Lecture 11: Gene Prediction

Study Chapter 6.11-6.14

Problem Set #1: Study Session
2/19 from 5pm-7pm in SN011

Gene Prediction: Computational Challenge

- Gene: A sequence of nucleotides coding for protein

- Gene Prediction Problem: Determine the beginning and end positions of genes in a genome

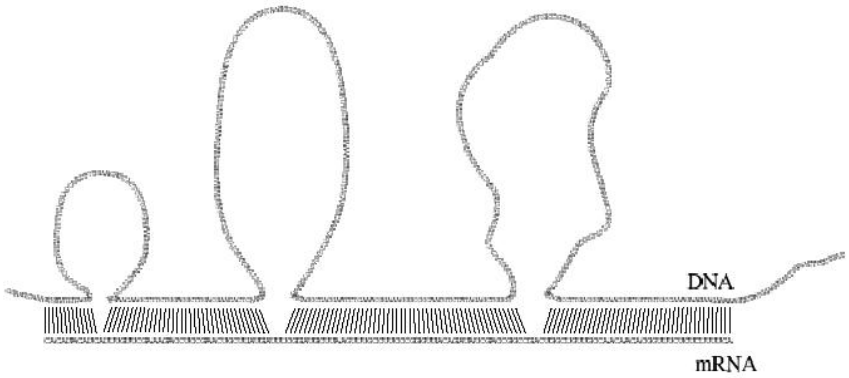
Gene Prediction: Computational Challenge

aatgcatgcggctatgctaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgcta
atgcatgcggctatgcaagctgggatccgatgactatgctaagctgggatccgatgacaatgcatgcggcta
tgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatgacaatgcatgcggctatgcta
atgaatggtcttgggatttaccttggaatatgctaatgcatgcggctatgctaagctgggatccgatgacaatg
catgcggctatgctaatgcatgcggctatgcaagctgggatccgatgactatgctaagctgcggctatgctaa
tgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctatgcaag
ctgggatcctgcggctatgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatgaca
atgcatgcggctatgctaatgaatggtcttgggatttaccttggaatatgctaatgcatgcggctatgctaagct
gggaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctat
gcaagctgggatccgatgactatgctaagctgcggctatgctaatgcatgcggctatgctaagctcatgcgg
ctatgctaagctgggaatgcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaat
gcatgcggctatgcaagctgggatccgatgactatgctaagctgcggctatgctaatgcatgcggctatgcta
agctcggctatgctaatgaatggtcttgggatttaccttggaatgctaagctgggatccgatgacaatgcatg
cggctatgctaatgaatggtcttgggatttaccttggaatatgctaatgcatgcggctatgctaagctgggaat
gcatgcggctatgctaagctgggatccgatgacaatgcatgcggctatgctaatgcatgcggctatgcaagc
tgggatccgatgactatgctaagctgcggctatgctaatgcatgcggctatgctaagctcatgcgg

Gene!

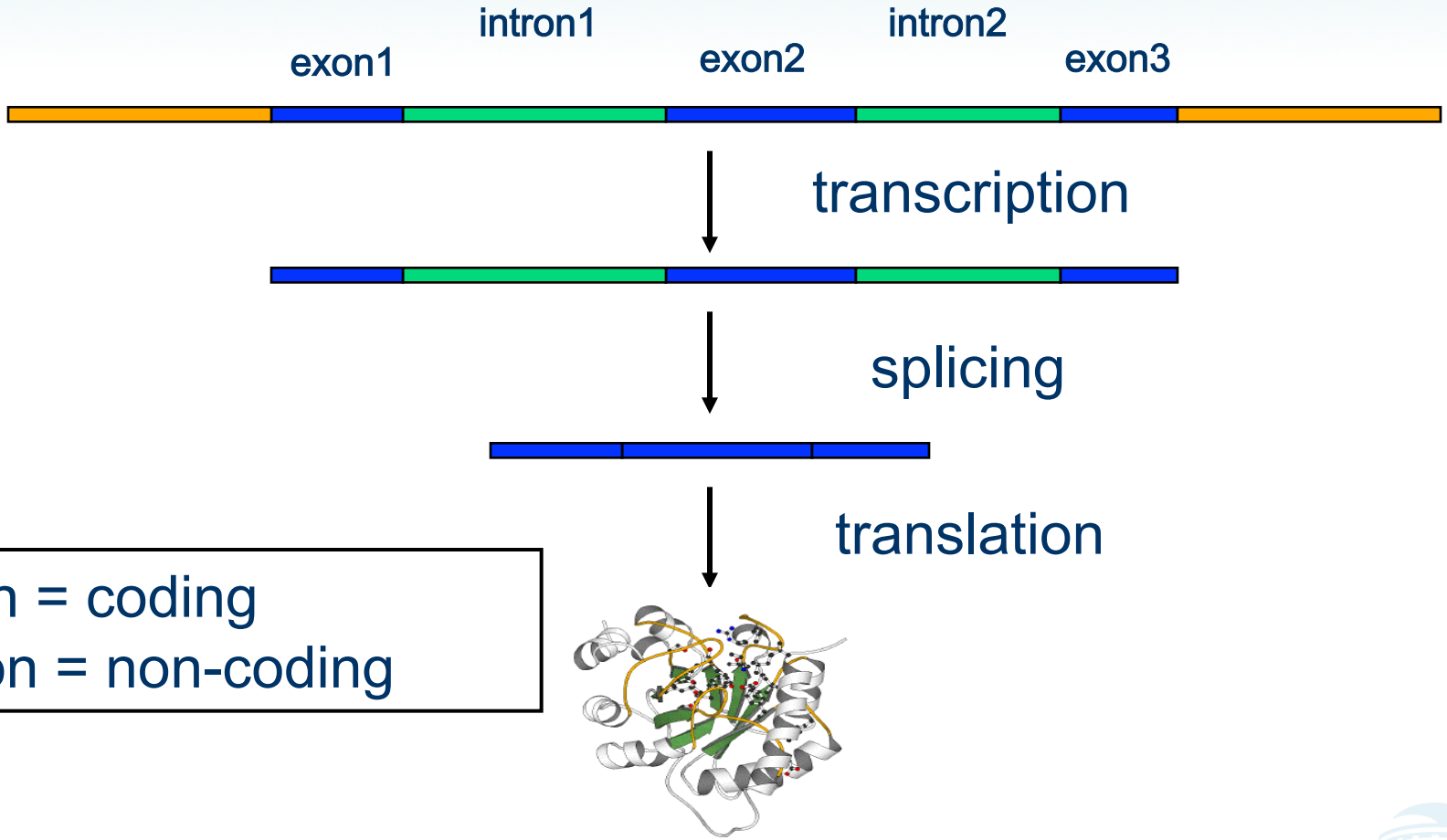
Exons and Introns

- In eukaryotes, the gene is a combination of coding segments (**exons**) that are interrupted by non-coding segments (**introns**)
- This makes computational gene prediction in eukaryotes even more difficult



- Prokaryotes, one-cell animals without a cell nucleus (i.e. bacteria, archaea) don't have introns - Genes in prokaryotes are continuous
- What are the hints that a gene is nearby?

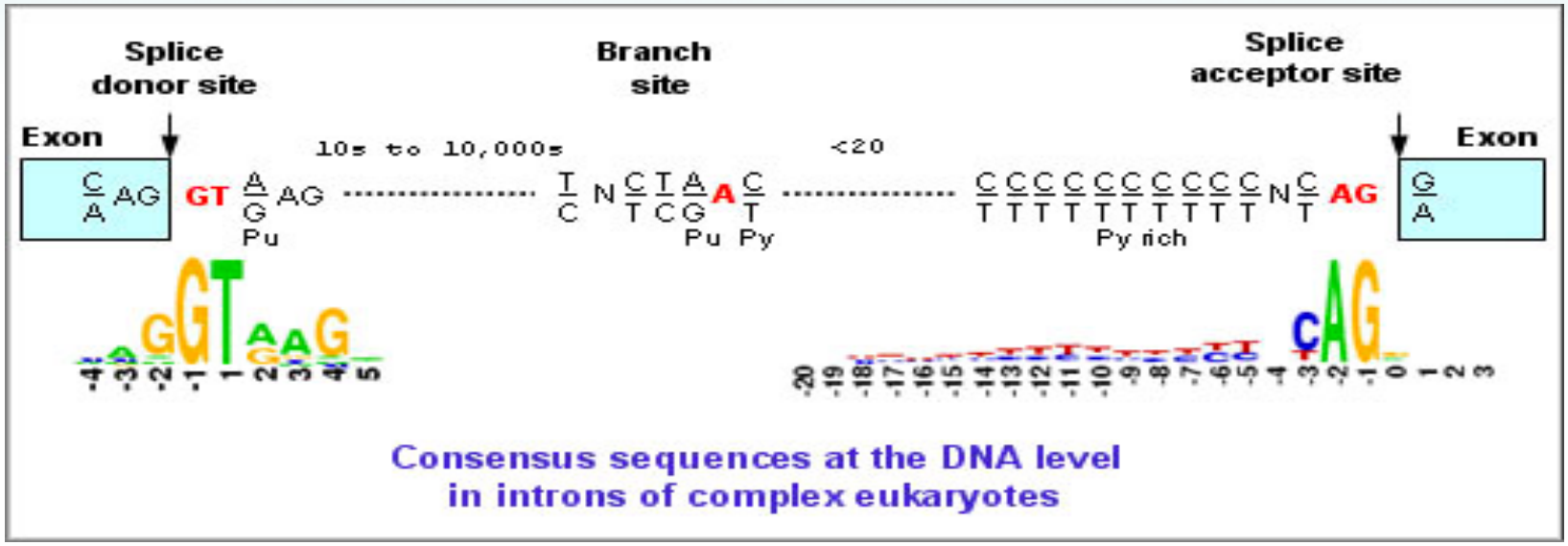
Central Dogma and Splicing



exon = coding
intron = non-coding

Batzoglou

Hint 1: Splicing Signals



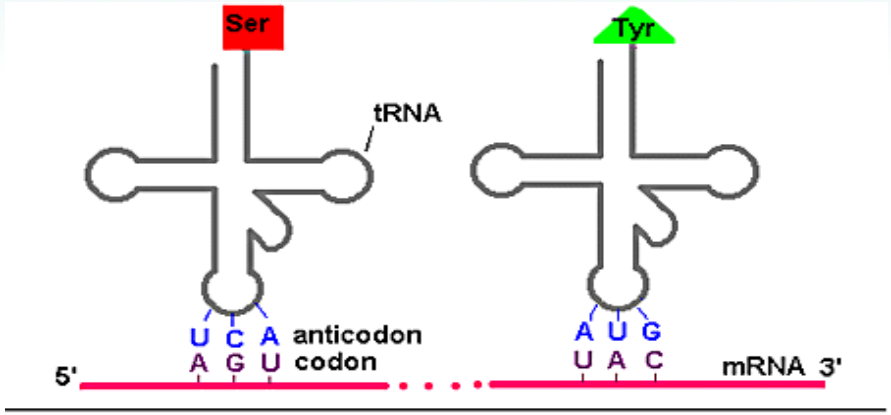
- The genome provides subtle hints of where exon/intron boundaries might occur
- The dinucelotides GT and AG on the left- and right-hand sides of an intron are highly conserved. (immediately adjacent to the exons)

Two Approaches to Gene Prediction

- **Statistical**: coding segments (exons) have typical sequences on either end and use different subwords than non-coding segments (introns).

- **Similarity-based**: many human genes are similar to genes in mice, chicken, or even bacteria. Therefore, already known mouse, chicken, and bacterial genes may help to find human genes.

Hint 2: Genetic Code and Stop Codons



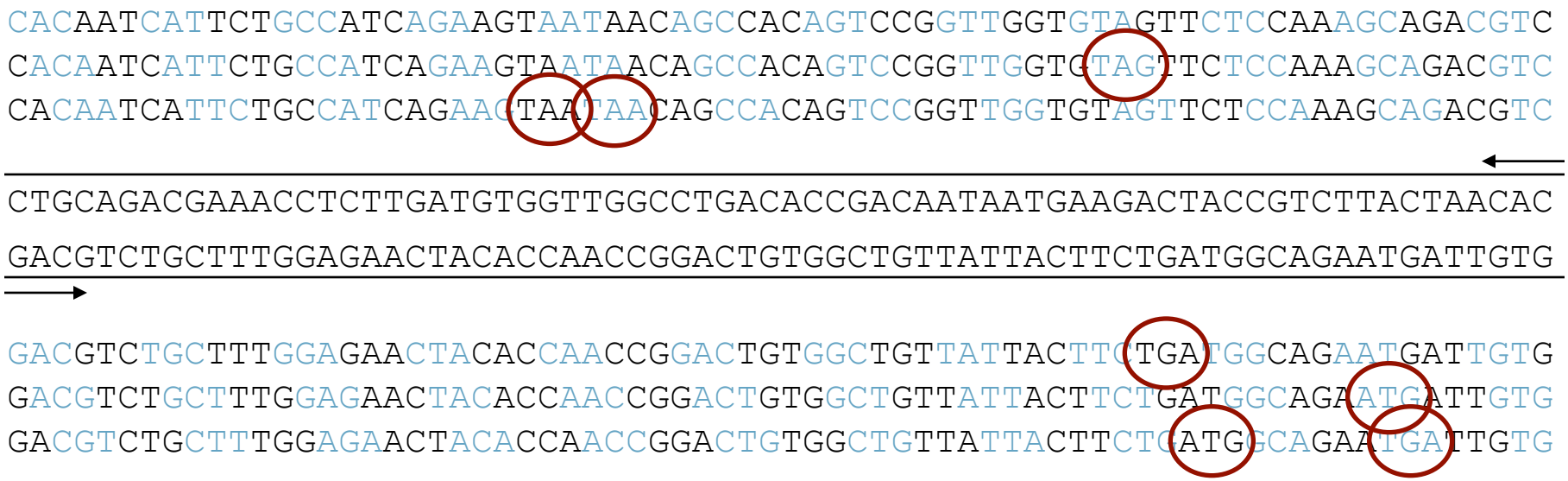
2nd base in codon

1st base in codon	U	C	A	G	3rd base in codon
U	Phe Phe Leu Leu	Ser Ser Ser Ser	Tyr Tyr STOP STOP	Cys Cys STOP Trp	U C A G
C	Leu Leu Leu Leu	Pro Pro Pro Pro	His His Gln Gln	Arg Arg Arg Arg	U C A G
A	Ile Ile Ile Met	Thr Thr Thr Thr	Asn Asn Lys Lys	Ser Ser Arg Arg	U C A G
G	Val Val Val Val	Ala Ala Ala Ala	Asp Asp Glu Glu	Gly Gly Gly Gly	U C A G

The Genetic Code

TAA, TAG and TGA correspond to 3 Stop codons that (together with Start codon ATG) delineate Open Reading Frames

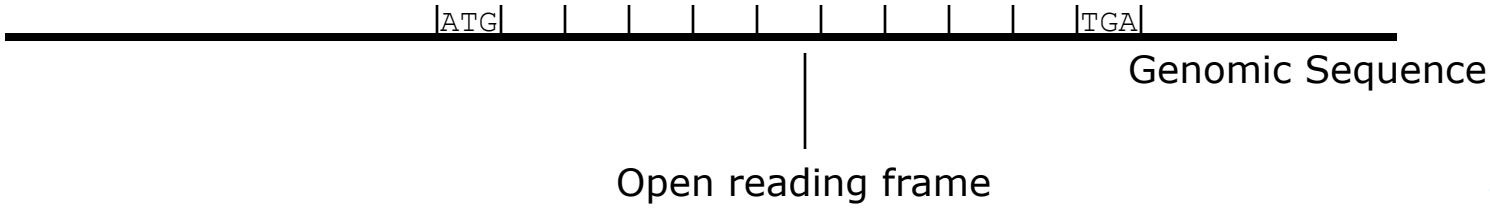
Six Frames in a DNA Sequence



- start codons - ATG
- stop codons – TAA, TAG, TGA

Open Reading Frames (ORFs)

- Detect potential coding regions by looking at **ORFs**
  - A genome of length *n* is comprised of (*n*/3) codons
  - Stop codons break genome into segments between consecutive Stop codons
  - The subsegments of these that start from the Start codon (ATG) are ORFs
    - ORFs in different frames may overlap



Long versus Short ORFs

- Long open reading frames may be a gene
  - At random, we should expect one stop codon every (64/3) ~= 21 codons
  - **However**, genes are usually much longer than this
- A basic approach is to scan for ORFs whose length exceeds certain threshold
  - This is naïve because some genes (e.g. some neural and immune system genes) are relatively short

Testing ORFs: Codon Usage

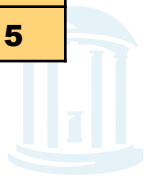
- Create a 64-element hash table and count the frequencies of codons in an ORF
- Amino acids typically have more than one codon, but in nature certain codons are used more often
- Uneven use of the codons may characterize a real gene
- This compensates for pitfalls of the ORF length test

Codon Usage in the Human Genome

	T			C			A			G		
T	TTT	Phe	46	TCT	Ser	19	TAT	Tyr	44	TGT	Cys	46
	TTC	Phe	54	TCC	Ser	22	TAC	Tyr	56	TGC	Cys	54
	TTA	Leu	8	TCA	Ser	15	TAA	Stop	30	TGA	Stop	47
	TTG	Leu	13	TCG	Ser	5	TAG	Stop	24	TGG	Trp	100
C	CTT	Leu	13	CCT	Pro	29	CAT	His	42	CGT	Arg	8
	CTC	Leu	20	CCC	Pro	32	CAC	His	58	CGC	Arg	18
	CTA	Leu	7	CCA	Pro	28	CAA	Gln	27	CGA	Arg	11
	CTG	Leu	40	CCG	Pro	11	CAG	Gln	73	CGG	Arg	20
A	ATT	Ile	36	ACT	Thr	25	AAT	Asn	47	AGT	Ser	15
	ATC	Ile	47	ACC	Thr	36	AAC	Asn	53	AGC	Ser	24
	ATA	Ile	17	ACA	Thr	28	AAA	Lys	43	AGA	Arg	21
	ATG	Met	100	ACG	Thr	11	AAG	Lys	57	AGG	Arg	21
G	GTT	Val	18	GCT	Ala	27	GAT	Asp	46	GGT	Gly	16
	GTC	Val	24	GCC	Ala	40	GAC	Asp	54	GGC	Gly	34
	GTA	Val	12	GCA	Ala	23	GAA	Glu	42	GGA	Gly	25
	GTG	Val	46	GCG	Ala	11	GAG	Glu	58	GGG	Gly	25

Codon Usage in the Mouse Genome

	T			C			A			G		
T	TTT	Phe	39	TCT	Ser	16	TAT	Tyr	41	TGT	Cys	45
	TTC	Phe	61	TCC	Ser	21	TAC	Tyr	59	TGC	Cys	55
	TTA	Leu	7	TCA	Ser	21	TAA	Stop	27	TGA	Stop	27
	TTG	Leu	10	TCG	Ser	6	TAG	Stop	45	TGG	Trp	100
C	CTT	Leu	13	CCT	Pro	26	CAT	His	35	CGT	Arg	6
	CTC	Leu	21	CCC	Pro	30	CAC	His	65	CGC	Arg	13
	CTA	Leu	14	CCA	Pro	34	CAA	Gln	24	CGA	Arg	10
	CTG	Leu	35	CCG	Pro	10	CAG	Gln	76	CGG	Arg	13
A	ATT	Ile	27	ACT	Thr	24	AAT	Asn	41	AGT	Ser	13
	ATC	Ile	43	ACC	Thr	28	AAC	Asn	59	AGC	Ser	23
	ATA	Ile	29	ACA	Thr	40	AAA	Lys	58	AGA	Arg	34
	ATG	Met	100	ACG	Thr	8	AAG	Lys	42	AGG	Arg	25
G	GTT	Val	14	GCT	Ala	23	GAT	Asp	39	GGT	Gly	17
	GTC	Val	24	GCC	Ala	36	GAC	Asp	61	GGC	Gly	32
	GTA	Val	19	GCA	Ala	30	GAA	Glu	49	GGA	Gly	29
	GTG	Val	43	GCG	Ala	11	GAG	Glu	51	GGG	Gly	15



Codon Usage and Likelihood Ratio

- An ORF is more "believable" than another if it has more "likely" codons
- Do sliding window calculations to find ORFs that have the "likely" codon usage
- Allows for higher precision in identifying true ORFs; much better than merely testing for length.
- However, average vertebrate exon length is 130 nucleotides, which is often too small to produce reliable peaks in the likelihood ratio
- Further improvement: **in-frame hexamer count** (frequencies of pairs of consecutive codons)

Similarity-Based Approach

- Genes in different organisms are similar
- The similarity-based approach uses known genes in one genome to predict (unknown) genes in another genome
- **Problem:** Given a known gene and an unannotated genome sequence, find a set of substrings of the genomic sequence whose concatenation best fits the gene
  - An alignment problem!

Chaining Local Alignments

- Locate codon substrings that match a given protein subsequence, putative (candidate) exons
- Define a putative exons as 3-tuples (*l, r, w*)

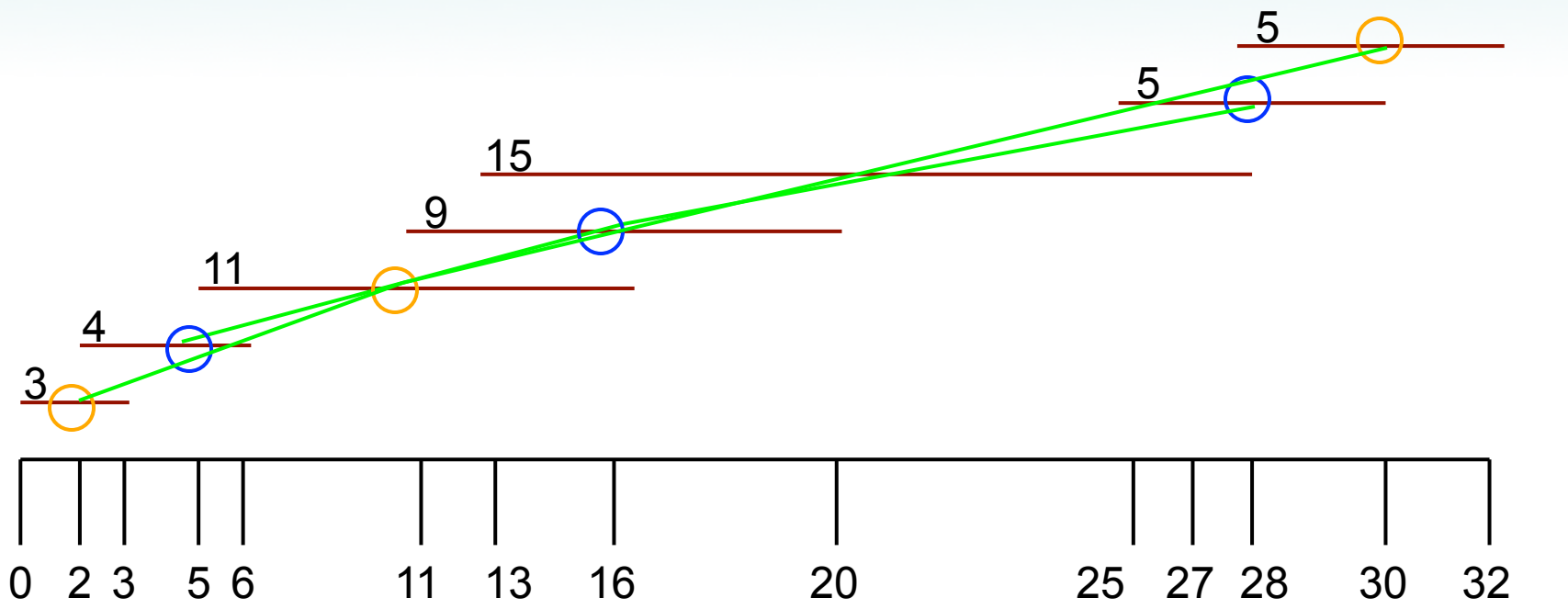
  l = *left starting position*

  r = *right ending position*

  w = *weight* based on some scoring function

  w(# of amino acid's matched, codon freq, …)
- Look for a maximum chain of substrings
  - Chain: a set of non-overlapping nonadjacent intervals.

Exon Chaining Problem



- Locate the beginning and end of each interval (*2n* points)
- Find the "best" path

Exon Chaining Problem: Formulation

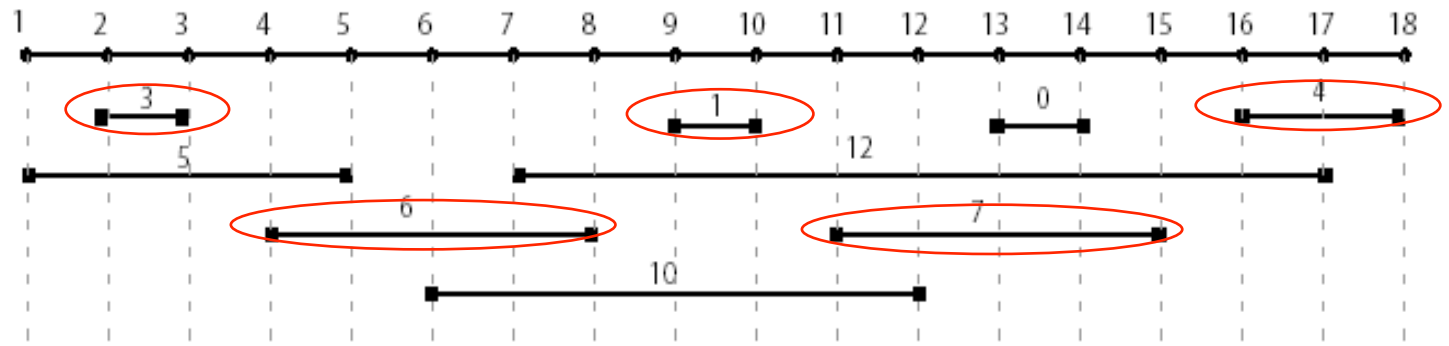
- **Exon Chaining Problem:** Given a set of putative exons, find a maximum set of non-overlapping putative exons

- **Input**: a set of weighted intervals (putative exons)

- **Output**: A maximum chain of intervals from this set

Would a greedy algorithm solve this problem?

Exon Chaining Problem: Graph Representation



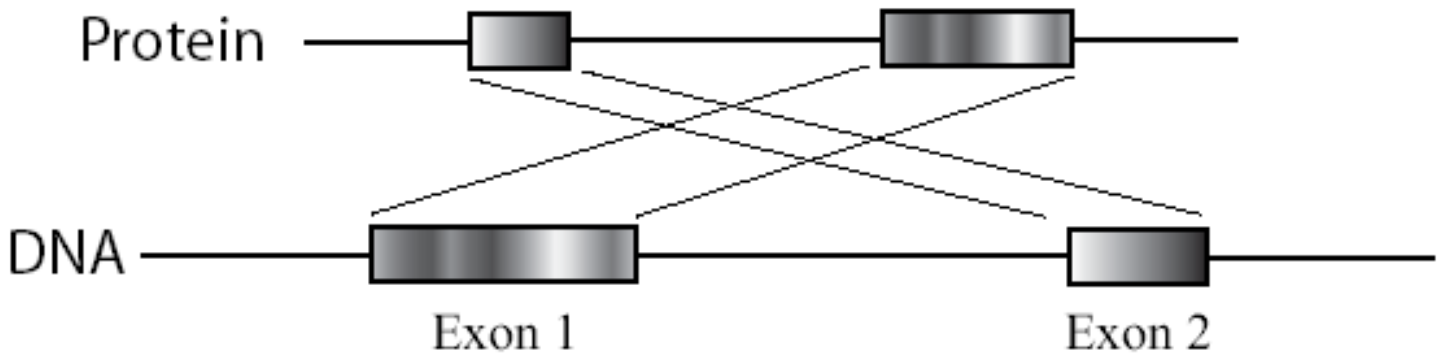
- A greedy solution takes the highest scoring chain first, followed by largest one left in the uncovered range, and so on until none can be taken, the score is the sum of all taken chains
- This problem can be solved with dynamic programming in O(*n*) time, by constructing a graph

Exon Chaining Algorithm

ExonChaining (G, n) //Graph, number of intervals

```
1  for i ← to 2n
2    s_i ← 0
3  for i ← 1 to 2n
4    if vertex v_i in G corresponds to right end of the interval I
5      j ← index of vertex for left end of the interval I
6      w ← weight of the interval I
7      s_j ← max {s_j + w, s_{i-1}}
8    else
9      s_i ← s_{i-1}
10 return s_{2n}
```

Exon Chaining: Deficiencies



- Poor definition of the putative exon endpoints
- Optimal interval chain may not correspond to a valid alignment
  - We enforce genomic order but not protein order
  - First interval may correspond to a suffix, whereas second interval may correspond to a prefix
  - Combination of such intervals is not a valid alignment

Spliced Alignment

- Mikhail Gelfand and colleagues proposed a **spliced alignment** approach of using a protein *within one genome to reconstruct* the exon-intron structure of *a (related) gene in another genome*.
  - Begins by selecting either all putative exons between potential acceptor and donor sites or by finding all substrings similar to the target protein (as in the Exon Chaining Problem).
  - This set is further filtered in a such a way that attempt to retain all true exons, but allow some false ones. (Many false positives, but no false negatives)

Spliced Alignment Problem: Formulation

- **Goal**: Find a chain of blocks in a genomic sequence that best fits a target sequence
- **Input**: Genomic sequences *G*,
  target sequence *T*,
  and a set of candidate exons *B*.
- **Output**: A chain of exons Γ such that the global alignment score between Γ* and T is maximum among all chains of blocks from *B*.

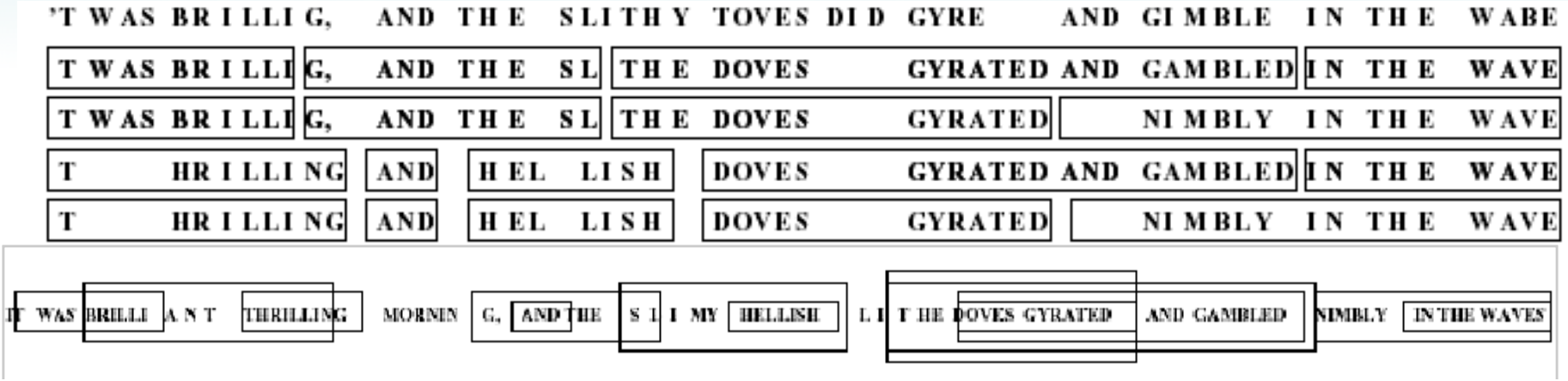
Γ* - concatenation of all exons from chain Γ

Lewis Carroll Example

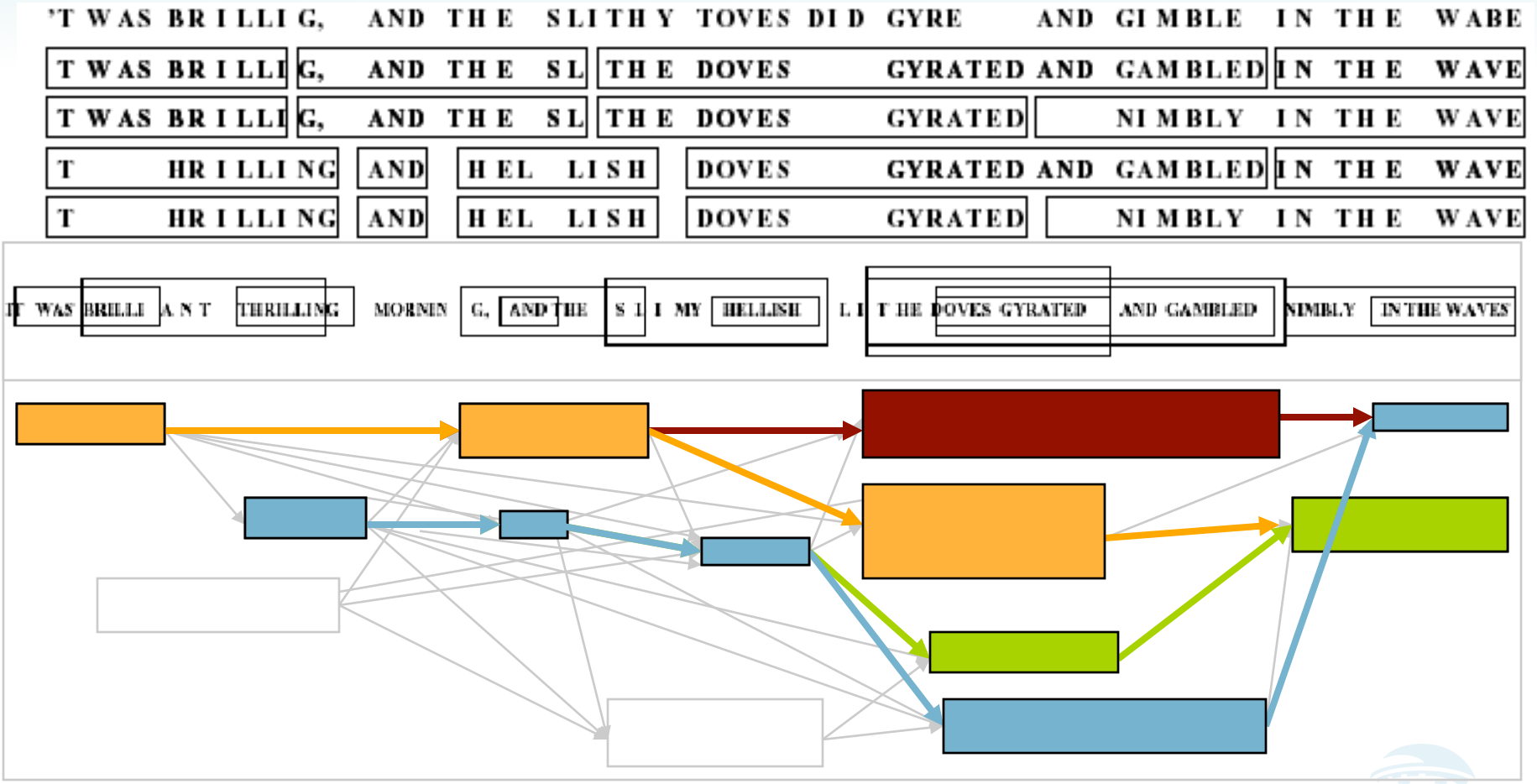
'T WAS BR I LLI G, AND TH E SLI TH Y TOVES DI D GYRE AND GI MBLE I N TH E WABE

IT WAS BRILLIANT THRILLING MORNING AND THE SLIMY HELLISH LITHE DOVES GYRATED AND GAMBLED NIMBLY IN THE WAVES

Lewis Carroll Example



Lewis Carroll Example



Exon Chaining vs Spliced Alignment

- In Spliced Alignment, every path spells out a string obtained by concatenating the labels of its vertices.
- The overall path gives an optimal alignment score between concatenated labels (blocks) and target sequence
- Defines weight as the sum of vertex weights rather than as the sum of edge weights as in Spliced Alignment
- Exon Chaining assumes the positions and weights of exons are pre-defined

How to solve using Dynamic Programming?

Spliced Alignment: Idea

- Compute the best alignment between *i*-prefix of genomic sequence *G* and *j*-prefix of target *T:*

  S(i,j)

- But what is *"i*-prefix*"* of *G?*
- There may be a few *i*-prefixes of *G* depending on which block *B* we are in.
- Compute the best alignment between *i*-prefix of genomic sequence *G* and *j*-prefix of target *T* ***under the assumption*** that the alignment uses the block *B* at position *i,*

  S(i,j,B)

- Two cases to consider*, block B* starts at i, or it does not

Spliced Alignment Recurrence

If *i* is not the starting vertex of block *B*:

$$S(i,j,B) = \max \begin{cases} S(i-1,j,B) - \sigma \\ S(i,j-1,B) - \sigma \\ S(i-1,j-1,B) + \delta(g_i, t_j) \end{cases}$$

Recall σ was the cost of an indel

If *i* is the starting vertex of block *B*:

$$S(i,j,B) = \max \begin{cases} S(i-1,j,B) - \sigma \\ \max_{\text{all blocks B' preceding B}} S(end(B'), j-1, B') - \delta(g_i, t_j) \\ \max_{\text{all blocks B' preceding B}} S(end(B'), j, B') - \sigma \end{cases}$$

Spliced Alignment Solution

- After computing the three-dimensional table *S(i, j, B)*, the score of the optimal spliced alignment is:

$$\max_{\text{all blocks } B} S(end(B), length(T), B)$$

Spliced Alignment: Complications

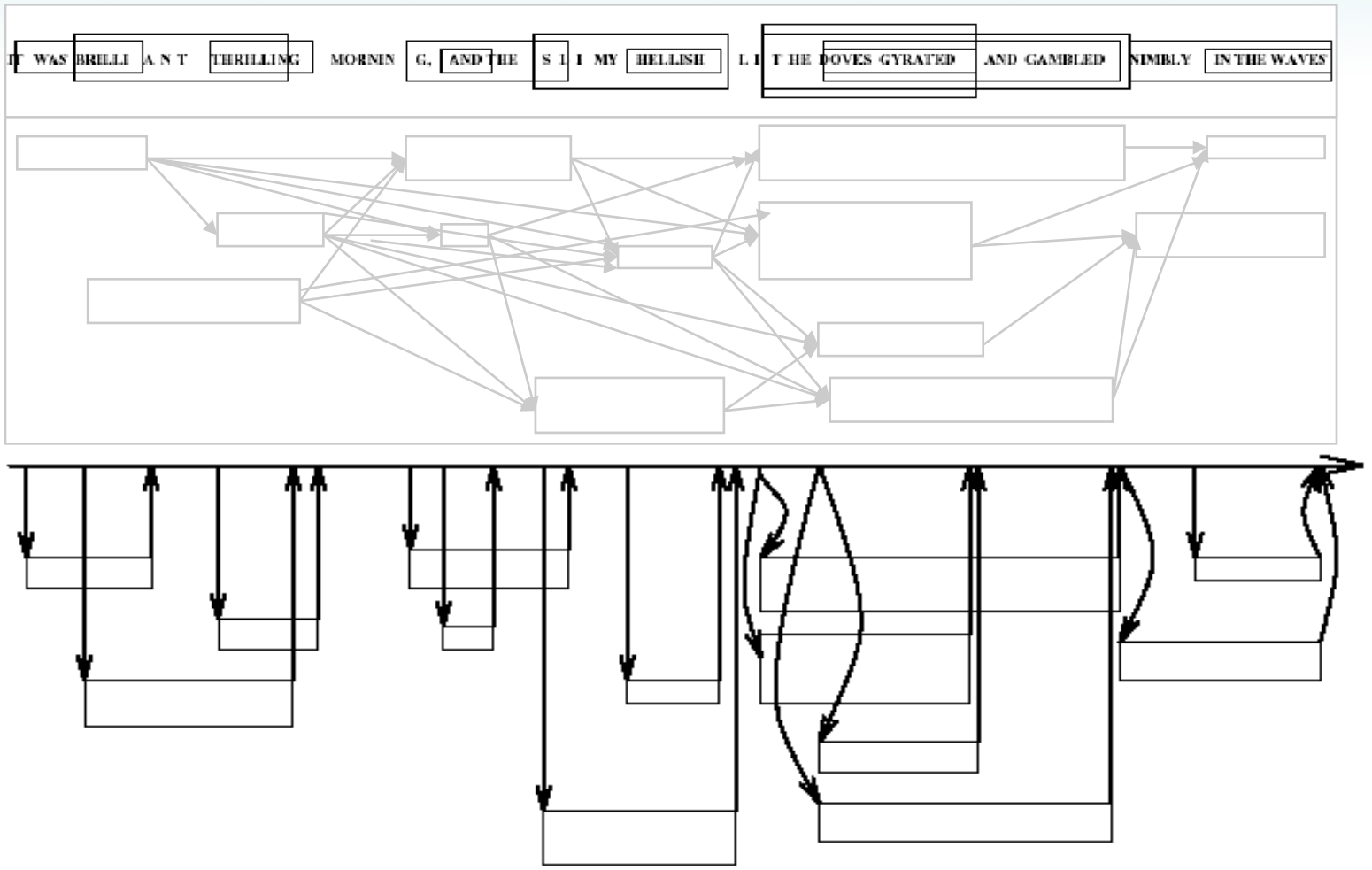
- Considering multiple *i*-prefixes leads to slow down. running time:

$$O(mn^2 |B|)$$

where m is the target length, n is the genomic sequence length and $|B|$ is the number of blocks.

- A ***mosaic effect***: short exons are easily combined to fit any target protein

Spliced Alignment: Speedup



Spliced Alignment: Speedup

$P(i,j) = \max_{\text{all blocks } B \text{ preceding position } i} S(end(B), j, B)$