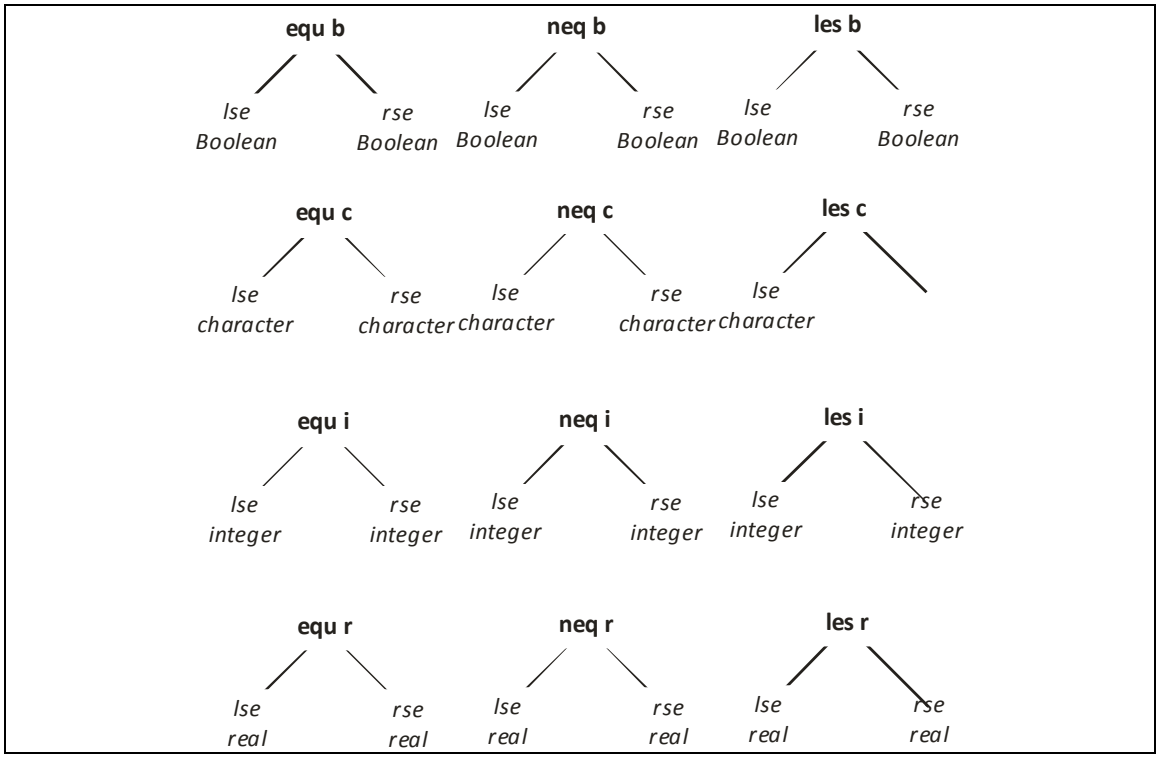
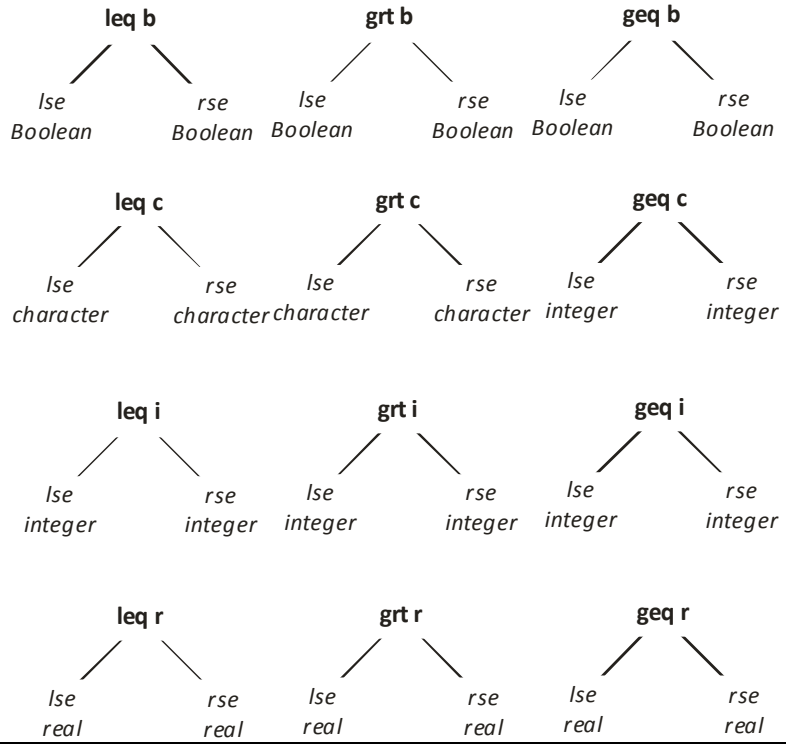
40	*expression*	→	*simple_expression*
41	*expression*	→	*simple_expression relop simple_expression*
42	*relop*	→	**=**
43	*relop*	→	**<>**
44	*relop*	→	**<**
45	*relop*	→	**<=**
46	*relop*	→	**>**
47	*relop*	→	**>=**

40	*expression*	→	*simple_expression* $$=$1

41 *expression* → *simple_expression relop simple_expression*

```
Exp* expression(Exp* lse,string* relop,Exp* rse)
    if *relop="="    return expression(lse,rse,"equ")
    if *relop="<>"   return expression(lse,rse,"neq")
    if *relop="<"    return expression(lse,rse,"les")
    if *relop="<="   return expression(lse,rse,"leq")
    if *relop=">"    return expression(lse,rse,"grt")
    if *relop=">="   return expression(lse,rse,"geq")
    else error("invalid relop")
```





Exp expression*(*Exp* l*,*Exp* r*,*string relop*)

1. Both expression *l* and *r* must have the same type.
2. Create PCode P with the following
   2.1. Label: “”
   2.2. PCode Op: parameter *relop*.
   2.3. Operand 1: Use the TypeChar for the Type of operand, parameter *l* or *r*.
   2.4. Operand 2: “”
3. Create an expression E with the following
   3.1. Left sub expression: parameter l.
   3.2. Right sub expression: parameter r.
   3.3. PCode: PCode P created in step 2 above
   3.4. Type: Boolean
4. Return expression E.