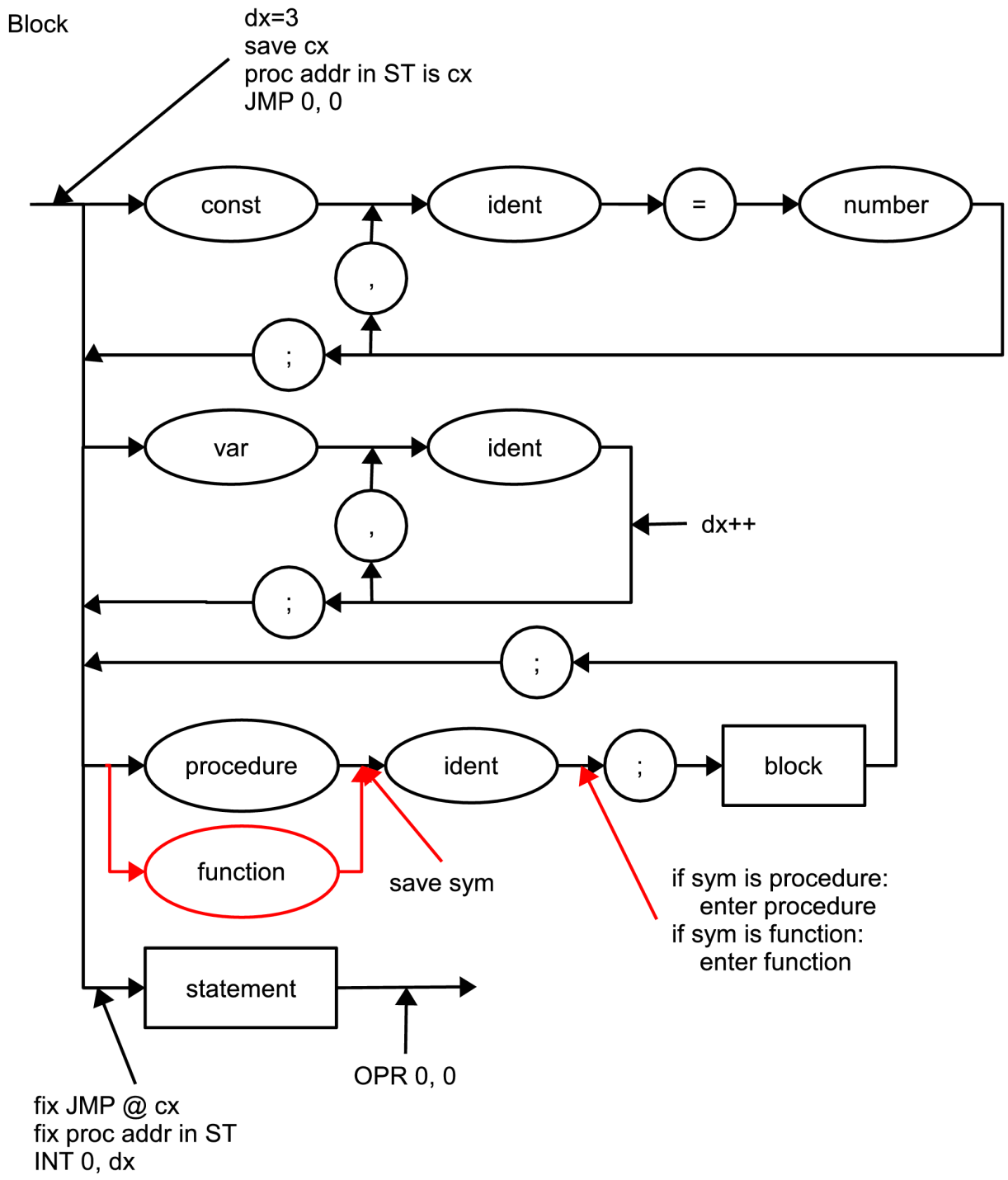
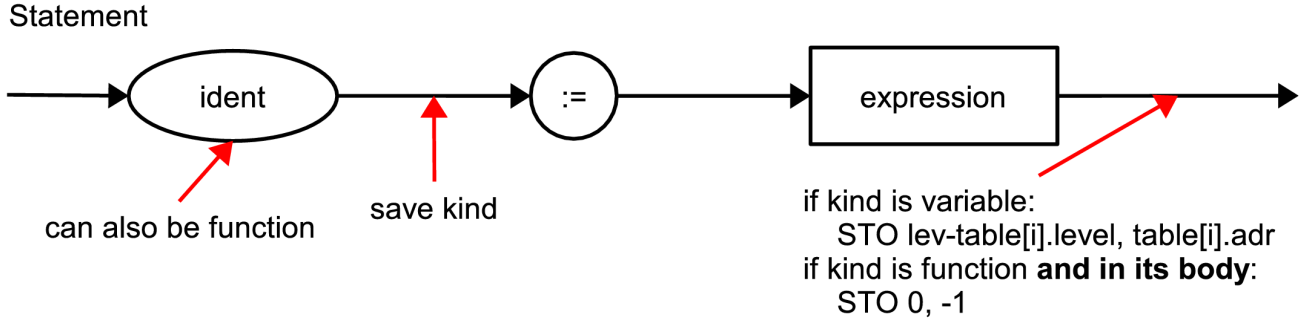
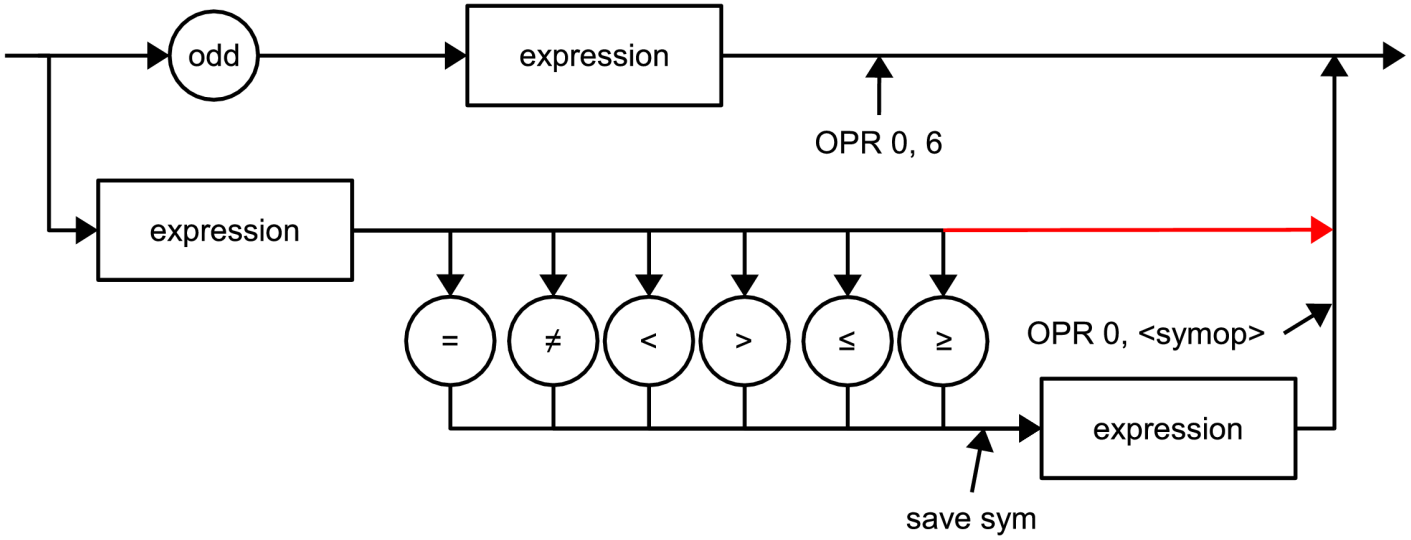
Syntax Diagrams (adding Functions, AND, OR, and NOT)

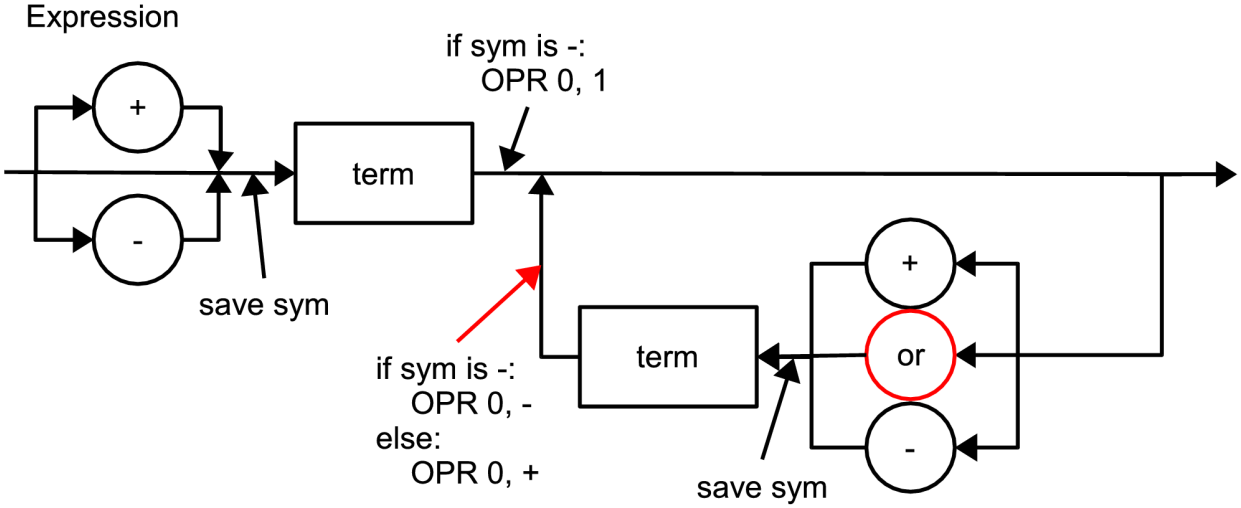




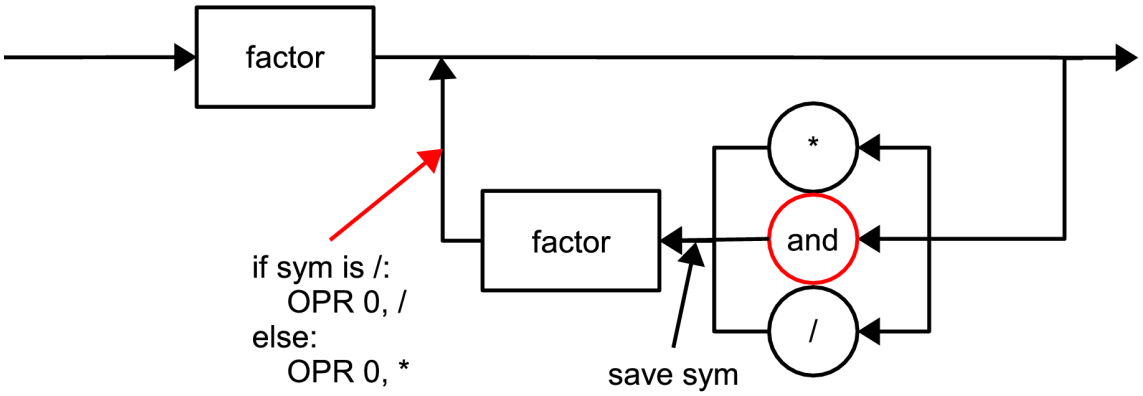
General Expression

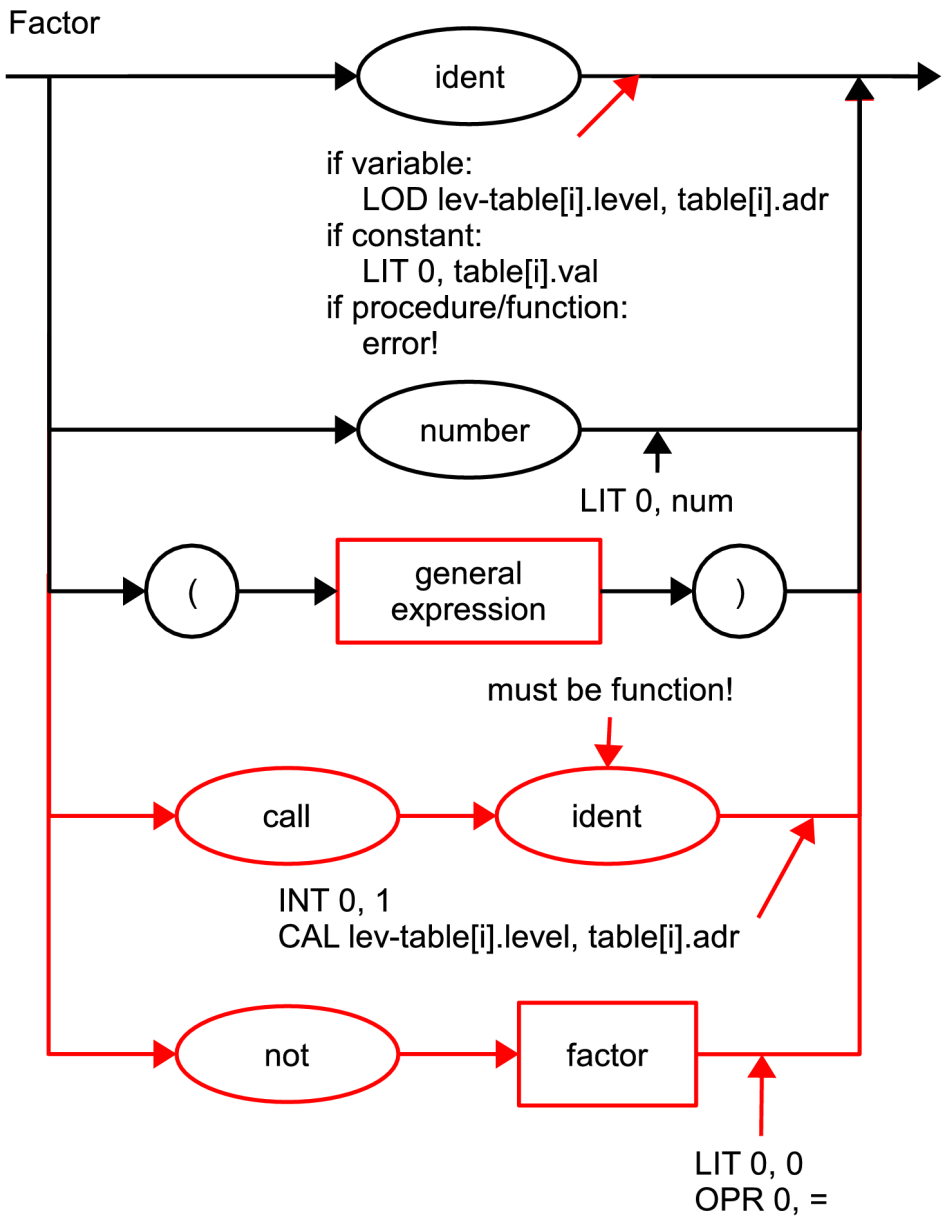


Expression



Term





for all additions
- add the reserved word
- add the symbol

functions
- add the ST type
- allow declaring functions in block
- allow calling functions in factor
  - ident must be function
  - we'll increase the stack to make space for the return value
  - consider this making a little attic
- allow returning values in statement (like assigning to a variable)
  - but we must be in the right function!
  - we'll return the value “under the carpet”
    - that's the “attic” space we made earlier in the call

AND
    add in term (same operation as TIMES)
OR
    add in expression (same operation as PLUS)
NOT
    add in factor
    we just push 0 on the stack and check for equality
        0 -> 1
        1 -> 0
but to add AND, OR, and NOT, we must somehow accept “compound conditions”
    let's call this a general expression instead
    so we rename condition to generalexpression (and fix all current calls to condition)
    we also allow the option of having no relation in the second branch of generalexpression
    then we redirect (<expression>) in factor to (<generalexpression>)
        this now allows “compound conditions”
        but we'll have to get used to adding the parentheses to the PL/0 source code