```
#Title: FormalLimitToInfinity
#Author: Gord Clement, July 2010
#Description: For a given function, point and N value,
     this procedure calculates the infinite limit of the function at the given point,
#                  the maximum delta value for the finite limit and displays an animation demonstrating
#                  all possible delta values
#Usage:
#Call: FormalLimitAtInfinity( function, variable, point, N )
#function: function to use in animation
#variable: the variable the limit is taken with respect to
#point: point at which the limit is taken (numeric)
#N : N value used to calculate delta value
FormalLimitToInfinity := proc(expr, var, pt, N)
    #local variable declarations
    local del, limval, output, xupper, xlower, xrange, i, leftend, rightend;
    #calculate limit
    limval := limit(expr, var=pt);
    #initialize delta
    del := 0;
     #limit is to positive infinity code
     if (limval=infinity) then
         if (N < 0) then
             #Error message
             output := "Error: Limit tends to infinity, but given N is not positive";
         else
             #solve all ranges of x satsifying y > N
             xrange := solve(expr > N) :
             #if only one x-range
             if (nops([xrange]) = 1) then
                 #extract end-points
                 xlower := op(1, xrange);
                 xlower := op(1, xlower);
                 xupper := op(2, xrange);
                 xupper := op(1, xupper);
             #many x-ranges
             else
                 #step through all ranges
                 for i from 1 to nops([xrange]) do
                     #extract end-points
                     leftend := op(1, xrange[i]);
                     leftend := op(1, leftend);
                     rightend := op(2, xrange[i]);
                     rightend := op(1, rightend);
                     #check if limit point is on the boundary
                     if (simplify(pt - leftend) = 0) then
                             xupper := rightend;
                     elif (simplify(pt - rightend) = 0) then
                             xlower := leftend;
                     end if;
                     #check if limit point is in the range
                     if  (is(pt in xrange[i])) then
                             #extract end-points
```

```
                    xlower := op(1, xrange[i]);
                    xlower := op(1, xlower);
                    xupper := op(2, xrange[i]);
                    xupper := op(1, xupper);
                end if
            end do;
        end if;
        #calculate delta
        del := min(abs(xupper - pt), abs(xlower - pt));
        #display animation
        plots[animate](plot, [[[var, expr, var=pt - 3 ·del..pt - 0.01·del], [var, expr, var=pt
 + 0.01 ·del..pt + 3·del], [t, N, t=pt - 3·del..pt + 3·del], [pt - delta, t, t = 1/2 ·N..5 N], [pt
 + delta, t, t=0.5·N..5·N], [pt, t, t = 1/2 ·N..5·N], [t, subs(var=pt - delta, expr), t=pt - delta..pt
 + delta], [t, subs(var=pt + delta, expr), t=pt - delta..pt + delta]], horizontal=pt - 3 ·del..pt
 + 3·del, vertical=0..5·N,  labels=["", ""], thickness=[3, 3, 2, 2, 2, 1, 1, 1], color=[black,
black, red, blue, blue, black, red, red], linestyle=[solid, solid, solid, solid, solid, dash, dash, dash]],
 delta=0..del)
    end if;
  #limit is to negative infinity
  elif (limval=-infinity) then
       if (N ≥ 0) then
          #Error message
          output := "Error: Limit tends to negative infinity, but given N is not negative";
       else
          #find x-range containing the limit point satisfying y < N as before
          xrange := solve(expr < N) :
          if (nops([xrange]) = 1) then
               xlower := op(1, xrange);
               xlower := op(1, xlower);
               xupper := op(2, xrange);
               xupper := op(1, xupper);
          else
               for i from 1 to nops([xrange]) do
                    leftend := op(1, xrange[i]);
                    leftend := op(1, leftend);
                    rightend := op(2, xrange[i]);
                    rightend := op(1, rightend);
                    if (simplify(pt - leftend) = 0) then
                         xupper := rightend;
                    elif (simplify(pt - rightend) = 0) then
                         xlower := leftend;
                   end if;
                    if  (is(pt in xrange[i])) then
                       xlower := op(1, xrange[i]);
                       xlower := op(1, xlower);
                       xupper := op(2, xrange[i]);
                       xupper := op(1, xupper);
```

```
                    end if
                end do;
            end if;
            #calculate delta
            del := min(abs(xupper − pt), abs(xlower − pt));
            #display animation
            plots[animate](plot, [[[var, expr, var=pt − 3 ·del..pt − 0.01·del], [var, expr, var=pt
 + 0.01 ·del..pt + 3·del], [t, N, t=pt − 3·del..pt + 3·del], [pt − delta, t, t=5·N..0.5 N], [pt
 + delta, t, t=5·N..0.5·N], [pt, t, t=5·N..0.5·N], [t, subs(var=pt − delta, expr), t=pt − delta..pt
 + delta], [t, subs(var=pt + delta, expr), t=pt − delta..pt + delta]], horizontal=pt − 3 ·del..pt
 + 3·del, vertical=5·N..0,  labels=["", ""], thickness=[3, 3, 2, 2, 2, 1, 1, 1], color=[black,
black, red, blue, blue, black, red, red], linestyle=[solid, solid, solid, solid, solid, dash, dash, dash]],
 delta=0..del)
        end if;
    else
        #Error message
        output
:= "Error:  Limit must tend to ± infinity, try using FormalLimit or FormalLimitAtInfinity";
    end if:


end proc:
```

$$FormalLimitToInfinity\left(\frac{-1}{(x-1)^2}, x, 1, -100\right)$$