Compact course notes

Pure Math 432/632, Winter 2012

First order logic and Computability

Professor: B. Csima
transcribed by: J. Lazovskis
University of Waterloo
April 9, 2012

Contents

Note: Not all theorems are proved that are presented.

1 First-order logic syntax

For the purposes of this course, we use naive set theory and assume the Axiom of Choice.

1.1 Definitions

Definition 1.1.1. An alphabet A is a non-empty set of symbols.

- A string or word a over an alphabet A is a finite sequence of symbols from A.
- The length of a word a is the total number of symbols in a, counting repetitions.

Remark 1.1.2. We use the following notation for readability:
- A^* denotes the set of all possible words over A
- $\square$ denotes the empty word, i.e. the word of no symbols

Definition 1.1.3. The alphabet of a first-order language A contains the following symbols:

a.	$v_0, v_1, v_2, \ldots$	variables
b.	$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$	not, and, or, implies, if and only if
c.	$\forall, \exists$	for all, there exists
d.	$\equiv$	equality
e.	$($, $)$	parentheses

Accompanying A is a (possibly empty) set S being the union of the following sets:

f. For every $n \in \mathbb{N}$, a set of n-ary relation symbols
g. For every $n \in \mathbb{N}$, a set of n-ary function symbols
h. A finite set of constant symbols

Therefore the symbol set S determines a first-order language, and $A_S = A \cup S$ is its alphabet

Example 1.1.4. The symbol set of groups is $S_{gr} := \{0, e\}$.

Definition 1.1.5. The arity of relations and functions refers to the number of symbols they state a relation about or act on, and is denoted in the superscript, such as R^n or f^n. Irrespective of the arity, a function always outputs a single symbol.

Definition 1.1.6. The following words in A_S^* are termed S-terms:
T1. every variable in A
T2. every constant symbol in S
T3. $ft_1 \ldots t_n$ for f an n-ary function and $t_1, \ldots, t_n$ all S-terms

The set of all S-terms is denoted by T^S.

Definition 1.1.7. The following words in A_S^* are termed S-formulae:
F1. $t_1 \equiv t_2$ for t_1, t_2 S-terms
F2. $Rt_1 \ldots t_n$ for R an n-ary relation symbol and $t_1, \ldots, t_n$ S-terms
F3. $\neg\varphi$ for φ an S-formula
F4. $(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \rightarrow \psi), (\varphi \leftrightarrow \psi)$ for φ, ψ S-formulae
F5. $\forall x\varphi$ and $\exists x\varphi$ for φ an S-formula and x a variable

The set of all S-formulae of length n is denoted by L^S.

Remark 1.1.8. If S is at most countable, then T^S and L^S are at most countable also.

Definition 1.1.9. The function var acts on an S-term and outputs the set of variables occuring in this term. Thus, if x is a variable, c is a constant, f is an n-ary relation and $t_1, \ldots, t_n$ are S-terms, then

$$\begin{aligned} \operatorname{var}(x) &:= \{x\} \\ \operatorname{var}(c) &:= \varnothing \\ \operatorname{var}(ft_1 \ldots t_n) &:= \operatorname{var}(t_1) \cup \cdots \cup \operatorname{var}(t_n) \end{aligned}$$

Definition 1.1.10. The function SF assigns to each formula the set of its subformulae, and is defined by:

$$\begin{aligned} SF(t_1 = t_2) &:= \{t_1 = t_2\} \\ SF(Rt_1 \dots t_n) &:= \{Rt_1 \dots t_n\} \\ SF(\neg\varphi) &:= \{\neg\varphi\} \cup SF(\varphi) \\ SF((\varphi * \psi)) &:= \{(\varphi * \psi)\} \cup SF(\varphi) \cup SF(\psi) \\ SF(Qx\varphi) &:= \{Qx\varphi\} \cup SF(\varphi) \end{aligned}$$

where $* \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$ and $Q \in \{\forall, \exists\}$.

Definition 1.1.11. Given an S-formula φ, each of the variables in $\text{var}(\varphi)$ are either bound or free. The function free, that produces the set of free variables of an S-formula, is defined as follows:

$$\begin{aligned} \text{free}(t_1 = t_2) &:= \text{var}(t_1) \cup \text{var}(t_2) \\ \text{free}(Rt_1 \dots t_n) &:= \text{var}(t_1) \cup \dots \cup \text{var}(t_n) \\ \text{free}(\neg\varphi) &:= \text{free}(\varphi) \\ \text{free}((\varphi * \psi)) &:= \text{free}(\varphi) \cup \text{free}(\psi) \\ \text{free}(Qx\varphi) &:= \text{free}(\varphi) \backslash \{x\} \end{aligned}$$

Example 1.1.12. In $\forall xRxyz$, the variable x is bound and y, z are free

1.2 Meaning

Definition 1.2.1. Let φ be an S-formula. If $\text{free}(\varphi) = \varnothing$, then φ is termed a sentence.

Definition 1.2.2. Define L_0^S to be the set of S-sentences. In general,

$$L_n^S := \{\varphi \mid \varphi \text{ is an } S\text{-formula and } |\text{free}(\varphi)| = n\}$$

Definition 1.2.3. An S-structure is a pair $\mathfrak{A} = (A, \mathfrak{a})$ of a set A and an assignment $\mathfrak{a}$ on S such that

1. A is non-empty
2. $\mathfrak{a}$ is defined by the following rules:
   - **i.** $\mathfrak{a}(R) = R^{\mathfrak{A}} = R^A$ is an n-ary relation on A
   - **ii.** $\mathfrak{a}(f) = f^{\mathfrak{A}} = f^A$ is an n-ary function on A
   - **iii.** $\mathfrak{a}(c) = c^{\mathfrak{A}} = c^A$ is an element of A

Remark 1.2.4. If β is an assignment in an S-structure $\mathfrak{A}$ with $a \in A$ and x is a variable, then define the assignment

$$\beta\frac{a}{x}(y) := \begin{cases} \beta(y) & \text{if } y \neq x \\ a & \text{if } y = x \end{cases}$$

Definition 1.2.5. An S-interpretation is a pair $\mathfrak{I} = (\mathfrak{A}, \beta)$ of an S-structure $\mathfrak{A}$ and an assignment β in $\mathfrak{A}$, that acts on S-terms, such that

1. $\mathfrak{I}\frac{a}{x} = (\mathfrak{A}, \beta\frac{a}{x})$
2. the action of $\mathfrak{I}$ is defined by the following rules:
   - **i.** $\mathfrak{I}(x) = \beta(x)$ for x a variable
   - **ii.** $\mathfrak{I}(c) = c^{\mathfrak{A}}$ for c a constant
   - **iii.** $\mathfrak{I}(ft_1 \dots t_n) = f^{\mathfrak{A}}(\mathfrak{I}(t_1), \dots, \mathfrak{I}(t_n))$ for $t_1, \dots, t_n$ S-terms

Definition 1.2.6. Given a formula φ, an interpretation $\mathfrak{I}$ is termed a model of φ (written $\mathfrak{I} \vDash \varphi$, pronounced "$\mathfrak{I}$ satisfies φ") when the following conditions are satisfied:

$\mathfrak{I} \models t_1 \equiv t_2$	if and only if	$\mathfrak{I}(t_1) = \mathfrak{I}(t_2)$
$\mathfrak{I} \models Rt_1 \dots t_n$	if and only if	$R^{\mathfrak{A}}(\mathfrak{I}(t_1), \dots, \mathfrak{I}(t_n))$ holds
$\mathfrak{I} \models \neg\varphi$	if and only if	not $(\mathfrak{I} \models \varphi)$
$\mathfrak{I} \models (\varphi * \psi)$	if and only if	$(\mathfrak{I} \models \varphi) * (\mathfrak{I} \models \psi)$
$* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$		$* \in$ {and, or, implies, if and only if}
$\mathfrak{I} \models Qx\varphi$	if and only if	$Qa \in A\ \mathfrak{I}\frac{a}{x} \models \varphi$
$Q \in \{\forall, \exists\}$		$Q \in$ {for all, there exists}

Definition 1.2.7. Let Φ be a possibly infinite set of S-formulae. Then for an S-interpretation $\mathfrak{I}$, we say $\mathfrak{I} \models \Phi$ iff $\mathfrak{I} \models \varphi$ for all $\varphi \in \Phi$.

Definition 1.2.8. Let Φ be a set of formulae and φ a formula. Then we write $\Phi \models \varphi$ (pronounced "φ is a consequence of Φ") iff for every interpretation $\mathfrak{I}$ with $\mathfrak{I} \models \Phi$, the expression $\mathfrak{I} \models \varphi$ holds.

1.3 Validity

Definition 1.3.1. A formula φ is termed valid iff $\varnothing \models \varphi$, that is, when for all interpretations $\mathfrak{I}$, $\mathfrak{I} \models \varphi$.

Definition 1.3.2. A formula φ is termed satisfiable (written Sat(φ)) if there exists an interpretation which is a model of φ. A set of formulas Φ is satsifiable if there exists an interpretation which is a model for every φ in Φ.

Lemma 1.3.3. For all Φ and all φ, $\Phi \models \varphi$ iff not $\text{Sat}(\Phi \cup \{\neg\varphi\})$.

Definition 1.3.4. Two formulae φ, ψ are termed equivalent (written $\varphi \Dashv\models \psi$) iff $\varphi \models \psi$ and $\psi \models \varphi$. Therefore we may eliminate some symbols:

$$\begin{aligned}
\varphi \wedge \psi &\Dashv\models \neg(\neg\varphi \vee \neg\psi) \\
\varphi \rightarrow \psi &\Dashv\models \neg\varphi \vee \psi \\
\varphi \leftrightarrow \psi &\Dashv\models \neg(\varphi \vee \psi) \vee \neg(\neg\varphi \vee \neg\psi) \\
\forall x\varphi &\Dashv\models \neg\exists x\neg\varphi
\end{aligned}$$

So the connectives $\wedge, \rightarrow, \leftrightarrow$ and the quantifier $\forall$ are superfluous. We no longer consider them in our language, but we continue to employ them as shorthand.

Lemma 1.3.5. [Coincidence lemma]
Let $\mathfrak{I}_1 = (\mathfrak{A}_1, \beta_1)$ be an S_1-interpretation and $\mathfrak{I}_2 = (\mathfrak{A}_2, \beta_2)$ be an S_2-interpretation, with $S = S_1 \cap S_2$ and t an S-term and φ an S-formula.

1. If $\mathfrak{I}_1$ and $\mathfrak{I}_2$ agree on the S-symbols in t and var(t), then $\mathfrak{I}_1(t) \equiv \mathfrak{I}_2(t)$

2. If $\mathfrak{I}_1$ and $\mathfrak{I}_2$ agree on the S-symbols in φ and free(φ), then $(\mathfrak{I}_1 \models \varphi) \Dashv\models (\mathfrak{I}_2 \models \varphi)$.

Proof: **1.** will be done by induction.

$$\begin{aligned}
\mathfrak{I}_1(c) = c^{\mathfrak{A}_1} &= c^{\mathfrak{A}_2} = \mathfrak{I}_2(c) \\
\mathfrak{I}_1(x) = \beta_1(x) &= \beta_2(x) = \mathfrak{I}_2(x) \\
\mathfrak{I}_1(ft_1 \dots t_n) = f^{\mathfrak{A}_1}(\mathfrak{I}_1(t_1) \dots \mathfrak{I}_1(t_n)) &= f^{\mathfrak{A}_2}(\mathfrak{I}_2(t_1) \dots \mathfrak{I}_2(t_n)) \\
&= f^{\mathfrak{A}_2}(\mathfrak{I}_2(t_1) \dots \mathfrak{I}_2(t_n)) \\
&= \mathfrak{I}_2(ft_1, \dots t_n)
\end{aligned}$$

2. will also be done by induction.

$$\begin{aligned}
\mathfrak{I}_1 \models t_1 \equiv t_2 &\text{ iff } \mathfrak{I}_1(t_1) \equiv \mathfrak{I}_1(t_2) \\
&\text{ iff } \mathfrak{I}_2(t_1) \equiv \mathfrak{I}_2(t_2) \\
&\text{ iff } \mathfrak{I}_2 \models t_1 \equiv t_2
\end{aligned}$$

Now suppose $\varphi = \exists x\psi$.
Then $\mathfrak{I}_1 \models \exists x\psi$ iff there exists $a \in A$ such that $\mathfrak{I}_1\frac{a}{x} \models \psi$.
Note that $\text{free}(\psi) \subset \text{free}(\varphi) \cup \{x\}$.
Since $\mathfrak{I}_1, \mathfrak{I}_2$ agree on $\text{free}(\varphi)$, we have that $\mathfrak{I}_1\frac{a}{x}$ and $\mathfrak{I}_2\frac{a}{x}$ agree on $\text{free}(\psi)$.
Also, $\mathfrak{I}_1\frac{a}{x}$ and $\mathfrak{I}_2\frac{a}{x}$ agree on $\{x\}$.
Hence they both agree on $\text{free}(\varphi)$.
Thus $\mathfrak{I}_1 \models \exists x\psi$ iff there exists $a \in A$ such that $\mathfrak{I}_1\frac{a}{x} \models \psi$
iff there exists $a \in A$ such that $\mathfrak{I}_2\frac{a}{x} \models \psi$
iff $\mathfrak{I}_2 \models \exists x\psi$ ■

Remark 1.3.6. If $\mathfrak{I} = (\mathfrak{A}, \beta)$ and $\text{free}(\varphi) = \{v_0, \ldots, v_{n-1}\}$ with $\beta(v_i) = a_i \in A$ for all i, then

1. $\mathfrak{I} \models \varphi$ is equivalent to $\mathfrak{A} \models \varphi[a_0, \ldots, a_{n-1}]$
2. $\mathfrak{I}(t)$ is equivalent to $t^{\mathfrak{A}}[a_0, \ldots, a_{n-1}]$
3. if φ is a sentence and $\mathfrak{I} \models \varphi$, then $\mathfrak{A} \models \varphi$

Definition 1.3.7. Let S, S' be symbol sets with $S \subset S'$ and $\mathfrak{A} = (A, \mathfrak{a})$ an S-structure and $\mathfrak{A}' = (A, \mathfrak{a}')$ an S'-structure so that $\mathfrak{a}, \mathfrak{a}'$ agree on S. Then

· $\mathfrak{A}$ is termed a reduct of $\mathfrak{A}'$
· $\mathfrak{A}'$ is termed an expansion of $\mathfrak{A}$, expressed $\mathfrak{A} = \mathfrak{A}'|_S$

Moreover, we note that by the coincidence lemma,

$$\mathfrak{A} \models \varphi[a_0, \ldots, a_{n-1}] \text{ iff } \mathfrak{A}' \models \varphi[a_0, \ldots, a_{n-1}]$$

Definition 1.3.8. Let $\mathfrak{A}, \mathfrak{B}$ be S-structures. Then a map $\pi : \mathfrak{A} \to \mathfrak{B}$ is an isomorphism iff

1. π is a bijection between A and B
2. if $R \in S$ and $a_1, \ldots, a_n \in A$, then $(a_1, \ldots, a_n) \in R^{\mathfrak{A}}$ iff $(\pi(a_1), \ldots, \pi(a_n)) \in R^{\mathfrak{B}}$
3. if $f \in S$ and $a_1, \ldots, a_n \in A$, then $\pi(f^{\mathfrak{A}}(a_1, \ldots, a_n))$ iff $f^{\mathfrak{B}}(\pi(a_1), \ldots, \pi(a_n))$
4. for all $c \in S$, $\pi(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$

If such a π exists, then $\mathfrak{A}$ and $\mathfrak{B}$ are termed isomorphic, and described $\mathfrak{A} \cong \mathfrak{B}$.

Lemma 1.3.9. [Isomorphism lemma]
If $\mathfrak{A}, \mathfrak{B}$ are isomorphic S-structures, then for all S-sentences φ, $\mathfrak{A} \models \varphi \iff \mathfrak{B} \models \varphi$.

Definition 1.3.10. Let $\mathfrak{A}, \mathfrak{B}$ be S-structures. Then $\mathfrak{A}$ is a substructure of $\mathfrak{B}$ iff

1. $A \subset B$
2. i. $R \in S \implies R^{\mathfrak{A}} = R^{\mathfrak{B}} \cap A^n$
ii. $f \in S \implies f^{\mathfrak{A}} = f^{\mathfrak{B}}|_{A^n}$
iii. $c \in S \implies c^{\mathfrak{A}} = c^{\mathfrak{B}}$

This relationship is then expressed $\mathfrak{A} \subset \mathfrak{B}$.

Lemma 1.3.11. [Substructure lemma]
Let $\mathfrak{A}, \mathfrak{B}$ be S-structures with $\mathfrak{A} \subset \mathfrak{B}$ and $\varphi \in L_n^S$ universal. Then for all $a_0, \ldots, a_{n-1} \in A$,

$$\mathfrak{B} \models \varphi[a_0, \ldots, a_{n-1}] \quad \text{implies} \quad \mathfrak{A} \models \varphi[a_0, \ldots, a_{n-1}]$$

Proposition 1.3.12. Let $\mathfrak{A}, \mathfrak{B}$ be S-structures with $\mathfrak{A} \subset \mathfrak{B}$ and $\varphi \in L_0^S$ existential. Then

$$\mathfrak{A} \models \varphi \quad \text{implies} \quad \mathfrak{B} \models \varphi$$

Definition 1.3.13. For arbitrary terms $t_0, \ldots, t_r$ and pairwise distinct variables of φ $x_0, \ldots, x_r$, define

$$\varphi\frac{t_0, \ldots, t_r}{x_0, \ldots, x_r} := \varphi \text{ with } x_i \text{ replaced by } t_i \text{ for all } i$$

Lemma 1.3.14. [Substitution lemma]

1. For every term t, $\mathfrak{I}\left(t\frac{t_1, \ldots, t_r}{x_0, \ldots, x_r}\right) = \mathfrak{I}\frac{\mathfrak{I}(t_1) \ldots, \mathfrak{I}(t_r)}{x_0, \ldots, x_r}(t)$
2. For every formula φ, $\mathfrak{I} \models \varphi\frac{t_1, \ldots, t_r}{x_0, \ldots, x_r}$ iff $\mathfrak{I}\frac{\mathfrak{I}(t_1) \ldots, \mathfrak{I}(t_r)}{x_0, \ldots, x_r} \models \varphi$

2 Sequent calculus

2.1 Consistency

Definition 2.1.1. A non-empty sequence of formulae Γ is termed a sequent. A set of rules associated with it is termed a sequent calculus $\mathfrak{S}$.

Definition 2.1.2. A formula φ is termed formally provable or derivable from a set of formulae Φ iff there are finitely many formulae (the antecedents) $\varphi_1, \ldots, \varphi_n$ such that given them, one may obtain φ (the succedent). This is expressed $\Phi \vdash \varphi$.

If $\varphi_1, \ldots, \varphi_n$ are in a sequence of formulae Γ, then we write $\vdash \Gamma\varphi$ with the same meaning.

Theorem 2.1.3. [Soundness theorem]
For a sequent Γ, if $\vdash \Gamma\varphi$, then $\Gamma \models \varphi$. Moreover, if $\Phi \vdash \varphi$, then there exists a sequence of formulae Γ from Φ such that $\vdash \Gamma\varphi$.

Definition 2.1.4. A set of formulae Φ is termed consistent and denoted $\text{Con}(\Phi)$ iff there is no formula φ such that $\Phi \vdash \varphi$ and $\Phi \vdash \neg\varphi$. If this occurs, then Φ is termed inconsistent and denoted $\text{Inc}(\Phi)$.

Lemma 2.1.5. $\text{Inc}(\Phi)$ iff for all φ, $\Phi \vdash \varphi$.

Proof: ($\Leftarrow$): Let $\varphi = v_0 \equiv v_0$.
So $\Phi \vdash \varphi$ and $\Phi \vdash \neg\varphi$, so $\text{Inc}(\Phi)$.

($\Rightarrow$): Suppose $\text{Inc}(\Phi)$.
Let φ be arbitrary.
Then there is ψ such that $\Phi \vdash \psi$ and $\Phi \vdash \neg\psi$.
So there are sequents $\Gamma_1, \Gamma_2 \subset \Phi$ such that $\vdash \Gamma_1\psi$ and $\vdash \Gamma_2\neg\psi$.
Since $\vdash \Gamma_1\psi$, we have $\vdash \Gamma_1\Gamma_2\neg\varphi\psi$ by (Ant).
Since $\vdash \Gamma_2\neg\psi$, we have $\vdash \Gamma_1\Gamma_2\neg\varphi\neg\psi$ by (Ant).
Thus $\vdash \Gamma_1\Gamma_2\varphi$ by (Ctr).
Since $\Gamma_1, \Gamma_2 \subset \Phi$, we have $\Phi \vdash \varphi$. ■

Corollary 2.1.6. $\text{Con}(\Phi)$ iff there is some formula that is not derivable from Φ.

Lemma 2.1.7. $\text{Con}(\Phi)$ iff $\text{Con}(\Phi_0)$ for all finite sets $\Phi_0 \subset \Phi$.

Lemma 2.1.8. $\text{Sat}(\Phi)$ implies $\text{Con}(\Phi)$

Lemma 2.1.9. For all Φ and φ:
1. $\Phi \vdash \varphi$ iff $\text{Inc}(\Phi \cup \{\neg\varphi\})$
2. if $\text{Con}(\Phi)$, then either $\text{Con}(\Phi \cup \{\varphi\})$ or $\text{Con}(\Phi \cup \{\neg\varphi\})$.

Lemma 2.1.10. For $n \in \mathbb{N}$, let S_n be symbol sets such that $S_0 \subset S_1 \subset S_2 \subset \cdots$. Let Φ_n be a set of S_n-formulae so that $\text{Con}_{S_n}(\Phi_n)$ and $\Phi_1 \subset \Phi_2 \subset \cdots$. Let $S = \bigcup_{n\in\mathbb{N}} S_n$ and $\Phi = \bigcup_{n\in\mathbb{N}} \Phi_n$. Then $\text{Con}_S(\Phi)$.

2.2 Completeness

Definition 2.2.1. A set of formulae Φ is termed negation complete iff for every formula φ, either $\Phi \vdash \varphi$ or $\Phi \vdash \neg\varphi$.

Definition 2.2.2. A set of formulae Φ contains witnesses iff for every formula of the form $\exists x\varphi$, there is a term t such that $\Phi \vdash \exists x\varphi \to \varphi\frac{t}{x}$.

Lemma 2.2.3. Suppose Φ is consistent, negation complete, and contains witnesses. Then

1. $\Phi \vdash \neg\varphi$ iff not $\Phi \vdash \varphi$
2. $\Phi \vdash (\varphi \vee \psi)$ iff $\Phi \vdash \varphi$ or $\Phi \vdash \psi$
3. $\Phi \vdash \exists x\varphi$ iff there is a term t such that $\Phi \vdash \varphi\frac{t}{x}$

Definition 2.2.4. Let Φ be a set of formulae and t_1, t_2 terms. Then define the relation $\sim$ by

$$t_1 \sim t_2 \quad \text{iff} \quad \Phi \vdash t_1 \equiv t_2$$

Then $\sim$ is an equivalence relation.

Lemma 2.2.5. If $t_1 \sim t_1', \ldots, t_n \sim t_n'$, then for an n-ary function symbol $f \in S$, $ft_1 \ldots t_n \sim ft_1' \ldots t_n'$. Moreover, for an n-ary relation symbol $R \in S$,

$$\Phi \vdash Rt_1 \ldots t_n \quad \text{iff} \quad \Phi \vdash Rt_1' \ldots t_n'$$

Definition 2.2.6. Define the following symbols:

$$T^S := \{t \mid t \text{ is an } S\text{-term}\}$$
$$\bar{t} := \{t' \in T^S \mid t \sim t'\}$$
$$T^\Phi := \{\bar{t} \mid t \in T^S\}$$

And the S-structure $\mathfrak{T}^\Phi$ over T^S such that

$$\text{for } n\text{-ary } R \in S,\ R^{\mathfrak{T}^\Phi}\overline{t_1} \ldots \overline{t_n} \ \text{ iff } \ \Phi \models Rt_1, \ldots t_n$$
$$\text{for } n\text{-ary } f \in S,\ f^{\mathfrak{T}^\Phi}(\overline{t_1} \ldots \overline{t_n}) = \overline{ft_1, \ldots t_n}$$
$$\text{for } c \in S,\ c^{\mathfrak{T}^\Phi} = \bar{c}$$

And for an assignment β, let

$$\beta^\Phi(x) = \bar{x}$$

Therefore we have constructed $\mathfrak{I}^\Phi = (\mathfrak{T}^\Phi, \beta^\Phi)$, the term interpretation associated with Φ.

Theorem 2.2.7. Let Φ be a consistent set of formulae which is negation complete and contains witnesses. Then Φ is satisfiable.

Lemma 2.2.8. Let S be at most countable with $\Phi \subset L^S$ consistent and free(Φ) finite. Then there exists $\Theta \supset \Phi$ which is consistent, negation complete, and contains witnesses. Moreover, this implies that Θ and Φ are satisfiable.

Definition 2.2.9. Let S be an arbitrary symbol set. To each $\varphi \in L^S$ associate a constant c_φ such that $c_\varphi = c_\psi$ iff $\varphi \equiv \psi$. Then define

$$S^* := S \cup \{c_{\exists x\varphi} \mid \exists x\varphi \in L^S\}$$
$$W(S) := \{(\exists x\varphi \to \varphi\tfrac{c_{\exists x\varphi}}{x}) \mid \exists x\varphi \in L^S\}$$

Lemma 2.2.10. For $\Phi \subset L^S$, if $\mathrm{Con}_S(\Phi)$, then $\mathrm{Con}_{S*}(\Phi \cup W(S))$.

Definition 2.2.11. Let M be a set and U a non-empty set of subsets of M. Then a non-empty set $D \subset U$ is termed a <u>chain</u> of U iff for all $V_1, V_2 \in D$, either $V_1 \subset V_2$ or $V_2 \subset V_1$.

Lemma 2.2.12. [Zorn]
If $\bigcup_{V\in D} V \in U$ for every chain $D \subset U$, then U has a maximal element. That is, there is some $U_0 \in U$ such that there does not exist $U_1 \in U$ with $U_0 \subsetneq U_1$.

Theorem 2.2.13. [Completeness]

$$\Phi \models \varphi \quad \text{iff} \quad \Phi \vdash \varphi$$
$$\mathrm{Sat}(\Phi) \quad \text{iff} \quad \mathrm{Con}(\Phi)$$

2.3 Ideas of Leopold Lowenheim and Thoralf Skolem

Theorem 2.3.1. [Lowenheim, Skolem]
Every satisfiable and at most countable set of formulae is satisfiable over a domain which is at most countable.

Proof: Let Φ be an at most countable set of S-sentences which is satisfiable and hence consistent.
There are at most countably many S-symbols in Φ, as every S-formula contains finitely many symbols.
Therefore WLOG S is at most countable.
By previous knowledge, there exists an interpretation $\mathfrak{I}$ that satisfies Φ with terms ranging over T^S.
Since T^S is at most countable, A is at most countable. ∎

Corollary 2.3.2. Every at most countable set of formulae that is satisfiable over an infinite domain is satisfiable over a countable domain.

Theorem 2.3.3. [Compactness]
We combine a previous theorem with a new one, together for the clear analogy:
1a. $\text{Con}(\Phi)$ iff $\text{Con}(\Phi_0)$ for all finite $\Phi_0 \subset \Phi$
1b. $\Phi \vdash \varphi$ iff $\Phi_0 \vdash \varphi$ for some finite $\Phi_0 \subset \Phi$

2a. $\text{Sat}(\Phi)$ iff $\text{Sat}(\Phi_0)$ for all finite $\Phi_0 \subset \Phi$
2b. $\Phi \vDash \varphi$ iff $\Phi_0 \vDash \varphi$ for some finite $\Phi_0 \subset \Phi$

Theorem 2.3.4. Let Φ be a set of formulae which is satisfiable over arbitrarily large finite domains. Then Φ is also satisfiable over an infinite domain.

Theorem 2.3.5. [Lowenheim, Skolem - "downward" variant]
Let $\Phi \subset L^S$ be satisfiable. Then Φ is satisfiable over a domain of cardinality at most $|L^S|$.

Theorem 2.3.6. [Lowenheim, Skolem - "upward" variant]
Let $\Phi \subset L^S$ be satisfiable over an infinite domain. Then for every set A there is a model of Φ which contains at least as many elements as A.

Theorem 2.3.7. [Lowenheim, Skolem, Tarski]
Let $\Phi \subset L^S$ be satisfiable over an infinite domain. Then for any $\kappa \geqslant |\Phi|$, Φ has a model of cardinality κ.

2.4 Elementary classes

Definition 2.4.1. Let Φ be a set of S-sentences. Define the class of models of Φ by

$$\text{Mod}^S(\Phi) := \{\mathfrak{A} \mid \mathfrak{A} \text{ is an } S\text{-structure}, \mathfrak{A} \vDash \Phi\}$$

Definition 2.4.2. Let $\mathfrak{K}$ be a class of S-structures. Then
1. $\mathfrak{K}$ is termed elementary iff there is an S-sentence φ such that $\mathfrak{K} = \text{Mod}^S(\varphi)$
2. $\mathfrak{K}$ is termed Δ-elementary iff there is a set Φ of S-sentences such that $\mathfrak{K} = \text{Mod}^S(\Phi)$

Remark 2.4.3. Any elementary class is Δ-elementary. Moreover, a Δ-elementary class may be described as the intersection of elementary classes.

· The class of fields is elementary.

· The class of fields with characteristic p prime is elementary.

Definition 2.4.4. Let $\mathfrak{A}, \mathfrak{B}$ be S-structures. Then $\mathfrak{A}$ and $\mathfrak{B}$ are termed elementarily equivalent, denoted $\mathfrak{A} \equiv \mathfrak{B}$, iff for every S-sentence φ, $\mathfrak{A} \vDash \varphi$ iff $\mathfrak{B} \vDash \varphi$.

Definition 2.4.5. A set Φ of S-sentences is termed independent iff there is no $\varphi \in \Phi$ such that $\Phi \backslash \{\varphi\} \vdash \varphi$.

Definition 2.4.6. Let $\mathfrak{A}$ be an S-structure. Then define the theory of $\mathfrak{A}$ to be

$$\text{Th}(\mathfrak{A}) = \{\varphi \in L_0^S \mid \mathfrak{A} \models \varphi\}$$

Lemma 2.4.7. Let $\mathfrak{A}, \mathfrak{B}$ be S-structures. Then $\mathfrak{B} \equiv \mathfrak{A}$ iff $\mathfrak{B} \models \text{Th}(\mathfrak{A})$.

· Note that by the isomorphism lemma, $\{\mathfrak{B} \mid \mathfrak{B} \cong \mathfrak{A}\} \subset \{\mathfrak{B} \mid \mathfrak{B} \equiv \mathfrak{A}\}$.

Theorem 2.4.8. Let $\mathfrak{A}$ be an S-structure. Then
1. if $\mathfrak{A}$ is infinite, then $\{\mathfrak{B} \mid \mathfrak{B} \cong \mathfrak{A}\}$ is not Δ-elementary
2. $\{\mathfrak{B} \mid \mathfrak{B} \equiv \mathfrak{A}\}$ is Δ-elementary
Moreover, $\{\mathfrak{B} \mid \mathfrak{B} \equiv \mathfrak{A}\}$ is the smallest Δ-elementary class containing $\mathfrak{A}$.

Definition 2.4.9. Consider $S_{ar} := (+, \cdot, 0, 1)$ and $\mathfrak{N} := (\mathbb{N}, +^{\mathbb{N}}, \cdot^{\mathbb{N}}, 0^{\mathbb{N}}, 1^{\mathbb{N}})$. A structure which is elementarily equivalent but not ismorphic to $\mathfrak{N}$ is termed a non-standard model of arithmeric.

In general, $\mathfrak{A}$ is a non-standard model of $\mathfrak{B}$ iff $\mathfrak{A} \equiv \mathfrak{B}$ and $\mathfrak{A} \not\cong \mathfrak{B}$.

Theorem 2.4.10. There exists a countable non-standard model of arithmetic.

Proof: Let $\Psi = \text{Th}(\mathfrak{N}) \cup \{\neg x \equiv 0, \neg x \equiv 1, \neg x \equiv \underline{2}, \ldots\}$.
Let $\Phi \subset \Psi$ be finite.
So there exists $m \in \mathbb{N}$ such that for all $n \geqslant m$, $\neg x \equiv \underline{m} \notin \Phi$.
Then $(\mathfrak{N}, \beta)$ is a model for Φ if $\beta(x) = n$.
By the completeness theorem, there is a model of Ψ, so by Lowenheim-Skolem, since Ψ is countable, Ψ has an at most countable model, say $(\mathfrak{A}, \beta)$.
Observe that $\mathfrak{A}$ is elementarily equivalent to $\mathfrak{N}$, since $\mathfrak{A} \models \text{Th}(\mathfrak{N})$.
Also note that $\mathfrak{A} \not\cong \mathfrak{N}$, since an ismorphism must map n to $\underline{n}^{\mathfrak{A}}$, but there is nothing to map $\beta(x)$ to. ∎

Note that above we have used the convention $\underline{n} := \underbrace{1 + 1 + \cdots + 1}_{n \text{ times}} = \underbrace{ff \cdots f}_{n \text{ times}} 1$ for f the successor function.

2.5 Abstraction and simplification

Definition 2.5.1. An S-formula φ is termed term-reduced iff its atomic subformulae have one of the following forms, where $y, x, x_1, \ldots, x_n$ are variables and c is a constant.

$$\begin{array}{ll} Rx_1 \ldots x_n & x \equiv y \\ fx_1 \ldots x_n \equiv x & c \equiv x \end{array}$$

Theorem 2.5.2. For every S-formula φ there is a logically equivalent term-reduced formula φ^*.

Note that $\text{free}(\varphi) = \text{free}(\varphi^*)$.

Definition 2.5.3. A symbol set S is termed relational iff it contains only relation symbols.

Definition 2.5.4. To every symbol set S associate a relational symbol set S^r containing:
· all relation symbols in S
· for every n-ary function symbol $f \in S$, an $(n+1)$-ary relation symbol F
· for every constant symbol $c \in S$, a unary relation symbol C

To every S-structure $\mathfrak{A}$ associate an S^r structure $\mathfrak{A}^r$ by:
· $R^{\mathfrak{A}} = R^{\mathfrak{A}}$
· $F^{\mathfrak{A}^r}$ = the graph of $f^{\mathfrak{A}}$
· $C^{\mathfrak{A}^r}$ = the graph of $c^{\mathfrak{A}}$

Theorem 2.5.5. For S-structures $\mathfrak{A}, \mathfrak{B}$, $\mathfrak{A} \equiv \mathfrak{B}$ iff $\mathfrak{A}^r \equiv \mathfrak{B}^r$.

Definition 2.5.6. A formula which is the disjunction of conjunctions of atomic and negated atomic formulae is termed a formula in <u>disjunctive normal form</u>. Similarly, a formula which is the conjunction of disjunctions of atomic and negated atomic formulae is termed a formula in <u>conjunctive normal form</u>.

Theorem 2.5.7. If φ is quantifier-free, then φ is logically equivalent to a formula φ_0 in disjunctive normal form and φ_1 in conjunctive normal form.

<u>*Proof:*</u> We prove only that $\varphi \dashv\models \varphi_0$ here.
Suppose $\varphi \in L_r^S$.
Let $\{\varphi_0, \dots, \varphi_n\}$ be the atomic formulae appearing in φ.
For an S-structure $\mathfrak{A}$ and $\bar{a} = (a_1, \dots, a_r) \in A^r$, let $\psi_{(\mathfrak{A},\bar{a})} = \psi_0 \wedge \cdots \wedge \psi_n$, where

$$\psi_i = \begin{cases} \varphi_i & \text{if } \mathfrak{A} \models \varphi_i[\bar{a}] \\ \neg\varphi_i & \text{if } \mathfrak{A} \models \neg\varphi_i[\bar{a}] \end{cases}$$

Note that $\mathfrak{A} \models \psi_{(\mathfrak{A},\bar{a})}[\bar{a}]$, and there are at most 2^{n+1} formulae of the form $\psi_{(\mathfrak{A},\bar{a})}$.
Let $\chi := \bigvee\{\psi_{(\mathfrak{A},\bar{a})} \mid \mathfrak{A} \text{ is an } S\text{-structure, } \bar{a} \in A^r \text{ and } \mathfrak{A} \models \varphi[\bar{a}]\}$.
Note that $\chi \in L_r^S$ is in disjunctive normal form.
<u>Claim</u>: χ is logically equivalent to φ.
<u>Proof of claim</u>: Suppose $\mathfrak{B} \models \varphi[\bar{b}]$.
Then $\psi_{(\mathfrak{B},\bar{b})}$ is the disjunct of χ, and since $\mathfrak{B} \models \psi_{(\mathfrak{B},\bar{b})}[\bar{b}]$, we have $\mathfrak{B} \models \chi[\bar{b}]$.
Now suppose $\mathfrak{B} \models \chi[\bar{b}]$.
Then there is some S-structure $\mathfrak{A}$ and some $\bar{a} \in A^r$ with $\mathfrak{A} \models \varphi[\bar{a}]$ such that $\mathfrak{B} \models \psi_{(\mathfrak{A},\bar{a})}[\bar{b}]$.
Then for each atmoic formula φ_i appearing in φ, $\mathfrak{B} \models \varphi_i[\bar{b}]$ iff $\mathfrak{A} \models \varphi_i[\bar{a}]$.
Since φ is obtained from the φ_i using only $\vee$ and $\neg$, we have that $\mathfrak{B} \models \varphi[\bar{b}]$ iff $\mathfrak{A} \models \varphi[\bar{a}]$.
So since $\mathfrak{A} \models \varphi[\bar{a}]$, we have $\mathfrak{B} \models \varphi[\bar{b}]$. ■

Definition 2.5.8. A formula which has the from $\varphi = Q_1x_1 \dots Q_nx_n\varphi$ for $Q_i \in \{\exists, \forall\}$ for all i and φ quantifier free is termed a formula in <u>prenex normal form</u>.

- $\cdot$ $Q_1x_1 \dots Q_nx_n$ is termed the <u>prefix</u> of φ
- $\cdot$ φ_0 is termed the <u>matrix</u> of φ

Theorem 2.5.9. Every formula φ is logically equivalent to a formula ψ in prenex normal form with $\text{free}(\varphi) = \text{free}(\psi)$.

<u>*Proof:*</u> Let $\varphi \sim \psi$ denote $\varphi \dashv\models \psi$.
We note that:
1. $\varphi \sim \psi$ implies $\neg\varphi \sim \neg\psi$
2. $\varphi_0 \sim \psi_0$ and $\varphi_1 \sim \psi_1$ implies $(\varphi_0 \vee \varphi_1) \sim (\psi_0 \vee \psi_1)$
3. $\varphi \sim \psi$ implies $Qx\varphi \sim Qx\psi$
4. $\neg Qx\varphi \sim Q^{-1}x\neg\varphi$
5. $x \notin \text{free}(\varphi)$ implies $(qx\varphi \vee \psi) \sim Qx(\varphi \vee \psi)$ and $(\psi \vee Qx\varphi) \sim Qx(\psi \vee \varphi)$
6. $\varphi \vee \psi \sim \psi \vee \varphi$
For $\varphi \in L^S$, let $\text{qn}(\varphi)$ be the number of quantifiers occuring in φ.
We prove the theorem by induction on n.
Let $P(n)$ be the statement "For φ with $\text{qn}(n) \leqslant n$, there is $\psi \in L^S$ in prenex normal form such that $\varphi \sim \psi$, $\text{free}(\varphi) = \text{free}(\psi)$ and $\text{qn}(\varphi) = \text{qn}(\psi)$".
<u>$n = 0$:</u> If $\text{qn}(\varphi) = 0$, we can set $\psi = \varphi$.
<u>$n > 0$:</u> Suppose $\varphi = \neg\varphi'$.
Then $\text{qn}(\varphi') = \text{qn}(\varphi)$ and $\text{free}(\varphi') = \text{free}(\varphi)$.
By the induction hypothesis, there is a formula $Qx\chi$ that is a prenex normal form for φ with $\text{qn}(Qx\chi) = \text{qn}(\varphi')$ and $\text{free}(Qx\chi) = \text{free}(\varphi')$.

Then $\varphi' \sim Qx\chi$ implies $\varphi \equiv \neg\varphi' \sim \neg Qx\chi$ by **1.** above.
Further, $\neg Qx\chi \sim Q^{-1}x\neg\chi$ by **4.** above.
Note $\text{free}(\neg\chi) = \text{free}(\chi)$ and $\text{qn}(\neg\chi) = \text{qn}(\chi) = \text{qn}(\varphi) - 1 \leqslant n-1$
Since $P(n-1)$ holds, there is a prenex normal form ψ for $\neg\chi$ with $\text{qn}(\psi) = \text{qn}(\chi)$ and $\text{free}(\psi) = \text{free}(\chi)$.
Thus $Q^{-1}x\psi$ is the desired prenex normal form for φ by **3.** above.

Suppose $\varphi = (\varphi' \vee \varphi'')$ and $\text{qn}(\varphi) > 0$.
WLOG assume $\text{qn}(\varphi') > 0$.
By the induction hypothesis, there is a formula $Qx\chi$ that is a prenex normal form for φ' with $\text{free}(Qx\chi) = \text{free}(\varphi')$ and $\text{qn}(Qx\chi) = \text{qn}(\varphi')$.
Let y be a variable which does not occur in $Qx\chi$ or φ''.
Then $Qx\chi \sim Qy\chi\frac{y}{x}$.
So by **2.** and **5.** above,

$$\varphi = (\varphi' \vee \varphi'') \sim (Qy\chi\frac{y}{x} \vee \varphi'') \sim Qy(\chi\frac{y}{x} \vee \varphi'')$$

So $\text{qn}(\chi\frac{y}{x} \vee \varphi'') = \text{qn}(\varphi) - 1 \leqslant n-1$.
Since $P(n-1)$ holds, there is a prenex normal form ψ for $\chi\frac{y}{x} \vee \varphi''$ with $\text{qn}(\psi) = \text{qn}(\chi\frac{y}{x} \vee \varphi'')$ and $\text{free}(\psi) = \text{free}(\chi\frac{y}{x} \vee \varphi'')$.
Then $Qy\psi$ is the desired prenex normal form for φ.
We also note that

$$\begin{aligned}\text{free}(Qy\psi) &= \text{free}(\chi\tfrac{y}{x} \vee \varphi'')\backslash\{y\} \\ &\subset \text{free}(\chi)\backslash\{x\} \cup \text{free}(\varphi'') \\ &= \text{free}(Qx\chi) \cup \text{free}(\varphi'') \\ &= \text{free}(\varphi') \cup \text{free}(\varphi'') \\ &= \text{free}(\varphi)\end{aligned}$$

Suppose $\varphi = \exists x\varphi'$.
Since $\text{qn}(\varphi') \leqslant n-1$, there is a prenex normal form ψ with $\varphi' \sim \psi$ and $\text{free}(\psi) = \text{free}(\varphi')$ and $\text{qn}(\psi) = \text{qn}(\varphi')$.
So $\exists x\psi$ is the desired prenex normal form for φ. ■

Remark 2.5.10. A countably infinite symbol set may be viewed as being defined over a finite alphabet.

3 Programming logic

3.1 Heuristic

Definition 3.1.1. A procedure P may run on inputs of words over a language. It may have an output and it may halt.

Definition 3.1.2. Let A be an alphabet, $W \subset A^*$ and P a procedure. Then
1. P is a decision procedure for W iff for every input $\xi \in A^*$, P eventually stops, having (before stopping) given exactly one output η such that
$\eta = \square$ iff $\xi \in W$
$\eta \neq \square$ iff $\xi \notin W$
2. P is an enumeration procedure for W if P, having been initiated, yields eventually as output any word in in W, in any order, with possible repetition.
Then we may describe W by saying that
i. W is decideable iff there exists a decision procedure for W
ii. W is enumerable iff there exists an enumeration procedure for W

Remark 3.1.3. If A is a finite alphabet, then A^* is enumerable.

Remark 3.1.4. The set $\{\varphi \in L_0^{S_\infty} \mid \models \varphi\}$ is enumerable.

Proof: By the completeness theorem, we need to enumerate all S_∞-sentences such that $\vdash \varphi$.
We may list all words over the language, checking if each word is a formula.
For each $n \in \mathbb{N}$, form all the (finite) combinations of the first n formulae in the list.
Check, for each combination, if it is a derivation ending with a sentence φ.
If so, list φ. ■

Theorem 3.1.5. Every decideable set is enumerable.

Theorem 3.1.6. A subset $W \subset A^*$ is decideable iff W and $A^*\backslash W$ are enumerable.

Proof: ($\Rightarrow$) Clearly a decision procedure P for W can be made into a decision procedure P' for $A^*\backslash W$.
By the above theorem, A and $A^*\backslash W$ are both enumerable.

($\Leftarrow$) Suppose W and $A^*\backslash W$ are enumerable by P and P'.
To decide whether $\xi \in W$, run P and P' until one lists ξ.
Exactly one will list ξ, as $W \cap A^*\backslash W = \varnothing$, and $W \cup A^*\backslash W = A^*$. ■

Definition 3.1.7. A computable function $f : A^* \to B^*$ is a function for which there is a procedure P that with input $\xi \in A^*$ halts with output $f(\xi) \in B^*$.

3.2 Formal

Definition 3.2.1. A register R is an indefinitely large unit of memory in which a word may be stored. We assume that an indefinite amount of register machines are avilable for use.

Definition 3.2.2. Fix an alphabet $A = \{a_0, \ldots, a_n\}$. A register program P over an alphabet A is a finite sequence $\alpha_0, \ldots, \alpha_k$ of instructions of the type below.

1 LET $R_i = R_i + a_j$	[add-instruction]	
2 LET $R_i = R_i - a_j$	[sub-instruction]	if a_j is not last in R_i, do nothing
3 IF $R_i = \square$ THEN L' ELSE L_0 OR $\cdots$ OR L_r	[jump-instruction]	if a_j is last, do L_j
4 PRINT	[print-instruction]	output the word in R_0
5 HALT	[halt-instruction]	stop the procedure

Above we assume $0 \leqslant j \leqslant n$, $i \in \mathbb{N}$, and $R_1, R_2, \ldots$ are register machines.

We assume certain properties of register machines:
1. α_i has label i
2. every jump-instruction refers to labels $\leqslant k$
3. only the last line, α_k, is a halt-instruction

Definition 3.2.3. A program P is started with the a word $\xi \in A^*$ if P begins the computation with ξ in R_0 and $\square$ in the remaining registers.

· If P started with ξ and reaches the halt-instruction, we write $P : \xi \to$ HALT. Otherwise, write $P : \xi \to \infty$.
· If P started with ξ and prints exactly one word η and later halts, we write $P : \xi \to \eta$.

Definition 3.2.4. To abbreviate a special instance of rule **3.** we equivalently say:

$$\texttt{IF } R_0 = \square \texttt{ THEN } L' \texttt{ ELSE } L' \texttt{ OR } \cdots \texttt{ OR } L'$$
$$\Updownarrow$$
$$\texttt{GOTO } L'$$

Definition 3.2.5. Let $W \subset A^*$. A program P decides W iff for all $\xi \in A^*$

$$\begin{array}{lll} P : \xi \to \square & \text{iff} & \xi \in W \\ P : \xi \to \eta & \text{iff} & \xi \notin W \text{ and } \eta \neq \square \end{array}$$

Then W is termed register decidable iff there is a program P that decides W.

Definition 3.2.6. Let $W \subset A^*$.

· A program P enumerates W iff P started with $\square$ and prints exactly all the words in W, with possible repetitions, and in any order.

· W is register enumerable iff there exists a program that enumerates W.

Definition 3.2.7. Let A, B be alphabets and $F : A^* \to B^*$.

· A program P over $A \cup B$ computes F iff for all $\xi \in A^*$, $P : \xi \to F(\xi)$.

· F is register-computable iff there is a program that computes F.

Remark 3.2.8. The left column comes from the definitions above. Church conjectures the right column.

$$\begin{array}{rcl} R\text{-decidable} & \Longrightarrow & \text{decidable} \\ R\text{-enumerable} & \Longrightarrow & \text{enumerable} \\ R\text{-computable} & \Longrightarrow & \text{computable} \end{array} \qquad \begin{array}{rcl} \text{decidable} & \Longrightarrow & R\text{-decidable} \\ \text{enumerable} & \Longrightarrow & R\text{-enumerable} \\ \text{computable} & \Longrightarrow & R\text{-computable} \end{array}$$

4 The limits of first-order logic

4.1 Undecidability

Let $A = \{a_0, \dots, a_r\}$. Let $B = A \cup \{\mathtt{A}, \mathtt{B}, \dots, \mathtt{Z}\} \cup \{0, 1, \dots, 9\} \cup \{=, +, -, \square, \S\}$. Then to every program P we associate a unique word over B. For example,

$$\begin{array}{l} 0\ \mathtt{LET}\ R_1 = R_1 + a_0 \\ 1\ \mathtt{PRINT} \\ 2\ \mathtt{HALT} \end{array} \quad \Longrightarrow \quad \mathtt{0LETR1{=}R2{+}}a_0\mathtt{\S 1PRINT\S 2HALT}$$

Consider a lexicographic ordering of B^*. Then for a program P over A, we can find its equivalent under association in this ordering, say it is at position n. Then define $\xi_P = \underbrace{a_0 \dots a_0}_{n \text{ times}}$ to be the Godel number of P.

Lemma 4.1.1. Let $\Pi = \{\xi_P \mid P \text{ is a program over } A\}$. Then Π is deciedable.

Proof: Given a word in A^*, check whether it is of the form $\underbrace{a_0 \dots a_0}_{n \text{ times}}$.

If not, then it is not in Π.

If yes, loon at the nth word in the ordering of B^*.

Check whether this codes a program over A.

Since the word is of finite length, we can check it. ■

Theorem 4.1.2. [Undecidability of the halting problem]

a. The set $\Pi'_{\mathtt{HALT}} = \{\xi_P \mid P \text{ is a program over } A \text{ and } P : \xi_P \to \mathtt{HALT}\}$ is not R-decidable

b. The set $\Pi_{\mathtt{HALT}} = \{\xi_P \mid P \text{ is a program over } A \text{ and } P : \square \to \mathtt{HALT}\}$ is not R-decidable

Proof: (**a.**) Suppose there exists a program P_0 that decides $\Pi'_{\mathtt{HALT}}$.

Then for all P,

$$\begin{array}{lll} P_0 : \xi_P \to \square & \text{iff} & P : \xi_P \to \mathtt{HALT} \\ P_0 : \xi_P \to \eta & \text{iff} & P : \xi_P \to \infty \text{ for } \eta \neq \square \end{array}$$

From P_0 we obtain a program P_1 by making the substitution

$$k\ \mathtt{HALT} \implies k\ \mathtt{IF}\ R_0 = \square\ \mathtt{THEN}\ k\ \mathtt{ELSE}\ k+1\ \mathtt{OR}\ \cdots\ \mathtt{OR}\ k+1$$

And adding the line

$$k+1 \ \texttt{HALT}$$

Then for this program P_1 we have that

$$\begin{array}{lll} P_1 : \xi_P \to \infty & \text{iff} & P : \xi_P \to \texttt{HALT} \\ P_1 : \xi_P \to \texttt{HALT} & \text{iff} & P : \xi_P \to \infty \end{array}$$

But then P_1 has a Godel number, so $P_1 : \xi_{P_1} \to \infty$ iff $P_1 : \xi_{P_1} \to \texttt{HALT}$.
This is a contradiction.

(**b.**) We design a procedure, that produces P^+ from P such that $\xi_P \in \Pi'_{\texttt{HALT}}$ iff $\xi_{P^+} \in \Pi_{\texttt{HALT}}$.
Given P, compute ξ_P with n instances of a_0.
Let P^+ be the program that begins with

$$\begin{array}{c} 0 \ \texttt{LET} \ R_0 = R_0 + a_0 \\ \vdots \\ n-1 \ \texttt{LET} \ R_0 = R_0 + a_0 \end{array}$$

followed by the lines of P, all incremented by n.
Clearly, $P : \xi_P \to \texttt{HALT}$ iff $P^+ : \square \to \texttt{HALT}$.
Now the result follows from (**a.**). ■

Lemma 4.1.3. $\Pi_{\texttt{HALT}}$ is enumerable.

Proof: For every $n \in \mathbb{N}$, get the finitely many programs with Godel number $\leqslant n$.
Start each program with $\square$, run for n steps, print the Godel number of the programs that halt. ■

Corollary 4.1.4. $A^* \backslash \Pi_{\texttt{HALT}}$ is not enumerable.

Definition 4.1.5. Let P be a program with instructions $\alpha_0, \ldots, \alpha_k$ and let $n \in \mathbb{N}$ be the maximal index of registers appearing in P. Then an $(n+2)$-tuple of rational numbers

$$(L, m_0, \ldots, m_n)$$

with $0 \leqslant L \leqslant k$ is termed the configuration of P after s steps if P started with $\square$, runs for at least s steps and after s steps L is to be executed next while the registers $R_0, \ldots, R_n$ contain the numbers $m_0, \ldots, m_n$, respectively.

In the above circumstances, the $(n+1)$-tuple $(0, \ldots, 0)$ is termed the initial configuration of P.

Remark 4.1.6. Since S_∞ has countably many function, relation, and constant symbols of each arity, we enumerate them and denote them by writing

$$\begin{array}{rl} R^n_m & \text{for the } m\text{th } n\text{-ary relation symbol} \\ f^k_\ell & \text{for the } \ell\text{th } k\text{-ary function symbol} \\ c_j & \text{for the } j\text{th constant symbol} \end{array}$$

Theorem 4.1.7. [Undecidability of first order logic]
The set $\{\varphi \in L_0^{S_\infty} \mid \models \varphi\}$ of valid S_∞ sentences is undecidable.

Proof: Let $A = \{1\}$, and identify words over A with natural numbers.
We assign to every program P in an effective way an S_∞ sentence φ_P such that $\models \varphi_P$ iff $P : \square \to \texttt{HALT}$.
This will show that $\Pi = \{\varphi \in L_0^{S_{ar}} \mid \models \varphi\}$ is undecidable.

Suppose the contrary.
Let $\xi \in A^*$ decide if $\xi \in \Pi$.
If not, $\xi \notin \Pi_{\texttt{HALT}}$.
If yes, compute P so that $\xi = \eta_P$.
Compute φ_P.
Use the decision procedure to decide whether $\models \varphi$.
If yes, $\xi \in \Pi_{\texttt{HALT}}$.
If no, $\xi \notin \Pi_{\texttt{HALT}}$.
So we have a decision procedure for $\Pi_{\texttt{HALT}}$.
This is a contradiction.

Now we define φ_P.
Let P be a program with instructions $\alpha_0, \dots, \alpha_k$.
Compute the smallest $n \in \mathbb{N}$ such that the registers occuring in P are among $R_0, \dots, R_n$.
Since α_k is the only halt-instruction, $P : \square \to \texttt{HALT}$ iff there exist $s, m_0, \dots, m_n \in \mathbb{N}$ such that $(k, m_0, \dots, m_n)$ is the configuration of P after s steps.
Let $R = R_0^{n+3}$ and $<= R_0^2$ and $f = f_0^1$ and $c = c_0$, all in S_∞.
Let $S = \{R, <, f, c\} \subset S_\infty$.
We associate to P an S-structure $\mathfrak{A}_P$ that describes P.
Set $\mathfrak{A}_P = \mathbb{N}$ and interpret $<$ by $<^{\mathbb{N}}$, c by 0, f by the successor function, R by $\{(s, L, m_0, \dots, m_n) \mid (L, m_0, \dots, m_n)$ is the configuratiton of P after s steps$\}$.

Now we define an S-sentence ψ_P that will appear in φ_P.
We want ψ_P to have the following properties:
(a). $\mathfrak{A}_p \models \psi_P$
(b). if $\mathfrak{A}$ is an S-structure with $\mathfrak{A} \models \psi_P$ and $RsLm_0 \dots m_n$, then $\mathfrak{A} \models R\overline{s}\overline{L}\overline{m_0}, \dots, \overline{m_n}$.
Let ψ_0 be the sentence describing that $f, c, <$ work as desired.

$$\psi_0 := \text{``}< \text{is an ordering"} \wedge \forall x(c < x \vee c \equiv x) \wedge \forall x(x < fx) \wedge \forall x \forall z(x < z \to (fx < z \vee fx \equiv z))$$

For $\alpha = \alpha_0, \dots, \alpha_{k-1}$ we define ψ_α by the following rules:
· If α is "L `LET` $R_i = R_i + 1$" then

$$\psi_\alpha = \forall x \forall y_0 \dots \forall y_n (Rx\overline{L}y_0 \dots y_n \to Rfx\overline{L+1}y_0 \dots y_{i-1}fy_iy_{i+1} \dots y_n)$$

· If α is "L `LET` $R_i = R_i - 1$" then

$$\psi_\alpha = \forall x \forall y_0 \dots \forall y_n (Rx\overline{L}y_0 \dots y_n \to ((y_i \equiv 0 \wedge Rfx\overline{L}y_0 \dots y_n) \vee (\neg y_i \equiv 0 \wedge \exists u(fu = y_i \wedge Rfx\overline{L+1}y_0 \dots y_{i-1}uy_{i+1} \dots y_n))))$$

· If α is "L `IF` $R_i = \square$ `THEN` L' `ELSE` L_0" then

$$\psi_\alpha = \forall x \forall y_0 \dots \forall y_n (Rx\overline{L}y_0 \dots y_n \to ((y_i \equiv 0 \wedge Rfx\overline{L'}y_0 \dots y_n) \vee (\neg y_i \equiv 0 \wedge Rfx\overline{L_0}y_0 \dots y_n)))$$

· If α is "L `PRINT`" then

$$\psi_\alpha = \forall x \forall y_0 \dots \forall y_n (Rx\overline{L}y_0 \dots y_n \to Rfx\overline{L+1}y_0 \dots y_n)$$

Let $\psi_P = \psi_0 \wedge R00 \cdots 0 \wedge \psi_{\alpha_0} \wedge \cdots \wedge \psi_{\alpha_{k-1}}$.
Then ψ_P satisfies **(a).** and **(b).** by induction.
Let $\varphi_P = \psi_P \to \exists x \exists y_o \dots \exists y_n Rx\overline{L}y_0 \dots y_n$.

Now we claim that φ_P is valid iff $P : \square \to \texttt{HALT}$.
Suppose $\models \varphi_P$.
Then $\mathfrak{A}_P \models \varphi_P$.
Thus $\mathfrak{A}_P \models \exists x \exists y_0 \dots \exists y_n Rx\overline{L}y_0 \dots y_n$.

So there are $s, m_0, \dots, m_n \in \mathfrak{A}_P$ such that $(s, k, m_0, \dots, m_n) \in R$.
In other words, the program P reaches the halt-configuration after s steps.
Thus $P : \square \to \texttt{HALT}$.

Suppose $P : \square \to \texttt{HALT}$, so P has a halt-configuration $(s, k, m_0, \dots, m_n)$.
Let $\mathfrak{A}$ be an arbitrary S-structure.
If $\mathfrak{A} \not\models \psi_P$, then $\mathfrak{A} \models \varphi_P$, as any result follows from a false statement.
If $\mathfrak{A} \models \psi_P$, then by **(b).** we have $\mathfrak{A} \models R\bar{s}\bar{k}\overline{m_0}, \dots, \overline{m_n}$.
So $\mathfrak{A} \models \exists x \exists y_0 \dots \exists y_n Rx\bar{k}y_0 \dots y_n$.
So then $\mathfrak{A} \models \varphi_P$.
Since $\mathfrak{A}$ was arbitrary, φ_P is valid. ∎

Definition 4.1.8. A set $T \subset L_0^S$ is termed a <u>theory</u> iff $\text{Sat}(T)$ and T is closed under logical consequence, i.e. $T = \{\varphi \mid T \models \varphi\}$. We define associated sets for general $\Phi \subset L^S$.

$$\Phi^{\models} := \{\varphi \in L^S \mid \Phi \models \varphi\}$$
$$\Phi^{\vdash} := \{\varphi \in L^S \mid \Phi \vdash \varphi\}$$

By the completeness theorem, we know that these two sets are equal.

4.2 Axiomatization

Definition 4.2.1. Let Φ_{PA} consist of the following S^{ar} sentences:

$$\begin{aligned}
&\forall x \neg x + 1 \equiv 0 \\
&\forall x x + 0 \equiv x \\
&\forall x x \cdot 0 \equiv 0 \\
&\forall x \forall y (x + 1 \equiv y + 1 \to x \equiv y) \\
&\forall x \forall y x + (y + 1) \equiv (x + y) + 1 \\
&\forall x \forall y x (y + 1) \equiv x \cdot y + x
\end{aligned}$$

And for all $x_1, \dots, x_n, y$ and all $\varphi \in L^{S_{ar}}$ such that $\text{free}(\varphi) \subset \{x_1, \dots x_n\}$, the sentence

$$\forall x_1 \dots \forall x_n \left(\left(\varphi \frac{0}{y} \wedge \forall y \left(\varphi \to \varphi \frac{y+1}{y} \right) \right) \to \forall y \varphi \right)$$

Then Φ_{PA} is termed the set of <u>first-order Peano axioms</u>.

· We note that $\mathfrak{N} \models \Phi_{PA}$, or equivalently, $\Phi_{PA}^{\models} \subset \text{Th}(\mathfrak{N})$.

Definition 4.2.2. A theory T is termed <u>R-axiomatizable</u> if there is an R-decidable set Φ such that $T = \Phi^{\models}$.
A theory T is termed <u>finitely axiomatizable</u> if there is a finite set Φ such that $T = \Phi^{\models}$.

Theorem 4.2.3. An R-axiomatizable theory is R-enumerable.

<u>Proof:</u> Let T be a theory.
Let Φ be an R-decidable (or enumerable) set of S-sentences such that $T = \Phi^{\models}$.
Generate systematically all derivable sequents.
Check for each whether the members of the antecedent belong to Φ.
If yes, and the succedent is a sentence, list the succedent. ∎

Definition 4.2.4. A theory $T \subset L_0^S$ is termed <u>complete</u> iff for every S-sentence φ we have $\varphi \in T$ or $\neg\varphi \in T$.
As a special case, for structures $\mathfrak{A}$, the theory <u>$\text{Th}(\mathfrak{A})$ is</u> always complete.

Theorem 4.2.5.

i. Every R-axiomatizable, complete theory is R-decidable.

ii. Every R-enumerable, complete theory is R-decidable.

Proof: (**i.**) Since R-axiomatizable implies R-enumerable, a proof of **ii.** will suffice.

(**ii.**) Execute the enumeration of T until either φ or $\neg\varphi$ is enumerated.

If φ is enumerated, then $\varphi \in T$.

If $\neg\varphi$ is enumerated, then $\varphi \notin T$, since T is satisfiable. ■

The folowing two lemmas will be used to prove the subsequent theorem.

Lemma 4.2.6. [β-FUNCTION LEMMA]
There is a function $\beta : \mathbb{N}^3 \to \mathbb{N}$ such that

1. for every sequence $(a_0, \ldots, a_r)$ over $\mathbb{N}$ there are $t, p \in \mathbb{N}$ such that for all $0 \leqslant i \leqslant r$, $\beta(t,p,i) = a_i$

2. β is definable in $L^{S_{ar}}$ - there is an S_{ar}-formula $\varphi_\beta(t,p,i,a)$ such that $\mathfrak{N} \models \varphi_\beta[t,p,i,a]$ iff $\beta(t,p,i) = a$

Lemma 4.2.7. [χ_P-LEMMA]
Given a program P, one may effectively associate to it a formula $\chi_P(v_0, \ldots, v_{2n+2})$ such that for all $\ell_0, \ldots, \ell_n, L, m_0, \ldots, m_n \in \mathbb{N}$ we have $\mathfrak{N} \models \chi_P[\ell_o, \ldots, \ell_n, L, m_0, \ldots, m_n]$ iff P, beginning with the configuration $(0, \ell_0, \ldots, \ell_n)$ after finitely many steps reaches the configuration $(L, m_0, \ldots, m_n)$.

Proof: We would like $\chi_P(x_0, \ldots, x_n, z, y_0, \ldots, y_n)$ to formalize the folowing:

$$\left(\begin{array}{c} \text{There is } s \in \mathbb{N} \text{ and a sequence of configurations } (c_i)_{i=0}^{s} \text{ such that:} \\ c_0 = (0, x_0, \ldots, x_n) \\ c_s = (z, y_0, \ldots, y_n) \\ \text{and for all } 0 \leqslant i < s,\ \text{we have } c_i \xrightarrow{P} c_{i+1} \end{array}\right)$$

Equivalently this may be stated as:

$$\left(\begin{array}{c} \text{There is } s \in \mathbb{N} \text{ and a sequence} \\ \underbrace{a_0, \ldots, a_{n+1}}_{c_0}, \underbrace{a_{n+2}, \ldots, a_{(n+2)+(n+1)}}_{c_1}, \ldots\ldots, \underbrace{a_{s(n+2)}, \ldots, a_{s(n+2)+(n+1)}}_{c_s} \\ \text{such that} \\ a_0 = 0, a_1 = x_0, \ldots, a_{n+1} = x_n, \ldots\ldots, a_{s(n+2)} = z, a_{s(n+2)+1} = y_0, \ldots, a_{s(n+2)+(n+1)} = y_n \\ \text{and for all } 0 \leqslant i < s, \\ \big(a_{i(n+1)}, \ldots, a_{i(n+2)+(n+1)}\big) \xrightarrow{P} \big(a_{(i+1)(n+1)}, \ldots, a_{(i+1)(n+2)+(n+1)}\big) \end{array}\right)$$

Using β from above, we complete the construction by setting

$$\begin{aligned} \chi_P(x_0, \ldots, x_n, z, y_0, \ldots, y_n) = \exists s \exists p \exists t \Big(& \varphi_\beta(t,p,0,0) \wedge \varphi_\beta(t,p,1,x_0) \wedge \cdots \wedge \varphi_\beta(t,p,\overline{n+1} x_n) \wedge \varphi_\beta(t,p,\overline{s(n+2)},z) \wedge \cdots \\ & \cdots \wedge \varphi_\beta(t,p,\overline{s(n+2)+(n+1)},y_n)) \\ & \wedge \forall i \Big(i < s \to \forall u \forall u_0 \cdots \forall u_n \forall u' \forall u'_0 \cdots \forall u'_n \Big(\varphi_\beta(t,p,\overline{i(n+2)},u) \wedge \cdots \\ & \cdots \wedge \varphi_\beta(t,p,\overline{i(n+2)+(n+1)},u_n) \wedge \varphi_\beta(t,p,\overline{(i+1)(n+2)},u') \wedge \cdots \\ & \cdots \wedge \varphi_\beta(t,p,\overline{(i+1)(n+2)+(n+1)},u'_n) \to \text{"}(u,u_0,\ldots,u_n) \xrightarrow{P} (u',u'_0,\ldots,u'_n)\text{"}\Big)\Big) \end{aligned}$$

■

Theorem 4.2.8. Th($\mathfrak{N}$) (commonly termed arithmetic) is not R-decidable.

Proof: We effectively assign to every register program P over $A = \{1\}$ an S_{ar}-sentence φ_P.

This φ_P is such that $\mathfrak{N} \models \varphi_P$ iff $P : \square \to \texttt{HALT}$.

Then $\mathrm{Th}(\mathfrak{N})$ will be undecidable, since $\Pi_{\texttt{HALT}}$ is undecidable.

As before, given P, we may compute its list of instructions $\alpha_0, \ldots, \alpha_k$ (for only α_k the HALT-instruction), and n the least number such that all registers by P used are among $R_0, \ldots, R_n$.

Using the χ_P-lemma, we have χ_P that describes how P operates, and we set

$$\varphi_P = \exists v_0 \ldots \exists v_n \chi_P(\underbrace{0, \ldots, 0}_{n+1 \text{ zeros}}, \overline{k}, v_0, \ldots, v_n)$$

Then we will have that

$$\begin{aligned} \mathfrak{N} \models \varphi_P \quad &\text{iff} \quad \mathfrak{N} \models \chi_P[0, \ldots, 0, k, m_0, \ldots, m_n] \text{ for some } m_0, \ldots, m_n \in \mathbb{N} \\ &\text{iff} \quad P \text{ beginning with the configuration } (0, \ldots, 0) \text{ after finitely many steps} \\ &\qquad \text{reaches configuration } (k, m_0, \ldots, m_n) \\ &\text{iff} \quad P : \square \to \texttt{HALT} \end{aligned}$$

This completes the proof. ■

Corollary 4.2.9. Arithmetic is neither R-axiomatizable nor R-enumerable. Therefore, with respect to a previous statement, $\Phi_{PA}^{\models} \subsetneq \mathrm{Th}(\mathfrak{N})$.

4.3 Representation

Theorem 4.3.1.

i. Given an n-ary decidable relation R over $\mathbb{N}$, there exists an S_{ar}-formula $\varphi(v_0, \ldots, v_{n-1})$ such that for all $\ell_0, \ldots, \ell_{n-1} \in \mathbb{N}$

$$R\ell_0 \ldots \ell_{n-1} \quad \text{iff} \quad \mathfrak{N} \models \varphi\left[\,\overline{\ell_0}, \ldots, \overline{\ell_{n-1}}\,\right]$$

ii. Given an n-ary computable function $f : \mathbb{N} \to \mathbb{N}$, there is an S_{ar}-formula $\varphi(v_0, \ldots, v_n)$ such that for all $(\ell_0, \ldots, \ell_n)$

$$f(\ell_0, \ldots, \ell_{n-1}) = \ell_n \quad \text{iff} \quad \mathfrak{N} \models \varphi\left[\,\overline{\ell_0}, \ldots \overline{\ell_n}\,\right]$$

Proof: The required functions are conjunctions of χ_P at each stage of a program P that decides R (and f). ■

Definition 4.3.2. Let $\Phi \subset L_0^{S_{ar}}$. An r-ary relation R on $\mathbb{N}$ is termed representable in Φ iff there is an S_{ar}-formula $\varphi(v_0, \ldots, v_{r-1})$ such that for all $n_0, \ldots n_{r-1} \in \mathbb{N}$

if $Rn_0 \ldots n_{r-1}$, then $\Phi \vdash \varphi[\,\overline{n_0}, \ldots, \overline{n_{r-1}}\,]$
if $\neg Rn_0 \ldots n_{r-1}$, then $\Phi \vdash \neg\varphi[\,\overline{n_0}, \ldots, \overline{n_{r-1}}\,]$.

In this case, we say that φ represents R in Φ.

Definition 4.3.3. An r-ary function f on $\mathbb{N}$ is termed representable in $\Phi \subset L_0^{S_{ar}}$ iff there is an S_{ar}-formula $\varphi(v_0, \ldots, v_r)$ such that for all $n_0, \ldots, n_r \in \mathbb{N}$, then

if $f(n_0, \ldots, n_{r-1}) = n_r$, then $\Phi \vdash \varphi[\,\overline{n_0}, \ldots, \overline{n_r}\,]$
if $f(n_0, \ldots, n_{r-1}) \neq n_r$, then $\Phi \vdash \neg\varphi[\,\overline{n_0}, \ldots, \overline{n_r}\,]$

In this case, we say that φ represents f in Φ.

Remark 4.3.4. If $\Phi = \mathrm{Th}(\mathfrak{N})$, then we call the set of representable functions and relations in Φ arithmetic.

Lemma 4.3.5.

i. If Φ is inconsistent, then every function and relation over $\mathbb{N}$ is representable in Φ.

ii. If $\Phi \subset \Phi' \subset L_0^{S_{ar}}$, then all functions and relations representable in Φ are representable in Φ'.

iii. Let Φ be consistent. If Φ is R-decidable, then every relation representable in Φ is R-decidable, and every function representable in Φ is R-computable.

Definition 4.3.6. Let $\phi \subset L_r^{S_{ar}}$. Then Φ allows representations if all R-decidable relations and all R-computable functions over $\mathbb{N}$ are representable in Φ.

Theorem 4.3.7. $\text{Th}(\mathfrak{N})$ allows representations.

Theorem 4.3.8. Φ_{PA} allows representations.

4.4 Incompleteness

Definition 4.4.1. Let S be a symbol set. If L^S is enumerable, then we define the Godel number of some S-formula φ to be the position that φ appears in in some numbering of L^S, and denote it by n_φ.

Theorem 4.4.2. [Fixed point theorem]
Suppose that Φ allows representations. Then for every $\psi \in L_1^{S_{ar}}$ there is a $\varphi \in L_0^{S_{ar}}$ such that $\Phi \vdash \varphi \leftrightarrow \psi(\overline{n_\varphi})$.

Proof: Suppose that Φ allows representations and $\psi \in L_1^{S_{ar}}$.

Define a computable function $F : \mathbb{N}^2 \to \mathbb{N}$ by

$$F(n,m) = \begin{cases} n_{\chi(\overline{m})} & \text{if } \ n = n_\chi \ \text{ for some } \ \chi \in L_1^{S_{ar}} \\ 0 & \text{else} \end{cases}$$

Thus we have that if $\chi \in L_1^{S_{ar}}$, then $F(n_\chi, m) = n_{\chi(\overline{m})}$.

Since Φ allows representations, there is an $\alpha \in L_3^{S_{ar}}$ such that for all $m, n, k \in \mathbb{N}$,

$F(n,m) = k \implies \Phi \vdash \alpha(\overline{n}, \overline{m}, \overline{k})$

$F(n,m) \neq k \implies \Phi \vdash \neg\alpha(\overline{n}, \overline{m}, \overline{k})$

Let $\beta(x) = \forall z(\alpha(x,x,z) \to \psi(z))$ and let $\varphi = \beta(\overline{n_\beta}) = \forall z(\alpha(\overline{n_\beta}, \overline{n_\beta}, z) \to \psi(z))$.

We claim that $\Phi \vdash \varphi \leftrightarrow \psi(\overline{n_\varphi})$.

Proof of claim: Note that $\beta \in L_1^{S_{ar}}$, so $F(n_\beta, n_\beta) = n_{\beta(\overline{n_\beta})}$.

However, $\beta(\overline{n_\beta}) = \varphi$, so $F(n_\beta, n_\beta) = n_\varphi$.

Thus $\Phi \vdash \alpha(\overline{n_\beta}, \overline{n_\beta}, \overline{n_\varphi})$. By definition of φ, we have $\Phi \cup \{\varphi\} \vdash \alpha(\overline{n_\beta}, \overline{n_\beta}, \overline{n_\varphi}) \to \psi(\overline{n_\varphi})$.

Therefore $\Phi \vdash \varphi \to \psi(\overline{n_\varphi})$.

By above, $\Phi \vdash \exists^{=1} z\alpha(\overline{n_\beta}, \overline{n_\beta}, z)$ and so $\Phi \vdash \forall z(\alpha(\overline{n_\beta}, \overline{n_\beta}, z) \to z = \overline{n_\varphi})$.

Thus $\Phi \vdash \psi(\overline{n_\varphi}) \to (\forall z(\alpha(\overline{n_\beta}, \overline{n_\beta}, z) \to \psi(z)))$.

Therefore $\Phi \vdash \psi(\overline{n_\varphi}) \to \varphi$. ■

Lemma 4.4.3. Suppose that Φ is consistent and allows representations. Then $\Phi^\vdash$ is not representable in Φ.

Proof: Suppose that $\chi(v)$ represents $\Phi^\vdash$ in Φ.

Then for any $n \in \mathbb{N}$,

$$n \in \Phi^\vdash \implies \Phi \vdash \chi(\overline{n})$$
$$n \notin \Phi^\vdash \implies \Phi \vdash \neg\chi(\overline{n})$$

In particular, if $\alpha \in L_1^{S_{ar}}$, then

$$\Phi \vdash \alpha \implies \Phi \vdash \chi(\overline{n_\alpha})$$
$$\Phi \nvdash \alpha \implies \Phi \vdash \neg\chi(\overline{n_\alpha})$$

Since Φ is consistent, we must have that $\Phi \not\vdash \alpha$ iff $\Phi \vdash \neg\chi(\overline{n_\alpha})$.
By the fixed point theorem applied to $\neg\chi$ and Φ, there exists $\varphi \in L_0^{S_{ar}}$ such that $\Phi \vdash \varphi \leftrightarrow \neg\chi(\overline{n_\varphi})$.
But then $\Phi \vdash \varphi$ iff $\neg\chi(\overline{n_\varphi})$ iff $\Phi \not\vdash \varphi$.
This is a contradiction.
Hence $\Phi^{\vdash}$ is not representable in Φ. ■

Theorem 4.4.4. [Tarski]
1. Suppose that Φ is consistent and allows representations. Then $\Phi^{\models}$ is not representable in Φ.
2. $\mathrm{Th}(\mathfrak{N})$ is not representable in $\mathrm{Th}(\mathfrak{N})$.

Proof: (**1.**) By completeness, $\Phi^{\vdash} = \Phi^{\models}$.

(**2.**) $\mathrm{Th}(\mathfrak{N})$ allows representations, and $\mathrm{Th}(\mathfrak{N})^{\vdash} = \mathrm{Th}(\mathfrak{N})$.
Apply the above theorem. ■

Consider $\Phi \subset L_0^{S_{ar}}$ decidable and allowing representations. Let us fix an enumeration of all S_{ar} derivations, i.e. all sequents in the derivation calculus of S_{ar}. Define a binary relation H by

$$Hnm \quad \Longleftrightarrow \quad \Big(\text{the } m\text{th derivation ends with a sequent } \psi_0 \dots \psi_{k-1}\varphi \text{ with } \psi_i \in \Phi \ \forall\, i \text{ and } n = n_\varphi\Big)$$

Since Φ is decidable, H is decidable, and $\Phi \vdash \varphi$ iff there is an $m \in \mathbb{N}$ such that $Hn_\varphi m$. Since Φ allows representations, there is some $\varphi_H(x, y) \in L_2^{S_{ar}}$ that represents H in Φ. Then we define

$$\mathrm{Der}_\Phi(x) := \exists y \varphi_H(x, y) \qquad\qquad \mathrm{Consis}_\Phi := \neg\mathrm{Der}_\Phi(\overline{n_{\neg 0 \equiv 0}})$$

With these formulae we may encode the derivability of a formula and the consistency of a set. They will be also used to prove the theorems below. So if x is the Godel number of some formula χ, then

$$\Big(\Phi \text{ derives } \chi\Big) \quad \text{iff} \quad \Big(\Phi \vdash \chi\Big) \quad \text{iff} \quad \Big(\Phi \vdash \mathrm{Der}_\Phi(x)\Big)$$

$$\Big(\Phi \text{ is consistent}\Big) \quad \text{iff} \quad \Big(\Phi \vdash \varphi \text{ iff not } \Phi \vdash \neg\varphi\Big) \quad \text{iff} \quad \Big(\Phi \vdash \mathrm{Consis}_\Phi\Big)$$

Theorem 4.4.5. [First incompleteness - Godel]
Suppose that Φ is consistent, R-decidable, and allows representations. Then there is an S_{ar}-sentence φ such that neither $\Phi \vdash \varphi$ nor $\Phi \vdash \neg\varphi$.

Proof: Assume no such φ exists.
Then $\Phi^{\vdash}$ is complete.
So $\Phi^{\vdash}$ is consistent and R-enumerable, hence R-decidable.
Since Φ allows representations, $\Phi^{\vdash}$ is not representable by Tarski.
This is a contradiction.
Hence such a φ exists. ■

· For the followng lemma, we choose $\neg\mathrm{Der}_\Phi(v_0) \in L_1^{S_{ar}}$, so then by the fixed point theroem we can find $\varphi \in L_0^{S_{ar}}$ such that $\Phi \vdash \varphi \leftrightarrow \neg\mathrm{Der}_\Phi(\overline{n_\varphi})$.

Lemma 4.4.6. If Φ is consistent, then not $\Phi \vdash \varphi$.

Proof: Suppose $\Phi \models \varphi$.
Let m be such that $Hn_\varphi m$.
Then $\Phi \vdash \varphi_H(\overline{n_\varphi}, m)$, so $\Phi \vdash \mathrm{Der}_\Phi(\overline{n_\varphi})$.
But $\Phi \vdash \varphi \leftrightarrow \neg\ \mathrm{Der}_\Phi(\overline{n_\varphi})$, so $\Phi \vdash \neg\ \mathrm{Der}_\Phi(\overline{n_\varphi})$.

Therefore Φ is inconsistent. ■

· It is technically tedious, but possible, to show that, with φ as above,

$$\Phi \vdash \text{Consis}_\Phi \rightarrow \neg \text{Der}_\Phi(\overline{n_\varphi})$$

Theorem 4.4.7. [Second incompleteness - Godel]
Suppose that $\Phi \supset \Phi_{PA}$ is consistent and R-decidable. Then not $\Phi \vdash \text{Consis}_\Phi$.

Proof: Suppose that $\Phi \vdash \text{Consis}_\Phi$.
Then $\Phi \vdash \neg \text{ Der}_\Phi(\overline{n_\varphi})$.
Since φ was a fixed point (i.e. $\Phi \vdash \varphi \leftrightarrow \neg \text{ Der}_\Phi(\overline{n_\varphi})$), we have that $\Phi \vdash \varphi$.
Then by the above lemma, Φ is inconsistent. ■

5 Elementary equivalence revisited

5.1 Partial and finite isomorphisms

Definition 5.1.1. Let $\mathfrak{A}$ and $\mathfrak{B}$ be S-structures. A map $p : A \to B$ is termed a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ if the following conditions are satisfied:
1. p is an injective homomorphism
2. for every n-ary $R \in S$ and $a_1, \dots, a_n \in A$, we have $R^\mathfrak{A} a_1 \dots a_n$ iff $R^\mathfrak{B} p(a_1) \dots p(a_n)$
3. for every n-ary $f \in S$ and $a, a_1, \dots, a_n \in A$, we have $f^\mathfrak{A}(a_1, \dots, a_n) = a$ iff $f^\mathfrak{B}(p(a_1), \dots, p(a_n)) = p(a)$
4. for $c \in S$ and $a \in \text{dom}(p)$, we have $c^\mathfrak{A} = a$ iff $c^\mathfrak{B} = p(a)$

The set of all such isomorphisms is denoted by

$$\text{Part}(\mathfrak{A}, \mathfrak{B}) := \{p \mid p : A \to B \text{ is a partial isomorphism from } \mathfrak{A} \text{ to } \mathfrak{B}\}$$

Note that the empty map, as well as any restriction of a (partial) isomorphism is a partial isomorphism.

Remark 5.1.2. If S is relational, then for $a_1, \dots, a_r \in A$ and $b_1, \dots, b_r \in B$, equivalently
1. By setting $p(a_i) = b_i$ the function p determines a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$
2. For every $\varphi \in L^S_r$ atomic, $\mathfrak{A} \models \varphi[a_1, \dots, a_r]$ iff $\mathfrak{B} \models \varphi[b_0, \dots, b_r]$

Proof: (**1.** $\Rightarrow$ **2.**) Suppose $R \in S$ is n-ary for $\{a_{i_1}, \dots, a_{i_n}\} \subset \{a_1, \dots, a_r\}$ and $R = Ra_{i_1} \dots a_{i_n}$, so

$$\begin{aligned} \mathfrak{A} \models R[a_1, \dots a_r] &\text{ iff } (a_{i_1}, \dots, a_{i_n}) \in R^\mathfrak{A} \\ &\text{ iff } (p(a_{i_1}), \dots, p(a_{i_n})) \in R^\mathfrak{B} \\ &\text{ iff } (b_{i_1}, \dots, b_{i_n}) \in R^\mathfrak{B} \\ &\text{ iff } \mathfrak{B} \models R[b_{i_1}, \dots, b_{i_n}] \end{aligned}$$

$$\begin{aligned} \mathfrak{A} \models v_i \equiv v_j[a_0, \dots, a_{r-1}] &\text{ iff } a_i = a_j \\ &\text{ iff } p(a_i) = p(a_j) \\ &\text{ iff } b_i = b_j \\ &\text{ iff } \mathfrak{B} \models v_i \equiv v_j[b_0, \dots, b_{r-1}] \end{aligned}$$

(**2.** $\Rightarrow$ **1.**) Here we use injectivity.
Consider $v_i = v_j$, so then

$$\begin{aligned} a_i = a_j &\text{ iff } \mathfrak{A} \models v_i \equiv v_j[a_0, \dots, a_{r-1}] \\ &\text{ iff } \mathfrak{B} \models v_i \equiv v_j[b_0, \dots, b_{r-1}] \\ &\text{ iff } b_i = b_j \end{aligned}$$

■

Definition 5.1.3. Given maps p, q, we say that q is an extension of p iff $\text{dom}(p) \subset \text{dom}(q)$ and $q|_{\text{dom}(p)} = p$. This relationship is expressed as $p \subset q$.

Definition 5.1.4. Two S-structures $\mathfrak{A}, \mathfrak{B}$ are termed finitely isomorphic iff there exists a sequence $(I_n)_{n=1}^{\infty}$ such that every I_n is a non-empty set of partial isomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$ satisfying

Forth-property: For every $p \in I_{n+1}$ and $a \in A$, there is $q \in I_n$ such that $p \subset q$ and $a \in \text{dom}(q)$
Back-property: For every $p \in I_{n+1}$ and $b \in B$, there is $q \in I_n$ such that $p \subset q$ and $b \in \text{range}(q)$

For such a sequence, we write $(I_n)_{n=1}^{\infty} : \mathfrak{A} \cong_f \mathfrak{B}$.

Definition 5.1.5. Two S-structures $\mathfrak{A}, \mathfrak{B}$ are termed partially isomorphic if there exists $I \subset \text{Part}(\mathfrak{A}, \mathfrak{B})$ non-empty such that
1. for all $a \in A$ and $p \in I$ there is $q \in I$ with $p \subset q$ and $a \in \text{dom}(q)$
2. for all $b \in B$ and $p \in I$ there is $q \in I$ with $p \subset q$ and $b \in \text{range}(q)$

This relationship is expressed as $\mathfrak{A} \cong_p \mathfrak{B}$.

Lemma 5.1.6.
1. If $\mathfrak{A} \cong \mathfrak{B}$, then $\mathfrak{A} \cong_p \mathfrak{B}$.
2. If $\mathfrak{A} \cong_p \mathfrak{B}$, then $\mathfrak{A} \cong_f \mathfrak{B}$.
3. If $\mathfrak{A} \cong_f \mathfrak{B}$ and A is finite, then $\mathfrak{A} \cong \mathfrak{B}$.
4. If $\mathfrak{A} \cong_p \mathfrak{B}$ and A, B are at most countable, then $\mathfrak{A} \cong \mathfrak{B}$.

Proof: (**1.**) If $\pi : \mathfrak{A} \cong \mathfrak{B}$, then $I : \mathfrak{A} \cong_p \mathfrak{B}$ for $I = \{\pi\}$.

(**2.**) If $I : \mathfrak{A} \cong_p \mathfrak{B}$, then $(I_n)_{n=1}^{\infty} : \mathfrak{A} \cong_f \mathfrak{B}$ for $I_n = I$ for all n.

(**3.**) Suppose that $(I_n)_{n=1}^{\infty} : \mathfrak{A} \cong_f \mathfrak{B}$ and $A = \{a_1, \ldots, a_r\}$.
Choose $p_0 \in I_{r+1}$.
Then for $0 \leqslant i \leqslant r$, given $p_i \in I_{r+1-i}$, choose $p_{i+1} \in I_{r-i}$ such that $p_i \subset p_{i+1}$ and $a_{i+1} \in \text{dom}(p_{i+1})$.
Now $p_r \in I_1$ is a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ with $\text{dom}(p_r) = A$.
So to show $p_r : A \to B$, it suffices to show $\text{range}(p_r) = B$.
Suppose there exists $b \in B$ with $b \notin \text{range}(p_r)$.
Then there exists $p_{r+1} \in I_1$ with $b \in \text{range}(p_{r+1})$.
This is a contradiction, as $\text{dom}(A) = A$ and p_{r+1} is injective.

(**4.**) If A or B are finite, the result follows from (**2.**) and (**3.**).
So suppose that $A = \{a_0, a_1, \ldots\}$ and $B = \{b_0, b_1, \ldots\}$.
Choose $p_0 \in I_1$.
For $i = 2r + 1$, choose $p_i \in I$ with $p_{i-1} \subset p_i$ and $a_r \in \text{dom}(p_i)$.
For $i = 2r + 2$, choose $p_i \in I$ with $p_{i-1} \subset p_i$ and $b_r \in \text{range}(p_i)$.
Then $p = \bigcup_{n=1}^{\infty} p_n$ is an isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$. ■

5.2 Dense orderings

Definition 5.2.1. A dense ordering is a set of formulae Φ that satisfy the following sentences.

$$\begin{aligned}
&\forall x \neg x < x \\
&\forall x \forall y \forall z ((x < y \wedge y < z) \to x < z) \\
&\forall x \forall y (x < y \vee x \equiv y \vee y < x) \\
&\forall x \forall y (x < y \to \exists z (x < z \wedge z < y)) \\
&\forall x \exists y x < y \\
&\forall x \exists y y < x
\end{aligned}$$

This set of sentences is denoted by Φ_{dord}.

Theorem 5.2.2. Any two countable dense orderings without endpoints are isomorphic. That is, a dense ordering without endpoints is a model of Φ_{dord}.

Proof: By the previous lemma, it suffices to show that any two countable dense linear orderings are partially isomorphic.

Set $\mathfrak{A} = (A, <^A)$ and $\mathfrak{B} = (B, <^B)$ be countable dense linear orderings.

Claim: $I : \mathfrak{A} \cong_p \mathfrak{B}$ for $I = \{p \mid p \in \text{part}(\mathfrak{A}, \mathfrak{B}) \text{ and } \text{dom}(p) \text{ is finite}\}$.

Proof of claim: Since $p = \varnothing \in I$, $I \neq \varnothing$.

First we check that it satisfies the forth property.

For $p \in I$, suppose $\text{dom}(p) = \{a_1, \ldots, a_n\}$.

Note that $\mathfrak{A}$ puts an order on $a_1, \ldots, a_n$, which is equivalent to the ordering that $\mathfrak{B}$ puts on $p(a_1), \ldots, p(a_n)$.

So for $a \in A$, $\mathfrak{A}$ determines where a is relative to $a_1, \ldots, a_n$,

Since $\mathfrak{B}$ is dense, there is some $b \in B$ with the same position, but with respect to $p(a_1), \ldots, p(a_n)$.

So $p \cup \{(a, b)\}$ is a finito partial isomorphism extending p with a in its domain.

The back property is proved similarly.

Since we have a partial isomorphism, we have an isomorphism. ■

Definition 5.2.3. A successor ordering is a set of formulae Φ that satisfy the following sentences.

$$\begin{array}{l} \forall x(\neg x \equiv 0 \leftrightarrow \exists y \sigma y \equiv x) \\ \forall x \forall y(\sigma x \equiv \sigma y \to x \equiv y) \\ \forall x \neg \sigma x \equiv x \\ \forall x \neg \sigma\sigma x \equiv x \\ \forall x \neg \sigma\sigma\sigma x \equiv x \\ \quad \vdots \end{array}$$

This set of sentences is denoted by Φ_σ, where σ is the successor function. For shorthand notation, for $a \in A$ of $\mathfrak{A}$ a successor structure, we let

$$a^{(n)} := \underbrace{\sigma^A \cdots \sigma^A}_{n \text{ times}} a$$

Proposition 5.2.4. Any two models of Φ_σ are finitely isomorphic.

Proof: For every $n \in \mathbb{N}$, define a function d_n by

$$\begin{array}{rccl} d_n : & A \times A & \to & \mathbb{N} \cup \{0\} \\ & (a, a') & \mapsto & \begin{cases} m & \text{if } a^{(m)} \equiv a' \text{ and } m \leqslant 2^n \\ -m & \text{if } a'^{(m)} \equiv a \text{ and } m \leqslant 2^n \\ \infty & \text{else} \end{cases} \end{array}$$

Suppose that $\mathfrak{A}$ and $\mathfrak{B}$ are models of Φ.

We will show that $(I_n)_{n=1}^\infty : \mathfrak{A} \cong_f \mathfrak{B}$ for

$$I_n = \{p \in \text{part}(\mathfrak{A}, \mathfrak{B}) \mid |\text{dom}(p)| < \infty,\ 0^A \in \text{dom}(p),\ d_n(a, a') = d_n(p(a), p(a'))\ \forall\ a, a' \in \ \text{dom}(p)\}$$

We note that $I_n \neq \varnothing$, as $(0^A, 0^B) \in I_n$.

Forth property: Suppose $p \in I_{n+1}$ and $a \in A$.

Case 1: There is an $a' \in \text{dom}(p)$ such that $d_n(a, a') \leqslant 2^n$.

In this case, choose $b \in B$ such that $d_n(p(a'), b) = d_n(a', a)$.

Let $q = p \cup (a, b)$.
Since $p \in I_{n+1}$, q is an isomorphism preserving distances.

Case 2: There is no such a'.
Choose b such that $d_n(p(a'), b) = \infty$ for all $a' \in \mathrm{dom}(p)$.
Let $q = p \cup (a, b)$.

The back prorperty is done in a symmetrical fashion. ■

Lemma 5.2.5. For a theory $T \subset L_0^S$, the following are equivalent.
1. T is complete
2. Any two models of T are elementarily equivalent.

Proof: (**1.⇒2.**) Let $\mathfrak{A}, \mathfrak{B}$ be models of T with $\varphi \in L_0^S$.
Then either $\varphi \in T$ or $\neg\varphi \in T$.
If $\varphi \in T$, then $\mathfrak{A} \models \varphi$ and $\mathfrak{B} \models \varphi$, or else $\mathfrak{A} \not\models \varphi$ and $\mathfrak{B} \not\models \varphi$.
Therefore $\mathfrak{A} \equiv \mathfrak{B}$.

(**2.⇒1.**) Let $\varphi \in L_0^S$ and suppose $\mathfrak{A} \models T$.
If $\mathfrak{A} \models \varphi$, then $\mathfrak{B} \models \varphi$ for all models $\mathfrak{B}$ of T, and so $\varphi \in T$.
If $\mathfrak{A} \not\models \varphi$, then $\mathfrak{A} \models \neg\varphi$ and $\mathfrak{B} \models \neg\varphi$ for all models $\mathfrak{B}$ of T, and so $\neg\varphi \in T$.
Therefore T is complete. ■

Proposition 5.2.6.
1. The theory $\Phi_{dord}^{\models}$ of dense orderings is complete and R-decidable.
2. The theory $\Phi_{\sigma}^{\models}$ of successor structures is complete and R-decidable.

Definition 5.2.7. For a formula φ, define the quantifier rank to be a function that enumerates the makimum number of nested quantifiers in φ.

$$\begin{aligned} \mathrm{qr}(\varphi) &:= 0 \quad \text{if } \varphi \text{ is atomic} \\ \mathrm{qr}(\neg\varphi) &:= \mathrm{qr}(\varphi) \\ \mathrm{qr}(\varphi \vee \psi) &:= \max\{\mathrm{qr}(\varphi), \mathrm{qr}(\psi)\} \\ \mathrm{qr}(\exists x\varphi) &:= \mathrm{qr}(\varphi) + 1 \end{aligned}$$

Lemma 5.2.8. Let $(I_n)_{n=1}^{\infty} : \mathfrak{A} \cong_f \mathfrak{B}$. Then for every formula φ, if $\varphi \in L_r^S$ and $\mathrm{qr}(\varphi) \leqslant n$ with $p \in I_n$ so that $a_0, \ldots, a_{r-1} \in \mathrm{dom}(p)$, then $\mathfrak{A} \models \varphi[a_0, \ldots a_{r-1}]$ iff $\mathfrak{B} \models \varphi[p(a_0), \ldots, p(a_{r-1})]$.

Proof: This will be done by induction on formulae.

(**i.**) For φ atomic, this is a restatement of a remark proved earlier.

(**ii.**) If $\varphi = \neg\psi$ for $\varphi \in L_r^S$ with $\mathrm{qr}(\varphi) \leqslant n$, and the result holds for ψ and $p \in I_n$ with $a_0, \ldots, a_{r-1} \in \mathrm{dom}(p)$, then

$$\begin{aligned} \mathfrak{A} \models \varphi[a_0, \ldots, a_{r-1}] \;\; &\text{iff} \;\; \mathfrak{A} \not\models \psi[a_0, \ldots, a_{r-1}] \\ &\text{iff} \;\; \mathfrak{B} \not\models \psi[p(a_0), \ldots, p(a_{r-1})] \\ &\text{iff} \;\; \mathfrak{B} \models \varphi[p(a_0), \ldots, p(a_{r-1})] \end{aligned}$$

(**iii.**) If $\varphi = \psi_0 \vee \psi_1$, then $\mathrm{qr}(\psi_0), \mathrm{qr}(\psi_1) \leqslant \mathrm{qr}(\varphi) \leqslant n$.
The rest of this part is straightforward.

(**iv.**) Suppose $\varphi = \exists x\psi$ and $\varphi \in L_r^S$ with $\mathrm{qr}(\varphi) \leqslant n$, and the result holds for ψ and $p \in I_n$ with $a_0, \ldots, a_{r-1} \in \mathrm{dom}(p)$.

By the coincidence lemma, we may assume WLOG that $\varphi = \exists v_r \psi$.
Now note that $\mathrm{qr}(\psi) \leqslant n-1$, so then

$$\begin{aligned}
\mathfrak{A} \models \varphi[a_0, \dots, a_{r-1}] &\text{ iff } \exists a \in A \text{ such that } \mathfrak{A} \models \psi[a_0, \dots, a_{r-1}, a] \\
&\text{ iff } \exists a \in A,\ q \in I_{n-1},\ q \supset p,\ a \in \mathrm{dom}(q),\ \mathfrak{A} \models \psi[a_0, \dots, a_{r-1}, a] \\
&\text{ iff } \exists a \in A,\ q \in I_{n-1},\ q \supset p,\ \mathfrak{B} \models \psi[p(a_0), \dots, p(a_{r-1}), q(a)] \\
&\text{ iff } \exists b \in B,\ q \in I_{n-1},\ b \in \mathrm{range}(q),\ \mathfrak{B} \models \psi[p(a_0), \dots, p(a_{r-1}), b] \\
&\text{ iff } \mathfrak{B} \models \varphi[p(a_0), \dots, p(a_{r-1})]
\end{aligned}$$

■

Definition 5.2.9. For a symbol set S, define $\Phi_r := \{\varphi \in L_r^S \mid \varphi \text{ is atomic or negated atomic}\}$. This set is finite for all r.

Definition 5.2.10. We introduce some notation to help out with the proof of Fraisse's theorem.

- For an r-tuple $(a_0, \dots, a_{r-1}) \in A^r$, we write $\overset{r}{a}$.
- Let $\mathfrak{A}, \mathfrak{B}$ be S-structures with $\overset{r}{a} \in A^r$ and $\overset{r}{b} \in B^r$. Then we write

$$\overset{r}{a} \to \overset{r}{b} \in \mathrm{Part}(\mathfrak{A}, \mathfrak{B}) \text{ iff } p(a_i) = b_i \text{ for } i \leqslant r \text{ defines a partial isomorphism from } \mathfrak{A} \text{ to } \mathfrak{B}$$

- Define formulae $\varphi^n_{\mathfrak{B}, \overset{r}{b}} \in L_r^S$ such that

$$\mathfrak{B} \models \varphi^n_{\mathfrak{B}, \overset{r}{b}}[\overset{r}{b}] \text{ and if } \mathfrak{A} \models \varphi^n_{\mathfrak{B}, \overset{r}{b}}[\overset{r}{a}]$$
then
$$\overset{r}{a} \to \overset{r}{b} \in \mathrm{Part}(\mathfrak{A}, \mathfrak{B}) \text{ which may be extended back and forth } n \text{ times}$$

These formulae are formally defined by induction on n as below, given $\mathfrak{B}$. The above is shown in the proof of Fraisse. We again use shorthand below, by letting $\overset{r}{b}\, b = (b_0, \dots, b_{r-1}, b)$.

$$\begin{aligned}
\varphi^0_{\mathfrak{B}, \overset{r}{b}} &= \bigwedge \{\varphi \in \Phi_r \mid \mathfrak{B} \models \varphi[\overset{r}{b}]\} \\
\varphi^{n+1}_{\mathfrak{B}, \overset{r}{b}} &= \forall v_r \bigvee \{\varphi^n_{\mathfrak{B}, \overset{r}{b} b} \mid b \in B\} \wedge \bigwedge \{\exists v_r \varphi^n_{\mathfrak{B}, \overset{r}{b} b} \mid b \in B\}
\end{aligned}$$

Since each Φ_r is finite, it follows by induction on n that the following set is finite.

$$\left\{ \varphi^n_{\mathfrak{B}, \overset{r}{b}} \text{ is an } S\text{-sentence and } \overset{r}{b} \in B \right\}$$

Thus the conjunctions and disjunctions are finite, so $\varphi^n_{\mathfrak{B}, \overset{r}{b}} \in L_r^S$.

Lemma 5.2.11.

i. $\varphi^n_{\mathfrak{B}, \overset{r}{b}} \in L_r^S$ and $\mathrm{qr}\left(\varphi^n_{\mathfrak{B}, \overset{r}{b}}\right) = n$

ii. $\mathfrak{B} \models \varphi^n_{\mathfrak{B}, \overset{r}{b}}[\overset{r}{b}]$

Proof: (**i.**) This is clear by induction on n.

(**ii.**) For $n = 0$, this is immediate.
Suppose this holds for n and for all r.
Then for all $\overset{r}{b}, b' \in B$, we have that $\mathfrak{B} \models \varphi^n_{\mathfrak{B}, \overset{r}{b} b'}[\overset{r}{b}, b']$.

So for all $b' \in B$, $\mathfrak{B} \models \bigvee\{\varphi^n_{\mathfrak{B},\overset{r}{b}b} \mid b \in B\}[\overset{r}{b}, b']$ and $\mathfrak{B} \models \exists v_r \varphi^n_{\mathfrak{B},\overset{r}{b}b'}[\overset{r}{b}]$.

So $\mathfrak{B} \models \forall v_r \bigvee\{\varphi^n_{\mathfrak{B},\overset{r}{b}b} \mid b \in B\}[\overset{r}{b}]$ and $\mathfrak{B} \models \bigwedge\{\exists v_r \varphi^n_{\mathfrak{B},\overset{r}{b}b'} \mid b \in B\}[\overset{r}{b}]$.

Therefore $\mathfrak{B} \models \varphi^{n+1}_{\mathfrak{B},\overset{r}{b}}[\overset{r}{b}]$. ■

Theorem 5.2.12. [Fraisse]
Let S be a finite symbol set and $\mathfrak{A}, \mathfrak{B}$ be S-structures. Then $\mathfrak{A} \equiv \mathfrak{B}$ iff $\mathfrak{A} \cong_f \mathfrak{B}$.

<u>*Proof:*</u> By a previous theorem, it suffices to prove the satement for relational symbol sets.

($\Leftarrow$) From the above lemma, if $\mathfrak{A} \cong_f \mathfrak{B}$, then for all $\varphi \in L_0^S$, $\mathfrak{A} \models \varphi$ iff $\mathfrak{B} \models \varphi$.
Therefore $\mathfrak{A} \equiv \mathfrak{B}$.

($\Rightarrow$) Let $\mathfrak{A}$ be an S-structure such that $\mathfrak{A} \equiv \mathfrak{B}$.

<u>Claim</u>: If $\mathfrak{A} \models \varphi^n_{\mathfrak{B},\overset{r}{b}}[\overset{r}{a}]$, then $\overset{r}{a} \to \overset{r}{b} \in \text{Part}(\mathfrak{A}, \mathfrak{B})$.

We prove this claim by induction on n.
Suppose that $\mathfrak{A} \models \varphi^0_{\mathfrak{B},\overset{r}{b}}[\overset{r}{a}]$.

Then for every atomic $\psi \in L_r^S$, $\mathfrak{A} \models \psi[\overset{r}{a}]$ iff $\mathfrak{B} \models \psi[\overset{r}{b}]$.
Then $\overset{r}{a} \to \overset{r}{b} \in \text{Part}(\mathfrak{A}, \mathfrak{B})$ by the old remark.
Suppose the result holds for $n \geqslant 1$ and $\mathfrak{A} \models \varphi^{n+1}_{\mathfrak{B},\overset{r}{b}}[\overset{r}{a}]$.
Fix any $a \in A$.
Since $\mathfrak{A} \models \forall v_r \bigvee\{\varphi^n_{\mathfrak{B},\overset{r}{b}} \mid b \in B\}[\overset{r}{a}]$, there is $b \in B$ such that $\mathfrak{A} \models \varphi^n_{\mathfrak{B},\overset{r}{b}b}[\overset{r}{a}, a]$.

Then by the induction hypothesis, $\overset{r}{a}\, a \to \overset{r}{b}\, b \in \text{Part}(\mathfrak{A}, \mathfrak{B})$, and so $\overset{r}{a} \to \overset{r}{b} \in \text{Part}(\mathfrak{A}, \mathfrak{B})$.

For S-structures $\mathfrak{A}, \mathfrak{B}$ and $n \in \mathbb{N}$, let

$$J_n := \left\{ \overset{r}{a} \to \overset{r}{b} \;\middle|\; r \in \mathbb{N},\ \overset{r}{a} \in A^r,\ \overset{r}{b} \in B^r,\ \mathfrak{A} \models \varphi^n_{\mathfrak{B},\overset{r}{b}}[\overset{r}{a}] \right\}$$

Then we claim that:
(a) $J_n \subset \text{Part}(\mathfrak{A}, \mathfrak{B})$
(b) $(J_n)_{n \in \mathbb{N}}$ has back and forth properties
(c) if $n > 0$ and $\mathfrak{A} \models \varphi^n_{\mathfrak{B}} \left(= \varphi^n_{\mathfrak{B},\overset{0}{b}} \right)$, then $\varnothing \in J_n$, hence $J_n \neq \varnothing$.

For (a), this was the previous claim.

For (b), let us first check the forth property.
Suppose that $p = \overset{r}{a} \to \overset{r}{b} \in J_{n+1}$ and $a \in A$.
Then $\mathfrak{A} \models \varphi^{n+1}_{\mathfrak{B},\overset{r}{b}}[\overset{r}{a}]$, so $\mathfrak{A} \models \forall v_r \bigvee\{\varphi^N_{\mathfrak{B},\overset{r}{b}b} \mid b \in B\}[\overset{r}{a}]$.
So there is some $b \in B$ such that $\mathfrak{A} \models \varphi^n_{\mathfrak{B},\overset{r}{b}b}[\overset{r}{a}, a]$.
So $\overset{r}{a}\, a \to \overset{r}{b}\, b \in J_n$ and extends p to a.

Now let us check the back property.
Suppose that $p = \overset{r}{a} \to \overset{r}{b} \in J_{n+1}$ and $b \in B$.
Since $\mathfrak{A} \models \bigwedge\{\exists v_r \varphi^n_{\mathfrak{B},\overset{r}{b}b} \mid b \in B\}[\overset{r}{a}]$, there is $a \in A$ such that $\mathfrak{A} \models \varphi^n_{\mathfrak{B},\overset{r}{b}b}[\overset{r}{a}\, a]$.
That is, $\overset{r}{a}\, a \to \overset{r}{b}\, b \in J_n$ with b in its range.

For (c), suppose that $\mathfrak{A} \equiv \mathfrak{B}$.
If $n > 0$, then $\mathfrak{B} \models \varphi^n_{\mathfrak{B}}$, so as $\mathfrak{A} \models \varphi^n_{\mathfrak{B}}$, clearly $J_n \neq \varnothing$.

This proves the claims.

Therefore $(J_n)_{n\in\mathbb{N}} : \mathfrak{A} \cong_f \mathfrak{B}$. ■

Fraisse's theorem implies that any two dense linear orderings are elementarily equivalent S-structures.

6 Computability

6.1 Turing machines

Definition 6.1.1. A Turing machine is a finite program with finitely many states that has access to a read-only (oracle) and a read-write (work) infinite tape.

Definition 6.1.2. A Turing program is a finite list of instructions of the form

$$q_i X Y q_j Z D_1 D_2$$

where q_i, q_j are states, $X, Y, Z \in \{0, 1\}$ and $D_1, D_2 \in \{L, R\}$.

Example 6.1.3. Suppose that a Turing machine is in state q_i and is reading X on the oracle tape and Y on the work tape, and if $q_i X Y q_j Z D_1 D_2$ is an instruction in the program, then the following programs add 1.

$$q_1 01 q_2 1RL$$
$$q_2 00 q_0 1RL$$

Proposition 6.1.4. We can effectively list all the Turing programs.

Let $P_0, P_1, \ldots$ be such a list. To each program P_i we associate a partial function φ_i as follows:

· If P_i started with $n+1$ 1's on the work tape, nothing on the oracle tape, with the work tape reading head at the left-most 1 and in state q_1, eventually reaches a halting state q_0, then we write $\varphi_i(n)\downarrow$, and let $\varphi_i(n)$ be the number of 1's on the work tape.

· If P_i started on input n and never halts, we write $\varphi_i(n)\uparrow$.

Definition 6.1.5. A set $A \subset \mathbb{N}$ is termed computable iff there is $i \in \mathbb{N}$ such that $\chi_A = \varphi_i$, where χ is the traditional characteristic function.

Definition 6.1.6. A set $A \subset \mathbb{N}$ is termed computably enumerable iff there is $i \in \mathbb{N}$ such that $W_i = \mathrm{dom}(\varphi_i) = \{n \mid \varphi_i(n)\downarrow\} = A$.

· Now we have $W_0, W_1, \ldots$ as an effective listing of all uniqe c.e. sets.

Definition 6.1.7. A function $f : \mathbb{N} \to \mathbb{N}$ is termed partial computable iff there is $i \in \mathbb{N}$ such that $f = \varphi_i$.

· f is computable iff it is partial computable and $\mathrm{dom}(\varphi_i) = \mathbb{N}$ for the same i

· f is total iff it is defined for all input values

So as to alleviate tedious proofs, we accept Church's thesis for Turing machines.

Definition 6.1.8. For $s, x, y \in \mathbb{N}$, we write $\varphi_{e,s}(x) = y$ (and $\varphi_{e,s}(x)\downarrow$) iff program P_e started with input x and empty oracle tape, halts within s steps and outputs y. If after s steps this program has not halted, we write $\varphi_{e,s}(x)\uparrow$.

Definition 6.1.9. Define the standard pairing function (which is injective) by

$$\begin{array}{rrcl} \langle\ ,\ \rangle : & \mathbb{N}\times\mathbb{N} & \to & \mathbb{N} \\ & (x,y) & \mapsto & \frac{1}{2}(x^2+2xy+y^2+3x+y) \end{array}$$

Then a binary relation R is termed computable iff $\{\langle x,y\rangle \mid (x,y)\in R\}$ is computable.

Definition 6.1.10. We write that $A \subset \Sigma_1$ (and say "A is Σ_1") iff there is a computable relation $R(x, y)$ such that for all $k \in \mathbb{N}$, $x \in A$ iff there is $y \in \mathbb{N}$ such that $R(x, y)$.

Theorem 6.1.11. A set A is c.e. iff A is Σ_1.

Proof: ($\Rightarrow$) If $A = W_e$, then $x \in A$ iff $x \in W_e$ iff there is s such that $x \in W_{e,s}$.
So A is Σ_1.

($\Leftarrow$) If A is Σ_1, then there is a computable relation $R(x, y)$ such that $x \in A$ iff there exists y such that $R(x, y)$,

Consider the program P that on input x asks, for each $y \in \mathbb{N}$ in turn, whether $(x, y) \in R$ and halts with output y for the first y with affirmative response.

Since R is computable, by Church's thesis there is an index e such that $P = P_e$, so $A = W_e$.

That is, A is c.e. ∎

Theorem 6.1.12. A non-empty set A is c.e. iff it is the range of a computable function.

Proof: ($\Leftarrow$) Suppose $A = \text{range}(f)$ for f computable.
Then $n \in A$ iff there is an x such that $f(x) = n$, so A is Σ_1, and hence c.e.

($\Rightarrow$) Suppose $A = W_e$ is non-epmty.
For $a \in A$, define $f(\langle x, s\rangle) = \left\{\begin{smallmatrix} x & x \in W_{e,s} \\ a & \text{else} \end{smallmatrix}\right.$
Then f is computable and has range A. ∎

Theorem 6.1.13. There is no effective listing of the computable functions.

Proof: Suppose that $f_0, f_1, \ldots$ is an effective listing.
Then $g(n) = f_n(n) + 1$ would be computable.
But $g \neq f_n$ for any n, so such a list cannot exist. ∎

Definition 6.1.14. Define the following sets:

$$K = \{e \mid \varphi_e(e)\downarrow\}$$
$$K_0 = \{\langle e, n\rangle \mid \varphi_e(n)\downarrow\}$$

Theorem 6.1.15. K is not computable

Proof: If K were computable, so would $g(x) = \left\{\begin{smallmatrix} \varphi_x(x)+1 & x \in K \\ 0 & \text{else} \end{smallmatrix}\right.$
So $g = \varphi_e$ for some e, and g is total.
Then $\varphi_e(e)\downarrow$, so $\varphi_e(e) = g(e) = \varphi_e(e) + 1$, a contradiction. ∎

Corollary 6.1.16. K_0 is not computable.

Definition 6.1.17. For sets A, B, we write $A \leqslant_m B$ (and say "A is many-one reducable to B") iff there is a computable function f such that $x \in A$ iff $f(x) \in B$. In the case where such a function f is injective, we write $A \leqslant_1 B$ (and say "A is one-reducable to B).

Therefore we have that $K \leqslant_m K_0$.

Theorem 6.1.18.
1. If $A \leqslant_m B$ and B is computable, then A is computable.
2. If $A \leqslant_m B$ and B is c.e., then A is c.e.

Proof: Suppose that $A \leqslant_m B$ via a function f.

(**1.**) Suppose that B is computable. To compute whether $x \in A$, first compute $f(x)$, then compute whether $f(x) \in B$.

(**2.**) If B is c.e., then $B = W_e$ for some e.
So $x \in A$ iff $f(x) \in W_e$ iff there is s such that $f(x) \in W_{e,s}$.
So B is Σ_1, therefore c.e. ■

Theorem 6.1.19. [$s - m - n$ THEOREM]
If $\Psi(x, y)$ is a partial computable function on two variables, then there exists an injective function f such that $\Psi(x, y) = \varphi_{f(x)}(y)$

· This theorem shows that $K_0 \leqslant_m K$.

Definition 6.1.20. The sets K and K_0 are termed complete, that is, they are able to uniformly compute any c.e. set.

6.2 Turing reducibility

Note that if A is a non-computable c.e. set, then $\overline{A} \nleqslant_m A$, which complicates things. Turing reducibility circumvents this difficulty.

Definition 6.2.1. For sets A, B, we write $A \leqslant_T B$ iff there is a Turing program P_e such that if B is on the oracle tape and P_e started on input n (i.e. $n + 1$ on work tape) and halts after finitely many steps with

1 on work tape if $n \in A$
0 on tape if $n \notin A$

Then we write $\Phi_e^B = A$.

Remark 6.2.2. If program P_e with oracle B started with input x and halts after s steps with y on the work tape, then we write $\Phi_{e,s}^B(x) = y$ (and $\Phi_e^B(x) \downarrow$). Therefore if $\Phi_e^B(x) = y$ then there is some finite segment (convex set) $\sigma \subset B$ such that $\Phi_e^\sigma(x) \downarrow$ also.

Definition 6.2.3. For a set $A \subset \mathbb{N}$, define the jump of A by

$$A' := \{x \mid \Phi_x^A(x) \downarrow\}$$

We say that a y is A'-computable iff $y \in A'$, or equivalently, that A computes y.

Proposition 6.2.4. [PROPERTIES OF THE JUMP]
1. A' is c.e. in A
2. $A <_T A'$
3. If B is c.e. in A, then $B \subset A'$
4. If $B \leqslant_T A$, then $B' \leqslant_T A'$

Definition 6.2.5. If $\varnothing \leqslant_T A \leqslant \varnothing''$ and $A' \equiv_T \varnothing'$, then we say that A is low. If $A \leqslant_T \varnothing'$ and $A' \equiv_T \varnothing''$, then we say that A is high.

Remark 6.2.6. Note that all computable sets are low. Also, if $A \equiv_T \varnothing'$, then A is high.

6.3 Special non-computable sets

First we wish to construct a low set that is not computable. We will build this set A in stages by finite binary strings α_s, and ultimately $A = \bigcup_s \{\alpha\}$.

At each stage $s+1$ we wil have $\alpha_{s+1} \supset \alpha_s$. Then A will not be computable, but will be $\varnothing'$-computable - to compute whether $x \in A$, we will run the construction using an $\varnothing$-oracle until a stage s for which $x \in \mathrm{dom}(\alpha_s)$, so then $x \in A$ iff $\alpha_s(x) = 1$.

As we build A, we must meet for each $e \in \mathbb{N}$ the requirement $R_e : A \neq \varphi_e$, which will ensure that A is not computable - it will be met at stage $2e+2$ of the construction. And in order to make A low, we will ensure that at stage $2e+1$, it will be decided whether or not $\Phi_e^A(e) \downarrow$. Since the construction will be $\varnothing'$ computable, this will ensure that $A' \leqslant_T \varnothing$.

Theorem 6.3.1. There exists a low set A that is not computable.

Proof: Construct the set A in the following manner:

Stage 0: Let $\alpha_0 = \square$.

Stage $s+1 = 2e+1$: Given α_s, put to the oracle the question $\exists\sigma\exists t \left(\sigma \supset \alpha_s \wedge \Phi_{e,t}^{\sigma}(e)\downarrow\right)$.
As it is a Σ_1-question, we can effectively find the appropriate location to check the $\varnothing'$-oracle.
If we find 1, set $\alpha_{s+1} = \sigma$.
If we find 0, set $\alpha_{s+1} = \alpha_s \frown 0$, where $\frown$ indicates string concatenation.

Stage $s+1 = 2e+2$: For $n = |\alpha_s|$, put to the oracle the question $\exists t\, (\varphi_{e,t}(n)\downarrow \wedge \varphi_{e,t}(n) \equiv 0)$.
Similarly to above, we can effectively find the appropriate location to check the $\varnothing'$-oracle.
If we find 1, set $\alpha_{s+1} = \alpha_s \frown 1$.
If we find 0, set $\alpha_{s+1} = \alpha_s \frown 0$.

Let $A = \bigcup_s \{\alpha_s\}$.
Since the construction is $\varnothing'$-computable, we have that $A \leqslant_T \varnothing$.

The set A is low because $\varnothing'$ computes at stage $s+1 = 2e+1$ if $e \in A'$, i.e. if $\Phi_e^A(e)\downarrow$.
If the answer to $\exists\sigma\exists t \left(\sigma \supset \alpha_s \wedge \Phi_{e,t}^{\sigma}(e)\downarrow\right)$ was "yes", then $e \in A'$, since $\Phi_{e,t}^{\alpha_{s+1}}(e)\downarrow$, and $\alpha_{s+1} \in A'$.
If the answer was "no", then $e \notin A'$.
Indeed, $e \in A'$ implies there exists $\tau \subset A$ and $t \in A$ such that $\Phi_{e,t}^{\tau}(e)\downarrow$.
Let σ be such that $\sigma \supset \tau, \alpha_s$, then this σ and t would show that the answer would have been "yes".

The set A is not computable.
Assume for contradiction that $A = \varphi_e$ for some e, and consider step $s+1 = 2e+2$ with $n = |\alpha_s|$.
If $\varphi_e(n) = 0$, then there exists t such that $\varphi_{e,t}(n) = 0$, so $A(n) = \alpha_{s+1}(n) = 1 \neq 0$.
If $\varphi_e(n) = 1$, then it is not the case that there exists t such that $\varphi_{e,t}(n) = 0$, so $A(n) = \alpha_{s+1} = 0 \neq 1$.
So $\varphi_e \neq A$. ■

Definition 6.3.2. Given a set $X \subset \mathbb{N}$ and $n \in \mathbb{N}$, define the following set:

$$X \upharpoonright n := \{x \in X \mid x < n\}$$

Lemma 6.3.3. [Limit lemma]
A total function $g : \mathbb{N} \to \mathbb{N}$ is $\varnothing'$-computable iff there exists a computable function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that for all $x \in \mathbb{N}$, $g(x) = \lim_{s\to\infty} [f(x,s)]$.

Proof: ($\Leftarrow$) Suppose that $g(x) = \lim_{s\to\infty} [f(x,s)]$ for f computable.
Then $\varnothing'$ can compute $g(x)$ as follows:

For each s, put to $\varnothing'$ the question $\exists t(t > s \wedge f(x,t) \neq f(x,s))$.
Since $g(x) = \lim_{s\to\infty}[f(x,s)]$, there must be some s for which the answer is "no".
So after finitely many steps, $\varnothing'$ will find such an s, and know that $g(x) = f(x,s)$.

($\Rightarrow$) Suppose that $g \leqslant_T \varnothing'$.
Then $g = \Phi_e^K$ for some e.
Let $\{K_s\}_{s\in W}$ be an enumeration of K.
Define a function f by

$$f(x,s) = \begin{cases} \Phi_{e,s}^{K_s}(x) & \text{if it is defined} \\ 0 & \text{else} \end{cases}$$

Note that $\Phi_{e,s}^{K_s}(x)$ is computable, as it is bounded by s steps.
Since $g(x) = \Phi_e^K(x)$, there is some initial segment $\sigma \subset K$ and some t_0 such that $g(x) = \Phi_{e,t_0}^{\sigma}(x)$.
Since $\{K_s\}$ is a c.e. approximation to K, there is a stage t_1 such that $K_{t_1} \upharpoonright |\sigma| = K \upharpoonright |\sigma|$.
Let $s = \max\{t_0, t_1\}$.
Then $\Phi_{e,s}^{K_s}(x) = \Phi_e^K(x) = g(x)$, and $\Phi_{e,s}^{K_s}(x)\downarrow$.
So $f(x,s) = g(x)$. ■

Definition 6.3.4. If $\Phi_{e,s}^X(n)\downarrow$ for some Turing program P_e at s steps, we call the largest number of X on the oracle tape that was queried the use of the computation.

Definition 6.3.5. Define the following set, for $e \in \mathbb{N}$.

$$X^{[e]} := \{\langle e, x\rangle \mid x \in X\}$$

Next we wish to construct a low c.e. set that is not computable. We will build A in steps, such that $A_{s+1} \supset A_s$, and $A = \lim_{s\to\infty}[A_s]$. In the end A will satisfy the following conditions for all $e \in \mathbb{N}$.

$$\begin{aligned} \text{for non-computability} \quad P_e : &\quad A \neq \varphi_e \\ \text{for being low} \quad N_e : &\quad \exists^\infty s\big(\Phi_{e,s}^A(e)\downarrow \to \Phi_e^A(e)\downarrow\big) \end{aligned}$$

When A will meet all of N_e, we will use an auxiliary function $f(e,s) = 1$ whenever $\Phi_{e,s}^{A_s}(e)\downarrow$ and 0 otherwise, so that $A'(e) = \lim_{s\to\infty}[f(e,s)]$. Then we will have that A' is limit computable, and so $A' \leqslant_T \varnothing'$.

Theorem 6.3.6. There exists a low c.e. set A that is not computable.

Proof: Let $x_{e,s}$ be witnesses at stage s so that $x_e = \lim_{s\to\infty}[x_{e,s}]$ exists with $A(x_e) \neq \varphi_e(x_e)$.
Construct A as follows.
Stage 0: Let $r(e,0) = 0$ and $x_{e,0} = \langle e, 0\rangle$.

Stage $s+1$: Suppose $\varphi_{e,s+1}(x_{e,s})\downarrow$ and $\varphi_{e,s+1}(x_{e,s}) = 0$ for some P_e that is not satisfied.
Enumerate $x_{e,s}$ into A_{s+1}, so that P_e may be declared satisfied.
For all $e \leqslant s$, if $\Phi_{e,s+1}^{A_{s+1}}(e)\downarrow$, let $r(e,s+1)$ be the use of the computation.
For all $i \leqslant s$, let $x_{i,s+1}$ be the least y such that $y \in \mathbb{N}^{[i]}$ with $y \notin A_{s+1}$ and $y > r(e,s+1)\ \forall\ e < i$.

Let $A = \lim_{s\to\infty}[A_s]$.
For each e, there is at most one stage s when $x_{e,s}$ is enumerated into A.
If $X_{e,s}$ is enumerated into A at stage s, then P_e is satisfied, and there is no further enumeration.

For all $e \in \mathbb{N}$, the limit $\lim_{s\to\infty}[r(e,s)]$ exists and is finite.
Let s be a stage where, for $i \leqslant e$, if $x_{i,t}$ is ever going to be enumerated into A, then it has happened by stage s.

Then by above, such an s exists.
Suppose there is a stage $s' > s$ where $r(e, s') \neq 0$.
Then $\Phi_{e,s'}^{A_{s'}}(e)\downarrow$, and $r(e, s')$ is the use of the computation.
As $s' > s$ and all $x_{e,t} > r(e, s')$ for unsatisfied P_e that might be satisfied, there will be no enumeration below $r(e, s')$ in A, so $\Phi_{e,s'}^{A_{s'}} = \Phi_e^A(e)$ and $r(e, t) = r(e, s')$ for all $t \geqslant s'$.

To meet N_0, check if $\Phi_{0,s}^{A_s}(0)\downarrow$ at some stage s.
If this happens, do not enumerate 0 into A below $r(e, s)$.
The N_e conditions for $e \in \mathbb{N}$ are all met.
Let s be such that $r(e, s) = \lim_{t\to\infty}[r(e, t)]$.
Then if $r(e, s) \neq 0$, then $\Phi_{e,t}^{A_t}(e) = \Phi_e^A(e)$ for all $t \geqslant s$ by the above discussion.
If $r(e, s) = 0$, then $\Phi_{e,t}^{A_t}(e)\uparrow$ for all $t \geqslant s$.

To meet P_0, i.e. to ensure that $A \neq \varphi_0$, wait until a stage s when $\varphi_{0,s}(0)\downarrow$ and $\varphi_{0,s}(0) = 0$.
If this never happens, then $0 \notin A$, so $A(0) = 0 \neq \varphi_0(0)$.
If at stage s we have $\varphi_{0,s}(0)\downarrow$ and $\varphi_{0,s}(0) = 0$, then we enumerate $0 \in A_{s+1}$, so $A(0) = 1 \neq 0 = \varphi_0(0)$.
If $\Phi_{e,s}^{A_s}(e)\downarrow$ and $\Phi_{e,s}^{A_s}(e) \neq \Phi_e^A(e)$, then at stage $t > s$, some $x < r(e, t)$ was enumerated into A_s.
The P_e conditions for $e \in \mathbb{N}$ are all met.
Let s be such that $r(i, s) = \lim_{t\to\infty}[r(i, t)]$ for all $i < e$.
Then $x_{e,t} = x_{e,s}$ for all $t \geqslant s$.
Let $x_e = \lim_{t\to\infty}[x_{e,t}]$.
If $\varphi_e(x_e)\downarrow$ and $\varphi_e(x_e) = 0$, then $\varphi_e(x_{e,t})\downarrow$ and $\varphi_e(x_{e,t}) = 0$ for some $t \geqslant s$.
At such a stage t, if P_e was not yet satisfied, we enumerate $x_{e,t}$ into A_t, so $A(x_e) \neq \varphi_e(x_e)$.
If P_e was already satisfied, then $\varphi_e(x_{e,\tilde{s}})\downarrow$ with $\varphi_e(x_{e,\tilde{s}}) = 0$.
Moreover, $x_{e,\tilde{s}} \in A$ for some $\tilde{s} \leqslant t$, so $A(x_{e,\tilde{s}}) \neq \varphi_e(x_{e,\tilde{s}})$.
If $\varphi_e(x_e) \neq 0$, then $\varphi_{e,t}(x_{e,t}) \neq 0$ at any t after $x_{e,t} = x_e$.
Thus $x_e \notin A$, so $A(x_e) = 0 \neq \varphi_e(x_e)$.

Therefore A is not computable, low, and c.e. ■