Review

• A Subtheory of Number Theory
• Representable Relations
• Church’s Thesis Revisited

Outline

• Representable Functions
• A Catalog of Representable Sets
• Gödel Numbers
• Fixed-Point Lemma
• Tarski Undefinability Theorem
• Gödel Incompleteness Theorem
• Second-Order Logic
• Skolemization

Source: Enderton, 2.6,3.3-3.5,4.1,4.2.

Numeralwise Determined Formulas

We just showed that a relation is representable in $\mathcal{C}_n A_E$ if we can find a formula that defines it in $\mathcal{N}$ and is numeralwise determined by $A_E$. The following theorem is helpful for establishing numeralwise determination.

Theorem

1. Any atomic formula is numeralwise determined by $A_E$.

2. If $\phi$ and $\psi$ are numeralwise determined by $A_E$, then so are $\neg \phi$ and $\phi \rightarrow \psi$.

3. If $\phi$ is numeralwise determined by $A_E$, then so are the following formulas (obtained by “bounded quantification”):

   \[ \forall x (x < y \rightarrow \phi) \text{ and } \exists x (x < y \land \phi). \]

Proof sketch The first two are easy. The $\forall$ case follows from the $\exists$ case. The $\exists$ case can be shown by using the fact that $\phi$ is numeralwise determined and considering two cases, one in which $\phi$ is true for some $x < y$ and one in which $\phi$ is false for every $x < y$. 
Representable Functions

Often functions are more convenient than relations. Suppose \( f : \mathbb{N}^m \rightarrow \mathbb{N} \) is an \( m \)-place function on the natural numbers.

A formula \( \phi \) in which at most \( v_1, \ldots, v_{m+1} \) occur free functionally represents \( f \) in the theory \( Cn A_E \) if for every \( a_1, \ldots, a_m \) in \( \mathbb{N} \),
\[ A_E \vdash \forall v_{m+1} \left[ \phi(S^{a_1}0, \ldots, S^{a_m}0, v_{m+1}) \iff v_{m+1} = Sf(a_1, \ldots, a_m)0 \right]. \]

Theorem
If \( \phi \) functionally represents \( f \) in \( Cn A_E \), then it also represents \( f \) as a relation in \( Cn A_E \).

Proof
Since \( \phi \) functionally represents \( f \), we have for any \( a_1, \ldots, a_{m+1} \),
\[ A_E \vdash \phi(S^{a_1}0, \ldots, S^{a_m}0, S^{a_{m+1}}0) \iff S^{a_{m+1}}0 = Sf(a_1, \ldots, a_m)0. \]

Since we can easily deduce whether the right half is true or false, it follows that we can deduce the left half or the negation of the left half.

\[ \square \]

Representable Functions

Consider the equation \( v_{m+1} = t \) where the free variables of \( t \) are among \( v_1, \ldots, v_m \). If \( f \) is a function which denotes the value of \( t \) at \( (a_1, \ldots, a_m) \), then clearly the equation defines \( f \) in \( \mathbb{N} \).

Also, since we showed that any atomic formula is numeralwise determined by \( A_E \), it follows that the equation represents \( f \) as a relation.

Finally, since
\[ \forall v_{m+1} \left[ v_{m+1} = t(S^{a_1}0, \ldots, S^{a_m}0) \iff v_{m+1} = Sf(a_1, \ldots, a_m)0 \right] \]
logically equivalent to \( t(S^{a_1}0, \ldots, S^{a_m}0) = Sf(a_1, \ldots, a_m)0 \), which is easily deducible in \( \mathbb{N} \), the equation functionally represents \( f \) as well.

As a result, we have the following:
1. The successor function is functionally represented by \( v_2 = Sv_1 \).
2. The \( m \)-place constant function with value \( b \) is functionally represented by \( v_{m+1} = Sb0 \).
3. Projection on variable \( i \) is functionally represented by \( v_{m+1} = v_i \).
4. Addition, multiplication, and exponentiation are represented by
   \[ v_3 = v_1 + v_2, \quad v_3 = v_1 \times v_2, \quad \text{and} \quad v_3 = v_1 Ev_2. \]

Representable Functions

The converse of the previous theorem is not true in general. A formula \( \phi \) may in fact represent a function \( f \) as a relation, but the stronger requirement of uniqueness may not be deducible.

However, by modifying \( \phi \), we can get a formula that works.

Theorem
If \( f \) is a function that is representable as a relation in \( Cn A_E \), then we can find a formula \( \phi \) that functionally represents \( f \) in \( Cn A_E \).

Proof Idea
If \( \theta \) represents \( f \) as a relation, then take \( \phi \) to be
\[ \theta(v_1, \ldots, v_m, v_{m+1}) \land \forall z (z < v_{m+1} \rightarrow \neg \theta(v_1, \ldots, v_m, z)). \]

\[ \square \]
Representable Functions

Theorem

If the $m + 1$-place function $g$ is representable and if for every $a_1, \ldots, a_m$ there is a $b$ such that $g(a_1, \ldots, a_m, b) = 0$, then we can find a formula that represents the $m$-place function $f$, where $f(a_1, \ldots, a_m) = \min b$ such that $g(a_1, \ldots, a_m, b) = 0$.

Using traditional notation, we can write this as $f(\vec{a}) = \min b [g(\vec{a}, b) = 0]$.

Proof

We have that $f(\vec{a}) = b$ iff $g(\vec{a}, b) = 0$ and for every $c < b$, $g(\vec{a}, c) \not= 0$. If a formula representing $f$ is simply:

$\psi(v_1, \ldots, v_m, v_{m+1}, 0) \land \forall y (y < v_{m+1} \rightarrow \neg \psi(v_1, \ldots, v_m, y, 0))$.  

A Catalog of Representable Sets

8. For each $m$, the function whose value at $a_0, \ldots, a_m$ is $\langle a_0, \ldots, a_m \rangle$ is representable.

9. There is a representable function (whose value at $\langle a, b \rangle$ is written $(a)_b$) such that for $b \leq m$, $(a_0, \ldots, a_m)_b = a_b$.

10. Say that $b$ is a sequence number iff for some $m \geq -1$ and some $a_0, \ldots, a_m, b = \langle a_0, \ldots, a_m \rangle$. The set of sequence numbers is representable.

11. There is a representable function $lh$ such that $lh \langle a_0, \ldots, a_m \rangle = m + 1$.

12. There is a representable function (whose value at $\langle a, b \rangle$ is called the restriction of $a$ to $b$, written $a^b$) such that for any $b \leq m + 1$,

$\langle a_0, \ldots, a_m \rangle^b = \langle a_0, \ldots, a_{b-1} \rangle$.

13. (Primitive recursion) With a $k + 1$-place function $f$ we associate another function $\tilde{f}$ such that $\tilde{f}(a, b_1, \ldots, b_k)$ encodes the values of $f(j, b_1, \ldots, b_k)$ for all $j < a$. Specifically, let $\tilde{f}(a, b) = \langle f(0, b), \ldots, f(a - 1, b) \rangle$.

A Catalog of Representable Sets

1. A relation $R$ is representable iff its characteristic function $K_R$ is

$K_R(\vec{a}) = 1$ if $\vec{a} \in R$, 0 otherwise.

2. If $R$ is a representable binary relation and $g, f$ are representable functions, then $\langle \vec{a} \rangle \langle f(\vec{a}), g(\vec{a}) \rangle \in R$ is representable.

3. If $R$ is a representable binary relation, then so is $P = \{ \langle a, b \rangle \mid \text{for some } c \leq b, \langle a, c \rangle \in R \}$.

4. The divisibility relation $\{ \langle a, b \rangle \mid a \text{ divides } b \text{ in } \mathbb{N} \}$ is representable.

5. The set of primes is representable.

6. The set of pairs of adjacent primes is representable.

7. The function whose value at $a$ is $p_a$, the $(a + 1)^{th}$ prime, is representable.

We can use this last fact to encode a finite sequence of numbers into a single number as follows:

$\langle a_0, \ldots, a_m \rangle = p_{a_0+1}^a \cdots p_{a_m+1}^a$.

A Catalog of Representable Sets

Theorem

Let $g$ and $h$ be representable functions, and assume that $f(0, b) = g(b)$,

$f(a + 1, b) = h(f(a, b), a, b)$.

Then $f$ is representable.

14. For a representable function $F$, the function whose value at $a, \vec{b}$ is

$\prod_{i < a} F(i, \vec{b})$

is representable. Similarly for $\sum_{i < a} F(i, \vec{b})$

15. Define the concatenation of $a$ and $b$, $a \cdot b$, as

$a \cdot b = \prod_{i < lh(b)} p_{i+lh(a)}^{b_i+1}$.

This is a representable function of $a$ and $b$, and

$\langle a_1, \ldots, a_m \rangle \cdot \langle b_1, \ldots, b_n \rangle = \langle a_1, \ldots, a_m, b_1, \ldots, b_n \rangle$.

16. Let $s_{i < a} f(i) = f(0) * f(1) * \cdots * f(a - 1)$. For a representable function $F'$, the function whose value at $a, \vec{b}$ is $s_{i < a} F'(i, \vec{b})$ is representable.
Arithmetization of Syntax

So far, we introduced the notion of representability and showed that many functions and relations are representable in $\mathcal{Cn} A_E$.

By encoding the syntax of first-order logic using natural numbers, we can encode facts about the terms and formulas of logic as relations in $\mathcal{N}$.

We can then use our results about representability to show that there are some surprising limits to what can be represented in $\mathcal{Cn} A_E$.

We begin by assigning a number to each symbol in our formal language.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Encoded symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall$</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>$\exists$</td>
<td>4</td>
</tr>
<tr>
<td>$\neg$</td>
<td>5</td>
</tr>
<tr>
<td>$&lt;$</td>
<td>6</td>
</tr>
<tr>
<td>$\rightarrow$</td>
<td>7</td>
</tr>
<tr>
<td>$+$</td>
<td>8</td>
</tr>
<tr>
<td>$\times$</td>
<td>10</td>
</tr>
<tr>
<td>$E$</td>
<td>12</td>
</tr>
<tr>
<td>$v_1$</td>
<td>11</td>
</tr>
<tr>
<td>$v_2$</td>
<td>13</td>
</tr>
<tr>
<td>$v_k$</td>
<td>$9 + 2k$</td>
</tr>
</tbody>
</table>

Note that this encoding could be modified to accommodate any countable signature (the symbols in the signature are assigned to the even numbers).

Let $h$ be the function which maps each symbol to its encoding.

Gödel Numbers

For an expression $\epsilon = s_0 \cdots s_n$ of the language, we define its Gödel number, $\#(\epsilon)$ by

$$\#(s_0, \ldots, s_n) = \langle h(s_0), \ldots, h(s_n) \rangle.$$  

Example

$$\#(\exists v_3 v_3 = 0) = \#(\neg \forall v_3 (\neg = v_3 0)) = \langle 1, 5, 0, 15, 1, 5, 9, 15, 2, 3, 3 \rangle = 2^2 \cdot 3^6 \cdot 5^1 \cdot 7^{16} \cdot 11^2 \cdot 13^6 \cdot 17^{10} \cdot 19^{16} \cdot 23^3 \cdot 29^4 \cdot 31^4.$$  

A set of expressions corresponds to a set of Gödel numbers:

$$\#\Phi = \{ \#(\epsilon) | \epsilon \in \Phi \}.$$  

Gödel Numbers

We now show that various relations and functions involving Gödel numbers are representable.

1. The set of Gödel numbers of variables is representable.

Proof

A formula which defines this set is:

$$\exists b (b < a \land a = \langle 11 + 2 \cdot b \rangle).$$

This formula makes use of bounded quantification, the equality relation, arithmetic constants, sequence numbers, and function composition, all of which we showed to be representable earlier.
Gödel Numbers

2. The set of Gödel numbers of terms is representable.

**Proof idea**

Let \( f \) be the corresponding characteristic function (i.e. the function whose value is 1 if its input is the Gödel number of a term and 0 otherwise).

Then,

\[
f(a) = \begin{cases} 
1 & \text{if } a \text{ is the Gödel number of a variable,} \\
1 & \text{if } \exists i < a \text{ s.t. } \exists k < a \\
\forall j < lh(i) \left( f\left( (i)_j \right) = 1 \right) \text{ and } \\
k \text{ is the value of } h \text{ at some } (lh(i))-\text{place function symbol and} \\
a = \langle k \rangle \ast \ast \forall j < lh(i) (i)_j \\
0 & \text{otherwise.}
\end{cases}
\]

3. The set of Gödel numbers of atomic formulas is representable.

4. The set of Gödel numbers of wffs is representable.

5. There is a representable function \( Sb \) such that for a term or formula \( \alpha \), variable \( x \), and term \( t \),

\[
Sb(\#\alpha, \#x, \#t) = \#(\alpha^x_t).
\]

6. The function whose value at \( n \) is \(#(S^n0)\) is representable.

7. There is a representable relation \( Fr \) such that for a term or formula \( \alpha \) and a variable \( x \), \( \langle \#\alpha, \#x \rangle \in Fr \) iff \( x \) occurs free in \( \alpha \).

8. The set of Gödel numbers of sentences is representable.

9. There is a representable relation \( Sbl \) such that for a formula \( \alpha \), variable \( x \), and term \( t \), \( \langle \#\alpha, \#x, \#t \rangle \in Sbl \) iff \( t \) is substitutable for \( x \) in \( \alpha \).

10. The relation \( Gen \), where \( \langle a, b \rangle \in Gen \) iff \( a \) is the Gödel number of a formula and \( b \) is the Gödel number of a generalization of that formula, is representable.

Church’s Thesis (Again)

Recall that Church’s thesis states a relation is decidable iff the relation is recursive.

We have just shown that every recursive relation is representable in the theory \( Cn A_E \). This means that \( Cn A_E \) must be powerful enough to represent any decision procedure.

Using techniques like those we have just seen (representability using Gödel numbers), it can be shown that any model of computation can be mirrored using \( Cn A_E \).

In what follows, the terms recursive and decidable are used interchangeably.

11. The set of Gödel numbers of tautologies is representable.

17. The set of Gödel numbers of logical axioms is representable.

Let \( G(\langle \alpha_0, \ldots, \alpha_n \rangle) = \langle \#\alpha_0, \ldots, \#\alpha_n \rangle \).

18. For a finite set \( A \) of formulas,

\[
\{ G(D) | D \text{ is a deduction from } A \}
\]

is representable.

19. Any recursive relation is representable in \( Cn A_E \).

21. If \( \# A \) is recursive and \( Cn A \) is a complete theory, then \( \# Cn A \) is recursive.
**Fixed-Point Lemma**

Suppose $\beta$ is a formula which defines (in $N$) some subset $A$ of $N$. How do we interpret the following formulas:

- $\beta(S^n0)$
- $\beta(S^{#\sigma})$
- $\sigma \iff \beta(S^{#\sigma})$

The fixed-point lemma gives us the surprising result that for any such formula $\beta$, we can always find a sentence $\sigma$ such that the last formula not only is true in $N$, but is derivable from $A_E$.

**Fixed-Point Lemma**

We have $h(#\alpha) = #(\alpha(S^{#\alpha}0))$, $\gamma$ defines $\{#\alpha \mid h(#\alpha) \text{ is in the set defined by } \beta\}$, and $\gamma = \gamma(S^{#\gamma}0)$.

$\sigma$ holds in $N$ iff $\gamma(S^{#\gamma}0)$ holds in $N$
- $\gamma$ holds in $N$
- $\#\gamma$ in the set defined by $\beta$
- $h(\#\gamma)$ is in the set defined by $\beta$
- $\#\gamma(S^{#\gamma}0)$ is in the set defined by $\beta$
- $\#\sigma$ is in the set defined by $\beta$
- $\beta(S^{#\sigma}0)$ is true in $N$.

This shows that $\sigma$ holds in $N$ iff $\beta(S^{#\sigma}0)$ holds in $N$, i.e.

$\models_N \sigma \iff \beta(S^{#\sigma}0)$.

However, we need to show that this fact is deducible from $A_E$:

$A_E \vdash \sigma \iff \beta(S^{#\sigma}0)$.

The proof that this is derivable follows from the fact that $\theta$ functionally represents $h$ and the definition of $\sigma$.

**Tarski Undefinability Theorem**

**Theorem**

The set $\# Th N$ is not definable in $N$.

**Proof**

Suppose that $\beta$ were a formula which defined the set $\# Th N$. Applying the fixed-point lemma to $\neg \beta$, we get a sentence $\sigma$ such that

$\models_N \sigma \iff \neg \beta(S^{#\sigma}0)$.

and thus,

$\models_N \sigma \iff \neg \beta(S^{#\sigma}0)$.

So, if $\sigma \in Th N (\models_N \sigma)$, then its Gödel number is not in the set $\beta$ defines, meaning that $\beta$ cannot define $\# Th N$.

On the other hand, if $\sigma \notin Th N (\not\models_N \sigma)$, then its Gödel number is in the set $\beta$ defines, meaning that $\beta$ cannot define $\# Th N$.

\[ \square \]
**Tarski Undefinability Theorem**

**Corollary**
The set $\# \text{Th}_N$ is not recursive.

**Proof**
Any recursive set is definable in $N$.

In other words, $\# \text{Th}_N$ (and thus $\text{Th}_N$) is not decidable.

*What happened?*
Essentially, what we have shown is that the language and structure of $N$ are powerful enough that for any decidable set $D$, we can always find a sentence whose meaning in $N$ is “I am in (or not in) $D$”.

Thus, if $\text{Th}_N$ were decidable, there would be a sentence that says “I am not in $\text{Th}_N$”. If the sentence is in $\text{Th}_N$, then it’s not, and if it’s not, then it is. Contradiction.

---

**Gödel Incompleteness Theorem**

**Theorem**
If $A \subseteq \text{Th}_N$ and $\# A$ is recursive, then $\text{Cn} A$ is not a complete theory.

**Proof**
Suppose $\text{Cn} A$ is complete. Since $A \subseteq \text{Th}_N$, it follows that $\text{Cn} A \subseteq \text{Th}_N$. And since $\text{Cn} A$ is complete and $\text{Th}_N$ is satisfiable, we must have $\text{Cn} A = \text{Th}_N$. But then $\# Cn A = \# \text{Th}_N$ is recursive (by item 21). But every recursive set is definable in $N$, and we just showed that $\# \text{Th}_N$ is not definable in $N$.

Hence, $\text{Th}_N$ cannot be axiomatized.

---

**Second-Order Logic**

Consider the first-order formula $\exists x (Px \rightarrow \forall x Px)$.

This formula is valid, meaning it is true in every model with every variable assignment.

In particular, it is true regardless of how we interpret $P$.

Because of this, we may be tempted to write $\forall P \exists x (Px \rightarrow \forall x Px)$.

Second-order logic allows us to give in to this temptation.

---

**Second-Order Logic: Syntax**

The logical symbols of second-order logic include all those used in first-order logic as well as the following:

- **Predicate variables**: For each positive integer $n$, we have the $n$-place predicate variables $X^n_1, X^n_2, \ldots$.

- **Function variables**: For each positive integer $n$, we have the $n$-place function variables $F^n_1, F^n_2, \ldots$.

The first-order variables $v_1, v_2, \ldots$ will be called *individual* variables to avoid confusion.
Second-Order Logic: Syntax

A signature $\Sigma$ consists of constant, function, and predicate symbols exactly as in first-order logic.

Terms

The terms over a signature $\Sigma$ are defined inductively as follows:

- $U$ = all expressions over the logical symbols and symbols in $\Sigma$.
- $B$ = individual variables and constants from $\Sigma$.
- $F$ = the set of term-building operations obtained from the function symbols in $\Sigma$ and the logical function variables.

To be more precise about $F$, let $f$ be a function symbol (either from $\Sigma$ or a function variable $F^n_i$ of arity $n$). The term-building operation $F_f$ is defined as follows:

$$F_f(\alpha_1, \ldots, \alpha_n) = f\alpha_1, \ldots, \alpha_n$$

The set of terms is the set of expressions generated from the constant symbols and individual variables by the $F_f$ operations.

Atomic Formulas

Atomic formulas are, as before, a predicate symbol applied to 0 or more terms. The predicate symbol can be equality, a predicate symbol from $\Sigma$, or a predicate variable $X^n_i$.

Formulas

As in first-order logic, we have the following formula-building operations:

- $E_\neg(\alpha) = (\neg \alpha)$
- $E_\rightarrow(\alpha, \beta) = (\alpha \rightarrow \beta)$
- $Q_i(\alpha) = \forall v_i \alpha$

We add the following additional formula-building operations:

- $Q^n_{i,n}(\alpha) = \forall X^n_i \alpha$
- $Q^n_{i,n}(\alpha) = \forall F^n_i \alpha$

The set of well-formed formulas (wffs or just formulas) is the set of expressions generated from the atomic formulas by the formula building operations. A free variable is defined as before, and a sentence is a formula with no free variables.

Second-Order Logic: Semantics

A model $M$ serves the same function in second-order logic as it does in first-order logic: it provides a domain and an interpretation for each symbol in the signature $\Sigma$.

However, the role of the variable assignment function $s$ must be extended.

Let $V$ be the set of all variables: individual, predicate, or function.

We define a variable assignment $s$ to be a function on $V$ which assigns to each variable in $V$ a suitable object:

- $s(v_i)$ is a member of $\text{dom}(M)$.
- $s(X^n_i)$ is an $n$-ary relation on $\text{dom}(M)$, and
- $s(F^n_i)$ is an $n$-ary function on $\text{dom}(M)$.

Given a wff $\phi$, we say that $M$ satisfies $\phi$ with $s$ and write $\models_M \phi[s]$ if $\phi$ is true in the model $M$ with variable assignment $s$.

We now define this formally.

Terms

We first define the extension $\bar{s}$, a function from the set of all terms into the domain of $M$:

1. For each variable $x$, $\bar{s}(x) = s(x)$.
2. For each constant symbol $c$, $\bar{s}(c) = c^M$.
3. If $t_1, \ldots, t_n$ are terms and $f$ is an $n$-ary function symbol from $\Sigma$, then $\bar{s}(f(t_1, \ldots, t_n)) = f^M(\bar{s}(t_1), \ldots, \bar{s}(t_n))$.
4. If $t_1, \ldots, t_n$ are terms, then $\bar{s}(F^n_{i_1}t_1, \ldots, t_n) = s(F^n_{i_1})(\bar{s}(t_1), \ldots, \bar{s}(t_n))$.

Atomic Formulas

1. $\models_M t_1 = t_2[s]$ if $\bar{s}(t_1) = \bar{s}(t_2)$.
2. For an $n$-ary predicate symbol $P$ from $\Sigma$, $\models_M Pt_1, \ldots, t_n[s]$ if $(\bar{s}(t_1), \ldots, \bar{s}(t_n)) \in P^M$.
3. $\models_M X^n_{i_1}t_1, \ldots, t_n[s]$ if $(\bar{s}(t_1), \ldots, \bar{s}(t_n)) \in s(X^n_{i_1})$. 
Second-Order Logic: Semantics

Formulas

1. $\models_M (\neg \phi)[s] \iff \not\models_M \phi[s]$.
2. $\models_M (\phi \to \psi)[s] \iff \not\models_M \phi[s] \text{ or } \models_M \psi[s]$.
3. $\models_M \forall x \phi[s] \iff \models_M \phi[s(x)[d]]$ for every $d \in \text{dom}(M)$.
4. $\models_M \forall X^\alpha \phi[s] \iff \models_M \phi[s(X^\alpha[R])]$ for every $\alpha$-ary relation $R$ on $\text{dom}(M)$.
5. $\models_M \forall F^\alpha \phi[s] \iff \models_M \phi[s(F^\alpha[f])]$ for every $\alpha$-place function $f$ on $\text{dom}(M)$.

where, as before, $s(x)[d]$ signifies the function which is the same as $s$ everywhere except at $x$ where its value is $d$.

For notational convenience, we will omit the subscripts and superscripts from the $X$ and $F$ variables when they are unnecessary.

Second-Order Logic

The previous example shows that the compactness theorem fails for second-order logic.

Theorem

There is an unsatisfiable set of second-order sentences, every finite subset of which is satisfiable.

Proof

Let $\lambda_\infty$ be a second-order sentence which axiomatizes the infinite models. Consider the set $\{\lambda_\infty, \lambda_2, \lambda_3, \ldots\}$. Every finite subset of this set is satisfiable in some finite model. However, the entire set is not satisfiable in any finite model or infinite model.

The Löwenheim-Skolem theorem also fails for second-order logic. For example, there is a sentence in the second-order language of equality (empty signature) that is true in a model iff its cardinality is $2^{\aleph_0}$.

Second-Order Logic

Recall that for each $n \geq 2$, we have a first-order sentence $\lambda_n$ which translates, “there are at least $n$ distinct objects”:

$\lambda_2 = \exists x \exists y x \neq y,$
$\lambda_3 = \exists x \exists y \exists z (x \neq y \wedge x \neq z \wedge y \neq z),$

... Earlier, we showed that the class of all infinite models is axiomatized by $\lambda_2, \lambda_3, \ldots$, but is not axiomatized by any single first-order sentence.

However, there is a second-order sentence which axiomatizes the infinite models. A set is infinite iff there is an ordering on it having no last element:

$\exists X \forall u \forall v \forall w (Xuv \rightarrow Xvw) \wedge \forall u \neg Xuu \wedge \forall u \exists v Xuv$.

Alternatively, a set is infinite if there is a one-to-one function that is not onto:

$\exists F \forall x \forall y (Fx = Fy \rightarrow x = y) \wedge \exists z \forall x Fx \neq z]$. 

Prenex Normal Form

A prenex formula is one of the form $Q_1x_1 \cdots Q_nx_n\alpha$, where each $Q_i$ is a quantifier and $\alpha$ is quantifier-free.

Theorem

For any formula we can find a logically equivalent prenex formula.

Proof

We use the following quantifier identities which are easy to prove.

• $\neg \forall x \alpha \leftrightarrow \exists x \neg \alpha$
• $\neg \exists x \alpha \leftrightarrow \forall x \neg \alpha$
• $(\alpha \rightarrow \forall x \beta) \leftrightarrow \forall x (\alpha \rightarrow \beta)$, for $x$ not free in $\alpha$.
• $(\alpha \rightarrow \exists x \beta) \leftrightarrow \exists x (\alpha \rightarrow \beta)$, for $x$ not free in $\alpha$.
• $(\forall x \alpha \rightarrow \beta) \leftrightarrow \exists x (\alpha \rightarrow \beta)$, for $x$ not free in $\beta$.
• $(\exists x \alpha \rightarrow \beta) \leftrightarrow \forall x (\alpha \rightarrow \beta)$, for $x$ not free in $\beta$.

By using these identities together with renaming of bound variables, it is easy to manipulate any formula to a logically equivalent prenex formula.
Skolemization

Skolem Normal Form Theorem

For any first-order formula, there is a logically equivalent second-order formula consisting of:
1. First a string (possibly empty) of existential individual and function quantifiers, followed by
2. A string (possibly empty) of universal individual quantifiers, followed by
3. A quantifier-free formula

Proof

First, given any first order formula, we can find an equivalent formula \( \phi \) in prenex normal form. If \( \phi \) is in the form described above, we are done.

Otherwise, \( \phi \) must contain a sub-formula of the form

\[
\forall v_1 \ldots \forall v_n \exists z \psi(z).
\]

This formula is equivalent to

\[
\exists F \forall v_1 \ldots \forall v_n \psi(Fv_1 \ldots v_n).
\]

By repeatedly applying this rule (from the outside inwards), every nested existential quantifier can be eliminated. The result is a formula of the form described above.

Example

Find an equisatisfiable universal formula:

\[
\exists y_1 \forall x_1 \left[ (\forall y_2 \exists x_2 \ x_1 + y_2 < x_2) \rightarrow (\forall x_3 \exists y_3 \ x_1 + x_3 > y_3) \right]
\]

iff

\[
\exists y_1 \forall x_1 \exists y_2 \forall x_2 \exists y_3 \left[ (x_1 + y_2 < x_2) \rightarrow (y_1 + x_3 > y_3) \right]
\]

iff

\[
\exists y_1 \exists F_1 \forall x_1 \forall x_2 \exists y_3 \left[ (x_1 + F_1 x_2 < x_2) \rightarrow (y_1 + x_3 > y_3) \right]
\]

iff

\[
\exists y_1 \exists F_1 \exists F_2 \forall x_1 \forall x_2 \forall x_3 \left[ (x_1 + F_1 x_2 < x_2) \rightarrow (y_1 + x_3 > F_2 x_1 x_2 x_3) \right]
\]

which is satisfiable iff

\[
\forall x_1 \forall x_2 \forall x_3 \left[ (x_1 + F_1 x_2 < x_2) \rightarrow (y_1 + x_3 > F_2 x_1 x_2 x_3) \right]
\]

is satisfiable.

Skolemization

Recall that a universal formula is a first-order prenex formula all of whose quantifiers are universal. Similarly, an existential formula is a first-order prenex formula all of whose quantifiers are existential.

Corollary

For any first-order \( \phi \), there is a universal formula \( \theta \) in an expanded language containing function symbols such that \( \phi \) is satisfiable iff \( \theta \) is satisfiable.

Corollary

For any first-order \( \phi \), there is an existential formula \( \theta \) in an expanded language containing function symbols such that \( \phi \) is valid iff \( \theta \) is valid.

This also gives us undecidability results for universal and existential formulas.

Corollary

1. The set of Gödel numbers of satisfiable universal first-order sentences is not recursive.
2. The set of Gödel numbers of valid existential first-order sentences is not recursive.

Proof

(2) Given any sentence \( \sigma \), we can effectively find an existential sentence that is valid iff \( \sigma \) is valid. Thus, if we could solve the validity problem for existential sentences, we could solve it for arbitrary sentences, contradicting Church’s Theorem.