Compléments de Programmation

Benoit Donnet
Année Académique 2023 - 2024

LIÈGE université

Agenda

- Chapitre 1: Raisonnement Mathématique
- Chapitre 2: Construction de Programme
- Chapitre 3: Introduction à la Complexité
- Chapitre 4: Récursivité
- Chapitre 5: Types Abstraits de Données
- Chapitre 6: Listes
- Chapitre 7: Piles
- Chapitre 8: Files
- Chapitre 9: Elimination de la Récursivité

Agenda

- Chapitre 1: Raisonnement Mathématique
  - Rappels sur la Logique
  - Quantificateurs Logiques
  - Quantificateurs Numériques
  - Formalisation d'un Énoncé
  - Preuve par Récurrence
  - Relation d'Ordre
  - Induction

Agenda

- Chapitre 1: Raisonnement Mathématique
  - Rappels sur la Logique
    - ✓ Notations Logiques
    - ✓ Opérateurs Logiques
    - ✓ Prédicat
  - Quantificateurs Logiques
  - Quantificateurs Numériques
  - Formalisation d'un Énoncé
  - Preuve par Récurrence
  - Relation d'Ordre
  - Induction

Notations Logiques

- Notations logiques
  - *True* est la propriété vraie
  - *False* est la propriété fausse
  - $\neg p$ signifie "non p"
  - $p \wedge q$ signifie "p et q"
  - $p \vee q$ signifie "p ou q"
  - $p \Rightarrow q$ signifie "p implique q"
    - ✓ si p est vraie, alors q aussi
    - ✓ si p est fausse, alors p ⇒ q est vraie, quelque soit q
  - $\forall x, p$ signifie "pour tout x, p est vrai"
    - ✓ en général, x apparaît dans p
  - $\exists x, p$ signifie "il existe au moins un x tel que p est vrai"
    - ✓ en général, x apparaît dans p

Opérateurs Logiques

- Rappel sur les opérateurs logiques

p	q	¬ p	p ∧ q	p ∨ q	p ⇒ q	p ⇔ q
True	True	False	True	True	True	True
True	False	False	False	True	False	False
False	True	True	False	True	True	False
False	False	True	False	False	True	True

Opérateurs Logiques (2)

- Propriétés
  1. *commutativité*
     - ✓ $p \wedge q == q \wedge p$
     - ✓ $p \vee q == q \vee p$
  2. *tiers exclu*
     - ✓ $p \vee \neg p ==$ True
  3. *contradiction*
     - ✓ $p \wedge \neg p ==$ False
  4. *associativité*
     - ✓ $(p \wedge q) \wedge r == p \wedge (q \wedge r)$
     - ✓ $(p \vee q) \vee r == p \vee (q \vee r)$
  5. *distributivité*
     - ✓ $p \wedge (q \vee r) == (p \wedge q) \vee (p \wedge r)$
     - ✓ $p \vee (q \wedge r) == (p \vee q) \wedge (p \vee r)$

Opérateurs Logiques (3)

- Propriétés (cont.)
  6. *De Morgan*
     - ✓ $\neg(p \wedge q) == \neg p \vee \neg q$
     - ✓ $\neg(p \vee q) == \neg p \wedge \neg q$
  7. *simplification du* $\vee$
     - ✓ $p \vee p == p$
     - ✓ $p \vee$ False $= p$
     - ✓ $p \vee$ True $=$ True
     - ✓ $p \vee (p \wedge q) == p$
  8. *simplification du* $\wedge$
     - ✓ $p \wedge p == p$
     - ✓ $p \wedge$ False $=$ False
     - ✓ $p \wedge$ True $= p$
     - ✓ $p \wedge (p \vee q) = p$

Opérateurs Logiques (2)

- Propriétés (cont.)
  - $p \Rightarrow q == \neg p \vee q$
  - $p \wedge q \Rightarrow p ==$ True
    - ✓ valide quelque soit les valeurs de p et q
    - ✓ **tautologie**

Prédicat

- **Prédicat**
  - fonction produisant une valeur booléenne
- On peut exprimer beaucoup de choses sous la forme d'un prédicat
- Exemple

t:

0	4	4	6	8	12	15

- $\forall i, 0 \leq i \leq 6, \forall j, i < j \leq 6, t[i] \leq t[j]$
- $\forall\ i, 0 \leq i \leq 5, t[i] \leq t[i+1]$

Agenda

- Chapitre 1: Raisonnement Mathématique
  - Rappels sur la Logique
  - Quantificateurs Logiques
    - ✓ Quantificateur Universel
    - ✓ Quantificateur Existentiel
    - ✓ Variable Libre/Liée
  - Quantificateurs Numériques
  - Formalisation d'un Énoncé
  - Preuves par Récurrence
  - Relation d'Ordre
  - Induction

Quant. Universel

- Soient
  - P, Q, deux prédicats
  - (l), une liste (non vide) d'identificateurs
- $\forall\ (l)\ \cdot\ (P \Rightarrow Q)$
  - prédicat
  - **quantification universelle**
- Dans le cas où (l) se réduit à un seul identificateur i
  - $\forall\ i\ \cdot\ (P \Rightarrow Q)$
- Interprétation
  - pour tout i tel que P alors Q (est vrai)

Quant. Universel (2)

- Exemple
  - soit T, un tableau de N valeurs entières positives
  - $\forall\ i\ \cdot\ (i \in 0\ \ldots\ N-1 \Rightarrow T[i] > 0)$
- Lecture
  - pour tout i de l'intervalle 0 ... N-1, $T[i]$ est strictement positif
  - tous les éléments de T sont strictement positifs
- Le quantificateur ∀ est en quelque sorte une généralisation de l'opérateur ∧
  - $T[0] > 0 \wedge T[1] > 0 \wedge \ldots \wedge T[N-1] > 0$

Quant. Universel (3)

- Quelques propriétés
  1. $\forall\ i \cdot (False \Rightarrow P) == True$
  2. $\forall\ i \cdot (P \wedge Q \Rightarrow R) == \forall\ i \cdot (P \Rightarrow (Q \Rightarrow R))$
  3. $\forall\ i \cdot (P \Rightarrow Q) \wedge \forall\ i \cdot (P \Rightarrow R) == \forall\ i \cdot (P \Rightarrow (Q \wedge R))$
  4. $\forall\ i \cdot (P \Rightarrow Q) \wedge \forall\ i \cdot (R \Rightarrow Q) == \forall\ i \cdot ((P \vee R) \Rightarrow Q)$
  5. $\forall\ i \cdot (P \Rightarrow Q) \vee \forall\ j \cdot (R \Rightarrow S) == \forall\ (i, j) \cdot ((P \wedge R) \Rightarrow (Q \vee S))$

Quant. Universel (4)

- La première règle peut s'interpréter comme suit
  - lorsque la variable prend ses valeurs sur l'ensemble vide, la quantification est vraie, quel que soit le terme P
- Ainsi
  - $\forall\ i \cdot (i \in 0\ \ldots\ -1 \Rightarrow T[i] = 0) == True$
  - $\forall\ i \cdot (i \in 0\ \ldots\ -1 \Rightarrow T[i] \neq 0) == True$

Quant. Universel (5)

- Dans le cadre du cours, nous allons plutôt utiliser la notation suivante:
  - $\forall\ i, P, Q$
- Exemples
  - $\forall\ i,\ i \in 0\ \ldots\ N-1,\ T[i] > 0$
  - $\forall\ i,\ 0 \leq i \leq N-1,\ T[i] > 0$

Quant. Existentiel

- Soient
  - P, Q, deux prédicats
  - (l), une liste (non vide) d'identificateurs
- $\exists\ (l) \cdot (P \wedge Q)$
  - prédicat
  - **<u>quantification existentielle</u>**
- Dans le cas où (l) se réduit à un seul identificateur i
  - $\exists\ i \cdot (P \wedge Q)$
- Interprétation
  - il existe (au moins) un i tel que P, pour lequel Q (est vrai)

Quant. Existentiel (2)

- Exemple
  - soit T, un tableau de N valeurs entières dont au moins une est positive
  - $\exists\ i \cdot (i \in 0\ \ldots\ N-1 \wedge T[i] > 0)$
- Lecture
  - il existe (au moins) un i de l'intervalle 0 ... N-1 tel que $T[i]$ est positif
- Le quantificateur ∃ est en quelque sorte une généralisation de l'opérateur ∨
  - $T[0] > 0 \vee T[1] > 0 \vee \ldots \vee T[N-1] > 0$

Quant. Existentiel (3)

- Quelques propriétés
  1. $\exists\ i \cdot (False \wedge P) = = False$
  2. $\exists\ i \cdot (P \wedge Q) \vee \exists\ i \cdot (P \wedge R) = = \exists\ i \cdot (P \wedge (Q \vee R))$
  3. $\exists\ i \cdot (P \wedge Q) \vee \exists\ i \cdot (R \wedge Q) = = \exists\ i \cdot ((P \vee R) \wedge Q)$
  4. $\exists\ i \cdot (P \wedge Q) \vee \exists\ j \cdot (R \wedge S) = = \exists\ (i, j) \cdot ((P \wedge R) \wedge (Q \wedge S))$

Quant. Existentiel (4)

- Les Règles de De Morgan se généralisent aux quantificateurs logiques
  - $\neg(\exists\ i \cdot (P \wedge Q)) = = \forall\ i \cdot (P \Rightarrow \neg Q)$
  - $\neg(\forall\ i \cdot (P \Rightarrow Q)) = = \exists\ i \cdot (P \wedge \neg Q)$

Quant. Existentiel (5)

- Dans le cadre du cours, nous allons plutôt utiliser la notation suivante:
  - $\exists\ i, P, Q$
- Exemples
  - $\exists\ i,\ i \in 0\ \ldots\ N-1, T[i] > 0$
  - $\exists\ i,\ 0 \leq i \leq N-1, T[i] > 0$

Variable Libre/Liée

- La présence des quantificateurs pose des problèmes relatifs au nom des variables
- Une variable est dite **liée** quand le nom que l'on utilise pour définir un objet n'a pas d'importance
  - il s'agit typiquement de variables introduites par des quantificateurs
  - exemple
    - ✓ $\forall\ x \cdot (x \Rightarrow y)$ a le même sens que $\forall\ z \cdot (z \Rightarrow y)$
- Une variable est dite **libre** quand le nom que l'on utilise pour définir un objet a de l'importance
  - il s'agit typiquement de variables non introduites par des quantificateurs
  - exemple
    - ✓ $\forall\ x \cdot (x \Rightarrow y)$ n'a pas le même sens que $\forall\ x \cdot (x \Rightarrow t)$

Variable Libre/Liée (2)

- Une même variable peut être libre et liée dans une formule
- Plus exactement, elle peut avoir des occurrences libres et des occurrences liées
- **<u>Occurrence</u>** d'une variable?
  - endroit où la variable apparaît dans la formule (sauf derrière un quantificateur)
- Exemple
  - $(\forall\ x, R(\mathbf{x})) \land P(\mathbf{x})$
    - ✓ occurrence liée de x
    - ✓ occurrence libre de x

Variable Libre/Liée (3)

- On dit qu'une variable est **<u>libre dans une formule</u>** si elle possède au moins une occurrence libre dans cette formule
- On dit qu'une variable est **<u>liée dans une formule</u>** si toutes les occurrences de la variable dans la formule sont liées

Agenda

- Chapitre 1: Raisonnement Mathématique
  - Rappels sur la Logique
  - Quantificateurs Logiques
  - Quantificateurs Numériques
    - ✓ Somme
    - ✓ Produit
    - ✓ Minimum/Maximum
    - ✓ Dénombrement
  - Formalisation d'un Énoncé
  - Preuve par Récurrence
  - Relation d'Ordre
  - Induction

Somme

- La somme d'une suite de termes est représentée à l'aide de la notation Σ
- Notation

$$\sum_{P} Q$$

- Exemples

$$0 + 1 + \ldots + N - 1 = \sum_{i \in 0 \ldots N-1} i$$

$$0 + 1 + \ldots + N - 1 = \sum_{i=0}^{N-1} i$$

Somme (2)

- Propriétés
  - $\sum_{i \in \varnothing} A = 0$
  - $\sum_{i=0}^{-1} A = 0$
  - $\sum_{P} Q + \sum_{P} R = \sum_{P} (Q + R)$

Produit

- Le produit d'une suite de termes est représenté à l'aide de la notation Π
- Notation

$$\prod_{P} Q$$

- Exemples

$$T[0] \times T[1] \times \ldots \times T[N-1] = \prod_{i \in 0 \ldots N-1} T[i]$$

$$T[0] \times T[1] \times \ldots \times T[N-1] = \prod_{i=0}^{N-1} T[i]$$

Produit (2)

- Propriétés
  - $\prod_{i \in \emptyset} A = 1$
  - $\prod_{i=1}^{0} A = 1$

Minimum/Maximum

- Le minimum d'une suite de termes est représenté à l'aide de la notation *min*
  - de manière identique, le maximum est représenté par la notation *max*
- Notations
  - $min_P Q$
  - $max_P Q$
- Exemples
  - $minimum\{T[0], T[1], ..., T[\mathrm{N}-1]\} == min_{i \in 0...N-1}(T[i])$
  - $maximum\{T[0], T[1], ..., T[\mathrm{N}-1]\} == max_{i \in 0...N-1}(T[i])$

Minimum/Maximum (2)

- Propriétés
  - $min_{x \in \varnothing} A == +\infty$
  - $min_{0 \leq i \leq -1} A_i == +\infty$
  - $min_{P \wedge Q} A == min(min_P A, min_Q A)$
  - $max_{x \in \varnothing} A == -\infty$
  - $max_{0 \leq i \leq -1} A_i == -\infty$
  - $max_{P \wedge Q} A == max(max_P A, max_Q A)$
  - $min_P A + max_P A == 0$

Dénombrement

- Si
  - i est une variable libre
  - P et Q des prédicats
- $\#i \cdot (P \mid Q)$
  - **quantificateur de dénombrement**
- Ce quantificateur se définit par
  - $\#i \cdot (P \mid Q) \equiv \Sigma_{P \wedge Q}(1)$
- Interprétation
  - le nombre de fois où le prédicat Q est vrai lorsque P (est vrai)

Dénombrement (2)

- Exemple
  - soit T, un tableau de N valeurs entières
  - $\#i \cdot (i \in 0 \ \dots \ N-1 \,|\, T[i] = 0)$
- Lecture
  - représente le nombre de 0 existants dans T

Agenda

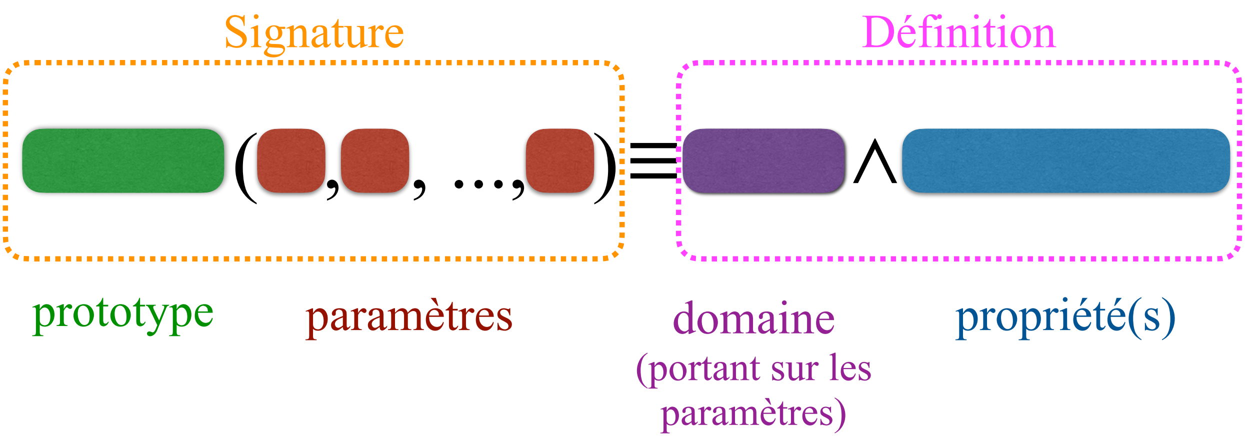
- Chapitre 1: Raisonnement Mathématique
  - Rappels sur la Logique
  - Quantificateurs Logiques
  - Quantificateurs Numériques
  - Formalisation d'un Énoncé
    - ✓ Principe
    - ✓ Format Général d'une Notation
    - ✓ Exemples
  - Preuve par Récurrence
  - Relation d'Ordre
  - Induction

Principe

- Il peut être intéressant de représenter une propriété à l'aide d'une notation mathématique
  - **<u>formalisation</u>** du problème
- La notation peut être réutilisée par la suite
  - sans se soucier de comment elle a été définie
- Intérêt?
  - condenser des expressions formelles
    - ✓ spécifications
    - ✓ Invariant Formel
    - ✓ cfr. Chap. 2

Format Général

- Le format général d'une notation mathématique est le suivant :



Exemples

- Exemple 1
  - un entier X est pair

$$\text{pair}(X) \equiv X \,\%\, 2 = 0$$

Exemples (2)

- Exemple 2
  - Toutes les valeurs d'un tableau sont strictement positives

$$\text{val\_pos}(T, N) \equiv \forall i,\ 0 \leq i \leq N-1,\ T[i] > 0$$

Exemples (3)

- Exemple 3
  - somme des éléments d'un sous-tableau

$$\text{som\_stab}(T, N, d, f) \equiv 0 \leq d \leq f \leq N-1 \wedge \sum_{i=d}^{f} T[i]$$

Agenda

- Chapitre 1: Raisonnement Mathématique
  - Rappels sur la Logique
  - Quantificateurs Logiques
  - Quantificateurs Numériques
  - Formalisation d'un Énoncé
  - Preuve par Récurrence
    - ✓ Principe
    - ✓ Exemples
  - Relation d'Ordre
  - Induction

Principe

- Raisonnement consistant à démontrer une propriété portant sur tous les *entiers naturels*
  - $\forall\ n,\ n \in \mathbb{N},\ P(n)$
- Exemple
  - $$P(n) = \sum_{i=1}^{n} i = \frac{n \times (n+1)}{2}$$
  - $$P(n) = 1 + 2 + \ldots + n = \frac{n \times (n+1)}{2}$$

Principe (2)

- Principe de la méthode
  1. prouver *P(0)*
     - ✓ **cas de base**
  2. $\forall\ n,\ n \in \mathbb{N},\ P(n) \Rightarrow P(n+1)$
     - ✓ prouver *P(n+1)* en utilisant *P(n)* comme hypothèse (de récurrence)
     - ✓ **étape de récurrence**

Exemples

- Preuve de $\sum_{i=1}^{n} i = \frac{n \times (n+1)}{2}$

1. cas de base

$$\sum_{i=1}^{0} i = 0$$

2. étape de récurrence

$$\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1)$$
$$= \frac{n \times (n+1)}{2} + (n+1)$$
$$= \frac{(n+1) \times (n+2)}{2}$$
$$= \frac{(n+1) \times ((n+1)+1)}{2}$$

Exemples (2)

- Proposition
  - $2^0 + 2^1 + \ldots + 2^n = 2^{n+1} - 1, \forall\, n \in \mathbb{N}$
- Preuve
  - Cas de base

$$2^0 = 1$$
$$= 2 - 1$$
$$= 2^{0+1} - 1$$

  - Etape de récurrence

$$2^0 + 2^1 + \ldots + 2^n + 2^{n+1} = (2^0 + 2^1 + \ldots + 2^n) + 2^{n+1}$$
$$= (2^{n+1} - 1) + 2^{n+1}$$
$$= 2 \times 2^{n+1} - 1$$
$$= 2^{(n+1)+1} - 1$$

Agenda

- Chapitre 1: Raisonnement Mathématique
  - Rappels sur la Logique
  - Quantificateurs Logiques
  - Quantificateurs Numériques
  - Formalisation d'un Énoncé
  - Preuve par Récurrence
  - Relation d'Ordre
    - ✓ Définitions de Base
    - ✓ Relation Binaire
    - ✓ Majorant/Minorant
    - ✓ Ensemble Bien Fondé
  - Induction

Définitions de Base

- Quelques définitions liées aux relations d'ordre
  - élément x appartenant à un ensemble E
    - ✓ $x \in E$
  - ensemble n'ayant aucun élément
    - ✓ **ensemble vide**
    - ✓ $\varnothing$
  - A est un **sous-ensemble** de B
    - ✓ A est inclus dans B
    - ✓ $A \subseteq B$
  - $P(E)$ **ensemble des parties de l'ensemble** E
    - ✓ $P(E) = \{A \mid A \subseteq E\}$
    - ✓ les éléments de $P(E)$ sont des ensembles
    - ✓ $P(E)$ contient toujours E et $\varnothing$
    - ✓ exemple pour $E = \{0, 1\}$
      - $P(E) = \{\{0\}, \{1\}, \{0, 1\}, \varnothing\}$

Définitions de Base (2)

- Quelques définitions liées aux relations d'ordre (suite)
  - le **produit cartésien** de deux ensembles, A et B
    - ✓ ensemble des couples (a, b) tels que $a \in A$ et $b \in B$
    - ✓ notation: $A \times B$
  - une **opération binaire** sur un ensemble E est une application de $E \times E \to E$
    - ✓ l'opération peut être associative, commutative, posséder un élément neutre
    - ✓ un ensemble muni d'une loi de composition interne associative et possédant un élément neutre est appelé un **monoïde**

Relation Binaire

- Une **relation binaire**, $\mathcal{R}$, sur E est une partie de $E \times E$ ou une application de $E \times E \longrightarrow \{True, False\}$
- Propriétés des relations binaires
  - *réflexive*
    - ✓ $\forall\, x \in E, x \mathcal{R} x$
  - *irréflexive*
    - ✓ $\forall\, x, y \in E, x \mathcal{R} y \Rightarrow x \neq y$
  - *symétrique*
    - ✓ $\forall\, x, y \in E, x \mathcal{R} y \Rightarrow y \mathcal{R} x$
  - *antisymétrique*
    - ✓ $\forall\, x, y \in E, x \mathcal{R} y \wedge y \mathcal{R} x \Rightarrow x = y$
  - *transitive*
    - ✓ $\forall\, x, y, z \in E, x \mathcal{R} y \wedge y \mathcal{R} z \Rightarrow x \mathcal{R} z$

Relation Binaire (2)

- Une relation **<u>d'équivalence</u>** est réflexive, symétrique, et transitive
- Une relation **<u>d'ordre large</u>** est réflexive, antisymétrique, et transitive
- Une relation **<u>d'ordre strict</u>** est irréflexive, antisymétrique, et transitive
- L'ordre est **<u>total</u>** lorsque 2 éléments quelconques de l'ensemble sont comparables par la relation
  - sinon l'ordre est **<u>partiel</u>**

Majorant/Minorant

- Soit E', une partie d'un ensemble ordonné $(E, \leq)$
  - $x \in E$ est un **<u>majorant</u>** de E' si $\forall\ y, y \in E', y \leq x$
  - un élément $y \in E$ qui n'a aucun majorant est dit élément **maximal**
  - $Maj(E')$ est l'ensemble des majorants de E'
  - $Maj(E') \cap E'$ a au plus un élément
    - ✓ si non vide, l'élément unique est appelé **maximum**
- On définit de manière identique minorant, élément minimal, minimum et Min(E')

Majorant/Minorant (2)

- On appelle **<u>borne supérieure</u>** x d'une partie E' le plus petit des majorants
  - $(\forall\ y,\ y \in E', y \leq x)$
  - $(\forall\ z,\ z \in E, (\forall\ y,\ y \in E', y \leq z) \Rightarrow x \leq z)$
- On appelle **<u>borne inférieure</u>** x d'une partie E' le plus grand des minorants

Ensemble Bien Fondé

- Une relation d'ordre sur un ensemble E est **<u>bien fondée</u>** si il n'y a pas de suite infinie strictement décroissante d'éléments de E
  - $\mathscr{R}$ est *bien fondée* ssi
    $\nexists\ x_1, x_2, \ldots, x_i, \ldots (x_i \in E)$
    telle que $x_{i+1}\ \mathscr{R}\ x_i, \forall\ i = 1,2,\ldots, i, \ldots$
  - notation:
    - ✓ $x < y$
    - ✓ $(E, <)$
- Un **<u>bon ordre</u>** est un ordre total bien fondé
- Exemple
  - $(\mathbb{N}, <)$
    - ✓ 237 > ... > 125 > ... > 12 > ... > 7 > ... > 0

Ensemble Bien Fondé (2)

- Un ensemble ordonné $(E, <)$ est **bien fondé** ssi tout sous-ensemble $S \subseteq E$ non vide ($S \neq \emptyset$) possède un élément minimum

Agenda

- Chapitre 1: Raisonnement Mathématique
  - Rappels sur la Logique
  - Quantificateurs Logiques
  - Quantificateurs Numériques
  - Formalisation d'un Énoncé
  - Preuve par Récurrence
  - Relation d'Ordre
  - Induction
    - ✓ Principe
    - ✓ Preuve par Induction
    - ✓ Induction Complète
    - ✓ Définition Inductive

Principe

- Si on a $(P(0), P(1), \ldots, P(n-1) \Rightarrow P(n)), \forall\, n \in \mathbb{N}$
- Alors $\forall\, n \in \mathbb{N}, P(n)$
- **Induction complète** (dans $\mathbb{N}$)
- Autre formulation
  - Si $\forall\, n \in \mathbb{N}, (\forall\, m, m \in \mathbb{N}, 0 \leq m < n, P(m)) \Rightarrow P(n)$
  - Alors $\forall\, n \in \mathbb{N}, P(n)$
- Pour prouver *P(n)*, on peut utiliser chacune des hypothèses
  - *P(0), P(1), …, P(n-1)*
  - pas de cas de base explicite

Preuve par Induction

- Exemple 1
  - $T(1) = 1$
  - $T(n) = 1 + T(1) + T(2) + \ldots + T(n\text{-}2) + T(n\text{-}1)$ $(n > 1)$
    - ✓ $T(1) = 1$
    - ✓ $T(2) = 2$
    - ✓ $T(3) = 4$
    - ✓ $T(4) = 8$
  - conjecture
    - ✓ $\forall\, n \geq 1 : T(n) = 2^{n-1}$

Preuve par Induction (2)

- Preuve
  - soit $n \geq 1$, quelconque
  - $n = 1$

    $T(n) = T(1) = 1 = 2^{1-1} = 2^0 = 2^{n-1}$
  - $n > 1$

    $T(n) = 1 + T(1) + T(2) + ... + T(n-1)$

    $= 1 + \boxed{2^0 + 2^1 + ... + 2^{n-2}}$ (n-1) × hypothèse d'induction

    $= 1 + (2^{(n-2)+1} - 1)$

    $= 2^{n-1}$

Preuve par Induction (3)

- Exemple 2
  - nombres de Fibonacci
    - ✓ $F_0 = 0$
    - ✓ $F_1 = 1$
    - ✓ $F_{i+2} = F_{i+1} + F_i,\ i \in \mathbb{N}$
  - nombre d'or
    - ✓ $G = \dfrac{1+\sqrt{5}}{2}$
    - ✓ propriétés
      1. $G^2 = G+1$
      2. $G^{i+2} = G^{i+1} + G^i$
  - proposition
    - ✓ $G^{n-1} \leq F_{n+1} \leq G^n, \forall\ n \in \mathbb{N}$

Preuve par Induction (4)

- Preuve?
- Idée

$F_n = \boxed{F_{n-1}} + \boxed{F_{n-2}}$

$P(n\text{-}1)$ $P(n\text{-}2)$

$F_{n+1} = \boxed{F_n} + \boxed{F_{n-1}}$

$\Rightarrow n \geq 2$

il faut traiter à part les cas n=0 et n=1

Preuve par Induction (5)

- Preuve (cont.)

1. $n = 0$

$G^{-1} \leq F_1 \leq G^0$?

$$\frac{2}{1+\sqrt{5}} \leq 1 \leq 1 \text{ ?}$$

2. $n = 1$

$G^0 \leq F_2 \leq G^1$?

$$1 \leq 1 \leq \frac{1+\sqrt{5}}{2} \text{ ?}$$

Preuve par Induction (6)

- Preuve (cont.)

  3. $n \geq 2$

G^{n-3}	$\leq$	F_{n-1}	$\leq$	G^{n-2}	(H.I. n-1 $\geq$ 1)
G^{n-2}	$\leq$	F_n	$\leq$	G^{n-1}	(H.I. $n \geq 1$)
$G^{n-2} + G^{n-3}$	$\leq$	$F_n + F_{n-1}$	$\leq$	$G^{n-1} + G^{n-2}$	(arithm.)
G^{n-1}	$\leq$	F_{n+1}	$\leq$	G^n	(prop. 2 de G, déf. F_{n+1}, $n \geq 1$)

Induction Complète

- Soit *(E, <)*, un ensemble bien fondé
  - on veut prouver $\forall\ x, x \in E, P(x)$
- Utilisation du principe d'induction
  - Si $\forall\ x, x \in E, (\forall\ y, y \in E, y \prec x, P(y)) \Rightarrow P(x)$
  - Alors $\forall\ x, x \in E, P(x)$
- Induction complète par rapport à une relation bien fondée

Définition Inductive

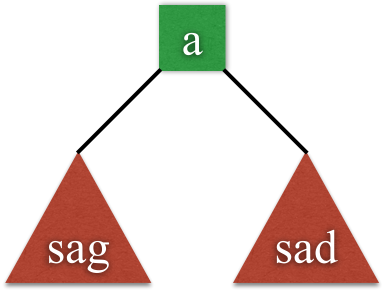
- Les définitions inductives sont très fréquentes en informatique
- La définition d'un ensemble X peut prendre la forme suivante
  - certains éléments de X sont donnés explicitement
  - les autres éléments sont donnés à partir d'éléments appartenant déjà à X
- Exemple
  - la partie X de $\mathbb{N}$ est définie par
    - ✓ $0 \in X$
    - ✓ $n \in X \Rightarrow n + 1 \in X$

Définition Inductive (2)

- Exemple 1
  - **<u>expression</u>** (cfr. INFO0946)
    - ✓ une variable, dénotée par son identificateur
      - moyennePoints
    - ✓ un littéral
      - 'a', 5, -45
    - ✓ un opérateur appliqué à une (ou deux) expression(s)
      - x++
      - a + (b/2)

Définition Inductive (3)

- Exemple 2
  - **<u>arbre binaire</u>** (définition inductive)
    - ✓ l'ensemble vide
      - $\varnothing$
    - ✓ un sommet
      - (a)
    - ✓ un sommet avec deux sous-arbres
      - (sag, a, sad)



Définition Inductive (4)

- On peut représenter l'ordre des opérations arithmétiques à l'aide d'un arbre binaire
  - $(x + y) \times (x - (y + z))$