

# Unification-Based Grammar

Bob Neveln  
Bob Alps

January 16, 2011

## Abstract

Context-free languages are easily parsed. Language used for the expression of mathematics needs to be unambiguous. A.P. Morse devised a method for generating an essentially context-free mathematical language which formalizes the common practice of using the definitions occurring in a mathematical text as the basis for the mathematical language of that text. Morse obtained the unambiguity and prefix-free properties of any such language by placing syntactic constraints on the set of definienda. Although effective they lacked generality. We show here that greater generality can be achieved using a unification-based constraint.

## 1 Introduction

One of Morse's basic ideas about mathematical syntax was that any well done set of mathematical definitions implicitly defines the grammar of a formal language. Morse's syntax is essentially a method for translating a set of mathematical definitions into a grammar together with a set of conventions about the way that bound variables work. The role of a grammar is to distinguish the well-formed expressions of a language from all other expressions. The determination of the scopes of bound variables is an additional task, one for which these grammars do not suffice. In Section 7 conventions for this purpose are established. It is also shown there that the presence of bound variables can add a layer to the notion of "unambiguous" which is not definable in terms of context-free grammars.

## 2 The Symbols

Given that mathematical language is to be built using a set of (terminal) symbols  $\Sigma$  which is a disjoint union  $\Sigma = \mathcal{C} \cup \mathcal{V} \cup \mathcal{P} \cup \mathcal{U}$  of:

- $\mathcal{C}$  a set of constants
- $\mathcal{V}$  a set of object variables
- $\mathcal{P}$  a set of (second order) predicate variables
- $\mathcal{U}$  a set of (second order) function variables.

The elements of  $\mathcal{P} \cup \mathcal{U}$  are also referred to as *schemators*. We further assume that there is an arity function  $a : \mathcal{P} \cup \mathcal{U} \rightarrow \mathbf{N}$ , which assigns a natural number to each schemator. The arity of a schemator  $p$  may be zero only if  $p \in \mathcal{P}$  and in this case  $p$  is called a sentence variable. A schemator of arity  $n$  followed by  $n$  object variables is called a *schematic expression*.

### 3 Grammars

We need a set  $N = \{S, F, T, V\}$  of four non-terminal symbols,  $S$  for “start,”  $F$  for “formula,”  $T$  for “term,” and  $V$  for “variable”, a set of *formula signatures*  $\mathcal{S}_F$ , a set of *term signatures*  $\mathcal{S}_T$ , such that the set of *signatures*  $\mathcal{S} = \mathcal{S}_F \cup \mathcal{S}_T \subset (\mathcal{C} \cup \{F, T, V\})^*$ , and a set  $R$  consisting of the following production rules:

1.  $S \rightarrow F$
2.  $S \rightarrow T$
3.  $F \rightarrow f$ , for each  $f \in \mathcal{S}_F$
4.  $F \rightarrow p \underbrace{T \dots T}_{n \text{ terms}}$  for each  $p \in \mathcal{P}$  and where the arity of  $p$  is  $n$
5.  $T \rightarrow t$ , for each  $t \in \mathcal{S}_T$
6.  $T \rightarrow u \underbrace{T \dots T}_{n \text{ terms}}$  for each  $u \in \mathcal{U}$  and where the arity of  $u$  is  $n$
7.  $T \rightarrow V$
8.  $V \rightarrow v$ , for each  $v \in \mathcal{V}$

Since without loss of generality we may assume that  $(V \notin \mathcal{S}_T)$  we see that the number of rules in  $R$ ,  $\#R = \#\mathcal{S}_F + \#\mathcal{P} + \#\mathcal{S}_T + \#\mathcal{U} + 3$ . The grammar  $(N, \Sigma, R, S)$  generates the language. The formulas of the language are generated by the grammar  $(N, \Sigma, R, F)$  and the terms by  $(N, \Sigma, R, T)$ .<sup>1</sup> Notice that the schemators do not occur in signatures. Further, as will be seen later  $V$  occurs in signatures only to mark index variable locations. In section 8 it will be shown how the set of signatures  $\mathcal{S}$  is obtained from the definitions and primitive terms and formulas of the language.

For each signature,  $b$  with  $n$  occurrences of the non-terminals  $F, T$ , and  $V$  we may form a term or formula by replacing these non-terminals by formulas, terms, and variables. We may represent the resulting term or formula by  $b(f_1, \dots, f_n)$  where each  $f_i$  is a formula, term, or variable replacing the  $i$ -th occurrence in  $b$  of a non-terminal  $F, T$ , or  $V$  respectively. We note that constants can appear at the beginning of terms and formulas only by virtue of the rules in 5 and 3 above.

**Theorem** If  $t$  is any term or formula beginning with a constant then  $t$  can be represented as  $b(f_1, \dots, f_n)$  for some signature  $b$  and components  $f_i$ .

---

<sup>1</sup>Morse’s syntax made no distinction between terms and formulas. To describe his grammars using this formalism, we should do away with the non-terminal symbols  $S$  and  $F$ , delete the rules in 1 through 4 and make  $T$  the initial symbol.

## 4 Two Desirable Properties

When a language is generated from such a grammar the distinguishing features of the language result from the kinds of expressions which constitute  $\mathcal{S}$ . For example a prefix (or Polish) style of syntax results if each signature  $b \in \mathcal{S}$  begins with a unique constant. Among the desirable properties of a language with a prefix grammar are that it is:

1. Unambiguous. Each term or formula beginning with a constant stems from a unique signature.
2. Prefix-Free. No term or formula of the language begins another.

When a language has a grammar of the type given in section 3 the property of being unambiguous, which is ordinarily stated in terms of the uniqueness of a leftmost derivation, can be stated as follows:

**Definition of Unambiguous** If  $b(f_1, \dots, f_n)$  is  $c(g_1, \dots, g_m)$  then  $b$  is  $c$ , ( $m = n$ ), and  $f_i$  is  $g_i$  for  $i = 1, \dots, n$ .

The prefix-free property can be stated as follows:

**Definition of Prefix-Free** If  $b(f_1, \dots, f_n)$  is an initial segment of  $c(g_1, \dots, g_m)$  then  $b(f_1, \dots, f_n)$  is  $c(g_1, \dots, g_m)$ .

In [3], Morse obtained conditions on an arbitrary set of signatures sufficient to guarantee that the resulting language would have these two properties.<sup>2</sup> We seek conditions which are necessary as well as sufficient.

## 5 Conditions on the Set of Signatures

We have that all signatures begin with some constant. Conversely we define an *introducer* as a constant that occurs as the initial symbol of some signature.

The following two conditions were considered by Morse:

1. No signature may be the initial segment of some other signature, nor is this allowed if the  $V$ 's in the signatures are changed to  $T$ 's.
2. No introducer may occur in a signature except as the initial symbol.

To get the two desired properties of section 4 the first of these two conditions is clearly necessary. The second however is so strong that it rules out the standard absolute value notation. Morse modified this condition in an ad hoc way to accommodate the absolute value sign using a notion of *flanker*. The resulting condition, although weaker, still rules out some rather standard usages such as that of parentheses in the power set notation  $\mathcal{P}(X)$  where, because it is an introducer, the left parenthesis is disallowed from the position it occupies in this notation. More in spirit

---

<sup>2</sup>Morse's constraints are given on pages 154-155 of [4] and pages 113-114 of [3]. The sufficiency of these constraints is shown in [5].

with the rest of Morse's work would be the determination of conditions only as strong as necessary.

Given two signatures  $b$  and  $c$ , we seek to determine whether there are  $f_1, \dots, f_n$  and  $g_1, \dots, g_m$  so that  $b(f_1, \dots, f_n)$  is an initial segment of  $c(g_1, \dots, g_m)$ . If so we say that  $b$  *prefix-unifies* with  $c$ . Almost by definition then we have the following theorem.

**Theorem** If no signature prefix-unifies with another signature then the resulting language is unambiguous and prefix-free.

The value of the prefix-unification concept depends on its effective determinacy.

## 6 Prefix-Unification Algorithm

We begin by defining concepts that are used in the algorithm.

For any expression  $s$  of length  $n$ , we use  $s[k]$  to denote the  $k$ -th symbol of  $s$ , where  $1 \leq k \leq n$ .

A *signature-list* is a list  $[b, m_1, \dots, m_k]$  whose first element is a signature and such that if the length of  $b$  is  $n$  there are  $k$  additional entries in the list where  $1 \leq k \leq n$ . If  $k = n$ , we say the signature-list is *complete*; otherwise it is *incomplete*.

We recursively define a *signature match* as a signature-list  $[b, m_1, \dots, m_k]$ , where for each  $i \in \{1, \dots, k\}$ :

1. if  $b[i] = V$  or  $b[i] \in \mathcal{C}$  then  $m_i$  is  $b[i]$ ,
2. if  $b[i] = T$  then  $m_i$  is either  $V$  or  $T$  or a signature match that is a complete signature-list with a term signature as the first item of the list, and
3. if  $b[i] = F$  then  $m_i$  is either  $F$  or a signature match that is a complete signature-list with a formula signature as the first item of the list.

A *complete signature match* is a signature match that is a complete signature-list. An *incomplete signature match* is a signature match that is an incomplete signature-list.

A *partial-unification list* is a list of signature matches each of which, except for possibly the last, is incomplete.

A partial-unification list is *reduced* if its final signature match is incomplete or if its length is 1.

The *reduction* of a partial-unification list  $A$  is the result of applying the following algorithm to  $A$ :

Repeat:

If  $A$  is reduced, then halt the algorithm and return  $A$ .

Otherwise, let the last two signature matches of  $A$  be  $[s, m_1, \dots, m_j]$  and  $M$  where  $M$  is a complete signature match, and remove both matches from the list  $A$  and append the signature match  $[s, m_1, \dots, m_j, M]$  to  $A$ .

A *partial unification* is an ordered pair  $(A, B)$  of partial-unification lists.

In the following prefix-unification algorithm a set  $X$  of partial unifications is maintained.  $X$  changes as the algorithm proceeds. If  $X$  becomes empty, then the algorithm has shown that no unification exists and it returns 0. If a prefix-unification is found then 1 is returned.

Using the above definitions, the algorithm may now be described.

The input to the algorithm consists of two signatures  $b$  and  $c$  having the same initial symbol  $\alpha_0$ .

Let  $A_0 = [[b, \alpha_0]]$ .

Let  $B_0 = [[c, \alpha_0]]$ .

Let  $X = \{(A_0, B_0)\}$ .

Repeat:

If  $X = \emptyset$  halt the algorithm and return 0.

Let  $Y = \emptyset$ .

For each partial unification  $(A, B) \in X$ ,

Let  $A'$  be the reduction of  $A$ .

Let  $B'$  be the reduction of  $B$ .

If either  $A'$  or  $B'$  ends with a complete signature match then halt the algorithm and return 1.

Otherwise add the pair  $(A', B')$  to  $Y$ .

Let  $X = Y$ .

Let  $Y = \emptyset$ .

For each partial unification  $(A, B)$  in  $X$ ,

Let  $[s, \ell_1, \dots, \ell_j]$  be the final signature match of  $A$ .

Let  $[t, m_1, \dots, m_k]$  be the final signature match of  $B$ .

Let  $\alpha = s[j + 1]$ .

Let  $\beta = t[k + 1]$ .

Since every symbol of a signature is one of  $V, T, F$ , or a constant, we can cover all possibilities for  $\alpha$  and  $\beta$  in 16 cases. Note that the first two numbered steps below cover 4 of the 16 cases.

1. If  $\alpha = \beta$  then:

Let  $A'$  be  $A$  with the final signature match of

$A$ ,  $[s, \ell_1, \dots, \ell_j]$ , replaced by  $[s, \ell_1, \dots, \ell_j, \alpha]$ .

Let  $B'$  be  $B$  with its final signature match

$[t, m_1, \dots, m_k]$  replaced by  $[t, m_1, \dots, m_k, \beta]$ .

Add the pair  $(A', B')$  to the set  $Y$ .

2. If  $\alpha, \beta \in \mathcal{C}$  and  $\alpha \neq \beta$  then make no change to  $Y$  since no extension is possible.

3. If  $\alpha = T$  and  $\beta = F$  or  $\alpha = F$  and  $\beta = T$  then make no change to  $Y$  since no extension is possible.

4. If  $\alpha = V$  and  $\beta \in \{F\} \cup \mathcal{C}$  or  $\beta = V$  and  $\alpha \in \{F\} \cup \mathcal{C}$  then make no change to  $Y$  since no extension is possible.

5. If  $\alpha = T$  and  $\beta = V$  or  $\alpha = V$  and  $\beta = T$  then:

- Let  $A'$  be  $A$  with the final signature match of  $A$ ,  $[s, \ell_1, \dots, \ell_j]$ , replaced by  $[s, \ell_1, \dots, \ell_j, V]$ .  
Let  $B'$  be  $B$  with its final signature match  $[t, m_1, \dots, m_k]$  replaced by  $[t, m_1, \dots, m_k, V]$ .  
Add the pair  $(A', B')$  to the set  $Y$ .
6. If  $\alpha = T$  and  $\beta \in \mathcal{C}$  then for each term signature  $u \in \mathcal{S}_T$ :  
If the initial symbol of  $u$  is  $\beta$ :  
Let  $A'$  be  $A$  with  $[u, \beta]$  appended.  
Let  $B'$  be  $B$  with the final signature match replaced by  $[t, m_1, \dots, m_k, \beta]$ .  
Add the pair  $(A', B')$  to the set  $Y$ .
  7. If  $\alpha = F$  and  $\beta \in \mathcal{C}$  then for each formula signature  $u \in \mathcal{S}_F$ :  
If the initial symbol of  $u$  is  $\beta$ :  
Let  $A'$  be  $A$  with  $[u, \beta]$  appended.  
Let  $B'$  be  $B$  with the final signature match replaced by  $[t, m_1, \dots, m_k, \beta]$ .  
Add the pair  $(A', B')$  to the set  $Y$ .
  8. If  $\beta = T$  and  $\alpha \in \mathcal{C}$  then for each term signature  $u \in \mathcal{S}_T$ :  
If the initial symbol of  $u$  is  $\alpha$ :  
Let  $B'$  be  $B$  with  $[u, \alpha]$  appended.  
Let  $A'$  be  $A$  with the final signature match replaced by  $[s, \ell_1, \dots, \ell_k, \alpha]$ .  
Add the pair  $(A', B')$  to the set  $Y$ .
  9. If  $\beta = F$  and  $\alpha \in \mathcal{C}$  then for each formula signature  $u \in \mathcal{S}_F$ :  
If the initial symbol of  $u$  is  $\alpha$ :  
Let  $B'$  be  $B$  with  $[u, \alpha]$  appended.  
Let  $A'$  be  $A$  with the final signature match replaced by  $[s, \ell_1, \dots, \ell_k, \alpha]$ .  
Add the pair  $(A', B')$  to the set  $Y$ .

Let  $X = Y$ .

Clearly this algorithm terminates only under favorable conditions. We seek enhancements to this algorithm, checking for repeated states, which will allow more partial unifications to be discarded.

## 7 Bound Variables

Bound variables cause a problem in creating context free grammars for Morse languages because Morse's rules require that all the bound variables of a definiendum be distinct. We must first note in passing that definienda (as opposed to composite expressions) with more than a single bound variable are not that common in mathematics. For example, mathematics

actually carried out in first order logic has none at all since the only bound variables used in a strict first order language occur in quantification and these have single not multiple occurrences. Nonetheless to illustrate the point suppose the following expression is used as a definiendum for a triple sum: ‘ $\sum x, y, z \in A \times B \times C \underline{u}''xyz$ ’ In this expression ‘ $x$ ’, ‘ $y$ ’, and ‘ $z$ ’ are bound. So according to Morse’s treatment the expressions

- ‘ $\sum t, t, z \in A \times B \times C \underline{u}''ttz$ ’
- ‘ $\sum t, y, t \in A \times B \times C \underline{u}''tyt$ ’
- ‘ $\sum x, t, t \in A \times B \times C \underline{u}''xtt$ ’
- ‘ $\sum t, t, t \in A \times B \times C \underline{u}''ttt$ ’

cannot be defined, whereas if ‘ $\sum V, V, V \in T \times T \times TT$ ’ is the signature corresponding to the original triple sum definiendum then these other expressions **must** be defined since each occurrence of  $V$  is replaced independently in a context-free grammar. We resolve this difficulty by requiring that these additional forms receive definitions.<sup>3</sup> Because of the way that bound variable replacement is defined these additional forms are completely independent of each other and of the original so that the mathematician is free to define them in whatever way is convenient. Of the definienda corresponding to a signature, exactly one will have distinct index variables. We shall call this the principal definiendum.

For principal definienda we require that:

1. All the schemators in the definiendum occur just once.
2. Each schemator occurs as the initial symbol of a schematic expression.
3. All the variables which occur in some schematic expression occur exactly once not in a schematic expression and these occurrences correspond to occurrences of  $V$  in the signature. These variables we refer to as the *index* variables.
4. All the remaining variables occur just once.

We further require that definienda which are not principal be obtained from the principal definiendum by substituting some index variable for other index variables.

Each of the four conditions above is included in Morse’s rules. His version of the third condition was stronger in that he required that each bound variable occur in each schematic expression. Our weakened condition has notable consequences, as noted below.

The problem of defining the scope of each bound variable remains. In standard first order logic the scope of a bound variable is simply the entire form in which it occurs. As noted by Morse actual mathematical practice is not always this simple. For example formulas such as

$$\int_0^x xdx = \int_0^x ydy$$

---

<sup>3</sup>If there are  $n$  occurrences of a bound variable then the number of these forms is  $B(n)$ , the Bell number. The first few Bell numbers are:  $B(1) = 1, B(2) = 2, B(3) = 5, B(4) = 15$ .

are often accepted without question. Morse accommodated this practice by localizing the scope of a bound variable so that it did not necessarily include the entire form. We proceed to localize it further.

The scope of each bound variable of the definiendum is by convention the set of schematic expressions in which that variable occurs.

Applying this convention to the triple sum definiendum we see that all the index variables occur in a single schematic expression. This expression therefore constitutes their shared scope. The variables ‘ $A$ ’, ‘ $B$ ’, and ‘ $C$ ’ are not included in the scope. Consequently we may infer that

$$\sum x, y, z \in x \times y \times z \underline{u}''xyz \equiv \sum u, v, w \in x \times y \times z \underline{u}''uvw$$

from

$$\sum u, v, w \in x \times y \times z \underline{u}''uvw \equiv \sum u, v, w \in x \times y \times z \underline{u}''uvw$$

simply by virtue of indicial variable substitution.

The rule of inference for indicial variable substitution may be stated as follows:

**Indicial Variable Substitution Rule**

If  $A$  is a constituent of a theorem  $T$  and  $\alpha$  is an index variable of  $A$ , and  $\beta$  is a variable which does not occur in the scope of  $\alpha$  in  $A$  and  $B$  is obtained from  $A$  by replacing  $\alpha$  by  $\beta$  and  $T'$  is obtained from  $T$  by replacing  $A$  by  $B$  then  $T'$  is a theorem.

Incidentally this rule provides a different solution to the following problem raised by Morse: <sup>4</sup> Suppose that ‘ $(\bigwedge *xy\underline{u}x\underline{v}y \equiv \bigwedge x \bigwedge y(\underline{u}x \rightarrow \underline{v}y))$ ’ were allowed as a definition. In this case the formula ‘ $\bigwedge *xy\underline{u}'xy\underline{v}'xy$ ’ could not be translated using that definition. However according to the rule just given this formula translates as follows: ‘ $\bigwedge s \bigwedge t(\underline{u}'sy \rightarrow \underline{v}'xt)$ ’ where ‘ $x$ ’ and ‘ $y$ ’ are free variables in the translated formula. Using this rule the need for Morse’s rule A.9 disappears. The argument however is important in that it justifies all four of the rules stated in section 7.

## 8 From Basic Forms to Signatures

In section 1 Morse’s syntax was described as a method of obtaining a grammar from a set of definitions. In section 3 the process of deriving a grammar from the set of signatures was described. We now describe how the set of signatures is obtained from a set of definitions.

We begin with the left sides of all definitions. These we refer to as definienda. To these we add all primitive or undefined terms and formulas. The resulting set we call the set of basic forms. The indicial variables of a basic form are the variables occurring in some schematic expression of the form. To obtain the signature of a basic form we replace each schematic expression by T or F depending on whether its initial symbol is in  $\mathcal{U}$  or  $\mathcal{P}$ . Of the remaining variables the indicial variables are replaced by  $V$  and the others by  $T$ . The result is the signature.

---

<sup>4</sup>See page 161 of [4] or page 119 or [3].

## 9 Conclusion

A.P. Morse devised rules for generating an essentially context-free dynamic mathematical language, permitting the language to expand, according to common practice, through the addition of definitions. In order to obtain the unambiguity and prefix properties of such a language, Morse included rules that placed syntactic constraints on the set of definienda. These constraints involved restrictions on the use of constant symbols which were used as the first symbol of some definiendum. Although effective, they lacked generality. We show here that greater generality can be achieved using a unification-based constraint.

## References

- [1] R.A. Alps. *A Translation Algorithm for Morse Systems*, PhD dissertation, Northwestern University, 1979.
- [2] R.A. Alps and R.C. Neveln, A Predicate Logic Based on Indefinite Description and Two Notions of Identity. *Notre Dame Journal of Formal Logic* 22(3) 1981.
- [3] A.P. Morse. *A Theory of Sets* Academic Press 1965.
- [4] A.P. Morse. *A Theory of Sets* Second Edition. Academic Press 1984.
- [5] R.C. Neveln. *Basic Theory of Morse Languages*, PhD dissertation, Northwestern University, 1975.
- [6] Bob Neveln and Bob Alps. *Writing and Checking Complete Proofs in T<sub>E</sub>X* PracJourn 1- 2007.