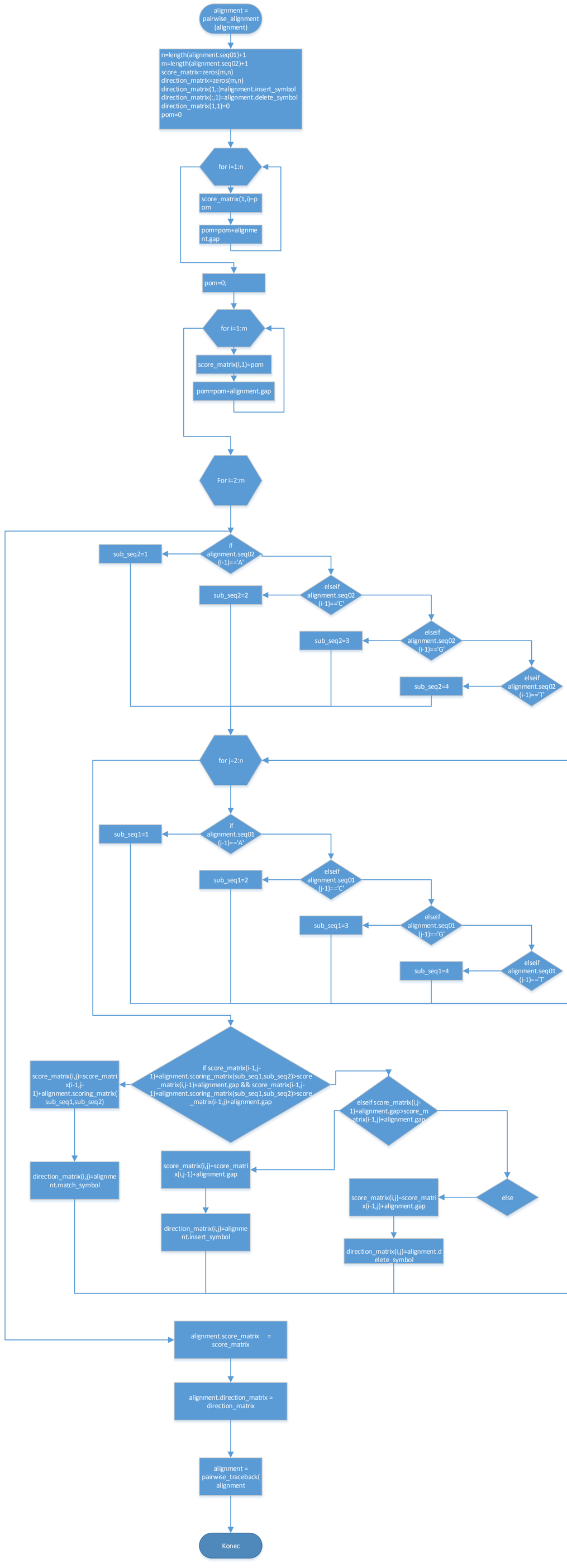
alignment = pairwise_alignment (alignment)
n=length(alignment.seq01)+1
m=length(alignment.seq02)+1
score_matrix=zeros(m,n)
direction_matrix=zeros(m,n)
direction_matrix(1,:)=alignment.insert_symbol
direction_matrix(:,1)=alignment.delete_symbol
direction_matrix(1,1)=0
pom=0
for i=1:n
score_matrix(1,i)=pom
pom=pom+alignment.gap
pom=0;
for i=1:m
score_matrix(i,1)=pom
pom=pom+alignment.gap
For i=2:m
if alignment.seq02 (i-1)=='A'
sub_seq2=1
elseif alignment.seq02 (i-1)=='C'
sub_seq2=2
elseif alignment.seq02 (i-1)=='G'
sub_seq2=3
elseif alignment.seq02 (i-1)=='T'
sub_seq2=4
for j=2:n
if alignment.seq01 (j-1)=='A'
sub_seq1=1
elseif alignment.seq01 (j-1)=='C'
sub_seq1=2
elseif alignment.seq01 (j-1)=='G'
sub_seq1=3
elseif alignment.seq01 (j-1)=='T'
sub_seq1=4
if score_matrix(i-1,j-1)+alignment.scoring_matrix(sub_seq1,sub_seq2)>score_matrix(i,j-1)+alignment.gap && score_matrix(i-1,j-1)+alignment.scoring_matrix(sub_seq1,sub_seq2)>score_matrix(i-1,j)+alignment.gap
score_matrix(i,j)=score_matrix(i-1,j-1)+alignment.scoring_matrix(sub_seq1,sub_seq2)
direction_matrix(i,j)=alignment.match_symbol
elseif score_matrix(i,j-1)+alignment.gap>score_matrix(i-1,j)+alignment.gap
score_matrix(i,j)=score_matrix(i,j-1)+alignment.gap
direction_matrix(i,j)=alignment.insert_symbol
else
score_matrix(i,j)=score_matrix(i-1,j)+alignment.gap
direction_matrix(i,j)=alignment.delete_symbol
alignment.score_matrix = score_matrix
alignment.direction_matrix = direction_matrix
alignment = pairwise_traceback(alignment
Konec