Logical Foundations of CS – CS5303
$\mathfrak{I}$nduction

These notes were written using Mathématiques pour l'Informatique, by André Arnold and Irène Guessarian.

They complete the course given during the first week of class, on proofs, inductive definitions, and proofs by induction.

---

1 Inductive definitions

Let's review a couple of examples:

Example 1 *The part X of $\mathbb{N}$ inductively defined by:*

(B) $0 \in X$

(I) $n \in X \;\Rightarrow\; n+1 \in X$

is simply $\mathbb{N}$. Therefore (B) and (I) constitute an inductive definition of $\mathbb{N}$.

Example 2 *Let $A = \{(,)\}$ the alphabet made of both parentheses. The set $D \subseteq A^*$ of well-formed expressions with parenthesis, called the Dyck language, is defined by:*

(B) $\epsilon \in D$

(I) $x, y \in D \;\Rightarrow\; (x), xy \in D$

Example 3 *The set E of well parenthesed expressions, made of identifiers from A, and operators $+$ and $\times$, is the subset of $(A \cup \{+, \times\})^*$ inductively defined as follows:*

(B) $A \subseteq E$

(I) $e,\ f \in E \;\Rightarrow\; (e+f),\ (e \times f)\ \in E$

Example 4 *The set BT of binary trees whose elements are elements of alphabet A is the subset of $(A \cup \{\emptyset, (,), ,\})^*$ inductively defined as follows:*

(B) $\emptyset \in\ AB$ *(it is the empty tree)*

(I) $l,\ r\ \in AB \;\Rightarrow\; \forall a \in A,\ (a, l, r) \in AB$ *(the tree of root a, of left subtree l and right subtree r)*

Definition 1 *An inductive definition of a set X is non ambiguous is for all $x \in X$, there exist an unique way to get x by applying the inductive rules.*

Example 5 *The following inductive definition of $\mathbb{N}^2$ is ambiguous:*

(B) $(0,0) \in \mathbb{N}^2$

(I_1) $(n,m) \in \mathbb{N}^2 \;\Rightarrow\; (n+1,m) \in \mathbb{N}^2$

(I_2) $(n,m) \in \mathbb{N}^2 \;\Rightarrow\; (n,m+1) \in \mathbb{N}^2$

2 Proof by induction

Example 6 *Let's prove by induction that all strings in the Dyck language D contain as many "(" as ")". For all $x \in D$, $l(x)$ denotes the number of "(" and $r(x)$ denotes the number of ")". Let $P(x)$ the property we want to prove: $P(x) : l(x) = r(x)$.*

(B) the only element of the basis is ϵ, the empty string. So it satisfies P:

$$l(\epsilon) = r(\epsilon) = 0$$

(I) Let $x,\ y \in D$, such that $P(x)$ and $P(y)$. Let $z = xy$:

$$l(z) = l(x) + l(y) = r(x) + r(y) = r(z)$$

Therefore, $P(xy)$. Similarly we show that $P((x))$.

We deduce from the above that $\forall x \in D$, $l(x) = r(x)$.

3 Inductively defined functions

Example 7 *The factorial function: $\mathbb{N} \rightarrow \mathbb{N}$ is inductively defined by:*

(B) $fact(0) = 1$

(I) $fact(n+1) = (n+1) \times fact(n)$

Here we use the inductive definition of $\mathbb{N}$ as a support for the inductive definition of $fact$. We can also note it as follows:

$$fact(n) = \begin{cases} 1 & \textit{if } n = 0 \\ n \times fact(n-1) & \textit{otherwise} \end{cases}$$

Example 8 *Let us consider the set E inductively defined in Example 3. The expressions of E are in infix notation. Postfix notation can also be used to define parenthesed expressions. For instance, the postfix expression of $((a \times (b+c)) + d)$ is $abc + \times d+$.*

The transformation from infix to postfix can be inductively defined as follows:

(B) $\forall a \in A,\ post(a) = a$

(I) $\forall e,\ f \in E,\ post((e+f)) = post(e)post(f)+$, *and* $post((e \times f)) = post(e)post(f)\times$

Example 9 *The height of a binary tree can be inductively defined as follows:*

(B) $h(\emptyset) = 0$

(I) $\forall l,\ r \in AB$, $\forall a \in A$, $h((a,l,r)) = 1 + max(h(l), h(r))$

Example 10 *The in-order traversal of a binary tree is the sequence of values/elements in the tree read from left to right. Note: different trees can result in the same in-order traversal (exercise: find two such trees). The inductive definition of the in-order traversal is as follows:*

$$In - trav(x) = \begin{cases} \epsilon & \text{if } x = \emptyset \\ in - trav(l).a.in - trav(r) & \text{if } x = (a,l,r) \end{cases}$$

4 Set closure

Given the inductive definition of sets, we now know how to generate a set $C(E)$ given a set E and an operation/inductive rule C on E.

Let C be a monotonic application, i.e., $\forall A \subseteq A'$, $C(A) \subseteq C(A')$.

We say that a set A is C-closed if $C(A) \subseteq A$

Property 1 *Let I be a set of indices. Let A_i be C-closed, $\forall i \in I$. Then $\cap_{i \in I} A_i$ is also C-closed.*

Definition 2 *Let A be a subset of E. Then*

$$\bigcap_{A \subseteq B \subseteq E,\ B \text{ is } C\text{-closed}} B$$

is still C-closed, and contains A. It is denoted $\widehat{C}(A)$. It is the closure of A under C.

Property 2

- *If A' is C-closed and contains A, then $\widehat{C}(A) \subseteq A'$.*
- $A \subseteq \widehat{C}(A)$.
- $\widehat{C}(\widehat{C}(A)) = \widehat{C}(A)$.
- $A \subseteq A' \Rightarrow \widehat{C}(A) \subseteq \widehat{C}(A')$.

Exercise 1 *Let $E = \mathbb{N}$, and $C(A) = \{n+m \mid n \in A,\ m \in A\}$. Let $k\mathbb{N} = \{kn \mid n \in \mathbb{N}\}$. Show that: if $A \subseteq k\mathbb{N}$ then $\widehat{C}(A) \subseteq k\mathbb{N}$.*