definition of this function has a new clause for each r:

$$Reg(j,e,x,n+1) = H_r(Reg((i)_1,e,x,n)) \text{ if } (i)_0 = 3 \ \& \ (i)_3 = r \ \& \ (i)_2 = j.$$

This means that in the definition of $T_k^{\Phi}(e,\vec{x},y)$, H_r appears only in contexts $H_r(X)$ where X designates a number appearing in a register during the P–computation from $\vec{x}$ and hence $< y$. Thus we may replace $H_r(X)$ by $(\overline{H_r}(y))_X$.

If Φ is $H_1,\dots,H_m$, we write $\overline{\Phi}(z)$ for $\overline{H_1}(z),\dots,\overline{H_m}(z)$. The above can be summarized as follows: there is a recursive relation $T_{k,m}$ such that

$$(1) \qquad T_k^{\Phi}(e,\vec{x},y) \leftrightarrow T_{k,m}(e,\vec{x},y,\overline{\Phi}(y)).$$

Thus if $\{e\}^{\Phi}(\vec{x}) \simeq z$ with computation number y, and $\overline{\Phi}(y) = \overline{\Phi'}(y)$, then $\{e\}^{\Phi'}(\vec{x}) \simeq z$.

13. The Arithmetical Hierarchy

We are now going to study the effect of using unbounded quantifiers in definitions of relations. From now on, we agree that n designates a non–zero number. The results of this section are due to Kleene.

A relation R is <u>arithmetical</u> if it has an explicit definition

$$(1) \qquad R(\vec{x}) \leftrightarrow Qy_1 \dots Qy_n P(\vec{x},y_1,\dots,y_n)$$

where each Qy_i is either $\exists y_i$ or $\forall y_i$ and P is recursive. We call $Qy_1 \dots Qy_n$ the <u>prefix</u> and $P(\vec{x},y_1,\dots,y_n)$ the <u>matrix</u> of the definition. We are chiefly interested in the prefix, since it measures how far the definition is from being recursive.

We shall first see how prefixes can be simplified. As z runs through all number, $(z)_0,(z)_1$ runs through all pairs of numbers. It follows that

$$\forall x \forall y R(x,y) \leftrightarrow \forall z R((z)_0,(z)_1)$$

and

$$\exists x \exists y R(x,y) \leftrightarrow \exists z R((z)_0,(z)_1).$$

Using these equivalences, we can replace two adjacent universal quantifiers in a prefix by a single such quantifier, and similarly for existential quantifiers. For example, a definition

$$R(x) \leftrightarrow \forall y \forall z \exists v P(x,y,z,v)$$

can be replaced by

$$R(x) \leftrightarrow \forall w \exists v P(x,(w)_0,(w)_1,v).$$

Of course, the matrix has changed; but it is still a recursive function of its variables because $(w)_0$ and $(w)_1$ are recursive functions of w. This sort of simplification of a prefix is called contraction of quantifiers.

A prefix is alternating if it does not contain two successive existential quantifiers or two successive universal quantifiers. A prefix is Π^0_n if it is alternating, has n quantifiers, and begins with $\forall$. A prefix is Σ^0_n if it is alternating, has n quantifiers, and begins with $\exists$. A relation is Π^0_n if it has an explicit definition with a Π^0_n prefix and a recursive matrix; similarly for Σ^0_n. A relation is Δ^0_n if it is both Π^0_n and Σ^0_n. We sometimes use Π^0_n for the class of Π^0_n relations; similarly for Σ^0_n and Δ^0_n.

13.1. PROPOSITION. Every arithmetical relation is Π^0_n or Σ^0_n for some n.

Proof. By contraction of quantifiers. □

13.2. PROPOSITION. If R is Π^0_n or Σ^0_n, then R is Δ^0_k for every $k > n$. If R is recursive, then R is Δ^0_n for all n.

Proof. By adding superfluous quantifiers. For example, suppose that R is Π^0_2; say $R(x) \leftrightarrow \forall y \exists z P(x,y,z)$. To show that R is Δ^0_3, we note that

$$R(x) \leftrightarrow \forall y \exists z \forall w P(x,y,z)$$

$$\leftrightarrow \exists w \forall y \exists z P(x,y,z).$$ □

A relation P is many–one reducible, or simply reducible, to a relation Q if it has a definition

$$P(\vec{x}) \leftrightarrow Q(F_1(\vec{x}),\ldots,F_n(\vec{x}))$$

where each F_i is total and recursive. If P is reducible to Q and Q is recursive, then P is recursive. From this we obtain the following result.

13.3. PROPOSITION. If P is reducible to Q and Q is Π^0_n, then P is Π^0_n. The same holds with Σ^0_n or Δ^0_n in place of Π^0_n. □

The contraction formulas show that R and <R> are reducible to one another; so R is Π^0_n iff <R> is Π^0_n, and similarly for Σ^0_n and Δ^0_n.

We now consider the effect of applying propositional connectives to arithmetical relations. The key tools are the prenex rules, which are certain rules for bringing quantifiers to the front of an expression. They are

$$\neg QxR(x) \leftrightarrow Q'x\neg R(x),$$

$$QxR(x) \vee P \leftrightarrow Qx(R(x) \vee P),$$

$$P \vee QxR(x) \leftrightarrow Qx(P \vee R(x)),$$

$$QxR(x) \,\&\, P \leftrightarrow Qx(R(x) \,\&\, P),$$

$$P \,\&\, QxR(x) \leftrightarrow Qx(P \,\&\, R(x)),$$

where Q is either $\forall$ or $\exists$ and Q' is $\exists$ if Q is $\forall$ and $\forall$ if Q is $\exists$. These rules are well known and easily seen to be valid.

From the first rule (and the fact that $\neg$ is a recursive symbol) we see that the negation of a Π^0_n relation is Σ^0_n and the negation of a Σ^0_n relation is Π^0_n. For example, to see that the negation of a Π^0_2 relation is Σ^0_2, note that by the prenex rules

$$\neg\forall x\exists yP \leftrightarrow \exists x\forall y\neg P.$$

The next four rules together with contraction of quantifiers show that the disjunction and conjunction of two Π^0_n relations is Π^0_n; and similarly for Σ^0_n. For example, to treat the disjunction of two Π^0_2 relations, observe that by the prenex rules

$$\forall x\exists yP \vee \forall z\exists wQ \leftrightarrow \forall x\forall z\exists y\exists w(P \vee Q)$$

and then use contraction of quantifiers.

Now consider a definition $R(\vec{x}) \leftrightarrow \forall yP(\vec{x},y)$ where P is arithmetical. By replacing P in this definition by the right side of the definition of P and then using contraction of quantifiers if possible, we see that if P is Π^0_n, then R is Π^0_n; and if P is Σ^0_n, then R is Π^0_{n+1}. A similar result holds if $\forall y$ is replaced by $\exists y$.

Now we consider the effect of bounded quantifiers. We need the following equivalences:

$$(\forall y < x)\forall z R(x,y,z) \leftrightarrow \forall z(\forall y < x)R(x,y,z),$$

$$(\exists y < x)\exists z R(x,y,z) \leftrightarrow \exists z(\exists y < x)R(x,y,z),$$

$$(\forall y < x)\exists z R(x,y,z) \leftrightarrow \exists z(\forall y < x)R(x,y,(z)_y),$$

$$(\exists y < x)\forall z R(x,y,z) \leftrightarrow \forall z(\exists y < x)R(x,y,(z)_y).$$

The first two of these are obvious. Both sides of the third says that there is a sequence $z_0, \ldots, z_{x-1}$ such that $R(x,y,z_y)$ for all $y < x$. Now replace R by $\neg R$ in the third equivalence, bring the negation signs to the front by means of the prenex rules, and then drop the negations signs from the front of both sides of the equivalence. We then obtain the fourth equivalence.

Now consider a definition $R(\vec{x},z) \leftrightarrow (Qy < z)P(\vec{x},y,z)$ where P is arithmetical. Substitute the right side of the definition of P for P. We can then apply the above equivalences to bring all of the unbounded quantifiers to the left of $(Qy < z)$. Since bounded quantifiers are recursive, we may now consider $(Qy < z)$ as part of the matrix. It follows that if P is Π^0_n, then so is R; and similarly for Σ^0_n.

We can summarize our results in the following table, which gives the classification of various combinations of P and Q in terms of the classifications of P and Q.

P,Q	$\neg P$	$P \vee Q$	$P \& Q$	$\forall xP$	$\exists xP$	$(Qx<y)P$
Π^0_n	Σ^0_n	Π^0_n	Π^0_n	Π^0_n	Σ^0_{n+1}	Π^0_n
Σ^0_n	Π^0_n	Σ^0_n	Σ^0_n	Π^0_{n+1}	Σ^0_n	Σ^0_n
Δ^0_n	Δ^0_n	Δ^0_n	Δ^0_n	Π^0_n	Σ^0_n	Δ^0_n

(The last row of the table follows from the first two rows.) To treat the case in which P and Q do not have the same classification, we use 13.2. For example, if P is Π^0_2 and Q is Σ^0_2, then P and Q are Δ^0_3, and we can use the last row of the table. To treat $\rightarrow$ and $\leftrightarrow$, we replace $X \rightarrow Y$ by $\neg X \vee Y$ and $X \leftrightarrow Y$ by $(X \rightarrow Y)$ & $(Y \rightarrow X)$. Every recursion theorist should learn this table.

The classification of the arithmetical relations into Π_n^0 and Σ_n^0 relations is called the arithmetical hierarchy. We have not yet shown that the classes in this hierarchy are distinct.

Let Φ be a class of k–ary relations. We say that a $(k+1)$–ary relation Q enumerates Φ if for every R in Φ, there is a e such that $R(\vec{x}) \leftrightarrow Q(\vec{x},e)$ for all $\vec{x}$.

13.4. ARITHMETICAL ENUMERATION THEOREM. For every n and k, there is a $(k+1)$–ary Π_n^0 relation which enumerates the class of k–ary Π_n^0 relations; and similarly with Σ_n^0 for Π_n^0.

Proof. We suppose that $n = 2$; other values of n are similar. Suppose that R is Π_2^0; say $R(\vec{x}) \leftrightarrow \forall y \exists z P(\vec{x},y,z)$ where P is recursive. Let e be an index of χ_P. Then

$$R(\vec{x}) \leftrightarrow \forall y \exists z(\{e\}(\vec{x},y,z) \simeq 0)$$
$$\leftrightarrow \forall y \exists z \exists s(\{e\}_s(\vec{x},y,z) \simeq 0).$$

If we let $Q(\vec{x},e)$ be the right side of this equation, then Q is Π_2^0 by 8.4 and the table; so Q is the desired enumerating relation for Π_2^0. By the table, $\neg Q$ is the desired enumerating relation for Σ_2^0. □

Suppose that R is a binary relation which enumerates the class Φ of sets. We can use the diagonal method to define a set A which is not in Φ. Since we want $A(x)$ to be different from $R(x,e)$ when $x = e$, we set $A(e) \leftrightarrow \neg R(e,e)$. To put it another way, let D be the diagonal set defined by $D(e) \leftrightarrow R(e,e)$. Then if R enumerates Φ, $\neg D$ is not in Φ.

13.5. ARITHMETICAL HIERARCHY THEOREM. For each n, there is a Π_n^0 unary relation which is not Σ_n^0, hence not Π_k^0 or Σ_k^0 for any $k < n$. The same holds with Π_n^0 and Σ_n^0 interchanged.

Proof. We prove the first half; the second half is similar. Let P be a binary Π_n^0 relation which enumerates the class of unary Π_n^0 relations, and define $D(e) \leftrightarrow P(e,e)$. By 13.3, D is Π_n^0. By the above discussion, $\neg D$ is not Π_n^0; so by the table, D is not Σ_n^0. By 13.2, D is not Π_k^0 or Σ_k^0 for any $k < n$. □

The Arithmetical Hierarchy Theorem shows that there are no inclusions among the classes Π_n^0 and Σ_n^0 other than those given by 13.2.

The Arithmetical Enumeration Theorem is false for Δ_n^0 relations; for if it were true, we could use the proof of the Arithmetical Hierarchy Theorem to show that there is a Δ_n^0 relation which is not Δ_n^0.

Let Φ be a set of total functions. If Q is any concept defined in terms of recursive functions, we can obtain a definition of Q in Φ or relative to Φ by replacing recursive everywhere in the definition of Q by recursive in Φ. For example, R is arithmetical in Φ if it has a definition (1) where P is recursive in Φ; and R is Π_n^0 in Φ if it has such a definition in which the prefix is Π_n^0. We shall assume that this is done for all past and future definitions.

Now let us consider how the results of this section extend to the relativized case. Up to the Enumeration Theorem, everything extends without problems. The rest extends to finite Φ but not to arbitrary Φ. For example, if Φ is the set of all reals, then every unary relation is recursive in Φ and hence Π_n^0 and Σ_n^0 in Φ for all n. Thus the Hierarchy Theorem fails. Since the Hierarchy Theorem is a consequence of the Enumeration Theorem, the Enumeration Theorem also fails.

14. Recursively Enumerable Relations

A relation R is semicomputable if there is an algorithm which, when applied to the inputs $\vec{x}$, gives an output iff $R(\vec{x})$. If F is the function computed by the algorithm, then the algorithm applied to $\vec{x}$ gives an output iff $\vec{x}$ is in the domain of F. Hence R is semicomputable iff it is the domain of a computable function.

As an example, let A be the set of n such that $x^n + y^n = z^n$ holds for some positive integers x, y, and z. Then A is semicomputable; the algorithm with input n tests each triple (x,y,z) in turn to see if $x^n + y^n = z^n$. On the other