4

Induction

> *Even though this proposition may have an infinite number of cases, I shall give a very short proof of it assuming two lemmas. The first, which is self evident, is that the proposition is valid for the second row. The second is that if the proposition is valid for any row then it must necessarily be valid for the following row.*
>
> — Blaise Pascal
> *Traité du Triangle Arithmetique*, c. 1654

This chapter discusses **induction**, a classic proof technique for proving first-order theorems with universal quantifiers. Section 4.1 begins with **stepwise induction**, which may be familiar to the reader from earlier education. Section 4.2 then introduces **complete induction** in the context of arithmetic. Complete induction is theoretically equivalent in power to stepwise induction but sometimes produces more concise proofs. Section 4.3 generalizes complete induction to **well-founded induction** in the context of arithmetic and recursive data structures. Finally, Section 4.4 covers a form of well-founded induction over logical formulae called **structural induction**. It is useful for reasoning about correctness of decision procedures and properties of logical theories and their interpretations.

We apply induction in various ways throughout the book. Structural induction is applied in proofs. Additionally, induction is the basis for the program verification methods of Chapter 5.

4.1 Stepwise Induction

We review stepwise induction for arithmetic and then show that it extends naturally to other theories, such as the theory of lists T_{cons}.

Arithmetic

Recall from Chapter 3 that the theory of Peano arithmetic T_{PA} formalizes arithmetic over the natural numbers. Its axioms include an instance of the (induction) axiom schema

$$F[0] \ \wedge \ (\forall n.\ F[n] \rightarrow F[n+1]) \ \rightarrow \ \forall x.\ F[x]$$

for each Σ_{PA}-formula $F[x]$ with only one free variable x. This axiom schema says that to prove $\forall x.\ F[x]$ — that is, $F[x]$ is T_{PA}-valid for all natural numbers x — it is sufficient to do the following:

- For the **base case**, prove that $F[0]$ is T_{PA}-valid.
- For the **inductive step**, assume as the **inductive hypothesis** that for some arbitrary natural number n, $F[n]$ is T_{PA}-valid. Then prove that $F[n+1]$ is T_{PA}-valid under this assumption.

These two steps comprise the **stepwise induction principle** for Peano (and Presburger) arithmetic.

Example 4.1. Consider the theory T^{+}_{PA} obtained from augmenting T_{PA} with the following axioms:

- $\forall x.\ x^0 = 1$ (exp. zero)
- $\forall x, y.\ x^{y+1} = x^y \cdot x$ (exp. successor)
- $\forall x, z.\ exp_3(x, 0, z) = z$ (exp_3 zero)
- $\forall x, y, z.\ exp_3(x, y+1, z) = exp_3(x, y, x \cdot z)$ (exp_3 successor)

The first two axioms define exponentiation x^y, while the latter two axioms define a ternary function $exp_3(x, y, z)$.

Let us prove that the following formula is T^{+}_{PA}-valid:

$$\forall x, y.\ exp_3(x, y, 1) = x^y \ . \tag{4.1}$$

We need to choose either x or y as the induction variable. Considering the exp_3 axioms, it appears that y is the smarter choice: (exp_3 successor) defines exp_3 recursively by considering the predecessor of $y+1$.

Therefore, we prove by stepwise induction on y that

$$F[y] : \ \forall x.\ exp_3(x, y, 1) = x^y \ .$$

For the **base case**, we prove

$$F[0] : \ \forall x.\ exp_3(x, 0, 1) = x^0 \ .$$

But $x^0 = 1$ by (exp. zero), and $exp_3(x, 0, 1) = 1$ by (exp_3 zero).

Assume as the inductive hypothesis that for arbitrary natural number n,

$$F[n] : \ \forall x.\ exp_3(x, n, 1) = x^n \ . \tag{4.2}$$

We want to prove that

$$F[n+1]: \ \forall x.\ exp_3(x, n+1, 1) = x^{n+1} \ . \tag{4.3}$$

By (exp_3 successor), we have

$$exp_3(x, n+1, 1) = exp_3(x, n, x \cdot 1) \ .$$

Unfortunately, the inductive hypothesis (4.2) does not apply to the left side of the equation since $n \neq n+1$, and it does not apply to the right side of the equation because the third argument is $x \cdot 1$ rather than 1. Continuing to apply axioms is unlikely to bring us closer to the proof. Thus, we have failed to prove the property.

What went wrong in the proof? Did we choose the wrong induction variable? Would x have worked better? In fact, it is often the case that the property must be **strengthened** to allow the induction to go through. A stronger theorem provides a stronger inductive hypothesis.

Let us strengthen the property to be proved to

$$\forall x, y, z.\ exp_3(x, y, z) = x^y \cdot z \ . \tag{4.4}$$

It clearly implies the desired property (4.1): just choose $z = 1$.

Again, we must choose the induction variable. Based on (exp_3 successor), we use y again. Thus, we prove by stepwise induction on y that

$$F[y]: \ \forall x, z.\ exp_3(x, y, z) = x^y \cdot z \ .$$

For the base case, we prove

$$F[0]: \ \forall x, z.\ exp_3(x, 0, z) = x^0 \cdot z \ .$$

From (exp_3 zero), we have $exp_3(x, 0, z) = z$, while from (exp. zero), we have $x^0 \cdot z = 1 \cdot z = z$.

Assume as the inductive hypothesis that

$$F[n]: \ \forall x, z.\ exp_3(x, n, z) = x^n \cdot z \tag{4.5}$$

for arbitrary natural number n. We want to prove that

$$F[n+1]: \ \forall x, z'.\ exp_3(x, n+1, z') = x^{n+1} \cdot z' \ , \tag{4.6}$$

where we have renamed z to z' for convenience. We have

$$\begin{aligned} exp_3(x, n+1, z') &= exp_3(x, n, x \cdot z') && (exp_3 \text{ successor}) \\ &= x^n \cdot (x \cdot z') && \text{IH (4.5)}, \ z \mapsto x \cdot z' \\ &= x^{n+1} \cdot z' && (\text{exp. successor}) \end{aligned}$$

finishing the proof. The annotation $z \mapsto x \cdot z'$ indicates that $x \cdot z'$ is substituted for z when applying the inductive hypothesis (4.5). This substitution is justified because z is universally quantified. Renaming z to z' avoids confusion during the application of the inductive hypothesis in the second line. ■

Lists

We can define stepwise induction over recursive data structures such as lists (see Chapters 3 and 9). Consider the theory of lists T_{cons}. **Stepwise induction** in T_{cons} is defined according to the following schema

$$(\forall\ \mathsf{atom}\ u.\ F[u])\ \wedge\ (\forall u, v.\ F[v] \rightarrow F[\mathsf{cons}(u, v)])\ \rightarrow\ \forall x.\ F[x]$$

for Σ_{cons}-formulae $F[x]$ with only one free variable x. The notation $\forall\ \mathsf{atom}\ u.\ F[u]$ abbreviates $\forall u.\ \mathsf{atom}(u) \rightarrow F[u]$. In other words, to prove $\forall x.\ F[x]$ — that is, $F[x]$ is T_{cons}-valid for all lists x — it is sufficient to do the following:

- For the **base case**, prove that $F[u]$ is T_{cons}-valid for an arbitrary $\mathsf{atom}\ u$.
- For the **inductive step**, assume as the **inductive hypothesis** that for some arbitrary list v, $F[v]$ is valid. Then prove that for arbitrary list u, $F[\mathsf{cons}(u, v)]$ is T_{cons}-valid under this assumption.

These steps comprise the **stepwise induction principle** for lists.

Example 4.2. Consider the theory T^{+}_{cons} obtained from augmenting T_{cons} with the following axioms:

- $\forall\ \mathsf{atom}\ u.\ \forall v.\ \mathit{concat}(u, v) = \mathsf{cons}(u, v)$ (concat. atom)
- $\forall u, v, x.\ \mathit{concat}(\mathsf{cons}(u, v), x) = \mathsf{cons}(u, \mathit{concat}(v, x))$ (concat. list)
- $\forall\ \mathsf{atom}\ u.\ \mathit{rvs}(u) = u$ (reverse atom)
- $\forall x, y.\ \mathit{rvs}(\mathit{concat}(x, y)) = \mathit{concat}(\mathit{rvs}(y), \mathit{rvs}(x))$ (reverse list)
- $\forall\ \mathsf{atom}\ u.\ \mathit{flat}(u)$ (flat atom)
- $\forall u, v.\ \mathit{flat}(\mathsf{cons}(u, v))\ \leftrightarrow\ \mathsf{atom}(u)\ \wedge\ \mathit{flat}(v)$ (flat list)

The first two axioms define the *concat* function, which concatenates two lists together. For example,

$$\begin{aligned}&\mathit{concat}(\mathsf{cons}(a, b), \mathsf{cons}(b, \mathsf{cons}(c, d)))\\ &\quad = \mathsf{cons}(a, \mathsf{cons}(b, \mathsf{cons}(b, \mathsf{cons}(c, \mathsf{cons}(d))))) .\end{aligned}$$

The next two axioms define the *rvs* function, which reverses a list. For example,

$$\mathit{rvs}(\mathsf{cons}(a, \mathsf{cons}(b, c))) = \mathsf{cons}(c, \mathsf{cons}(b, a)) .$$

Note, however, that *rvs* is undefined on lists like $\mathsf{cons}(\mathsf{cons}(a, b), c)$, for $\mathsf{cons}(\mathsf{cons}(a, b), c)$ cannot result from concatenating two lists together. Therefore, the final two axioms define the *flat* predicate, which evaluates to $\top$ on a list iff every element is an atom. For example, $\mathsf{cons}(a, \mathsf{cons}(b, c))$ is *flat*, but $\mathsf{cons}(\mathsf{cons}(a, b), c)$ is not because the first element of the list is itself a list.

Let us prove that the following formula is T^{+}_{cons}-valid:

$$\forall x.\ \mathit{flat}(x)\ \rightarrow\ \mathit{rvs}(\mathit{rvs}(x)) = x\ . \tag{4.7}$$

For example,

$$\begin{aligned} rvs(rvs(\mathsf{cons}(a, \mathsf{cons}(b, c)))) &= rvs(\mathsf{cons}(c, \mathsf{cons}(b, a))) \\ &= \mathsf{cons}(a, \mathsf{cons}(b, c)) \end{aligned}$$

We prove by stepwise induction on x that

$$F[x] : \ \mathit{flat}(x) \ \rightarrow \ rvs(rvs(x)) = x \ .$$

For the base case, we consider arbitrary $\mathsf{atom}\ u$ and prove

$$F[u] : \ \mathit{flat}(u) \ \rightarrow \ rvs(rvs(u)) = u \ .$$

But $rvs(rvs(u)) = u$ follows from two applications of (reverse atom).

Assume as the inductive hypothesis that for arbitrary list v,

$$F[v] : \ \mathit{flat}(v) \ \rightarrow \ rvs(rvs(v)) = v \ . \tag{4.8}$$

We want to prove that for arbitrary list u,

$$F[\mathsf{cons}(u, v)] : \ \mathit{flat}(\mathsf{cons}(u, v)) \ \rightarrow \ rvs(rvs(\mathsf{cons}(u, v))) = \mathsf{cons}(u, v) \ . \tag{4.9}$$

Consider two cases: either $\mathsf{atom}(u)$ or $\neg\mathsf{atom}(u)$.

If $\neg\mathsf{atom}(u)$, then

$$\mathit{flat}(\mathsf{cons}(u, v)) \ \Leftrightarrow \ \mathsf{atom}(u) \ \wedge \ \mathit{flat}(v) \ \Leftrightarrow \ \bot \ ,$$

by (flat list) and assumption. Therefore, (4.9) holds since its antecedent is $\bot$.

If $\mathsf{atom}(u)$, then we have that

$$\mathit{flat}(\mathsf{cons}(u, v)) \ \Leftrightarrow \ \mathsf{atom}(u) \ \wedge \ \mathit{flat}(v) \ \Leftrightarrow \ \mathit{flat}(v)$$

by (flat list). Furthermore,

$$\begin{aligned} &rvs(rvs(\mathsf{cons}(u, v))) \\ &\quad = rvs(rvs(\mathit{concat}(u, v))) && \text{(concat. atom)} \\ &\quad = rvs(\mathit{concat}(rvs(v), rvs(u))) && \text{(reverse list)} \\ &\quad = \mathit{concat}(rvs(rvs(u)), rvs(rvs(v))) && \text{(reverse list)} \\ &\quad = \mathit{concat}(u, rvs(rvs(v))) && \text{(reverse atom)} \\ &\quad = \mathit{concat}(u, v) && \text{IH (4.8), since } \mathit{flat}(v) \\ &\quad = \mathsf{cons}(u, v) && \text{(concat. atom)} \end{aligned}$$

which finishes the proof. ■

4.2 Complete Induction

Complete induction is a form of induction that sometimes yields more concise proofs. For the theory of arithmetic T_{PA} it is defined according to the following schema

$$(\forall n.\ (\forall n'.\ n' < n \to F[n']) \to F[n]) \ \to\ \forall x.\ F[x]$$

for Σ_{PA}-formulae $F[x]$ with only one free variable x. In other words, to prove $\forall x.\ F[x]$ — that is, $F[x]$ is T_{PA}-valid for all natural numbers x — it is sufficient to follow the **complete induction principle**:

- Assume as the **inductive hypothesis** that for arbitrary natural number n and for every natural number n' such that $n' < n$, $F[n']$ is T_{PA}-valid. Then prove that $F[n]$ is T_{PA}-valid.

It appears that we are missing a base case. In practice, a case analysis usually requires at least one base case. In other words, the base case is implicit in the structure of complete induction. For example, for $n = 0$, the inductive hypothesis does not provide any information — there does not exist a natural number $n' < 0$. Hence, $F[0]$ must be shown separately without assistance from the inductive hypothesis.

Example 4.3. Consider another augmented version of Peano arithmetic, T^*_{PA}, that defines integer division. It has the usual axioms of T_{PA} plus the following:

- $\forall x, y.\ x < y \ \to\ quot(x, y) = 0$ (quotient less)
- $\forall x, y.\ y > 0 \ \to\ quot(x + y, y) = quot(x, y) + 1$ (quotient successor)
- $\forall x, y.\ x < y \ \to\ rem(x, y) = x$ (remainder less)
- $\forall x, y.\ y > 0 \ \to\ rem(x + y, y) = rem(x, y)$ (remainder successor)

These axioms define functions for computing integer quotients $quot(x, y)$ and remainders $rem(x, y)$. For example, $quot(5, 3) = 1$ and $rem(5, 3) = 2$. We prove two properties, which the reader may recall from grade school, about these functions. First, we prove that the remainder is always less than the divisor:

$$\forall x, y.\ y > 0 \ \to\ rem(x, y) < y\ . \tag{4.10}$$

Then we prove that

$$\forall x, y.\ y > 0 \ \to\ x = y \cdot quot(x, y) + rem(x, y)\ . \tag{4.11}$$

For property (4.10), (remainder successor) suggests that we apply complete induction on x to prove

$$F[x] :\ \forall y.\ y > 0 \ \to\ rem(x, y) < y\ . \tag{4.12}$$

Thus, for the inductive hypothesis, assume that for arbitrary natural number x,

$$\forall x'.\ x' < x \ \to\ \underbrace{\forall y.\ y > 0 \ \to\ rem(x', y) < y}_{F[x']}\ . \tag{4.13}$$

Let y be an arbitrary positive natural number. Consider two cases: either $x < y$ or $\neg(x < y)$.

If $x < y$, then

$$\begin{aligned} rem(x, y) &= x && \text{(remainder less)} \\ &< y && \text{by assumption } x < y \end{aligned}$$

as desired.

If $\neg(x < y)$, then there is a natural number n, $n < x$, such that $x = n + y$. Compute

$$\begin{aligned} rem(x, y) &= rem(n + y, y) && x = n + y \\ &= rem(n, y) && \text{(remainder successor)} \\ &< y && \text{IH (4.13)}, x' \mapsto n, \text{ since } n < x \end{aligned}$$

finishing the proof of this property.

For property (4.11), (remainder successor) again suggests that we apply complete induction on x to prove

$$G[x] : \ \forall y.\ y > 0 \ \rightarrow \ x = y \cdot quot(x, y) + rem(x, y)\ . \tag{4.14}$$

Thus, for the inductive hypothesis, assume that for arbitrary natural number x,

$$\forall x'.\ x' < x \ \rightarrow \ \underbrace{\forall y.\ y > 0 \ \rightarrow \ x' = y \cdot quot(x', y) + rem(x', y)}_{G[x']}\ . \tag{4.15}$$

Let y be an arbitrary positive natural number. Consider two cases: either $x < y$ or $\neg(x < y)$.

If $x < y$, then

$$\begin{aligned} &y{\cdot}quot(x, y) + rem(x, y) \\ &\quad = y \cdot 0 + rem(x, y) && \text{(quotient less)} \\ &\quad = x && \text{(remainder less)} \end{aligned}$$

as desired.

If $\neg(x < y)$, then there is a natural number $n < x$ such that $x = n + y$. Compute

$$\begin{aligned} &y{\cdot}quot(x, y) + rem(x, y) \\ &\quad = y \cdot quot(n + y, y) + rem(n + y, y) && x = n + y \\ &\quad = y \cdot (quot(n, y) + 1) + rem(n + y, y) && \text{(quotient successor)} \\ &\quad = y \cdot (quot(n, y) + 1) + rem(n, y) && \text{(remainder successor)} \\ &\quad = (y \cdot quot(n, y) + rem(n, y)) + y \\ &\quad = n + y && \text{IH (4.15)}, x' \mapsto n, \text{ since } n < x \\ &\quad = x && x = n + y \end{aligned}$$

finishing the proof of this property. ■

In the next section, we generalize complete induction so that we can apply it in other theories.

4.3 Well-Founded Induction

A binary predicate $\prec$ over a set S is a **well-founded relation** iff there does not exist an infinite sequence $s_1, s_2, s_3, \ldots$ of elements of S such that each successive element is less than its predecessor:

$$s_1 \succ s_2 \succ s_3 \succ \cdots ,$$

where $s \prec t$ iff $t \succ s$. In other words, each sequence of elements of S that decreases according to $\prec$ is finite.

Example 4.4. The relation $<$ is well-founded over the natural numbers. Any sequence of natural numbers decreasing according to $<$ is finite:

$$1023 > 39 > 30 > 29 > 8 > 3 > 0 .$$

However, the relation $<$ is not well-founded over the rationals. Consider the infinite decreasing sequence

$$1 > \frac{1}{2} > \frac{1}{3} > \frac{1}{4} > \cdots ,$$

that is, the sequence $s_i = \frac{1}{i}$ for $i \geq 0$. ■

Example 4.5. Consider the theory $T^{\mathsf{PA}}_{\mathsf{cons}}$, which includes the axioms of T_{cons} and T_{PA} and the following axioms:

- $\forall\ \mathsf{atom}\ u, v.\ u \preceq_{\mathsf{c}} v \ \leftrightarrow\ u = v$ ($\preceq_{\mathsf{c}}$ (1))
- $\forall\ \mathsf{atom}\ u.\ \forall v.\ \neg\mathsf{atom}(v) \ \rightarrow\ \neg(v \preceq_{\mathsf{c}} u)$ ($\preceq_{\mathsf{c}}$ (2))
- $\forall\ \mathsf{atom}\ u.\ \forall v, w.\ u \preceq_{\mathsf{c}} \mathsf{cons}(v, w) \ \leftrightarrow\ u = v \ \vee\ u \preceq_{\mathsf{c}} w$ ($\preceq_{\mathsf{c}}$ (3))
- $\forall u_1, v_1, u_2, v_2.\ \mathsf{cons}(u_1, v_1) \preceq_{\mathsf{c}} \mathsf{cons}(u_2, v_2)$
  $\leftrightarrow\ (u_1 = u_2 \ \wedge\ v_1 \preceq_{\mathsf{c}} v_2) \ \vee\ \mathsf{cons}(u_1, v_1) \preceq_{\mathsf{c}} v_2$ ($\preceq_{\mathsf{c}}$ (4))
- $\forall x, y.\ x \prec_{\mathsf{c}} y \ \leftrightarrow\ x \preceq_{\mathsf{c}} y \ \wedge\ x \neq y$ ($\prec_{\mathsf{c}}$)
- $\forall\ \mathsf{atom}\ u.\ |u| = 1$ (length atom)
- $\forall u, v.\ |\mathsf{cons}(u, v)| = 1 + |v|$ (length list)

The first four axioms define the sublist relation $\preceq_{\mathsf{c}}$. $x \preceq_{\mathsf{c}} y$ holds iff x is a (not necessarily strict) sublist of y. The next axiom defines the strict sublist relation: $x \prec_{\mathsf{c}} y$ iff x is a strict sublist of y. The final two axioms define the length function, which returns the number of elements in a list.

The strict sublist relation $\prec_{\mathsf{c}}$ is well-founded on the set of all lists. One can prove that the number of sublists of a list is finite; and that its set of strict sublists is a superset of the set of strict sublists of any of its sublists. Hence, there cannot be an infinite sequence of lists descending according to $\prec_{\mathsf{c}}$. ■

Well-founded induction generalizes complete induction to arbitrary theory T by allowing the use of any binary predicate $\prec$ that is well-founded in the domain of every T-interpretation. It is defined in the theory T with well-founded relation $\prec$ by the following schema

$$(\forall n.\ (\forall n'.\ n' \prec n \to F[n']) \to F[n]) \ \to\ \forall x.\ F[x]$$

for Σ-formulae $F[x]$ with only one free variable x. In other words, to prove the T-validity of $\forall x.\ F[x]$, it is sufficient to follow the **well-founded induction principle**:

- Assume as the **inductive hypothesis** that for arbitrary element n and for every element n' such that $n' \prec n$, $F[n']$ is T-valid. Then prove that $F[n]$ is T-valid.

Complete induction in T_{PA} of Section 4.2 is a specific instance of well-founded induction that uses the well-founded relation $<$.

A theory of lists augmented with the first five axioms of Example 4.5 has well-founded induction in which the well-founded relation is $\prec_{\mathsf{c}}$.

Example 4.6. Consider proving the trivial property

$$\forall x.\ |x| \geq 1 \tag{4.16}$$

in $T^{\mathsf{PA}}_{\mathsf{cons}}$, which was defined in Example 4.5. We apply well-founded induction on x using the well-founded relation $\prec_{\mathsf{c}}$ to prove

$$F[x]:\ |x| \geq 1\ . \tag{4.17}$$

For the inductive hypothesis, assume that

$$\forall x'.\ x' \prec_{\mathsf{c}} x\ \to\ \underbrace{|x'| \geq 1}_{F[x']}\ . \tag{4.18}$$

Consider two cases: either $\mathsf{atom}(x)$ or $\neg\mathsf{atom}(x)$.

In the first case $|x| = 1 \geq 1$ by (length atom).

In the second case x is not an atom, so $x = \mathsf{cons}(u, v)$ for some u, v by the (construction) axiom. Then

$$\begin{aligned}
|x| &= |\mathsf{cons}(u, v)| \\
&= 1 + |v| && \text{(length list)} \\
&\geq 1 + 1 && \text{IH (4.18), } x' \mapsto v, \text{ since } v \prec_{\mathsf{c}} \mathsf{cons}(u, v) \\
&\geq 1
\end{aligned}$$

as desired. Exercise 4.2 asks the reader to prove formally that $\forall u, v.\ v \prec_{\mathsf{c}} \mathsf{cons}(u, v)$.

This property is also easily proved using stepwise induction. ■

In applying well-founded induction, we need not restrict ourselves to the intended domain D of a theory T. A useful class of well-founded relations are **lexicographic relations**. From a finite set of pairs of sets and well-founded relations $(S_1, \prec_1), \ldots, (S_m, \prec_m)$, construct the set

$$S = S_1 \times \cdots \times S_m \ ,$$

and define the relation $\prec$:

$$(s_1, \ldots, s_m) \prec (t_1, \ldots, t_m) \ \Leftrightarrow \ \bigvee_{i=1}^{m} \left(s_i \prec_i t_i \ \wedge \ \bigwedge_{j=1}^{i-1} s_j = t_j \right)$$

for $s_i, t_i \in S_i$. That is, for elements $s : (s_1, \ldots, s_m)$, $t : (t_1, \ldots, t_m)$ of S, $s \prec t$ iff at some position i, $s_i \prec_i t_i$, and for all preceding positions j, $s_j = t_j$. For convenience, we abbreviate $(s_1, \ldots, s_m)$ by $\overline{s}$ and thus write, for example, $\overline{s} \prec \overline{t}$.

Lexicographic well-founded induction has the form

$$(\forall \overline{n}.\ (\forall \overline{n}'.\ \overline{n}' \prec \overline{n} \rightarrow F[\overline{n}']) \rightarrow F[\overline{n}]) \ \rightarrow \ \forall \overline{x}.\ F[\overline{x}]$$

for Σ-formula $F[\overline{x}]$ with only free variables $\overline{x} = \{x_1, \ldots, x_m\}$. Notice that the form of this induction principle is the same as well-founded induction. The only difference is that we are considering tuples $\overline{n} = (n_1, \ldots, n_m)$ rather than single elements n.

Example 4.7. Consider the following puzzle. You have a bag of red, yellow, and blue chips. If only one chip remains in the bag, you take it out. Otherwise, you remove two chips at random:

1. If one of the two removed chips is red, you do not put any chips in the bag.
2. If both of the removed chips are yellow, you put one yellow chip and five blue chips in the bag.
3. If one of the chips is blue and the other is not red, you put ten red chips in the bag.

These cases cover all possibilities for the two chips. Does this process always halt?

We prove the following property: *for all bags of chips, you can execute the choose-and-replace process only a finite number of times before the bag is empty.* Let the triple

$$(\#\text{yellow},\ \#\text{blue},\ \#\text{red})$$

represent the current state of the bag. Such a tuple is in the set of triples of natural numbers $S : \mathbb{N}^3$. Let $<_3$ be the natural lexicographic extension of $<$ to such triples. For example,

$$(11, 13, 3) \not<_3 (11, 9, 104) \quad \text{but} \quad (11, 9, 104) <_3 (11, 13, 3) \ .$$

We prove that for arbitrary bag state (y, b, r) represented by the triple of natural numbers y, b, and r, only a finite number of steps remain.

For the base cases, consider when the bag has no chips (state $(0,0,0)$) or only one chip (one of states $(1,0,0)$, $(0,1,0)$, or $(0,0,1)$). In the first case, you are done; in the second set of cases, only one step remains.

Assume for the inductive hypothesis that for any bag state (y',b',r') such that

$$(y',b',r') <_3 (y,b,r) \ ,$$

only a finite number of steps remain. Now remove two chips from the current bag, represented by state (y,b,r). Consider the three possible cases:

1. *If one of the two removed chips is red, you do not put any chips in the bag.* Then the new bag state is $(y-1,b,r-1)$, $(y,b-1,r-1)$, or $(y,b,r-2)$. Each is less than (y,b,r) by $<_3$.
2. *If both of the removed chips are yellow, you put one yellow chip and five blue chips in the bag.* Then the new bag state is $(y-1,b+5,r)$, which is less than (y,b,r) by $<_3$.
3. *If one of the chips is blue and the other is not red, you put ten red chips in the bag.* Then the new bag state is $(y-1,b-1,r+10)$ or $(y,b-2,r+10)$. Each is less than (y,b,r) by $<_3$.

In all cases, we can apply the inductive hypothesis to deduce that only a finite number of steps remain from the next state. Since only one step of the process is required to get to the next state, there are only a finite number of steps remaining from the current state (y,b,r). Hence, the process always halts. ■

Example 4.8. Consider proving the property

$$\forall x,y.\ x \preceq_{\mathsf{c}} y \ \rightarrow\ |x| \leq |y| \tag{4.19}$$

in $T^{\mathsf{PA}}_{\mathsf{cons}}$. Let $\prec^2_{\mathsf{c}}$ be the natural lexicographic extension of $\prec_{\mathsf{c}}$ to pairs of lists. That is, $(x_1,y_1) \prec^2_{\mathsf{c}} (x_2,y_2)$ iff $x_1 \prec_{\mathsf{c}} x_2 \vee (x_1 = x_2 \ \wedge\ y_1 \prec_{\mathsf{c}} y_2)$.

We apply lexicographic well-founded induction to pairs (x,y) to prove

$$F[x,y] :\ x \preceq_{\mathsf{c}} y \ \rightarrow\ |x| \leq |y| \ . \tag{4.20}$$

For the inductive hypothesis, assume that

$$\forall x',y'.\ (x',y') \prec^2_{\mathsf{c}} (x,y) \ \rightarrow\ \underbrace{x' \preceq_{\mathsf{c}} y' \ \rightarrow\ |x'| \leq |y'|}_{F[x',y']} \ . \tag{4.21}$$

Now consider arbitrary lists x and y. Consider two cases: either $\mathsf{atom}(x)$ or $\neg\mathsf{atom}(x)$.

If $\mathsf{atom}(x)$, then

$$\begin{array}{ll} |x| = 1 & (\mathsf{length\ atom}) \\ \quad \leq |y| & \text{Example 4.6} \end{array}$$

Hence, regardless of whether $x \preceq_{\mathsf{c}} y$, we have that $|x| \leq |y|$ so that (4.20) holds.

If $\neg\mathsf{atom}(x)$, then consider two cases: either $\mathsf{atom}(y)$ or $\neg\mathsf{atom}(y)$. If $\mathsf{atom}(y)$, then

$$x \preceq_{\mathsf{c}} y \;\Leftrightarrow\; \bot$$

by ($\preceq_{\mathsf{c}}$ (2)); therefore, (4.20) holds trivially.

For the final case, we have that $\neg\mathsf{atom}(x)$ and $\neg\mathsf{atom}(y)$. Then $x = \mathsf{cons}(u_1, v_1)$ and $y = \mathsf{cons}(u_2, v_2)$ for some lists u_1, v_1, u_2, v_2. We have

$$\begin{aligned} x \preceq_{\mathsf{c}} y &\;\Leftrightarrow\; \mathsf{cons}(u_1, v_1) \preceq_{\mathsf{c}} \mathsf{cons}(u_2, v_2) && \text{assumption} \\ &\;\Leftrightarrow\; (u_1 = u_2 \;\wedge\; v_1 \preceq_{\mathsf{c}} v_2) \;\vee\; \mathsf{cons}(u_1, v_1) \preceq_{\mathsf{c}} v_2 && (\preceq_{\mathsf{c}}\ (4)) \end{aligned}$$

The disjunction suggests two possibilities. Consider the first disjunct. Because $v_1 \prec_{\mathsf{c}} \mathsf{cons}(u_1, v_1) = x$, we have that

$$(v_1, v_2) \prec_{\mathsf{c}}^2 (x, y) \;,$$

allowing us to appeal to the inductive hypothesis (4.21): from $v_1 \preceq_{\mathsf{c}} v_2$, deduce that $|v_1| \leq |v_2|$. Then with two applications of (length list), we have

$$|x| \leq |y| \;\Leftrightarrow\; 1 + |v_1| \leq 1 + |v_2| \;\Leftrightarrow\; |v_1| \leq |v_2| \;.$$

Therefore, $|x| \leq |y|$ and (4.20) holds for this case.

Suppose the second disjunct ($\mathsf{cons}(u_1, v_1) \preceq_{\mathsf{c}} v_2$) holds. We again look to the inductive hypothesis (4.21). We have

$$(\mathsf{cons}(u_1, v_1), v_2) \prec_{\mathsf{c}}^2 (x, y)$$

because $\mathsf{cons}(u_1, v_1) = x$ and $v_2 \prec_{\mathsf{c}} \mathsf{cons}(u_2, v_2) = y$. Therefore, the inductive hypothesis tells us that $|x| \leq |v_2|$, while (length list) implies that $|v_2| < |y|$. In short,

$$|x| \leq |v_2| < |y| \;,$$

which implies $|x| \leq |y|$ as desired, completing the proof. ■

Example 4.9. Augment the theory of Presburger arithmetic $T_{\mathbb{N}}$ (see Chapters 3 and 7) with the following axioms to define the Ackermann function:

- $\forall y.\ ack(0, y) = y + 1$ (ack left zero)
- $\forall x.\ ack(x + 1, 0) = ack(x, 1)$ (ack right zero)
- $\forall x, y.\ ack(x + 1, y + 1) = ack(x, ack(x + 1, y))$ (ack successor)

The Ackermann function grows quickly for increasing arguments:

- $ack(0, 0) = 1$
- $ack(1, 1) = 3$

- $ack(2,2) = 7$
- $ack(3,3) = 61$
- $ack(4,4) = 2^{2^{2^{2^{16}}}} - 3$

One might expect that proving properties about the Ackermann function would be difficult.

However, lexicographic well-founded induction allows us to reason about certain properties of the function. Define $<_2$ as the natural lexicographic extension of $<$ to pairs of natural numbers. Now consider input arguments to ack and the resulting arguments in recursive calls:

- (ack left zero) does not involve a recursive call.
- In (ack right zero), $(x+1, 0) >_2 (x, 1)$.
- In (ack successor),
  - $(x+1, y+1) >_2 (x+1, y)$, and
  - $(x+1, y+1) >_2 (x, ack(x+1, y))$.

As the arguments decrease according to $<_2$ with each level of recursion, we conclude that the computation of $ack(x, y)$ halts for every x and y. In Chapter 5, we show that finding well-founded relations is a general technique for showing that functions always halt.

Additionally, we can induct over the execution of ack to prove properties of the ack function itself. Let us prove that

$$\forall x, y.\ ack(x, y) > y \tag{4.22}$$

is $T_{\mathbb{N}}^{ack}$-valid. We apply lexicographic well-founded induction to the arguments of ack to prove

$$F[x, y] :\ \ ack(x, y) > y \tag{4.23}$$

for arbitrary natural numbers x and y. For the inductive hypothesis, assume that

$$\forall x', y'.\ (x', y') <_2 (x, y) \ \rightarrow\ \underbrace{ack(x', y') > y'}_{F[x', y']}\ . \tag{4.24}$$

Consider three cases: $x = 0$, $x > 0\ \wedge\ y = 0$, and $x > 0\ \wedge\ y > 0$.

If $x = 0$, then $ack(0, y) = y + 1 > y$ by (ack left zero), as desired.

If $x > 0\ \wedge\ y = 0$, then

$$ack(x, 0) = ack(x - 1, 1)$$

by (ack right zero). Since

$$(x' : x - 1,\ y' : 1) <_2 (x, y)\ ,$$

the inductive hypothesis (4.24) tells us that

$$ack(x-1,1) > 1 \ .$$

Therefore, we have

$$ack(x,0) = ack(x-1,1) > 1 \ ,$$

so $ack(x,0) > 0$ as desired.

For the final case, $x > 0 \ \wedge \ y > 0$, we have

$$ack(x,y) = ack(x-1, ack(x,y-1))$$

by (ack successor). Since

$$(x' : x-1, \ y' : ack(x,y-1)) <_2 (x,y) \ ,$$

the inductive hypothesis (4.24) implies that

$$ack(x-1, ack(x,y-1)) > ack(x,y-1) \ .$$

Furthermore, since

$$(x' : x, \ y' : y-1) <_2 (x,y) \ ,$$

the inductive hypothesis (4.24) implies that

$$ack(x,y-1) > y-1 \ .$$

All together, then, we have

$$ack(x,y) = ack(x-1, ack(x,y-1)) > ack(x,y-1) > y-1 \ ;$$

hence, $ack(x,y) > (y-1)+1 = y$, completing the proof. ■

4.4 Structural Induction

Induction has many other applications outside of reasoning about the validity of first-order formulae. In this section, we introduce the proof technique of **structural induction** for proving properties about formulae themselves. Structural induction is applied in Section 2.7, in analyzing the quantifier elimination procedures of Chapter 7, and in other applications throughout the book.

Define the **strict subformula relation** over FOL formulae as follows: two formulae F_1 and F_2 are related by the strict subformula relation iff F_1 is a strict subformula of F_2. The strict subformula relation is well founded over the set of FOL formulae since every formula, having only a finite number of symbols, has only a finite number of strict subformulae; and each of its strict subformulae has fewer strict subformulae that it does. To prove a desired property of FOL formulae, instantiate the well-founded induction principle with the strict subformula relation:

- Assume as the **inductive hypothesis** that for arbitrary FOL formula F and for every strict subformula G of F, G has the desired property. Then prove that F has the property.

Since atoms do not have strict subformulae, they are treated as base cases. This induction principle is the **structural induction principle**.

Example 4.10. Exercise 1.3 asks the reader to prove that certain logical connectives are redundant in the presence of others. Formally, the exercise is asking the reader to prove the following claim: Every propositional formula F is equivalent to a propositional formula F' constructed with only the logical connectives $\top$, $\wedge$, and $\neg$.

There are three base cases to consider:

- The formula $\top$ can be represented directly as $\top$.
- The formula $\bot$ is equivalent to $\neg\top$.
- Any propositional variable P can be represented directly as P.

For the inductive step, consider formulae G, G_1, and G_2, and assume as the inductive hypothesis that each is equivalent to formulae G', G'_1, and G'_2, respectively, which are constructed only from the connectives $\top$, $\vee$, and $\neg$ (and propositional variables, of course). We show that each possible formulae that can be constructed from G, G_1, and G_2 with only one logical connective is equivalent to another constructed with only $\top$, $\vee$, and $\neg$:

- $\neg G$ is equivalent to $\neg G'$ from the inductive hypothesis.
- By considering the truth table in which the four possible valuations of G_1 and G_2 are considered, one can establish that $G_1 \vee G_2$ is equivalent to $\neg(\neg G'_1 \wedge \neg G'_2)$. By the inductive hypothesis, the latter formula is constructed only from propositional variables, $\top$, $\wedge$, and $\neg$.
- By similar reasoning, $G_1 \rightarrow G_2$ is equivalent to $\neg(G'_1 \wedge \neg G'_2)$, which satisfies the claim.
- Similar reasoning handles $G_1 \leftrightarrow G_2$ as well.

Hence, the claim is proved.

Note that the main argument is essentially similar to the answer that the reader might have provided in answering Exercise 1.3. Structural induction merely provides the basis for lifting the truth-table argument to a general statement about propositional formulae. ■

Structural induction is also useful for reasoning about interpretations of formulae, as the following example shows.

Example 4.11. This example relies on several basic concepts of set theory; however, even the reader unfamiliar with set theory can understand the application of structural induction without understanding the actual claim.

Consider $\Sigma_{\mathbb{Q}}$-formulae $F[x_1, \ldots, x_n]$ in which the only predicate is $\leq$, the only logical connectives are $\vee$ and $\wedge$, and the only quantifier is $\forall$. We

prove that the set of satisfying $T_{\mathbb{Q}}$-interpretations of F (intuitively, those $T_{\mathbb{Q}}$-interpretations that assign to $x_1, \ldots, x_n$ values from $\mathbb{Q}^n$ that satisfy F) describes a closed subset of $\mathbb{Q}^n$.

For the base case, consider any inequality $\alpha \leq \beta$ with free variables $x_1, \ldots, x_n$. From basic set theory, the set of satisfying points is closed.

For the inductive step, consider formulae G, G_1, and G_2 constructed as specified. Assume as the inductive hypothesis that the satisfying $T_{\mathbb{Q}}$-interpretations for each comprise closed sets. Consider applying the allowed logical connectives and quantifier:

- $G_1 \wedge G_2$: The set described by this formula is the set-theoretic intersection of the sets described by G_1 and G_2, and is thus closed by the inductive hypothesis and set theory.
- $G_1 \vee G_2$: Similarly, the set described by this formula is the set-theoretic union of the sets described by G_1 and G_2, and is thus closed by the inductive hypothesis and set theory.
- $\forall x.\ G$: Consider subformula G with free variable x (if x is not free in G, then the formula is equivalent to just G, which describes a closed set by the inductive hypothesis). For each value $\frac{a}{b} \in \mathbb{Q}$, consider the formula

  $$G_{\frac{a}{b}} :\ G \ \wedge\ bx \leq a \ \wedge\ a \leq bx\ .$$

  The set described by each $G_{\frac{a}{b}}$ is closed according to the inductive hypothesis and reasoning similar to the previous cases. From set theory, the conjunction of all such sets is still closed, so the set of satisfying $T_{\mathbb{Q}}$-interpretations of $\forall x.\ G$ describes a closed set.

The induction is complete, so the claim is proved.

Results from Chapter 7 prove that $\exists$ also preserves closed sets in $T_{\mathbb{Q}}$. ■

Remark 4.12. Example 4.11 considers a subset of FOL formulae. However, this subset is by definition closed under conjunction, disjunction, and universal quantification: if F, F_1, and F_2 are in the subset, then so are $F_1 \wedge F_2$, $F_1 \vee F_2$, and $\forall x.\ F$; and conversely. In other words, all strict subformulae of a formula in the subset are also in the subset, so that structural induction is applicable.

The proof of Lemma 2.31 provides another example of the application of structural induction.

4.5 Summary

This chapter covers several induction principles in several first-order theories:

- *Stepwise induction* is presented in the context of integer arithmetic and lists. The induction principle requires defining a step such as adding one or constructing a list with one more element.

- *Complete induction* is presented in the context of integer arithmetic. The induction principle relies on the well-foundedness of the $<$ predicate. Rather than assuming that the desired property holds for one element n and proving the property for the case $n+1$ as in stepwise reduction, one assumes that the property holds for all elements $n' < n$ and proves that it holds for n. This stronger assumption sometime yields easier or more concise proofs.
- *Well-founded induction* generalizes complete induction to other theories; it is presented in the context of lists and lexicographic tuples. The induction principle requires a well-founded relation over the domain.
- *Structural induction* is an instance of well-founded induction in which the domain is formulae and the well-founded relation is the strict subformula relation.

Besides being an important tool for proving first-order validities, induction is the basis for both verification methodologies studied in Chapter 5. Structural induction also serves as the basis for the quantifier elimination procedures studied in Chapter 7.

Bibliographic Remarks

The induction proofs in Examples 4.1, 4.3, and 4.9 are taken from the text of Manna and Waldinger [55].

Blaise Pascal (1623–1662) and Jacob Bernoulli (1654–1705) are recognized as having formalized stepwise and complete induction, respectively. Less formal versions of induction appear in texts by Francesco Maurolico (1494–1575); Rabbi Levi Ben Gershon (1288–1344), who recognized induction as a distinct form of mathematical proof; Abu Bekr ibn Muhammad ibn al-Husayn Al-Karaji (953–1029); and Abu Kamil Shuja Ibn Aslam Ibn Mohammad Ibn Shaji (850–930) [97]. Some historians claim that Euclid may have applied induction informally.

Exercises

4.1 (T^{+}_{cons})**.** Prove the following in T^{+}_{cons}:

(a) $\forall u, v.\ \mathit{flat}(u) \ \wedge\ \mathit{flat}(v) \ \rightarrow\ \mathit{flat}(\mathit{concat}(u, v))$
(b) $\forall u.\ \mathit{flat}(u) \ \rightarrow\ \mathit{flat}(\mathit{rvs}(u))$

4.2 ($T^{\mathsf{PA}}_{\mathsf{cons}}$)**.** Prove or disprove the following in $T^{\mathsf{PA}}_{\mathsf{cons}}$:

(a) $\forall u.\ u \preceq_{\mathsf{c}} u$
(b) $\forall u, v, w.\ \mathsf{cons}(u, v) \preceq_{\mathsf{c}} w \ \rightarrow\ v \preceq_{\mathsf{c}} w$
(c) $\forall u, v.\ v \prec_{\mathsf{c}} \mathsf{cons}(u, v)$

4.3 ($T^{+}_{\mathsf{cons}} \cup T^{\mathsf{PA}}_{\mathsf{cons}}$). Prove the following in $T^{+}_{\mathsf{cons}} \cup T^{\mathsf{PA}}_{\mathsf{cons}}$:

(a) $\forall u, v.\ |\mathit{concat}(u, v)| = |u| + |v|$
(b) $\forall u.\ \mathit{flat}(u) \ \rightarrow\ |\mathit{rvs}(u)| = |u|$

4.4 (Chips). Does the process of Example 4.7 still halt if

(a) in Step 1, you return one red chip to the bag?
(b) in Step 1, you add one blue chip?
(c) in Step 1, you add one blue chip; and in Step 3, you return the blue chip to the bag but do not add any other chips?

4.5 (Strict sublist). Modify Example 4.8 to prove

$$\forall x, y.\ x \prec_{\mathsf{c}} y \ \rightarrow\ |x| < |y| \ .$$

4.6 (Structural induction). Prove that every first-order formula F is equivalent to a first-order formula F' constructed with only the logical connectives $\top$, $\wedge$, and $\neg$ and the quantifier $\forall$.

4.7 (Finite number of sublists). Prove that the number of sublists of a list (defined in $T^{\mathsf{PA}}_{\mathsf{cons}}$) is finite.

4.8 ($^{\star}\prec_{\mathsf{c}}$ is well-founded). Prove that $\prec_{\mathsf{c}}$, defined in $T^{\mathsf{PA}}_{\mathsf{cons}}$, is well-founded over lists. To avoid circularity, do not apply well-founded induction in this proof. *Hint*: Prove that $\prec_{\mathsf{c}}$ is transitive and **irreflexive** ($\forall u.\ \neg(u \prec_{\mathsf{c}} u)$); then apply Exercise 4.7.