Practical Foundations for Programming Languages

Robert Harper
Carnegie Mellon University

Spring, 2009

[Draft of January 21, 2009 at 12:59pm.]

Preface

This is a working draft of a book on the foundations of programming languages. The central organizing principle of the book is that programming language features may be seen as manifestations of an underlying type structure that governs its syntax and semantics. The emphasis, therefore, is on the concept of *type*, which codifies and organizes the computational universe in much the same way that the concept of *set* may be seen as an organizing principle for the mathematical universe. The purpose of this book is to explain this remark.

This is very much a work in progress, with major revisions made nearly every day. This means that there may be internal inconsistencies as revisions to one part of the book invalidate material at another part. Please bear this in mind!

Corrections, comments, and suggestions are most welcome, and should be sent to the author at `rwh@cs.cmu.edu`.

Contents

Part I

Judgements and Rules

Chapter 1

Inductive Definitions

Inductive definitions are an indispensable tool in the study of programming languages. In this chapter we will develop the basic framework of inductive definitions, and give some examples of their use.

1.1 Objects and Judgements

We start with the notion of a *judgement*, or *assertion*, about an *object* of study. We shall make use of many forms of judgement, including examples such as these:

n nat	n is a natural number
$n = n_1 + n_2$	n is the sum of n_1 and n_2
a ast	a is an abstract syntax tree
τ type	τ is a type
$e : \tau$	expression e has type τ
$e \Downarrow v$	expression e has value v

A judgement states that one or more objects have a property or stand in some relation to one another. The property or relation itself is called a *judgement form*, and the judgement that an object or objects have that property or stand in that relation is said to be an *instance* of that judgement form. A judgement form is also called a *predicate*, and the objects constituting an instance are its *subjects*.

We will use the meta-variable P to stand for an unspecified judgement form, and the meta-variables a, b, and c to stand for unspecified objects. We write $a\ P$ for the judgement asserting that P holds of a. When it is not important to stress the subject of the judgement, we write J to stand for

an unspecified judgement. For particular judgement forms, we freely use prefix, infix, or mixfix notation, as illustrated by the above examples, in order to enhance readability.

We are being intentionally vague about the universe of objects that may be involved in an inductive definition. The rough-and-ready rule is that any sort of finite construction of objects from other objects is permissible. In particular, we shall make frequent use of the construction of composite objects of the form $o(a_1, \ldots, a_n)$, where $a_1, \ldots, a_n$ are objects and o is an n-argument operator. This construction includes a special case the formation of n-tuples, $(a_1, \ldots, a_n)$, in which the tupling operator is left implicit. (In Chapters 8 and 6 we will formalize these and richer forms of objects, called abstract syntax trees.)

1.2 Inference Rules

An *inductive definition* of a judgement form consists of a collection of *rules* of the form

$$\frac{J_1 \quad \ldots \quad J_k}{J} \tag{1.1}$$

in which J and $J_1, \ldots, J_k$ are all judgements of the form being defined. The judgements above the horizontal line are called the *premises* of the rule, and the judgement below the line is called its *conclusion*. If a rule has no premises (that is, when k is zero), the rule is called an *axiom*; otherwise it is called a *proper rule*.

An inference rule may be read as stating that the premises are *sufficient* for the conclusion: to show J, it is enough to show $J_1, \ldots, J_k$. When k is zero, a rule states that its conclusion holds unconditionally. Bear in mind that there may be, in general, many rules with the same conclusion, each specifying sufficient conditions for the conclusion. Consequently, if the conclusion of a rule holds, then it is not necessary that the premises hold, for it might have been derived by another rule.

For example, the following rules constitute an inductive definition of the judgement a `nat`:

$$\frac{}{\texttt{zero nat}} \tag{1.2a}$$

$$\frac{a \texttt{ nat}}{\texttt{succ(}a\texttt{) nat}} \tag{1.2b}$$

These rules specify that a `nat` holds whenever either a is `zero`, or a is `succ(`b`)` where b `nat`. Taking these rules to be exhaustive, it follows that a `nat` iff a is a natural number written in unary.

Similarly, the following rules constitute an inductive definition of the judgement a tree:

$$\frac{}{\texttt{empty tree}} \tag{1.3a}$$

$$\frac{a_1 \texttt{ tree} \quad a_2 \texttt{ tree}}{\texttt{node}(a_1;a_2) \texttt{ tree}} \tag{1.3b}$$

These rules specify that a tree holds if either a is `empty`, or a is $\texttt{node}(a_1;a_2)$, where a_1 tree and a_2 tree. Taking these to be exhaustive, these rules state that a is a binary tree, which is to say it is either empty, or a node consisting of two children, each of which is also a binary tree.

The judgement $a = b$ nat defining equality of a nat and b nat is inductively defined by the following rules:

$$\frac{}{\texttt{zero} = \texttt{zero nat}} \tag{1.4a}$$

$$\frac{a = b \texttt{ nat}}{\texttt{succ}(a) = \texttt{succ}(b) \texttt{ nat}} \tag{1.4b}$$

In each of the preceding examples we have made use of a notational convention for specifying an infinite family of rules by a finite number of patterns, or *rule schemes*. For example, Rule (1.2b) is a rule scheme that determines one rule, called an *instance* of the rule scheme, for each choice of object a in the rule. We will rely on context to determine whether a rule is stated for a *specific* object, a, or is instead intended as a rule scheme specifying a rule for *each choice* of objects in the rule. (In Chapter 3 we will remove this amiguity by introducing parameterization of rules by objects.)

A collection of rules is considered to define the *strongest* judgement that is *closed under*, or *respects*, those rules. To be closed under the rules simply means that the rules are *sufficient* to show the validity of a judgement: J holds *if* there is a way to obtain it using the given rules. To be the *strongest* judgement closed under the rules means that the rules are also *necessary*: J holds *only if* there is a way to obtain it by applying the rules. The sufficiency of the rules means that we may show that J holds by *deriving* it by composing rules. Their necessity means that we may reason about it using *rule induction*.

1.3 Derivations

To show that an inductively defined judgement holds, it is enough to exhibit a *derivation* of it. A derivation of a judgement is a composition of rules, starting with axioms and ending with that judgement. It may be thought

of as a tree in which each node is a rule whose children are derivations of its premises. We sometimes say that a derivation of J is *evidence* for the validity of an inductively defined judgement J.

We usually depict derivations as trees with the conclusion at the bottom, and with the children of a node corresponding to a rule appearing above it as evidence for the premises of that rule. Thus, if

$$\frac{J_1 \quad \ldots \quad J_k}{J}$$

is an inference rule and $\nabla_1, \ldots, \nabla_k$ are derivations of its premises, then

$$\frac{\nabla_1 \quad \ldots \quad \nabla_k}{J} \tag{1.5}$$

is a derivation of its conclusion. In particular, if $k = 0$, then the node has no children.

For example, this is a derivation of `succ(succ(succ(zero)))` `nat`:

$$\dfrac{\dfrac{\dfrac{\overline{\texttt{zero nat}}}{\texttt{succ(zero) nat}}}{\texttt{succ(succ(zero)) nat}}}{\texttt{succ(succ(succ(zero))) nat}} \,. \tag{1.6}$$

Similarly, here is a derivation of `node(node(empty;empty);empty)` `tree`:

$$\dfrac{\dfrac{\overline{\texttt{empty tree}} \quad \overline{\texttt{empty tree}}}{\texttt{node(empty;empty) tree}} \quad \overline{\texttt{empty tree}}}{\texttt{node(node(empty;empty);empty) tree}} \,. \tag{1.7}$$

To show that an inductively defined judgement is derivable we need only find a derivation for it. There are two main methods for finding derivations, called *forward chaining*, or *bottom-up construction*, and *backward chaining*, or *top-down construction*. Forward chaining starts with the axioms and works forward towards the desired conclusion, whereas backward chaining starts with the desired conclusion and works backwards towards the axioms.

More precisely, forward chaining search maintains a set of derivable judgements, and continually extends this set by adding to it the conclusion of any rule all of whose premises are in that set. Initially, the set is empty; the process terminates when the desired judgement occurs in the set. Assuming that all rules are considered at every stage, forward chaining will

eventually find a derivation of any derivable judgement, but it is impossible (in general) to decide algorithmically when to stop extending the set and conclude that the desired judgement is not derivable. We may go on and on adding more judgements to the derivable set without ever achieving the intended goal. It is a matter of understanding the global properties of the rules to determine that a given judgement is not derivable.

Forward chaining is undirected in the sense that it does not take account of the end goal when deciding how to proceed at each step. In contrast, backward chaining is goal-directed. Backward chaining search maintains a queue of current goals, judgements whose derivations are to be sought. Initially, this set consists solely of the judgement we wish to derive. At each stage, we remove a judgement from the queue, and consider all rules whose conclusion is that judgement. For each such rule, we add the premises of that rule to the back of the queue, and continue. If there is more than one such rule, this process must be repeated, with the same starting queue, for each candidate rule. The process terminates whenever the queue is empty, all goals having been achieved; any pending consideration of candidate rules along the way may be discarded. As with forward chaining, backward chaining will eventually find a derivation of any derivable judgement, but there is, in general, no algorithmic method for determining in general whether the current goal is derivable. If it is not, we may futilely add more and more judgements to the goal set, never reaching a point at which all goals have been satisfied.

1.4 Rule Induction

Since an inductively defined judgement holds only if there is some derivation of it, we may prove properties of such judgements by *rule induction*, or *induction on derivations*. The principle of rule induction states that to show that a property $\mathcal{P}$ holds of a judgement J whenever J is derivable, it is enough to show that $\mathcal{P}$ is *closed under*, or *respects*, the rules defining J. Writing $\mathcal{P}(J)$ to mean that $\mathcal{P}$ holds of J, we say that $\mathcal{P}$ respects the rule

$$\frac{J_1 \quad \ldots J_k}{J}$$

if $\mathcal{P}(J)$ holds whenever $\mathcal{P}(J_1), \ldots, \mathcal{P}(J_k)$. The assumptions $\mathcal{P}(J_1), \ldots, \mathcal{P}(J_k)$ are the *inductive hypotheses*, and $\mathcal{P}(J)$ is the *inductive conclusion*, corresponding to that rule.

The principle of rule induction is simply the expression of the definition of an inductively defined judgement form as the *strongest* judgement form closed under the rules comprising the definition. This means that the judgement form is both (a) closed under those rules, and (b) sufficient for any other property also closed under those rules. The former property means that a derivation is evidence for the validity of a judgement; the latter means that we may reason about an inductively defined judgement form by rule induction.

If $\mathcal{P}(J)$ is closed under a set of rules defining a judgement form, then so is the conjunction of $\mathcal{P}$ with the judgement itself. This means that when showing $\mathcal{P}$ to be closed under a rule, we may inductively assume not only that $\mathcal{P}(J_i)$ holds for each of the premises J_i, but also that J_i itself holds as well. We shall generally take advantage of this without explicit mentioning that we are doing so.

When specialized to Rules (1.2), the principle of rule induction states that to show $\mathcal{P}(a\ \mathsf{nat})$ whenever $a\ \mathsf{nat}$, it is enough to show:

1. $\mathcal{P}(\mathtt{zero}\ \mathsf{nat})$.
2. $\mathcal{P}(\mathtt{succ}(a)\ \mathsf{nat})$, assuming $\mathcal{P}(a\ \mathsf{nat})$.

This is just the familiar principle of *mathematical induction* arising as a special case of rule induction.

Similarly, rule induction for Rules (1.3) states that to show $\mathcal{P}(a\ \mathsf{tree})$ whenever $a\ \mathsf{tree}$, it is enough to show

1. $\mathcal{P}(\mathtt{empty}\ \mathsf{tree})$.
2. $\mathcal{P}(\mathtt{node}(a_1; a_2)\ \mathsf{tree})$, assuming $\mathcal{P}(a_1\ \mathsf{tree})$ and $\mathcal{P}(a_2\ \mathsf{tree})$.

This is called the principle of *tree induction*, and is once again an instance of rule induction.

As a simple example of a proof by rule induction, let us prove that natural number equality as defined by Rules (1.4) is reflexive:

Lemma 1.1. *If* $a\ \mathsf{nat}$, *then* $a = a\ \mathsf{nat}$.

Proof. By rule induction on Rules (1.2):

Rule (1.2a) Applying Rule (1.4a) we obtain $\mathtt{zero} = \mathtt{zero}\ \mathsf{nat}$.

Rule (1.2b) Assume that $a = a\ \mathsf{nat}$. It follows that $\mathtt{succ}(a) = \mathtt{succ}(a)\ \mathsf{nat}$ by an application of Rule (1.4b).

□

As another example of the use of rule induction, we may show that the predecessor of a natural number is also a natural number. While this may seem self-evident, the point of the example is to show how to derive this from first principles.

Lemma 1.2. *If* $\mathtt{succ}(a)$ nat, *then* a nat.

Proof. It is instructive to re-state the lemma in a form more suitable for inductive proof: if b nat and b is $\mathtt{succ}(a)$ for some a, then a nat. We proceed by rule induction on Rules (1.2).

Rule (1.2a) Vacuously true, since zero is not of the form $\mathtt{succ}(-)$.

Rule (1.2b) We have that b is $\mathtt{succ}(b')$, and we may assume both that the lemma holds for b' and that b' nat. The result follows directly, since if $\mathtt{succ}(b') = \mathtt{succ}(a)$ for some a, then a is b'.

□

Similarly, let us show that the successor operation is injective.

Lemma 1.3. *If* $\mathtt{succ}(a_1) = \mathtt{succ}(a_2)$ nat, *then* $a_1 = a_2$ nat.

Proof. It is instructive to re-state the lemma in a form more directly amenable to proof by rule induction. We are to show that if $b_1 = b_2$ nat then if b_1 is $\mathtt{succ}(a_1)$ and b_2 is $\mathtt{succ}(a_2)$, then $a_1 = a_2$ nat. We proceed by rule induction on Rules (1.4):

Rule (1.4a) Vacuously true, since zero is not of the form $\mathtt{succ}(-)$.

Rule (1.4b) Assuming the result for $b_1 = b_2$ nat, and hence that the premise $b_1 = b_2$ nat holds as well, we are to show that if $\mathtt{succ}(b_1)$ is $\mathtt{succ}(a_1)$ and $\mathtt{succ}(b_2)$ is $\mathtt{succ}(a_2)$, then $a_1 = a_2$ nat. Under these assumptions we have b_1 is a_1 and b_2 is a_2, and so $a_1 = a_2$ nat is just the premise of the rule. (We make no use of the inductive hypothesis to complete this step of the proof.)

□

Both proofs rely on some natural assumptions about the universe of objects; see Section 1.8 on page 13 for further discussion.

1.5 Iterated and Simultaneous Inductive Definitions

Inductive definitions are often *iterated*, meaning that one inductive definition builds on top of another. In an iterated inductive definition the premises of a rule

$$\frac{J_1 \quad \ldots \quad J_k}{J}$$

may be instances of either a previously defined judgement form, or the judgement form being defined. For example, the following rules, define the judgement a list stating that a is a list of natural numbers.

$$\frac{}{\texttt{nil list}} \tag{1.8a}$$

$$\frac{a \text{ nat} \quad b \text{ list}}{\texttt{cons}(a;b) \text{ list}} \tag{1.8b}$$

The first premise of Rule (1.8b) is an instance of the judgement form a nat, which was defined previously, whereas the premise b list is an instance of the judgement form being defined by these rules.

Frequently two or more judgements are defined at once by a *simultaneous inductive definition*. A simultaneous inductive definition consists of a set of rules for deriving instances of several different judgement forms, any of which may appear as the premise of any rule. Since the rules defining each judgement form may involve any of the others, none of the judgement forms may be taken to be defined prior to the others. Instead one must understand that all of the judgement forms are being defined at once by the entire collection of rules. The judgement forms defined by these rules are, as before, the strongest judgement forms that are closed under the rules. Therefore the principle of proof by rule induction continues to apply, albeit in a form that allows us to prove a property of each of the defined judgement forms simultaneously.

For example, consider the following rules, which constitute a simultaneous inductive definition of the judgements a even, stating that a is an even natural number, and a odd, stating that a is an odd natural number:

$$\frac{}{\texttt{zero even}} \tag{1.9a}$$

$$\frac{a \text{ odd}}{\texttt{succ}(a) \text{ even}} \tag{1.9b}$$

$$\frac{a \text{ even}}{\texttt{succ}(a) \text{ odd}} \tag{1.9c}$$

The principle of rule induction for these rules states that to show simultaneously that $\mathcal{P}(a\ \mathsf{even})$ whenever a even and $\mathcal{P}(a\ \mathsf{odd})$ whenever a odd, it is enough to show the following:

1. $\mathcal{P}(\mathtt{zero}\ \mathsf{even})$;
2. if $\mathcal{P}(a\ \mathsf{odd})$, then $\mathcal{P}(\mathtt{succ}(a)\ \mathsf{even})$;
3. if $\mathcal{P}(a\ \mathsf{even})$, then $\mathcal{P}(\mathtt{succ}(a)\ \mathsf{odd})$.

As a simple example, we may use simultaneous rule induction to prove that (1) if a even, then a nat, and (2) if a odd, then a nat. That is, we define the property $\mathcal{P}$ by (1) $\mathcal{P}(a\ \mathsf{even})$ iff a nat, and (2) $\mathcal{P}(a\ \mathsf{odd})$ iff a nat. The principle of rule induction for Rules (1.9) states that it is sufficient to show the following facts:

1. $\mathtt{zero}\ \mathsf{nat}$, which is derivable by Rule (1.2a).
2. If a nat, then $\mathtt{succ}(a)\ \mathsf{nat}$, which is derivable by Rule (1.2b).
3. If a nat, then $\mathtt{succ}(a)\ \mathsf{nat}$, which is also derivable by Rule (1.2b).

1.6 Defining Functions by Rules

A common use of inductive definitions is to define a function by giving an inductive definition of its *graph* relating inputs to outputs, and then showing that the relation uniquely determines the outputs for given inputs. For example, we may define the addition function on natural numbers as the relation $\mathsf{sum}(a;b;c)$, with the intended meaning that c is the sum of a and b, as follows:

$$\frac{b\ \mathsf{nat}}{\mathsf{sum}(\mathtt{zero};b;b)} \tag{1.10a}$$

$$\frac{\mathsf{sum}(a;b;c)}{\mathsf{sum}(\mathtt{succ}(a);b;\mathtt{succ}(c))} \tag{1.10b}$$

The rules define a ternary (three-place) relation, $\mathsf{sum}(a;b;c)$, among natural numbers a, b, and c. We may show that c is determined by a and b in this relation.

Theorem 1.4. *For every a nat and b nat, there exists a unique c nat such that $\mathsf{sum}(a;b;c)$.*

Proof. The proof decomposes into two parts:

1. (Existence) If a nat and b nat, then there exists c nat such that $\mathsf{sum}(a;b;c)$.
2. (Uniqueness) If a nat, b nat, c nat, c' nat, $\mathsf{sum}(a;b;c)$, and $\mathsf{sum}(a;b;c')$, then $c = c'$ nat.

For existence, let $\mathcal{P}(a\ \mathsf{nat})$ be the proposition *if* b nat *then there exists* c nat *such that* $\mathsf{sum}(a;b;c)$. We prove that if a nat then $\mathcal{P}(a\ \mathsf{nat})$ by rule induction on Rules (1.2). We have two cases to consider:

Rule (1.2a) We are to show $\mathcal{P}(\mathtt{zero}\ \mathsf{nat})$. Assuming b nat and taking c to be b, we obtain $\mathsf{sum}(\mathtt{zero};b;c)$ by Rule (1.10a).

Rule (1.2b) Assuming $\mathcal{P}(a\ \mathsf{nat})$, we are to show $\mathcal{P}(\mathtt{succ}(a)\ \mathsf{nat})$. That is, we assume that if b nat then there exists c such that $\mathsf{sum}(a;b;c)$, and are to show that if b' nat, then there exists c' such that $\mathsf{sum}(\mathtt{succ}(a);b';c')$. To this end, suppose that b' nat. Then by induction there exists c such that $\mathsf{sum}(a;b';c)$. Taking $c' = \mathtt{succ}(c)$, and applying Rule (1.10b), we obtain $\mathsf{sum}(\mathtt{succ}(a);b';c')$, as required.

For uniqueness, we prove that *if* $\mathsf{sum}(a;b;c_1)$, *then if* $\mathsf{sum}(a;b;c_2)$, *then* $c_1 = c_2$ nat by rule induction based on Rules (1.10).

Rule (1.10a) We have $a = \mathtt{zero}$ and $c_1 = b$. By an inner induction on the same rules, we may show that if $\mathsf{sum}(\mathtt{zero};b;c_2)$, then c_2 is b. By Lemma 1.1 on page 8 we obtain $b = b$ nat.

Rule (1.10b) We have that $a = \mathtt{succ}(a')$ and $c_1 = \mathtt{succ}(c_1')$, where $\mathsf{sum}(a';b;c_1')$. By an inner induction on the same rules, we may show that if $\mathsf{sum}(a;b;c_2)$, then $c_2 = \mathtt{succ}(c_2')$ nat where $\mathsf{sum}(a';b;c_2')$. By the outer inductive hypothesis $c_1' = c_2'$ nat and so $c_1 = c_2$ nat.

□

1.7 Modes

The statement that one or more arguments of a judgement is (perhaps uniquely) determined by its other arguments is called a *mode specification* for that judgement. For example, we have shown that every two natural numbers have a sum according to Rules (1.10). This fact may be restated as a mode specification by saying that the judgement $\mathsf{sum}(a;b;c)$ has *mode* $(\forall, \forall, \exists)$. The notation arises from the form of the proposition it expresses: *for all* a nat *and for all* b nat, *there exists* c nat *such that* $\mathsf{sum}(a;b;c)$. If we wish

to further specify that c is *uniquely* determined by a and b, we would say that the judgement $\mathsf{sum}(a;b;c)$ has mode $(\forall,\forall,\exists!)$, corresponding to the proposition *for all* a nat *and for all* b nat, *there exists a unique* c nat *such that* $\mathsf{sum}(a;b;c)$. If we wish only to specify that the sum is unique, *if it exists*, then we would say that the addition judgement has mode $(\forall,\forall,\exists^{\leq 1})$, corresponding to the proposition *for all* a nat *and for all* b nat *there exists at most one* c nat *such that* $\mathsf{sum}(a;b;c)$.

As these examples illustrate, a given judgement may satisfy several different mode specifications. In general the universally quantified arguments are to be thought of as the *inputs* of the judgement, and the existentially quantified arguments are to be thought of as its *outputs*. We usually try to arrange things so that the outputs come after the inputs, but it is not essential that we do so. For example, addition also has the mode $(\forall,\exists^{\leq 1},\forall)$, stating that the sum and the first addend uniquely determine the second addend, if there is any such addend at all. Put in other terms, addition of natural numbers has a (partial) inverse, namely subtraction! We could equally well show that addition has mode $(\exists^{\leq 1},\forall,\forall)$, which is just another way of stating that addition has a partial inverse over the natural numbers.

Often there is an intended, or *principal*, mode of a given judgement, which we often foreshadow by our choice of notation. For example, when giving an inductive definition of a function, we often use equations to indicate the intended input and output relationships. For example, we may re-state the inductive definition of addition (given by Rules (1.10)) using equations:

$$\frac{a\ \mathsf{nat}}{a + \mathtt{zero} = a\ \mathsf{nat}} \tag{1.11a}$$

$$\frac{a + b = c\ \mathsf{nat}}{a + \mathtt{succ}(b) = \mathtt{succ}(c)\ \mathsf{nat}} \tag{1.11b}$$

When using this notation we tacitly incur the obligation to prove that the mode of the judgement is such that the object on the right-hand side of the equations is determined as a function of those on the left. Having done so, we abuse notation, writing $a + b$ for the unique c such that $a + b = c$ nat.

1.8 Foundations

An inductively defined judgement form, such as a nat, may be seen as "carving out" a particular class of objects from an (as yet unspecified) *universe of discourse* that is rich enough to include the objects in question. That is, among the objects in the universe, the judgement a nat isolates those

objects of the form `succ(...succ(zero)...)`. But what, precisely, are these objects? And what sorts of objects are permissible in an inductive definition?

One answer to these questions is to fix in advance a particular set to serve as the universe over which all inductive definitions are to take place. This set must be proved to exist on the basis of the standard axioms of set theory, and the objects that we wish to use in our inductive definitions must be encoded as elements of this set. But what set shall we choose as our universe? And how are the various objects of interest encoded within it?

At the least we wish to include all possible *finitary trees* whose nodes are labelled by an element of an infinite set of *operators*. For example, the object `succ(succ(zero))` may be considered to be a tree of height two whose root is labelled with the operator `succ` and whose sole child is also so labelled and has a child labelled `zero`. Judgements with multiple arguments, such as $a + b = c$ `nat`, may be handled by demanding that the universe also be closed under formation of finite tuples $(a_1, \ldots, a_n)$ of objects. It is also possible to consider other forms of objects, such as *infinitary trees*, whose nodes may have infinitely many children, or *regular trees*, whose nodes may have ancestors as children, but we shall not have need of these in our work.

To construct a set of finitary objects requires that we fix a representation of trees and tuples as certain sets. This can be done, but the results are notoriously unenlightening.[1] Instead we shall simply assert that such a set exists (that is, can be constructed from the standard axioms of set theory). The construction should ensure that we can construct any finitary tree, and, given any finitary tree, determine the operator at its root and the set of trees that are its children.

While many will feel more secure by working within set theory, it is important to keep in mind that accepting the axioms of set theory is far more dubious, foundationally speaking, than just accepting the existence of finitary trees without recourse to encoding them as sets. Moreover, there is a significant disadvantage to working with sets. If we use abstract sets to model computational phenomena, we incur the additional burden of showing that these set-theoretic constructions can all be implemented on a computer. In contrast, it is intuitively clear how to represent finitary trees on a computer, and how to compute with them by recursion, so no further

[1] Perhaps you have seen the definition of the natural number 0 as the empty set, $\emptyset$, and the number $n + 1$ as the set $n \cup \{ n \}$, or the definition of the ordered pair $\langle a, b \rangle = \{ a, \{ a, b \} \}$. Similar coding tricks can be used to represent any finitary tree.

explanation is required.

1.9 Exercises

1. Give an inductive definition of the judgement $\mathsf{max}(a; b; c)$, where a nat, b nat, and c nat, with the meaning that c is the larger of a and b. Prove that this judgement has the mode $(\forall, \forall, \exists!)$.

2. Consider the following rules, which define the height of a binary tree as the judgement $\mathsf{hgt}(a; b)$.

$$\frac{}{\mathsf{hgt}(\mathtt{empty}; \mathtt{zero})} \tag{1.12a}$$

$$\frac{\mathsf{hgt}(a_1; b_1) \quad \mathsf{hgt}(a_2; b_2) \quad \mathsf{max}(b_1; b_2; b)}{\mathsf{hgt}(\mathtt{node}(a_1; a_2); \mathtt{succ}(b))} \tag{1.12b}$$

   Prove by tree induction that the judgement $\mathtt{hgt}$ has the mode $(\forall, \exists!)$, with inputs being binary trees and outputs being natural numbers.

3. Give an inductive definition of the judgement "∇ is a derivation of J" for an inductively defined judgement J of your choice.

4. Give an inductive definition of the forward-chaining and backward-chaining search strategies.

Chapter 2

Hypothetical Judgements

A *categorical* judgement is an unconditional assertion about some object of the universe. The inductively defined judgements given in Chapter 1 are all categorical. A *hypothetical judgement* expresses an *entailment* between one or more *hypotheses* and a *conclusion*. We will consider two notions of entailment, called *derivability* and *admissibility*. Derivability expresses the stronger of the two forms of entailment, namely that the conclusion may be deduced directly from the hypotheses by composing rules. Admissibility expresses the weaker form, that the conclusion is derivable from the rules whenever the hypotheses are also derivable. Both forms of entailment share a common set of *structural* properties that characterize conditional reasoning. One consequence of these properties is that derivability is stronger than admissibility (but the converse fails, in general). We then generalize the concept of an inductive definition to admit rules that have not only categorical, but also hypothetical, judgements as premises. Using these we may enrich the rule set with new axioms that are available for use within a specified premise of a rule.

2.1 Derivability

For a given set, $\mathcal{R}$, of rules, we define the *derivability* judgement, written $J_1, \ldots, J_k \vdash_{\mathcal{R}} K$, where each J_i and K are categorical, to mean that we may derive K using rules $\mathcal{R} \cup \{ J_1, \ldots, J_k \}$.[1] That is, we treat the *hypotheses*, or *antecedents*, of the judgement, $J_1, \ldots, J_n$ as *temporary axioms*, and derive the *conclusion*, or *consequent*, by composing rules in $\mathcal{R}$. That is, evidence for a

[1] Here we are treating the judgements J_i as axioms, or rules with no premises.

hypothetical judgement consists of a derivation of the conclusion from the hypotheses using the rules in $\mathcal{R}$.

We often use capital Greek letters, frequently Γ or Δ, to stand for a finite set of categorical judgements, writing $\Gamma \vdash_{\mathcal{R}} K$ to mean that K is derivable from rules $\mathcal{R} \cup \Gamma$ (that is, regarding the hypotheses as axioms). We sometimes write $\vdash_{\mathcal{R}} \Gamma$, where Γ is the finite set $\{ J_1, \ldots, J_k \}$, to mean that $\vdash_{\mathcal{R}} J_i$ for each $1 \leq i \leq k$. The derivability judgement $J_1, \ldots, J_n \vdash_{\mathcal{R}} J$ is sometimes expressed by saying that the rule

$$\frac{J_1 \quad \ldots \quad J_n}{J} \tag{2.1}$$

is *derivable* according to the rules $\mathcal{R}$.

For example, the derivability judgement

$$a \ \mathtt{nat} \vdash \mathtt{succ(succ(}a\mathtt{))\ nat} \tag{2.2}$$

is valid relative to Rules (1.2) for *any* choice of object a. Evidence for this is provided by the derivation

$$\frac{\dfrac{a \ \mathtt{nat}}{\mathtt{succ(}a\mathtt{)\ nat}}}{\mathtt{succ(succ(}a\mathtt{))\ nat}}, \tag{2.3}$$

which composes Rules (1.2), starting with a nat as an axiom, and ending with succ(succ(a)) nat. This may be equivalently expressed by stating that the rule

$$\frac{a \ \mathtt{nat}}{\mathtt{succ(succ(}a\mathtt{))\ nat}} \tag{2.4}$$

is derivable relative to Rules (1.2).

It follows directly from the definition of derivability that it is stable under extension with new rules.

Theorem 2.1 (Uniformity)**.** *If* $\Gamma \vdash_{\mathcal{R}} J$*, then* $\Gamma \vdash_{\mathcal{R} \cup \mathcal{R}'} J$*.*

Proof. Any derivation of J from $\mathcal{R} \cup \Gamma$ is also a derivation from $\mathcal{R} \cup \mathcal{R}' \cup \Gamma$, since the presence of additional rules does not influence the validity of the derivation. □

Derivability enjoys a number of *structural properties* that follow from its definition, independently of the rule set, $\mathcal{R}$, in question.

Reflexivity Every judgement is a consequence of itself: $\Gamma, J \vdash J$. Each hypothesis justifies itself as conclusion.

Weakening If $\Gamma \vdash J$, then $\Gamma, K \vdash J$. Entailment is not influenced by unexercised options.

Exchange If $\Gamma_1, J_1, J_2, \Gamma_2 \vdash J$, then $\Gamma_1, J_2, J_1, \Gamma_2 \vdash J$. The relative ordering of the axioms is immaterial.

Contraction If $\Gamma, J, J \vdash K$, then $\Gamma, J \vdash K$. We may use a hypothesis as many times as we like in a derivation.

Transitivity If $\Gamma, K \vdash J$ and $\Gamma \vdash K$, then $\Gamma \vdash J$. If we replace an axiom by a derivation of it, the result is a derivation of its consequent without that hypothesis.

These properties may be summarized by saying that derivability is *structural*.

Theorem 2.2. *For any rule set $\mathcal{R}$, the derivability judgement $\Gamma \vdash_{\mathcal{R}} J$ is structural.*

Proof. Reflexivity follows directly from the meaning of derivability. Weakening follows directly from uniformity. Exchange and contraction are inherent in treating rules as sets. Transitivity is proved by rule induction on the first premise. □

2.2 Admissibility

Admissibility, written $\Gamma \models_{\mathcal{R}} J$, is a weaker form of hypothetical judgement stating that $\vdash_{\mathcal{R}} \Gamma$ implies $\vdash_{\mathcal{R}} J$. That is, the conclusion J is derivable from rules $\mathcal{R}$ whenever the assumptions Γ are all derivable from rules $\mathcal{R}$. In particular if any of the hypotheses are *not* derivable relative to $\mathcal{R}$, then the judgement is vacuously true. The admissibility judgement $J_1, \ldots, J_n \models_{\mathcal{R}} J$ is sometimes expressed by stating that the rule,

$$\frac{J_1 \quad \ldots \quad J_n}{J}, \tag{2.5}$$

is *admissible* relative to the rules in $\mathcal{R}$.

For example, the admissibility judgement

$$\mathsf{succ}(a)\ \mathsf{nat} \models a\ \mathsf{nat} \tag{2.6}$$

is valid, because any derivation of $\mathtt{succ}(a)\ \mathtt{nat}$ from Rules (1.2) must contain a sub-derivation of $a\ \mathtt{nat}$ from the same rules, which justifies the conclusion. This may equivalently be expressed by saying that the rule

$$\frac{\mathtt{succ}(a)\ \mathtt{nat}}{a\ \mathtt{nat}} \tag{2.7}$$

is admissible relative to Rules (1.2).

In contrast to derivability the admissibility judgement is *not* stable under extension to the rules. For example, if we enrich Rules (1.2) with the axiom

$$\frac{}{\mathtt{succ(junk)\ nat}} \tag{2.8}$$

(where `junk` is some object for which $\mathtt{junk}\ \mathtt{nat}$ is not derivable), then the admissibility (2.6) is *invalid*. This is because Rule (2.8) has no premises, and there is no composition of rules deriving $\mathtt{junk}\ \mathtt{nat}$.

This example shows that admissibility is sensitive to which rules are *absent* from, as well as to which rules are *present* in, an inductive definition.

The structural properties of derivability given by Theorem 2.2 on the preceding page ensure that derivability is stronger than admissibility.

Theorem 2.3. *If* $\Gamma \vdash_{\mathcal{R}} J$*, then* $\Gamma \models_{\mathcal{R}} J$*.*

Proof. Repeated application of the transitivity of derivability shows that if $\Gamma \vdash_{\mathcal{R}} J$ and $\vdash_{\mathcal{R}} \Gamma$, then $\vdash_{\mathcal{R}} J$. □

To see that the converse fails, observe that there is no composition of rules such that

$$\mathtt{succ(junk)\ nat} \vdash_{(1.2)} \mathtt{junk\ nat},$$

yet the admissibility judgement

$$\mathtt{succ(junk)\ nat} \models_{(1.2)} \mathtt{junk\ nat}$$

holds vacuously.

Evidence for admissibility may be thought of as a mathematical function transforming derivations $\nabla_1, \ldots, \nabla_n$ of the hypotheses into a derivation ∇ of the consequent. Therefore, the admissibility judgement enjoys the same structural properties as derivability, and hence is a form of hypothetical judgement:

Reflexivity If J is derivable from the original rules, then J is derivable from the original rules: $J \models J$.

Weakening If J is derivable from the original rules assuming that each of the judgements in Γ are derivable from these rules, then J must also be derivable assuming that Γ and also K are derivable from the original rules: if $\Gamma \mid\!= J$, then $\Gamma, K \mid\!= J$.

Exchange The order of assumptions in an iterated implication does not matter.

Contraction Assuming the same thing twice is the same as assuming it once.

Transitivity If $\Gamma, K \mid\!= J$ and $\Gamma \mid\!= K$, then $\Gamma \mid\!= J$. If the assumption K is used, then we may instead appeal to the assumed derivability of K.

Theorem 2.4. *The admissibility judgement $\Gamma \mid\!=_{\mathcal{R}} J$ is structural.*

Proof. Follows immediately from the definition of admissibility as stating that if the hypotheses are derivable relative to $\mathcal{R}$, then so is the conclusion. □

2.3 Hypothetical Inductive Definitions

It is useful to enrich the concept of an inductive definition to permit rules with derivability judgements as premises and conclusions. Doing so permits us to introduce *local hypotheses* that apply only in the derivation of a particular premise, and also allows us to constrain inferences based on the *global hypotheses* in effect at the point where the rule is applied.

A *hypothetical inductive definition* consists of a collection of *hypothetical rules* of the form

$$\frac{\Gamma\,\Gamma_1 \vdash J_1 \quad \ldots \quad \Gamma\,\Gamma_n \vdash J_n}{\Gamma \vdash J}\ . \tag{2.9}$$

The hypotheses Γ are the *global hypotheses* of the rule, and the hypotheses Γ_i are the *local hypotheses* of the ith premise of the rule. Informally, this rule states that J is a derivable consequence of Γ whenever each J_i is a derivable consequence of Γ, augmented with the additional hypotheses Γ_i. Thus, one way to show that J is derivable from Γ is to show, in turn, that each J_i is derivable from $\Gamma\,\Gamma_i$. The derivation of each premise involves a "context switch" in which we extend the global hypotheses with the local hypotheses of that premise, establishing a new global hypothesis set for use within that derivation.

Often a hypothetical rule is given for each choice of global context, without restriction. In that case the rule is said to be *pure*, because it applies irrespective of the context in which it is used. A pure rule, being stated uniformly for all global contexts, may be given in *implicit* form, as follows:

$$\frac{\Gamma_1 \vdash J_1 \quad \ldots \quad \Gamma_n \vdash J_n}{J} \ . \tag{2.10}$$

This formulation omits explicit mention of the global context in order to focus attention on the local aspects of the inference.

Sometimes it is necessary to restrict the global context of an inference, so that it applies only a specified *side condition* is satisfied. Such rules are said to be *impure*. Impure rules generally have the form

$$\frac{\Gamma\,\Gamma_1 \vdash J_1 \quad \ldots \quad \Gamma\,\Gamma_n \vdash J_n \quad \Psi}{\Gamma \vdash J} \ , \tag{2.11}$$

where the condition, Ψ, limits the applicability of this rule to situations in which it is true. For example, Ψ may restrict the global context of the inference to be empty, so that no instances involving global hypotheses are permissible.

What does it mean to use the derivability consequence relation for a set of rules in one of the rules of that very set? One way to justify this is to consider a hypothetical inductive definition to be an ordinary inductive definition of a *formal derivability judgement*, $\Gamma \vdash J$, for which the following *structural rules* must be admissible:

$$\frac{}{\Gamma, J \vdash J} \tag{2.12a}$$

$$\frac{\Gamma \vdash J}{\Gamma, K \vdash J} \tag{2.12b}$$

$$\frac{\Gamma \vdash K \quad \Gamma, K \vdash J}{\Gamma \vdash J} \tag{2.12c}$$

In the common case that all rules in a definition are pure, the structural rules (2.12b) and (2.12c) are easily seen to be admissible by rule induction. It is necessary to include Rule (2.12a) explicitly to ensure that the formal derivability enjoys the expected structural properties.

Just as with an ordinary inductive definition, we say that a property, $\mathcal{P}$, of judgements $\Gamma \vdash J$ is *closed under* a hypothetical rule

$$\frac{\Gamma\,\Gamma_1 \vdash J_1 \quad \ldots \quad \Gamma\,\Gamma_n \vdash J_n}{\Gamma \vdash J} \tag{2.13}$$

if, and only if,

$$\mathcal{P}(\Gamma\,\Gamma_1 \vdash J_1),\ \ldots,\ \mathcal{P}(\Gamma\,\Gamma_n \vdash J_n)\ \textit{imply}\ \mathcal{P}(\Gamma \vdash J).$$

For example, $\mathcal{P}$ is closed under reflexivity iff $\mathcal{P}(\Gamma, J \vdash J)$.

For a given set of hypothetical rules, $\mathcal{R}$, the judgement $\Gamma \vdash_{\mathcal{R}} J$ is defined to be the strongest formal entailment closed under $\mathcal{R}$. It follows that this judgement is closed under the rules $\mathcal{R}$ in that if $\Gamma\,\Gamma_1 \vdash_{\mathcal{R}} J_1$, and $\ldots$ and $\Gamma\,\Gamma_n \vdash_{\mathcal{R}} J_n$, then $\Gamma \vdash_{\mathcal{R}} J$. The principle of *hypothetical rule induction* is valid: to show that $\mathcal{P}(\Gamma \vdash J)$ whenever $\Gamma \vdash_{\mathcal{R}} J$, it is enough to show, for each rule of the form (2.9), if $\mathcal{P}(\Gamma, \Gamma_1 \vdash J_1)$, and $\ldots$ and $\mathcal{P}(\Gamma, \Gamma_n \vdash J_n)$, then $\mathcal{P}(\Gamma \vdash J)$. We will make frequent use of this induction principle throughout the book.

2.4 Exercises

1. Prove that if all rules in a hypothetical inductive definition are pure, then the structural rules of weakening (Rule (2.12b)) and transitivity (Rule (2.12c)) are admissible.

2. Define $\Gamma' \vdash \Gamma$ to mean that $\Gamma' \vdash J_i$ for each J_i in Γ. Show that $\Gamma \vdash J$ iff whenever $\Gamma' \vdash \Gamma$, it follows that $\Gamma' \vdash J$. *Hint*: from left to right, appeal to transitivity of entailment; from right to left, consider the case of $\Gamma' = \Gamma$.

3. Show that it is dangerous to permit admissibility judgements in the premise of a rule. *Hint*: show that using such rules one may "define" an inconsistent judgement form J for which we have $a\ J$ iff it is *not* the case that $a\ J$.

Chapter 3

Generic Judgements

Just as hypothetical judgements express reasoning under hypotheses, so *generic judgements* express reasoning *generically* with respect to unspecified objects. There are two forms of generic judgement, the *parametric*, or *uniform*, and the *non-parametric*, or *non-uniform*. The parametric form captures reasoning that is completely independent of one or more objects, which are represented by *parameters*, much as the derivability judgement expresses reasoning with respect to unjustified axioms. The non-parametric form captures reasoning that depends on the specific choice of objects, much as the admissibility judgement expresses reasoning based on the possible forms of derivation of the hypotheses. The concept of a hypothetical inductive definition may be generalized to admit either form of generic judgement as premises, resulting in the concept of a *generic inductive definition*, which we shall use heavily throughout the book.

3.1 Objects and Parameters

We will enrich the class of objects that we consider to include *parameters*, or *indeterminates*, that may be replaced by other objects, themselves perhaps involving parameters, by a process known as *substitution*, or *instantiation*. For the purposes of this chapter we need not be specific about the exact nature of objects, parameters, or substitution, but can instead rely only on a specification of a few operations and relations on them.

We assume given an infinite class of objects, called *parameters*, that are distinct from all other objects (so that we know a parameter when we see one, and cannot confuse parameters with other forms of object). We generally use the variables, x, y, and z to stand for parameters, and we use the

variables $\mathcal{X}$ and $\mathcal{Y}$ to stand for finite sets of parameters. We write $\mathcal{X}, x$ for $\mathcal{X} \cup \{ x \}$, and $\mathcal{X}\,\mathcal{Y}$ for $\mathcal{X} \cup \mathcal{Y}$.

We take as given a judgement $\mathcal{X} \vdash a$ obj specifying that a is an object all of whose parameters lie within the set $\mathcal{X}$. Observe that if $\mathcal{X} \vdash a$ obj and $x \notin \mathcal{X}$, then the parameter x cannot occur within the object a. If $\emptyset \vdash a$ obj, then we say that a is a *closed* object; otherwise, a is said to be an *open* object. Observe that the judgement $\mathcal{X} \vdash a$ obj is closed under expansion of the set of parameters, and that $\mathcal{X}, x \vdash x$ obj.

We further assume that parameters may be *renamed* at will, provided that no two parameters are confused in the process. If $\mathcal{X}, x \vdash a_x$ obj is an object involving parameter x, and $y \notin \mathcal{X}$, then $\mathcal{X}, y \vdash a_y$ obj as well. We also assume given a function, called *substitution*, or *instantiation*, and written $[a/x]b$, with the property that if $\mathcal{X} \vdash a$ obj and $\mathcal{X}, x \vdash b$ obj, then $\mathcal{X} \vdash [a/x]b$ obj. More generally, we make use of *simultaneous substitution*, written $[a_1, \ldots, a_n/x_1, \ldots, x_n]a$, with the evident meaning. We assume that substitution is associative in the sense that

$$[a/x][b/y]c = [[a/x]b/y][a/x]c.$$

In practice it is usually necessary to distinguish multiple categories of objects and parameters, and to restrict renaming and substitution to respect the classes of the objects and parameters involved. It is straightforward to generalize the material in this chapter to account for multiple categories of objects and parameters.

3.2 Rule Schemes

We begin by making precise the informal concepts of rules and rule schemes discussed in Chapter 1. Recall that a rule scheme is a rule that involves meta-variables standing for unspecified objects. An instance of a rule scheme is obtained by replacing these meta-variables with specific objects. We may now make these informal concepts precise using the mechanisms of parameterization and substitution.

A *rule scheme* is a configuration of the form

$$x_1, \ldots, x_n \;\Big|\; \frac{J_1 \quad \ldots \quad J_k}{J} \tag{3.1}$$

consisting of a rule prefixed by a finite set of parameters containing *all* of the parameters that may occur in the premises or conclusion of the rule

scheme. No other parameters may occur in the rule other than those specified in its prefix. An *instance* of the rule scheme (3.1) is obtained by substituting objects $a_1, \ldots, a_n$ for the parameters $x_1, \ldots, x_n$. For example, Rule (1.2b) may be presented as a rule scheme by writing it in the form

$$x \mid \frac{x\ \mathsf{nat}}{\mathtt{succ}(x)\ \mathsf{nat}}\ .$$

Its instances, obtained by replacing the parameter, x, by an object, a, are rules of the form

$$\frac{a\ \mathsf{nat}}{\mathtt{succ}(a)\ \mathsf{nat}}\ .$$

A rule scheme stands for the totality of its instances. Consequently, we say that a property, $\mathcal{P}$, is *closed under* a rule scheme iff it is closed under all of its instances. More precisely, $\mathcal{P}$ is closed under a rule scheme of the form (3.1) iff for every choice of objects $a_1, \ldots, a_n$, if

$$\mathcal{P}([a_1, \ldots, a_n/x_1, \ldots, x_n]J_1) \textit{ and } \ldots \textit{ and } \mathcal{P}([a_1, \ldots, a_n/x_1, \ldots, x_n]J_k),$$

then

$$\mathcal{P}([a_1, \ldots, a_n/x_1, \ldots, x_n]J).$$

The judgement form defined by a set, $\mathcal{R}$, of rule schemes is defined to be the strongest judgement form closed under the rule schemes in $\mathcal{R}$.

3.3 Uniform Genericity

The *parametric*, or *uniform*, *derivability* judgement, written $\mathcal{X} \mid \Gamma \vdash_{\mathcal{R}} J$, states that the categorical judgement J is derivable from rules $\mathcal{R}$ and hypotheses Γ uniformly in the parameters $\mathcal{X}$. (When Γ is empty, we abbreviate the uniform derivability judgement to $\mathcal{X} \mid_{\mathcal{R}} J$.) By "uniformly" we mean that the parameters $\mathcal{X}$ are to be regarded as *fresh* objects, distinct from all others, in the derivation of J from Γ according to the rule schemes, $\mathcal{R}$. Evidence for $\mathcal{X} \mid \Gamma \vdash_{\mathcal{R}} J$ consists of a *derivation scheme*, ∇, involving the parameters in $\mathcal{X}$ that composes rules in $\mathcal{R}$ to obtain J from the assumptions in Γ. An *instance* of a derivation scheme is obtained by replacing the parameters of the scheme with chosen objects to obtain a derivation.

Just as the hypotheses Γ are to be considered "local" to ∇, so also are the parameters $\mathcal{X}$. We do not distinguish two derivation schemes that differ only in the choice of parameter names used to represent the fresh objects of the derivation. The parameters serve only as a kind of "pronoun", referring to some unspecified concrete object to be determined later; the exact

"word" we use for the pronoun is not relevant, all that matters is that the referent be unambiguous.

For example, evidence for the uniform derivability judgement

$$x \mid x\ \mathsf{nat} \vdash_{(1.2)} \mathtt{succ(succ}(x)\mathtt{)}\ \mathsf{nat} \tag{3.2}$$

consists of the derivation scheme, ∇_x, given as follows:

$$\dfrac{\dfrac{\overline{x\ \mathsf{nat}}}{\mathtt{succ}(x)\ \mathsf{nat}}}{\mathtt{succ(succ}(x)\mathtt{)}\ \mathsf{nat}}\ . \tag{3.3}$$

We could just as well have used y throughout as a parameter of the scheme without affecting its meaning.

By choosing an object, a, we obtain an instance

$$a\ \mathsf{nat} \vdash \mathtt{succ(succ}(a)\mathtt{)}\ \mathsf{nat}, \tag{3.4}$$

of the uniform derivability judgement. Similarly, the derivation scheme ∇_x may be specialized to a to obtain the derivation

$$\dfrac{\dfrac{\overline{a\ \mathsf{nat}}}{\mathtt{succ}(a)\ \mathsf{nat}}}{\mathtt{succ(succ}(a)\mathtt{)}\ \mathsf{nat}}\ . \tag{3.5}$$

The resulting derivation is evidence for (3.4).

The parametric derivability judgement enjoys a collection of *structural properties* that follow directly from its meaning:

Renaming If $\mathcal{X}, x \mid \Gamma \vdash_{\mathcal{R}} J_x$, then $\mathcal{X}, y \mid \Gamma \vdash_{\mathcal{R}} J_y$, provided that $y \notin \mathcal{X}$.

Proliferation If $\mathcal{X} \mid \Gamma \vdash_{\mathcal{R}} J$ and $x \notin \mathcal{X}$, then $\mathcal{X}, x \mid \Gamma \vdash_{\mathcal{R}} J$.

Swapping If $\mathcal{X}_1, x_1, x_2, \mathcal{X}_2 \mid \Gamma \vdash_{\mathcal{R}} J$, then $\mathcal{X}_1, x_2, x_1, \mathcal{X}_2 \mid \Gamma \vdash_{\mathcal{R}} J$.

Duplication If $\mathcal{X}, x, x \mid \Gamma \vdash_{\mathcal{R}} J$, then $\mathcal{X}, x \mid \Gamma \vdash_{\mathcal{R}} J$.

Instantiation If $\mathcal{X}, x \mid \Gamma \vdash_{\mathcal{R}} J$, and $\mathcal{X} \vdash a\ \mathsf{obj}$, then $\mathcal{X} \mid [a/x]\Gamma \vdash_{\mathcal{R}} [a/x]J$.

Renaming states that a uniform derivability judgement is invariant under renaming of parameters. With respect to the structural properties of the hypothetical judgement, proliferation corresponds to weakening, swapping corresponds to exchange, duplication corresponds to contraction, and instantiation corresponds to transitivity.

Theorem 3.1. *The uniform derivability judgement is structural.*

Proof. Renaming follows from the identification of two parametric judgements that differ only in the names of bound variables. Proliferation is a direct consequence of uniformity. Swapping and duplication are inherent in considering the parameters of the judgement to be a set. Instantiation may be proved by rule induction on the judgement $\mathcal{X}, x \mid \Gamma \vdash_{\mathcal{R}} J$, where $x \notin \mathcal{X}$. Consider an arbitrary rule of the form (3.1), and suppose that we have evidence for each of the premises

$$\mathcal{X}, x \mid \Gamma \vdash_{\mathcal{R}} [a_1, \ldots, a_n / x_1, \ldots, x_n] J_i,$$

where $1 \leq i \leq k$. By the inductive hypothesis we have evidence for

$$\mathcal{X} \mid [a/x]\Gamma \vdash_{\mathcal{R}} [a/x][a_1, \ldots, a_n / x_1, \ldots, x_n] J_i$$

for each $1 \leq i \leq k$. By suitable renaming the parameter x may be chosen apart from $x_1, \ldots, x_n$, and hence does not occur in any premise of the rule. It follows that we have evidence for the judgement

$$\mathcal{X} \mid [a/x]\Gamma \vdash_{\mathcal{R}} [[a/x]a_1, \ldots, [a/x]a_n / x_1, \ldots, x_n] J_i.$$

We may then re-instantiate the same rule scheme to obtain the conclusion

$$\mathcal{X} \mid [a/x]\Gamma \vdash_{\mathcal{R}} [[a/x]a_1, \ldots, [a/x]a_n / x_1, \ldots, x_n] J,$$

which is to say

$$\mathcal{X} \mid [a/x]\Gamma \vdash_{\mathcal{R}} [a/x][a_1, \ldots, a_n / x_1, \ldots, x_n] J.$$

□

3.4 Non-Uniform Genericity

The *non-parametric*, or *non-uniform*, *derivability* judgement, $\mathcal{X} \parallel \Gamma \vdash_{\mathcal{R}} J$, where $\mathcal{X} = x_1, \ldots, x_n$, states that for every *closed* instance $a_1, \ldots, a_n$ of the parameters, the derivability judgement

$$[a_1, \ldots, a_n / x_1, \ldots, x_n]\Gamma \vdash [a_1, \ldots, a_n / x_1, \ldots, x_n] J$$

is valid. Since only closed instances are considered, evidence for non-uniform derivability may assign a different derivation for each instance. This is analogous to the distinction between admissibility and derivability

discussed in Chapter 2. Whereas derivability is uniform (does not depend on the evidence for the hypotheses, if any), admissibility is non-uniform (depends on the evidence for the hypotheses to determine evidence for the conclusion).

Theorem 3.2. *The non-uniform derivability judgement is structural.*

Proof. This is an immediate consequence of the definition, using familiar properties of universal quantification and implication. □

Consequently, uniform derivability is stronger than non-uniform derivability.

Theorem 3.3. *If* $\mathcal{X} \mid \Gamma \vdash_{\mathcal{R}} J$*, then* $\mathcal{X} \parallel \Gamma \vdash_{\mathcal{R}} J$*.*

Proof. Follows directly from Theorem 3.1 on the previous page. □

3.5 Generic Inductive Definitions

A *generic inductive definition* is a generalization of a hypothetical inductive definition in which we permit expansion not only of the rules, but also of the set of active parameters, in each premise of a rule. A *generic hypothetical rule* has the form

$$\frac{\mathcal{X}\,\mathcal{X}_1 \mid \Gamma\,\Gamma_1 \vdash J_1 \quad \ldots \quad \mathcal{X}\,\mathcal{X}_n \mid \Gamma\,\Gamma_n \vdash J_n}{\mathcal{X} \mid \Gamma \vdash J}\ . \tag{3.6}$$

The set, $\mathcal{X}$, is the set of *global parameters* of the inference, and, for each $1 \leq i \leq n$, the set $\mathcal{X}_i$ is the set of *fresh local parameters* of the ith premise. The local parameters are *fresh* in the sense that, by suitable renaming, they may be chosen to be disjoint from the global parameters of the inference. The pair $\mathcal{X} \mid \Gamma$ is called the *global context* of the rule, and each pair $\mathcal{X}_i \mid \Gamma_i$ is called the *local context* of the ith premise of the rule.

A generic rule is *pure* if it is stated for all choices of global context, subject only to the freshness requirement on local parameters. Such a rule may be written in *implicit form* as follows:

$$\frac{\mathcal{X}_1 \mid \Gamma_1 \vdash J_1 \quad \ldots \quad \mathcal{X}_n \mid \Gamma_n \vdash J_n}{J}\ . \tag{3.7}$$

This form of the rule stands for all rules of the form Rule (3.6) obtained by specifying the global context $\mathcal{X} \mid \Gamma$.

As with hypothetical inductive definitions, we regard a generic inductive definition as defining a *formal generic entailment*, written $\mathcal{X} \mid \Gamma \vdash J$, that is expresses uniform derivability with respect to the rules themselves. To ensure that the formal uniform derivability judgement is well-behaved, the following *structural rules* are admissible in any generic inductive definition:

$$\frac{\mathcal{X} \mid \Gamma \vdash J}{\mathcal{X}, x \mid \Gamma \vdash J} \tag{3.8a}$$

$$\frac{\mathcal{X}, x \mid \Gamma \vdash J \quad \mathcal{X} \vdash a \ \mathsf{obj}}{\mathcal{X} \mid [a/x]\Gamma \vdash [a/x]J} \tag{3.8b}$$

Admissibility of the structural properties is assured if all rules are pure, otherwise it must be explicitly proved for each generic inductive definition.

The principle of rule induction for a generic inductive definition states that to show $\mathcal{P}(\mathcal{X} \mid \Gamma \vdash J)$ whenever $\mathcal{X} \mid \Gamma \vdash J$ is derivable, it is enough to show that $\mathcal{P}$ is closed under the rules comprising the definition. Specifically, for each rule of the form (3.6), we must show that

$$\textit{if } \mathcal{P}(\mathcal{X}\, \mathcal{X}_1 \mid \Gamma\, \Gamma_1 \vdash J_1),\ \ldots,\ \mathcal{P}(\mathcal{X}\, \mathcal{X}_n \mid \Gamma\, \Gamma_n \vdash J_n),\ \textit{then } \mathcal{P}(\mathcal{X} \mid \Gamma \vdash J).$$

Observe that, by our identification of two generic judgements that differ only in the names of their parameters, the property, $\mathcal{P}$, is not permitted to distinguish between any two such judgements. This limits the class of properties that we may consider to those that are well-defined with respect to this identification, but experience shows that all natural properties (including those of interest in this book) respect this equivalence. This means that the proof of $\mathcal{P}(\mathcal{X} \mid \Gamma \vdash J)$ need only be carried out for one particular choice of parameters, and that this choice may be tacitly assumed to satisfy any finite freshness requirements we may wish to impose.

3.6 Exercises

Chapter 4

Transition Systems

Transition systems are used to describe the execution behavior of programs by defining an abstract computing device with a set, S, of *states* that are related by a *transition judgement*, $\mapsto$. The transition judgement describes how the state of the machine evolves during execution.

4.1 Transition Systems

An *(ordinary) transition system* is specified by the following judgements:

1. s state, asserting that s is a *state* of the transition system.
2. s final, where s state, asserting that s is a *final* state.
3. s initial, where s state, asserting that s is an *initial* state.
4. $s \mapsto s'$, where s state and s' state, asserting that state s may transition to state s'.

We require that if s final, then for no s' do we have $s \mapsto s'$. In general, a state s for which there is no $s' \in S$ such that $s \mapsto s'$ is said to be *stuck*, which may be indicated by writing $s \not\mapsto$. All final states are stuck, but not all stuck states need be final!

A *transition sequence* is a sequence of states $s_0, \ldots, s_n$ such that s_0 initial, and $s_i \mapsto s_{i+1}$ for every $0 \leq i < n$. A transition sequence is *maximal* iff $s_n \not\mapsto$, and it is *complete* iff it is maximal and, in addition, s_n final. Thus every complete transition sequence is maximal, but maximal sequences are not necessarily complete. A transition system is *deterministic* iff for every

state s there exists at most one state s' such that $s \mapsto s'$, otherwise it is *non-deterministic*.

A *labelled transition system* over a set of labels, I, is a generalization of a transition system in which the single transition judgement, $s \mapsto s'$ is replaced by an I-indexed family of transition judgements, $s \overset{i}{\mapsto} s'$, where s and s' are states of the system. In typical situations the family of transition relations is given by a simultaneous inductive definition in which each rule may make reference to any member of the family.

It is often necessary to consider families of transition relations in which there is a distinguished unlabelled transition, $s \mapsto s'$, in addition to the indexed transitions. It is sometimes convenient to regard this distinguished transition as labelled by a special, anonymous label not otherwise in I. For historical reasons this distinguished label is often designated by τ or ϵ, but we will simply use an unadorned arrow. The unlabelled form is often called a *silent* transition, in contrast to the labelled forms, which announce their presence with a label.

4.2 Iterated Transition

Let $s \mapsto s'$ be a transition judgement, whether drawn from an indexed set of such judgements or not.

The *iteration* of transition judgement, $s \mapsto^* s'$, is inductively defined by the following rules:

$$\frac{}{s \mapsto^* s} \tag{4.1a}$$

$$\frac{s \mapsto s' \quad s' \mapsto^* s''}{s \mapsto^* s''} \tag{4.1b}$$

It is easy to show that iterated transition is transitive: if $s \mapsto^* s'$ and $s' \mapsto^* s''$, then $s \mapsto^* s''$.

The principle of rule induction for these rules states that to show that $P(s, s')$ holds whenever $s \mapsto^* s'$, it is enough to show these two properties of P:

1. $P(s, s)$.
2. if $s \mapsto s'$ and $P(s', s'')$, then $P(s, s'')$.

The first requirement is to show that P is reflexive. The second is to show that P is *closed under head expansion*, or *converse evaluation*. Using this principle, it is easy to prove that $\mapsto^*$ is reflexive and transitive.

The *n-times iterated* transition judgement, $s \mapsto^n s'$, where $n \geq 0$, is inductively defined by the following rules.

$$\frac{}{s \mapsto^0 s} \tag{4.2a}$$

$$\frac{s \mapsto s' \quad s' \mapsto^n s''}{s \mapsto^{n+1} s''} \tag{4.2b}$$

Theorem 4.1. *For all states s and s', $s \mapsto^* s'$ iff $s \mapsto^k s'$ for some $k \geq 0$.*

Finally, we write $s \downarrow$ to indicate that there exists some s' final such that $s \mapsto^* s'$.

4.3 Simulation and Bisimulation

A *strong simulation* between two transition systems $\mapsto_1$ and $\mapsto_2$ is given by a binary relation, $s_1 \, S \, s_2$, between their respective states such that if $s_1 \, S \, s_2$, then $s_1 \mapsto_1 s_1'$ implies $s_2 \mapsto_2 s_2'$ for some state s_2' such that $s_1' \, S \, s_2'$. Two states, s_1 and s_2, are *strongly similar* iff there is a strong simulation, S, such that $s_1 \, S \, s_2$. Two transition systems are strongly similar iff each initial state of the first is strongly similar to an initial state of the second. Finally, two states are *strongly bisimilar* iff there is a single relation S such that both S and its converse are strong simulations.

A strong simulation between two labelled transition systems over the same set, I, of labels consists of a relation S between states such that for each $i \in I$ the relation S is a strong simulation between $\overset{i}{\mapsto}_1$ and $\overset{i}{\mapsto}_2$. That is, if $s_1 \, S \, s_2$, then $s_1 \overset{i}{\mapsto}_1 s_1'$ implies that $s_2 \overset{i}{\mapsto}_2 s_2'$ for some s_2' such that $s_1' \, S \, s_2'$. In other words the simulation must preserve labels, and not just transitions.

The requirements for strong simulation are rather stringent: every step in the first system must be mimicked by a similar step in the second, up to the simulation relation in question. This means, in particular, that a sequence of steps in the first system can only be simulated by a sequence of steps of the same length in the second—there is no possibility of performing "extra" work to achieve the simulation.

A *weak simulation* between transition systems is a binary relation between states such that if $s_1 \, S \, s_2$, then $s_1 \mapsto_1 s_1'$ implies that $s_2 \mapsto_2^* s_2'$ for some s_2' such that $s_1' \, S \, s_2'$. That is, every step in the first may be matched by zero or more steps in the second. A *weak bisimulation* is such that both it and its converse are weak simulations. We say that states s_1 and s_2 are *weakly (bi)similar* iff there is a weak (bi)simulation S such that $s_1 \, S \, s_2$.

The corresponding notion of weak simulation for labelled transitions involves the silent transition. The idea is that to weakly simulate the labelled transition $s_1 \overset{i}{\longmapsto}_1 s_1'$, we do not wish to permit multiple *labelled* transitions between related states, but rather to permit any number of *unlabelled* transitions to accompany the labelled transition. A relation between states is a *weak simulation* iff it satisfies both of the following conditions whenever $s_1 \, S \, s_2$:

1. If $s_1 \longmapsto_1 s_1'$, then $s_2 \longmapsto_2^* s_2'$ for some s_2' such that $s_1' \, S \, s_2'$.

2. If $s_1 \overset{i}{\longmapsto}_1 s_1'$, then $s_2 \longmapsto_2^* \overset{i}{\longmapsto}_2 \longmapsto_2^* s_2'$ for some s_2' such that $s_1' \, S \, s_2'$.

That is, every silent transition must be mimicked by zero or more silent transitions, and every labelled transition must be mimicked by a corresponding labelled transition, preceded and followed by any number of silent transitions. As before, a *weak bisimulation* is a relation between states such that both it and its converse are weak simulations. Finally, two states are *weakly (bi)similar* iff there is a weak (bi)simulation between them.

4.4 Exercises

1. Prove that S is a weak simulation for the ordinary transition system $\longmapsto$ iff S is a strong simulation for $\longmapsto^*$.