```
<exp>1 ::= <exp>2 - <term>

            if S(<exp>2) = rA then
               generate [SUB    S(<term>)]
            else
               begin
                  GETA (<exp>2)
                  generate [ SUB    S(<term>)]
               end
            S(<exp>1) := rA
            REGA := <exp>1


<term> ::= <factor>

            S(<term>) := S(<factor>)
            if S(<term>) = rA then
               REGA := <term>


<term>1 ::= <term>2 * <factor>

            if S(<term>2) = rA then
               generate [MUL    S(<factor>)]
            else if S(<factor>) = rA then
               generate [MUL    S(<term>2)]
            else
               begin
                  GETA (<term>2)
                  generate [MUL   S(<factor>)]
               end
            S(<term>1) := rA
            REGA := <term>1


<term>1 ::= <term>2 DIV <factor>

            if S(<term>2) = rA then
               generate [DIV    S(<factor>)]
            else
               begin
                  GETA (<term>2)
                  generate [DIV    S(<factor>)]
               end
            S(<term>1) := rA
            REGA := <term>1
```

Figure 5.19 (*cont'd*)