Chapter 3
Quantifier-free logic

The Language of First-Order Logic

Chapter 3, Section 3

First-Order Signature

Definition 5.3.1 A *first-order signature* (or *signature* when there is no danger of confusion with propositional signatures) is a 4-tuple (*Co*,*Fu*, *Re*, *r*) where

(1) Co is a set (possibly empty) of symbols called the *constant symbols* ;

(2) Fu is a set (possibly empty) of symbols called the *function symbols* ;

(3) Re is a set (possibly empty) of symbols called the *relation symbols* ;

(4) Co, Fu, Re are pairwise disjoint ;

(5) r is a function taking each symbol s in $Fe \cup Re$ to a positive integer $r(s)$ called the *rank* (or *arity*) of s. We say a symbol is *n-ary* if it has arity n ; *binary* means 2-ary.

Signature of Arithmetic

Example 5.3.2(a) The following signature will play an important role. The *signature of arithmetic*, σ_{arith}, has the following symbols :

(i) a constant symbol $\bar{0}$;

(ii) a function symbol $\bar{S}$ of arity 1 ;

(iii) two binary function symbols $\bar{+}$ and $\bar{\cdot}$.

We will usually present signatures in this informal style. To match Definition 5.3.1 we would put :

(1) $Co_{arith} = \{\bar{0}\}$.

(2) $Fu_{arith} = \{\bar{S}, \bar{+}, \bar{\cdot}\}$

(3) $Re_{arith} = \emptyset$.

(4) $r_{arith}(\bar{S}) = 1, r_{arith}(\bar{+}) = 2,\ r_{arith}(\bar{\cdot}) = 2$.

Then $\sigma_{arith} = (Co_{arith}, Fu_{arith}, Re_{arith}, r_{arith})$.

Signature of Arithmetic

Later we will use this signature to talk about the natural numbers. Then $\bar{0}$ will name the number 0, and the function symbols $\bar{+}$, $\bar{\cdot}$ will stand for plus and times. The symbol $\bar{S}$ will stand for the successor function

$$(n \to n + 1).$$

Signature of Groups

Example 5.3.2(b) The *signature of groups*, σ_{group}, has a constant symbol and two function symbols, namely,

(i) a constant symbol ‘e’;

(ii) a binary function symbol ‘$\cdot$’;

(iii) a 1-ary function symbol ‘ $^{-1}$’.

Signature of Linear Order

Example 5.3.2(c) Neither of the previous signatures has any relation symbol; here is one that does. The signature of linear orders, σ_{lo}, has no constant symbols, no function symbols, and one binary relation symbol $<$.

Parsing

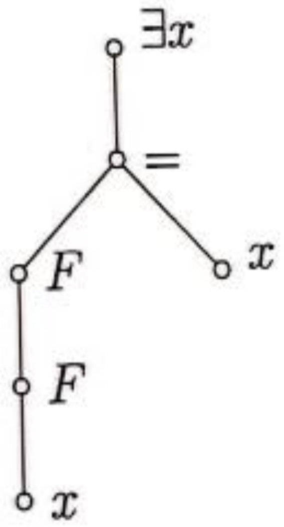
Suppose F is a 1-ary function symbol. Then the expression

(5.15) $$\exists x\ (F(F(x)) = x)$$

is a formula of LR, as will soon become clear. (It expresses that for some element $x, F(F(x))$ is equal to x.) If we parse it, we get the following parsing tree :

Parsing

(5.16)



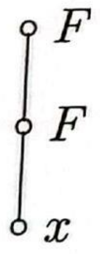
Variables, Parsing tree for terms

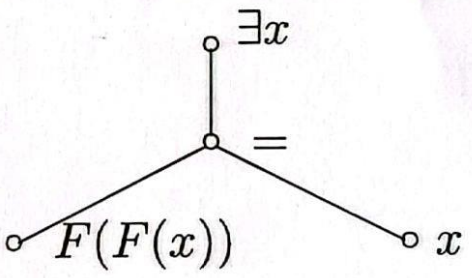
Just as in Chapter 3, we can reconstruct the formula by labelling the nodes of the parsing tree, starting at the leaves and working upwards.

The three nodes in the left-hand branch of the tree are the parsing tree of the term $F(F(x))$. It will make life simpler if we split the syntax of LR into two levels. The lower level consists of the *terms* of LR; the upper level consists of the *formulas*. Most formulas have terms inside them (the formula ⊥ is one of the exceptions). But in first-order logic, terms never have formulas inside them. So we can describe the terms first, and then treat them as ingredients when we go on to build formulas. Thus we split (5.16) into two parsing trees :

Variables, Parsing tree for terms

(5.17)





The first is a parsing tree for a term and the second is a parsing tree for a formula.

Variables, Parsing tree for terms

Definition 5.3.3

(a) The *variables* of LR are the infinitely many symbols

$$x_0, x_1, x_2, \cdots$$

which we assume are not in σ.

(b) Let σ be a signature. A *parsing tree* for terms of LR(σ) is a right-labelled tree where

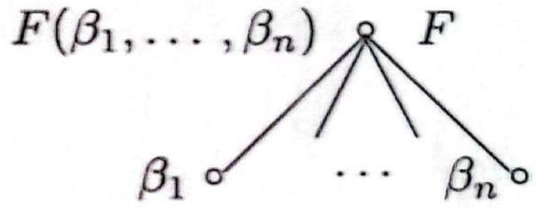
- if a node has arity 0, then its label is either a constant symbol of σ or a variable ;
- if a node has arity $n > 0$, then its label is a function symbol of σ with arity n.

In the heat of battle we will often use x, y, z, etc. as variables of LR, because this is common mathematical notation. But strictly the variables of LR are just the variables in (a) above.

Variables, Parsing tree for terms

We can read a term from its parsing tree by the following compositional definition:

$\alpha \quad \circ \alpha$



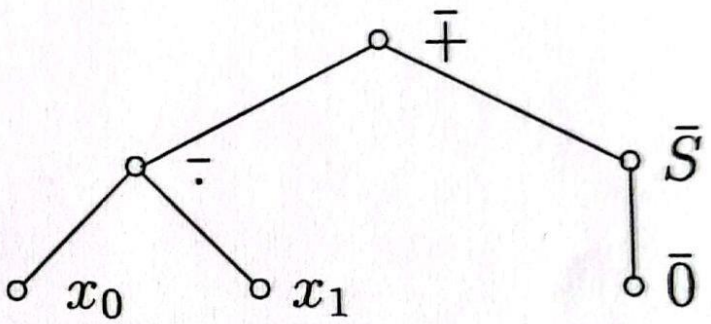
(5.18)

where α is a constant symbol or variable, and F is a function symbol of arity n.

Terms

For example, on the parsing tree

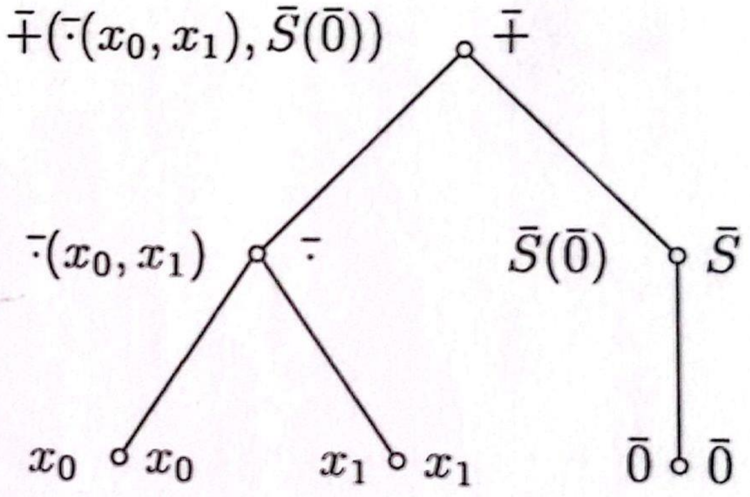
(5.19)



Terms

we build up the following labelling:

(5.20)



The left label on the root node is

$$\bar{+}(\bar{\cdot}(x_0, x_1), \bar{S}(\bar{0})). \tag{5.21}$$

Terms

Definition 5.3.4 Let σ be a signature. Then a *term* of LR(σ) is the associated term of a parsing tree for terms of LR(σ). A *term* of LR is a term of LR(σ) for some signature σ.

For example, the following are terms of LR(σ_{arith}) :

$$\bar{0} \qquad \bar{S}\big(\bar{S}(x_5)\big) \qquad \bar{+}(\bar{S}(\bar{0}), x_2) \qquad \bar{S}(\bar{\cdot}\,(x_4, \bar{0}))$$

In normal mathematical shorthand these would appear as

(5.23) $$0 \qquad S\big(S(x_5)\big) \qquad S(0) \,+\, x_2 \qquad S(x_4 \,\cdot\, 0)$$

Quantifiers

Definition 5.3.5 Let v be a variable. An expression $\forall v$ (read 'for all v') is called a *universal quantifier*. An expression $\exists v$ (read 'there is v') is called an *existential quantifier*. *Quantifiers* are universal quantifiers and existential quantifiers.

Parsing tree for formulas

Definition 5.3.6 Let σ be a signature. A *parsing tree* for formulas of LR(σ) is a right-labelled tree where

• every leaf is labelled with either $\perp$ or a term of LR(σ). if a leaf is labelled by a term, then it has a mother. Equivalently, if a parsing tree has just a single node, then its label must be $\perp$;

• every node that is not a leaf is labelled with one of the symbols $=, \neg, \wedge, \vee, \rightarrow, \leftrightarrow$ or a relation symbol of σ, or a quantifier ;

• every node labelled with a quantifier has arity 1, and its daughter node is not labelled with a term ;

Parsing tree for formulas

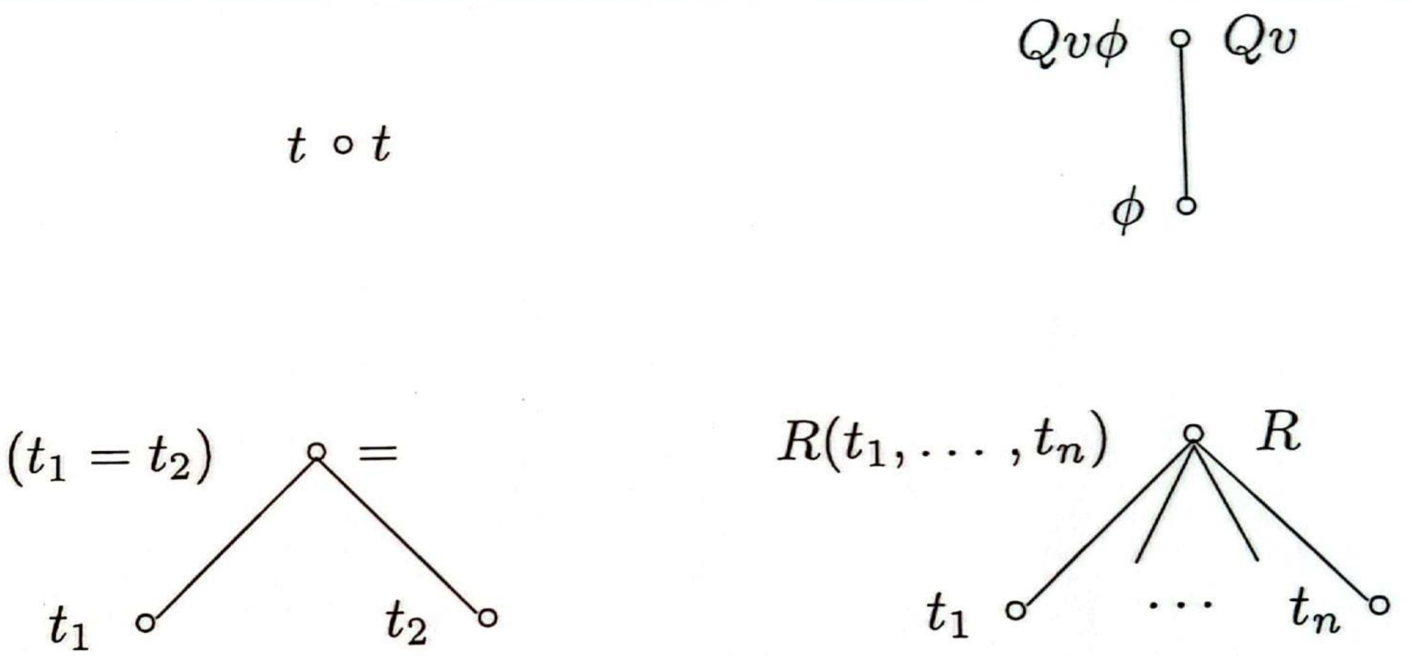
- every node labelled with $\neg$ has arity 1 and its daughter node is not labelled with a term ;
- every node labelled with one of $\wedge, \vee, \rightarrow, \leftrightarrow$ has arity 2 and neither of its daughters is labelled with a term ;
- if a node is labelled with $=$, its arity is 2 and its daughter nodes are labelled with terms ;
- if a node is labelled with a relation symbol R, its arity is the arity of R and its daughter nodes are labelled with terms.

Parsing tree for formulas

The compositional definition for building the associated formula of a parsing tree for formulas is as follows. It includes the clauses of (3.22) for the nodes labelled by truth function symbols, together with four new clauses for the leaves and the nodes labelled by a quantifier, '=' or a relation symbol :

Parsing tree for formulas

(5.24)

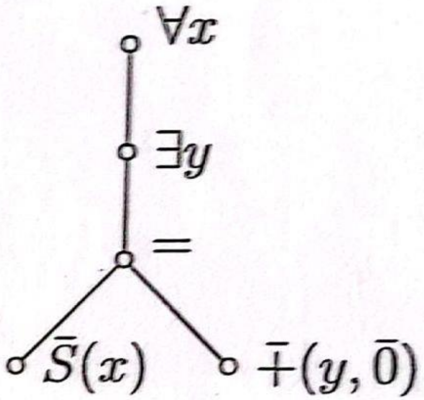


where t is a term, Qv is a quantifier and R is a relation symbol of σ with arity n.

Example

For example, here is an example of a parsing tree for a formula of $\mathrm{LR}(\sigma_{\mathrm{arith}})$:

(5.25)



Example

Applying (5.24) to (5.25) yields the following left labelling (where the right-hand labels are omitted to save clutter):

(5.26)

$$\forall x \exists y(\bar{S}(x) = \bar{+}(y, \bar{0}))$$

$$\exists y(\bar{S}(x) = \bar{+}(y, \bar{0}))$$

$$(\bar{S}(x) = \bar{+}(y, \bar{0}))$$

$$\bar{S}(x) \qquad \bar{+}(y, \bar{0})$$

Example

So the associated formula of (5.25), the label of the root, is

(5.27) $$\forall x\, \exists y\, (\bar{S}(x) = \bar{+}(y, \bar{0}))$$

In normal mathematical practice we would write (5.27) as

(5.28) $$\forall x\, \exists y\, (S(x) = y + 0)$$

As with terms, we regard (5.28) as shorthand for (5.27).

We note another very common shorthand: $s \neq t$ for the formula $\neg(s = t)$.

Formula (formal definition)

Definition 5.3.7 Let σ be a signature. Then a *formula* of LR(σ) is the associated formula of a parsing tree for formulas of LR(σ). A *formula* of LR is a formula of LR(σ) for some signature σ.

If the parsing trees were not of interest to us in their own right, we could boil down the definitions of 'term' and 'formula' to the following inductive definitions in the style of (3.5).

Terms (inductive definition)

First the terms :

(a) Every constant symbol of σ is a term of LR(σ).

(b) Every variable is a term of LR(σ).

(c) For every function symbol F of σ, if F has arity n and $t_1, \cdots, t_n$ are terms of LR(σ) then the expression $F(t_1, \cdots, t_n)$ is a term of LR(σ).

(d) Nothing is a term of LR(σ) except as implied by (a), (b), (c).

Formulas (inductive definition)

Then the formulas :

(a) If s, t are terms of LR(σ) then the expression $(s = t)$ is a formula of LR(σ).

(b) If R is a relation symbol of arity n in σ, and $t_1, \cdots, t_n$ are terms of LR(σ), then the expression $R(t_1, \cdots, t_n)$ is a formula of LR(σ).

(c) $\perp$ is a formula of LR(σ).

(d) If ϕ and ψ are formulas of LR(σ), then so are the following expressions :

$$(\neg\phi) \qquad (\phi \wedge \psi) \qquad (\phi \vee \psi) \qquad (\phi \to \psi) \qquad (\phi \leftrightarrow \psi)$$

(e) If ϕ is a formula of LR(σ) and v is a variable, then the expressions $\forall v\phi$ and $\exists v\phi$ are formulas of LR(σ).

(f) Nothing is a formula of LR(σ) except as implied by (a)–(e).

Unique parsing for LR ********

If someone gives us a term or a formula of LR(σ), then we can reconstruct its parsing tree by starting at the top and working downwards. This is intuitively clear, but as in Section 3.3 we want to prove it by showing a calculation that makes the construction automatic.

The first step is to find the *head*, that is, the symbol or quantifier that comes in at the root node. For terms this is easy: the head is the leftmost symbol in the term. If this symbol is a function symbol F, then the signature tells us its arity n, so we know that the term has the form $F(t_1, \cdots, t_n)$. But this does not tell us how to chop up the string $t_1, \cdots, t_n$ to find the terms t_1, t_2, etc. It is enough if we can locate the $n - 1$ commas that came with F, but this is not trivial since the terms t_1, etc. might contain commas too. Fortunately, the reasoning differs only in details from what we did already with LP, so we make it an exercise (Exercise 5.3.6). The corresponding reasoning for formulas is more complicated but not different in principle. The following theorem guarantees that each term or formula of LR(σ) has a unique parsing tree.

Unique parsing Theorem (Terms)

Theorem 5.3.8 (Unique Parsing Theorem for LR) *Let σ be a signature. Then no term of LR(σ) is also a formula of LR(σ). If t is a term of LR(σ), then exactly one of the following holds :*

(a) t is a variable.

(b) t is a constant symbol of LR(σ).

(c) t is $F(t_1, \cdots, t_n)$, where n is a positive integer, F is a function symbol of σ with arity n, and $t_1, \cdots, t_n$ are terms of LR(σ). Moreover, if t is also $G(s_1, \cdots, s_m)$ where G is an m-ary function symbol of LR(σ) and $s_1, \cdots, s_m$ are terms of LR(σ), then $n = m, F = G$ and each t_i is equal to s_i.

Unique parsing Theorem (Formulas)

If ϕ is a formula of LR(σ), then exactly one of the following is true :

(a) ϕ is $R(t_1, \cdots, t_n)$ where n is a positive integer, R is a relation symbol of σ with arity n, and $t_1, \cdots, t_n$ are terms of LR(σ). Moreover, if ϕ is also $P(s_1, \cdots, s_m)$ where P is an m-ary relation symbol of LR(σ) and $s_1, \cdots, s_m$ are terms of LR(σ), then $n = m, R = P$ and each t_i is equal to s_i.

(b) ϕ is $(s = t)$ for some terms s and t of LR(σ). Moreover, if ϕ is also $(s' = t')$ where s' and t' are terms of LR(σ), then s' is s and t' is t.

(c) ϕ is $\perp$.

Unique parsing Theorem (Formulas)

(d) ϕ *has exactly one of the forms*

$$(\phi_1 \wedge \phi_2)\ (\phi_1 \vee \phi_2)(\phi_1 \rightarrow \phi_2)\ (\phi_1 \leftrightarrow \phi_2),$$

where ϕ_1 *and* ϕ_2 *are uniquely determined formulas.*

(e) ϕ *is* $(\neg\psi)$ *for some uniquely determined formula* ψ*.*

(f) ϕ *has exactly one of the forms* $\forall x\psi$ *and* $\forall\exists x\psi$*, where* x *is a uniquely determined variable and* ψ *is a uniquely determined formula.*

Complexity

In the light of this theorem, we can define properties of terms and formulas by using properties of their parsing trees. For clauses (b) and (c) below, recall Definition 3.2.9.

Definition 5.3.9

(a) The *complexity* of a term is the height of its parsing tree. The complexity of a formula is the height of the parsing tree with leaves labelled by terms, and the corresponding edges, removed. A term or formula with complexity 0 is said to be *atomic* ; all other terms and formulas are said to be *complex*.

Subterms

(b) Let t be a term of LR. Then the *subterms* of t are the traces in t of the left labels on the nodes of the parsing tree of t. The *proper subterms* of t are all the subterms except t itself. The *immediate subterms* of t are the traces coming from the daughters of the root of the parsing tree.

Subformulas

(c) Let ϕ be a formula of LR. Then the *subformulas* of ϕ are the traces in ϕ of the left labels on the nodes of the parsing tree of ϕ with leaves labelled by terms, and the corresponding edges, removed. The *proper subformulas* of ϕ are all the subformulas except ϕ itself. The *immediate subformulas* of ϕ are the traces coming from the daughters of the root of the truncated parsing tree.

Quantifier – free formulas

(d) Let ϕ be a formula of LR. We say that ϕ is *quantifier-free* if $\forall$ and $\exists$ never occur in ϕ (or equivalently, if no nodes of the parsing tree of ϕ are right labelled with quantifiers). We say *qf formula* for *quantifier-free formula*. We say *qf* LR to mean the terms and qf formulas of LR.

Atomic terms and atomic formulas

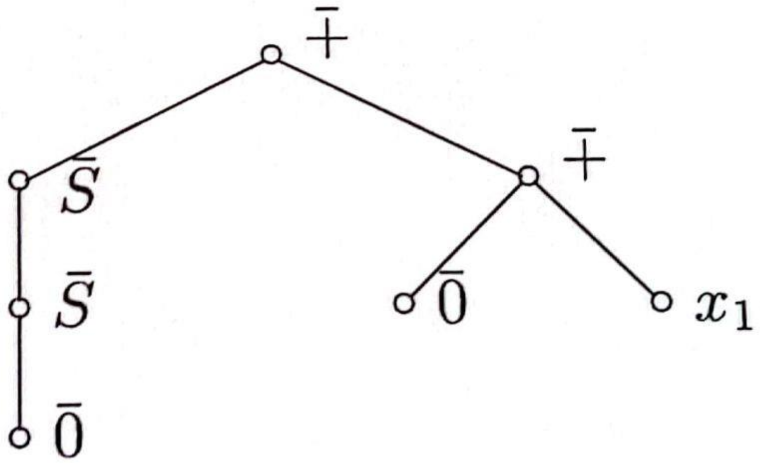
By this definition the atomic terms are exactly those that have no proper subterms ; in other words, they are the variables and the constant symbols. Likewise the atomic formulas are exactly those with no proper subformulas ; in other words, they are $\perp$ and formulas of the forms $(s = t)$ or $R(t_1, \cdots, t_n)$. Every atomic formula is quantifier-free. But there are qf formulas that are not atomic ; the smallest is the formula $(\neg\perp)$.

Example

For example, the expression

$$\bar{+}(\bar{S}(\bar{S}(\bar{0})), \bar{+}(\bar{0}, x_1))$$

is a term of $\mathrm{LR}(\sigma_{\mathtt{arith}})$, and it has the parsing tree



Example

It is a complex term with complexity 3. It has seven subterms, as follows:

$$\bar{+}(\bar{S}(\bar{S}(\bar{0})), \bar{+}(\bar{0}, x_1))$$

(immediate subterm)

(immediate subterm)