```
function ksum = subseq_kernel(str1, str2, q, lambda)
% ksum = subseq_kernel(str1, str2, q, lambda)
%
%   Computes subsequence kernel for dot product between `str1` and `str2`
%
%   Parameters:
%       str1   - a string to compute the kernel
%       str2   - a string to compute the kernel
%       q      - max subsequence length
%       lambda - decay parameter
%
%   Return:
%       ksum - result value for kernel computation between `str1` and `str2`
x1 = generate_feature_matrix(str1, q);
x2 = generate_feature_matrix(str2, q);

n = length(str1);
m = length(str2);
qp = q + 1;
kp = zeros(n, m, qp);
kp(:,:,1) = ones(n, m);
for z = 1:q
    for i = 1:length(x1{z})-1
        kpp = 0;
        for j = 1:length(x2{z})-1
            kpp = lambda * (kpp + (lambda * count_equalities(x1{z}{i}, x2{z}{j}) * kp(i,j,z)));
            kp(i+1,j+1,z+1) = lambda * kp(i,j+1,z+1) + kpp;
        end
    end
end

ksum = 0;
for z = 1:q
    for i = 1:length(x1{z})
        for j = 1:length(x2{z})
            ksum = ksum + (lambda * lambda * count_equalities(x1{z}{i}, x2{z}{j}) * kp(i,j,z));
        end
    end
end
end
```

Published with MATLAB® R2017b