Définitions Inductives

Définitions inductives en informatique

- Syntaxe concrete
- Syntaxe abstraite
- Règles de typage
- Règles d'évaluation

Le principe

Une définition inductive est caracterisée par :

- Une ou plusieures assertions
- Un ensemble de règles d'inférence pour dériver ces assertions

Example :

- Assertion : "`X est naturel`" ou "`X nat`"
- Règles d'inférence :

R1: `0 est naturel`
R2: Si `n est naturel`, alors `succ(n) est naturel`.

Notation

Les règles d'inférence sont notées

$$\frac{\texttt{Hypothèse}_1 \ldots \texttt{Hypothèse}_n}{\texttt{Conclusion}} \text{(Nom de la règle)}$$

- `Conclusion` est une assertion
- $\texttt{Hypothèse}_1 \ldots \texttt{Hypothèse}_n$ sont des assertions
- En général $n \geq 0$. Si $n = 0$ la règle est un axiome

Remark : On pourra omettre quelques fois le nom de la règle

Exemple (règle unaire)

Les entiers naturels

$$\frac{}{0\ \texttt{est naturel}}\,(Nat0) \qquad \frac{n\ \texttt{est naturel}}{succ(n)\ \texttt{est naturel}}\,(Nat+)$$

Exemple (règle binaire)

Les arbres binaires

$$\frac{}{\textit{vide}\ \texttt{est un arbre binaire}}\ \text{(Abin-nil)}$$

$$\frac{A_1\ \texttt{est un arbre binaire} \qquad A_2\ \texttt{est un arbre binaire}}{\textit{node}(A_1, A_2)\ \texttt{est un arbre binaire}}\ \text{(Abin-ind)}$$

Exemple

Les mots sur un alphabet A

$$\frac{}{\epsilon\ \mathtt{mot}} \qquad \frac{a \in A \qquad n\ \mathtt{mot}}{a.n\ \mathtt{mot}}$$

Exemple (plusieurs axiomes, règles unaires et binaires)

Les expressions de la logique propositionnelle sur l'alphabet A

$$\frac{p \in A}{p \ \mathtt{expr}}$$

$$\frac{A_1 \ \mathtt{expr} \qquad A_2 \ \mathtt{expr}}{A_1 \vee A_2 \ \mathtt{expr}} \qquad \frac{A_1 \ \mathtt{expr} \qquad A_2 \ \mathtt{expr}}{A_1 \wedge A_2 \ \mathtt{expr}}$$

$$\frac{A_1 \ \mathtt{expr} \qquad A_2 \ \mathtt{expr}}{A_1 \supset A_2 \ \mathtt{expr}} \qquad \frac{A \ \mathtt{expr}}{\neg A \ \mathtt{expr}}$$

Exemple (plusieures assertions)

Les forêts de type `T`

$$\frac{}{avide \in \texttt{arbre T}} \qquad \frac{}{fvide \in \texttt{foret T}}$$

$$\frac{t \in \texttt{T} \qquad f \in \texttt{foret T}}{node(t, f) \in \texttt{arbre T}} \qquad \frac{A \in \texttt{arbre T} \qquad f \in \texttt{foret T}}{add(A, f) \in \texttt{foret T}}$$

Dérivation d'une assertion

Une assertion A est dérivable ssi

- A est un axiome

  $$\frac{}{A}$$

- ou il y a une règle de la forme

  $$\frac{A_1 \qquad\qquad A_n}{A}$$

  telle que $A_1, \ldots, A_n$ sont dérivables

Exercise :

1. Montrer que $succ(succ(succ(0)))$ `nat` est dérivable.
2. Donner le terme qui dénote la forêt suivante et montrer comment la construire avec les règles précédentes:

```
avide        3               6
           /   \            / | \
          44    6          6  7  8
           |    |         /   |   \
       avide  avide   avide   5    avide
                             / \
                        avide   avide
```

Ensemble inductif

Un ensemble inductif est le plus petit ensemble engendré par un système de règles d'inférence.

Preuves par Induction

Preuves par induction

- Induction sur les entiers
  - Induction mathématique
  - Induction complète
  - Équivalence
- Induction bien fondée
- Induction structurelle
- Induction sur un ensemble inductif

Induction sur les entiers I (induction mathématique)

Theorem

Soit P une propriété sur les entiers. Supposons

(CB) $P(0)$,

(CI) $\forall n \in \mathbb{N}. P(n) \supset P(n+1)$,

alors $\forall n \in \mathbb{N}. P(n)$

Exemples

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

Mais comment prouver

1. “Tout entier est décomposable en produit de nombres premiers”
2. “Si n est divisible par 3, alors $fib(n)$ est pair, sinon $fib(n)$ est impair”.

Induction sur les entiers II (induction complète)

Theorem

Soit P une propriété sur les entiers. Supposons

(CB) $P(0)$,

(CI) $((\forall k \in \mathbb{N}.(k < n \supset P(k))) \supset P(n))$,

alors $\forall n \in \mathbb{N}.P(n)$

Équivalence des deux principes

Malgré l'apparente supériorité du deuxième principe, on prouve

Theorem

Induction mathématique et complète sont équivalentes.

Théorème fondamental du cours

Theorem

Tous le monde est d'accord avec le professeur.

Proof.

On montre, par induction sur le nombre de personnes dans l'amphi, que tout groupe de n personnes contenant le professeur est d'accord avec lui.
Cas de base: il y a seulement le professeur, trivial.
Cas inductif: on suppose l'enoncé vrai pour tout groupe de n personnes, et on le prouve pour tout groupe de $n+1$.
Numérotons de 1 à $n+1$ les personnes en question, de façon que le professeur soit le numéro n, et considérons le groupe A des premières n et le groupe B des dernières n personnes.
Les deux groupes contiennent le professeur et sont de taille $n < n+1$, donc on peut appliquer l'hypothèse d'induction et en déduire qu'ils sont tous d'accord avec le professeur (qui est dans les deux), ce qui nous permet de conclure. $\square$

Corollary : Le professeur a toujours raison.

vrai ou faux?

Principe d’induction bien fondée

Un ensemble $\mathcal{A}$, un ordre strict $>$ et une propriété P sur $\mathcal{A}$

Principe d’induction :

Si

(CB) Pour tout élément minimal $y \in \mathcal{A}.P(y)$.

(CI) $((\forall z \in \mathcal{A}.(z < x \supset P(z))) \supset P(x))$

“le fait que $P(z)$ soit vérifiée pour tout élément $z{<}x$ implique $P(x)$”

alors

$\forall x \in \mathcal{A}.P(x)$

Ce principe est-il toujours bien défini?

Soit $>$ un ordre strict.

Theorem

Si $>$ est bien fondé, alors le principe d'induction est correct.

Theorem

Si le principe d'induction est correct, alors $>$ est bien fondé.

Corollary : Le principe d'induction est correct pour les ensembles inductifs.

Corollary : Le principe d'induction structurelle est correct.

Exemples

- Les mots :
  $P(m)$ est la propriété :
  $concat(concat(m, v_1), v_2), = concat(m, concat(v_1, v_2))$
- Les arbres binaires :
  $P(a)$ est la propriété : $feuilles(a) = noeuds\_internes(a) + 1$

Induction sur Quelques sur d'ordres bien fondés

- Ordre lexicographique
- Ordre multi-ensemble
- Combinaisons

Soit $>_{A_i}$ un ordre strict sur l'ensemble $\mathcal{A}_i$.
Ordre lexicographique sur le produit de 2 ensembles:

$$(x, y) >_{lex} (x', y') \text{ ssi } (x >_{A_1} x') \text{ ou } (x = x' \text{ et } y >_{A_2} y')$$

Example :

$$(4, "abc") >_{lex} (3, "abc") >_{lex} (2, "abcde") >_{lex} (2, "bcde") >_{lex}$$
$$(2, "e") >_{lex} (1, "e") >_{lex} (0, \epsilon)$$

Ordre lexicographique sur le produit de n ensembles

Si chaque $>_{A_i}$ est un ordre strict sur l'ensemble $\mathcal{A}_i$, alors $>_{lex}$ est un ordre strict qui permet de comparer deux n-uplets de la manière suivante:

$$(x_1, \ldots, x_n) >_{lex} (x'_1, \ldots, x'_n) \text{ ssi } \quad \begin{array}{l} \exists 1 \leq j \leq n \\ (x_j >_{A_j} x'_j \text{ and } \forall 1 \leq i < j \;\; x_i = x'_i) \end{array}$$

Theorem

Si chaque $>_{A_i}$ est un ordre strict bien fondé sur $\mathcal{A}_i$, alors l'ordre lexicographique $>_{lex}$ sur le produit de $\mathcal{A}_1 \times \ldots \times \mathcal{A}_n$ est un ordre strict bien fondé sur $\mathcal{A}_1 \times \ldots \times \mathcal{A}_n$.

Advertissement : $>_{lex}$ n'est pas l'ordre du dictionaire!!

Exemple : la fonction d'Ackerman

Montrer par induction que la fonction suivante termine.

Ackerman(0,n)	=	n+1
Ackerman(m+1,0)	=	Ackerman(m,1)
Ackerman(m+1,n+1)	=	Ackerman(m,Ackerman(m+1,n))

Definition

Soit $\mathcal{A}$ un ensemble. Un multi-ensemble de base $\mathcal{A}$ est une fonction $\mathcal{M} : \mathcal{A} \to \mathbb{N}$. Le multi-ensemble $\mathcal{M}$ est fini si $\mathcal{M}(x) > 0$ seulement pour un nombre fini d'éléments de $\mathcal{A}$.

Notation : $\{a, a, b\}$.

Ordres multi-ensembles

Definition

$\mathcal{M} >_{mul} \mathcal{N}$ ssi $\mathcal{N}$ s'obtient à partir de $\mathcal{M}$ en appliquant la règle suivante un nombre fini de fois : enlever un élément x de $\mathcal{M}$ et le remplacer par un nombre fini d'éléments plus petits que x (par rapport à l'ordre $>$).

Notation :

$$\{5, 3, 1, 1\}$$

Example :

$$\{5, 3, 1, 1\} >_{mul} \{4, 3, 3, 1\}$$

Theorem

Si $>_A$ est un ordre strict bien fondé sur $\mathcal{A}$, alors $>_{mul}$ est un ordre strict bien fondé sur les multi-ensembles de base $\mathcal{A}$.

Exemple

Un homme possède une somme d'argent en euros. Chaque jour il procède de la façon suivante:

- soit il jette une pièce de monnaie dans une fontaine,
- ou bien il change l'un de ses billets à la banque par un nombre arbitraire de pièces de monnaie de valeur quelconque.

Montrer que ce processus termine, c'est à dire, que dans un temps fini l'homme est ruiné.