consists of a binary function symbol $F$. Hence by 11.2 and 11.3, it will suffice to construct a model $M'$ of PO such that $M$ is definable in an inessential extension of $M'$.

Let $M_1 = |M| \cup \{1,2,3\}$, where 1,2,3 are objects not in $|M|$. Let $M_2$ be the set of ordered pairs $(x,i)$ where $x \in |M|$ and $i \in \{1,2,3\}$. Let $M_3$ be the set of ordered triples $(x,y,z)$ such that $x,y,z \in |M|$ and $F_M(x,y) = z$. Let $|M'| = M_1 \cup M_2 \cup M_3$. We define $<_{M'}$ as follows. If $x \in M_1$, $w <_{M'} x$ is false for all $w$. If $<x,i> \in M_2$, then $w <_{M'} <x,i>$ holds for $w = x$ and $w = i$. If $<x,y,z> \in M_3$, then $w <_{M'} <x,y,z>$ if $w$ is one of $<x,1>$, $<y,2>$, $<z,3>$, $x$, $y$, $z$, 1, 2, or 3. Clearly $M'$ is a partially ordered set.

For $x,y,z \in |M'|$ we have

$$x \in |M| \leftrightarrow \forall y \neg(y < x) \;\&\; x \neq 1 \;\&\; x \neq 2 \;\&\; x \neq 3,$$

$$F(x,y) = z \leftrightarrow x,y,z \in |M| \;\&\; \exists u \exists x_1 \exists y_1 \exists z_1 (x < x_1 \;\&\; y < y_1 \;\&\; z < z_1 \;\&$$
$$1 < x_1 \;\&\; 2 < y_1 \;\&\; 3 < z_1 \;\&\; x_1 < u \;\&\; y_1 < u \;\&\; z_1 < u).$$

(In proving the second equivalence from right to left, one should first note that we must have $x_1, y_1, z_1 \in M_2$ and $u \in M_3$.) It follows easily that $M$ is definable in $M''$, where $M''$ is an inessential expansion of $M'$ formed by adding three new constants to represent 1, 2, and 3.

It follows that PO is undecidable. It also follows that a theory whose language consists of one binary relation symbol and which has no axioms is undecidable.

Many other strongly undecidable structures can be constructed by these methods. However, the proof that $M$ is definable in $M'$ often requires a very detailed analysis of $M$ and $M'$.

## 12. Relative Recursion

Let $\Phi$ be a set of *total* functions. We generalize the notion of computable to allow us to use the values of the functions in $\Phi$ at any arguments we wish in the course of the computation. Following Turing, we picture the computation as