

# Chapter 4

## Formal Verification

### 4.1 Introduction to Propositional Logic

**Definition 4.1.1** *Propositional Formulas* (also called boolean formulas) are built from *atoms*  $p_1, p_2, p_3, \dots$ , the unary connective  $\neg$  (NOT), the binary connectives  $\wedge, \vee$  (AND, OR, respectively), and parenthesis  $(, )$ . We define propositional formulas by structural induction:

1. Any atom  $p$  is a formula,
2. if  $\alpha, \beta$  are formulas, then so are  $\neg\alpha, (\alpha \wedge \beta), (\alpha \vee \beta)$ , and  $(\alpha)$ .

**Example 4.1.1**  $p, (p \vee q), (\neg(p \wedge q) \wedge (\neg p \vee \neg q))$

We use  $\{a, b, c, \dots, x, y, z, \dots, p, q, r, \dots\}$ , as different metavariables that stand for atoms. We will use  $\alpha \supset \beta$  (implies) to express  $(\neg\alpha \vee \beta)$ , and  $\alpha \leftrightarrow \beta$  (is equivalent to) to express  $((\alpha \supset \beta) \wedge (\beta \supset \alpha))$

**Exercise 4.1.1** Show how to define well-formed boolean formulas over the variables  $\{a, b, c, \dots, z\}$  with a context-free grammar. Also, once you have given a grammar, transform it to Chomsky Normal Form.

**Exercise 4.1.2** Note that the grammar in exercise 4.1.1 only generates propositional formulas with variables in the set  $\{a, b, \dots, z\}$ . How would you modify this grammar so that *all* atoms  $p_1, p_2, p_3, \dots$ , could be generated? (**Hint:** You need to show how to generate all possible subscripts.)

**Theorem 4.1.1 (Unique Readability Theorem)** Suppose that  $\alpha, \beta, \alpha', \beta'$  are formulas,  $c, c'$  are binary connectives, and  $(\alpha c \beta) =_{\text{synt}} (\alpha' c' \beta')$ . Then  $\alpha =_{\text{synt}} \alpha'$  and  $\beta =_{\text{synt}} \beta'$  and  $c =_{\text{synt}} c'$ .

Note that this theorem says that the grammar for generating formulas is unambiguous.

Here  $\alpha =_{\text{synt}} \alpha'$  denotes that  $\alpha$  and  $\alpha'$  are equal as string of symbols (syntactic identity, rather than semantic identity).

PROOF: Assign weights to all symbols as follows:

0 to  $\neg$   
 1 to  $\wedge, \vee, ($   
 $-1$  to  $), p$ , for each atom  $p$

Define the weight of a formula to be the sum of the weights of all the symbols in it. We now show the following auxiliary lemma.

**Lemma 4.1.1** The weight of any formula  $\alpha$  is  $-1$ , but the weight of any proper initial segment is  $\geq 0$ . (Hence no proper initial segment of a formula is a formula).

PROOF: (Of lemma 4.1.1) By structural induction on the length of  $\alpha$ . In the basis case,  $w(p) = -1$ , for any atom  $p$ . In the induction step there are three cases:  $\neg\alpha$ ,  $(\alpha \wedge \beta)$  and  $(\alpha \vee \beta)$ . □ □

**Exercise 4.1.3** Finish the induction step of the proof of lemma 4.1.1.

**Definition 4.1.2** A *truth assignment* is a map  $\tau : \{\text{atoms}\} \longrightarrow \{\text{T}, \text{F}\}$ . Here  $\{\text{T}, \text{F}\}$  represents True and False (sometimes denoted 1,0). The truth assignment  $\tau$  can be extended to assign either T or F to every formula, as follows:

1.  $(\neg\alpha)^\tau = \text{T}$  iff  $\alpha^\tau = \text{F}$
2.  $(\alpha \wedge \beta)^\tau = \text{T}$  iff  $\alpha^\tau = \text{T}$  and  $\beta^\tau = \text{T}$
3.  $(\alpha \vee \beta)^\tau = \text{T}$  iff  $\alpha^\tau = \text{T}$  or  $\beta^\tau = \text{T}$

**Definition 4.1.3** We define some basic notions:

1. We say that  $\tau$  *satisfies*  $\alpha$  iff  $\alpha^\tau = \text{T}$ .
2.  $\tau$  *satisfies* a set of formulas  $\Phi$  iff  $\tau$  satisfies all  $\alpha \in \Phi$ .
3.  $\Phi$  is *satisfiable* if some  $\tau$  satisfies it; otherwise,
4.  $\Phi$  is *unsatisfiable*
5.  $\Phi \models \alpha$  ( $\alpha$  is a *logical consequence* of  $\Phi$ ) iff  $\tau$  satisfies  $\alpha$  for every  $\tau$  such that  $\tau$  satisfies  $\Phi$ .
6. A formula  $\alpha$  is *valid* iff  $\models \alpha$  (i.e.,  $\alpha^\tau = \text{T}$  for all  $\tau$ ).

7. A valid propositional formula is called a *tautology*.
8.  $\alpha$  and  $\beta$  are *equivalent* formulas (written  $\alpha \iff \beta$ ) iff  $\alpha \models \beta$  and  $\beta \models \alpha$ .  
Note that ‘ $\iff$ ’ and ‘ $\leftrightarrow$ ’ have different meanings (‘ $\iff$ ’ is a semantic notion in that it represents semantic equivalence, and ‘ $\leftrightarrow$ ’ is a syntactic notion in that it is a connective).

**Example 4.1.2** The following are tautologies:  $p \vee \neg p, p \supset p, \neg(p \wedge \neg p)$ .

Logical consequence:  $(p \wedge q) \models (p \vee q)$ .

Equivalence:  $\neg(p \vee q) \iff (\neg p \wedge \neg q)$  (De Morgan’s Law).

**Exercise 4.1.4** Show that if  $\Phi \models \alpha$  and  $\Phi \cup \{\alpha\} \models \beta$ , then  $\Phi \models \beta$ .

**Exercise 4.1.5** Prove the following **Duality Theorem**: Let  $\alpha'$  be the result of interchanging  $\vee$  and  $\wedge$  in  $\alpha$ , and replacing  $p$  by  $\neg p$  for each atom  $p$ . Then  $\alpha \iff \alpha'$ .

**Exercise 4.1.6** Prove the **Craig Interpolation Theorem**: Let  $\alpha$  and  $\beta$  be any two propositional formulas. Let  $\text{Var}(\alpha)$  be the set of variables that occur in  $\alpha$ . Let  $S = \text{Var}(\alpha) \cap \text{Var}(\beta)$ . Assume  $S$  is not empty. If  $\alpha \supset \beta$  is valid, then there exists a formula  $\gamma$  such that  $\text{Var}(\gamma) = S$ , called an “interpolant” such that  $\alpha \supset \gamma$  and  $\gamma \supset \beta$  are both valid.

## 4.2 Gentzen’s system PK

One way to establish that a formula  $\alpha$  with  $n$  atoms is a tautology is to verify that  $A^\tau = \top$  for all  $2^n$  truth assignments  $\tau$  to the atoms of  $A$ . A similar exhaustive method can be used to verify that  $\Phi \models \alpha$  (if  $\Phi$  is finite).

Another way, is to use the notion of a formal proof; one such proof system in Gentzen’s system PK. In the propositional sequent calculus system PK, each line in a proof is a *sequent* of the form:

$$S = \alpha_1, \dots, \alpha_k \rightarrow \beta_1, \dots, \beta_l$$

where  $\rightarrow$  is a new symbol, and  $\alpha_1, \dots, \alpha_k$  and  $\beta_1, \dots, \beta_l$  are sequences of formulas ( $k, l \geq 0$ ) called *cedents* (*antecedent* and *succedent*, respectively).

A truth assignment  $\tau$  *satisfies* the sequent  $S$  iff  $\tau$  falsifies some  $\alpha_i$  or  $\tau$  satisfies some  $\beta_i$ , i.e., iff  $\tau$  satisfies the formula:

$$\alpha_S = (\alpha_1 \wedge \dots \wedge \alpha_k) \supset (\beta_1 \vee \dots \vee \beta_l).$$

If the antecedent is empty,  $\rightarrow \alpha$  is equivalent to  $\alpha$ , and if the succedent is empty,  $\alpha \rightarrow$  is equivalent to  $\neg \alpha$ . If both antecedent and succedent are empty, then  $\rightarrow$  is false (unsatisfiable).

$$\frac{\Gamma_1, \alpha, \beta, \Gamma_2 \rightarrow \Delta}{\Gamma_1, \beta, \alpha, \Gamma_2 \rightarrow \Delta} \quad \frac{\Gamma \rightarrow \Delta_1, \alpha, \beta, \Delta_2}{\Gamma \rightarrow \Delta_1, \beta, \alpha, \Delta_2}$$

$$\frac{\Gamma, \alpha, \alpha \rightarrow \Delta}{\Gamma, \alpha \rightarrow \Delta} \quad \frac{\Gamma \rightarrow \Delta, \alpha, \alpha}{\Gamma \rightarrow \Delta, \alpha}$$

$$\frac{\Gamma \rightarrow \Delta}{\alpha, \Gamma \rightarrow \Delta} \quad \frac{\Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta, \alpha}$$

Table 4.1: Weak structural rules of PK: exchange, contraction, weakening.

$$\frac{\Gamma \rightarrow \Delta, \alpha \quad \alpha, \Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta}$$

Table 4.2: The cut rule of PK.

We have notions of satisfiability, validity, logical consequence, etc., for sequents, analogous to those for formulas (see definition 4.1.3).

**Example 4.2.1** Some examples of valid sequents:  $\alpha \rightarrow \alpha$ ,  $\rightarrow \alpha$ ,  $\neg\alpha$ ,  $\alpha, \alpha \supset \beta \rightarrow \beta$ ,  $\alpha \wedge \neg\alpha \rightarrow$

**Definition 4.2.1** A formal *proof* in PK is a finite rooted tree in which the nodes are labeled with sequents. The sequent at the root (bottom) is what is being proved: the *endsequent*. The sequents at the leaves (top) are *logical axioms*, and must be of the form  $\alpha \rightarrow \alpha$ , where  $\alpha$  is a formula. Each sequent other than the logical axioms must follow from its parent sequent(s) by one of the rules of inference (given in tables 4.1, 4.2, and 4.3).

**Exercise 4.2.1** Give PK proofs for the valid sequents  $\neg(P \vee Q) \rightarrow \neg P \wedge \neg Q$ ,  $\neg P \wedge \neg Q \rightarrow \neg(P \vee Q)$ , and  $(P_1 \wedge (P_2 \wedge (P_3 \wedge P_4))) \rightarrow (((P_1 \wedge P_2) \wedge P_3) \wedge P_4)$ .

$$\frac{\Gamma \rightarrow \Delta, \alpha}{\neg\alpha, \Gamma \rightarrow \Delta} \quad \frac{\alpha, \Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta, \neg\alpha}$$

$$\frac{\alpha, \beta, \Gamma \rightarrow \Delta}{(\alpha \wedge \beta), \Gamma \rightarrow \Delta} \quad \frac{\Gamma \rightarrow \Delta, \alpha \quad \Gamma \rightarrow \Delta, \beta}{\Gamma \rightarrow \Delta, (\alpha \wedge \beta)}$$

$$\frac{\alpha, \Gamma \rightarrow \Delta \quad \beta, \Gamma \rightarrow \Delta}{(\alpha \vee \beta), \Gamma \rightarrow \Delta} \quad \frac{\Gamma \rightarrow \Delta, \alpha, \beta}{\Gamma \rightarrow \Delta, (\alpha \vee \beta)}$$

Table 4.3: Rules of PK for eliminating connectives.

**Exercise 4.2.2** Show that the contraction rules can be derived from the cut rule (with exchanges and weakenings).

**Exercise 4.2.3** Suppose that we allowed  $\supset$  as a primitive connective, rather than one introduced by definition. Give the appropriate left and right introduction rules for  $\supset$ . Do the same for  $\leftrightarrow$ .

**Lemma 4.2.1 (Rule Soundness Principle)** For each PK rule, the sequent on the bottom is a logical consequence of the sequent(s) on the top.

**Exercise 4.2.4** Prove this lemma.

**Theorem 4.2.1 (Soundness)** Each sequent provable in PK is valid.

PROOF: We show that the endsequent in every PK proof is valid, by induction on the number of sequents in the proof.

For the Basis Case, the proof is a single line; an axiom  $\alpha \rightarrow \alpha$ . This is obviously valid.

For the induction step, one need only verify for each rule, if all top sequents are valid, then the bottom sequent is valid. This follows from the Rule Soundness Principle.  $\square$

**Theorem 4.2.2 (Inversion Principle)** For each PK rule, except weakening, if the bottom sequent is valid, then all top sequents are valid.

**Exercise 4.2.5** Prove the Inversion Principle.

**Exercise 4.2.6** Give an example, with the weakening rule, for which this principle fails.

**Theorem 4.2.3 (Completeness)** Every valid propositional sequent is provable in PK without using cut or contraction.

PROOF: We show that every valid sequent  $\Gamma \rightarrow \Delta$  has a PK proof, by induction on the total number of connectives  $\wedge, \vee, \neg$ , occurring in  $\Gamma \rightarrow \Delta$ .

Basis Case: zero connectives, so every formula in  $\Gamma \rightarrow \Delta$  is an atom, and since it is valid, some atom  $p$  must be in both  $\Gamma$  and  $\Delta$ . Hence  $\Gamma \rightarrow \Delta$  can be derived from  $p \rightarrow p$  by weakenings and exchanges.

Induction Step: suppose  $\gamma$  is non-atomic, in  $\Gamma$  or  $\Delta$ . Then it is of the form  $\neg\alpha, (\alpha \wedge \beta), (\alpha \vee \beta)$ . Then,  $\Gamma \rightarrow \Delta$  can be derived by one of the connective elimination rules, using exchanges. The top sequent(s) will have one fewer connective than  $\Gamma \rightarrow \Delta$ , and are valid by the Inversion Principle; hence they have PK proofs by the IH.  $\square$

**Exercise 4.2.7** What are the five rules *not* used in the induction step in the above proof?

**Exercise 4.2.8** Consider  $\text{PK}'$ , which is like  $\text{PK}$ , but where the axioms must be of the form  $p \rightarrow p$ , i.e.,  $\alpha$  must be atomic in the logical axioms. Prove the above four principles for  $\text{PK}'$ .

**Exercise 4.2.9** Suppose that  $\{\rightarrow \beta_1, \dots, \rightarrow \beta_n\} \vDash \Gamma \rightarrow \Delta$ . Give a  $\text{PK}$  proof of  $\Gamma \rightarrow \Delta$  where all the leaves are either logical axioms  $\alpha \rightarrow \alpha$ , or one of the non-logical axioms  $\rightarrow \beta_i$ . (**Hint:** your proof will require the use of the cut rule.) Now give a proof of the fact that given a finite  $\Phi$  such that  $\Phi \vDash \Gamma \rightarrow \Delta$ , there exists a  $\text{PK}$  proof of  $\Gamma \rightarrow \Delta$  where all the leaves are logical axioms or sequents in  $\Phi$ .

**Exercise\* 4.2.1** Redo exercise 4.2.9, that is show that if  $\Phi \vDash \Gamma \rightarrow \Delta$  then there is a  $\text{PK}$  proof of  $\Gamma \rightarrow \Delta$  where all the leaves are logical axioms or sequents in  $\Phi$ , and all the cuts are restricted to formulas which appear in the sequents of  $\Phi$ . (Such proofs are called *anchored*, and this is a proof of *anchored implicational completeness*.)

### 4.3 Introduction to First Order Logic

**Definition 4.3.1** Let  $\mathcal{L} = [f_1, \dots, f_n; R_1, \dots, R_m]$  be a language, where  $f_i$  is an  $k_i$ -ary function, and  $R_j$  is an  $l_j$ -ary predicate. We define  $\mathcal{L}$ -terms as follows:

1. Every variable is a term:  $a, b, c, \dots, x, y, z, \dots$ ,
2. if  $f$  is an  $n$ -ary function symbol and  $t_1, t_2, \dots, t_n$  are terms, then so is  $ft_1t_2 \dots t_n$ .

A 0-ary function symbol is called a *constant* (we use  $c, d, e$  as a metasympol for constants).

**Example 4.3.1**  $f$  is binary and  $g$  is unary, then  $fgex, fxy, gfege$  are terms.

**Exercise 4.3.1** Show the Unique Readability Theorem for terms.

**Example 4.3.2** *The language of arithmetic* is  $\mathcal{L}_A = [0, s, +, \cdot, =]$ , where 0 is a constant,  $s$  is a unary function called *successor*,  $+$ ,  $\cdot$  are binary functions, and  $=$  is a binary predicate. We use *infix* notation, rather than the formal *prefix* notation for  $+$ ,  $\cdot$ , that is,  $(t_1 \cdot t_2) \stackrel{\text{synt}}{=} \cdot t_1 t_2$ ,  $(t_1 + t_2) \stackrel{\text{synt}}{=} + t_1 t_2$ . Examples of  $\mathcal{L}_A$  terms are  $sss0$  and  $((x + sy) \cdot (ssz + s0))$ . Also, let  $t_1 = t_2$  stand for  $= t_1 t_2$  and  $t_1 \neq t_2$  for  $\neg = t_1 t_2$ .

**Definition 4.3.2** We define  $\mathcal{L}$ -formulas by structural induction: (i)  $Pt_1t_2\dots t_n$  is an *atomic* formula,  $P$  is an  $n$ -ary predicate symbol,  $t_1, t_2, \dots, t_n$  are terms. (ii) If  $\alpha, \beta$  are formulas, then so are  $\neg\alpha, (\alpha \vee \beta), (\alpha \wedge \beta)$ . (iii) If  $\alpha$  is a formula, and  $x$  a variable, then  $\forall x\alpha$  and  $\exists x\alpha$  are also formulas. We use  $P, Q, R, \dots$ , as metavariables for predicate symbols.

**Example 4.3.3** Let  $f$  be a unary function symbol,  $P$  a unary predicate symbol, and  $Q$  a binary predicate symbol. Then the following are well formed formulas:  $(\neg\forall xPx \vee \exists x\neg Px), (\forall x\neg Qxy \wedge \neg\forall zQfyz)$ .

**Exercise 4.3.2** Show that the set of (well-formed)  $\mathcal{L}$ -formulas can be given by a context-free grammar  $G_{\mathcal{L}}$ .

**Definition 4.3.3** An occurrence of  $x$  in  $\alpha$  is *bound* if it is in a subformula of  $\alpha$  of the form  $\forall x\beta$  or  $\exists x\beta$  (i.e., in the *scope* of a quantifier). Otherwise, the occurrence is *free*.

**Example 4.3.4**  $\exists y(x = y + y)$ :  $x$  is free, but  $y$  is bound. In  $Px \wedge \forall xQx$  the variable  $x$  occurs both as free and bound.

**Definition 4.3.4** A term  $t$  or formula  $\alpha$  are *closed* if they contain no free variables. A closed formula is called a *sentence*.

**Definition 4.3.5** A *structure* (or *interpretation*) gives meaning to terms and formulas. An  $\mathcal{L}$  structure  $\mathcal{M}$  consists of:

1. A nonempty set  $M$ , called the *universe of discourse*.
2. For each  $n$ -ary  $f$ ,  $f^{\mathcal{M}} : M^n \rightarrow M$ .
3. For each  $n$ -ary  $P$ ,  $P^{\mathcal{M}} \subseteq M^n$ .

If  $\mathcal{L}$  contains  $=$ ,  $=^{\mathcal{M}}$  must be the usual  $=$ . Equality gets a special treatment—it must always be the true equality. On the other hand,  $<^{\mathcal{M}}$  could be anything, not necessarily the order relation we are used to. Every  $\mathcal{L}$ -sentence becomes either true or false when interpreted by an  $\mathcal{L}$ -structure  $\mathcal{M}$ . If a sentence  $\alpha$  becomes true under  $\mathcal{M}$ , we say  $\mathcal{M}$  *satisfies*  $\alpha$ , or  $\mathcal{M}$  is a *model* for  $\alpha$ , and write  $\mathcal{M} \models \alpha$ . If  $\alpha$  has free variables, then they must get values from  $M$  (the universe of discourse), before  $\alpha$  can get a truth value under  $\mathcal{M}$ . An *object assignment*  $\sigma$  for a structure  $\mathcal{M}$  is a mapping from variables to the universe  $M$ ;  $t^{\mathcal{M}}[\sigma]$  is an element in  $M$ —given by the structure  $\mathcal{M}$  and the object assignment  $\sigma$ .  $\mathcal{M} \models \alpha[\sigma]$  means that  $\mathcal{M}$  satisfies  $\alpha$  when its free variables are assigned values by  $\sigma$ .

**Definition 4.3.6** The *basic semantic definitions*, also known as *Tarski semantics*, are a way of assigning meaning to terms and formulas. We define the interpretation of terms,  $t^{\mathcal{M}}[\sigma]$ , by structural induction:

1.  $x^{\mathcal{M}}[\sigma]$  is  $\sigma(x)$
2.  $(ft_1t_2 \dots t_n)^{\mathcal{M}}[\sigma]$  is  $f^{\mathcal{M}}(t_1^{\mathcal{M}}[\sigma], t_2^{\mathcal{M}}[\sigma], \dots, t_n^{\mathcal{M}}[\sigma])$

If  $x$  is a variable, and  $m \in M$ , then  $\sigma(m/x)$  is the same object assignment as  $\sigma$ , but  $x$  is mapped to  $m$ . We define the interpretation of formulas,  $\mathcal{M} \models \alpha[\sigma]$ , by structural induction as well:

1.  $\mathcal{M} \models (Pt_1 \dots t_n)[\sigma]$  iff  $(t_1^{\mathcal{M}}[\sigma], \dots, t_n^{\mathcal{M}}[\sigma]) \in P^{\mathcal{M}}$ .
2.  $\mathcal{M} \models \neg\alpha[\sigma]$  iff  $\mathcal{M} \not\models \alpha[\sigma]$ .
3.  $\mathcal{M} \models (\alpha \wedge \beta)[\sigma]$  iff  $\mathcal{M} \models \alpha[\sigma]$  and  $\mathcal{M} \models \beta[\sigma]$ .
4.  $\mathcal{M} \models (\alpha \vee \beta)[\sigma]$  iff  $\mathcal{M} \models \alpha[\sigma]$  or  $\mathcal{M} \models \beta[\sigma]$ .
5.  $\mathcal{M} \models (\forall x\alpha)[\sigma]$  iff  $\mathcal{M} \models \alpha[\sigma(m/x)]$  for all  $m \in M$ .
6.  $\mathcal{M} \models (\exists x\alpha)[\sigma]$  iff  $\mathcal{M} \models \alpha[\sigma(m/x)]$  for some  $m \in M$ .

If  $t$  is closed, we write  $t^{\mathcal{M}}$ . If  $\alpha$  is a sentence, we write  $\mathcal{M} \models \alpha$ .

**Exercise 4.3.3** Suppose that  $\mathcal{L} = [; R, =]$  ( $R$  binary predicate) and let  $\mathcal{M}$  be an  $\mathcal{L}$ -structure with universe  $\mathbb{N}$  and such that  $(m, n) \in R^{\mathcal{M}}$  iff  $m \leq n$ . What is the truth value of  $\exists x \forall y Rxy$  and  $\exists y \forall x Rxy$  under  $\mathcal{M}$ ?

**Definition 4.3.7** The *standard structure*  $\mathbb{N}$  for the language  $\mathcal{L}_A$  has universe  $M = \mathbb{N} = \{0, 1, 2, \dots\}$ ,  $s^{\mathbb{N}}(n) = n + 1$ , and  $0, +, \cdot, =$  get their usual meaning on the natural numbers.

**Exercise 4.3.4** Is  $\forall x \forall y \exists z (x + z = y \vee y + z = x)$  satisfied by the standard structure? What about  $\forall x \exists y (y + y = x)$ ?

**Definition 4.3.8** Let  $\Phi$  denote a set of formulas.

1.  $\alpha$  is *satisfiable* iff  $\mathcal{M} \models \alpha[\sigma]$  for some  $\mathcal{M}$  &  $\sigma$ .
2.  $\mathcal{M} \models \Phi[\sigma]$  iff  $\mathcal{M} \models \alpha[\sigma]$  for all  $\alpha \in \Phi$ .
3.  $\Phi \models \alpha$  iff for all  $\mathcal{M}$  &  $\sigma$ , if  $\mathcal{M} \models \Phi[\sigma]$  then  $\mathcal{M} \models \alpha[\sigma]$ . ( $\alpha$  is a *logical consequence* of  $\Phi$ —second use of “ $\models$ ”)
4.  $\alpha$  is *valid*,  $\models \alpha$ , iff  $\mathcal{M} \models \alpha[\sigma]$  for all  $\mathcal{M}$  &  $\sigma$ .
5.  $\alpha$  &  $\beta$  are *logically equivalent*,  $\alpha \iff \beta$ , iff for all  $\mathcal{M}$  &  $\sigma$ , ( $\mathcal{M} \models \alpha[\sigma]$  iff  $\mathcal{M} \models \beta[\sigma]$ ).

Note that  $\models$  is a symbol of the “meta language” (English), as opposed to  $\wedge, \vee, \exists, \dots$  which are symbols of first order logic.

If  $\Phi$  is just one formula, i.e.,  $\Phi = \{\beta\}$ , then we write  $\beta \models \alpha$  in place of  $\{\beta\} \models \alpha$ .

**Exercise 4.3.5** Show that  $(\forall x\alpha \vee \forall x\beta) \models \forall x(\alpha \vee \beta)$  for all formulas  $\alpha$  and  $\beta$ . What about  $\forall x(\alpha \vee \beta) \models (\forall x\alpha \vee \forall x\beta)$ ?

**Definition 4.3.9 (Substitution)** If  $t, u$  are terms, then

$t(u/x)$  result of replacing all occurrences of  $x$  in  $t$  by  $u$   
 $\alpha(u/x)$  result of replacing all **free** occurrences of  $x$  in  $\alpha$  by  $u$

The semantics of substitution are as follows:  $(u(t/x))^{\mathcal{M}}[\sigma] = u^{\mathcal{M}}[\sigma(m/x)]$  where  $m = t^{\mathcal{M}}[\sigma]$ .

**Example 4.3.5** Let  $\mathcal{M}$  be  $\underline{\mathbb{N}}$  (the standard structure) for  $\mathcal{L}_A$ . Suppose  $\sigma(x) = 5$  and  $\sigma(y) = 7$ . Let:

$u$  be the term  $x + y$   
 $t$  be the term  $ss0$

Then:

$$u(t/x) \text{ is } ss0 + y \text{ and so } (u(t/x))^{\underline{\mathbb{N}}}[\sigma] = 2 + 7 = 9$$

On the other hand,  $m = t^{\underline{\mathbb{N}}} = 2$ , so  $u^{\underline{\mathbb{N}}}[\sigma(m/x)] = 2 + 7 = 9$ .

**Exercise 4.3.6** Prove  $(u(t/x))^{\mathcal{M}}[\sigma] = u^{\mathcal{M}}[\sigma(m/x)]$  where  $m = t^{\mathcal{M}}[\sigma]$ , by structural induction on  $u$ .

Does exercise 4.3.6 apply to formulas  $\alpha$ ? That is, is it true that  $\mathcal{M} \models \alpha(t/x)[\sigma]$  iff  $\mathcal{M} \models \alpha[\sigma(m/x)]$ , where  $m = t^{\mathcal{M}}[\sigma]$ ? The answer is no. For example, suppose  $\alpha$  is  $\forall y \neg(x = y + y)$ . This says “ $x$  is odd”. But  $\alpha(x + y/x)$  is  $\forall y \neg(x + y = y + y)$  which is always false, regardless of the value of  $\sigma(x)$ . The problem is that  $y$  in the term  $x + y$  got “caught” by the quantifier  $\forall y$ .

**Definition 4.3.10** A term  $t$  is *freely substitutable for  $x$  in  $\alpha$*  iff no free occurrence of  $x$  in  $\alpha$  is in a subformula of  $\alpha$  of the form  $\forall y\beta$  or  $\exists y\beta$ , where  $y$  occurs in  $t$ .

**Theorem 4.3.1 (Substitution)** If  $t$  is freely substitutable for  $x$  in  $\alpha$  then for all structures  $\mathcal{M}$  and all object assignments  $\sigma$ ,  $\mathcal{M} \models \alpha(t/x)[\sigma]$  iff  $\mathcal{M} \models \alpha[\sigma(m/x)]$ , where  $m = t^{\mathcal{M}}[\sigma]$ .

**Exercise 4.3.7** Prove the Substitution Theorem. (Hint. Use structural induction on  $\alpha$  and the BSDs.)

If a term  $t$  is not freely substitutable for  $x$  in  $\alpha$ , it is because some variable  $y$  in  $t$  gets “caught” by a quantifier  $\forall y$  or  $\exists y$  in  $\alpha$ . One way to fix this is simply rename the bound variable  $y$  in  $\alpha$  to some new variable  $z$ . This renaming does not change the meaning of  $\alpha$ .

## 4.4 Gentzen's system LK

Henceforth, let  $a, b, c, \dots$  denote free variables and let  $x, y, z, \dots$  to denote bound variables.

**Definition 4.4.1** A first order formula  $\alpha$  is called a *proper formula* if it satisfies the restriction that it has no free occurrence of any “bound” variable and no bound occurrence of any “free” variable. Similarly a *proper term* has no “bound” variable.

Notice that a subformula of a proper formula is not necessarily proper, and a proper formula may contain terms which are not proper.

The sequent system LK is an extension of the propositional system PK where now all formulas in the sequent  $\alpha_1, \dots, \alpha_k \rightarrow \beta_1, \dots, \beta_l$  must be proper formulas.

The system LK is PK together with the following four rules for introducing quantifiers:

$$\frac{\alpha(t), \Gamma \rightarrow \Delta}{\forall x \alpha(x), \Gamma \rightarrow \Delta} \qquad \frac{\Gamma \rightarrow \Delta, \alpha(b)}{\Gamma \rightarrow \Delta, \forall x \alpha(x)}$$

$$\frac{\alpha(b), \Gamma \rightarrow \Delta}{\exists x \alpha(x), \Gamma \rightarrow \Delta} \qquad \frac{\Gamma \rightarrow \Delta, \alpha(t)}{\Gamma \rightarrow \Delta, \exists x \alpha(x)}$$

The rules have the following restrictions:

1.  $t$  must be a proper term.
2.  $\alpha(t)$  (respectively  $\alpha(b)$ ) is the result of substituting  $t$  (respectively  $b$ ) for all free occurrences of  $x$  in  $\alpha(x)$ . (Note that  $t, b$  can be freely substituted for  $x$  in  $\alpha(x)$  because  $\forall x \alpha(x), \exists x \alpha(x)$  are proper formulas.)
3. The free variable  $b$  must not occur in the conclusion in  $\forall$  right and  $\exists$  left.

**Exercise 4.4.1** Show that the four new rules are sound.

**Exercise 4.4.2** Give a specific example of a sequent  $\Gamma \rightarrow \Delta, \alpha(b)$  which is valid, but the bottom sequent  $\Gamma \rightarrow \Delta, \forall x \alpha(x)$  is not valid, because the restriction on  $b$  is violated ( $b$  occurs in  $\Gamma$  or  $\Delta$  or  $\forall x \alpha(x)$ ). Do the same for  $\exists$  left.

An LK proof of a valid first order sequent can be obtained using the same method as in the propositional case. Write the goal sequent at the bottom, and move up using the introduction rules in reverse.

If there is a choice about which quantifier to remove next, choose  $\forall$  right or  $\exists$  left (working backward), since these rules carry a restriction.

## 4.5 Peano Arithmetic

Recall the language of arithmetic,  $\mathcal{L}_A = [0, s, +, \cdot, =]$ .

### Axioms for PA

- P1  $\forall x (sx \neq 0)$   
 P2  $\forall x \forall y (sx = sy \supset x = y)$   
 P3  $\forall x (x + 0 = x)$   
 P4  $\forall x \forall y (x + sy = s(x + y))$   
 P5  $\forall x (x \cdot 0 = 0)$   
 P6  $\forall x \forall y (x \cdot sy = x \cdot y + x)$

And the **Induction Scheme**:

$$\forall y_1 \dots \forall y_k [(\alpha(0) \wedge \forall x (\alpha(x) \supset \alpha(sx))) \supset \forall x \alpha(x)] \quad (4.1)$$

where  $\alpha$  is any  $\mathcal{L}_A$ -formula, and (4.1) is a sentence.

### Equality Axioms

- E1  $\forall x (x = x)$   
 E2  $\forall x \forall y (x = y \supset y = x)$   
 E3  $\forall x \forall y \forall z ((x = y \wedge y = z) \supset x = z)$   
 E4  $\forall x_1 \dots \forall x_n \forall y_1 \dots \forall y_n (x_1 = y_1 \wedge \dots \wedge x_n = y_n) \supset f x_1 \dots x_n = f y_1 \dots y_n$   
 E5  $\forall x_1 \dots \forall x_n \forall y_1 \dots \forall y_n (x_1 = y_1 \wedge \dots \wedge x_n = y_n) \supset P x_1 \dots x_n \supset P y_1 \dots y_n$

where E4 and E5 hold for all  $n$ -ary function and predicate symbols.

In  $\mathcal{L}_A$   $s$  is unary,  $+$ ,  $\cdot$  are binary, and  $=$  is binary.

Let LK-PA be LK where the leaves are allowed to be P1-6 and E1-5, besides the usual axioms  $\alpha \rightarrow \alpha$ .

For example,  $\rightarrow \forall x (x = x)$  would be a valid leaf.

**Exercise 4.5.1** Show that LK-PA proves that all nonzero elements have predecessor.

**Exercise 4.5.2** Show that LK-PA proves the following: (i) the associative and commutative law of addition, (ii) the associative and commutative laws of multiplication, (iii) that multiplication distributes over addition. Specify carefully which axioms you are using.

## 4.6 Verification

Recall that  $\{\alpha\}P\{\beta\}$  means that if formula  $\alpha$  is true before execution of  $P$ ,  $P$  is executed and terminates, then formula  $\beta$  will be true, and  $\alpha, \beta$ , are called the precondition and postcondition of the program  $P$ , respectively. They are given as formulas of  $\mathcal{L}_A$ .

Using a finite set of rules for program verification, we want to show that  $\{\alpha\}P\{\beta\}$  holds, and conclude that the program is correct *with respect to the specification*  $\alpha, \beta$ .

**Rules for program verification:**

Consequence right and left:

$$\frac{\{\alpha\}P\{\beta\} \quad (\beta \supset \gamma)}{\{\alpha\}P\{\gamma\}} \quad \frac{(\gamma \supset \alpha) \quad \{\alpha\}P\{\beta\}}{\{\gamma\}P\{\beta\}}$$

Composition and assignment:

$$\frac{\{\alpha\}P_1\{\beta\} \quad \{\beta\}P_2\{\gamma\}}{\{\alpha\}P_1P_2\{\gamma\}} \quad \frac{x = t}{\{\alpha(t)\}x = t\{\alpha(x)\}}$$

If:

$$\frac{\{\alpha \wedge \beta\}P_1\{\gamma\} \quad \{\alpha \wedge \neg\beta\}P_2\{\gamma\}}{\{\alpha\} \text{ if } \beta \text{ then } P_1 \text{ else } P_2 \{\gamma\}}$$

While:

$$\frac{\{\alpha \wedge \beta\}P\{\alpha\}}{\{\alpha\} \text{ while } \beta \text{ do } P \{\alpha \wedge \neg\beta\}}$$

As an example, we verify the program `mult(A,B)` which computes  $y = A \cdot B$ :

```

1 mult(int A, int B) {
2   int a = A;
3   int b = B;
4   int y = 0;
5   while (b > 0) {
6     y = y + a;
7     b = b - 1; }

```

Each pass through the **while** loop adds  $a$  to  $y$ , but  $a \cdot b$  decreases by  $a$  because  $b$  is decremented by 1.

**Loop invariant:**

$$(y + (a \cdot b) = A \cdot B) \wedge b \geq 0$$

We want to show:

$$\{B \geq 0\} \text{mult}(A,B) \{y = AB\} \tag{4.2}$$

using the loop invariant (the “creative” part of the proofs of correctness). To save space, write  $tu$  instead of  $t \cdot u$ . Let  $t \geq u$  abbreviate the  $\mathcal{L}_A$ -formula  $\exists x(t = u + x)$ , and let  $t \leq u$  abbreviate  $u \geq t$ .

1	$\{y + a(b - 1) = AB \wedge (b - 1) \geq 0\} b = b - 1; \{y + ab = AB \wedge b \geq 0\}$	assignment
2	$\{(y + a) + a(b - 1) = AB \wedge (b - 1) \geq 0\} y = y + a; \{y + a(b - 1) = AB \wedge (b - 1) \geq 0\}$	assignment
3	$(y + ab = AB \wedge b - 1 \geq 0) \supset ((y + a) + a(b - 1) = AB \wedge b - 1 \geq 0)$	theorem
4	$\{y + ab = AB \wedge b - 1 \geq 0\} y = y + a; \{y + a(b - 1) = AB \wedge b - 1 \geq 0\}$	consequence left on 2 and 3
5	$\{y + ab = AB \wedge b - 1 \geq 0\} y = y + a; b = b - 1; \{y + ab = AB \wedge b \geq 0\}$	composition on 4 and 1
6	$(y + ab = AB) \wedge b \geq 0 \wedge b > 0 \supset (y + ab = AB) \wedge b - 1 \geq 0$	theorem
7	$\{(y + ab = AB) \wedge b \geq 0 \wedge b > 0\} y = y + a; b = b - 1; \{y + ab = AB \wedge b \geq 0\}$	consequence left on 5 and 6
8	$\{(y + ab = AB) \wedge b \geq 0\}$ while $(b > 0)$ { $y = y + a;$ $b = b - 1;$ } $\{y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$	while on 7
9	$\{(0 + ab = AB) \wedge b \geq 0\}$ int $y = 0; \{(y + ab = AB) \wedge b \geq 0\}$	assignment
10	$\{(0 + ab = AB) \wedge b \geq 0\}$ int $y = 0;$ while $(b > 0)$ { $y = y + a;$ $b = b - 1;$ } $\{y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$	composition on 9 and 8
11	$\{(0 + aB = AB) \wedge B \geq 0\}$ int $b = B; \{(0 + ab = AB) \wedge b \geq 0\}$	assignment
12	$\{(0 + aB = AB) \wedge B \geq 0\}$ int $b = B;$ int $y = 0;$ while $(b > 0)$ { $y = y + a;$ $b = b - 1;$ } $\{y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$	composition on 11 and 10
13	$\{(0 + AB = AB) \wedge B \geq 0\}$ int $a = A; \{(0 + aB = AB) \wedge B \geq 0\}$	assignment
14	$\{(0 + AB = AB) \wedge B \geq 0\}$ mult(int $A, \text{int } B) \{y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$	composition on 13 and 12
15	$B \geq 0 \supset ((0 + AB = AB) \wedge B \geq 0)$	theorem
16	$(y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)) \supset y = AB$	theorem
17	$\{B \geq 0\}$ mult(int $A, \text{int } B) \{y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$	consequence left on 15 and 14
18	$\{B \geq 0\}$ mult(int $A, \text{int } B) \{y = AB\}$	consequence right on 16 and 17

## 4.7 Answers to selected exercises

### Exercise 4.1.1.

$G_{\text{prop}} = (\{S\}, \{a - z, (, ), \wedge, \vee, \neg\}, \{S \longrightarrow a - z | (S \wedge S) | (S \vee S) | \neg S\}, S)$ .

We convert  $G_{\text{prop}}$  to Chomsky Normal Form:  $X_0 \longrightarrow a - z, X_1 \longrightarrow (, X_2 \longrightarrow ), X_3 \longrightarrow \wedge, X_4 \longrightarrow \vee, X_5 \longrightarrow \neg, X_6 \longrightarrow X_1 X_6, X_7 \longrightarrow X_0 X_7, X_8 \longrightarrow X_3 X_8, X_9 \longrightarrow X_0 X_2, X_{10} \longrightarrow X_1 X_9, X_{11} \longrightarrow X_0 X_{10}, X_{12} \longrightarrow X_4 X_{11}, X_{13} \longrightarrow X_0 X_2, X_{14} \longrightarrow X_5 X_{10}$

**Exercise 4.3.3.**  $\mathcal{M} \models \exists x \forall y Rxy$ , but  $\mathcal{M} \not\models \exists y \forall x Rxy$ .

**Exercise 4.3.4.** Yes to the first, no to the second.

**Exercise 4.3.5.** We prove this using BSDs: Let  $\mathcal{M}$  be any structure, and  $\sigma$  any object assignment. Suppose  $\mathcal{M} \models (\forall x\alpha \vee \forall x\beta)[\sigma]$ . Then,  $\mathcal{M} \models \forall x\alpha[\sigma]$  or  $\mathcal{M} \models \forall x\beta[\sigma]$ . Case (1):  $\mathcal{M} \models \forall x\alpha[\sigma]$ . Then,  $\mathcal{M} \models \alpha[\sigma(m/x)]$  for all  $m \in M$ . Then,  $\mathcal{M} \models (\alpha \vee \beta)[\sigma(m/x)]$  for all  $m \in M$ . So,  $\mathcal{M} \models \forall x(\alpha \vee \beta)[\sigma]$ . Case (2):  $\mathcal{M} \models \forall x\beta[\sigma]$ . Same idea.  $\therefore \mathcal{M} \models \forall x(\alpha \vee \beta)[\sigma]$  By the def of logical consequence,  $(\forall x\alpha \vee \forall x\beta) \models \forall x(\alpha \vee \beta)$ . The answer to the second question is no, not necessarily; to prove that the RHS is **not** a logical consequence of the LHS, we must exhibit: (i) a model  $\mathcal{M}$ , (ii) an object assignment  $\sigma$ , (iii) formulas  $\alpha, \beta$ , such that:  $\mathcal{M} \models \forall x(\alpha \vee \beta)[\sigma]$ , but  $\mathcal{M} \not\models (\forall x\alpha \vee \forall x\beta)[\sigma]$ . Let  $\alpha$  and  $\beta$  be  $Px$  and  $Qx$ , respectively ( $P, Q$  unary predicates). Now define  $\mathcal{M}$  and  $\sigma$ . Since the formulas are sentences, no need to define  $\sigma$ .  $\mathcal{M}$ : let the universe of discourse be  $M = \mathbb{N}$ . We still need to give meaning in  $\mathcal{M}$  to  $P, Q$ . Let  $P^{\mathcal{M}} = \{0, 2, 4, \dots\}$ , and  $Q^{\mathcal{M}} = \{1, 3, 5, \dots\}$ . Then:  $\mathcal{M} \models \forall x(Px \vee Qx)$  (bec. every number is even or odd). But,  $\mathcal{M} \not\models (\forall xPx \vee \forall xQx)$  (bec. it is not true that either all numbers are even or all numbers are odd).

**Exercise 4.5.1.** Let  $\alpha(x)$  be  $(x = 0 \vee \exists y(x = sy))$ . We outline the proof informally, but the proof can of course be formalized in LK-PA. Basis:  $x = 0$ , and LK-PK proves  $\alpha(0)$  easily:

$$\frac{\frac{\rightarrow \forall x(x = x)}{\rightarrow 0 = 0, \forall x(x = x)} \text{ weak \& exch} \quad \frac{0 = 0 \rightarrow 0 = 0}{\forall x(x = x) \rightarrow 0 = 0} \forall\text{-left}}{\rightarrow 0 = 0} \text{ Cut}}{\frac{\rightarrow 0 = 0}{\rightarrow 0 = 0, \exists y(0 = sy)} \text{ weak}} \frac{\rightarrow 0 = 0, \exists y(0 = sy)}{\rightarrow 0 = 0 \vee \exists y(0 = sy)} \forall\text{-right}$$

Induction Step: Show that LK-PA proves  $\forall x(\alpha(x) \supset \alpha(sx))$ , i.e., we must give an LK-PA proof of the sequent:

$$\rightarrow \forall x(\neg(x = 0 \vee \exists y(x = sy)) \vee (sx = 0 \vee \exists y(sx = sy)))$$

This is not difficult... From  $\alpha(0)$  and  $\forall x(\alpha(x) \supset \alpha(sx))$ , and using the axiom:

$$\rightarrow (\alpha(0) \wedge \forall x(\alpha(x) \supset \alpha(sx))) \supset \forall x\alpha(x)$$

we can now conclude (in just a few steps):  $\rightarrow \forall x\alpha(x)$  which is what we wanted to prove. Thus, LK-PA proves  $\forall x\alpha(x)$ .