```
module semi1;
    export main(:);
    edb arc(X,Y), path(X,Y);
    proc compPath(A : X,Y)
        rels newPath(X,Y);
        newPath(X,Y) := in(compPath(A)) & A(X,Y).
        path(X,Y) := {}.
        while newPath(X,Y) do
            path(X,Y) += newPath(X,Y).
            newPath(X,Y) := newPath(X,Z) &
                    in(compPath(A)) & A(Z,Y).
            newPath(X,Y) -= path(X,Y).
        end;
        return(A : X,Y) := in(compPath(A)) & path(X,Y).
    end
    prog main(:)
        rels temp(X,Y);
        arc(X,Y) := {}.
        arc(1,2).
        arc(2,3).
        arc(3,4).
        arc(4,5).
        arc(2,4).
        temp(X,Y) := compPath(arc,X,Y).
    end
end
```

Fig. 3. Seminaive path evaluation for arbitrary *arc* relation.