

Maths For Fun

## 1,2,3, soleil ! Dessiner la récurrence

*Dominique Duval*

Université de Grenoble

12 octobre 2009

### III – Coinduction

# Plan

RAPPELS

LISTES et FLOTS

COINDUCTION

MÉMOIRE

## Axiomes de Peano (2nd ordre)

L'ensemble des **entiers naturels** est un ensemble (noté  $\mathbb{N}$ ) avec un élément (noté 0) et une fonction de  $\mathbb{N}$  vers  $\mathbb{N}$  (notée  $\text{suc}$ ), qui vérifient :

1.  $\forall x \in \mathbb{N}, \text{suc}(x) \neq 0.$
2.  $\forall x, y \in \mathbb{N}, \text{suc}(x) = \text{suc}(y) \Rightarrow x = y.$
3.  $\forall P \subseteq \mathbb{N},$   
**Initialisation.** si  $0 \in P$   
**Hérédité.** et si  $\forall x \in \mathbb{N} (x \in P \Rightarrow \text{suc}(x) \in P)$   
**Conclusion.** alors  $P = \mathbb{N}$

L'axiome (3) est l'**axiome de récurrence**.

Dans le cours 1 on a exprimé les axiomes de Peano comme une condition d'initialité.

# RÉCURRENCE = initialité pour les naturels

La **signature des naturels**  $\Sigma_{\text{nat}}$  :

$$\mathbb{I} \xrightarrow{z} \mathbb{N} \xleftarrow{s} \mathbb{N}$$

Le **modèle des naturels**  $M_{\text{nat}}$  :

$$\{*\} \xrightarrow{0} \mathbb{N} \xleftarrow{\text{suc}} \mathbb{N}$$

Le modèle  $M_{\text{nat}}$  de  $\Sigma_{\text{nat}}$  est **initial** :

Pour tout ensemble  $X$  avec  $a \in X$  et  $b : X \rightarrow X$ ,  
il existe une **unique** fonction  $f : \mathbb{N} \rightarrow X$  telle que  
 $f(0) = a$  et  $f(\text{suc}(n)) = b(f(n))$  pour tout  $n \in \mathbb{N}$ .

$$\begin{array}{ccccc} \{*\} & \xrightarrow{0} & \mathbb{N} & \xleftarrow{\text{suc}} & \mathbb{N} \\ \downarrow \text{id} & = & \downarrow f & = & \downarrow f \\ \{*\} & \xrightarrow{a} & X & \xleftarrow{b} & X \end{array}$$

# INDUCTION structurelle = initialité pour une signature

**Théorème d'initialité** Soit  $\Sigma$  une signature :

$$\dots \quad X \xrightarrow{f} Y \quad \dots$$

La signature  $\Sigma$  a un **modèle initial**  $M_0$ ,  
il est **unique à iso près**,  
et c'est le modèle des **termes clos** engendrés par  $\Sigma$ .

Pour tout modèle  $M$  de  $\Sigma$   
il existe un unique morphisme  $m : M_0 \rightarrow M$ .

$$\begin{array}{ccccc} \dots & M_0(X) & \xrightarrow{M_0(f)} & M_0(Y) & \dots \\ & \downarrow m(X) & & \downarrow m(Y) & \\ \dots & M(X) & \xrightarrow{M(f)} & M(Y) & \dots \end{array}$$

# INDUCTION structurelle : exemples

- ▶ Modèle initial d'une signature.

Ex. Naturels. Ex. Syntaxe d'un langage.

- ▶ Modèle initial d'une signature **paramétrée**.

Ex. Listes sur  $\mathbb{A}$ .

$$\begin{array}{ccccc} \{*\} & \xrightarrow{\text{empty}} & \mathbb{A}^* & \xleftarrow{\text{cons}} & \mathbb{A} \times \mathbb{A}^* \\ \downarrow \text{id} & = & \downarrow f & = & \downarrow \text{id} \times f \\ \{*\} & \xrightarrow{a} & E & \xleftarrow{b} & \mathbb{A} \times E \end{array}$$

Ex. Arbres binaires sur  $\mathbb{A}$ .

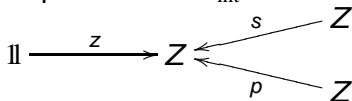
$$\begin{array}{ccccc} \{*\} & \xrightarrow{\text{emptyT}} & \mathbb{T} & \xleftarrow{\text{consT}} & \mathbb{A} \times \mathbb{T}^2 \\ \downarrow \text{id} & = & \downarrow f & = & \downarrow \text{id} \times f^2 \\ \{*\} & \xrightarrow{a} & E & \xleftarrow{b} & \mathbb{A} \times E^2 \end{array}$$

# INITIALITÉ pour une spécification équationnelle

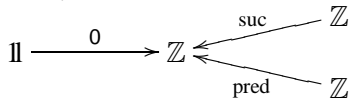
- Modèle initial d'une signature avec des équations.

Ex. "Récurrence" (sic...) sur les entiers relatifs.

Spécification équationnelle  $\Sigma_{\text{int}}$  :



Modèle initial  $M_{\text{int}}$  :



Pour tout ensemble  $X$  avec  $a \in X$ ,  $b : X \rightarrow X$  et  $c : X \rightarrow X$

tels que  $b(c(n)) = n$  et  $c(b(n)) = n$  pour tout  $n \in \mathbb{Z}$

il existe une unique fonction  $f : \mathbb{Z} \rightarrow X$  telle que

$f(0) = a$  et  $f(\text{suc}(n)) = b(f(n))$  et  $f(\text{pred}(n)) = c(f(n))$

pour tout  $n \in \mathbb{Z}$ .

# INITIALITÉ pour une spécification équationnelle

## Ex. Monoïdes

Le monoïde initial est  $\{*\}$  avec  $(*, *) \mapsto *$ .

## Ex. Anneaux (unitaires)

L'anneau initial est  $\mathbb{Z}$  avec  $+$  et  $\times$ .

## Ex. Anneaux avec $k$ éléments

Le modèle initial est  $\mathbb{Z}[X_1, \dots, X_k]$  avec  $+$  et  $\times$ .

Applications au **calcul formel** :

$$\mathbb{Z}[X_1, X_2] \rightarrow \mathbb{Q}(i, \sqrt{2}) \subseteq \mathbb{C}$$

# Plan

RAPPELS

**LISTES et FLOTS**

COINDUCTION

MÉMOIRE

# Listes : construction

Une **liste** sur  $\mathbb{A}$  est une liste finie d'éléments de  $\mathbb{A}$  :  
 $(x_1, x_2, \dots, x_n)$  pour un  $n \geq 0$

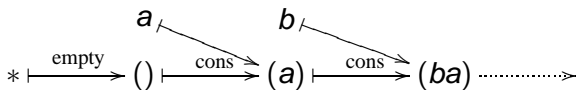
Notation :  $\mathbb{A}^*$

**Construction** des listes :

**empty** :  $\{*\} \rightarrow \mathbb{A}^*$  telle que  $* \mapsto ()$

**cons** :  $\mathbb{A} \times \mathbb{A}^* \rightarrow \mathbb{A}^*$  telle que  $(x, (x_1, \dots, x_n)) \mapsto (x, x_1, \dots, x_n)$

$$\{*\} \xrightarrow{\text{empty}} \mathbb{A}^* \xleftarrow{\text{cons}} \mathbb{A} \times \mathbb{A}^*$$



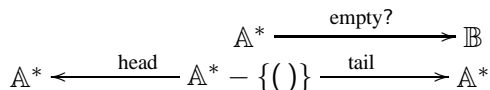
# Listes : destruction

**Destruction** ou **observation** des listes : ( $\mathbb{B} = \{0, 1\}$ )

**empty?** :  $\mathbb{A}^* \rightarrow \mathbb{B}$  telle que  $() \mapsto 1, (x, \dots) \mapsto 0$

**head** :  $\mathbb{A}^* - \{()\} \rightarrow \mathbb{A}$  telle que  $(x_1, x_2, \dots, x_n) \mapsto x_1$

**tail** :  $\mathbb{A}^* - \{()\} \rightarrow \mathbb{A}^*$  telle que  $(x_1, x_2, \dots, x_n) \mapsto (x_2, \dots, x_n)$



Pour simplifier on peut considérer les **flots**...

# Flots

Un **flot** (“**stream**”) sur  $\mathbb{A}$  est une liste infinie d’éléments de  $\mathbb{A}$  :

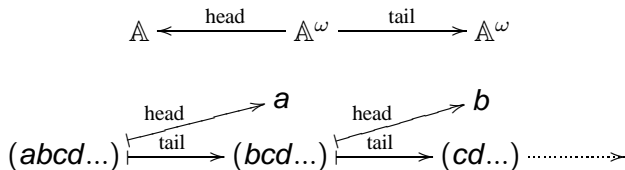
$(x_1, x_2, \dots, x_n, x_{n+1}, \dots)$

Notation :  $\mathbb{A}^\omega$

**head** :  $\mathbb{A}^\omega \rightarrow \mathbb{A}$  telle que  $(x_0, x_1, x_2, \dots) \mapsto x_0$

**tail** :  $\mathbb{A}^\omega \rightarrow \mathbb{A}^\omega$  telle que  $(x_0, x_1, x_2, \dots) \mapsto (x_1, x_2, \dots)$

Les fonctions **head** et **tail** servent à **observer** des flots :



# Symétrie ?

## Rappel.

Les *listes* forment le modèle *initial* de  $\Sigma_{\text{list}}$  :

$$\mathbb{1} \xrightarrow{e} L \xleftarrow{c} P \times L$$

d'où un raisonnement par *induction* sur les listes

## Théorème.

Les *flots* forment le modèle *terminal* de  $\Sigma_{\text{flot}}$  :

$$P \xleftarrow{h} F \xrightarrow{t} F$$

d'où un raisonnement par *coinduction* sur les flots

# Coinduction sur les flots

Pour tout ensemble  $X$  avec  $a : X \rightarrow \mathbb{A}$  et  $b : X \rightarrow X$ ,  
il existe une unique fonction  $F : X \rightarrow \mathbb{A}^\omega$  telle que  $\forall x \in X$

$$\boxed{\text{head}(F(x)) = a(x) \text{ et } \text{tail}(F(x)) = F(b(x))}$$

$$\begin{array}{ccccc} \mathbb{A} & \xleftarrow{a} & X & \xrightarrow{b} & X \\ \downarrow \text{id} & = & \downarrow F & = & \downarrow F \\ \mathbb{A} & \xleftarrow{\text{head}} & \mathbb{A}^\omega & \xrightarrow{\text{tail}} & \mathbb{A}^\omega \end{array}$$

Soit  $x \in X$ , on cherche  $F(x) = (y_1, y_2, \dots) \in \mathbb{A}^\omega$ .

$$\text{head}(F(x)) = a(x)$$

$$\text{donc } y_1 = a(x)$$

$$\text{tail}(F(x)) = F(b(x)) \text{ d'où}$$

$$\text{head}(\text{tail}(F(x))) = \text{head}(F(b(x))) = a(b(x))$$

$$\text{donc } y_2 = a(b(x))$$

...etc... (récurrence sur  $n$ ...)

$$\text{donc } y_n = a(b^{n-1}(x)) \text{ pour tout } n \geq 1.$$

# Plan

RAPPELS

LISTES et FLOTS

**COINDUCTION**

MÉMOIRE

# Induction et coinduction

- ▶ L'**induction** considère le **modèle initial** d'une signature.
- ▶ La **coinduction** considère le **modèle terminal** d'une signature.

# Modèle terminal

**Définition.** Un modèle  $M_0$  de  $\Sigma$  est **terminal** si pour tout modèle  $M$  de  $\Sigma$  il existe un unique morphisme  $m : M \rightarrow M_0$ .

$$\begin{array}{ccccc} \dots & M(X) & \xrightarrow{M(f)} & M(Y) & \dots \\ & \downarrow m(X) & & \downarrow m(Y) & \\ \dots & M_0(X) & \xrightarrow{M_0(f)} & M_0(Y) & \dots \end{array}$$

$=$

# Théorème de terminalité

Le “dual” du théorème d’initialité pour les **signatures** est très (trop) simple :

**Théorème de terminalité** (pour les signatures).

*Toute signature a un modèle terminal,  
c’est le modèle “des singletons”.*

Le “dual” du théorème d’initialité pour les **signatures paramétrées** est plus intéressant.

**Théorème de terminalité** (pour les signatures paramétrées).

*Une signature paramétrée a un modèle terminal  
si toutes ses opérations sont de la forme*

$$f_i : X \rightarrow P_i \times X^i \text{ avec } P_i \text{ paramètres et } i \in \mathbb{N}$$

## Exemple : flots

Signature  $\Sigma_{\text{flot}}$

$$P \xleftarrow{h} F \xrightarrow{t} F$$

$P$  est le type des **paramètres**

Soit  $\mathbb{A}$  l'ensemble des **arguments**

Le modèle  $M_{\text{flot}}$  des **flots** sur  $\mathbb{A}$

est **terminal** parmi les modèles de  $\Sigma_{\text{flot}}$  où  $P$  vaut  $\mathbb{A}$ .

$$\mathbb{A} \xleftarrow{\text{head}} \mathbb{A}^\omega \xrightarrow{\text{tail}} \mathbb{A}^\omega$$

# Un exercice (listes)

La signature  $\Sigma_{\text{list}}$  et son modèle **initial** :

$$\begin{array}{ccccc} \mathbb{I} & \xrightarrow{e} & L & \xleftarrow{c} & P \times L \\ \{*\} & \xrightarrow{\text{empty}} & \mathbb{A}^* & \xleftarrow{\text{cons}} & \mathbb{A} \times \mathbb{A}^* \end{array}$$

**Question.** Soit un ensemble  $\mathbb{A}$  et une fonction  $\varphi : \mathbb{A} \rightarrow \mathbb{A}$ . Définir  $\Phi : \mathbb{A}^* \rightarrow \mathbb{A}^*$  telle que

$$\Phi(x_0, x_1, \dots, x_n) = (\varphi(x_0), \varphi(x_1), \dots, \varphi(x_n))$$

**Réponse.** Par **induction**.

$\Phi(\text{empty}) = \text{empty} \text{ et } \Phi(\text{cons}(x, l)) = \text{cons}(\varphi(x), \Phi(l)).$

$$\begin{array}{ccccc} \{*\} & \xrightarrow{\text{empty}} & \mathbb{A}^* & \xleftarrow{\text{cons}} & \mathbb{A} \times \mathbb{A}^* \\ \downarrow \text{id} & = & \downarrow \Phi & = & \downarrow \text{id} \times \Phi \\ \{*\} & \xrightarrow{\text{empty}} & \mathbb{A}^* & \xleftarrow{\text{cons} \circ (\varphi \times \text{id})} & \mathbb{A} \times \mathbb{A}^* \end{array}$$

# Un exercice (flots)

La signature  $\Sigma_{\text{flot}}$  et son modèle **terminal** :

$$\begin{array}{ccccc} P & \xleftarrow{h} & F & \xrightarrow{t} & F \\ \mathbb{A} & \xleftarrow{\text{head}} & \mathbb{A}^\omega & \xrightarrow{\text{tail}} & \mathbb{A}^\omega \end{array}$$

**Question.** Soit un ensemble  $\mathbb{A}$  et une fonction  $\varphi : \mathbb{A} \rightarrow \mathbb{A}$ .  
Définir  $\Phi : \mathbb{A}^\omega \rightarrow \mathbb{A}^\omega$  telle que

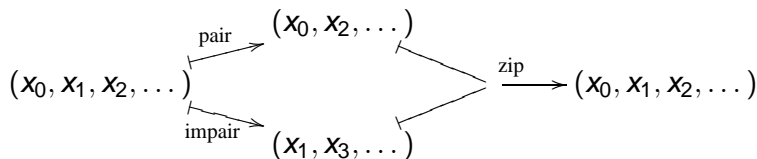
$$\Phi(x_0, x_1, \dots, x_n, \dots) = (\varphi(x_0), \varphi(x_1), \dots, \varphi(x_n), \dots)$$

**Réponse.** Par **coinduction**.

$$\text{head}(\Phi(s)) = \varphi(\text{head}(s)) \text{ et } \text{tail}(\Phi(s)) = \Phi(\text{tail}(s)).$$

$$\begin{array}{ccccc} \mathbb{A} & \xleftarrow{\varphi \circ \text{head}} & \mathbb{A}^\omega & \xrightarrow{\text{tail}} & \mathbb{A}^\omega \\ \downarrow \text{id} & = & \downarrow \Phi & = & \downarrow \Phi \\ \mathbb{A} & \xleftarrow{\text{head}} & \mathbb{A}^\omega & \xrightarrow{\text{tail}} & \mathbb{A}^\omega \end{array}$$

# Un exemple de coinduction



1. **Définir** les fonctions **pair** et **impair**.
2. **Prouver** deux lemmes :  $\text{pair}(\text{tail}(s)) = \text{impair}(s)$   
puis  $\text{impair}(\text{tail}(s)) = \text{tail}(\text{pair}(s))$  pour tout  $s \in \mathbb{A}^\omega$ .
3. **Définir** la fonction **zip**.
4. **Prouver** que  $\text{zip}(\text{pair}(s), \text{impair}(s)) = s$  pour tout  $s \in \mathbb{A}^\omega$ .

À terminer...

# Plan

RAPPELS

LISTES et FLOTS

COINDUCTION

**MÉMOIRE**

# La mémoire comme “boîte noire”

Dans le langage impératif **IMP**

toutes les variables sont de type entier

On note  $\mathbb{V}$  l'ensemble des **variables** (ou **identificateurs**)

Pour définir la sémantique de IMP on introduit l'ensemble  $\mathbb{S}$  des **états** de la mémoire avec

- ▶ une fonction **lookup** :  $\mathbb{S} \times \mathbb{V} \rightarrow \mathbb{Z}$   
pour **observer** la valeur d'une variable
- ▶ une fonction **update** :  $\mathbb{S} \times \mathbb{V} \times \mathbb{Z} \rightarrow \mathbb{S}$   
pour **modifier** la valeur d'une variable.

On a choisi  $\mathbb{S} = \mathbb{Z}^{\mathbb{V}}$  avec

$$\text{lookup}(s, X) = s(X)$$

$$\text{update}(s, X, n) = s[n/X]$$

**Pourquoi** ce choix ?

A priori, la mémoire est une “boîte noire” :  $\mathbb{S}$  est inconnu.

# Un modèle terminal pour observer la mémoire

$$X \in \mathbb{V} \longrightarrow \boxed{s \in \mathbb{S} ?} \longrightarrow n \in \mathbb{Z}$$

Signature  $\Sigma_{\text{mem}}$  :

$$\mathbb{S} \times \mathbb{V} \xrightarrow{!} \mathbb{Z}$$

avec  $V$  et  $Z$  paramètres, à interpréter par  $\mathbb{V}$  et  $\mathbb{Z}$ .

“Notre” modèle  $M_{\text{mem}}$  :

$$\mathbb{Z}^{\mathbb{V}} \times \mathbb{V} \xrightarrow{(s,X) \mapsto s(X)} \mathbb{Z}$$

**Proposition.** *Le modèle  $M_{\text{mem}}$  de  $\Sigma_{\text{mem}}$  est **terminal***

“**Donc**” le choix de  $\mathbb{S} = \mathbb{Z}^{\mathbb{V}}$  avec  $\text{lookup}(s, X) = s(X)$  est optimal pour **observer** l'état.

# Preuve de terminalité

**Proposition.** *Le modèle  $M_{\text{mem}}$  de  $\Sigma_{\text{mem}}$  est **terminal***

**Preuve.** Soit  $M$  :

$$\mathbb{S} \times \mathbb{V} \xrightarrow{\text{lookup}} \mathbb{Z}$$

On cherche  $f : \mathbb{S} \rightarrow \mathbb{Z}^{\mathbb{V}}$  tel que :

$$\begin{array}{ccc} \mathbb{S} \times \mathbb{V} & \xrightarrow{\text{lookup}} & \mathbb{Z} \\ \downarrow f \times \text{id} & & \downarrow \text{id} \\ \mathbb{Z}^{\mathbb{V}} \times \mathbb{V} & \xrightarrow{(s, X) \mapsto s(X)} & \mathbb{Z} \end{array}$$

c'est-à-dire  $f(s)(X) = \text{lookup}(s, X)$  pour tous  $s \in \mathbb{S}$  et  $X \in \mathbb{V}$ .  
Il existe un unique  $f$ , c'est  $f : s \mapsto (X \mapsto \text{lookup}(s, X))$ .

# Le modèle terminal pour modifier la mémoire

La signature  $\Sigma_{\text{mem}}$  et son modèle terminal  $M_{\text{mem}}$  :

$$\begin{array}{ccc} \mathbb{S} \times \mathbb{V} & \xrightarrow{I} & \mathbb{Z} \\ \mathbb{S} \times \mathbb{V} & \xrightarrow{\text{lookup}} & \mathbb{Z} \end{array}$$

où  $\mathbb{S} = \mathbb{Z}^{\mathbb{V}}$  et  $\text{lookup}(s, X) = s(X)$ .

**Proposition.** *Il existe une unique fonction*

*update :  $\mathbb{S} \times \mathbb{V} \times \mathbb{Z} \rightarrow \mathbb{S}$  telle que*

*lookup(update( $s, X, n$ ),  $Y$ ) = si ( $Y=X$ ) alors  $n$  sinon lookup( $s, Y$ )*

*et c'est la fonction update( $s, X, n$ ) =  $s[n/X]$ .*

“**Donc**” le choix de  $\mathbb{S} = \mathbb{Z}^{\mathbb{V}}$

avec  $\text{lookup}(s, X) = s(X)$  et  $\text{update}(s, X, n) = s[n/X]$

est optimal pour **observer** et pour **modifier** l'état.

## Preuve par coinduction

**Proposition.** *Il existe une unique fonction*

$\text{update} : \mathbb{S} \times \mathbb{V} \times \mathbb{Z} \rightarrow \mathbb{S}$  *telle que*

$\text{lookup}(\text{update}(s, X, n), Y) = \text{si } (Y=X) \text{ alors } n \text{ sinon } \text{lookup}(s, Y)$

**Preuve.** Soit  $M'$  :

$$\mathbb{S}' \times \mathbb{V} \xrightarrow{\text{lookup}'} \mathbb{Z}$$

où  $\mathbb{S}' = \mathbb{S} \times \mathbb{V} \times \mathbb{Z}$  et  $\text{lookup}' : \mathbb{S}' \times \mathbb{V} \rightarrow \mathbb{Z}$  :

$\text{lookup}'(s, X, n, Y) = \text{si } (Y=X) \text{ alors } n \text{ sinon } \text{lookup}(s, Y)$

Par terminalité, **il existe un unique**  $f : \mathbb{S}' \rightarrow \mathbb{S}$  tel que :

$$\begin{array}{ccc} \mathbb{S}' \times \mathbb{V} & \xrightarrow{\text{lookup}'} & \mathbb{Z} \\ \downarrow f \times \text{id} & & \downarrow \text{id} \\ \mathbb{S} \times \mathbb{V} & \xrightarrow{\text{lookup}} & \mathbb{Z} \end{array}$$

c'est-à-dire

$\text{lookup}(f(s, X, n), Y) = \text{si } (Y=X) \text{ alors } n \text{ sinon } \text{lookup}(s, Y)$