

## Leçon 11 - Structure logique

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques
2. Quant & ens.
3. Assertions
4. Méthodes dem.
  - 4.1  $\Rightarrow ?$
  - 4.2  $\Leftrightarrow ?$
  - 4.3 Absurde
  - 4.4 C.N., C.S.
  - 4.5 Contre-exemple
  - 4.6 Récurrence
  - 4.7 Algorithmique

## Leçon 11 - Structure logique

20 septembre 2024

⇒ **Reprendre quelques principes de base de démonstration**

⇒ **Deux raisonnements en particulier**

1. Cours mathématiques
2. Quantificateurs et notations ensemblistes
3. Vocabulaire sur les assertions
4. Principales méthodes de démonstration
  - 4.1. Démonstration d'une implication
  - 4.2. Démonstration d'une équivalence
  - 4.3. Raisonnement par l'absurde
  - 4.4. Conditions nécessaire, suffisante
  - 4.5. Contre-exemple
  - 4.6. Démonstration par récurrence
  - 4.6. Démonstration par algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

⇒ **Prendre quelques principes de base de démonstration**

⇒ **Deux raisonnements en particulier**

1. Cours mathématiques
2. Quantificateurs et notations ensemblistes
3. Vocabulaire sur les assertions
4. Principales méthodes de démonstration
  - 4.1. Démonstration d'une implication
  - 4.2. Démonstration d'une équivalence
  - 4.3. Raisonnement par l'absurde
  - 4.4. Conditions nécessaire, suffisante
  - 4.5. Contre-exemple
  - 4.6. Démonstration par récurrence
  - 4.6. Démonstration par algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Implication

Considérons deux assertions  $A$  et  $B$ . On veut démontrer que  $A \Rightarrow B$ .

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

# Implication

Considérons deux assertions  $A$  et  $B$ . On veut démontrer que  $A \Rightarrow B$ .

**Savoir-faire.**  $A \Rightarrow B$ . Raisonnement direct

On suppose  $A$  vraie, par une succession d'implications connues (calculs, résultats de théorèmes...), on prouve qu'alors  $B$  est vraie.

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques
2. Quant& ens.
3. Assertions
4. Méthodes dem.
  - 4.1 ⇒ ?
  - 4.2. ⇔ ?
  - 4.3. Absurde
  - 4.4. C.N., C.S.
  - 4.5. Contre-exemple
  - 4.6. Récurrence
  - 4.7. Algorithme

# Implication

Considérons deux assertions  $A$  et  $B$ . On veut démontrer que  $A \Rightarrow B$ .

**Savoir-faire.**  $A \Rightarrow B$ . Raisonnement direct

On suppose  $A$  vraie, par une succession d'implications connues (calculs, résultats de théorèmes...), on prouve qu'alors  $B$  est vraie.

## Exercice

Montrer que si  $(x, y)$  est élément de  $]0, 2[ \times ] - 2, 0[$  alors

$$\frac{1}{x} - \frac{1}{y} > 1$$

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques
2. Quant& ens.
3. Assertions
4. Méthodes dem.
  - 4.1 ⇒ ?
  - 4.2. ⇔ ?
  - 4.3. Absurde
  - 4.4. C.N., C.S.
  - 4.5. Contre-exemple
  - 4.6. Récurrence
  - 4.7. Algorithmique

# Implication

Considérons deux assertions  $A$  et  $B$ . On veut démontrer que  $A \Rightarrow B$ .

**Savoir-faire.**  $A \Rightarrow B$ . Raisonnement direct

On suppose  $A$  vraie, par une succession d'implications connues (calculs, résultats de théorèmes...), on prouve qu'alors  $B$  est vraie.

## Exercice

Montrer que si  $(x, y)$  est élément de  $]0, 2[ \times ]-2, 0[$  alors

$$\frac{1}{x} - \frac{1}{y} > 1$$

## Exercice

Soit  $f$  la fonction définie sur  $\mathbb{R}$  par  $f(x) = |x + \frac{3}{2}| - \frac{1}{2}$ . Montrer que

$$x \geq -1 \Rightarrow f(x) \geq 0$$

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1 ⇒ ?

4.2. ⇐ ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Implication avec disjonction de cas

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

On peut-être amené à faire une disjonction de cas :

## Exercice

Compléter l'énoncé suivant pour que la démonstration nécessite l'étude de deux cas séparés.

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

On peut-être amené à faire une disjonction de cas :

## Exercice

Compléter l'énoncé suivant pour que la démonstration nécessite l'étude de deux cas séparés.

Soit  $f$  la fonction définie par  $f : x \mapsto |x + \frac{3}{2}| + |x - 1| - \frac{1}{2}$

Montrer que  $x \geq -1 \Rightarrow f(x) \geq 0$ .

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

# Le chemin de la démonstration

## Heuristique. Démontrer que ...

Lorsqu'on cherche à démontrer un résultat, il y a fondamentalement deux attentes :

1. Trouver la (une) démonstration, satisfaisante i.e. qui donne la certitude du fait
2. Ecrire la démonstration, de sorte que toute personne qui lise la démonstration soit également persuadé du fait

Avec le temps du passage de 1 à 2, il faut donc trois temps dans la recherche d'une démonstration.

## Attention. Premier piège

Le temps 1 est le temps de l'analyse.

Le temps 2 est le temps de la synthèse.

Ce sont deux choses très différentes. Lorsque vous lisez une démonstration d'un théorème faite par un professeur, un corrigé dans un livre. . .vous ne voyez que le second temps, celui de la synthèse. Après la lecture, vous pouvez vous dire : « et oui, je vois que c'est vrai », mais vous n'avez pas appris *comment on trouve* la démonstration !!!

La seule solution est de chercher, chercher, chercher. . .et de ne pas se précipiter sur la (une) solution.

De même, si pour vous écrire une démonstration de cours lors d'une colle est uniquement un exercice de mémoire, alors c'est que vous n'avez pas compris le premier point, ni le point  $1 \rightarrow 2$  de la démonstration du fait considéré. *Un exemple ?*

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant & ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

## **Analyse** La métaphore du petit poucet

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmie

## Analyse La métaphore du petit poucet

### *Besoin de définitions*

*$f$  est surjective de  $E$  sur  $F$  si  $\forall y \in F, \exists x \in E$  tel que  $y = f(x)$ .*

*$f$  est injective de  $E$  (sur  $F$ ) si  $\forall x_1, x_2 \in E,$*

*$f(x_1) = f(x_2) \Leftrightarrow x_1 = x_2.$*

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1 ⇒ ?

4.2. ⇔ ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

## Analyse La métaphore du petit poucet

### *Besoin de définitions*

$f$  est surjective de  $E$  sur  $F$  si  $\forall y \in F, \exists x \in E$  tel que  $y = f(x)$ .

$f$  est injective de  $E$  (sur  $F$ ) si  $\forall x_1, x_2 \in E$ ,

$$f(x_1) = f(x_2) \Leftrightarrow x_1 = x_2.$$

### **Exemple** Exercice « de base »

Soit  $E$  un ensemble. Soit  $f : E \rightarrow E$ .

On suppose que  $f$  est surjective, montrer que  $f^2 (= f \circ f)$  est surjective.

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1 ⇒ ?

4.2. ⇐ ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

## Analyse La métaphore du petit poucet

### *Besoin de définitions*

$f$  est surjective de  $E$  sur  $F$  si  $\forall y \in F, \exists x \in E$  tel que  $y = f(x)$ .

$f$  est injective de  $E$  (sur  $F$ ) si  $\forall x_1, x_2 \in E,$

$f(x_1) = f(x_2) \Leftrightarrow x_1 = x_2.$

### **Exemple** Exercice « de base »

Soit  $E$  un ensemble. Soit  $f : E \rightarrow E.$

On suppose que  $f$  est surjective, montrer que  $f^2 (= f \circ f)$  est surjective.

### Exercice

Soit  $E$  un ensemble. Soit  $f : E \rightarrow E.$

On suppose que  $f^3 = f$ . Montrer que si  $f$  est injective alors  $f$  est surjective. Et réciproquement.

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1 ⇒ ?

4.2. ⇐ ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

## Proposition - Contraposée

$(\text{non } B \Rightarrow \text{non } A)$  s'appelle la contraposée de  $(A \Rightarrow B)$ .

Il est équivalent de prouver l'une ou l'autre de ces deux implications

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

## Proposition - Contraposée

$(\text{non } B \Rightarrow \text{non } A)$  s'appelle la contraposée de  $(A \Rightarrow B)$ .

Il est équivalent de prouver l'une ou l'autre de ces deux implications

## Démonstration

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

# Utilisation de la contraposée

Savoir-faire.  $A \Rightarrow B$ . Raisonnement par contraposée

On suppose donc  $B$  fausse et on prouve qu'alors  $A$  est fausse, comme précédemment

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

# Utilisation de la contraposée

## Savoir-faire. $A \Rightarrow B$ . Raisonnement par contraposée

On suppose donc  $B$  fausse et on prouve qu'alors  $A$  est fausse, comme précédemment

Le résultat de l'exercice suivant sera fréquemment exploité en analyse :

### Exercice

On considère un nombre réel  $x \geq 0$ . Montrer que

$$(\forall \epsilon > 0, 0 \leq x \leq \epsilon) \Rightarrow x = 0.$$

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

⇒ **Prendre quelques principes de base de démonstration**

⇒ **Deux raisonnements en particulier**

1. Cours mathématiques
2. Quantificateurs et notations ensemblistes
3. Vocabulaire sur les assertions
4. Principales méthodes de démonstration
  - 4.1. Démonstration d'une implication
  - 4.2. Démonstration d'une équivalence
  - 4.3. Raisonnement par l'absurde
  - 4.4. Conditions nécessaire, suffisante
  - 4.5. Contre-exemple
  - 4.6. Démonstration par récurrence
  - 4.6. Démonstration par algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Une première méthode

Deux possibilités pour prouver l'équivalence  $A \Leftrightarrow B$  :

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Une première méthode

Deux possibilités pour prouver l'équivalence  $A \Leftrightarrow B$  :

Savoir-faire.  $A \Leftrightarrow B$ . On procède en deux temps

1. On montre  $A \Rightarrow B$
2. On montre  $B \Rightarrow A$

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Une première méthode

Deux possibilités pour prouver l'équivalence  $A \Leftrightarrow B$  :

**Savoir-faire.**  $A \Leftrightarrow B$ . On procède en deux temps

1. On montre  $A \Rightarrow B$
2. On montre  $B \Rightarrow A$

## Exercice

On considère une fonction  $f$  définie sur  $\mathbb{R}$  à valeurs dans  $\mathbb{R}$ .

Montrer que

$(f \text{ est une fonction paire et impaire}) \Leftrightarrow (f \text{ est la fonction nulle}).$

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Une seconde méthode

Savoir-faire.  $A \Leftrightarrow B$ . On procède par équivalences connues successives.

Cette méthode est surtout utilisée pour des résolutions calculatoires.

Attention de ne pas en abuser : *il faut à chaque étape être sûr que l'on peut « remonter » les équivalences.*

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

# Une seconde méthode

Savoir-faire.  $A \Leftrightarrow B$ . On procède par équivalences connues successives.

Cette méthode est surtout utilisée pour des résolutions calculatoires.

Attention de ne pas en abuser : *il faut à chaque étape être sûr que l'on peut « remonter » les équivalences.*

## Exercice

Montrer que

$$\begin{cases} 2x + y = 2 \\ 3x + 4y = 3 \end{cases} \iff (x, y) = (1, 0)$$

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

Truc & Astuce pour le calcul. Ne pas abuser de  $\implies$

Il faut éviter le plus possible d'écrire  $\iff$  comme un tic de langage.

1. S'il n'est pas utile, on ne le note pas !
2. Si on choisit de le noter, on vérifie bien à chaque étape le sens  $\Leftarrow$  tout particulièrement

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

⇒ **Prendre quelques principes de base de démonstration**

⇒ **Deux raisonnements en particulier**

1. Cours mathématiques
2. Quantificateurs et notations ensemblistes
3. Vocabulaire sur les assertions
4. Principales méthodes de démonstration
  - 4.1. Démonstration d'une implication
  - 4.2. Démonstration d'une équivalence
  - 4.3. Raisonnement par l'absurde**
  - 4.4. Conditions nécessaire, suffisante
  - 4.5. Contre-exemple
  - 4.6. Démonstration par récurrence
  - 4.6. Démonstration par algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

**4.3. Absurde**

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

## Savoir-faire. Raisonement par l'absurde

Pour démontrer un certain énoncé, on fait l'hypothèse qu'il est faux et on aboutit à une contradiction.

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

## Savoir-faire. Raisonement par l'absurde

Pour démontrer un certain énoncé, on fait l'hypothèse qu'il est faux et on aboutit à une contradiction.

### Exercice

Montrer que le réel  $\sqrt{2}$  est irrationnel (i.e. n'est pas rationnel)

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

## Savoir-faire. Raisonement par l'absurde

Pour démontrer un certain énoncé, on fait l'hypothèse qu'il est faux et on aboutit à une contradiction.

### Exercice

Montrer que le réel  $\sqrt{2}$  est irrationnel (i.e. n'est pas rationnel)

**Remarque** Différence avec la contraposée

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

⇒ **Prendre quelques principes de base de démonstration**

⇒ **Deux raisonnements en particulier**

1. Cours mathématiques
2. Quantificateurs et notations ensemblistes
3. Vocabulaire sur les assertions
4. Principales méthodes de démonstration
  - 4.1. Démonstration d'une implication
  - 4.2. Démonstration d'une équivalence
  - 4.3. Raisonnement par l'absurde
  - 4.4. Conditions nécessaire, suffisante**
  - 4.5. Contre-exemple
  - 4.6. Démonstration par récurrence
  - 4.6. Démonstration par algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

**4.4. C.N., C.S.**

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Confusion fréquente. . .un peu de bon sens

Il s'agit simplement *d'un peu de bon sens* sur la signification des mots « nécessaire » et « suffisant ».

**Exemple**  $A$  : «  $x$  est le carré d'un entier » et  $B$  : «  $x \geq 0$  »

Qu'est-ce qui est nécessaire, qu'est-ce qui est suffisant (ou ne l'est pas) ?

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

## Confusion fréquente. . .un peu de bon sens

Il s'agit simplement *d'un peu de bon sens* sur la signification des mots « nécessaire » et « suffisant ».

**Exemple**  $A$  : «  $x$  est le carré d'un entier » et  $B$  : «  $x \geq 0$  »

Qu'est-ce qui est nécessaire, qu'est-ce qui est suffisant (ou ne l'est pas) ?

### Définition - Condition nécessaire. Condition suffisante

Plus généralement  $A \Rightarrow B$  peut se dire :

- ▶  $B$  est une condition nécessaire (**CN**) pour avoir  $A$  (puisque si on  $A$ , nécessairement on a  $B$ )
- ▶  $A$  est une condition suffisante (**CS**) pour avoir  $B$  (puisque'il suffit d'avoir  $A$  pour avoir  $B$ )

Rechercher une condition nécessaire et suffisante (**CNS**) pour avoir  $A$  revient donc à chercher  $B$  tel que  $A \Leftrightarrow B$ .

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

## Savoir-faire. Analyse-Synthèse

Certaines démonstrations, difficiles à gérer par équivalences, ou lorsque le résultat n'est pas donné, se font en deux phases :

1. phase d' "analyse" : on obtient une condition nécessaire (par implications successives par exemple) pour qu'une première hypothèse soit vérifiée
2. phase de "synthèse", ou phase de vérification : la condition précédemment obtenue est-elle suffisante ?

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

## Savoir-faire. Analyse-Synthèse

Certaines démonstrations, difficiles à gérer par équivalences, ou lorsque le résultat n'est pas donné, se font en deux phases :

1. phase d' "analyse" : on obtient une condition nécessaire (par implications successives par exemple) pour qu'une première hypothèse soit vérifiée
2. phase de "synthèse", ou phase de vérification : la condition précédemment obtenue est-elle suffisante ?

## Exercice

Montrer que toute fonction définie sur  $\mathbb{R}$  à valeurs dans  $\mathbb{R}$  s'écrit de manière unique comme somme d'une fonction paire et d'une fonction impaire.

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant & ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

# Analyse-Synthèse

## Exercice

Soit  $A, B, C$  et  $D$  quatre points du plan tel que  $AC = BD$  et  $(AB)$  non parallèle à  $(CD)$ .

Alors il existe une unique rotation du plan  $r$  tel que  $r(A) = B$  et  $r(C) = D$ .

Donner ses caractérisations

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1.  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

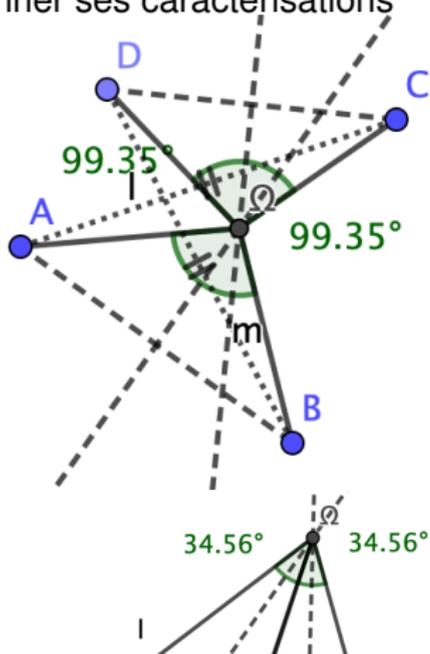
# Analyse-Synthèse

## Exercice

Soit  $A, B, C$  et  $D$  quatre points du plan tel que  $AC = BD$  et  $(AB)$  non parallèle à  $(CD)$ .

Alors il existe une unique rotation du plan  $r$  tel que  $r(A) = B$  et  $r(C) = D$ .

Donner ses caractérisations



⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1 ⇒ ?

4.2. ⇔ ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

⇒ **Prendre quelques principes de base de démonstration**

⇒ **Deux raisonnements en particulier**

1. Cours mathématiques
2. Quantificateurs et notations ensemblistes
3. Vocabulaire sur les assertions
4. Principales méthodes de démonstration
  - 4.1. Démonstration d'une implication
  - 4.2. Démonstration d'une équivalence
  - 4.3. Raisonnement par l'absurde
  - 4.4. Conditions nécessaire, suffisante
  - 4.5. Contre-exemple**
  - 4.6. Démonstration par récurrence
  - 4.6. Démonstration par algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

**4.5. Contre-exemple**

4.6. Récurrence

4.7. Algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

## Savoir-faire. Utilisation d'un contre-exemple

Lorsque l'on veut prouver qu'une implication est fausse, on cherche un exemple vérifiant l'hypothèse mais pas la conclusion, ce que l'on appelle un *contre-exemple*.

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmie

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

## Savoir-faire. Utilisation d'un contre-exemple

Lorsque l'on veut prouver qu'une implication est fautive, on cherche un exemple vérifiant l'hypothèse mais pas la conclusion, ce que l'on appelle un *contre-exemple*.

### Exercice

Soit  $f$  la fonction définie sur  $\mathbb{R}$  par  $f(x) = |x + \frac{3}{2}| - \frac{1}{2}$ . Montrer que  $(x \geq -1)$  et  $(f(x) \geq 0)$  ne sont pas équivalents.

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

⇒ **Prendre quelques principes de base de démonstration**

⇒ **Deux raisonnements en particulier**

1. Cours mathématiques
2. Quantificateurs et notations ensemblistes
3. Vocabulaire sur les assertions
4. Principales méthodes de démonstration
  - 4.1. Démonstration d'une implication
  - 4.2. Démonstration d'une équivalence
  - 4.3. Raisonnement par l'absurde
  - 4.4. Conditions nécessaire, suffisante
  - 4.5. Contre-exemple
  - 4.6. Démonstration par récurrence**
  - 4.6. Démonstration par algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

**4.6. Récurrence**

4.7. Algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

## Principe admis (axiome)

Soit  $P(n)$  (parfois notée  $\mathcal{P}_n$ ) une propriété portant sur l'entier  $n$ .

Si on a

$$\left\{ \begin{array}{l} P(0) \text{ vraie} \\ \forall n \in \mathbb{N}, (P(n) \text{ vraie} \Rightarrow P(n+1) \text{ vraie}) \end{array} \right.$$

alors  $P(n)$  est vraie pour tout entier  $n$ .

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

## Principe admis (axiome)

Soit  $P(n)$  (parfois notée  $\mathcal{P}_n$ ) une propriété portant sur l'entier  $n$ .

Si on a

$$\left\{ \begin{array}{l} P(0) \text{ vraie} \\ \forall n \in \mathbb{N}, (P(n) \text{ vraie} \Rightarrow P(n+1) \text{ vraie}) \end{array} \right.$$

alors  $P(n)$  est vraie pour tout entier  $n$ .

Un idée de démonstration ?

(A quoi est équivalent le principe de récurrence ?)

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmie

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

## Savoir-faire. Rédaction d'un raisonnement par récurrence

- Pour  $n = 0$ ,  $P(0)$  est vérifiée, avec vérification effective !  
(souvent deux simples calculs)
- Supposons la propriété vérifiée pour **un certain**  $n \geq 0$  (et surtout pas “pour tout”, parce que là, ce n'est plus la peine de faire une démonstration !).

⇒ Montrons que  $P(n + 1)$  est vraie.

Conclusion :  $\forall n \in \mathbb{N}, P(n)$  est vraie.

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

**Remarque** Démarrer à un autre nombre

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

# Quelques remarques

**Remarque** Démarrer à un autre nombre

Exercice

Montrer que pour tout  $n \in \mathbb{N}$ ,  $n < 2^n$ .

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

**Remarque** Démarrer à un autre nombre

Exercice

Montrer que pour tout  $n \in \mathbb{N}$ ,  $n < 2^n$ .

Exercice

Y a-t-il des erreurs dans les raisonnements suivants ? Où  
sont-elles ?

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

# Réurrences plus subtiles

## Savoir-faire. Réurrence à plusieurs pas (ou plusieurs termes)

Suivant la façon dont est énoncée la propriété de récurrence  $P(n)$  il peut être nécessaire

- d'initialiser la récurrence avec plusieurs ( $k$ ) valeurs de  $n$
- de supposer  $P(n), \dots, P(n+k-1)$  (il y en a aussi  $k$ ) vraies pour un certain  $n \geq 0$
- de prouver que ces  $k$  propriétés exactes entraînent  $P(n+k)$  vraie

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

## Réurrences plus subtiles

### Savoir-faire. Réurrence à plusieurs pas (ou plusieurs termes)

Suivant la façon dont est énoncée la propriété de récurrence  $P(n)$  il peut être nécessaire

- d'initialiser la récurrence avec plusieurs ( $k$ ) valeurs de  $n$
- de supposer  $P(n), \dots, P(n+k-1)$  (il y en a aussi  $k$ ) vraies pour un certain  $n \geq 0$
- de prouver que ces  $k$  propriétés exactes entraînent  $P(n+k)$  vraie

### Savoir-faire. Réurrence forte

- Pour  $n = 0$ ,  $P(0)$  est vérifiée (avec vérification effective !)
- Supposons la propriété vérifiée **jusqu'à** un certain  $n \geq 0$  (i.e  $P(0), P(1), \dots, P(n)$  vraies)
- Montrons que  $P(n+1)$  est vraie.

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Réurrence

4.7. Algorithmique

## Exercice

Soit  $(u_n)$  la suite définie par  $u_0 = \frac{2}{5}, u_1 = 1$  et pour tout entier naturel  $n$ ,

$$u_{n+2} = 5u_{n+1} - 6u_n.$$

Démontrer que pour tout  $n \in \mathbb{N}$ ,  $u_n = \frac{2^n + 3^n}{5}$ .

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Exercices

## Exercice

Soit  $(u_n)$  la suite définie par  $u_0 = \frac{2}{5}, u_1 = 1$  et pour tout entier naturel  $n$ ,

$$u_{n+2} = 5u_{n+1} - 6u_n.$$

Démontrer que pour tout  $n \in \mathbb{N}$ ,  $u_n = \frac{2^n + 3^n}{5}$ .

## Exercice

Montrer que tout entier  $n \geq 2$  se décompose en produit de nombres premiers.

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1.  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Exercices

## Exercice

Soit  $(u_n)$  la suite définie par  $u_0 = \frac{2}{5}, u_1 = 1$  et pour tout entier naturel  $n$ ,

$$u_{n+2} = 5u_{n+1} - 6u_n.$$

Démontrer que pour tout  $n \in \mathbb{N}$ ,  $u_n = \frac{2^n + 3^n}{5}$ .

## Exercice

Montrer que tout entier  $n \geq 2$  se décompose en produit de nombres premiers.

## Exercice

Montrer qu'un changement d'hypothèse de récurrence ramène une récurrence à plusieurs pas ou une récurrence forte à une récurrence faible.

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1.  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

⇒ **Prendre quelques principes de base de démonstration**

⇒ **Deux raisonnements en particulier**

1. Cours mathématiques
2. Quantificateurs et notations ensemblistes
3. Vocabulaire sur les assertions
4. Principales méthodes de démonstration
  - 4.1. Démonstration d'une implication
  - 4.2. Démonstration d'une équivalence
  - 4.3. Raisonnement par l'absurde
  - 4.4. Conditions nécessaire, suffisante
  - 4.5. Contre-exemple
  - 4.6. Démonstration par récurrence
  - 4.6. Démonstration par algorithme

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

## Heuristique. Exploitation d'un algorithme

Un algorithme peut permettre de démontrer, constructivement, l'existence d'un certain objet (ou d'une certaine fonction).

La difficulté est plutôt de démontrer que l'algorithme :

- ▶ se termine bien
- ▶ réalise bien ce que l'on désire

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Algorithme

## Heuristique. Exploitation d'un algorithme

Un algorithme peut permettre de démontrer, constructivement, l'existence d'un certain objet (ou d'une certaine fonction).

La difficulté est plutôt de démontrer que l'algorithme :

- ▶ se termine bien
- ▶ réalise bien ce que l'on désire

## Définition - Algorithme

Un algorithme est une suite finie de règles à appliquer dans un ordre déterminé à un nombre fini de données pour arriver avec certitude (c'est-à-dire sans indétermination ou sans ambiguïté), en un nombre fini d'étapes, à un certain résultat et cela indépendamment des données.

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Terminaison de l'algorithme

Pour démontrer que l'algorithme termine (que le nombre d'étapes est fini), on exploite un variant de boucle, en règle général, en suite d'entiers décroissante. Pour des boucles `for`, la terminaison de boucle est en règle générale immédiate.

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Terminaison de l'algorithme

Pour démontrer que l'algorithme termine (que le nombre d'étapes est fini), on exploite un variant de boucle, en règle général, en suite d'entiers décroissante. Pour des boucles `for`, la terminaison de boucle est en règle générale immédiate.

## Savoir-faire. Démontrer qu'une boucle se termine

On identifie une expression (variable) qui :

- ▶ est entière
- ▶ décroît strictement à chaque étape de la boucle

Alors, nécessairement, la boucle se termine.

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Terminaison de l'algorithme

Pour démontrer que l'algorithme termine (que le nombre d'étapes est fini), on exploite un variant de boucle, en règle général, en suite d'entiers décroissante. Pour des boucles `for`, la terminaison de boucle est en règle générale immédiate.

## Savoir-faire. Démontrer qu'une boucle se termine

On identifie une expression (variable) qui :

- ▶ est entière
- ▶ décroît strictement à chaque étape de la boucle

Alors, nécessairement, la boucle se termine.

## Exemple Division euclidienne

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

## Exercice

On considère le bout de programme suivant :

```
c=0
while p>0 :
    if c==0 :
        p=p-2
        c=1
    else :
        p=p+1
        c=0
```

1. Que fait ce programme ?
2. Les variables  $p$  et  $c$  sont-elles décroissantes, strictement ?
3. Montrer que la variable  $t=2p+3c$  est entière, strictement décroissante.

En déduire la terminaison de l'algorithme.

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Correction de l'algorithme

Il faut aussi savoir **démontrer** que le programme (avec de nombreuses répétitions de la boucle) réalise se que l'on souhaite. On utilise alors pour cela des *invariant de boucles*.

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Correction de l'algorithme

Il faut aussi savoir **démontrer** que le programme (avec de nombreuses répétitions de la boucle) réalise se que l'on souhaite. On utilise alors pour cela des *invariant de boucles*.

## Savoir-faire. Usage d'un invariant de boucle pour démontrer le résultat attendu

Pour démontrer qu'une boucle réalise bien le résultat attendu,

1. on cherche une expression qui reste constante tout au long des calculs de la boucle ;
2. on calcule sa valeur initiale (avant le début de la boucle) ;
3. on en déduit sa valeur finale ;
4. enfin connaissant les valeurs finales des variables du système (testées pour la sortie de boucle), on en déduit la valeur de la variable retournée par le programme.

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Exemple d'invariant de boucle

## Exemple Retour sur la division euclidienne

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Exemple d'invariant de boucle

**Exemple** Retour sur la division euclidienne

## Exercice

Montrer que le programme

```
1 def somme2(n) :  
2     S=0  
3     for i in range(1,n+1):  
4         S=S+i**2  
5     return (S)
```

calcule bien  $\sum_{i=1}^{100} i^2$

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow ?$

4.2.  $\Leftrightarrow ?$

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

## Exemple d'invariant de boucle

**Exemple** Retour sur la division euclidienne

### Exercice

Montrer que le programme

```

1  def somme2(n) :
2      S=0
3      for i in range(1,n+1):
4          S=S+i**2
5      return (S)

```

calcule bien  $\sum_{i=1}^{100} i^2$

### Exercice

On cherche à écrire un programme qui calcul  $n!$ .

1. Ecrire un programme avec une boucle `while`.
2. **Démontrer** que le programme se termine bien.
3. **Démontrer** que le programme effectue bien ce que l'on souhaite.

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1 ⇒ ?

4.2. ⇔ ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

## Objectifs

- ⇒ Reprendre quelques principes de base de démonstration
- ⇒ Deux raisonnements en particulier

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

## Objectifs

⇒ Reprendre quelques principes de base de démonstration

- ▶ Implications, double implication et équivalence

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

# Conclusion

## Objectifs

⇒ Reprendre quelques principes de base de démonstration

- ▶ Implications, double implication et équivalence
- ▶ C.N. et C.S. Méthode analyse-synthèse.

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Conclusion

## Objectifs

⇒ Reprendre quelques principes de base de démonstration

- ▶ Implications, double implication et équivalence
- ▶ C.N. et C.S. Méthode analyse-synthèse.
- ▶ Contraposée voire raisonnement par l'absurde

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

## Objectifs

- ⇒ Reprendre quelques principes de base de démonstration
- ⇒ Deux raisonnements en particulier

⇒ Principes de  
démonstration

⇒ Deux  
raisonnements

1. Cours  
mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique

## Objectifs

- ⇒ Reprendre quelques principes de base de démonstration
- ⇒ Deux raisonnements en particulier
  - ▶ Récurrence (suite d'assertions impliquées successivement)

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmie

# Conclusion

## Objectifs

- ⇒ Reprendre quelques principes de base de démonstration
- ⇒ Deux raisonnements en particulier
  - ▶ Récurrence (suite d'assertions impliquées successivement)
  - ▶ Utilisation d'un algorithme

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithme

# Conclusion

## Objectifs

- ⇒ Reprendre quelques principes de base de démonstration
- ⇒ Deux raisonnements en particulier

## Pour la prochaine fois

- ▶ Lecture du cours : chapitre 4. Nombres complexes
- ▶ Exercice n°220, 221, 233
- ▶ TD de jeudi :  
8h-10h : 74, 75, 79, 85b, 91  
10h-12h : 90, 76, 82, 85a, 93

⇒ Principes de démonstration

⇒ Deux raisonnements

1. Cours mathématiques

2. Quant& ens.

3. Assertions

4. Méthodes dem.

4.1  $\Rightarrow$  ?

4.2.  $\Leftrightarrow$  ?

4.3. Absurde

4.4. C.N., C.S.

4.5. Contre-exemple

4.6. Récurrence

4.7. Algorithmique