16

Representability of Recursive Functions

In the preceding chapter we connected our work on recursion with our work on formulas and proofs in one way, by showing that various functions associated with formulas and proofs are recursive. In this chapter we connect the two topics in the opposite way, by showing how we can 'talk about' recursive functions using formulas, and prove things about them in theories formulated in the language of arithmetic. In section 16.1 we show that for any recursive function f, we can find a formula ϕ_f *such that for any natural numbers a and b, if* $f(a) = b$ *then* $\forall y(\phi_f(\mathbf{a}, y) \leftrightarrow y = \mathbf{b})$ *will be true in the standard interpretation of the language of arithmetic. In section 16.2 we strengthen this result, by introducing a theory* **Q** *of* minimal arithmetic*, and showing that for any recursive function f, we can find a formula* ψ_f *such that for any natural numbers a and b, if* $f(a) = b$ *then* $\forall y(\psi_f(\mathbf{a}, y) \leftrightarrow y = \mathbf{b})$ *will be not merely true, but provable in* **Q**. *In section 16.3 we briefly introduce a stronger theory* **P** *of* Peano arithmetic*, which includes axioms of* mathematical induction*, and explain how these axioms enable us to prove results not obtainable in* **Q**. *The brief, optional section 16.4 is an appendix for readers interested in comparing our treatment of these matters here with other treatments in the literature.*

16.1 Arithmetical Definability

In Chapter 9, we introduced the language L^* of arithmetic and its standard interpretation $\mathcal{N}^*$. We now abbreviate 'true in the standard interpretation' to *correct*. Our goal in this chapter is to show that we can 'talk about' recursive functions in the language of arithmetic, and we begin by making talk about 'talking about' precise. We say a formula $F(x)$ of the language of arithmetic *arithmetically defines* a set S of natural numbers if and only if for all natural numbers a we have Sa if and only if $F(\mathbf{a})$ is correct. We say the set S is *arithmetically definable*, or *arithmetical* for short, if some formula arithmetically defines it. These notions naturally extend to two-place or many-place relations. A formula $F(x, y)$ arithmetically defines a relation R on natural numbers if and only if for all natural numbers a and b we have Rab if and only if $F(\mathbf{a}, \mathbf{b})$ is correct. The notions also naturally extend to functions, a function being counted as arithmetical if and only if its graph relation is arithmetical. Thus a one-place function f is arithmetical if and only if there is a formula $F(x, y)$ of the language of arithmetic such that for all a and b we have $f(a) = b$ if and only if $F(\mathbf{a}, \mathbf{b})$ is correct.

16.1 Examples (Basic functions). To give the most trivial example, the identity function $\mathrm{id} = \mathrm{id}^1_1$is arithmetically defined by the formula $y = x$, and more generally, id^n_i is arithmetically defined by the formula $y = x_i$, or if we want the other x_j to be mentioned, by the formula

$$x_1 = x_1 \,\&\, \ldots \,\&\, x_n = x_n \,\&\, y = x_i .$$

The zero function $\mathrm{const}_0(x) = 0$ is also arithmetically definable, by the formula $y = \mathbf{0}$, or if we want x to be mentioned, by the formula $x = x \;\&\; y = \mathbf{0}$. The successor, addition, and multiplication functions are arithmetically definable by the formulas $y = x'$, $y = x_1 + x_2$, and $y = x_1 \cdot x_2$.

16.2 Examples (Other arithmetical functions). Of course, it is no surprise that the functions we have just been considering are arithmetically definable, since they are 'built in': we have included in the language special symbols expressly for them. But their inverses, for which we have not built in symbols, are also arithmetical. The predecessor function is arithmetically definable by the following formula $F_{\mathrm{pred}}(x_1, y)$:

$$(x_1 = \mathbf{0} \,\&\, y = \mathbf{0}) \vee x_1 = y' .$$

The difference function $x_1 \dot{-} x_2$ is arithmetically defined by the following formula $F_{\mathrm{dif}}(x_1, x_2, y)$:

$$(x_1 < x_2 \,\&\, y = \mathbf{0}) \vee (x_1 = x_2 + y)$$

and the quotient and remainder functions $\mathrm{quo}(x_1, x_2)$ and $\mathrm{rem}(x_1, x_2)$ are arithmetically defined by the following formulas $F_{\mathrm{quo}}(x_1, x_2, y)$ and $F_{\mathrm{rem}}(x_1, x_2, y)$:

$$(x_2 = \mathbf{0} \,\&\, y = \mathbf{0}) \vee \exists u < x_2 x_1 = y \cdot x_2 + u$$
$$(x_2 = \mathbf{0} \,\&\, y = x_1) \vee (y < x_2 \,\&\, \exists u \leq x_1 \, x_1 = u \cdot x_2 + y).$$

On the other hand, it is not obvious how to define exponentiation, and as a temporary expedient we now expand the language of arithmetic by adding a symbol $\uparrow$, thus obtaining the language of *exponential arithmetic*. Its standard interpretation is like that of the original language arithmetic, with the denotation of $\uparrow$ being the usual exponentiation function. In terms of this expansion we define $\uparrow$*-arithmetical definability* in the obvious way. (The expression '$\uparrow$-arithmetical' may be pronounced 'exponential-arithmetical' or 'exp-arithmetical' for short.)

16.3 Examples ($\uparrow$-arithmetical functions). Examples of $\uparrow$-arithmetical functions include the exponential function itself, its inverses the logarithm functions (lo and lg of Example 7.11), and, what will be more significant for our present purposes, any number of functions pertaining to the coding of finite sequences of numbers by single numbers or pairs of numbers. For instance, in section 1.2 we found one serviceable if not especially elegant way of coding sequences by pairs for which the ith entry of the sequence coded by the pair (s, t) could be recovered using the function

$$\mathrm{entry}(i, s, t) = \mathrm{rem}(\mathrm{quo}(s, t^i), t)$$

This function is $\uparrow$-arithmetically definable by the following formula $F_{\text{ent}}(x_1, x_2, x_3, y)$:

$$\exists z \leq x_3 \uparrow x_1 (F_{\text{quo}}\ (x_2, x_3 \uparrow x_1, z)\ \&\ F_{\text{rem}}\ (z, x_2, y)).$$

For this just says that there is something that is the quotient on dividing x_2 by $x_3^{x_1}$, and whose remainder on dividing by x_2 is y, adding that it will be less than or equal to x_2 (as any quotient on dividing x_2 by anything must be).

Even after helping ourselves to exponentiation, it is still not obvious how to define *super*-exponentiation, but though not obvious, it is possible—in fact *any* recursive function can now be defined, as we next show.

16.4 Lemma. Every recursive function f is $\uparrow$-arithmetical.

Proof: Since we have already shown the basic functions to be definable, we need only show that if any of the three processes of composition, primitive recursion, or minimization is applied to $\uparrow$-arithmetical functions, the result is an $\uparrow$-arithmetical function. We begin with composition, the idea for which was already encountered in the last example. Suppose that f and g are one-place functions and that h is obtained from them by composition. Then clearly $c = h(a)$ if and only if

$$c = g(f(a))$$

which may be more long-windedly put as

there is something such that it is $f(a)$ and g(it) is c.

It follows that if f and g are $\uparrow$-arithmetically defined by ϕ_f and ϕ_g, then h is $\uparrow$-arithmetically defined by the following formula $\phi_h(x, z)$:

$$\exists y\, (\phi_f(x, y)\ \&\ \phi_g(y, z)).$$

[To be a little more formal about it, given any a, let $b = f(a)$ and let $c = h(a) = g(f(a)) = g(b)$. Since ϕ_f and ϕ_g define f and g, $\phi_f(\mathbf{a}, \mathbf{b})$ and $\phi_g(\mathbf{b}, \mathbf{c})$ are correct, so $\phi_f(\mathbf{a}, \mathbf{b})\ \&\ \phi_g(\mathbf{b}, \mathbf{c})$ is correct, so $\exists y(\phi_f(\mathbf{a}, y)\ \&\ \phi_g(y, \mathbf{c}))$ is correct, which is to say $\phi_h(\mathbf{a}, \mathbf{c})$ is correct. Conversely, if $\phi_h(\mathbf{a}, \mathbf{c})$ is correct, $\phi_f(\mathbf{a}, \mathbf{b})\ \&\ \phi_g(\mathbf{b}, \mathbf{c})$ and hence $\phi_f(\mathbf{a}, \mathbf{b})$ and $\phi_g(\mathbf{b}, \mathbf{c})$ must be correct for some b, and since ϕ_f defines f, this b can only be $f(a)$, while since ϕ_g defines g, c then can only be $g(b) = g(f(a)) = h(a)$.]

For the composition of a two-place function f with a one-place function g the formula would be

$$\exists y(\phi_f(x_1, x_2, y)\ \&\ \phi_g(y, z)).$$

For the composition of two one-place functions f_1 and f_2 with a two-place function g, the formula would be

$$\exists y_1 \exists y_2 (\phi_{f_1}(x, y_1)\ \&\ \phi_{f_2}(x, y_2)\ \&\ \phi_g(y_1, y_2, z))$$

and so on. The construction is similar for functions of more places.

Recursion is just a little more complicated. Suppose that f and g are one-place and three-place functions, respectively, and that the two-place function h is obtained

from them by primitive recursion. Writing i' for the successor of i, clearly $c = h(a, b)$ if and only if there exists a sequence σ with the following three properties:

entry 0 of σ is $h(a, 0)$
for all $i < b$, if entry i of σ is $h(a, i)$, then entry i' of σ is $h(a, i')$
entry b of σ is c.

These conditions may be restated equivalently thus:

entry 0 of σ is $f(a)$
for all $i < b$, entry i' of σ is $g(a, i,$ entry i of $\sigma)$
entry b of σ is c.

These conditions may be restated more long-windedly thus:

there is something that is entry 0 of σ and is $f(a)$
for all $i < b$, there is something that is entry i of σ, and
    there is something which is entry i' of σ, and
    the latter is $g(a, i,$ the former)
entry b of σ is c.

Moreover, instead of saying 'there is a sequence' we may say 'there are two numbers coding a sequence'. It follows that if f and g are $\uparrow$-arithmetically defined by ϕ_f and ϕ_g, then h is $\uparrow$-arithmetically defined by the formula $\phi_h(x, y, z) = \exists s \exists t \phi$, where ϕ is the conjunction of the following three formulas:

$$\exists u(F_{\text{ent}}(\mathbf{0}, s, t, u) \,\&\, \phi_f(x, u))$$
$$\forall w < y \, \exists u \exists v(F_{\text{ent}}(w, s, t, u) \,\&\, F_{\text{ent}}(w', s, t, v) \,\&\, \phi_g(x, w, u, v))$$
$$F_{\text{ent}}(y, s, t, z).$$

The construction is exactly the same for functions of more places.

Minimization is a little simpler. Suppose that f is a two-place function, and that the one-place function g is obtained from it by minimization. Clearly $g(a) = b$ if and only if

$f(a, b) = 0$ and
for all $c < b$, $f(a, c)$ is defined and is not 0.

These conditions may be restated more long-windedly thus:

$f(a, b) = 0$ and
for all $c < b$, there is something that is $f(a, c)$, and it is not 0.

It follows that if f is $\uparrow$-arithmetically defined by ϕ_f, then g is $\uparrow$-arithmetically defined by the following formula $\phi_g(x, y)$:

$$\phi_f(x, y, \mathbf{0}) \,\&\, \forall z < y \, \exists u(\phi_f(x, z, u) \,\&\, u \neq \mathbf{0}).$$

The construction is exactly the same for functions of more places.

On reviewing the above construction, it will be seen that the presence of the exponential symbol $\uparrow$ in the language was required only for the formula F_{ent}. If we

could find some *other* way to code sequences by pairs for which the entry function could be defined *without* using exponentiation, then we could forget about $\uparrow$. And in fact, a coding is possible for which

$$\text{entry}(i, s, t) = \text{rem}(s, t(i+1)+1)$$

so that for F_{ent} we may take

$$F_{\text{rem}}(x_2, x_3 \cdot (x_1 \mathbf{+ 1}) \mathbf{+ 1}, y).$$

That such a coding is possible is the content of the following lemma.

16.5 Lemma (β-function lemma). For every k and every $a_0, a_1, \ldots, a_k$ there exist s and t such that for all i with $0 \leq i \leq k$ we have $a_i = \text{rem}(s, t(i+1)+1)$.

Proof: This result follows directly from the proofs of two ancient and famous theorems of number theory, to be found in a prominent place in any textbook on that subject. Since this is not a textbook on number theory, we are not going to develop the whole subject from the foundations, but we do give an indication of the proof. The first ingredient is the *Chinese remainder theorem*, so called from the appearance (at least of special cases) of the theorem in the ancient *Mathematical Classic* of Sun Zi and the medieval *Mathematical Treatise in Nine Sections* of Qin Jiushao. This theorem states that given any numbers $t_0, t_1, \ldots, t_n$ no two of which have a common prime factor, and given any numbers $a_i < t_i$, there is a number s such that $\text{rem}(s, t_i) = a_i$ for all i from 0 to n. The proof is sufficiently illustrated by the case of two numbers t and u with no common prime factor, and two numbers $a < t$ and $b < u$. Every one of the tu numbers i with $0 \leq i < tu$ produces one of the tu pairs (a, b) with $a < t$ and $b < u$ on taking the remainders rem (s, t) and rem (s, u). To show that, as asserted by the theorem, every pair (a, b) is produced by some number s, it suffices to show that no two distinct numbers $0 \leq s < r < tu$ produce the same pair. If s and r do produce the same pair, then they leave the same remainder when divided by t, and leave the same remainder when divided by u. In that case, their difference $q = r - s$ leaves remainder zero when divided by either t or u. In other words, t and u both divide q. But when numbers with no common prime factor both divide a number, so does their product. Hence tu divides q. But this is impossible, since $0 < q < tu$.

The second ingredient comes from the proof in Euclid's *Elements of Geometry* that there exist infinitely many primes. Given any number n, we want to find a prime $p > n$. Well, let $N = n!$, so that in particular N is divisible by every prime $\leq n$. Then $N + 1$, like any number >1, has a prime factor p. (Possibly N is itself prime, in which case we have $p = N$.) But we cannot have $p \leq n$, since when N is divided by any number $\leq n$, there is a remainder of 1. A slight extension of the argument shows that any two distinct numbers $N \cdot i + 1$ and $N \cdot j + 1$ with $0 < i < j \leq n$ have no common prime factor. For if a prime p divides both numbers, it divides their difference $N(j - i)$. This is a product of factors $\leq n$, and when a prime divides a product of several factors, it must divide one of the factors; so p itself must be a number $\leq n$. But then p can*not* divide $N \cdot i + 1$ or $N \cdot j + 1$. Now given k and every $a_0, a_1, \ldots, a_k$, taking n larger than all of them, and letting t be a number divisible by every prime $\leq n$, no two of the

numbers $t_i = t(i+1)+1$ will have a common prime factor, and we will of course have $a_i < t_i$, so there will be an s such that $\text{rem}(s, t_i) = a_i$ for all i with $0 \leq i \leq k$.

Thus we have proved part (a) of the following.

16.6 Lemma

(a) Every recursive function f is arithmetical.
(b) Every recursive set is arithmetical.

Proof: As remarked just before the statement of the lemma, we already have (a). For (b), if R is an n-place recursive relation and f its characteristic function, then apply (a) to get a formula $\phi(x_1, \ldots, x_n, y)$ arithmetically defining f. Then the formula $\phi(x_1, \ldots, x_n, \mathbf{1})$ arithmetically defines R.

Further sharpening of the result depends on distinguishing different kinds of formulas. By a *rudimentary formula* of the language of arithmetic we mean a formula built up from atomic formulas using only negation, conjunction, disjunction, and bounded quantifications $\forall x < t$ and $\exists x < t$, where t may be any term of the language (not involving x). (Conditionals and biconditionals are allowed, too, since these officially are just abbreviations for certain constructions involving negation, conjunction, and disjunction. So are the bounded quantifiers $\forall x \leq t$ and $\exists x \leq t$, since these are equivalent to $\forall x < t'$ and $\exists x < t'$). By an $\exists$-*rudimentary formula* we mean a formula of form $\exists x F$ where F is rudimentary, and similarly for an $\forall$-*rudimentary formula*. (The negation of an $\exists$-rudimentary formula is equivalent to an $\forall$-rudimentary formula, and conversely.) Many major theorems of number theory are naturally expressible by $\forall$-rudimentary formulas.

16.7 Examples (Theorems of number theory). *Lagrange's theorem* that every natural number is the sum of four squares is naturally expressible by an $\forall$-rudimentary sentence as follows:

$$\forall x\, \exists y_1 < x\, \exists y_2 < x\, \exists y_3 < x\, \exists y_4 < x\, x = y_1 \cdot y_1 + y_2 \cdot y_2 + y_3 \cdot y_3 + y_4 \cdot y_4.$$

Bertrand's postulate, or *Chebyshev's theorem*, that there is a prime between any number greater than one and its double, is naturally expressible by an $\forall$-rudimentary sentence as follows:

$$\forall x(\mathbf{1} < x \rightarrow \exists y < 2 \cdot x(x < y \,\&\, \sim\exists u < y\, \exists v < y\, y = u \cdot v)).$$

Our present concern, however, will be with $\exists$-rudimentary formulas and with *generalized* $\exists$-rudimentary formulas, which include all formulas obtainable from rudimentary formulas by conjunction, disjunction, bounded universal quantification, bounded existential quantification, and unbounded existential quantification. Reviewing the proof of Lemma 16.6, one finds that the formulas defining the basic functions and the formula F_{ent} are rudimentary, and that the formula defining a composition of functions is obtained by conjunction, bounded quantification, and existential quantification from rudimentary formulas and the formulas defining the original functions, and similarly for recursion and minimization. Hence we have proved:

16.8 Lemma. Every recursive function is arithmetically definable by a generalized ∃-rudimentary formula.

The next refinement will be to get rid of the word 'generalized' here. Two formulas with, say, two free variables, $\phi(x, y)$ and $\psi(x, y)$, are called *arithmetically equivalent* if for all numbers a and b, $\phi(\mathbf{a}, \mathbf{b})$ is correct if and only if $\psi(\mathbf{a}, \mathbf{b})$ is correct. Clearly arithmetically equivalent formulas define the same relation or function. The condition that ϕ and ψ are arithmetically equivalent is equivalent to the condition that the biconditional

$$\forall x \forall y(\phi(x, y) \leftrightarrow \psi(x, y))$$

is correct. In particular, if ϕ and ψ are logically equivalent—in which case the biconditional is true not just in the standard interpretation, but in *any* interpretation—then they are arithmetically equivalent. The following lemma bears more than a passing resemblance to Corollary 7.15.

16.9 Lemma (Closure properties of ∃-rudimentary formulas).

(a) Any rudimentary formula is arithmetically equivalent to an ∃-rudimentary formula.
(b) The conjunction of two ∃-rudimentary formulas is arithmetically equivalent to an ∃-rudimentary formula.
(c) The disjunction of two ∃-rudimentary formulas is arithmetically equivalent to an ∃-rudimentary formula.
(d) The result of applying bounded universal quantification to an ∃-rudimentary formula is arithmetically equivalent to an ∃-rudimentary formula.
(e) The result of applying bounded existential quantification to an ∃-rudimentary formula is arithmetically equivalent to an ∃-rudimentary formula.
(f) The result of applying (unbounded) existential quantification to an ∃-rudimentary formula is arithmetically equivalent to an ∃-rudimentary formula.

Proof: For (a), ϕ is logically equivalent to $\exists w(w = w \,\&\, \phi)$ (and if ϕ is rudimentary, so is $w = w \,\&\, \phi$).

For (b), $\exists u \phi(u) \,\&\, \exists v \psi(v)$ is arithmetically equivalent to

$$\exists w \, \exists u < w \, \exists v < w \, (\phi(u) \,\&\, \psi(v)).$$

[and if $\phi(u)$ and $\psi(v)$ are rudimentary, so is $\exists u < w \, \exists v < w \, (\phi(u) \,\&\, \psi(v))$]. The implication in one direction is logical, and in the other direction we use the fact that for any two natural numbers u and v, there is always a natural number w greater than both.

For (c), $\exists u \phi(u) \vee \exists v \psi(v)$ is logically equivalent to $\exists w(\phi(w) \vee \psi(w))$.

For (d), $\forall z < y \, \exists u \phi(u, z)$ is arithmetically equivalent to

$$\exists w \, \forall z < y \, \exists u < w \, \phi(u, z).$$

The implication in one direction is logical, and in the other direction we use the fact that for any finitely many natural numbers $u_0, u_1, \ldots, u_{y-1}$ there is a number w that is greater than all the u_z.

For (e), $\exists z < y\, \exists u\, \phi(u, z)$ is logically equivalent to $\exists u\, \exists z < y\, \phi(u, z)$.

For (f), $\exists u\, \exists v\, \phi(u, v)$ is arithmetically equivalent to $\exists w\, \exists u < w\, \exists v < w\, \phi(u, v)$, much as in part (b).

Repeated application of Lemma 16.9, followed by combination with Lemma 16.8 give the following:

16.10 Proposition. Every generalized $\exists$-rudimentary formula is arithmetically equivalent to an $\exists$-rudimentary formula.

16.11 Lemma. Every recursive function is arithmetically definable by an $\exists$-rudimentary formula.

Call a function that is arithmetically definable by a rudimentary formula a *rudimentary function*. Can we go further and show every recursive function to be rudimentary? Not quite. The next lemma tells us how far we *can* go. It bears more than a passing resemblance to Proposition 7.17.

16.12 Lemma. Every recursive function is obtainable by composition from rudimentary functions.

Proof: Let f be a recursive function of, say, one place. (The proof for many-place functions is exactly the same.) We know f is arithmetically definable by an $\exists$-rudimentary formula $\exists z \phi(x, y, z)$. Let S be the relation arithmetically defined by ϕ, so that we have

$$Sabc \leftrightarrow \phi(\mathbf{a}, \mathbf{b}, \mathbf{c}) \text{ is correct.}$$

We have

$$f(a) = b \leftrightarrow \exists c\, Sabc.$$

We now introduce two auxiliary functions:

$$g(a) = \begin{cases} \text{the least } d \text{ such that} \\ \quad \exists b < d\, \exists c < d\, Sabc & \text{if such a } d \text{ exists} \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$h(a, d) = \begin{cases} \text{the least } b < d \text{ such that} \\ \quad \exists c < d\, Sabc & \text{if such a } b \text{ exists} \\ 0 & \text{otherwise.} \end{cases}$$

(Note that if f is total, then g is total, while h is always total.) These functions are rudimentary, being arithmetically definable by the following formulas $\phi_g(x, w)$ and $\phi_h(x, w, y)$:

$$\exists y < w\, \exists z < w\, \phi(x, y, z) \;\&\; \forall v < w\, \forall y < v\, \forall z < v \sim\phi(x, y, z)$$
$$\exists z < w\, \phi(x, y, z) \;\&\; \forall u < y\, \forall z < w \sim\phi(x, u, z)$$

and a little thought shows that $f(x) = h(x, g(x)) = h(\mathrm{id}(x), g(x))$, so $f = \mathrm{Cn}[h, \mathrm{id}, g]$ is a composition of rudimentary functions.

If T is a consistent theory in the language of arithmetic, we say a set S is *defined* in T by $D(x)$ if for all n, if n is in S, then $D(\mathbf{n})$ is a theorem of T, and if n is not in S, then $\sim D(\mathbf{n})$ is a theorem of T. S is *definable* in T if S is defined by some formula. Arithmetical definability is simply the special case where T is *true arithmetic*, the set of all correct sentences. The general notion of definability in a theory extends to relations, but definability of a function turns out to be less useful than a related notion. For the remainder of this chapter, unless otherwise noted, 'function' will mean 'total function'. Let f be a one-place function. (The definition we are about to give extends easily to many-place functions.) We say f is *representable* in T if there is a formula $F(x, y)$ such that whenever $f(a) = b$, the following is a theorem of T:

$$\forall y(F(\mathbf{a}, y) \leftrightarrow y = \mathbf{b}).$$

This is logically equivalent to the conjunction of the positive assertion

$$F(\mathbf{a}, \mathbf{b})$$

and the general negative assertion

$$\forall y(y \neq \mathbf{b} \rightarrow \sim F(\mathbf{a}, y)).$$

By contrast, definability would only require that we have the positive assertion and for each particular $c \neq b$ the relevant particular instance of the general negative assertion, namely, $\sim F(\mathbf{a}, \mathbf{c})$.

Now in the special case where T is true arithmetic, of course if each particular numerical instance is correct, then the universal generalization is correct as well, so representability and definability come to the same thing. But for other theories, each particular numerical instance may be a theorem without the universal generalization being a theorem, and representability is in general a stronger requirement than definability. Note that if T is a *weaker* theory than T^* (that is, if the set of theorems of T is a subset of the set of theorems of T^*), then the requirement that a function be representable in T is a *stronger* requirement than that it be representable in T^* (that is, representability in T implies representability in T^*). Thus far we have proved all recursive functions to be representable in true arithmetic. If we are to strengthen our results, we must consider weaker theories than that.

16.2 Minimal Arithmetic and Representability

We now introduce a finite set of *axioms of minimal arithmetic* **Q**, which, though not strong enough to prove major theorems of number theory, at least are correct and strong enough to prove all correct $\exists$-rudimentary sentences. By themselves, the axioms of **Q** would not be adequate for number theory, but any set of adequate axioms would have to include them, or at least to prove them (in which case the set might as well include them). Our main theorems (Theorems 16.13 and 16.15) apply to any theory T that contains **Q**, and since **Q** is weak, the theorems are correspondingly strong.

In displaying the list of axioms we make use of a traditional convention, according to which when displaying sentences of the language of arithmetic that begin with a