*The **Modifield** BNF for Wren*

The Wren BNF presented in the text is, unfortunately, not suitable for parsing using a top-down approach. I have modified the grammar slightly to make it suitable. There remains one problem area that is shaded in gray below, so I will give you that non-terminal function.

```
<program>   ::=   program IDENTIFIER is <block>

<block>     ::=   <decseq> begin <commandfqseq> end

<decseq>    ::=   <dec> <decseq>    |     λ

<dec> ::=   var <varlist> : <type> ;

<type>      ::=   integer     |     boolean

<varlist>   ::=   IDENTIFIER  |     IDENTIFIER , <varlist>

<commandseq>      ::=   <command>   |     <command> ; <command seq>

<command>   ::=   <assign>
            |     skip
            |     read IDENTIFIER
            |     write <intexpr>
            |     while <boolexpr> do <commandseq> end while
            |     if <boolexpr> then <commandseq> end if
            |     if <boolexpr> then <commandseq> else
                  <commandseq> end if

<assign>    ::=   IDENTIFIER := <intexpr>
            |     IDENTIFIER :=: <boolexpr>

<intexpr>   ::=   <intterm>   |     <intexpr> <weak_op> <intterm>

<intterm>   ::=   <element>   |     <intterm> <strong_op> <element>

<intelement>      ::=   NUMERAL
                  |     IDENTIFIER
                  |     ( <intexpr> )
                  |     - <intelement>

<boolexpr>  ::=   <boolterm>
            |     <boolexpr> or <boolterm>

<boolterm>  ::=   <boolelement>
            |     <boolterm> and <boolelement>

<boolelement>     ::=   true
                  |     false
                  |     not [ <boolexpr> ]
                  |     [ <boolexpr> ]
                  |     IDENTIFIER
                  |     <intexpr> <relation> <intexpr>

<relation>  ::=   <=  |  <  |  =  |  <>  |  >=  |  >

<weak op>   ::=   +     |     -

<strong op> ::=   *     |     /
```