```
(1)     module semi;
(2)         export main(:);
(3)         edb arc(X,Y), path(X,Y);
(4)         prog main(:)
(5)             rels newPath(X,Y);
(6)             arc(X,Y) := {}.
(7)             arc(1,2).
(8)             arc(2,3).
(9)             arc(3,4).
(10)            arc(4,5).
(11)            arc(2,4).
(12)            newPath(X,Y) := arc(X,Y).
(13)            path(X,Y) := {}.
(14)            while newPath(X,Y) do
(15)                path(X,Y) += newPath(X,Y).
(16)                newPath(X,Y) := newPath(X,Z) & arc(Z,Y).
(17)                newPath(X,Y) -= path(X,Y).
(18)            end;
(19)        end
(20)    end
```

Fig. 1. GLUE program to compute paths by seminaive evaluation.

```
(1)     module nim;
(2)         export main(:);
(3)         edb lt(X,Y), win(A,B,C);
(4)         prog main(:)
(5)             win(A,B,C) := interesting(A,B,C) & w(A,B,C).
(6)         end
(7)         interesting(A,B,C) :- lt(8,A) & lt(6,B) & lt(4,C).
(8)         move(A,B,C,X,B,C) :- interesting(A,B,C) & lt(A,X).
(9)         move(A,B,C,A,Y,C) :- interesting(A,B,C) & lt(B,Y).
(10)        move(A,B,C,A,B,Z) :- interesting(A,B,C) & lt(C,Z).
(11)        w(A,B,C) :- move(A,B,C,X,Y,Z) & !w(X,Y,Z).
(12)    end
```

Fig. 2. GLUE/NAIL program for win applied to game of nim.