Algorithms for a Peptide Computer

M. Sakthi Balan

Theoretical Computer Science Lab
Department of Computer Science and Engg.
Indian Institute of Technology
Chennai – 600036.

***Email**: sakthi@cs.iitm.ernet.in*

***URL**: www.cs.iitm.ernet.in/theory/sakthi*

Organization

- Natural Computing
- Biological Computing
- DNA Computing
- Peptide Computing
- Solving Hamiltonian Path Problem
- Solving Exact 3-Cover Set Problem
- Solving Satisfiability Problem
- Conclusion

"….It seems that progress in electronic hardware (and the corresponding software engineering) is not enough; for instance, the miniaturization is approaching the quantum boundary, where physical processes obey laws based on probabilities and non-determinism, something almost completely absent in the operation of "classical" computers. So, new breakthrough is needed…."

Computing with Cells and Atoms –

Cristian S. Calude and Gh. Paun

Natural Computing

- Biological Computing
- Quantum Computing

Biological Computing

- DNA Computing
- Peptide Computing

DNA Computing

- Uses DNA strands and Watson-Crick Complementarity as operation
- Highly ***non-deterministic***
- Massive ***parallelism***
- Solves NP-Complete Problems quite efficiently

Peptide Computing

- Uses peptides and antibodies
- Operation – binding of antibodies to epitopes in peptides
- ***Epitope*** – The site in peptide recognized by antibody
- Highly ***non-deterministic***
- Massive ***parallelism***

Peptide Computing Contd..

- Peptides – sequence of amino acids
- Twenty amino acids. Example – Glycine, Valine
- Connected by covalent bonds

Peptide Computing Contd..

- Antibodies recognizes epitopes by binding to it
- Binding of antibodies to epitopes has associated power called ***affinity***
- Higher priority to the antibody with larger affinity power

Computing
DNA Vs Peptide

- Four building blocks
- Adenine (A), Guanine(G), Cytosine (C), Thiamine (T)
- Only one reverse complement – Watson-Crick Complement
- Complement (A) = T and Complement (G) = C

- Twenty building blocks (20 amino acids)
- Example: Glycine, Valine
- Different antibodies can recognize different epitopes
- Binding affinity of antibodies can be different

Peptide Computing Model

- Peptides represent sample space of the problem
- Antibodies are used to select the correct solution of the problem (i.e. peptides)

Definition

- For finite sequence $M = m_1, m_2, \ldots, m_n$ the ***doubly duplicated sequence*** is

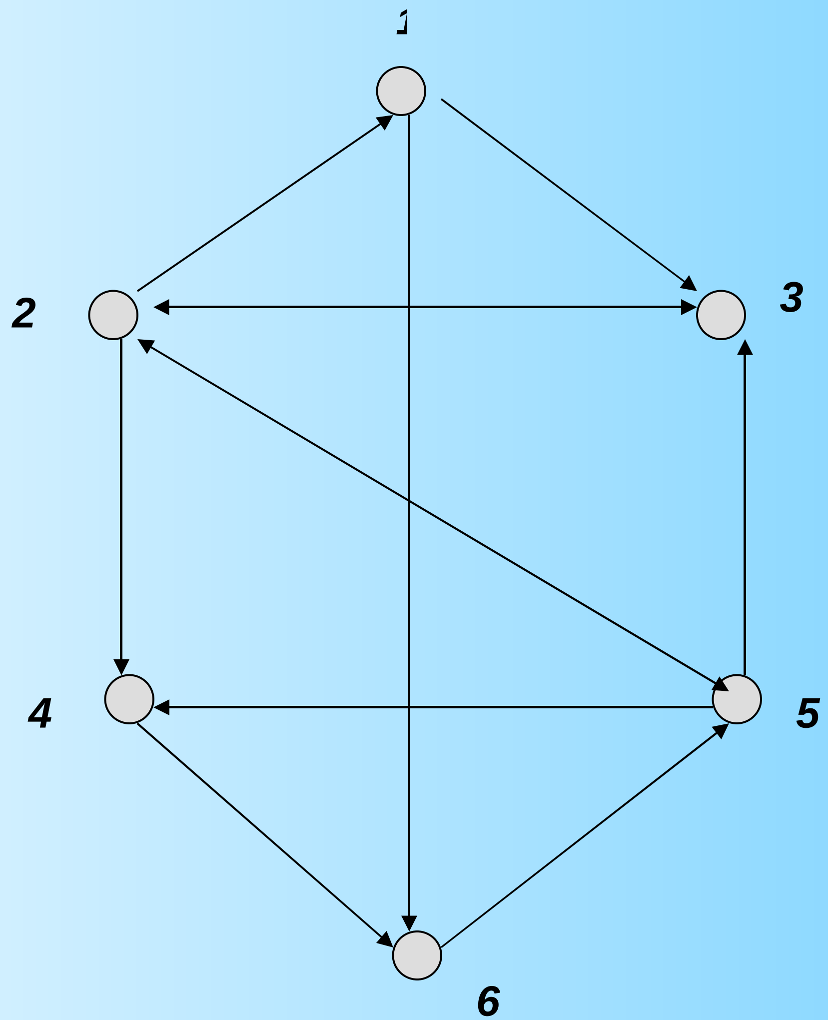
$$MM = m_1, m_1, m_2, m_2, \ldots, m_n, m_n$$

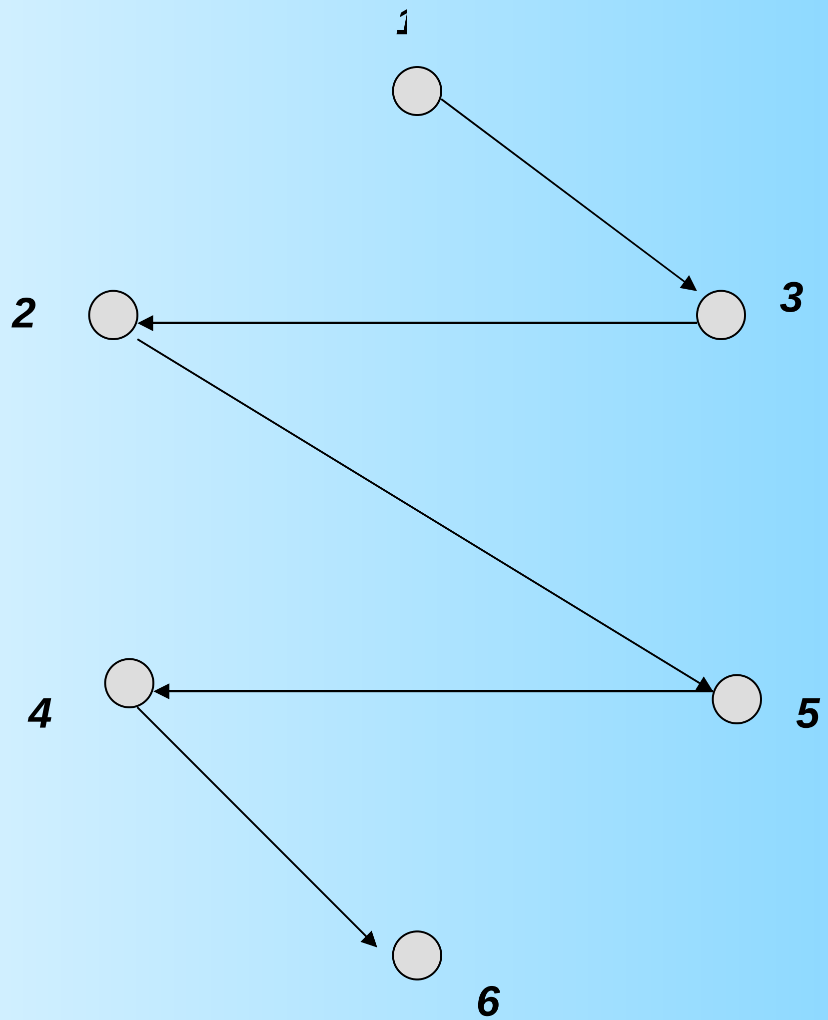
- Doubly duplicated permutation of a finite set S is

$\{mm \mid m$ is a permutation of the set $S\}$

Hamiltonian Path Problem

- $G = (V,E)$ is a directed graph
- $V = \{v_1, v_2, \ldots, v_n\}$ is the vertex set
- $E = \{e_{ij} \mid v_i$ is adjacent to $v_j\}$ is the edge set
- v_1 - source vertex, v_n – end vertex
- **Problem** – Test whether there exists a Hamiltonian path between v_1 and v_n


1
2
3
4
5
6


1
2
3
4
5
6

Peptides Formation

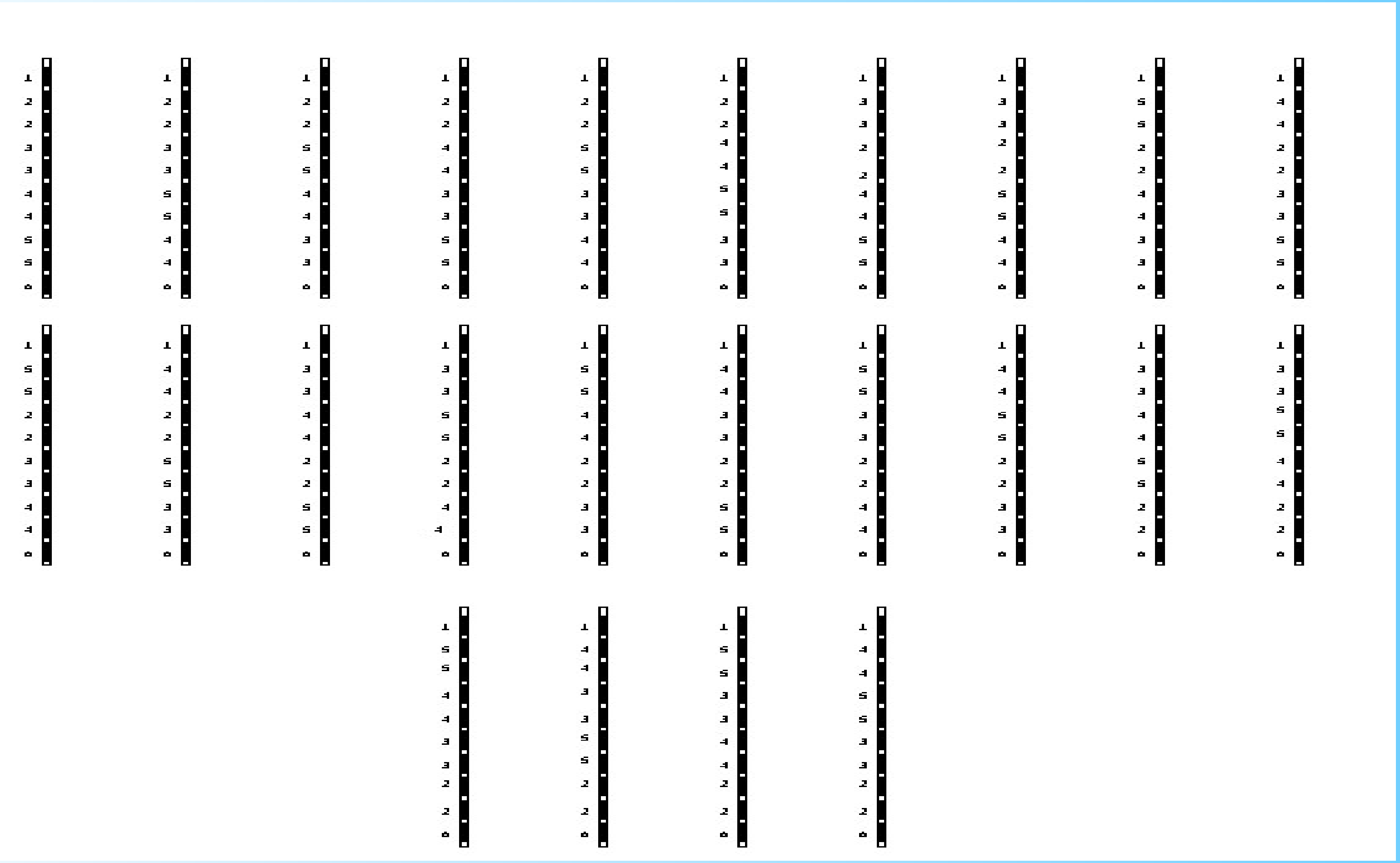
- Each vertex v_i has a corresponding epitope ep_i
- Each peptide has ep_1 on one extreme and ep_n on the other extreme
- All doubly duplicated permutations of

  $\{ep_2, \ldots, ep_{n-1}\}$ are formed in each of the peptide in between ep_1 and ep_n

Antibody Formation

- Form antibodies A_{ij} – site = $ep_i ep_j$ s.t. v_j is adj. to v_i
- Form antibodies B_{ij} – site = $ep_i ep_j$ s.t. v_j is not adj. to v_i
- Form antibody C – site is whole of peptide
- Affinity(B_{ij}) > Affinity(C)
- Affinity(C) > Affinity(A_{ij})

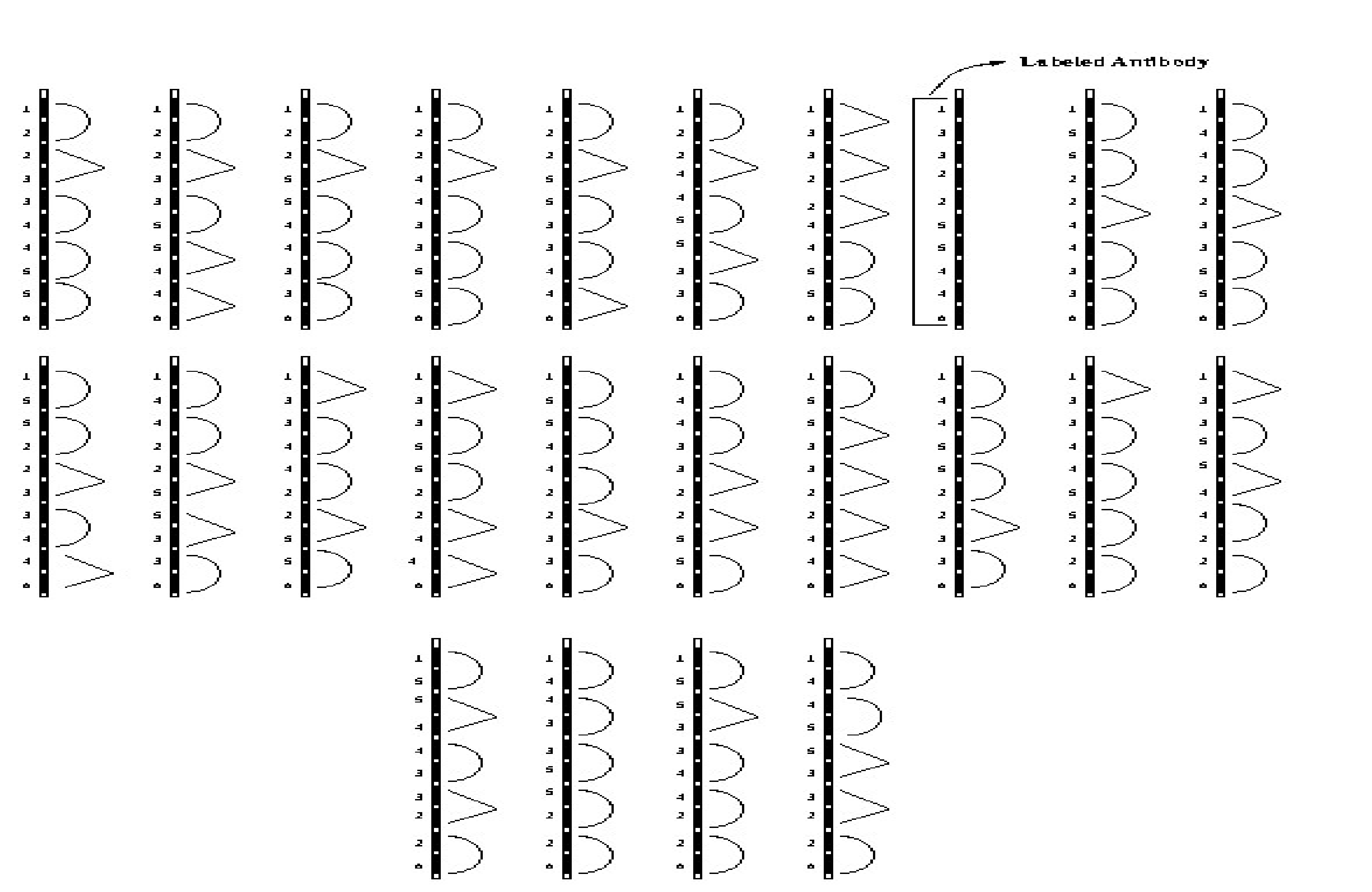
Peptide Solution Space



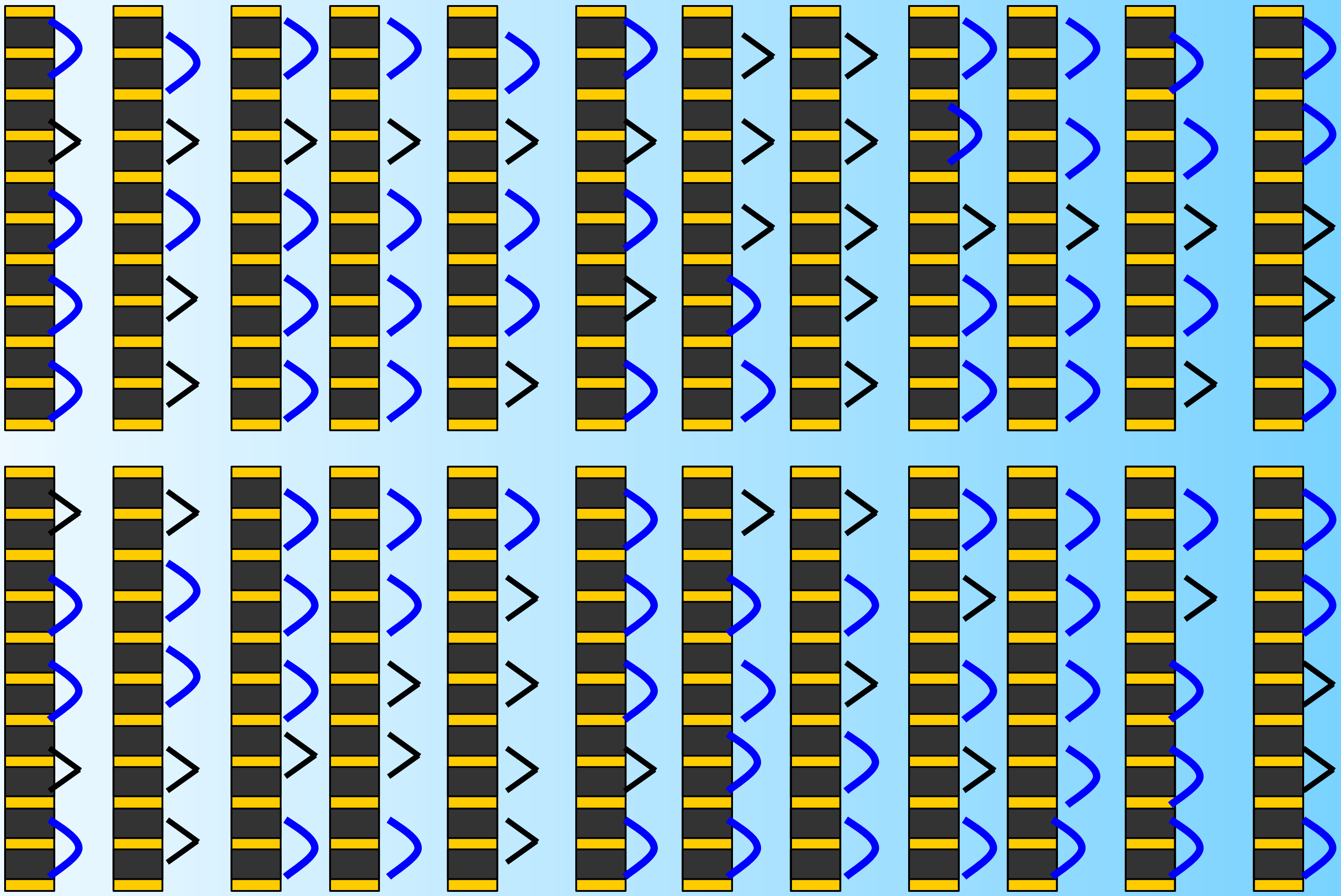
Algorithm

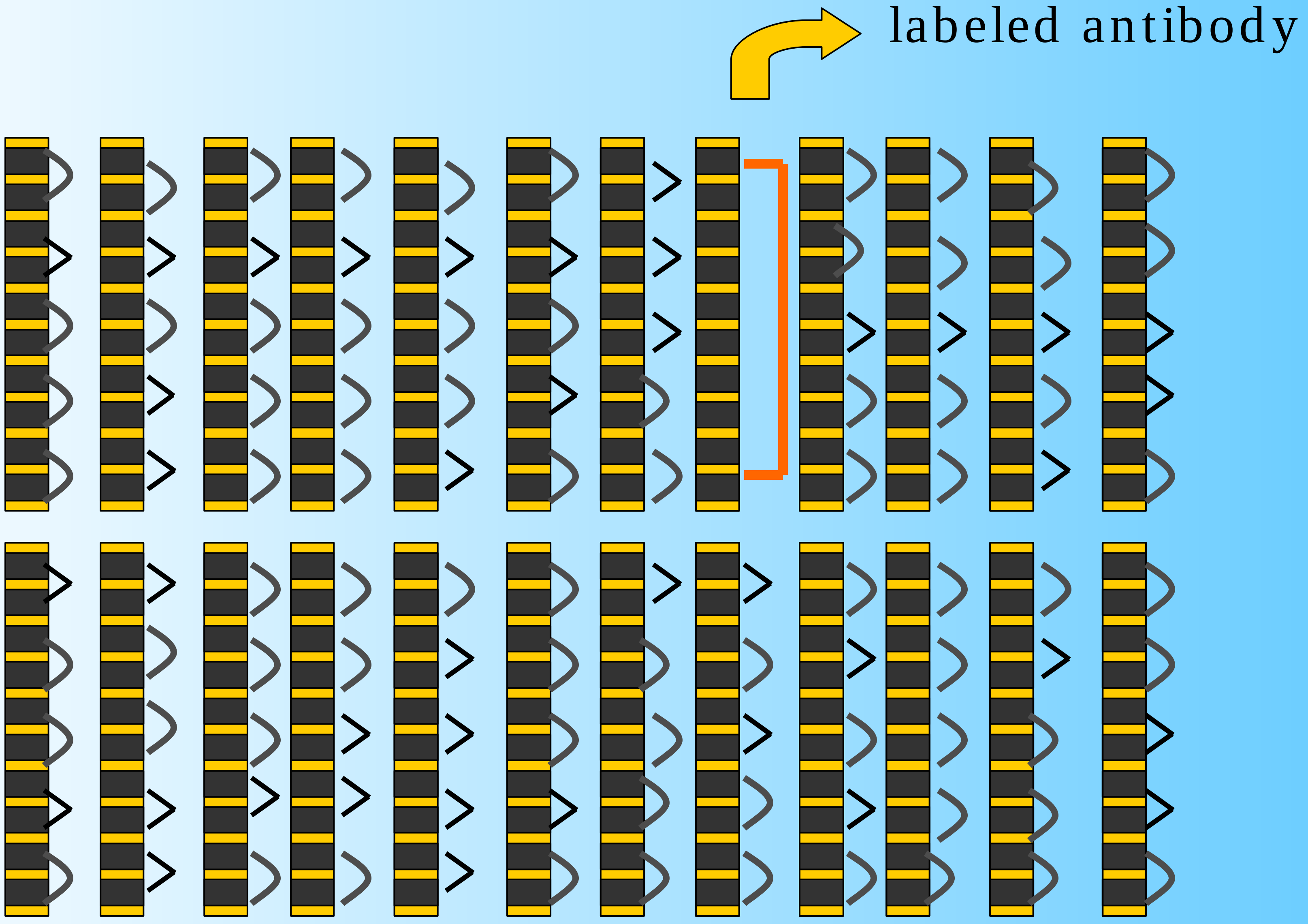
1. Take all the peptides in an aqueous solution
2. Add antibodies A_{ij}
3. Add antibodies B_{ij}
4. Add labeled antibody C
5. If fluorescence is detected answer is *yes* or else the answer is *no*

Peptides with Antibodies



Peptide with Antibodies




labeled antibody

Complexity

- Number of peptides = $(n-2)!$
- Length of peptides = $O(n)$
- Number of antibodies = $O(n^2)$
- Number of Bio-steps is *constant*

Exact Cover by 3-Sets Problem

- **Instance:** A finite set $X = \{x_1, x_2, \ldots, x_n\}$, $n = 3q$ and a collection C of 3-elements subsets of X
- **Question**: Does C contain an ***Exact Cover*** for X

Peptide Formation

- For each x_i a specific epitope ep_i is chosen
- For every permutation of the set $\{ep_i\}$ a peptide is chosen *s.t.* every subsequence of $ep_i\, ep_j\, ep_k$ is followed by the epitope ep_{ijk}

Example

$$X = \{x_1, x_2, \ldots, x_9\}$$

For permutation

$$x_1, x_7, x_9, x_2, x_6, x_4, x_3, x_5, x_8$$

Antibody Formation

- Form antibodies A_{ijk}, site = $ep_i\, ep_j\, ep_k$ if $\{x_i,x_j,x_k\}$ is in C
- Form antibodies B_{ijk}, site = $ep_i\, ep_j\, ep_k$ if $\{x_i,x_j,x_k\}$ is not in C
- Form colored antibody C, site is whole of peptide
- Affinity(B_{ijk}) > Affinity(C)
- Affinity(C) > Affinity(A_{ijk})

Algorithm

- Take all the antibodies in an aqueous solution.
- Add antibodies A_{ijk}
- Add antibodies B_{ijk}
- Add antibody C
- If fluorescence is detected the answer is *yes* otherwise *no*

Complexity

- Number of peptides = $n!$
- Length of peptides = $O(n)$
- Number of Antibodies = $O(n^3)$
- Number of Bio-steps is *constant*

Satisfiability Problem

Problem: Let F be a formula over n variables. Does there exists an assignment of truth value to every variable in F such that F becomes *true*.

Satisfiability Problem (Contd..)

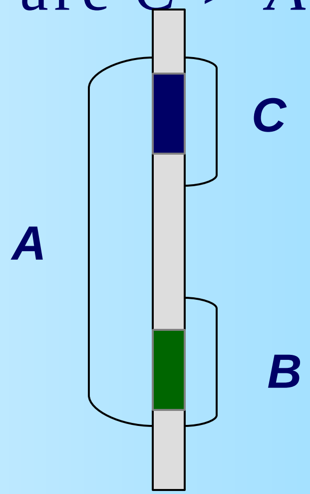
- Let F be a formula in conjunctive normal form.
- There are n variables in F.
- To find an assignment such that F is *true*.
- $N = 2^n$ assignments possible.

Example

- Let F = $(v_1 \text{ or } \neg v_2)$ and $\neg v_2$ and $(v_1 \text{ or } v_2)$
- Assignments are *(F,F)*, *(F,T)*, *(T,F)*, and *(T,T)*
- *(T,F)* satisfies *F*

Peptide Formation

- For each assignment prepare a peptide and different antibodies binding to overlapping epitopes.
- Binding affinities are $C > A > B$.

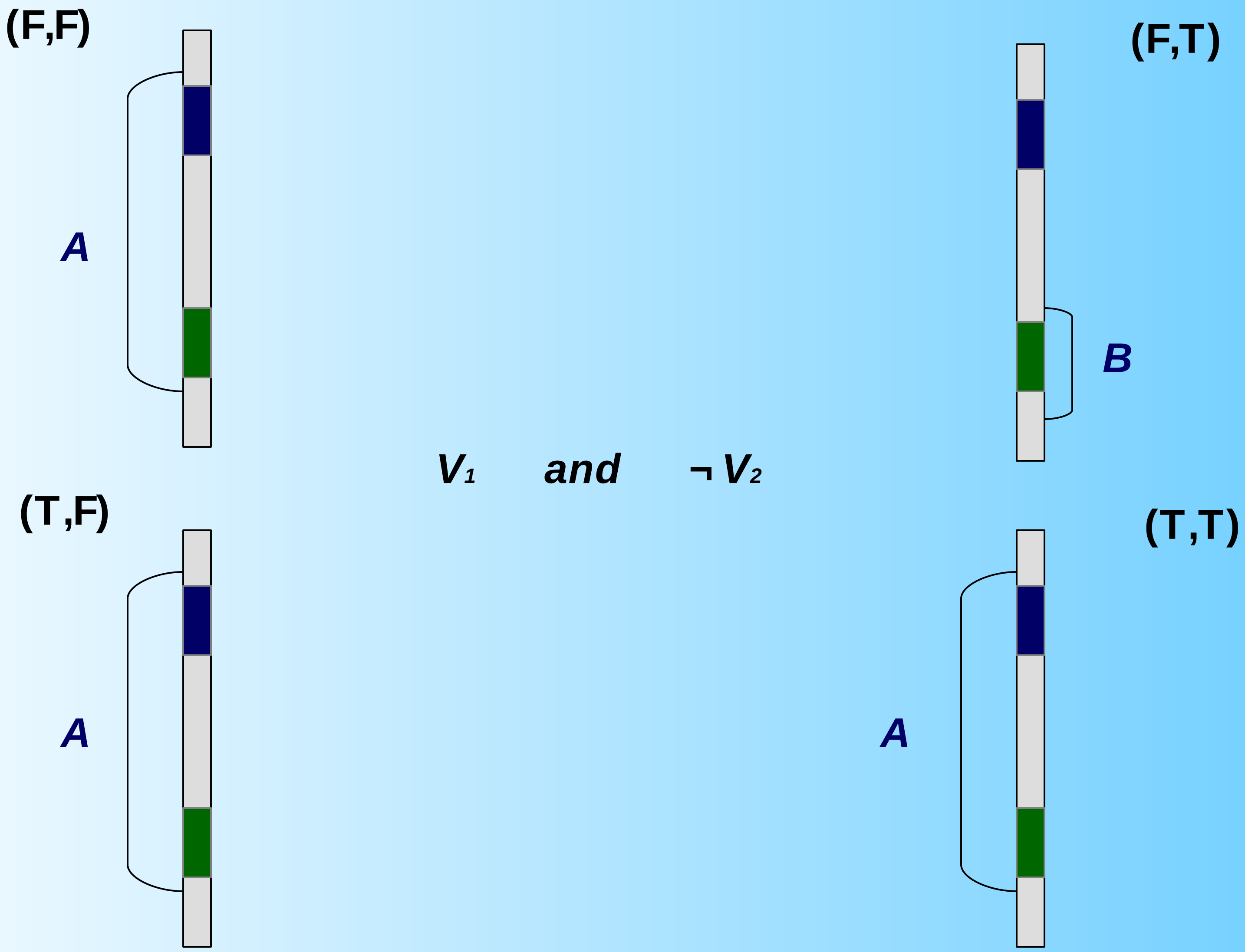


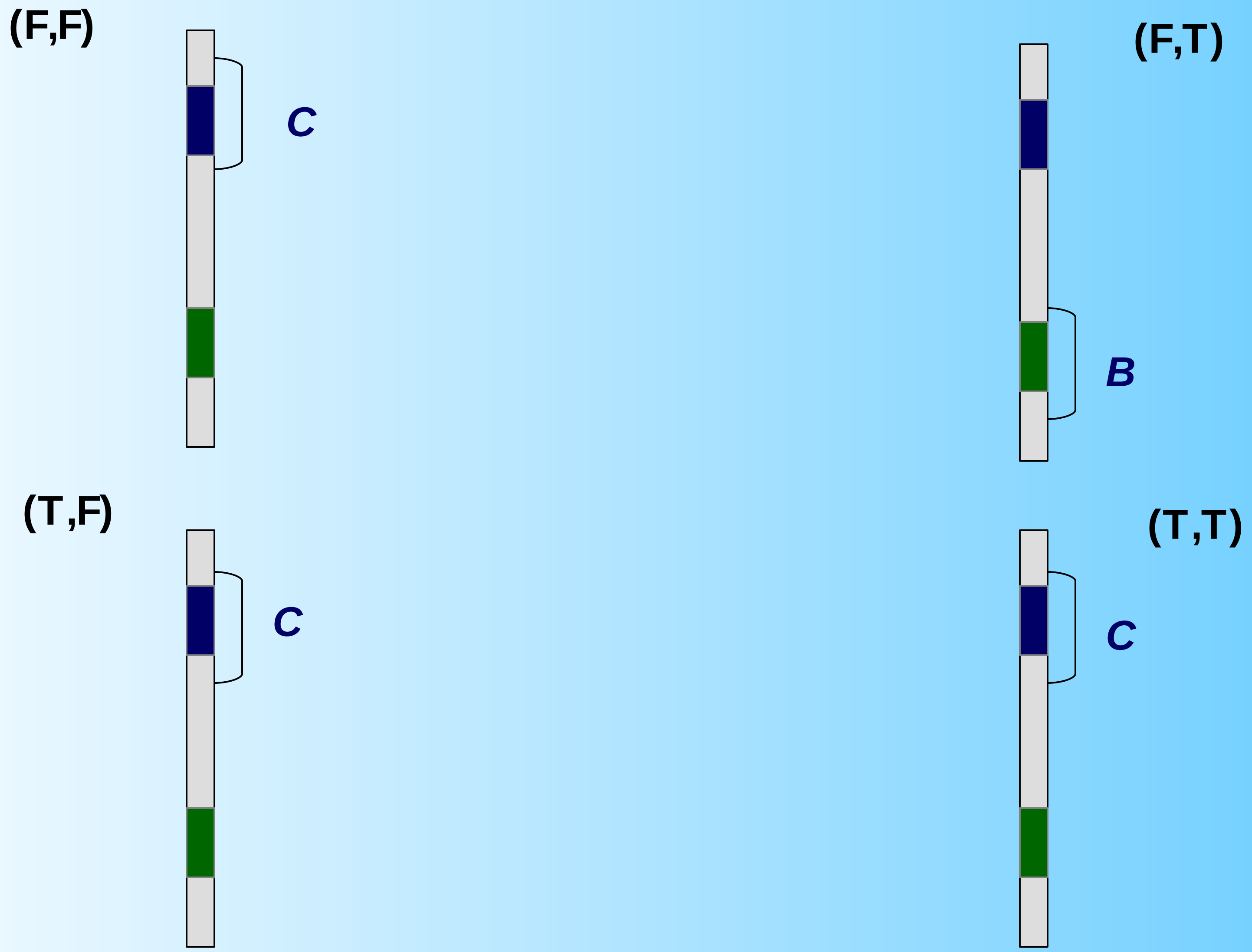
Peptide Formation (Contd..)

- Prepare partial solutions $G_1, G_2, \ldots, G_k$ where G_i contains antibody A if C_i is true under corresponding assignment X
- $G_1 = \{A_1, A_3, A_4\}$, $G_2 = \{A_1, A_3\}$, $G_3 = \{A_2, A_3, A_4\}$

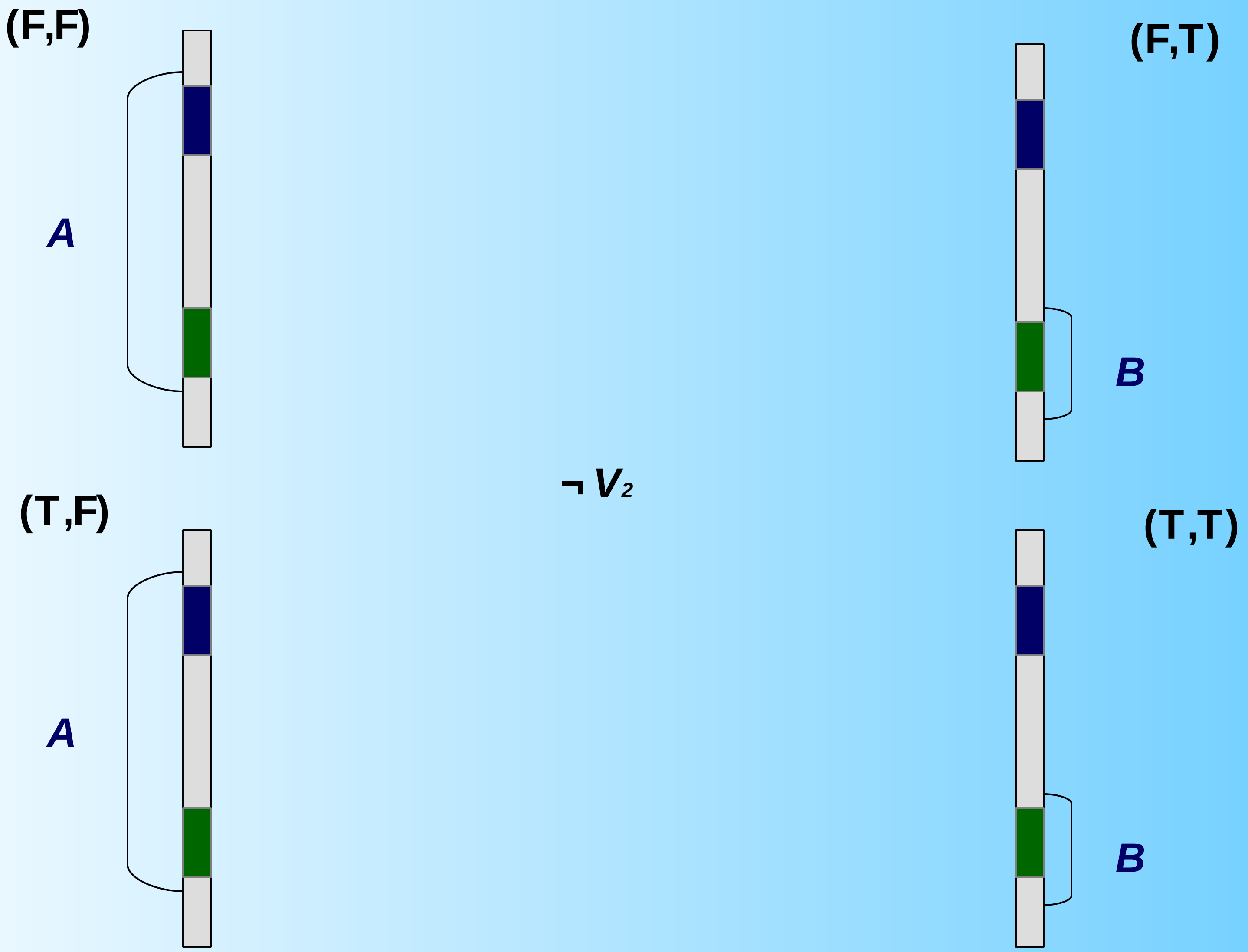
Algorithm

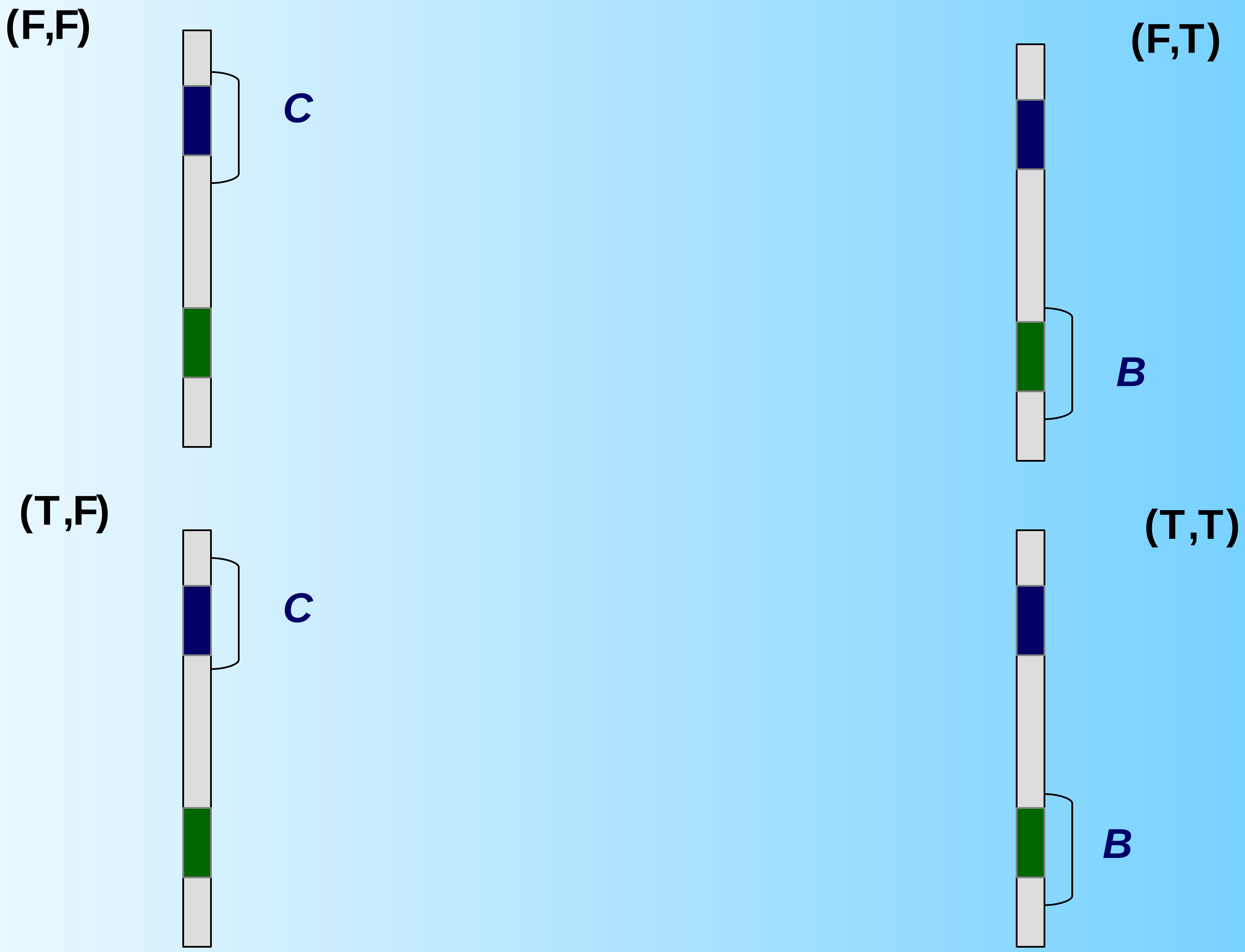
1. Let $m = k$
2. The antibody set G_k is added. The antibodies A of G_k bind to their epitopes.
3. Antibodies B are added. Antibodies B bind to all free binding sites for B.
4. Antibodies C are added.
5. Antibodies C are removed by adding epitope C in excess
6. All remaining anitbodies are covalently attached to their epitopes.
7. Let $m = m-1$. If $m > 0$ go to (2)
8. Add labeled antibodies A or B
9. Fluorescence is detected.

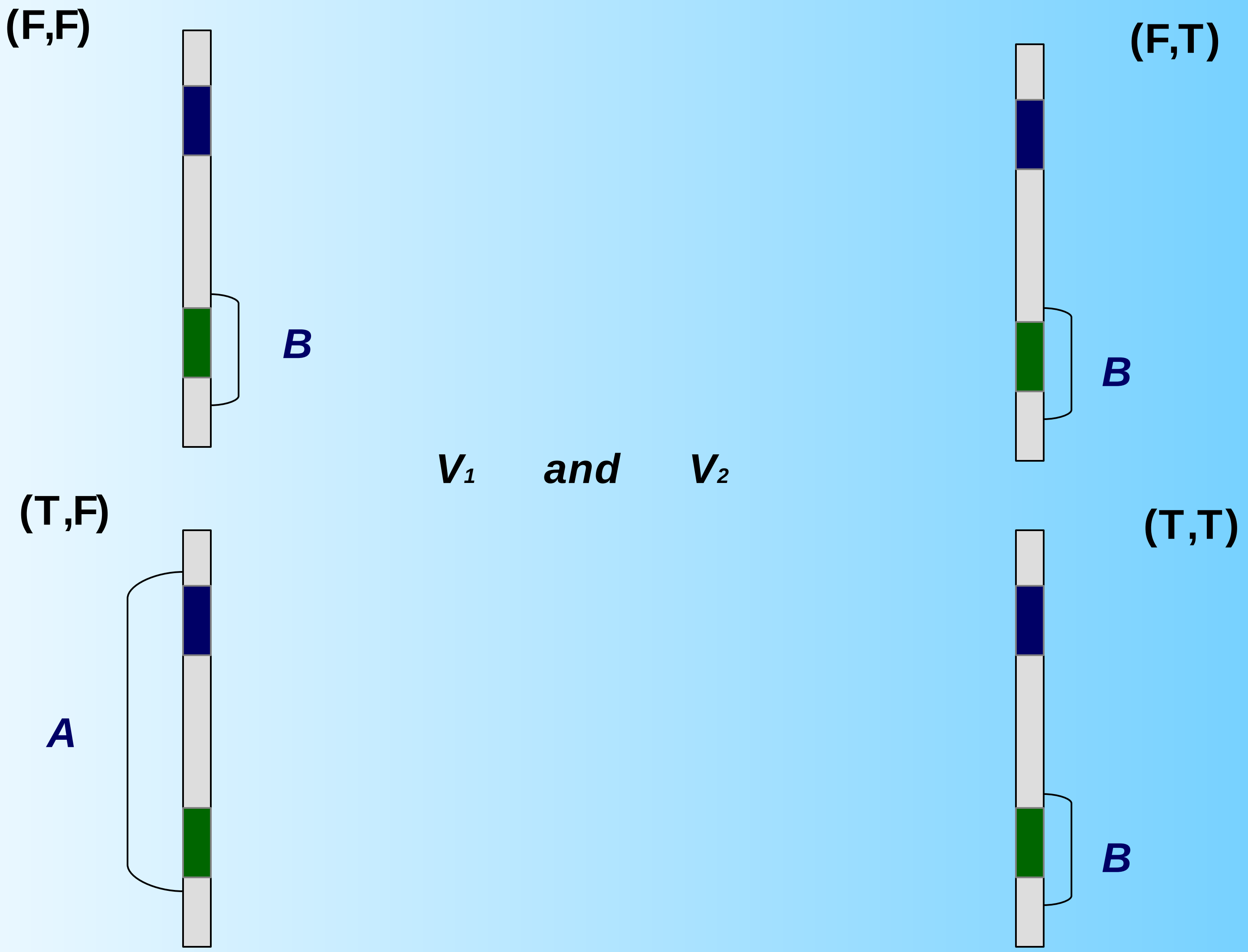

(F,F)
A
(F,T)
B
V1 and ¬V2
(T,F)
A
(T,T)
A

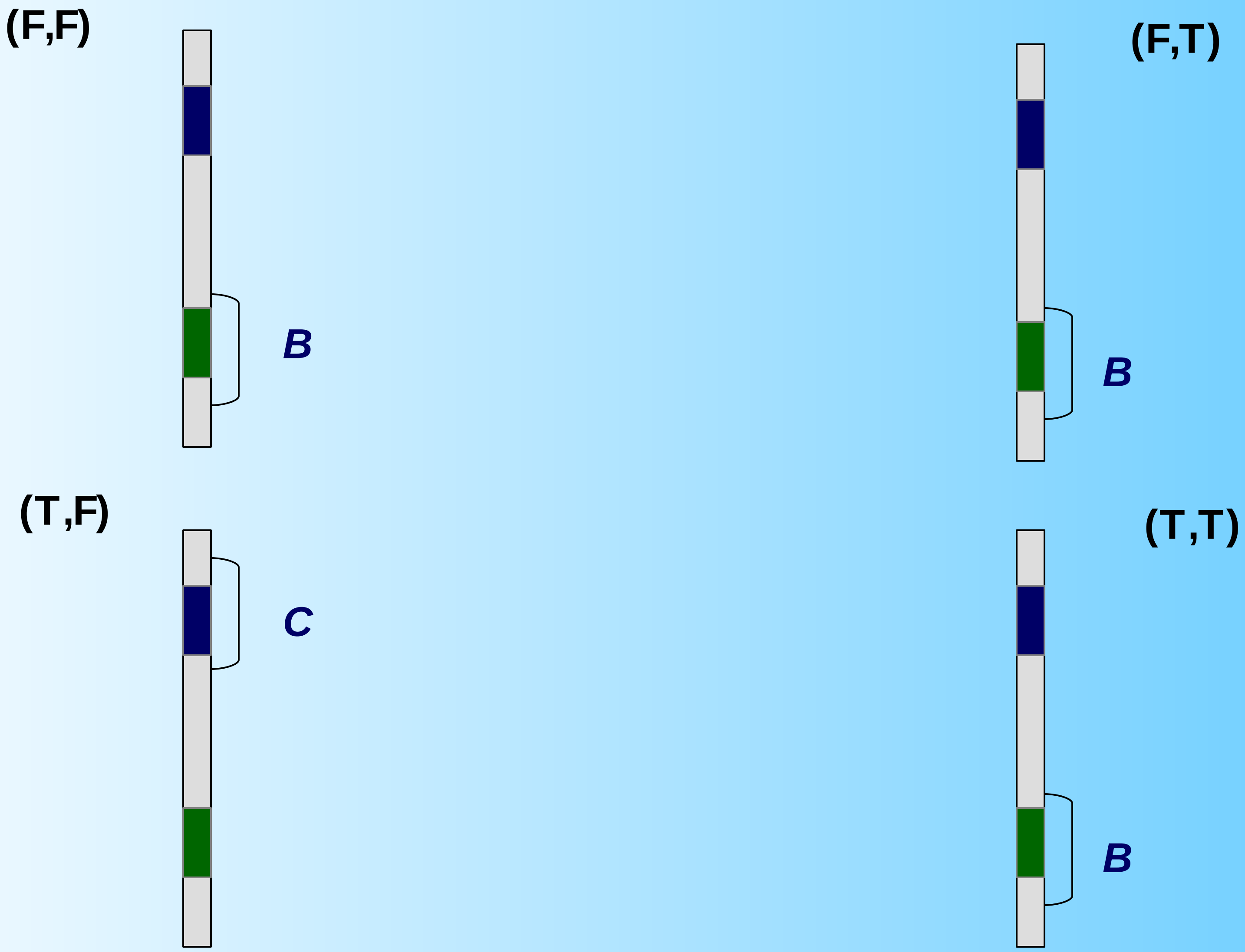


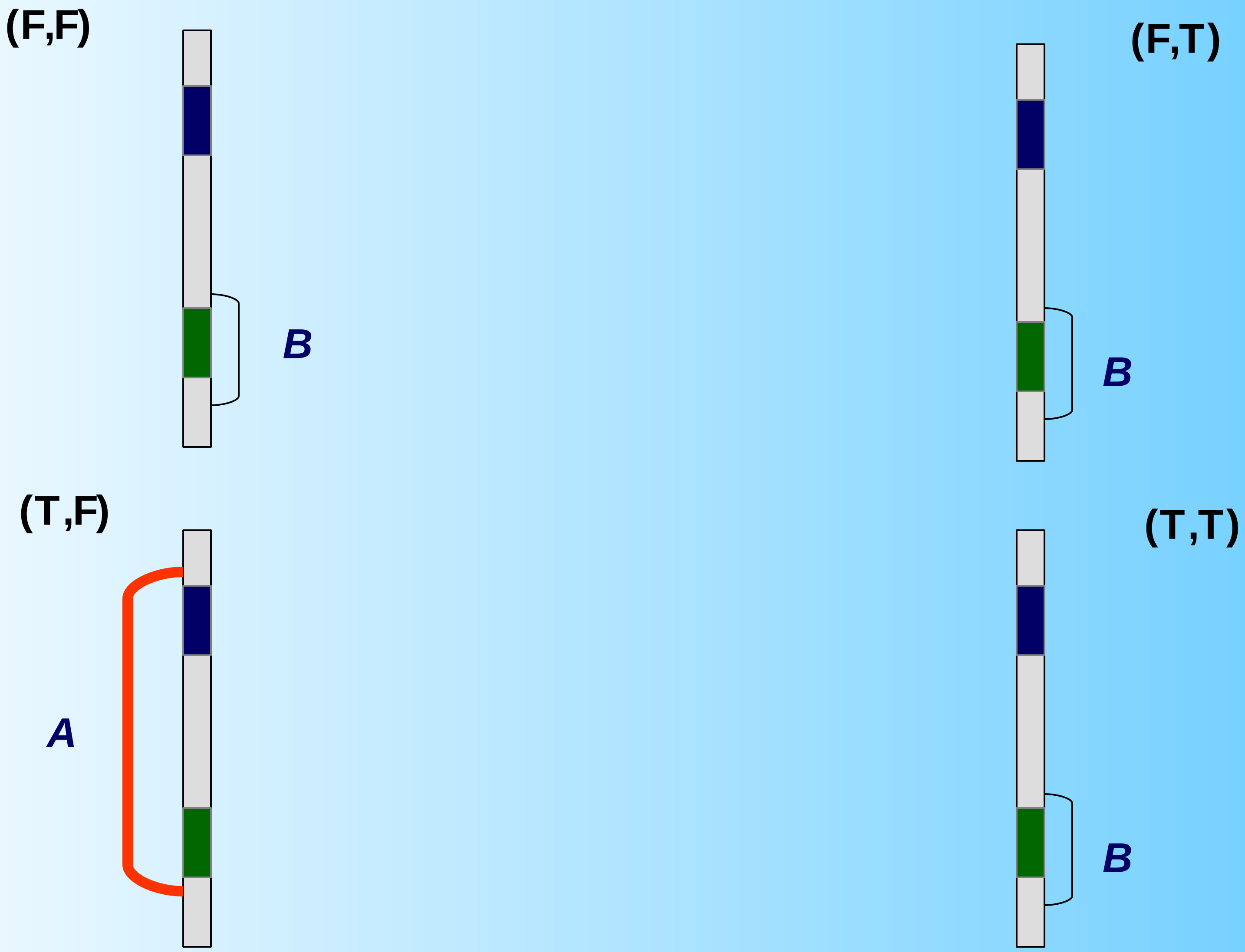
(F,F)
C
(F,T)
B
(T,F)
C
(T,T)
C


(F,F)
A
(F,T)
B
¬ V2
(T,F)
A
(T,T)
B


(F,F)
C
(F,T)
B
(T,F)
C
(T,T)
B


(F,F)
B
(F,T)
B
V1 and V2
(T,F)
A
(T,T)
B


(F,F)
B
(F,T)
B
(T,F)
C
(T,T)
B


(F,F)
B
(F,T)
B
(T,F)
A
(T,T)
B

What Next...

- Complexity Issues
- Cost effectiveness
- Implementation Difficulties
- Theoretical Model

Come forth into the light of things
Let nature be your teacher.

W. Wordsworth

Thank You