My topology exercises

Evgeny Markin

2023

Contents

Preface

Those are my solutions and notes for "A Concise Introduction to Mathematical Logic" (3rd edition) by Wolfgang Rautenberg

Chapter 1

Propositional Logic

1.1 Boolean Functions and Formulas

1.1.1

$f \in B_n$ is called linear if $f(x_1, ..., x_n) = a_0 + a_1x_1 + ... + a_nx_n$ for suitable coefficients $a_0, ..., a_n \in \{0, 1\}$

We firstly going to assume that $+$ is associative and commutative.

(a) Show that the above representation of a linear function f is unique

By constructing an appropriate table we can prove that

$$a_0 + a_1x_1 = b_0 + b_1x_1 \iff a_0 = b_0 \land a_1 = b_1$$

Assume that

$$\sum_{i<n} a_ix_i = \sum_{i<n} b_ix_i \iff \{a_n\} = \{b_n\}$$

Now assume that

$$\sum_{i<n} a_ix_i + a_nx_n = \sum_{i<n} b_ix_i + b_nx_n$$

we follow that if $a_n \neq b_n$, then without loss of generality we can assume that $a_n = 0$ and $b_n = 1$. Thus

$$\sum_{i<n} a_ix_i + x_n = \sum_{i<n} b_ix_i$$

Let $\{q_n\}$ be a vector of boolean variables. Substituting all the x's in $\sum_{i<n} a_ix_i$ for q's we're going to get result m. If $m = 0$, then we can set x_n to 1 to follow that

$$\sum_{i<n} a_iq_i + q_n = 1 \neq \sum_{i<n} b_ix_i$$

and if $m = 1$, then we can set $q_n = 1$ to also get

$$\sum_{i<n} a_i q_i + q_n = 0 \neq \sum_{i<n} b_i x_i$$

thus concluding that (attention to $\leq$)

$$\sum_{i\leq n} a_i x_i + a_n x_n = \sum_{i\leq n} b_i x_i + b_n x_n \Leftrightarrow \{a_n\} = \{b_n\}$$

now we can use the induction to conclude the desired result.

(b) Determine the number of n-ary Boolean functions

Since linear functions are determined uniquely by their coefficients, it's easy to say that there are 2^{n+1} n-ary linear Boolean functions

(c) Prove that each formula α in $\neg, +$ (i.e. α is a formula of the logical signature $\{\neg, +\}$) represents a linear Boolean functions.

We follow that

$$\neg(x + y) = \neg(\neg xy \vee \neg yx) = \neg(\neg xy) \wedge \neg(\neg yx) = (x \vee \neg y) \wedge (\neg x \vee y) =$$

The rest of the exercises are pretty trivial, so I'm gonna leave them alone

1.1.2 Semantic Equivalence and Normal Forms