20. STRUCTURAL INDUCTION

Mihai-Valentin DUMITRU
mihai.dumitru2201@upb.ro

January 2025

Structural induction is an extension of mathematical induction. We use mathematical induction to prove a property P of natural numbers (or of an infinite subset of the natural numbers, except the first few ones).

An induction proof consists of proving the *base case*, followed by the *inductive step*. These are two statements that require two different proofs.

The base case requires us to prove that the property holds for a particular number; usually small, very often 0. I.e. we need to prove $P(k)$ for some small $k \in \mathbb{N}$ (usually $k = 0$).

After having proven the base case, we need to prove the statement $P(n) \rightarrow P(n+1)$; i.e. "assuming an arbitrary natural number n has property P, it follows that $n+1$ also has this property".

These two proofs together shows that P holds for all natural numbers $\geq k$. Think back in the previous lecture, where we proposed the definition of a $Natural$ ADT for modelling natural numbers. We employed two constructors: $Zero$ and $Succ$, meaning that all natural numbers are either zero, or the successor of some other natural number.

It's important that *all* natural numbers can be constructed in this way. This is the key to mathematical induction: if the sentences $P(0)$ and $P(n) \rightarrow P(n+1)$ are correct (here, $+1$ should not be interpreted as "addition", but rather as the result of the successor function), then all numbers must have property P.

The key to extending induction to any ADT T is the following:

- take a base case for each nullary constructor, $C : T$; prove the sentence $P(C)$.
- take a base case for each external constructor, $E : A_1 \times A_2 \times \ldots \times A_n \rightarrow T$; prove the sentence $P(E(a_1, a_2, ..., a_n))$, where $a_1 \in A_1, a_2 \in A_2, ..., a_n \in A_n$ are arbitrary values of those types.
- for each internal constructor $I : T \times T \times \ldots \times T \times A_1 \times A_2 \times \ldots \times A_m \rightarrow T^\dagger$, assume the sentences $P(t_1), P(t_2), ..., P(t_n)$ (with $t_1, t_2, ..., t_n \in T$) and prove that under this assumption, $P(I(t_1, t_2, ..., t_n, a_1, a_2, ..., a_m))$, where $a_1 \in A_1, a_2 \in A_2, ..., a_m \in A_m$ are arbitrary values of those types.

  In other words, we are proving the sentence $P(t_1), P(t_2), ..., P(t_n) \rightarrow P(I(t_1, t_2, ..., t_n, a_1, a_2, ..., a_m))$.

Let's dive right in to some examples.

1 Natural numbers

In the previous lecture, we defined addition on natural numbers, using only two axioms. The first one tells us that any number added to zero equals to that other number; intuitively, we know that the symmetric sentence is also true: "zero added to any number equals that other number". But this is not one of the axioms. It doesn't have to be!

> **Theorem 20.1.** $P_1(n) \stackrel{\text{def}}{=} \texttt{add}(n, Zero) = n$
> $\forall n \in Natural, P_1(n)$

Proof. **Base case** ($n = Zero$): $\texttt{add}(Zero, Zero) \stackrel{ADD1}{=} Zero$

Inductive step We assume $\texttt{add}(n, Zero) = n$.

[†] Each internal constructor's domain must consist of a cartesian product with $n \geq 1$ occurrences of the defined ADT T, as well as $m \geq 0$ occurrences of some other ADT; to simplify the definition here, we moved all the occurrences of T at the beginning of the product, even though we usually allow them to be mixed among the occurrences of other ADTs.

We need to show $\texttt{add}(Succ(n), Zero) = Succ(n)$.

$\texttt{add}(Succ(n), Zero) \stackrel{ADD2}{=} Succ(\texttt{add}(n, Zero)) \stackrel{IH}{=} Succ(n)$.

□

The text above the equal sign shows the axiom which tells us that we can derive the right-hand-side from the left-hand-side. `IH` refers to the "induction hypothesis".

We can also prove the following:

Theorem 20.2. $P_2(n) \stackrel{\text{def}}{=} \forall m \in Natural \quad \texttt{add}(n, Succ(m)) = S(\texttt{add}(n, m))$
$\forall n \in Natural, P_2(n)$

Proof. **Base case** $(n = Zero)$: $\texttt{add}(Zero, Succ(m)) \stackrel{ADD1}{=} Succ(m) \stackrel{ADD1}{=} Succ(\texttt{add}(Zero, m))$

Note that the last equality is an abuse of our usual notation, because we actually derive the left-hand-side from the right-hand-side in a single step according to axiom `ADD1` (and not the other way around).

Inductive step We assume $\texttt{add}(n, Succ(m)) = Succ(\texttt{add}(n, m))$.

We need to show $\texttt{add}(Succ(n), S(m)) = Succ(\texttt{add}(Succ(n), m)$.

$\texttt{add}(Succ(n), Succ(m)) \stackrel{ADD2}{=} Succ(\texttt{add}(n, Succ(m))) \stackrel{IH}{=} Succ(Succ(\texttt{add}(n, m))) \stackrel{ADD2}{=} Succ(\texttt{add}(Succ(n), m))$ □

The commutativity of the addition operation is also familiar to us, yet not a part of the axioms that we defined. Again, this is actually a provable property.

Theorem 20.3. $\forall m, n \in Natural \quad \texttt{add}(m, n) = \texttt{add}(n, m)$

Note that our structural induction method can only be applied to *unary properties*, which say something about a particular value of the ADT.

What we need is to prove $P_{COMM}(n) : \forall m \in Natural \quad \texttt{add}(m, n) = \texttt{add}(n, m)$. We can prove this by structural induction on n, first showing $P_{COMM}(Zero)$, then $P_{COMM}(n) \rightarrow P_{COMM}(Succ(n))$.

The **base case** $P_{COMM}(Zero)$ is easy: we have to show that $\texttt{add}(m, Zero) = \texttt{add}(Zero, m)$; using axiom `ADD1` and the previously proven P_1, we can show both sides are equal to m.

For the **inductive step**, we assume $P_{COMM}(n)$ and show $P_{COMM}(Succ(n))$.

$P_{COMM}(Succ(n)) : \texttt{add}(m, Succ(n)) = \texttt{add}(Succ(n), m)$.

Using $P_2(m)$, we can reduce the left-hand-side to $Succ(\texttt{add}(m, n))$; we can reduce the right-hand-side to the same value, using the axiom `ADD1`.

Note that we "cheated" a bit, by anticipating important properties that are useful in our proof for commutativity. In practice, we might not know in advance, which properties are useful; drafting our proof, we get stuck on a particular property, which we should attempt to prove separately; we can then come back to our original proof, armed with a new theorem. The important thing is that, at the end, we should type out our proof nicely, reordering any intermediate results that we need.

2 Lists

Let's try to prove properties on lists. We will start with a simple one, similar to the first property we proved for the addition of $Natural$ numbers.

Theorem 20.4. $P_{l1}(l) : \texttt{append}(l, Empty) = l$

Proof. **Base case** $(l = Empty)$: $\texttt{append}(Empty, Empty) \stackrel{APP1}{=} Empty$

Inductive step We assume $\texttt{append}(l, Empty) = l$.

We need to show $\texttt{append}(Cons(x, l), Empty) = Cons(x, l)$.

$\texttt{append}(Cons(x, l), Empty) \stackrel{APP2}{=} Cons(x, \texttt{append}(l, Empty)) \stackrel{IH}{=} Cons(x, l)$

□

3 Binary Trees

Binary trees are a bit more interesting, because they require two hypotheses in the inductive step. Let's prove a very simple property about the relationship between the height and the size of a binary tree.

Theorem 20.5. $P_{SH}(t) \stackrel{\text{def}}{=} \texttt{height}(t) \leq \texttt{size}(t)$
$\forall t \in BTree, P_{SH}(t)$

Proof. **Base case** ($t = Nil$): $\texttt{height}(Nil) \stackrel{H1}{=} 0 \stackrel{SZ1}{=} \texttt{size}(Nil)$

Inductive step We assume $\texttt{height}(l) \leq \texttt{size}(l)$ and $\texttt{height}(r) \leq \texttt{size}(r)$.

We need to show $\texttt{height}(Node(e, l, r)) \leq \texttt{size}(Node(e, l, r))$.

$\texttt{height}(Node(e, l, r)) \stackrel{H2}{=} 1 + \max(\texttt{height}(l), \texttt{height}(r))$

Based on the two inductive hypothesis and the mathematical property (which we assume to be known):

$(a \leq b \wedge c \leq d) \rightarrow a + c \leq b + d$

we can conclude that the right-hand-side of our last relation is less than $1 + \max(\texttt{size}(l), \texttt{size}(r))$; but, using axiom SZ2, we can see this is exactly $\texttt{size}(Node(e, l, r))$.

□

Note the related, stronger property: $\texttt{height}(t) < \texttt{size}(t)$. The difference is that the sign is strict; this property is not true for Nil (for which both the height and size are 0), or for any leaf (for which both the height and size are 1), but it is true for any other tree.

So we could prove this property holds for almost all trees; in our structural induction proof, we just have to use trees with at least two levels as the base case.