# 15-399 Supplementary Notes: Complete Induction

Robert Harper

March 24, 2005

## Complete Induction

The principle of mathematical induction states that if a property $P$ holds of $\mathbf{0}$ and is closed under successor, then it holds for every natural number:[1]

$$P(\mathbf{0}) \wedge (\forall z \in \mathbf{nat}.P(z) \supset P(\mathbf{s}(z))) \supset \forall x \in \mathbf{nat}.P(x).$$

Re-phrasing a bit, the principal of mathematical induction states that $P$ is *universal*, if it is *closed*. Defining

$$\mathsf{ALL}(P) = \forall x \in \mathbf{nat}.P(x)$$

and

$$\mathsf{CLOSED}(P) = P(\mathbf{0}) \wedge \forall z \in \mathbf{nat}.P(z) \supset P(\mathbf{s}(z)),$$

the principle of mathematical induction may be re-stated in the form

$$\mathsf{CLOSED}(P) \supset \mathsf{ALL}(P).$$

While mathematically sufficient, it can be inconvenient to structure a proof so that it relies only on this basic principle. A more convenient form is the principle of *complete induction*,[2] which states that $P$ is universal, if it is *complete*:

$$\mathsf{COMPL}(P) \supset \mathsf{ALL}(P),$$

where

$$\mathsf{COMPL}(P) = \forall x \in \mathbf{nat}.(\forall z \in \mathbf{nat}.z < x \supset P(z)) \supset P(x).$$

It is helpful to define

$$\mathsf{LT}(x)(P) = \forall z \in \mathbf{nat}.z < x \supset P(z),$$

[1] We write $P(t)$ to mean $[t/x]P$, where $x$ is a designated free variable of $P$ over which we are reasoning by induction.

[2] Also called *strong induction* or *complete induction*.

so that we may re-state completeness in the form

$$\mathsf{COMPL}(P) = \forall x \in \mathbf{nat}.\mathsf{LT}(x)(P) \supset P(x).$$

Thus the principle of complete induction states that to show $\mathsf{ALL}(P)$, it is enough to show that, for a general $x \in \mathbf{nat}$, $P(x)$ holds under the inductive assumption that $P$ holds for every number less than $x$. Thus, there is no separate base case, and the inductive hypothesis states that $P$ holds not just for the immediate predecessor, but for all smaller numbers. (Do you see why this covers the base case implicitly?)

The principle of complete induction is derivable from ordinary mathematical induction. That is, we may *prove* the principle of complete induction as a theorem, using the principle of ordinary mathematical induction in the proof. It follows that any use of complete induction is simply a disguised use of ordinary mathematical induction.

# Proof of Complete Induction

**Theorem 0.1 (Complete Induction)**

$$\mathsf{COMPL}(P) \supset \mathsf{ALL}(P).$$

We first prove the following lemma:

**Lemma 0.1**

$$\mathsf{COMPL}(P) \supset \forall x \in \mathbf{nat}.\mathsf{LT}(x)(P).$$

**Proof:** Assume $\mathsf{COMPL}(P)$, and let $x \in \mathbf{nat}$ be arbitrary. We show $\mathsf{LT}(x)(P)$ by ordinary mathematical induction on $x$.

1. For the basis, we are to show $\mathsf{LT}(0)(P)$. Let $z \in \mathbf{nat}$ and assume $z < 0$. This is a contradiction; consequently, $P(z)$ holds, as required.

2. For the inductive step, assume $\mathsf{LT}(x')(P)$ to show $\mathsf{LT}(\mathbf{s}(x'))(P)$. Let $z \in \mathbf{nat}$ be such that $z < \mathbf{s}(x')$. By a simple lemma, either $z = x'$ or $z < x'$. We show $P(z)$ by cases:

   (a) If $z = x'$, then since $\mathsf{COMPL}(P)$, it is enough to show $\mathsf{LT}(x')(P)$. But this is precisely the inductive assumption!

   (b) If $z < x'$, then by the inductive hypothesis $P(z)$, as required.

□

We may now prove the complete induction theorem:

**Proof:** Assume that $\mathsf{COMPL}(P)$. Suppose that $x \in \mathbf{nat}$; we are to show $P(x)$. By the lemma, we have $\mathsf{LT}(\mathbf{s}(x))(P)$, and hence $P(x)$ since $x < \mathbf{s}(x)$. This completes the proof. □

What is the computational content of this proof?

Suppressing proof terms for propositions, the statement of the lemma erases (and simplifies) to the type

$$(\mathbf{nat} \to (\mathbf{nat} \to \tau) \to \tau) \to (\mathbf{nat} \to (\mathbf{nat} \to \tau)).$$

The proof of the lemma erases to a term, $H$, of this type satisfying the equations

$$\begin{array}{rcl} H(c)(\mathbf{0})(n) & \Leftrightarrow & \mathbf{abort} \\ H(c)(\mathbf{s}(m))(n) & \Leftrightarrow & c(m)(H(c)(m)) \quad (m = n) \\ H(c)(\mathbf{s}(m))(n) & \Leftrightarrow & H(c)(m)(n) \quad (n < m). \end{array}$$

The first argument to $H$ is the proof of completeness of the predicate $P$; the second governs the primitive recursion (it decreases by one on each recursive call); the third is the "real" argument, which may decrease by more than one on each call.

The statement of complete induction erases (and simplifies) to the type

$$(\mathbf{nat} \to (\mathbf{nat} \to \tau) \to \tau) \to (\mathbf{nat} \to \tau).$$

The proof of complete induction erases to a term, $I$, of this type such that

$$I(c)(m) \Leftrightarrow H(c)(\mathbf{s}(m))(m).$$

## Example

Pfenning gives a proof of the totality of the discrete logarithm function by complete induction. The discrete logarithm is defined by the following equations:

$$\begin{array}{rcl} \mathbf{lg}(\mathbf{0}) & \Leftrightarrow & \mathbf{0} \\ \mathbf{lg}(\mathbf{s}(m)) & \Leftrightarrow & \mathbf{s}(\mathbf{lg}(\mathbf{half}(\mathbf{s}(m)))) \end{array}$$

Informally, $\mathbf{lg}(n) = \lfloor \log_2(n+1) \rfloor$. Pfenning gives a proof of the proposition

$$\forall x \in \mathbf{nat}.\exists y \in \mathbf{nat}.\, \mathbf{lg}(x) = y.$$

This states that **lg** is a total function in the sense that every natural number has a discrete logarithm according to the foregoing definition.

The core of the proof is to show that the predicate $P(x)$ defined by

$$P(x) = \exists y \in \mathbf{nat}.\, \mathbf{lg}(x) = y$$

is complete, from which the result follows by complete induction. If $t$ is the (erased) term corresponding to the proof of completeness of $P(x)$, then $I(t)$ is (the erasure of) a proof of $\mathsf{ALL}(P)$. Consequently, if $m \in \mathbf{nat}$, then $I(t)(m) \Leftrightarrow n$, where $\mathbf{lg}(m) = n$. Expanding the definition of $I$ given above, we have $H(t)(\mathbf{s}(m))(m) \Leftrightarrow \mathbf{lg}(m)$ for every $m \in \mathbf{nat}$.

To see how this works, observe that Pfenning's proof of completeness for the predicate $P(x)$ defines a term $t$ of type $\mathbf{nat} \rightarrow (\mathbf{nat} \rightarrow \mathbf{nat}) \rightarrow \mathbf{nat}$ satisfying the equations

$$\begin{array}{rcl} t(\mathbf{0})(f) & \Leftrightarrow & \mathbf{0} \\ t(\mathbf{s}(m))(f) & \Leftrightarrow & \mathbf{s}(f(\mathbf{half}(\mathbf{s}(m)))) \end{array}$$

Plugging this into the equations for complete induction we obtain

$$\begin{array}{rcl} I(t)(m) & \Leftrightarrow & H(t)(\mathbf{s}(m))(m) \\ \\ H(t)(\mathbf{0})(n) & \Leftrightarrow & \mathbf{abort} \\ H(t)(\mathbf{s}(m))(n) & \Leftrightarrow & t(m)(H(t)(m)) \quad (n = m) \\ H(t)(\mathbf{s}(m))(n) & \Leftrightarrow & H(t)(m)(n) \quad (n < m) \end{array}$$

Plugging in the term $t$ obtained from the logarithm example, and simplifying, we obtain these equations:

$$\begin{array}{rcl} H(t)(\mathbf{s}(\mathbf{0}))(\mathbf{0}) & \Leftrightarrow & t(\mathbf{0})(H(t)(\mathbf{0})) \\ & \Leftrightarrow & \mathbf{0} \\ \\ H(t)(\mathbf{s}(\mathbf{s}(m')))(\mathbf{s}(m')) & \Leftrightarrow & t(\mathbf{s}(m'))(H(t)(\mathbf{s}(m'))) \\ & \Leftrightarrow & \mathbf{s}(H(t)(\mathbf{s}(m'))(\mathbf{half}(\mathbf{s}(m')))) \\ \\ H(t)(\mathbf{s}(\mathbf{s}(m')))(n) & \Leftrightarrow & H(t)(\mathbf{s}(m'))(n)\ (n < \mathbf{s}(m')) \end{array}$$

In essence $H$ computes by "counting down" in its second argument until it reaches one more than the third argument, then applies $t$, which calls $H$ with half of the third argument. The countdown repeats until finally the third argument is zero.

From these we may deduce the following equations for $I(t)$:

$$\begin{array}{rcl} I(t)(\mathbf{0}) & = & \mathbf{0} \\ I(t)(\mathbf{s}(m)) & = & \mathbf{s}(I(t)(\mathbf{half}(\mathbf{s}(m)))) \quad (m \in \mathbf{nat}). \end{array}$$

In other words $I(t)(m) = \mathbf{lg}(m)$ for every $m \in \mathbf{nat}$, as required, and is, moreover, primitive recursive (because $H$ is primitive recursive).