et universels, de la conjonction et de l'implication. Remarquons qu'en revanche, la règle d'utilisation de la disjonction (distinction de cas) reste en général explicite, puisqu'elle crée une rupture dans la structure de la preuve.

Par exemple, si on sait que :

$$\forall x\ (P(x) \Rightarrow \exists y\ Q(y) \wedge R(y))$$

et que l'on sait que $P(a)$ alors on peut en déduire en une étape que l'on a un b tel que $Q(b)$ et $R(b)$.

En Edukera cette règle se déclenche quand on clique sur une hypothèse et sur $\exists$.

Règle cloture d'une branche de preuve

Si on doit montrer une formule de la forme :

$$\exists x_1\ \ldots\ x_n P(x_1, \ldots, x_n) \vee Q(x_1, \ldots, x_n)$$

et que l'on a $P(u_1, \ldots, u_n)$ ou bien $Q(u_1, \ldots, u_n)$ alors cette branche de preuve est terminée.

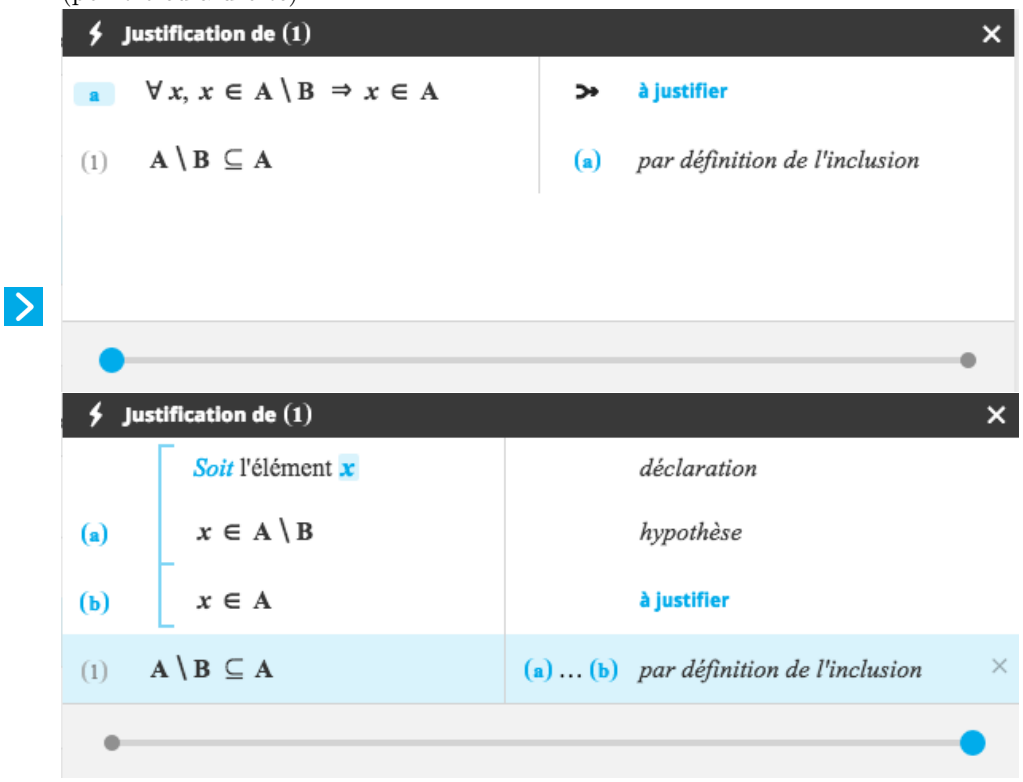
Règle de création de portée

Si on doit montrer une formule de la forme :

$$\forall x_1\ \ldots\ x_n P_1(x_1, \ldots, x_n) \wedge \ldots \wedge P_k(x_1, \ldots, x_n) \Rightarrow Q(x_1, \ldots, x_n)$$

Alors on peut supposer que l'on a des x_1, x_n qui vérifient les hypothèses P_i et il reste à montrer Q.

En Edukera cette règle s'appelle la règle de création de portée. Par exemple si on doit prouver qu'un ensemble est inclus dans un autre, en cliquant sur $=_{def}$ on peut simplement déplier la définition de l'inclusion (point bleu à gauche), on obtient alors la formule avec les « pour tout » et les « implications » ou bien déclencher dans la foulée la règle de création de portée (point bleu à droite).



3.3.2. *Règles de substitution.*

Règle substitution

Si on sait que deux formules A et B sont équivalentes alors on peut substituer A par B dans une preuve.

Voici quelques équivalences usuelles :

Non-contradiction	$A \wedge \neg A$	$\equiv$	$\bot$
Tiers-exclu	$A \vee \neg A$	$\equiv$	$\top$
Double négation	$\neg\neg A$	$\equiv$	A
Définition négation	$\neg A$	$\equiv$	$A \Rightarrow \bot$
Idempotence de ou	$A \vee A$	$\equiv$	A
Idempotence de et	$A \wedge A$	$\equiv$	A
Lois de de Morgan	$\neg(A \wedge B)$	$\equiv$	$\neg A \ \vee \ \neg B$
	$\neg(A \vee B)$	$\equiv$	$\neg A \ \wedge \ \neg B$
Associativité du ou	$(A \vee B) \vee C$	$\equiv$	$A \vee (B \vee C)$
Associativité du et	$(A \wedge B) \wedge C$	$\equiv$	$A \wedge (B \wedge C)$
Définition de l'implication	$A \Rightarrow B$	$\equiv$	$\neg A \vee B$
Négation de l'implication	$\neg(A \Rightarrow B)$	$\equiv$	$A \wedge \neg B$
Commutativité du ou	$A \vee B$	$\equiv$	$B \vee A$
Commutativité du et	$A \wedge B$	$\equiv$	$B \wedge A$
Définition de l'équivalence	$A \iff B$	$\equiv$	$(A \Rightarrow B) \wedge (B \Rightarrow A)$
Curryfication	$A \Rightarrow (B \Rightarrow C)$	$\equiv$	$(A \wedge B) \Rightarrow C$
Contraposée	$A \Rightarrow B$	$\equiv$	$\neg B \Rightarrow \neg A$
Transitivité de l'implication	$(A \Rightarrow B \wedge B \Rightarrow C)$	$\Rightarrow$	$(A \Rightarrow C)$
Modus ponens	$(A \wedge (A \Rightarrow B))$	$\Rightarrow$	B
Modus tolens	$((A \Rightarrow B) \wedge \neg B)$	$\Rightarrow$	$\neg A$

3.3.3. *Tiers exclu.* Le principe du tiers-exclu peut se déduire à l'aide du raisonnement par l'absurde. Il énonce que $A \vee \neg A$ pour toute formule A.

Exemple de preuve utilisant le tiers-exclu :

EXEMPLE 3.3 – Il existe deux nombres irrationnels tel que l'un à la puissance l'autre est rationnel.

$$\exists xy \ x \notin \mathbb{Q} \wedge y \notin \mathbb{Q} \wedge x^y \in \mathbb{Q}$$

Démonstration. D'après le principe du tiers-exclu, on a $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q} \vee \neg\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$.

Cas $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$**:** On choisit $x = \sqrt{2}$ et $y = \sqrt{2}$.

Cas $\sqrt{2}^{\sqrt{2}} \notin \mathbb{Q}$ **:** On choisit $x = \sqrt{2}^{\sqrt{2}}$ et $y = \sqrt{2}$.

On a $x^y = \left(\sqrt{2}^{\sqrt{2}}\right)^{\sqrt{2}} = \sqrt{2}^{\sqrt{2}\sqrt{2}} = \sqrt{2}^2 = 2$. D'où $x^y \in \mathbb{Q}$.

□

3.3.4. *Contraposée.* En utilisant le raisonnement par l'absurde on peut en déduire le raisonnement par contraposée.

Pour montrer $A \Rightarrow B$, il suffit de montrer $\neg B \Rightarrow \neg A$.

En Edukera la contraposée est disponible en cliquant sur :

Attention, il ne faut pas confondre la contraposée et la réciproque. La contraposée de $A \Rightarrow B$ est $\neg B \Rightarrow \neg A$, la réciproque est $B \Rightarrow A$.

3.3.5. *Sans perte de généralité.*

EXERCICE 3.4 – Réécrire la preuve suivante sans utiliser la règle "sans perte de généralité".

3.3.6. *Tautologie.* On dit qu'une formule est une tautologie quand sa validité ne dépend pas de la signification des symboles de fonctions et de prédicats qui apparaissent dans la formules.

Par exemple $1 + 1 = 2$ n'est pas une tautologie, car la validité de cette formule dépend de la définition de 1, $+$, 2 et $=$. En revanche, $1 = 0 \Rightarrow 1 = 0$ est une tautologie, car quelque soit la signification de 1, 0 et $=$, la formule est toujours valide. Autre exemple, $A \vee \neg A$ est une tautologie.

Pour s'assurer qu'une formule est une tautologie on peut faire la table de vérité. Si celle si ne contient que des vrai c'est une tautologie.

3.4. **Raisonnement/rédaction par chaînage avant ou chaînage arrière.** La raisonnement par chaînage avant consiste à partir des hypothèses pour aller jusqu'à la conclusion. Le raisonnement par chaînage arrière consiste à partir de la conclusion pour aller jusqu'aux hypothèses. En pratique en mathématiques, on rédige les démonstration en général en chaînage avant, mais il arrive parfois d'utiliser aussi le chaînage arrière. L'expression utilisée est alors « Il suffit de montrer que. »

Mais pour trouver une démonstration, on utilise souvent le chaînage arrière. Par exemple, si je dois prouver que deux droites sont parallèles, je peux me demander quels sont les théorèmes qui permettent d'arriver à cette conclusion, et me dire il suffit que je prouve que ce sont les côté d'un parallélogramme...

En Edukera, on réalise un raisonnement par chaînage arrière quand on clique sur le bouton « justifier ».
Mais en Edukera, comme la copie est électronique, on n'est pas nécessaire de rédiger la démonstration en utilisant la formule « Il suffit de montrer que... » la démonstration est présentée dans un ordre qui n'est pas celui dans lequel on a construit la preuve. En effet, Edukera complète le texte de la preuve, un peu comme un brouillon au fur et à mesure. Tant que la démonstration n'est pas terminée le texte comporte des trous : des propositions qui restent à justifier. La démonstration est terminée quand on a « bouché » tous les trous, qu'il ne reste plus rien à justifier.
En revanche, on réalise un raisonnement par chaînage avant quand on clique sur une hypothèse.
Les règles logiques disponibles en chaînage avant sont accompagnées d'une flèche vers le bas. Les règles logique disponibles en chaînage arrière sont accompagnées d'une flèche vers le haut.

3.5. **Technique de résolution.**

3.6. **Raisonnement par analyse-synthèse.** Lorsque l'on souhaite montrer l'existence d'un objet (ou de plusieurs objets) vérifiant un certain problème, il est parfois plus simple de procéder par analyse-synthèse. Tout d'abord on suppose que l'élément cherché existe et on cherche la forme qu'il doit avoir pour vérifier ce que l'on souhaite (analyse).

Ensuite on montre que l'élément (ou les éléments) ainsi construit existe bel et bien et qu'il vérifie le problème initial (synthèse).

La phase d'analyse ne fait pas partie de la preuve proprement dite, il s'agit d'une technique pour obtenir les témoins pour les existentiels.

Exemple 3.5 – Soit f une fonction de $\mathbb{R}$ dans $\mathbb{R}$, montrons que f s'écrit comme somme d'une fonction paire et d'une fonction impaire.

$$\forall f : \mathbb{R} \to \mathbb{R} \; \exists \, (P\, Q : \mathbb{R} \to \mathbb{R}) \;\; f = P + Q \wedge Paire(P) \wedge Impaire(Q)$$

Démonstration. Analyse : supposons qu'il existe P une fonction paire et Q une fonction impaire telles que $f = P + Q$. On aura alors $f(-x) = P(-x) + Q(-x) = P(x) - Q(x)$. D'où $P(x) = \frac{f(x)+f(-x)}{2}$ et $Q(x) = \frac{f(x)-f(-x)}{2}$.

Synthèse : P et Q ainsi définies sont bien paire et impaire respectivement. De plus $(P + Q)(x) = f(x)$. □

4. Axiomes : les ensembles

Nous l'avons dit en ouverture de ce document : en cherchant à formaliser ce que « faire des mathématiques » signifie, nous avons dégagé deux aspects importants. Primo, il faut se conformer à certaines règles de logique, que nous avons maintenant largement décrites (et qui sont utiles en informatique aussi). Secondo, il faut se donner un point de départ, sous forme d'axiomes. Nous en venons à ce point.

Nous allons décrire la théorie des ensembles, dans les grandes lignes. Le faire de manière très rigoureuse et formelle est difficile (allez voir la page Wikipedia sur « Zermelo-Fraenkel », vous aurez une idée !). Aussi, nous n'allons pas dresser une (très) longue liste de propriétés pour en faire nos axiomes. A la place, nous allons discuter de manière informelle de ce qu'est un ensemble (et votre compréhension intuitive du mot « ensemble » est mise à contribution).

Néanmoins, le fait est que tout ce qui suit, dans ce §4, est une collection d'axiomes, même si on ne le rappelle pas constamment.

4.1. **Ensembles et appartenance.** Les objets mathématiques peuvent être rangés dans des ensembles, que l'on écrit avec des accolades. Par exemple,

$$E = \{1, 2, 3\} \quad \text{et} \quad F = \{19, 11\}$$

sont des ensembles. On note $x \in X$ pour signifier que x appartient à X, et dans le cas contraire on emploie le symbole $\notin$; par exemple, on a $2 \in E$ et $3 \notin F$.

Un ensemble ne comprend jamais de « répétitions », et n'est pas ordonné : ainsi

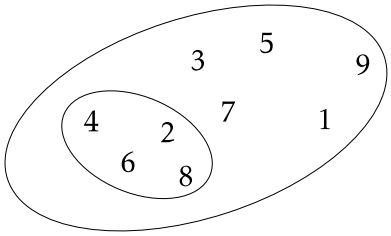
$$\{2, 2, 2, 3, 3\} = \{2, 3\} \quad \text{et} \quad \{3, 2, 1\} = \{1, 2, 3\}\,.$$

Il existe bien sûr des ensembles infinis, comme l'ensemble $\mathbb{N}$ des nombres entiers, dont nous reparlerons plus loin. Il y a également un ensemble vide, qui ne contient aucun élément : on le note $\emptyset$ ou, plus rarement, $\{\}$.

Lorsque tous les éléments d'un ensemble A sont aussi dans l'ensemble B, on dit que A est une partie de B, ou qu'il est inclus dans B, et on note $A \subset B$ (parfois $A \subseteq B$; nous ne ferons pas de différence). Par exemple

$$\{2, 4, 6, 8\} \subset \{1, 2, 3, 4, 5, 6, 7, 8, 9\}\,.$$

Les ensembles sont souvent dessinés comme des bulles, et pour représenter l'inclusion on place ces bulles les unes dans les autres, comme ci-dessous :



Remarque 4.1. Voici une chose qui paraît évidente : deux ensembles A et B sont égaux si et seulement si « appartenir à A » est équivalent à « appartenir à B » ; plus formellement, on dira que $A = B$ est équivalent à

$$\forall x \quad x \in A \iff x \in B.$$

(Dans une description vraiment axiomatique et formelle de la théorie des ensembles, il faudrait énoncer ça comme une « définition » de l'égalité !)

De manière similaire, $A \subset B$ signifie

$$\forall x \quad x \in A \implies x \in B.$$

De ces remarques très simples, on retient une chose : $A = B$ équivaut à $(A \subset B) \wedge (B \subset A)$, c'est maintenant bien clair. Or, ce qui peut apparaître comme une évidence est en fait à retenir : dans un exercice où l'on demande de montrer l'égalité de deux ensembles, disons $A = B$, il est très souvent utile de procéder en deux temps, et de montrer les deux inclusions $A \subset B$ et $B \subset A$ l'une après l'autre.

Revenons aux généralités. Fixant B, on peut considérer l'ensemble $\mathscr{P}(B)$ dont les éléments sont toutes les parties de B ; ainsi dans le cas où $B = \{1, 2, 3\}$, on a

$$\mathscr{P}(B) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}.$$

(On n'oublie ni la partie vide, ni B lui-même.)

Enfin, étant donnés deux ensembles A et B, on peut former leur produit cartésien noté $A \times B$, dont les éléments sont les paires (a, b) avec $a \in A$ et $b \in B$. Lorsque $A = \{1, 3\}$ et $B = \{2, 4, 6\}$ par exemple, on a

$$A \times B = \{(1, 2),\ (1, 4),\ (1, 6),\ (3, 2),\ (3, 4),\ (3, 6)\}.$$

On notera que pour les paires, l'ordre est important : ainsi l'élément $(1, 2)$ de $\mathbb{N} \times \mathbb{N}$ est différent de l'élément $(2, 1)$.

On écrit volontiers A^2 au lieu de $A \times A$. Sur le même modèle on peut considérer, pour tout entier $n \geq 1$, l'ensemble noté A^n dont les éléments sont de la forme $(a_1, a_2, \ldots, a_n)$ avec chaque $a_i \in A$ (on parle des « n-uplets d'éléments de A », et pour $n = 3, 4, 5$ on parle de triplets, quadruplets, quintuplets).

4.2. **Quelques constructions.** Lorsqu'on dispose d'un ensemble E, on peut s'intéresser aux éléments de E qui vérifient une certaine propriété P. Ceux-ci forment à nouveau un ensemble, que l'on note ainsi :

$$\{x \in E \mid P(x)\}.$$

(Parfois le $\mid$ est remplacé par deux points, ou par l'expression complète « tels que ». Il y a de nombreuses variantes et il faut s'habituer à des notations qui changent de temps en temps, en général pour éviter les lourdeurs.)

Par exemple, supposons que $A \subset E$. Alors le complémentaire de A dans E est par définition

$$\{x \in E \mid x \notin A\}.$$

On le note généralement $E - A$ ou $E \smallsetminus A$, parfois $\complement_E A$, et parfois même $\complement A$ ou A^c lorsque E est sous-entendu.

Autre exemple, si A et B sont deux parties de E, alors leur intersection est

$$A \cap B = \{x \in E \mid x \in A \wedge x \in B\},$$

leur union est

$$A \cup B = \{x \in E \mid x \in A \vee x \in B\}.$$

Exemple 4.2 – Prenons $E = \mathbb{N} \times \mathbb{N}$, puis

$$A = \{(n, m) \in \mathbb{N} \times \mathbb{N} \mid n = 0\},$$

et enfin

$$B = \{(n, m) \in \mathbb{N} \times \mathbb{N} \mid m = 0\}.$$

Alors $A \cap B = \{(0, 0)\}$. On peut également écrire

$$A \cup B = \{(n, m) \in \mathbb{N} \times \mathbb{N} \mid nm = 0\}.$$

Note : en pratique, on écrirait plutôt $A = \{(0,m) \in \mathbb{N} \times \mathbb{N}\}$ ou encore $A = \{(0,m) \mid m \in \mathbb{N}\}$, l'essentiel étant de se faire comprendre.

Il est très important de comprendre que la lettre x qui est employée ci-dessus dans la description des ensembles est muette, et peut être remplacée par n'importe quelle autre : on obtient rigoureusement les mêmes ensembles. Par exemple si

$$A = \{x \in \mathbb{N} \mid \text{il existe } y \in \mathbb{N} \text{ tel que } x = 2y\},$$

et si

$$B = \{a \in \mathbb{N} \mid \text{il existe } b \in \mathbb{N} \text{ tel que } a = 2b\},$$

alors $A = B =$ les nombres entiers pairs.

On ne peut pas utiliser tout et n'importe quoi pour décrire les ensembles. Pour se convaincre que les propriétés P comme ci-dessus ne peuvent pas être complètement arbitraires, voir l'exemple suivant.

Exemple 4.3 – Pour un entier n, considérons la propriété « n ne peut pas être décrit en moins de 16 mots ». Appelons cette propriété $P(n)$, et soit

$$A = \{n \in \mathbb{N} \mid P(n)\}.$$

Les mots de la langue française sont en nombre fini, donc en 16 mots on ne peut décrire qu'un nombre fini de nombres. Ainsi, A est infini et en particulier, non-vide. Soit alors a le plus petit élément de A. Ce nombre est « le plus petit nombre qui ne peut pas être décrit en moins de 16 mots ». On vient tout juste de décrire a en 15 mots !

C'est absurde. Et pour cause, la propriété $P(n)$ ne fait pas partie des propriétés mathématiques acceptables.

L'axiome selon lequel $\{x \in E \mid P(x)\}$ est un ensemble lorsque E est un ensemble peut paraître anodin. En réalité il est bien plus fin qu'on pourrait le croire.

Exemple 4.4 – Notre deuxième exemple utilise pour $P(x)$ la propriété « $x \notin x$ ». Celle-ci est parfaitement acceptable. C'est sa signification intuitive proche de zéro qui donne un parfum de paradoxe au raisonnement suivant, pourtant correct.

Montrons la chose suivante : pour tout ensemble E, il existe un ensemble A tel que $A \notin E$. En effet, soit

$$A = \{x \in E \mid x \notin x\}.$$

Si on avait $A \in E$, alors on constaterait que $A \in A$ équivaut à $A \notin A$, par définition. C'est absurde, donc $A \notin E$.

On énonce souvent ce résultat sous la forme suivante : il n'existe pas d'ensemble de tous les ensembles. Nous venons bien de le démontrer. S'il est tentant d'écrire quelque chose comme $U = \{x \mid x \text{ est un ensemble}\}$ pour essayer de le définir malgré tout, on se rend compte que cette expression n'est pas de la forme $\{x \in E \mid P(x)\}$, et donc ne désigne pas un ensemble. La présence de l'ensemble E pour « chapeauter » les x est essentielle.

Ceux que ça intéresse peuvent essayer « paradoxe de Russell » dans un moteur de recherche.

4.3. **Les ensembles en informatique.**

Voyons quelques commandes Python pour travailler avec les ensembles.

```
>>> a = [1,4,2,7,1,9,0,3,4,6,6,6,8,3]
>>> set(a)
{0, 1, 2, 3, 4, 6, 7, 8, 9}
```

« Set » est employé pour « ensemble », dans les mathématiques anglosaxones, et voici pourquoi la commande Python porte ce nom. Elle retire les doublons dans une liste, en gros (ici l'affichage des nombres a été fait dans l'ordre, mais c'est

uniquement visuel, il n'y a pas d'ordre dans l'objet `set(a)` ; essayez `set(a)[0]` pour obtenir le premier élément, et vous obtiendrez une erreur).

Certains langages de programmation autorisent aussi à définir un ensemble (ou une liste) en utilisant un « schéma de compréhension », c'est-à-dire une syntaxe proche de $\{x \in E \mid P(x)\}$. La propriété P est dans ce cas définie par une fonction qui associe à tout élément de a un booléen.

Par exemple en Python, on peut définir la liste des éléments de a qui vérifient la propriété $P(x) = x > 5$ de la manière suivantes :

```
>>> a= [1,4,2,7,1,9,0,3,4,6,6,6,8,3]
>>> b= [x for x in a if x > 5]; b
[7, 9, 6, 6, 6, 8]
>>> set(b)
{6, 7, 8, 9}
```

En informatique, le produit cartésien est très utilisé, tellement utilisé que ce concept apparaît sous différentes formes. Dans certains langages il apparaît explicitement : on peut construire le type correspondant au produit cartésien de types, les éléments étant représentés par des n-uplets comme en mathématiques. Par exemple en Ocaml le produit cartésien est noté `*` :

```
# let p=(2,"toto");;
val p : int * string = (2, "toto")
```

En Python :

```
>>> p = (2,"toto")
>>> type(p)
<type 'tuple'>
```

On doit souvent manipuler le produit cartésien de nombreux ensembles, il est alors pratique de donner un nom à chaque composante d'un produit cartésien. Par exemple quand on représente un point par le produit cartésien de deux nombres, on appelle abscisse le premier et ordonnée le second.

La plupart des langages de programmation permettent de définir un produit cartésien en donner un nom à chaque composante du produit cartésien. En C, çà s'appelle `struct`, en Python `namedtuple`, en Ocaml on appelle cela un enregistrement (*record* en anglais).

En Python :

```
>>> from collections import *
>>> Point = namedtuple('Point', ['x', 'y'])
>>> p = Point(2,3)
>>> p.x
2
>>> p.y
3
```

En Haskell :

```
type Point = {X :: Float, Y :: Float}
```

Dans les langages orientés objets, les éléments du produit cartésien sont appelés des attributs. Par exemple en Java :

```
public class Point {
    private final double x;
    private final double y;
    public Point(double x, double y) {
        this.x = x;
        this.y = y;
    }
    public double getX() { return x; }
    public double getY() { return y; }
}
```

5. Fonctions

5.1. **Qu'est-ce qu'une fonction ?** Étant donnés deux ensembles A et B, une fonction f de A vers B associe à tout élément $x \in A$ un élément $f(x) \in B$ et un seul. On peut traduire cette définition (un peu vague) en termes d'ensembles. Si l'on souhaite être extrêmement précis, on dira :

Définition 5.1 – Une fonction, ou application, est un triplet $f = (A, B, \Gamma)$ où :

(1) A est un ensemble, appelé le domaine de définition de f, ou parfois la source de f ;

(2) B est un ensemble, appelé le but de f ;

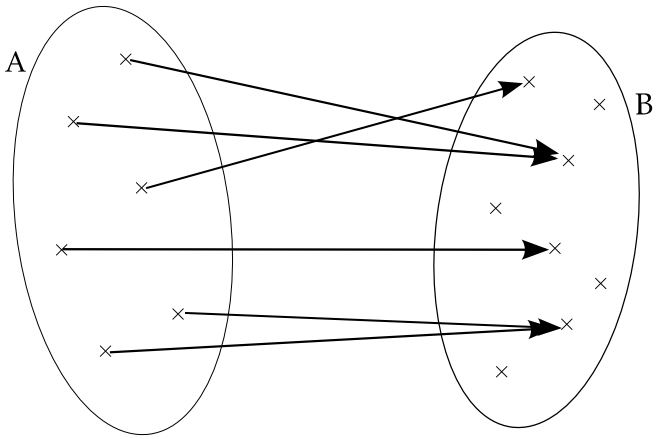
(3) Γ est un ensemble, qui est une partie de $A \times B$ et que l'on appelle le graphe de f, ayant la propriété suivante : pour chaque $x \in A$, il existe un unique $y \in B$ tel que $(x, y) \in \Gamma$. Ce y est noté $f(x)$.

On utilise la notation

$$f : A \longrightarrow B$$

pour indiquer que f est une fonction dont le domaine de définition est A et dont le but est B.

On représente typiquement une fonction $A \longrightarrow B$ de la manière suivante :



Chaque flèche sur ce dessin part d'un élément $x \in A$ et pointe sur $f(x)$. La caractéristique importante est que chaque point de A marque le début d'une flèche, et d'une seule.

Voyons quelques exemples.

Exemple 5.2 – Il y a une (et une seule) fonction $f\colon \mathbb{N} \to \mathbb{N}$ telle que $f(n) = 2n^2+1$. On utilise parfois la notation

$$\begin{array}{rccc} f\colon & \mathbb{N} & \longrightarrow & \mathbb{N} \\ & n & \mapsto & 2n^2+1 \end{array}$$

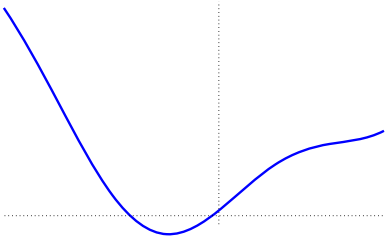
pour désigner cette fonction. C'est très souvent par des expressions, telles que $2n^2+1$, que l'on va définir les fonctions.

Ici le domaine de définition est $A = \mathbb{N}$, le but est $B = \mathbb{N}$, et le graphe de f est $\Gamma = \{(n, 2n^2+1) \mid n \in \mathbb{N}\}$.

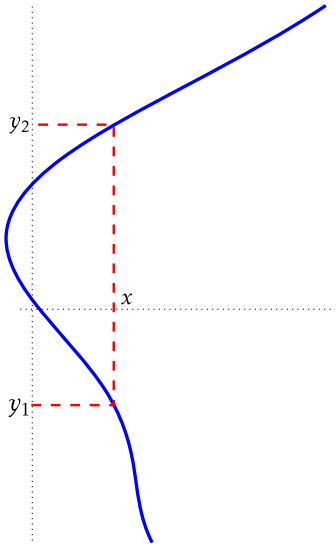
Exemple 5.3 – Voici une fonction dont la source et le but sont bien différents :

$$\begin{array}{rccc} f\colon & \mathbb{N} & \longrightarrow & \mathscr{P}(\mathbb{N}) \\ & n & \mapsto & \{n, n+1, n+2\}\,. \end{array}$$

Exemple 5.4 – Nous allons anticiper un peu et supposer que vous connaissez un minimum l'ensemble $\mathbb{R}$. On le représente par une droite, et $\mathbb{R}\times\mathbb{R}$ par un plan. Une fonction $A \to B$ avec $A \subset \mathbb{R}$ et $B \subset \mathbb{R}$ est donnée par son graphe, qui ressemble de près ou de loin à une courbe dans le plan. Par exemple la figure suivante représente un tel graphe.



La propriété caractéristique des graphes se voit bien sur le dessin : lorsque $A \subset \mathbb{R}$ et $B \subset \mathbb{R}$, une partie Γ de $A\times B$ est le graphe d'une fonction $A \to B$ si et seulement si *chaque droite verticale d'équation $x = a$ (avec $a \in A$) coupe Γ exactement en un point*. Si maintenant on fait subir une rotation à cette figure, obtient-on encore le graphe d'une fonction ?



La réponse est visiblement non : pour le x indiqué, il y a deux couples (x, y_1) et (x, y_2) qui appartiennent à la courbe. Ce n'est donc pas un graphe. Dans la suite nous allons étudier la propriété correspondante en utilisant cette fois des droites horizontales.

Concluons par quelques remarques. Soit $f\colon A \to B$ une fonction. L'ensemble A peut très bien être un produit cartésien, disons $A = A_1 \times A_2$, et dans ce cas un élément de A est une paire (a_1, a_2) avec $a_i \in A_i$. La plupart du temps, on va alors écrire $f(a_1, a_2)$ au lieu de $f\big((a_1, a_2)\big)$. Par exemple si f est la fonction $\mathbb{R}^2 \to \mathbb{R}$ définie par $(x, y) \mapsto \sin(xy)$, on va écrire $f(x, y) = \sin(xy)$, et non pas $f\big((x, y)\big)$. Non seulement on veut économiser sur les parenthèses, mais au delà de ça, on dira souvent que f est une fonction de plusieurs variables, pour insister sur le fait que f est définie sur un produit cartésien. Bien souvent, on veut étudier séparément la façon dont $f(x, y)$ dépend de x avec y fixé par exemple, ou bien le contraire (cf les dérivées partielles, que vous avez dû voir un peu en physique, ou ailleurs).

Remarque 5.5. Une autre façon de procéder est de voir une fonction à deux arguments comme une fonction qui prend le premier argument et renvoie une fonction qui attend le deuxième argument. La fonction f ci-dessus devient alors une fonction de l'ensemble $\mathbb{R}$ dans l'ensemble des fonctions $\mathbb{R} \to \mathbb{R}$, donc $f\colon \mathbb{R} \to (\mathbb{R} \to \mathbb{R})$, en écrivant $(\mathbb{R} \to \mathbb{R})$ pour *l'ensemble* de toutes les fonctions $\mathbb{R} \to \mathbb{R}$. Ce n'est pas une notation usuelle en mathématiques, mais en informatique, si !

En mathématiques et dans la plupart des langages de programmation, on utilise la première approche. Dans les langage de programmation fonctionnels comme par exemple OCaml, Haskell, ou Coq la tradition est d'utiliser la deuxième approche.

Les deux approches sont équivalentes au sens où l'on peut passer mécaniquement de l'une à l'autre. Le passage de la première à la seconde approche s'appelle la curryfication en l'honneur du logicien Haskell Curry.

Si f est une fonction de A^n dans B, on appelle n l'arité de la fonction ou son nombre d'arguments.

5.2. **Fonctions au sens informatique.** Attention, le mot fonction n'a pas le même sens en informatique et en mathématique.

Premièrement, toutes les fonctions au sens mathématique ne sont pas des fonctions au sens informatique.

En effet, dans les langages de programmation les fonctions sont définies de manière calculatoire. On définit une fonction en indiquant un programme pour calculer le résultat. Mathématiquement, on peut très bien définir une fonction sans donner la moindre indication sur comment calculer le résultat. Par exemple, on peut définir la fonction qui prend en argument une fonction $f : \mathbb{R} \to \mathbb{R}$ et renvoie le nombre de zéros de f si celui est fini et zéro sinon. Cette définition ne dit absolument pas comment calculer le résultat de la fonction. Ou encore, dans le même genre, considérer la fonction $\phi\colon \mathscr{P}(\mathbb{N}) \to \{0, 1\}$ qui associe à l'ensemble $A \subset \mathbb{N}$ l'entier 1 si A contient une infinité de nombres premiers, et 0 sinon. Mathématiquement, aucun problème avec cette définition, mais lorsque A est infini, la question de la valeur de $\phi(A)$ peut être tout à fait épineuse !

Pour certaines fonctions, on peut même démontrer qu'elles ne sont pas calculables. Un exemple célèbre est celui proposé par Alan Turing : la fonction qui à un programme associe 1 si le programme termine et 0 sinon n'est pas calculable.

Deuxièmement, il y a aussi des fonctions au sens informatique qui ne sont pas des fonctions au sens mathématique.

Voici quelques exemples :

- Les fonctions qui en plus de renvoyer un résultat, modifient l'état global de l'ordinateur, voire du monde :
  - Les fonctions comme `print` qui affichent quelque chose à l'écran.
  - Les fonctions qui permettent d'écrire un fichier sur le disque dur, ou d'envoyer un message sur le réseau, d'imprimer un document...
  - Les fonctions qui modifient le contenu d'une variable.

— Les fonctions dont le comportement ne dépend pas que des arguments.
  — Fonction dont le comportement dépend d'une variable globale.
  — Fonction qui lit un fichier sur le disque.
  — La fonction qui renvoie un nombre tiré au « hasard ».
  — La fonction qui renvoie la date du jour.

On dit que les fonctions qui ne font pas que retourner un résultat réalisent un effet de bord.

En informatique, on peut avoir des fonctions qui réalise un effet de bord mais qui ne retournent pas de résultat. Dans certains langages comme Pascal, on les appelle des procédures et il y a une syntaxe particulière pour les définir, mais dans la plupart des langages les procédures peuvent être définies comme les fonctions usuelles, le type de retour porte alors un non spécial. Par exemple en C ou en Java, on l'appelle `void`, en Ocaml on l'appelle `unit` et en Python `NoneType`.

```
>>> def f():
...     return;
...
>>> type(f())
<type 'NoneType'>
```

5.3. **Fonctions injectives.** (Nous revenons aux fonctions « mathématiques », et pas « informatiques ».)

Définition 5.6 – Soit $f : A \to B$ une fonction. Supposons que, pour tout choix de deux éléments distincts $x_1 \neq x_2$ dans l'ensemble A, on ait également $f(x_1) \neq f(x_2)$. On dit alors que f est injective, ou encore que f est une injection. En symboles, f est injective si

$$\forall x_1 \in A, \forall x_2 \in A, x_1 \neq x_2 \implies f(x_1) \neq f(x_2)\,.$$

Il existe bien des façons de reformuler ceci. Par exemple, f est injective si et seulement si l'égalité $f(x_1) = f(x_2)$ entraîne $x_1 = x_2$. C'est la règle de la contraposée ; en symboles, on écrit

$$\forall x_1 \in A, \forall x_2 \in A, f(x_1) = f(x_2) \implies x_1 = x_2\,.$$

Il est également bon de noter que f est injective si et seulement si l'équation

$$f(x) = b\,,$$

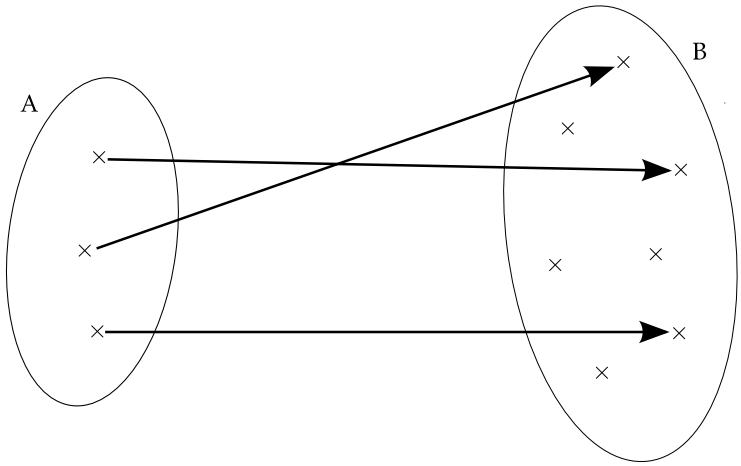
dont l'inconnue est $x \in A$ et qui comporte le paramètre $b \in B$, possède au maximum une solution.

Exemple 5.7 – La fonction $d : \mathbb{N} \to \mathbb{N}$ définie par $d(n) = 2n$, est injective : en effet si $2x_1 = 2x_2$, alors $x_1 = x_2$. L'équation $d(x) = b$ s'écrit $2x = b$; elle a une solution $x = \frac{b}{2}$ si b est pair, et aucune solution si b est impair.

Exemple 5.8 – La fonction $c : \mathbb{Z} \to \mathbb{N}$ définie par $c(n) = n^2$ n'est pas injective (ici $\mathbb{Z}$ est l'ensemble de tous les nombres entiers, positifs ou négatifs). En effet $c(n) = c(-n)$, de sorte que l'équation $c(x) = b$, qui s'écrit $x^2 = b$, peut posséder deux solutions, comme par exemple 2 et -2 qui sont solutions pour $b = 4$.

Exemple 5.9 – La fonction de l'ensemble des étudiants du groupe de TD vers l'ensemble des mots de la langue française, qui à un étuidant associe son prénom, n'est en général pas injective.

Voici comment on représente une fonction injective :

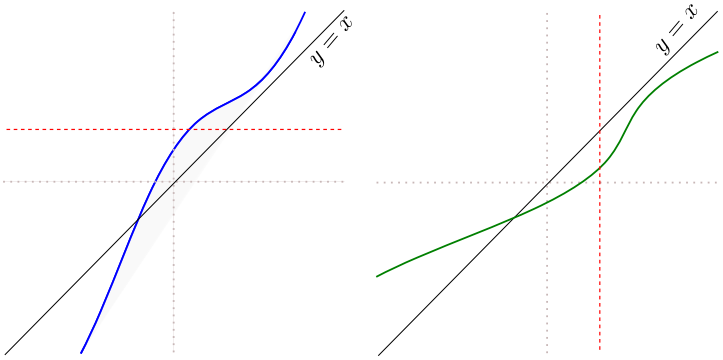


Cette fois-ci, les flèches pointent toutes vers des éléments différents.

EXEMPLE 5.10 – Revenons au cas particulier où $f : A \to B$ avec A et B des parties de $\mathbb{R}$. L'équation $f(x) = b$ possède une solution x lorsque le graphe de f comporte un point $(x, f(x))$ qui est également sur la droite horizontale d'équation $y = b$. La condition pour que f soit injective est donc que *les droites horizontales rencontrent le graphe de f en un point au maximum.*

Soit Γ le graphe de f. Faisons subir à ce graphe une symétrie par rapport à la droite d'équation $y = x$ (cette symétrie envoie le point (x, y) sur (y, x)). On obtient un ensemble Γ'. Lorsque f est injective, ce Γ' ne rencontre les droites verticales qu'en un point au plus. C'est-à-dire que Γ' est le graphe d'une fonction !

Cette discussion est illustrée sur la figure suivante.



Pour définir une fonction g dont le graphe serait Γ', il lui faudrait un ensemble de définition et un but. Les points de Γ' sont ceux de la forme $(f(x), x)$, et il est donc tentant de prendre pour ensemble de définition

$$\{f(x) \mid x \in A\} \subset B\,.$$

On va noter cet ensemble $f(A)$ dans la suite. Voyons ça.

5.4. **Fonctions surjectives et bijectives.**

DÉFINITION 5.11 – Soit $f : A \to B$ une fonction. On note $f(A)$, ou encore $\mathrm{Im}(f)$, l'ensemble

$$\{b \in B \mid \exists x \in A \text{ tel que } b = f(x)\}\,.$$

(En plus concis $f(A) = \{f(x) \mid x \in A\}$.) On dit que $f(A)$ est l'image de A par f.

Lorsque $f(A) = B$, on dit que f est surjective, ou encore que f est une surjection. En symboles, ceci signifie que

$$\forall b \in B, \exists a \in A, f(a) = b\,.$$

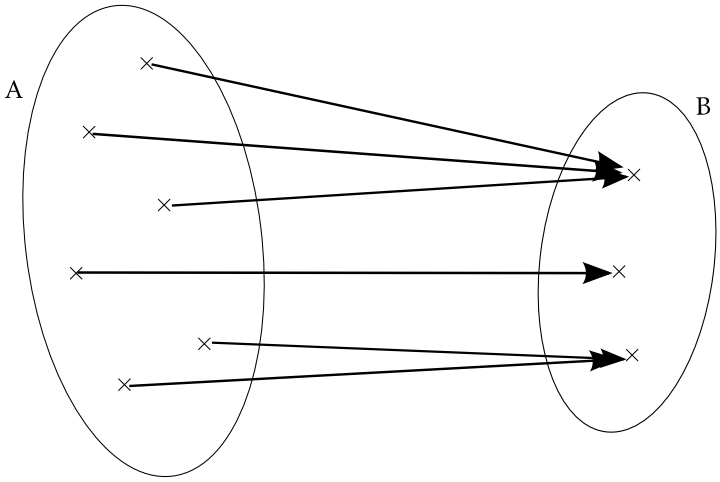
Ainsi f est surjective lorsque l'équation $f(x) = b$ possède au minimum une solution.

EXEMPLE 5.12 – La fonction $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ définie par $f(n, m) = n + m$ est surjective. En effet, si on se donne $b \in \mathbb{N}$, alors $f(b, 0) = b$. On a aussi $f(0, b) = b$, et même $f(1, b - 1) = b$, de sorte que f est loin d'être injective, par contre.

EXEMPLE 5.13 – La fonction $d : \mathbb{N} \to \mathbb{N}$ telle que $d(n) = 2n$ n'est pas surjective. En fait l'ensemble image $d(\mathbb{N})$ est l'ensemble des nombres pairs.

La fonction qui, à une carte de crédit en circulation, associe ses 19 chiffres (16 sur le devant et 3 derrière) n'est pas surjective : sinon, on pourrait inventer n'importe quelle combinaison de chiffres, et acheter tout ce qu'on veut. (Par ailleurs, il y a beaucoup moins de cartes de crédit en circulation que 10^{19}.)

Voici la représentation typique d'une fonction surjective :



Ici chaque élément de B est à l'extrémité d'au moins une flèche.

DÉFINITION 5.14 – Lorsqu'une fonction est à la fois injective et surjective, on dit qu'elle est bijective, ou encore que c'est une bijection.

Lorsque $f : A \to B$ est bijective, l'équation $f(x) = b$ possède une solution et une seule (comme précédemment, l'inconnue est x et b est un paramètre). Cette solution est notée $f^{-1}(b)$.

On obtient ainsi une fonction $f^{-1} : B \to A$, que l'on appelle la réciproque de f. On a alors :

PROPOSITION 5.15 – *Lorsque f est bijective, la fonction f^{-1} vérifie*

(1) $f^{-1}(f(a)) = a$ *pour* $a \in A$,

(2) $f(f^{-1}(b)) = b$ *pour* $b \in B$.

De plus, si on a une paire de fonctions $f\colon A \to B$ *et* $g\colon B \to A$ *telles que* $g(f(a)) = a$ *pour* $a \in A$ *et* $f(g(b)) = b$ *pour* $b \in B$, *alors* f *est une bijection et* $g = f^{-1}$.

Enfin, f^{-1} *est également une bijection lorsqu'elle existe, et*

$$(f^{-1})^{-1} = f\,.$$

Démonstration. L'élément $f^{-1}(b)$ vérifie (2) par définition. Pour le (1), étant donné a, soit $b = f(a)$; puisque f est injective, a est le seul élément de A qui vérifie cette équation, et c'est cet élément que l'on note $f^{-1}(b)$. Donc $a = f^{-1}(b) = f^{-1}(f(a))$.

Voyons la suite. Soient f et g comme dans la proposition. Si $f(a_1) = f(a_2)$, alors on a aussi $g(f(a_1)) = g(f(a_2))$, donc $a_1 = a_2$. Ainsi f est injective. De plus,

si $b \in B$ on a $b = f(g(b))$ donc b est bien dans l'image de f, ce qui montre que f est surjective. Finalement f est une bijection.

Partant de $f(f^{-1}(b)) = b = f(g(b))$, on tire $f^{-1}(b) = g(b)$, pour tout $b \in B$, car f est injective. Donc $f^{-1} = g$.

Par symétrie, on peut inverser les rôles de f et de g. Donc g est bijective et $g^{-1} = f$, c'est-à-dire que f^{-1} est bijective et que $(f^{-1})^{-1} = f$. □

EXEMPLE 5.16 – La fonction $s : \mathbb{Z} \to \mathbb{Z}$ définie par $s(n) = -n$ est une bijection. De plus, $s^{-1} = s$.

EXEMPLE 5.17 – (Autres exemples.) Vous connaissez depuis le lycée quelques paires de bijections réciproques : exponentielle et logarithme, cosinus et arccosinus, sinus et arcsinus, tangente et arctangente. Normalement, ces exemples ont été revus au S1.

5.5. **Quelques opérations sur les fonctions.**

DÉFINITION 5.18 – Soit $f\colon E \to F$ et $g\colon F \to G$ deux fonctions. La *composée* de f et g, notée $g \circ f$, est la fonction $E \to G$ définie par $g \circ f(x) = g(f(x))$ pour $x \in E$.

EXEMPLE 5.19 – Prenons $E = F = G = \mathbb{R}$, puis $f(x) = x + 1$ et $g(x) = x^2$. Alors $g \circ f(x) = x^2 + 2x + 1$. Par contre $f \circ g(x) = x^2 + 1 \ldots$

EXEMPLE 5.20 – Si E est un ensemble, la fonction $E \to E$, $x \mapsto x$ est appelée l'identité de E ; il y a de nombreuses notations, par exemple I_E. Avec ce langage, deux fonctions $f\colon E \to F$ et $g\colon F \to E$ vérifient la propriété

$$(g \circ f = I_E) \wedge (f \circ g = I_F)$$

si, et seulement si, ce sont des bijections réciproques, càd $g = f^{-1}$.

Pour finir, nous allons maintenant donner un sens à $f(A)$ lorsque A est une partie du domaine de définition de f, ainsi qu'à $f^{-1}(B)$ lorsque B est une partie du but de f. Attention, la notation $f^{-1}(B)$ va prendre un sens même lorsque f n'est pas une bijection (et que la notation f^{-1} toute seule n'a pas de sens). Plus précisément :

DÉFINITION 5.21 – Soit $f\colon E \to F$ une application. Pour $A \subset E$ on note

$$f(A) = \{y \in F \mid y = f(x) \text{ pour un } x \in A\} \subset F\,.$$

Pour $B \subset F$, on définit

$$f^{-1}(B) = \{x \in E \mid f(x) \in B\} \subset E\,.$$

EXEMPLE 5.22 – Pour $c\colon \mathbb{Z} \to \mathbb{N}$ telle que $c(n) = n^2$, on constate que $c(\{1,2,3\}) = \{1,4,9\}$, et $c^{-1}(\{1,4,9\}) = \{-3,-2,-1,1,2,3\}$. (La fonction c^{-1} n'existe pas.)

EXEMPLE 5.23 – Les ensembles de la forme $f^{-1}(\{y\})$ pour $y \in F$ sont souvent appelés les fibres de f. Avec ce langage on énonce que f est surjective si et seulement si ses fibres sont toutes non-vides, et f est injective si et seulement si ses fibres ont au plus un élément.

Lorsque f est bijective, et que la fonction f^{-1} existe, on a $f^{-1}(\{y\}) = \{f^{-1}(y)\}$.

Un peu de vocabulaire pour finir. On dit que $f(A)$ est l'image (directe) de A par f ; que $f^{-1}(B)$ est l'image réciproque de B par f ; on dit que les éléments de la fibre $f^{-1}(\{y\})$ sont les antécédents de y par f.

6. Relations

Soit E un ensemble. Informellement, une *relation* sur E est une propriété qui dépend de *deux* éléments de R, et qui peut être vraie ou fausse. Pour rendre ça rigoureux, on va définir une relation comme l'ensemble des couples pour lesquels la propriété est vraie :

Définition 6.1 – Soit E un ensemble. Une relation (binaire) sur E est un sous-ensemble de $E \times E$. Si R est une relation, on s'autorise à noter $x\,R\,y$ au lieu de $(x, y) \in R$. (Et on dira parfois « x est en relation avec y. »)

(Si cette définition vous trouble au début, commencez avec la définition informelle, elle est suffisante dans bien des situations.)

Exemple 6.2 – Anticipons un peu sur quelque chose qui va revenir plus loin : les entiers peuvent être comparés, c'est-à-dire que parfois, pour $x, y \in \mathbb{N}$, on a $x \leq y$. On constate que $\leq$ est une relation sur $\mathbb{N}$.

Sur $\mathscr{P}(E)$, l'ensemble des parties de l'ensemble (arbitraire) E, on a la relation $\subset$. Notez que dans cet exemple, on peut trouver $A, B \in \mathscr{P}(E)$ tel que l'on n'a ni $A \subset B$, ni $B \subset A$ (exercice : trouvez un exemple !).

Définition 6.3 – On dit qu'une relation R sur E est :

- réflexive si tout élément de E est en relation avec lui-même. R est réflexive $\underset{\text{def}}{\Leftrightarrow} \forall x\ x\ R\ x$
- anti-réflexive si aucun élément de E est en relation avec lui-même. R est anti-réflexive $\underset{\text{def}}{\Leftrightarrow} \forall x\ \neg x\ R\ x$
- transitive si quand x est en relation avec y et y avec z alors x est en relation avec z. R est transitive $\underset{\text{def}}{\Leftrightarrow} \forall xyz\ x\ R\ y \wedge y\ R\ z \Rightarrow x\ R\ z$
- symétrique si quand x est en relation avec y alors y est en relation avec x. R est symétrique $\underset{\text{def}}{\Leftrightarrow} \forall xy\ x\ R\ y \Rightarrow y\ R\ x$
- anti-symétrique si quand x est en relation avec y alors y n'est pas en relation avec x ou alors $x = y$. R est anti-symétrique $\underset{\text{def}}{\Leftrightarrow} \forall xy\ x\ R\ y \wedge y\ R\ x \Rightarrow x = y$

Exemple 6.4 – Les relations suivantes sont réflexives : $\leq$ sur $\mathbb{N}$, ainsi que $\subset$ sur $\mathscr{P}(E)$, ou $=$ sur n'importe quel ensemble.

Les relations suivantes sont anti-réflexives : $<$ sur $\mathbb{N}$, ou $\neq$ sur n'importe quel ensemble.

Les relations $\neq$ et $=$ sont symétriques.

La relation $<$ sur $\mathbb{N}$ est anti-symétrique.

Définition 6.5 – On dit qu'une relation R sur E est une relation d'équivalence si elle est réflexive, transitive et symétrique. Dans ce cas, si $x \in E$ on note

$$[x] = \{y \in E \mid y\ R\ x\},$$

et on appelle cet ensemble la classe d'équivalence de x. L'ensemble de toutes les classes d'équivalences est noté E/R : c'est un sous-ensemble de $\mathscr{P}(E)$. On dit parfois que c'est l'ensemble quotient de E par R.

Exemple 6.6 – Considérons l'ensemble suivant, dont les éléments sont des lettres colorées (attention si vous êtes en train de lire une version imprimée en noir et blanc de ce document !) :

$$E = \{A, A, B, B, B, C\}.$$

Sur E, on définit la relation R par $x\ R\ y \Leftrightarrow x$ et y sont de la même couleur. On vérifie tout de suite que c'est effectivement une relation d'équivalence. Les classes d'équivalence sont :

$$[A] = [B] = \{A, B\},$$
$$[A] = [B] = [C] = \{A, B, C\},$$
$$[B] = \{B\}.$$

L'ensemble quotient est

$$E/R = \{[A], [A], [B]\}.$$

(On aurait pu l'écrire de plusieurs façons.) Intuitivement, l'ensemble E/R est ici « bleu, rouge, vert », l'ensemble des couleurs qui sont en jeu. D'une manière générale, prendre le quotient d'un ensemble E par une relation d'équivalence R, pour former E/R, est quelque chose que l'on fait lorsqu'on veut identifier tous les éléments ayant une certaine caractéristique en commun (dans l'exemple, c'est la couleur), en oubliant leurs autres différences.

Autre exemple, sur le même ensemble, définissons $x\ S\ y \Leftrightarrow x$ et y représentent la même lettre de l'alphabet. C'est encore une relation d'équivalence (vérifiez-le), et cette fois-ci on a

$$[A] = [A] = \{A, A\},$$
$$[B] = [B] = [B] = \{B, B, B\},$$
$$[C] = \{C\}.$$

Et ainsi

$$E/S = \{[A], [B], [C]\}.$$

Voici un théorème dont l'énoncé ne devrait pas être surprenant, après cet exemple :

THÉORÈME 6.7 – *Soit R une relation d'équivalence sur E, et soient $x, y \in E$. Si $[x] \cap [y] \neq \emptyset$, alors $[x] = [y]$.*

L'énoncé contraposé, équivalent, est bien sûr $[x] \neq [y] \implies [x] \cap [y] = \emptyset$. En d'autres termes, les différentes classes d'équivalences sont disjointes. L'ensemble E se trouve « découpé » en paquets disjoints, les classes d'équivalences.

Démonstration. Soit $z \in [x] \cap [y]$ (un tel z existe par hypothèse). Donc $z\ R\ x$ et $z\ R\ y$ par définition, et par symétrie, on a $x\ R\ z$.

Si l'on prend un élément quelconque $t \in [x]$, alors $t\ R\ x$, donc par transitivité (deux fois !), on a $t\ R\ z$ puis $t\ R\ y$. Donc $t \in [y]$. Comme t était arbitraire, on vient de montrer que $[x] \subset [y]$.

En renversant les rôles de x et y dans l'argument, on montre également que $[y] \subset [x]$. Au final on a bien $[x] = [y]$. □

EXEMPLE 6.8 – Sur l'ensemble $\mathbb{Z}$ des entiers relatifs, ayant fixé un entier n, on définit une relation $\equiv$ en spécifiant que $x \equiv y$ signifie « n divise $x - y$ ». On dit parfois que x et y sont « congrus modulo n », et on utilise souvent une notation du type

$$x \equiv y \quad (n) \qquad \text{ou} \qquad x \equiv y \mod n$$

ou autre. À titre d'exercice, vous montrerez que $\equiv$ est une relation d'équivalence.

DÉFINITION 6.9 (ORDRE) – Une relation binaire est un ordre si elle est réflexive, transitive et anti-symétrique. Un ordre peut être total s'il permet de comparer deux éléments différents quelconques, il est dit partiel dans le cas contraire.

EXEMPLE 6.10 – La relation $\leq$ est un ordre total sur $\mathbb{N}$, et $\subset$ est un ordre partiel sur $\mathscr{P}(E)$.

Soient deux ensembles ordonnés $(A_1, \leq_1)$ et $(A_2, \leq_2)$. Il y a plusieurs manières de définir un ordre sur l'ensemble produit $A1 \times A_2$.

Définition 6.11 (Ordre lexicographique) – On définit l'ordre lexicographique sur les couples de la manière suivante. On dit qu'un couple est plus petit qu'un autre si la première composante de l'un est strictement plus petite que la première composante de l'autre, ou alors si les premières composantes sont égales, on compare les deuxièmes. $(x_1, x_2) \leq_{lex} (y_1, y_2) \underset{\text{def}}{\Leftrightarrow} x_1 <_1 y_1 \vee (x_1 = y_1 \wedge x_2 \leq_2 y_2)$

On peut généraliser l'ordre lexicographique sur les paires, à tout n-uplet, en considérant l'ordre des lettres dans l'alphabet, l'ordre lexicographique sur les mots de n lettres est l'ordre du dictionnaire.

7. Les entiers

7.1. **Axiomes.** C'est reparti pour une série d'axiomes. Nous allons lister les propriétés des entiers qui permettent de travailler, sans chercher à obtenir une liste minimale (ce qui rendrait les choses plus abstraites). Comme pour les ensembles, nous ne serons pas trop formels (mais un peu plus, tout de même !).

On peut faire remarquer que les mathématiciens n'ont pas tous la même attitude vis-à-vis des entiers. Pour certains, les nombres entiers peuvent être définis rigoureusement à partir de la théorie des ensembles – mais pour cela, il faut aller plus loin avec les ensembles, et utiliser des axiomes dont nous n'avons pas parlé dans ce poly. Dans cette optique, les propriétés que l'on va lister ci-dessous ne sont pas des axiomes, mais des théorèmes.

Pour d'autres mathématiciens, c'est tout l'inverse : ils ne ressentent pas particulièrement le besoin de parler d'ensembles, ignorent les axiomes précédents, mais par contre, prennent les nombres entiers (c'est-à-dire le contenu des paragraphes suivants) comment point de départ. (On dit que ces mathématiciens « travaillent dans l'axiomatique de Peano », du nom d'un mathématicien italien.)

Quant à nous, nous prenons une approche axiomatique pour les ensembles et pour les entiers. C'est le plus simple.

Il existe donc un ensemble noté $\mathbb{N}$, dont les éléments sont appelés les entiers naturels. Il a les propriétés remarquables suivantes.

(1) Il y a deux éléments $0 \in \mathbb{N}$ et $1 \in \mathbb{N}$, avec $0 \neq 1$.

(2) Il existe une fonction $a \colon \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$, appelée l'addition, avec les propriétés suivantes. Notons que l'on va écrire $n + m$ au lieu de $a(n, m)$.

   (a) $\forall n \in \mathbb{N}, 0 + n = n$. On dit que 0 est neutre.

   (b) $\forall a \in \mathbb{N}, \forall b \in \mathbb{N}, a + b = b + a$. On parle de commutativité de l'addition.

   (c) $\forall a \in \mathbb{N}, \forall b \in \mathbb{N}, \forall c \in \mathbb{N}, a + (b + c) = (a + b) + c$. On parle de l'associativité de l'addition.

(3) Il existe une relation d'ordre total $\leq$ sur $\mathbb{N}$. Elle vérifie $0 < 1$ et la propriété

$$\forall (n, a, b) \in \mathbb{N}^3, a < b \implies n + a < n + b\,.$$

En particulier, en ajoutant n à la relation $0 < 1$, on obtient $n < n + 1$, pour tout n. Comme cette dernière remarque s'applique à $n + 1$, on a $n + 1 < (n + 1) + 1 = n + (1 + 1) = n + 2$ (faut-il expliquer que l'on écrit 2 pour $1 + 1$, pour faire court ?). Donc $n < n + 1 < n + 2$, et en particulier $0 < 1 < 2$. On peut continuer comme ça autant qu'on veut (ci-dessous, nous parlerons de la technique de la récurrence, qui permet de rendre rigoureux l'idée de « continuer autant qu'on veut », dans bien des situations).

Une remarque importante est que si $n + a = n + b$, alors $a = b$ (pourquoi ?).

(4) Voici une propriété de « minimalité », qui dit que tous les nombres entiers peuvent s'obtenir par les moyens ci-dessus : pour tout entier $n \in \mathbb{N}$, ou bien $n = 0$, ou bien n est de la forme $m + 1$ pour un $m \in \mathbb{N}$. En symboles, la propriété ci-dessous est un axiome :

$$\forall n \in \mathbb{N}, (n = 0) \vee (\exists m \in \mathbb{N}, n = m + 1)\,.$$

(5) Autre propriété fondamentale : toute partie non vide de $\mathbb{N}$ possède un plus petit élément :

$$\forall A \subset \mathbb{N}, A \neq \emptyset \implies \exists a \in A, \forall n \in A, a \leq n\,.$$

Exercice : montrer que a est en fait unique !

(6) Il existe une fonction $\mu\colon \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$, appelée la multiplication, avec les propriétés suivantes. Notons que l'on écrit, en général $n \cdot m$ ou même nm au lieu de $\mu(n, m)$.

(a) $\forall n \in \mathbb{N}, 1 \cdot n = n$. On dit que 1 est neutre.

(b) $\forall a \in \mathbb{N}, \forall b \in \mathbb{N}, a \cdot b = b \cdot a$. On parle de commutativité de la multiplication.

(c) $\forall a \in \mathbb{N}, \forall b \in \mathbb{N}, \forall c \in \mathbb{N}, a \cdot (b \cdot c) = (a \cdot b) \cdot c$. On parle de l'associativité de la multiplication.

(d) $\forall a \in \mathbb{N}, \forall b \in \mathbb{N}, \forall c \in \mathbb{N}, a \cdot (b + c) = a \cdot b + a \cdot c$. C'est la propriété de distributivité de la multiplication par rapport à l'addition.

7.2. **Le principe de récurrence.** Normalement, ces axiomes ne vous surprennent pas : vous les utilisez sans y penser. Seule le (5) est peut-être à garder en tête ; si la forme du (4) vous étonne (un peu), sachez qu'on va surtout l'utiliser pour montrer le théorème suivant, qui est d'une importance capitale.

THÉORÈME 7.1 – *Soit A une partie de $\mathbb{N}$. On suppose que A vérifie les deux propriétés suivantes :*

(1) $0 \in A$,

(2) $\forall n \in \mathbb{N}, n \in A \implies n + 1 \in A$.

Alors $A = \mathbb{N}$.

Démonstration. Soit $B = \mathbb{N} \setminus A$. On veut montrer que $B = \emptyset$, et on procède par l'absurde : supposons donc que $B \neq \emptyset$. D'après l'axiome (5), B possède un plus petit élément, disons $b \in B$.

Puisque $0 \in A$, on a $0 \notin B$, et donc certainement $b \neq 0$. D'après l'axiome (4), on peut écrire $b = a + 1$ avec a entier. On a fait remarquer ci-dessus que $a < a + 1 = b$. Comme b est le plus petit élément de B, notre conclusion est que $a \notin B$. Par définition, ceci revient à dire que $a \in A$.

Mais alors, d'après la propriété (2), $a \in A$ entraîne que $a + 1 \in A$, soit $b \in A$, ce qui est absurde puisque $b \in B$. Cette contradiction montre que $B = \emptyset$, ou de manière équivalente, que $A = \mathbb{N}$. □

Ce théorème est à la base de ce qu'on appelle le principe de récurrence. L'idée est (maintenant) simple. Supposons qu'on souhaite montrer que pour tout entier n, une certaine propriété $P(n)$ est vraie. Alors il suffit de vérifier que

(1) $P(0)$ est vraie (on dit parfois que c'est l'initialisation) ;

(2) $\forall n \in \mathbb{N}, P(n) \implies P(n+1)$ (on dit parfois que c'est la propriété d'hérédité).

En effet, si on pose $A = \{n \in \mathbb{N} \mid P(n)\}$, alors l'ensemble A vérifie les propriétés requises dans le théorème, donc $A = \mathbb{N}$.

Exemple 7.2 – Voici l'exemple le plus bête possible. Supposons que l'on veuille montrer que tout entier n vérifie $0 \leq n$ (on ne l'a pas listé dans les axiomes !). La propriété $P(n)$ étant « $0 \leq n$ », on a bien $P(0)$ puisque $0 \leq 0$ (sachant que $\leq$ est une relation d'ordre) ; et si $P(n)$ est vraie, on a $0 \leq n$, et nous savons que $n < n+1$ donc $0 \leq n+1$, c'est-à-dire que $P(n+1)$ est vraie. D'après le principe de récurrence, $P(n)$ est vraie pour tout n.

Exemple 7.3 – Montrons qu'il n'existe pas d'entier n tel que $0 < n < 1$. Pour cela, soit $P(n)$ la propriété

$$(n = 0) \vee (n \geq 1).$$

Montrons-la par récurrence. On a certainement $P(0)$ car $0 = 0$ (nous avons initialisé). Montrons l'hérédité, et pour cela, supposons que $P(n)$ est vraie.

Il y a deux cas. Si $n = 0$, alors $n + 1 = 1$, donc $1 \leq n + 1$ puisque $1 \leq 1$; donc $P(n + 1)$ est vraie. Deuxième cas, si $1 \leq n$, alors $1 \leq n + 1$ car $n < n + 1$. Finalement, on a bien $P(n + 1)$ dans tous les cas. La preuve est terminée.

Exercice : montrez le même énoncé en considérant l'ensemble $A = \{n \in \mathbb{N} \mid 0 < n < 1\}$ (s'il est non vide, alors...).

Parfois, on s'intéresse à des propriétés de la forme $P(n)$ seulement définies (ou intéressantes) pour n suffisamment grand, disons pour $n \geq n_0$, avec $n_0 \in \mathbb{N}$. Dans ce cas, il suffit de montrer que $P(n_0)$ est vraie, et que $\forall n \geq n_0, P(n) \implies P(n+1)$. (Exercice : démontrez que ce principe est correct !)

En général, on annonce au lecteur « montrons $P(n)$ pour $n \geq n_0$ par récurrence ». Voici un exemple.

Exemple 7.4 – Nous allons montrer que si n, m sont des entiers tous les deux non-nuls, alors $n \cdot m \neq 0$ (on dit que la multiplication est « intègre »). Pour cela, fixons un entier $m \neq 0$, et soit $P(n)$ la propriété

$$n \cdot m \neq 0.$$

Montrons-la par récurrence pour $n \geq 1$ (d'après l'exemple précédent, $n \neq 0$ est bien équivalent à $n \geq 1$, donc c'est bien ce que l'on veut).

Pour $P(1)$, on a $1 \cdot m = m \neq 0$, pas de problème.

Maintenant, supposons que $P(n)$ est vraie, donc $n \cdot m \neq 0$. En particulier $0 < n \cdot m$ (d'après l'exemple ci-dessus !). En ajoutant m à cette inégalité, on obtient $m < n \cdot m + m = (n + 1) \cdot m$, et en particulier $0 < (n + 1) \cdot m$. C'est bien $P(n + 1)$.

Voici donc comment on procède pour démontrer formellement les propriétés « évidentes » des entiers. Assurez-vous de comprendre ces arguments, et de comprendre pourquoi il est nécessaire de les donner (au moins une fois). En faisant les exercices, vous allez graduellement vous rendre compte que ces choses ne sont pas difficiles, et nécessitent seulement un peu de soin. Avec l'expérience, vous aurez rapidement la conviction que toute propriété des entiers que vous considérez comme évidente depuis des années peut effectivement se démontrer, quitte à y passer du temps. Et vous pourrez alors de nouveau affirmer, par exemple, qu'il est clair qu'il n'y a pas d'entier n tel que $0 < n < 1$ (comme ci-dessus) : vos interlocuteurs et vous-mêmes tomberez immédiatement d'accord sur l'existence d'une preuve formelle.

Il est grand temps de donner un exemple un peu plus consistant de récurrence :

Exemple 7.5 – Montrons par récurrence que $3^{2n+6} - 2^n$ est divisible par 7, pour tout n. Voyons pour $n = 0$: on a $3^6 - 1 = 728 = 7 \times 104$, donc ça marche. Voyons l'hérédité, et supposons donc que $3^{2n+6} - 2^n = 7k$ pour un certain entier k. On écrit

$$3^{2(n+1)+6} - 2^{n+1} = 9(3^{2n+6}) - 2 \cdot 2^n = 7(3^{2n+6}) + 2(3^{2n+6} - 2^n) = 7(3^{2n+6} + 2k).$$

On a bien ici un multiple de 7, donc la preuve par récurrence est terminée.

EXEMPLE 7.6 – On dit qu'un entier $p > 1$ est *premier* lorsque les seuls entiers qui divisent p sont 1 et p lui-même. Montrons par récurrence que tout entier $n \geq 2$ est un produit de nombres premiers (résultat classique !).

Attention, ici la propriété $P(n)$ que l'on montre par récurrence sur n n'est PAS « n est un produit de nombres premiers » (essayez, ça marche mal). On va faire ce que l'on appelle parfois une récurrence forte, c'est-à-dire que l'on prend pour $P(n)$ la propriété « pour tout entier m avec $2 \leq m \leq n$, on peut écrire m comme un produit de nombres premiers ». Il est évident que démontrer $P(n)$ pour tout n revient exactement à ce que l'on voulait faire, mais il se trouve que c'est plus facile, dans ce cas précis.

En effet, voyons $P(2)$ (attention, ici ça commence à 2 !). Le nombre 2 est premier, et c'est le seul nombre entre 2 et 2, donc on a bien la propriété $P(2)$.

Voyons l'hérédire ; supposons $P(n)$. Un nombre m tel que $2 \leq m \leq n+1$ est soit $m = n+1$, soit il vérifie $2 \leq m \leq n$, donc comme $P(n)$ est vraie, il suffit de traiter le cas $m = n+1$. Si $n+1$ est un nombre premier, alors on a fini. Deuxième cas, si ce nombre n'est pas premier, alors $n+1 = m_1 m_2$ avec $m_1 \neq 1$ et $m_1 \neq n+1$; on en déduit que $m_2 \neq 1$ et $m_2 \neq n+1$. Ainsi m_1 et m_2 sont tous les deux entre 2 et n. D'après $P(n)$, on peut écrire $m_1 = p_1 p_2 \cdots p_k$ et $m_2 = q_1 q_2 \cdots q_\ell$ où les p_i et les q_i sont premiers. Pour finir $n+1 = m_1 m_2 = p_1 p_2 \cdots p_k q_1 q_2 \cdots q_\ell$, c'est bien un produit de nombres premiers, et on a terminé.

7.3. **Définitions par récurrence.** Commençons par quelques mots sur la récursivité en informatique. Au S1 vous avez vu qu'une fonction peut, dans la plupart des langages, s'appeler elle-même. C'est ce qu'on appelle une *fonction récursive*, et l'idée est très proche de celle de la récurrence sur les entiers (quoiqu'il y ait des différences aussi).

Par exemple, voici des exemples en Python et Ocaml de fonctions récursives, qui renvoient un booléen.

```
def pair(n):
    if (n==0):
        return(True)
    elif (n==1):
        return(False)
    else:
        return (pair(n-2))
```

```
let rec pair n =
  if n=0 then true else
  if n=1 then false else
  pair (n-2)
```

On pourrait utiliser l'une ou l'autre de ces fonctions pour donner une (autre) définition de ce qu'est un entier pair : on dit qu'un entier n est pair si `pair(n)` renvoie `True`, par exemple. C'est un exemple de définition par récurrence. En voici un autre :

DÉFINITION 7.7 – Soient a et n des entiers. On définit la n-ième puissance de a, notée a^n, par récurrence ainsi :

$$a^0 = 1, \qquad a^{n+1} = a \cdot a^n .$$

En d'autres termes, a^n c'est « $a \cdot a \cdots a$, le tout n fois », mais pour rendre ça rigoureux, mieux vaut la définition par récurrence ci-dessus. L'essence de cette définition est que a^n est l'entier renvoyé par `puissance(a, n)` où :

```
def puissance(a, n):
    if n == 0:
        return 1
    else:
        return a * puissance(a, n-1)
```

Un autre exemple classique est celui des suites définies par récurrence. Une suite d'éléments de l'ensemble X, au fait, c'est une fonction $u\colon \mathbb{N} \to X$; on note en général u_n au lieu de $u(n)$, et on emploie aussi souvent la notation $(u_n)_{n\geq 0}$ pour parler de u. Si $f\colon X \to X$ est une fonction, alors on peut définir par récurrence la suite u par les conditions

$$u_0 = x_0, \qquad u_{n+1} = f(u_n)\,,$$

où x_0 est un élément de X donné. Par exemple, on peut considérer la suite u d'éléments de $\mathbb{R}$ donnée par

$$u_0 = 1, \qquad u_{n+1} = 3u_n^2 - 1\,.$$

En Python ça serait

```
def u(n):
    if n == 0:
        return 1
    else:
        return 3*u(n)**2 - 1
```

(Ou même `u= lambda n : 1 if n==0 else 3*u(n-1)**2 - 1`.) Les premiers termes sont $u_0 = 1, u_1 = 2, u_2 = 11, u_3 = 362, u_4 = 393131$.

Remarque 7.8. *Attention, cette remarque aborde un point difficile ; en première lecture, on pourra l'ignorer, et se contenter des explications ci-dessus.* Nous n'avons pas formalisé complètement ce qu'est une « définition par récurrence ». Ou plus précisément, nos explications utilisent un algorithme. Or la règle du jeu, normalement, c'est de tout exprimer à l'aide des ensembles...

Prenons donc l'exemple des suites définies par récurrence. Imaginons le cadre général, celui d'une fonction $f\colon X \to X$ qui nous est donnée. Si $x_0 \in X$ est fixé, on souhaite définir la suite u par les formules

$$(*) \qquad u_0 = x_0, \qquad u_{n+1} = f(u_n)\,.$$

En d'autres termes, on souhaite que u_n soit l'élément `u(f, n, x0)` où :

```
def u(f, n, x0):
    if n == 0:
        return x0
    else:
        return f( u(f, n-1, x0) )
```

Intuitivement, tout est parfaitement clair, mais a-t-on vraiment défini une suite, c'est-à-dire une *fonction* $u\colon \mathbb{N} \to X$? Si oui, alors *quel est son graphe, concrètement* ? Une fonction n'est pas une suite d'instructions pour faire un calcul...

Il ne faut pas s'y méprendre : ce qui est délicat ici, c'est de « faire rentrer » la définition dans le cadre qu'on s'est donné, avec notre définition de « fonction », elle-même adoptée parce qu'on a voulu se ramener aux ensembles, par économie. C'est une difficulté artificielle, qui ne doit pas faire oublier que le concept de « définition par récurrence » est intuitivement assez simple.

La réponse ceci dit, avec tous les détails, est compliquée. Dans un premier temps, on démontre par récurrence sur N qu'il existe une fonction unique $u^{(N)}\colon \{0,1,2,\dots,N\} \to X$ telle que $u^{(N)}(n+1) = f(u^{(N)}(n))$ pour $0 \leq n \leq N-1$. En effet, ça marche bien pour $N=0$, et puis si $\Gamma^{(N)}$ est le graphe de $u^{(N)}$, on montre qu'il suffit d'ajouter la paire $(N+1, f(u^{(N)}(N))$ pour obtenir le graphe de $u^{(N+1)}$... (détails laissés en exercice).

Puis, on voit chaque graphe $\Gamma^{(N)}$ comme une partie de $\mathbb{N} \times X$, et on pose

$$\Gamma := \bigcup_{N \geq 0} \Gamma^{(N)}\,.$$

Enfin, on prend le temps de vérifier que Γ est bien le graphe d'une fonction u, qui vérifie $u(n+1) = f(u(n))$ pour $n \geq 0$. *Par définition,* la suite u « définie par (*) », c'est celle-ci.

Au risque de se répéter : la première définition donnée est parfaitement suffisante pour déterminer chaque u_n sans ambiguïté. Avec cette remarque, nous avons voulu prévenir une objection qui a pu vous venir à l'esprit, à savoir qu'une fonction n'est pas, techniquement, une suite d'instructions pour faire un calcul – c'est un graphe. En pratique, ce genre de pinaillage n'intéresse pas les mathématiciens, surtout une fois qu'on a vu, par un exemple, que la formalisation n'est jamais une véritable difficulté.

7.4. **Quand généraliser simplifie !**

Quand on souhaite prouver une propriété ou écrire un programme, il est parfois plus simple de prouver une propriété plus forte ou d'écrire un programme plus général. En effet, quand on réalise une preuve par récurrence le fait de prouver une propriété plus générale permet d'avoir une hypothèse de récurrence aussi plus générale. De même, en programmation, si on construit une fonction récursive résolvant un problème plus général, cela nous autorise à utiliser un appel récursif résolvant le problème plus général.

Prenons l'exemple célèbre des tours de Hanoï. Ce jeu consiste à déplacer des disques de diamètres différents d'une tour de « départ » à une tour d'« arrivée » en passant par une tour « intermédiaire », et ceci en un minimum de coups, tout en respectant les règles suivantes :

- on ne peut déplacer plus d'un disque à la fois ;
- on ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.

L'astuce pour résoudre se problème est de procéder par récurrence et de résoudre un problème plus général. Au lieu d'essayer seulement de déplacer les disques de la tour de « départ » à la tour d'« arrivée » en passant par la tour « intermédiaire », on va déplacer les disques d'un tour quelconque à une autre quelconque aussi en passant par la troisième. Une fois cette astuce trouvée, le problème devient facile à résoudre. Pour un disque la problème est trivial. Pour n disques, on sait que par la « magie » de la récursivité on peut déplacer $n-1$ disque comme on veut. Il suffit donc de les déplacer de « départ » vers « intermédiaire », de mettre le disque le plus gros à sa place sur la tour d'arrivée, et de déplacer à nouveau les $n-1$ disque de la tour « intermédiaire » à la tour d'« arrivée ».

7.5. **Manipulation de sommes et produits.**

On va discuter un peu des notations $\sum$ et $\prod$, et de leurs propriétés. Dans ce qui suit, disons pour être concrets que nous traitons toujours d'entiers naturels (les éléments de $\mathbb{N}$) ; mais en réalité, les règles s'appliquent sans problème aux nombres réels, aux complexes, et à vrai dire dans toutes les situations où l'on a une addition et une multiplication avec les propriétés de commutativité, associativité et distributivité. Vous étudierez ces

situations plus en détail dans le cours d'algèbre, le but ici étant de s'entraîner aux manipulations de symboles.

DÉFINITION 7.9 – Soit $f : \mathbb{N} \to \mathbb{N}$ une fonction (comme annoncé, la discussion pourrait se faire avec une fonction $\mathbb{N} \to \mathbb{R}$ ou $\mathbb{N} \to \mathbb{C}$ etc). On définit la somme $\sum_{i=k}^{n} f(i)$ quand $k \leq n$ par récurrence sur n de la manière suivante :

$$\sum_{i=k}^{n} f(i) =_{def} \begin{cases} f(k) & \text{si } n = k \\ f(n) + \sum_{i=k}^{n-1} f(i) & \text{si } n > k \end{cases}$$

Autrement dit, $\sum_{i=k}^{n} f(i)$ est `somme(f, k, n)` où :

```
def somme(f, k, n):
    if n == k :
        return f(k)
    else :
        return f(n) + somme(f, k, n-1)
```

LEMME 7.10 – *Soit λ une constante (qui ne dépend pas de i).*

(1) Décomposition (on peut "sortir" le premier terme de la somme) :

$$\sum_{i=k}^{n} f(i) = f(k) + \sum_{i=k+1}^{n} f(i)$$

(2) Décomposition (on peut "sortir" le dernier terme de la somme) :

$$\sum_{i=k}^{n} f(i) = \sum_{i=k}^{n-1} f(i) + f(n)$$

(3) Linéarité :

$$\sum_{i=k}^{n} \lambda a(i) = \lambda \sum_{i=k}^{n} a(i)$$

Linéarité ? :

(4)

$$\sum_{i=k}^{n} (a(i) + b(i)) = \sum_{i=k}^{n} a(i) + \sum_{i=k}^{n} b(i)$$

(5) Comptage :

$$\sum_{i=k}^{n} \lambda = (n + 1 - k)\lambda$$

(6) Découpage d'une somme : Si $k \leq p \leq n$ alors

$$\sum_{i=k}^{n} a(i) = \sum_{i=k}^{p} a(i) + \sum_{i=p}^{n} a(i)$$

(7) Décalage d'indice :

$$\sum_{i=k}^{n} a(i) = \sum_{i=k+m}^{n+m} a(i - m)$$

(8) Parcours à rebours :

$$\sum_{i=k}^{n} a(i) = \sum_{i=k}^{n} a(n - i)$$

Démonstration. Par récurrence sur n en utilisant les propriétés d'associativité, commutativité et distributivité de la somme et du produit. □

EXERCICE 7.11 – Donner une version rigoureuse de la preuve suivante du fait que $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$ (sans utiliser de récurrence) :

$$\begin{array}{ccccccc} 1 & + & 2 & + & \dots & + & n \\ n & + & n-1 & + & \dots & + & 1 \\ \hline n+1 & + & n+1 & + & \dots & + & n+1 \end{array}$$

DÉFINITION 7.12 – On définit le produit $\prod$ quand $k \leq n$ par récurrence sur n de la manière suivante :

$$\prod_{i=k}^{n} f(i) =_{def} \begin{cases} 1 & \text{si } k = n \\ f(n) \times \prod_{i=k}^{n-1} f(i) & \text{si } n > k \end{cases}$$

LEMME 7.13 – *Soit λ une constante (qui ne dépend pas de i).*

(1) Décomposition (on peut "sortir" le premier terme du produit) :

$$\prod_{i=k}^{n} f(i) = f(k) \prod_{i=k+1}^{n} f(i)$$

(2) Décomposition (on peut "sortir" le dernier terme du produit) :

$$\prod_{i=k}^{n} f(i) = f(n) \prod_{i=k}^{n-1} f(i)$$

(3)

$$\prod_{i=k}^{n} \lambda a(i) = \lambda^{n+1-k} \prod_{i=k}^{n} a(i)$$

(4) Associativité :

$$\prod_{i=k}^{n} a(i)b(i) = \prod_{i=k}^{n} a(i) \prod_{i=k}^{n} b(i)$$

(5) Comptage :

$$\prod_{i=k}^{n} \lambda = \lambda^{n+1-k}$$

(6) Découpage d'un produit : Si $k \leq p \leq n$ alors

$$\prod_{i=k}^{n} a(i) = \prod_{i=k}^{p} a(i) \prod_{i=p}^{n} a(i)$$

(7) Décalage d'indice :

$$\prod_{i=k}^{n} a(i) = \prod_{i=k+m}^{n+m} a(i-m)$$

(8) Parcours à rebours :

$$\prod_{i=k}^{n} a(i) = \prod_{i=k}^{n} a(n-i)$$

8. Dénombrement

Le dénombrement, c'est l'art et la manière de compter, dans des situations toujours plus subtiles.

Tous les cours sur le dénombrement commencent par donner un certain nombre de formules considérées comme « évidentes », avant d'aller plus loin. A vrai dire, sur ce sujet, chaque professeur va espérer que vous avez déjà vu des démonstrations ailleurs ! Les propriétés dont on a besoin sont, en effet, très intuitives, mais les preuves formelles – dont on doit convaincre le lecteur qu'on peut les produire, rappelez-vous – sont finalement un peu délicates.

C'est un terrain d'entraînement parfait pour notre cours : ici, nous allons montrer les formules de base du dénombrement, en essayant de vous faire sentir à chaque fois le passage de l'idée intuitive à la formalisation. Bien sûr, vous en retirerez des choses utiles (les nombres binomiaux, par exemple, se retrouvent partout), mais c'est aussi un prétexte pour vous faire étudier une rédaction mathématique.

8.1. **Fondations.**

Définition 8.1 – Soit X un ensemble. S'il existe un entier n et une bijection

$$\{1, 2, \dots, n\} \longrightarrow X\,,$$

alors on dira que X est de cardinal n, ou encore que X possède n éléments. On dit que X est fini lorsqu'il existe un n tel que X est de cardinal n.

Le seul ensemble dont on dit qu'il est de cardinal 0 est l'ensemble vide (il est fini).

Remarque 8.2.
(1) Dans toute cette partie, on écrit $\{1, 2, \dots, n\}$ comme raccourci pour $\{x \in \mathbb{N} \mid 1 \leq x \leq n\}$.
(2) Si $f\colon \{1, 2, \dots, n\} \longrightarrow X$ est une telle bijection, alors on a bien envie d'employer la notation $x_i = f(i)$, de sorte que $X = \{x_1, x_2, \dots, x_n\}$ (car f est surjective), et $x_i \neq x_j$ quand $i \neq j$ (car f est injective).
(3) Supposons qu'il existe une bijection entre X et l'ensemble Y. Alors si X est de cardinal n, on peut dire la même chose de Y. Pourquoi ?

Voici les propriétés que l'on considère fondamentales.

Lemme 8.3 – *Soit X un ensemble.*

(1) Soit $A \subset X$ une partie de X. Si X est fini, alors A aussi. De plus, si A est de cardinal m, et si X est de cardinal n, alors $m \leq n$.

(2) On suppose que

$$X = A_1 \cup A_2 \cup \cdots \cup A_k\,,$$

où les parties A_i vérifient $A_i \cap A_j = \emptyset$ pour $i \neq j$. De plus, on suppose que A_i est de cardinal n_i, pour chaque indice i. Alors X est de cardinal $n_1 + n_2 + \cdots + n_k$.

Démonstration. Comme la démonstration est un peu longue, voici une esquisse. Vous pourrez compléter les arguments, à titre d'exercice. Il est possible de combler chaque détail !

Commençons par le (1). Les étapes sont :
— On démontre qu'on peut se ramener au cas de $X = \{1, \dots, n\}$ pour n entier. (Il faut donc montrer que toute partie de $\{1, \dots, n\}$ est finie, de cardinal $\leq n$.)
— On peut le faire par récurrence sur n. Toute partie de $\{1, \dots, n+1\}$ est soit une partie de $\{1, \dots, n\}$, soit est de la forme $A' \cup \{n+1\}$ où A' est une partie de $\{1, ..., n\}$, et on considère les deux cas séparément.

Voyons maintenant le point (2).

— Il suffit de montrer le cas $k = 2$: le cas général se fait alors facilement par récurrence.
— Il faut montrer que $\{1, \dots, n+m\}$ est l'union disjointe de $\{1, \dots, n\}$ et de $\{n+1, \dots, n+m\}$, pour n, m entiers. (Ça nécessite de faire remarquer qu'il n'y a pas d'entier x tel que $n < x < n + 1$.)
— Il faut trouver une bijection entre $\{1, \dots, m\}$ et $\{n + 1, \dots, n + m\}$. On prendra $x \mapsto x + n$, dont la réciproque est $x \mapsto x - n$ (il faut donc un peu parler des entiers relatifs et de leurs propriétés (ou ce qui revient au même, des propriétés de la soustraction), ce qu'on n'a pas fait dans ce poly).
— On montre un résultat intermédiaire : si A_1 est en bijection avec A'_1, et A_2 avec A'_2, et si $A_1 \cap A_2 = \emptyset$ et $A'_1 \cap A'_2 = \emptyset$, alors il y a une bijection entre $A_1 \cup A_2$ et $A'_1 \cup A'_2$.

□

Voyons des exemples. On peut, pour commencer, compter le nombre d'éléments dans un produit cartésien.

LEMME 8.4 – *Soit X un ensemble de cardinal n, et Y un ensemble de cardinal m. Alors le produit cartésien $X \times Y$ est de cardinal nm.*

Démonstration. Écrivons $X = \{x_1, \dots, x_n\}$, comme dans la remarque ci-dessus, et posons $A_i = \{(x_i, y) \mid y \in Y\}$, pour $1 \leq i \leq n$. Alors $X \times Y$ est l'union des sous-ensembles $A_1, A_2, \dots, A_n$, qui sont disjoints. D'après le lemme, le cardinal de $X \times Y$ est donc la somme des cardinaux des A_i.

Maintenant, fixons i, et regardons l'application $Y \to A_i$ définie par $y \mapsto (x_i, y)$. C'est une bijection, dont la réciproque $A_i \to Y$ est donnée par $(x_i, y) \mapsto y$. Donc le cardinal de A_i est m, indépendamment de i.

Finalement, le cardinal de $X \times Y$ est $m + m + \cdots + m$, avec n fois le nombre m ; c'est donc bien nm. □

COROLLAIRE 8.5 – *Si X possède n éléments, et si $k \geq 1$ est un entier, alors X^k possède n^k éléments.*

Démonstration. Récurrence sur k, laissée en exercice. □

On va maintenant compter le nombre de façons de choisir k éléments distincts dans un ensemble X de cardinal n. En d'autres termes, pour tout ensemble X, posons

$$\mathscr{A}_k(X) = \{(a_1, a_2, \dots, a_k) \mid a_i \neq a_j \text{ pour } i \neq j\},$$

de sorte que $\mathscr{A}_k(X) \subset X^k$. (Un élément de $\mathscr{A}_k(X)$ est parfois appelé un k-arrangement d'éléments de X, d'où la notation.)

THÉORÈME 8.6 – *Soit X un ensemble de cardinal n, et soit $k \geq 1$ un entier. Alors l'ensemble $\mathscr{A}_k(X)$ est de cardinal*

$$n(n-1)(n-2)\cdots(n-k+1)$$

si $k \leq n$.

Essentiellement, l'idée est qu'il faut d'abord choisir a_1, pour lequel on a n choix, puis on choisit a_2, pour lequel on a $(n-1)$ choix, puis a_3 avec $(n-2)$ choix, etc. La démonstration qui suit ne dit pas autre chose.

Démonstration. On va procéder par récurrence sur k. La propriété est exactement celle du théorème, c'est-à-dire, l'entier k étant donné : pour tout ensemble X de cardinal $n \geq k$, le cardinal de $\mathscr{A}_k(X)$ est $n(n-1)(n-2)\cdots(n-k+1)$.

Pour $k = 1$, l'ensemble $\mathscr{A}_1(X)$ étant X lui-même, on constate bien qu'il est de cardinal n.

Supposons donc que $\mathscr{A}_{k-1}(X)$ est de cardinal $n(n-1)\cdots(n-(k-1)+1)$ pour tout ensemble X de cardinal $n \geq k-1$, et montrons la propriété pour k. Soit $X = \{x_1, x_2, \ldots, x_n\}$ un ensemble de cardinal n. Posons

$$A_i = \{(x_i, a_1, a_2, \ldots, a_{k-1}) \mid a_s \neq a_t \text{ pour } s \neq t, \text{ et } a_s \neq x_i\} \subset \mathscr{A}_k(X)\,.$$

Alors $\mathscr{A}_k(X)$ est l'union disjointe des parties A_i, pour $1 \leq i \leq n$. (C'est la traduction de : « on commence par choisir le premier élément, qui est l'un des x_i ».)

Calculons le cardinal de A_i. On considère l'application

$$\mathscr{A}_{k-1}(X \smallsetminus \{x_i\}) \longrightarrow A_i$$

définie par $(a_1, \ldots, a_{k-1}) \mapsto (x_i, a_1, \ldots, a_{k-1})$. C'est une bijection, dont la réciproque est donnée par $(x_i, a_1, \ldots, a_{k-1}) \mapsto (a_1, \ldots, a_{k-1})$. Donc le cardinal de A_i est aussi celui de $\mathscr{A}_{k-1}(X \smallsetminus \{x_i\})$. Par récurrence, puisque $X \smallsetminus \{x_i\}$ est de cardinal $n-1$, le cardinal de A_i est finalement

$$m(m-1)\cdots(m-(k-1)+1)$$

avec $m = n-1$, c'est-à-dire

$$(n-1)(n-2)\cdots(n-k+1)\,.$$

C'est le même cardinal pour chaque indice i. Pour $\mathscr{A}_k(X)$, il faut donc faire la somme de ce nombre n fois, ce qui donne

$$n(n-1)(n-2)\cdots(n-k+1)$$

pour le cardinal de $\mathscr{A}_k(X)$. □

Exemple 8.7 – Le cas $k = n$ est intéressant. Pour fabriquer un élément de $\mathscr{A}_n(X)$, il faut finalement prendre *tous* les éléments de X, mais... dans un certain ordre ! Le cardinal de $\mathscr{A}_n(X)$ est donc le nombre de façons de « trier » les éléments de X. Le théorème affirme que ce nombre est $n(n-1)(n-2)\cdots\times 3\times 2\times 1$.

Pour $X = \{1,2,3\}$ par exemple, on doit avoir $3\times 2\times 1 = 6$ façons de faire. Et en effet, les éléments de $\mathscr{A}_3(\{1,2,3\})$ sont

$$(1,2,3), (1,3,2), (2,3,1), (2,1,3), (3,1,2), (3,2,1)\,.$$

Définition 8.8 – Le produit des nombres entiers de 1 à n est appelé factorielle n, et est noté $n!$. (Par convention, on pose aussi $0! = 1$; on vous laisse méditer sur ce choix.)

Donc $n!$ est le cardinal de $\mathscr{A}_n(\{1,2,\ldots,n\})$. Avec cette notation, le théorème précédent affirme que le cardinal de $\mathscr{A}_k(X)$ est $n!/(n-k)!$, si X possède n éléments.

Le petit résultat ci-dessous est assez évident, mais il aide souvent à la rédaction.

Lemme 8.9 – *Soit $f\colon X \to Y$ une fonction. On suppose que Y est fini, de cardinal n. De plus, on suppose qu'il existe un entier m tel que, pour tout $y \in Y$, l'ensemble $f^{-1}(\{y\})$ est de cardinal m. Alors X est fini, de cardinal nm.*

(À titre d'exercice, vous montrerez que les hypothèses entraînent que f est surjective, si X est non-vide.)

Démonstration. L'ensemble X est l'union disjointe des parties $f^{-1}(\{y\})$ (pourquoi ?), indexées par $y \in Y$. Il y a n parties, chacune de cardinal m, d'où le résultat. □

Définition 8.10 – Soit X un ensemble, et k un entier. On note $\mathscr{P}_k(X)$ l'ensemble des parties de X de cardinal k.

On va maintenant calculer le cardinal de $\mathscr{P}_k(X)$, lorsque X est fini. Voici la version informelle. On a calculé le cardinal de $\mathscr{A}_k(X)$, c'est $n!/(n-k)!$, mais on peut aussi le calculer d'une autre façon : pour déterminer un élément de $\mathscr{A}_k(X)$, on peut d'abord choisir un ensemble de k éléments de X, c'est-à-dire un élément de $\mathscr{P}_k(X)$, et ensuite, il faut mettre les éléments de cet ensemble dans l'ordre. Il y a r façons de choisir l'ensemble, où l'entier r est le cardinal de $\mathscr{P}_k(X)$, et il y a $k!$ façons de faire le tri, on l'a vu. Donc le cardinal de $\mathscr{A}_k(X)$ est aussi $rk!$, d'où la valeur de r.

Voyons une version plus détaillée.

THÉORÈME 8.11 – *Soit X de cardinal n, et soit k un entier, avec $0 \leq k \leq n$. Alors l'ensemble $\mathscr{P}_k(X)$ est de cardinal*

$$\frac{n!}{k!(n-k)!}\,.$$

Démonstration. On définit une fonction

$$f\colon \mathscr{A}_k(X) \longrightarrow \mathscr{P}_k(X)$$

par $f\big((x_1,\dots,x_k)\big) = \{x_1,\dots,x_k\}$. (Prenez bien le temps de comprendre la définition de f.)

Si $y = \{x_1,\dots,x_k\} \in \mathscr{P}_k(X)$, alors $f^{-1}(\{y\}) = \mathscr{A}_k(y)$. (Là encore, n'allez pas trop vite en lisant ceci.) En particulier, le cardinal de $f^{-1}(\{y\})$ est $k!$, puisque le cardinal de y est k.

D'après le dernier lemme, si on note r le cardinal de $\mathscr{P}_k(X)$ (que l'on cherche à calculer !), alors $rk!$ est le cardinal de $\mathscr{A}_k(X)$. D'où $rk! = n!/(n-k)!$, et enfin $r = n!/(k!(n-k!))$ comme annoncé. □

8.2. **Les nombres binomiaux.**

DÉFINITION 8.12 – On note, pour $0 \leq k \leq n$,

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}\,.$$

Les nombres de cette forme (pour diverses valeurs de k et n) sont appelés nombres binomiaux. Lorsque $k < 0$ ou $k > n$, on pose $\binom{n}{k} = 0$.

On a donc montré que le cardinal de $\mathscr{P}_k(X)$ est $\binom{n}{k}$, lorsque X est de cardinal n. (C'est encore vrai pour $k < 0$ ou $k > n$, car $\mathscr{P}_k(X)$ est alors vide !)

THÉORÈME 8.13 – *Les nombres binomiaux ont les propriétés suivantes.*

(1) $\displaystyle\binom{n}{k} = \binom{n}{n-k}.$

(2) $\displaystyle\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}.$

Démonstration. On laisse le (1) en exercice facile (il y a au moins 2 démonstrations bien différentes, les voyez-vous ?). Voyons le (2) (on donne une seule démonstration, et là encore, on vous laisse imaginer une deuxième, plus courte mais moins parlante...)

Soit X un ensemble de cardinal $n+1$, et soit $x_0 \in X$. Écrivons $\mathscr{P}_{k+1}(X) = A \cup B$, où

$$A = \{E \in \mathscr{P}_{k+1}(X) \mid x_0 \in E\}\,,$$

et $B = \mathscr{P}_{k+1}(X) \smallsetminus A$ (de sorte que $A \cap B = \emptyset$).

Le cardinal de A est $\binom{n}{k}$. En effet, informellement, une partie de X qui est dans A est de la forme $\{x_0, x_1, ..., x_k\}$, donc il faut choisir les k éléments $x_1, ..., x_k$

dans $X \smallsetminus \{x_0\}$ qui est de cardinal n. Plus formellement, on a une bijection $A \longrightarrow \mathscr{P}_k(X \smallsetminus \{x_0\})$ définie par $E \mapsto E \smallsetminus \{x_0\}$; sa réciproque est $F \mapsto F \cup \{x_0\}$.

Un peu de la même façon, on prouve que le cardinal de B est $\binom{n}{k+1}$. En effet B est en bijection avec $\mathscr{P}_{k+1}(X \smallsetminus \{x_0\})$ (et même mieux : il est en fait égal à cet ensemble, quand on y réfléchit).

Puisque $A \cap B = \emptyset$, le cardinal de $\mathscr{P}_{k+1}(X)$, qui est $\binom{n+1}{k+1}$, est aussi la somme $\binom{n}{k} + \binom{n}{k+1}$. Voilà qui prouve le point (2). □

À l'aide de la propriété (1), on peut calculer très facilement les nombres ci-dessus. On va construire un tableau, qui contient dans la n-ième ligne et dans la k-ième colonne le nombre $\binom{n}{k}$; par ailleurs, on convient de ne pas écrire les zéros dans le tableau (on laisse un blanc). Les premières lignes sont donc :

$$\begin{array}{cccc} 1 & & & \\ 1 & 1 & & \\ 1 & 2 & 1 & \\ 1 & 3 & 3 & 1 \end{array}$$

C'est le *triangle de Pascal* qui apparaît peu à peu. Pour écrire la ligne suivante, on utilise (1) comme ceci :

$$\begin{array}{cccc} 1 & & & \\ 1 & 1 & & \\ 1 & 2 & 1 & \\ 1 & \boxed{3} & \boxed{3} & 1 \\ & & \boxed{6} & \end{array}$$

(Chaque nombre est donc la somme de celui immédiatement au-dessus et de celui à gauche au-dessus ; vérifiez que la ligne suivante est $1, 4, 6, 4, 1$). Le calcul est très rapide, et il suffit de se rappeler que la toute première ligne (pour $n = 0$) contient juste un 1 (pour $k = 0$). On gagne cependant un peu de temps à mémoriser quelques lignes.

L'utilisation la plus fameuse des nombres $\binom{n}{k}$ est la *formule du binôme de Newton* :

THÉORÈME 8.14 – *Soient a et b des entiers (ou des réels ou des complexes, ou en fait n'importe quels éléments avec lesquels les règles de calcul usuelles s'appliquent.) Alors pour tout entier n on a*

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k} .$$

Par exemple on a

$$(a+b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4 ,$$

d'après le triangle de Pascal. Précisons qu'un terme de la forme $a^k b^{n-k}$ est appelé un *binôme* (car il fait intervenir deux éléments différents), et que c'est la formule de Newton qui vaut aux nombres $\binom{n}{k}$ le nom de coefficients binomiaux.

Donnons une idée de démonstration, pour commencer. On fait quelques développements « sans permuter a et b » :

$$(a+b)^2 = (a+b)(a+b) = a^2 + ab + ba + b^2 .$$

Puis

$$\begin{aligned} (a+b)^3 &= (a^2 + ab + ba + b^2)(a+b) \\ &= a^3 + aba + ba^2 + b^2a + a^2b + ab^2 + bab + b^3 . \end{aligned}$$

Ensuite on regroupe les termes, puisque aba, a^2b et ba^2 sont égaux, par exemple. On trouve $a^3+3a^2b+3ab^2+b^3$, puisqu'il y avait trois termes donnant a^2b, un seul donnant a^3, etc.

Cette petite analyse nous fait réaliser qu'un terme de $(a+b)^n$, après développement mais avant de faire permuter a et b et de regrouper, est de la forme $x_1x_2\cdots x_n$ avec chaque $x_i \in \{a,b\}$; et par ailleurs, chaque écriture $x_1x_2\cdots x_n$ se retrouve une fois et une seule.

Quand on veut regrouper les termes, il faut compter combien de fois on trouve a^kb^{n-k}, pour chaque k. Facile : pour obtenir a^kb^{n-k} à partir de $x_1x_2\cdots x_n$, il s'agit de choisir quels indices $i_1,\dots,i_k$ vérifient $x_{i_1}=x_{i_2}=\cdots=x_{i_k}=a$, et pour les autres indices $x_j=b$. Il faut donc choisir un sous-ensemble de k éléments parmi l'ensemble $\{1,\dots,n\}$ de tous les indices, donc il y a $\binom{n}{k}$ façons de le faire. Finalement, on obtient un terme $\binom{n}{k}a^kb^{n-k}$, pour chaque k, comme on voulait le montrer.

Pour rédiger une démonstration plus formelle (mais qui explique moins bien la formule), on va faire une récurrence.

Démonstration. la propriété $P(n)$ est « la formule pour $(a+b)^n$ est vraie pour n ». On vérifie sans peine $P(0)$ qui affirme que $(a+b)^0=1$, et aussi $P(1)$ qui affirme que $(a+b)^1=a+b$, et ci-dessus nous avons vérifié $P(2)$ et $P(3)$. L'initialisation est donc largement faite ($P(0)$ seule aurait suffi).

Supposons donc que $P(n)$ est vraie et penchons nous sur la propriété $P(n+1)$. On écrit

$$\begin{aligned}(a+b)^{n+1} &= (a+b)^n(a+b)\\ &= \left(\sum_{k=0}^{n}\binom{n}{k}a^kb^{n-k}\right)(a+b)\,.\end{aligned}$$

Ici on a utilisé $P(n)$ pour développer $(a+b)^n$. Le reste n'est que calcul, l'étape essentielle étant l'utilisation de la formule (1) pour les coefficients binomiaux :

$$\begin{aligned}(a+b)^{n+1} &= \sum_{k=0}^{n}\binom{n}{k}a^{k+1}b^{n-k}+\sum_{k=0}^{n}\binom{n}{k}a^kb^{n+1-k}\\ &= \sum_{k=0}^{n+1}\binom{n}{k-1}a^kb^{n+1-k}+\sum_{k=0}^{n}\binom{n}{k}a^kb^{n+1-k}\\ &= \sum_{k=0}^{n+1}\left(\binom{n}{k-1}+\binom{n}{k}\right)a^kb^{n+1-k}\\ &= \sum_{k=0}^{n+1}\binom{n+1}{k}a^kb^{n+1-k}\,.\end{aligned}$$

(Notez comment, pour obtenir la deuxième égalité, nous avons remplacé k par $k-1$ dans la somme de gauche ; la somme est inchangée si l'on fait maintenant varier k de 1 à $n+1$ au lieu de 0 à n, mais nous avons même laissé k démarrer à 0 car $\binom{n}{-1}=0$.) On a bien obtenu $P(n+1)$ en supposant $P(n)$, et par le principe de récurrence, on sait maintenant que $P(n)$ est vraie pour tout n. □