UPWARD PARTITIONED BOOK EMBEDDING

Hugo A. Akitaya, Erik D. Demaine, Adam Hesterberg,
and Quanquan C. Liu

Book Embedding

- Introduced in 1979 by Bernhart and Keinen [BK79]
- Book Embedding: Given undirected graph $G = (V, E)$:
  - Linear ordering, π, of vertices into spine
  - Disjoint partition of edges into sets so that each set can be embedded into page
- Pages join together to form book and book thickness is the minimum number of pages in any book embedding of G
- Applications in VLSI design, parallel process scheduling, others

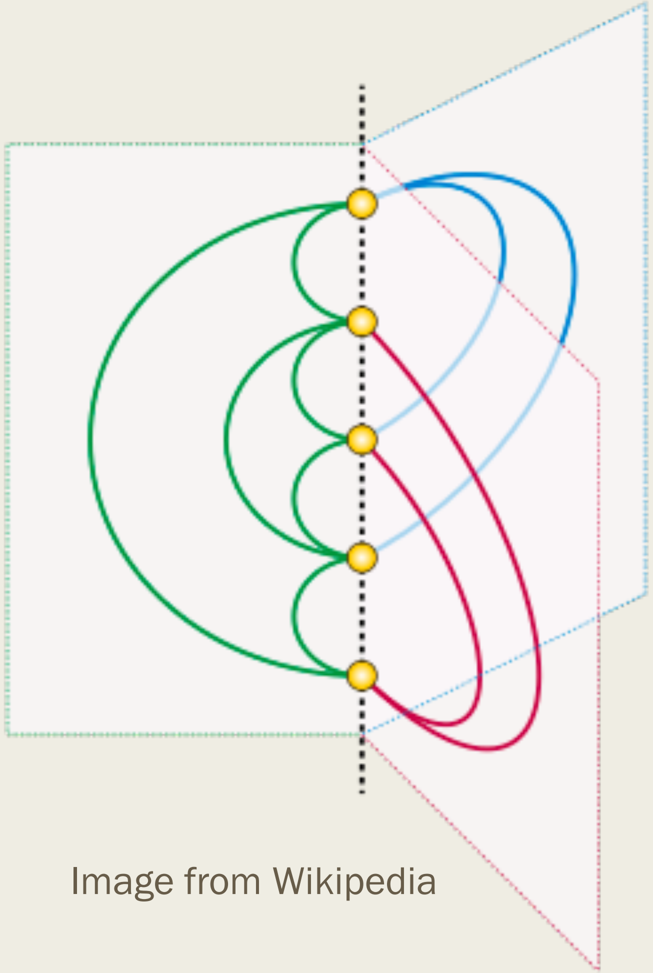
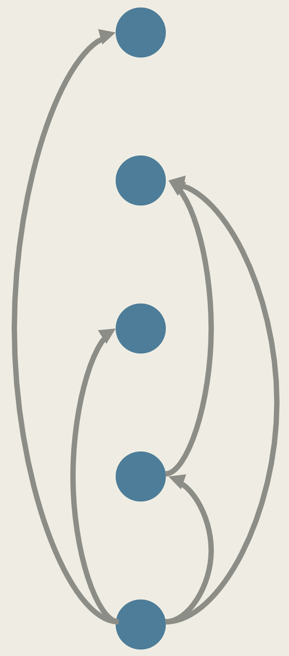


Image from Wikipedia

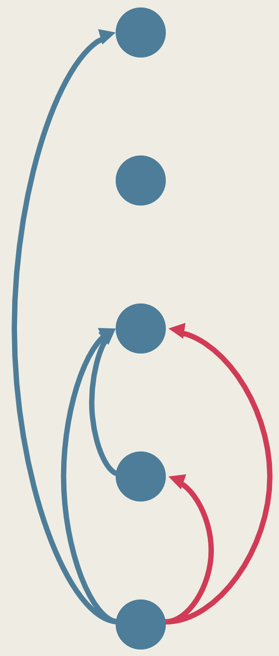
Directed Graphs and Partitioned Problem

- Upward book embedding [HP97, HPT99]: Given directed DAG, embed the DAG such that the ordering on the spine is in topological order



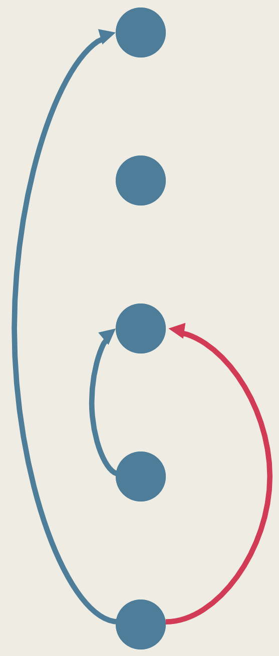
Directed Graphs and Partitioned Problem

- Upward book embedding [HP97, HPT99]: Given directed DAG, embed the DAG such that the ordering on the spine is in topological order
- Upward partitioned book embedding (Upward): Given partition of edges into k pages (k disjoint sets of edges), linearly order vertices on spine



Directed Graphs and Partitioned Problem

- Upward book embedding [HP97, HPT99]: Given directed DAG, embed the DAG such that the ordering on the spine is in topological order
- Upward partitioned book embedding (Upward): Given partition of edges into k pages (k disjoint sets of edges), linearly order vertices on spine
- Upward matching-partitioned book embedding (Matching): Edge partitions form a matching

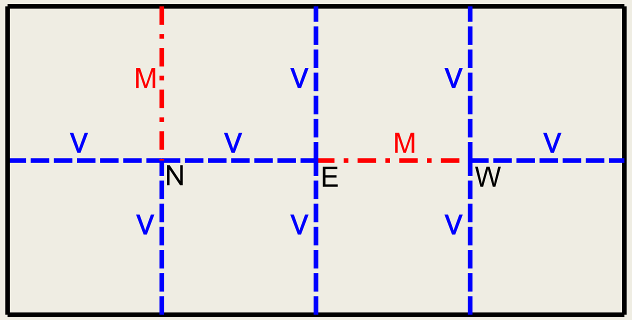


Results: Old and New (All Partitioned)

Type	$k = 1$	$k = 2$	$k = 3$	$k \geq 4$
Undirected	$O(n)$ [BK79]	$O(n)$ [HN14]	NP-complete [ALN15]	NP-complete [ALN15]
Upward	$O(n)$ [HP99]	*OPEN*	NP-complete [Theorem 1]	NP-complete [Theorem 1]
Matching	$O(n)$ [HP99]	$O(n)$ [Theorem 3]	*OPEN*	NP-complete [Theorem 2]

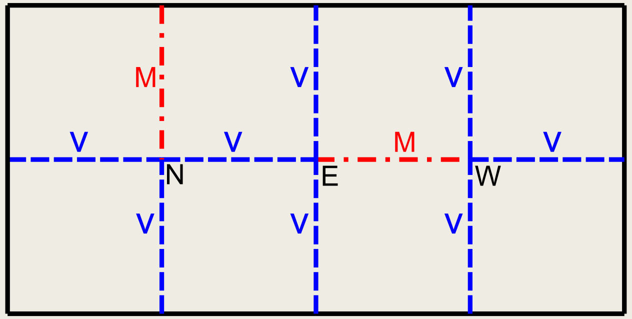
Connections to Origami

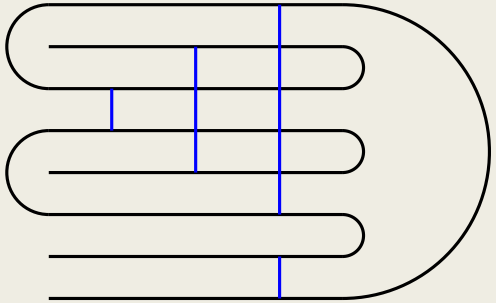
- k page Matching motivated by map folding problem
- Map folding problem: Given an $m \times n$ grid pattern with specified mountains and valleys, find a flat folded state
- Reduces to Matching



Connections to Origami

- k page Matching motivated by map folding problem
- Map folding problem: Given an $m \times n$ grid pattern with specified mountains and valleys, find a flat folded state
- Reduces to Matching
- $1 \times n \to 2$ page Matching (linear time) and $2 \times n \to 3$ page Matching
- $2 \times n$ map folding currently has complicated $O(n^9)$ algorithm
- Open Question: Can we get a better algorithm for $2 \times n$ map folding via reduction to 3 page Matching?





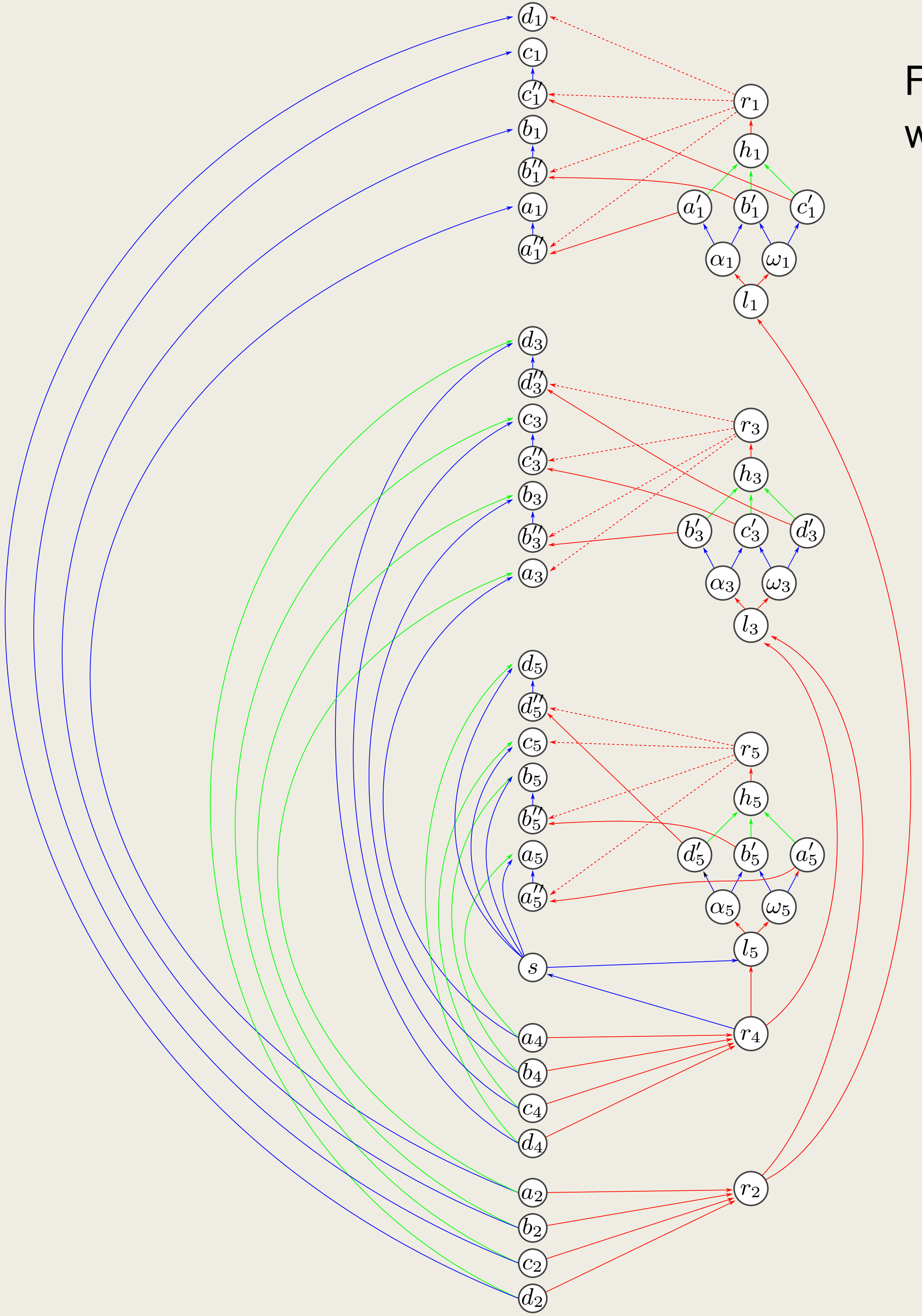
3 page Upward is NP-complete

- Reduction from NP-complete problem BETWEENNESS
- Betweenness:
  - Given L variables and C clauses where $\langle a, b, c\rangle$ is a clause
  - Find total ordering ϕ such that $\phi(a) < \phi(b) < \phi(c)$ or $\phi(c) < \phi(b) < \phi(a)$ is true for all clauses
- Given instance (L, C) of betweenness construct instance of 3 page Matching (G, P) where $P = \{RED, BLUE, GREEN\}$
- A solution π to (G, P) corresponds with a solution ϕ to (L, C)

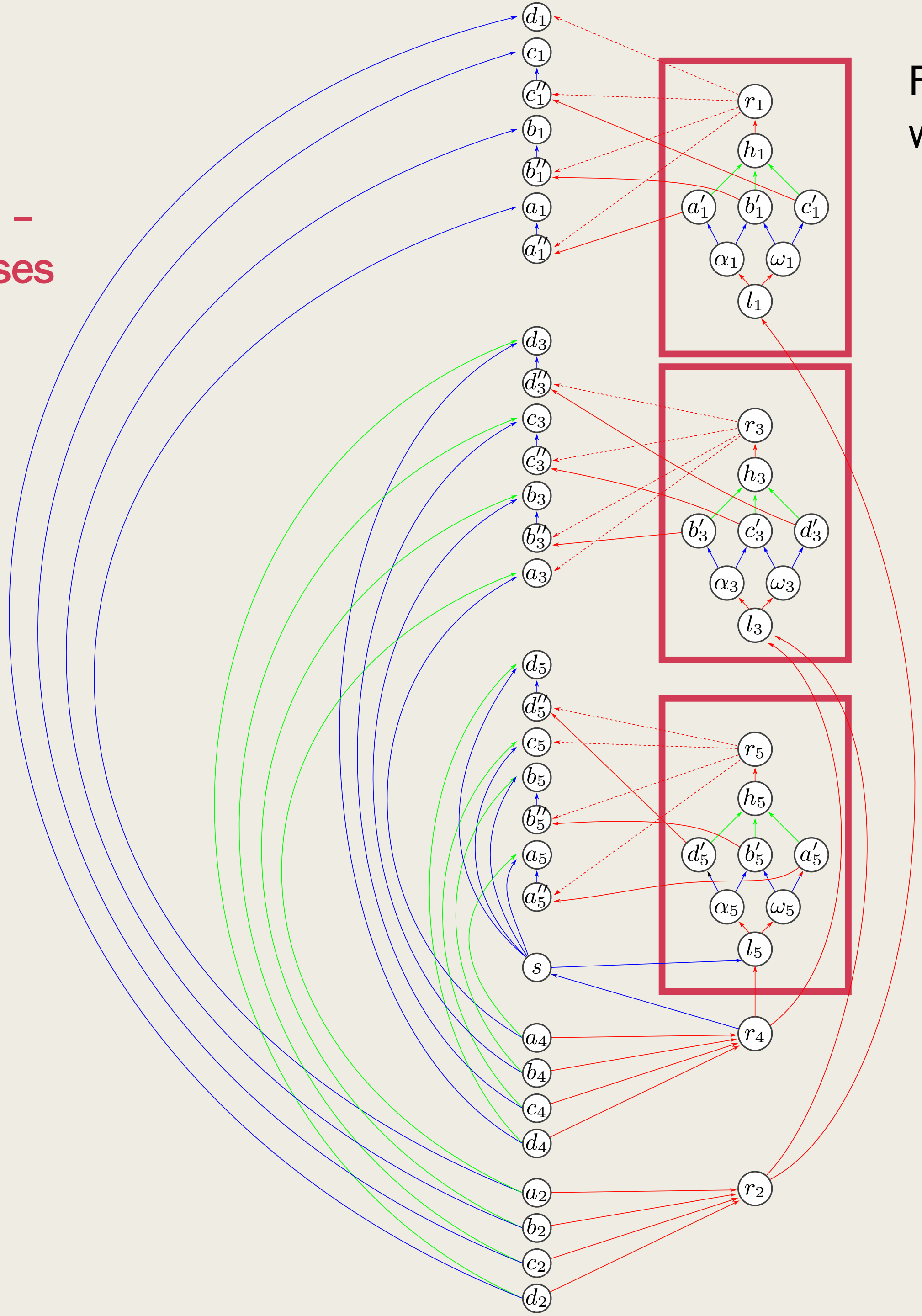
Full construction of (G, P) from (L, C)

where $L = \{a, b, c, d\}$ and

$$C = \{\langle d, b, a\rangle, \langle b, c, d\rangle, \langle a, b, c\rangle\}$$



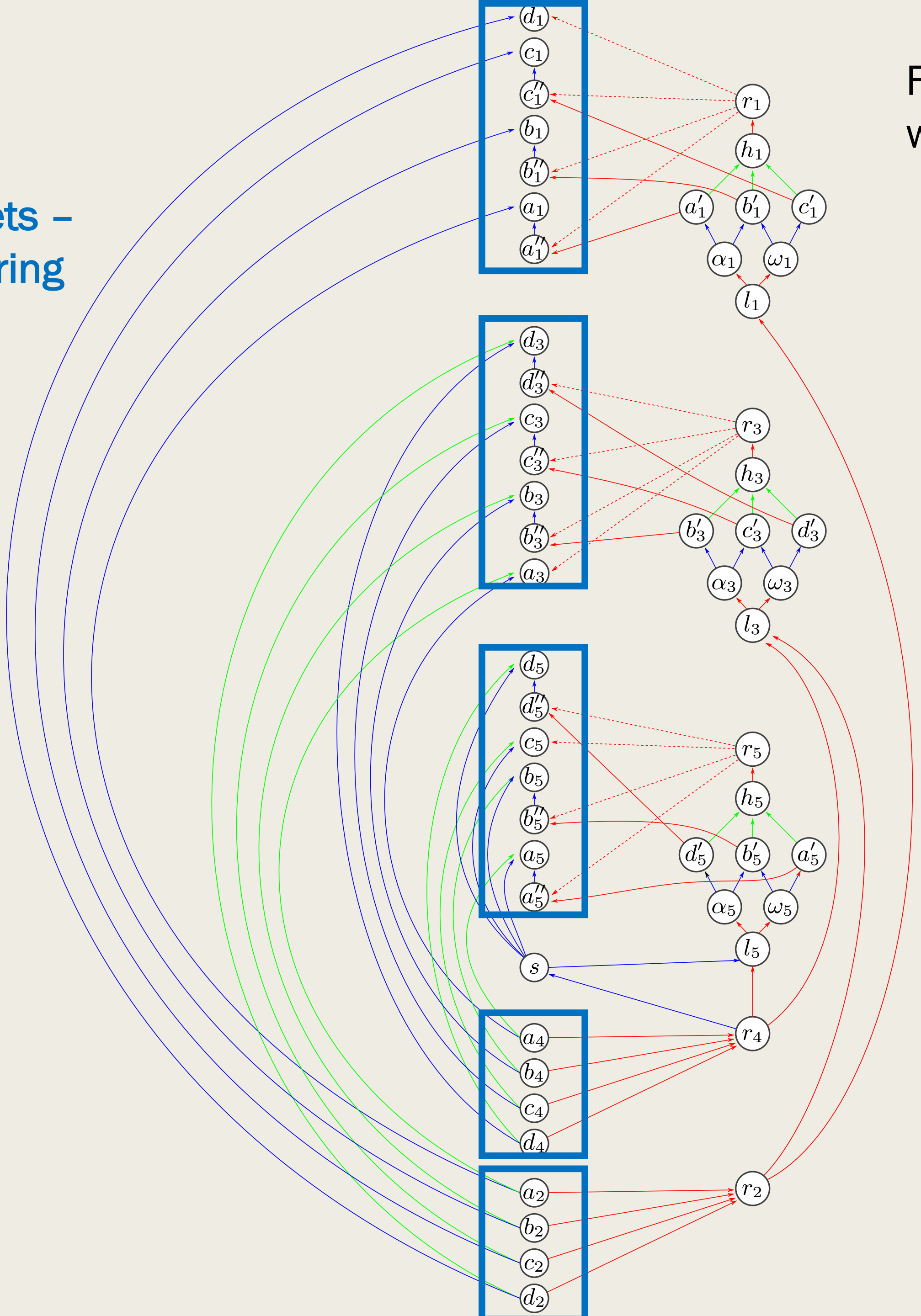
Ordered Triple Gadgets – used to represent clauses



Full construction of (G, P) from (L, C) where $L = \{a, b, c, d\}$ and

$C = \{\langle d, b, a\rangle, \langle b, c, d\rangle, \langle a, b, c\rangle\}$

Order preserving gadgets – used to represent ordering of variables

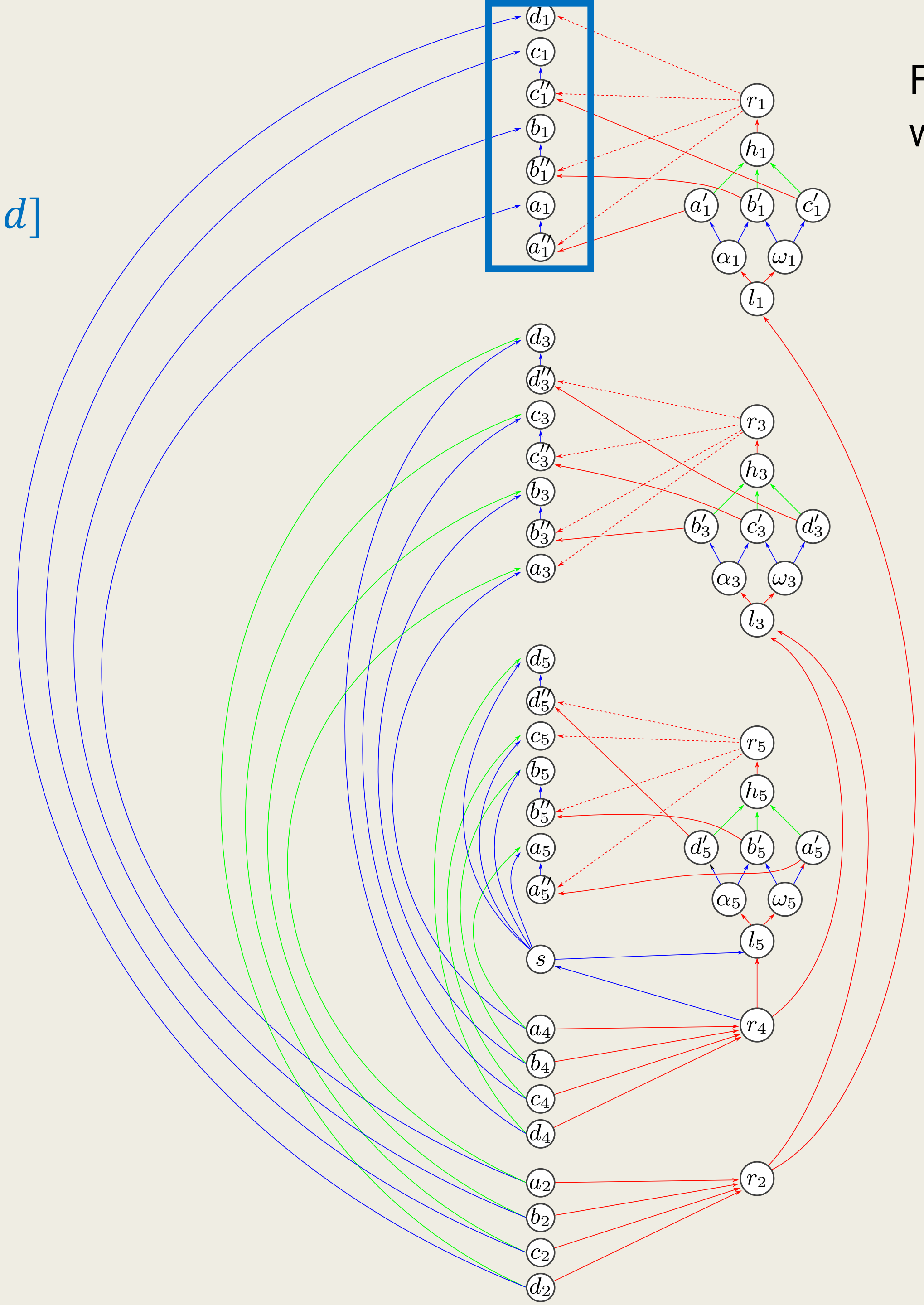


Full construction of (G, P) from (L, C) where $L = \{a, b, c, d\}$ and

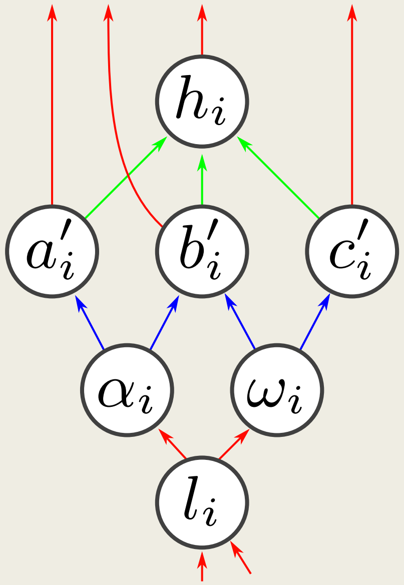
$$C = \{\langle d, b, a\rangle, \langle b, c, d\rangle, \langle a, b, c\rangle\}$$

Full construction of (G, P) from (L, C) where $L = \{a, b, c, d\}$ and
$C = \{\langle d, b, a\rangle, \langle b, c, d\rangle, \langle a, b, c\rangle\}$

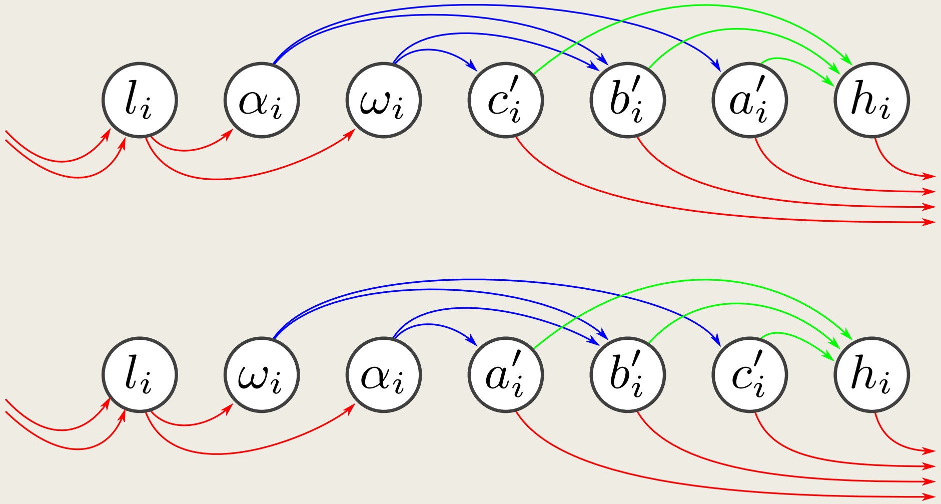
Ordering $\phi = [a, b, c, d]$



Ordered Triple Gadgets

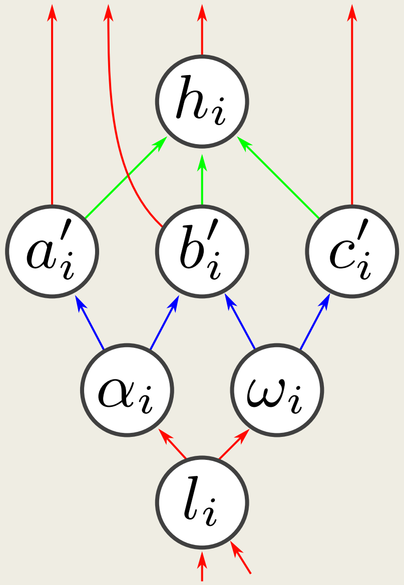


Ordered triple gadget for clause $\langle a, b, c \rangle$



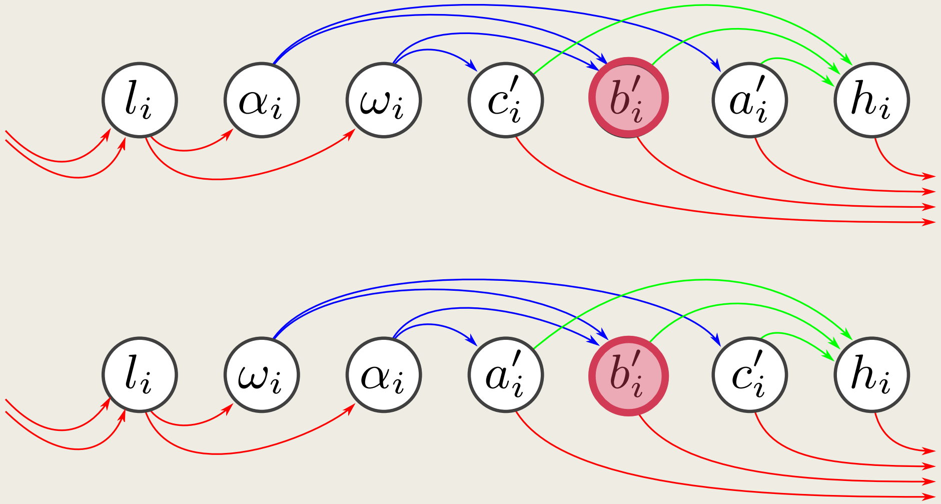
Two ways of embedding ordered triple gadget

Ordered Triple Gadgets



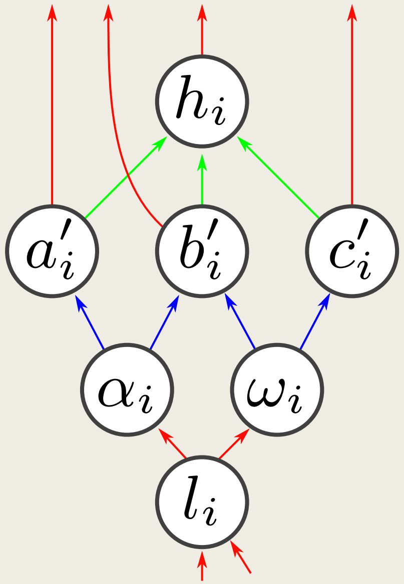
Ordered triple gadget for clause
$\langle a, b, c \rangle$

b always stays between *a* and *c*



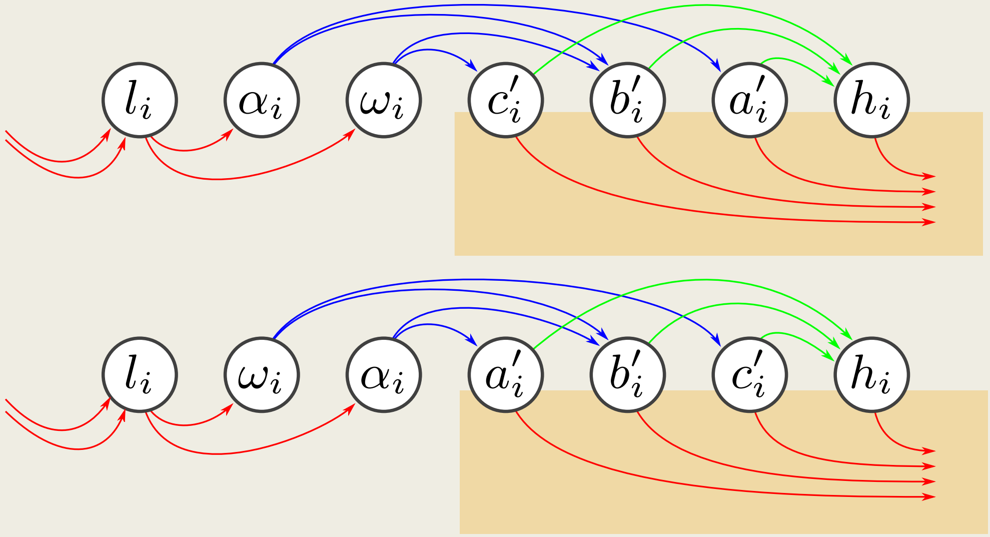
Two ways of embedding ordered triple gadget

Ordered Triple Gadgets



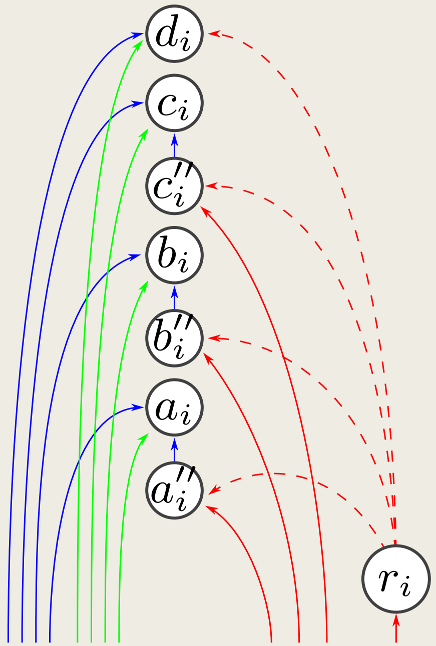
Ordered triple gadget for clause $\langle a, b, c \rangle$

Red edges connect to order preserving gadgets

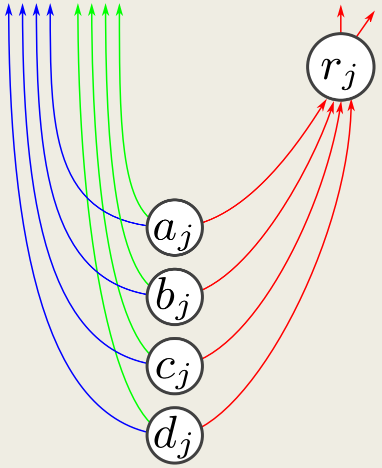


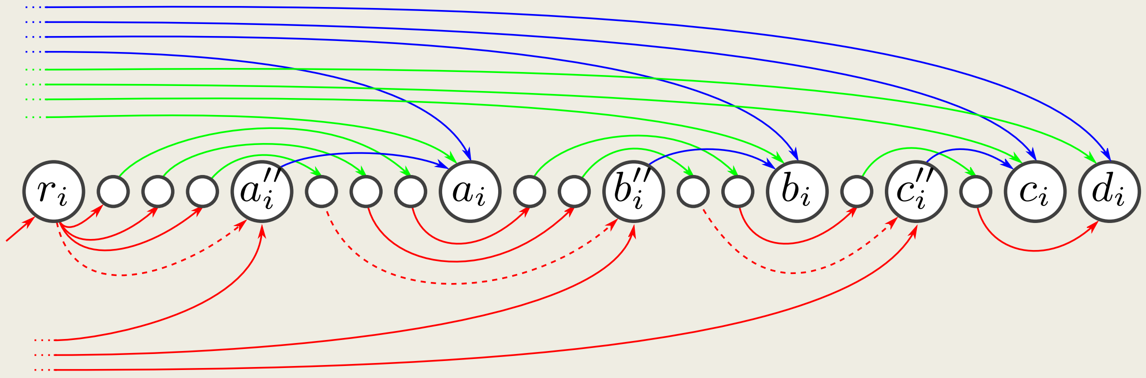
Two ways of embedding ordered triple gadget

Order Preserving Gadgets

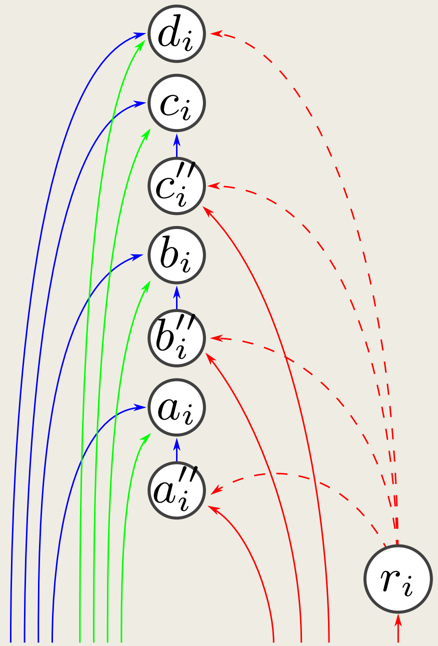


Odd index order preserving gadget

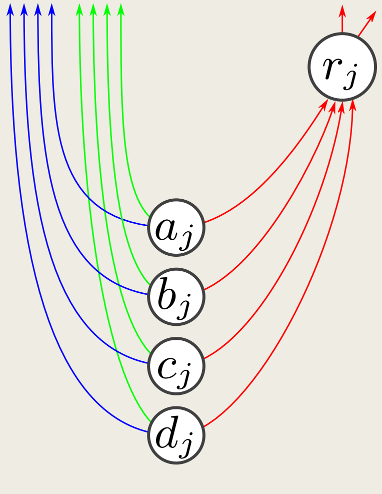




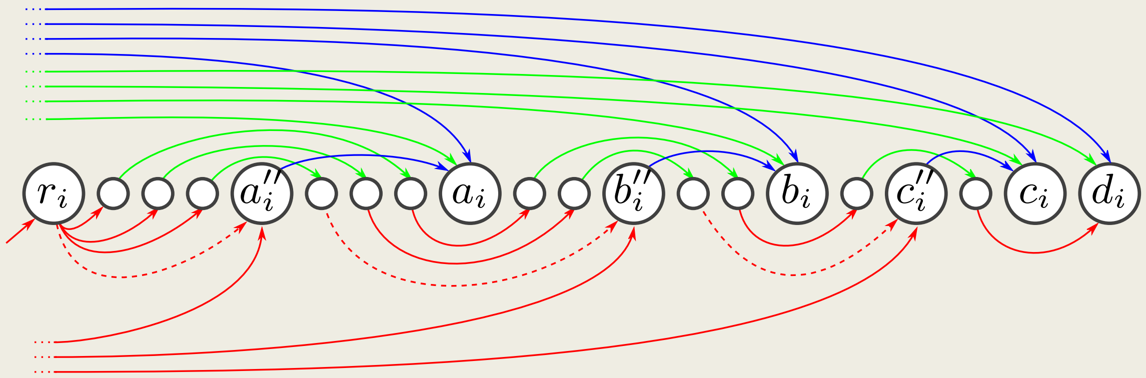
Order Preserving Gadgets



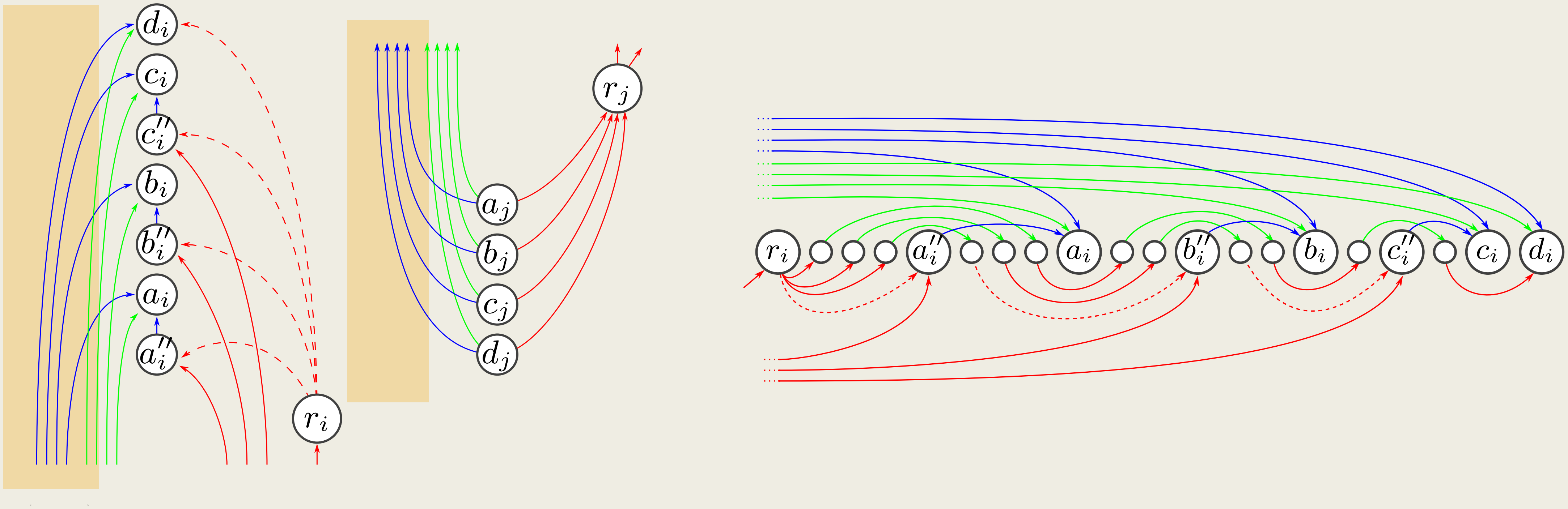
Odd index order preserving gadget



Even index order preserving gadget

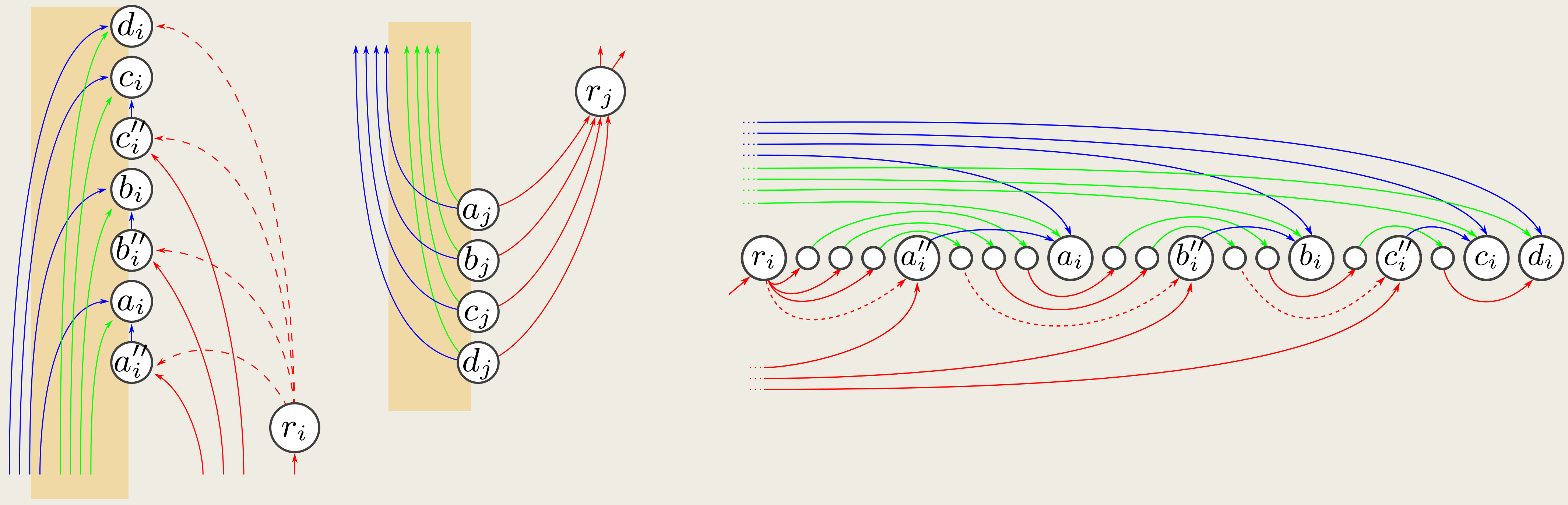


Order Preserving Gadgets



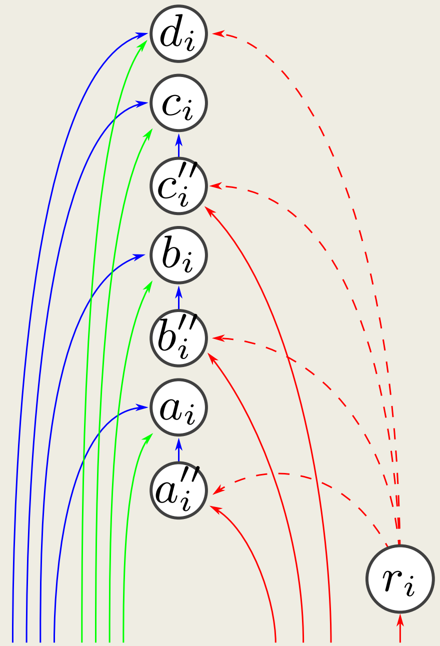
Blue edges go from even index *j* to odd index *j - 1*

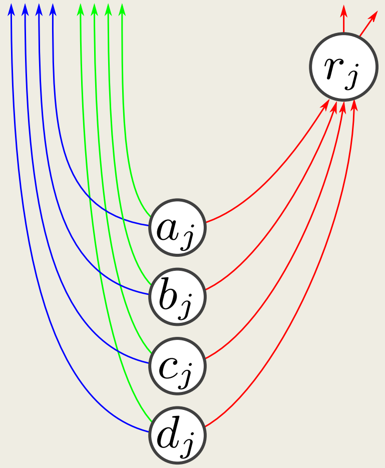
Order Preserving Gadgets

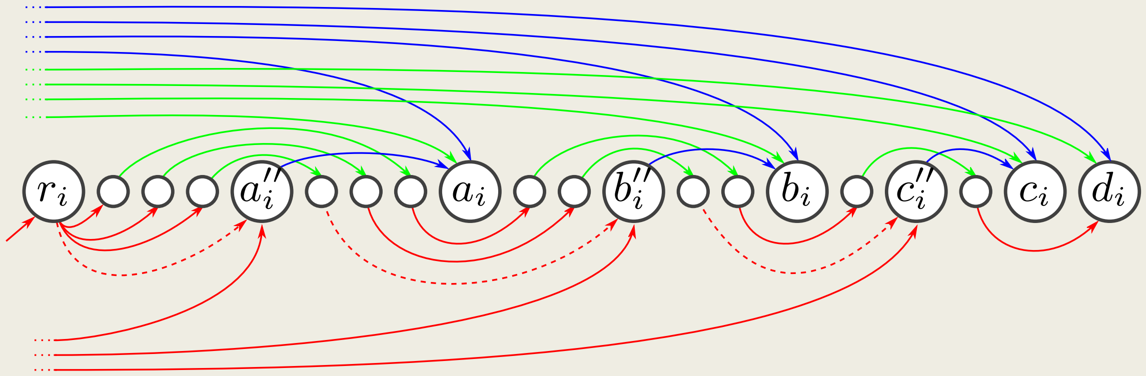


Green edges go from even index *j* to odd index *j* + *1*

Order Preserving Gadgets

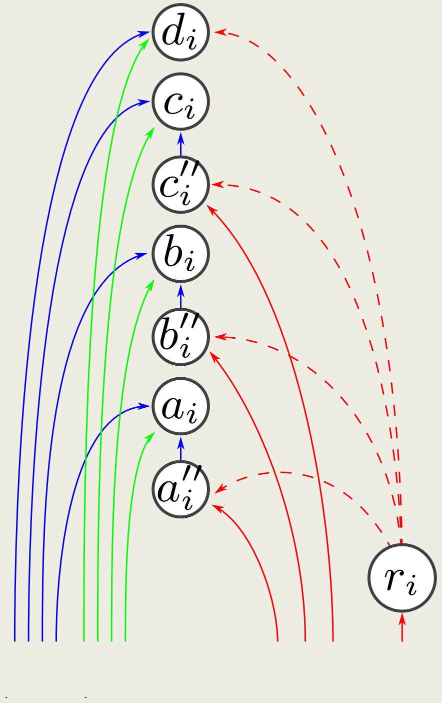


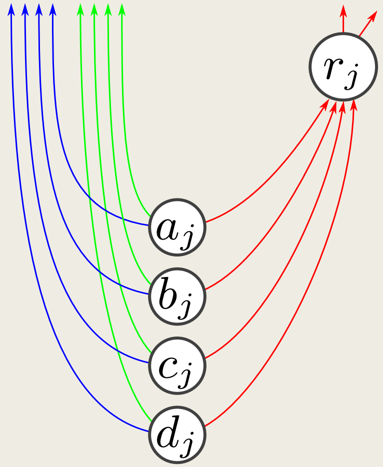


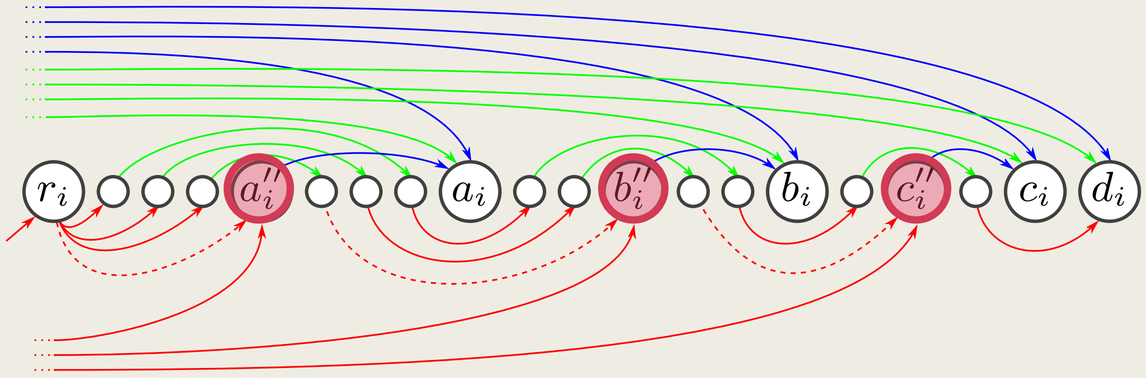


Valid embedding of order preserving gadget

Order Preserving Gadgets

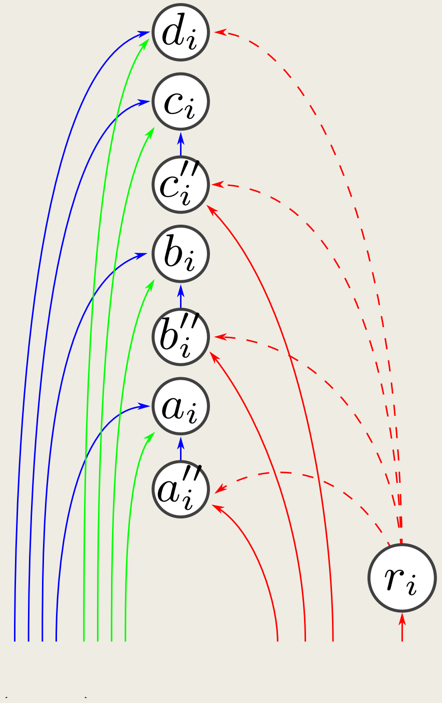


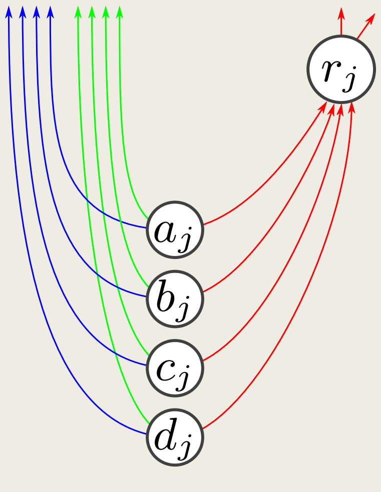


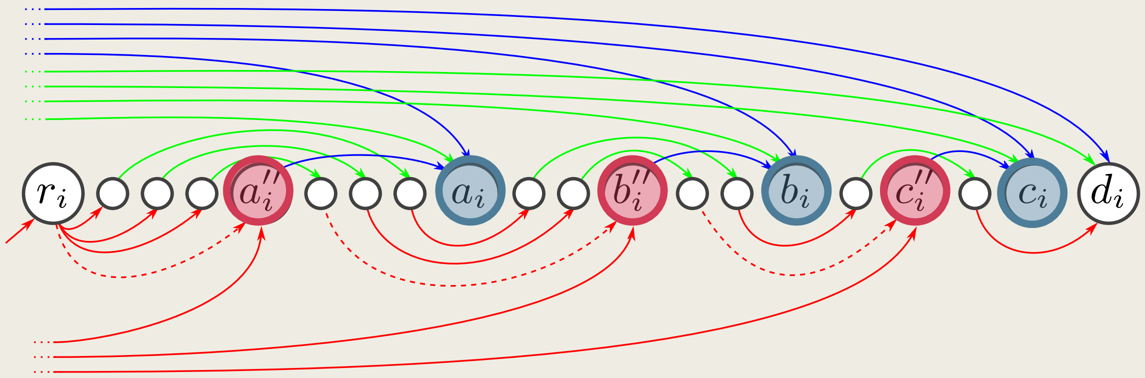


Red edges from ordered triple gadget ensures valid ordering of variables in clauses

Order Preserving Gadgets

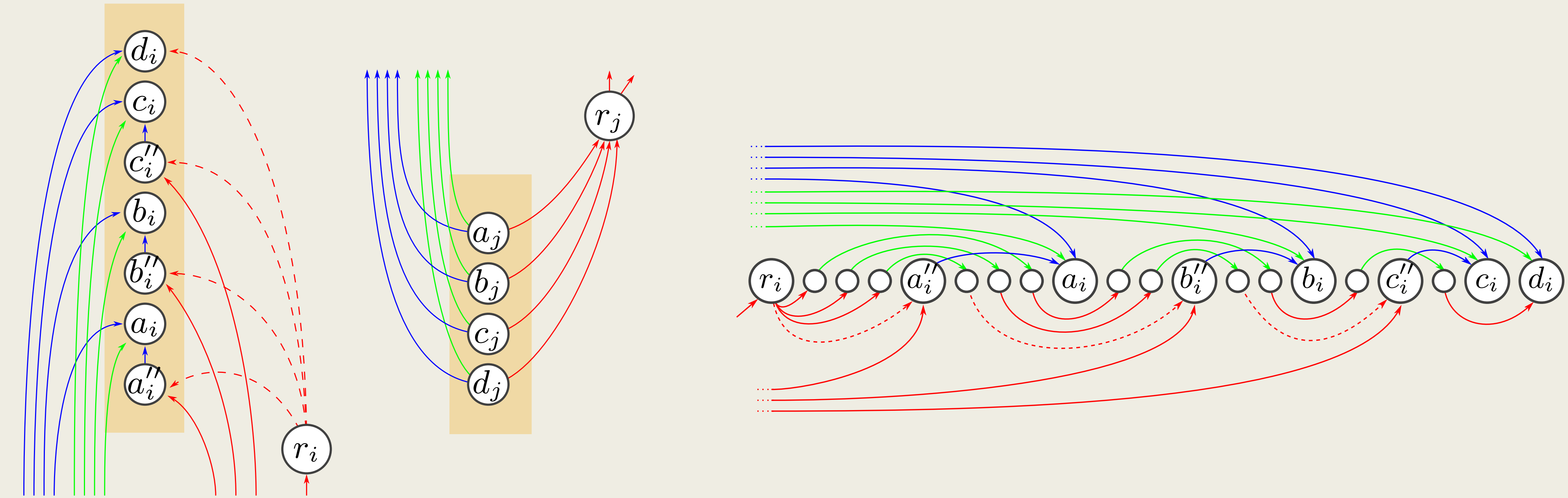






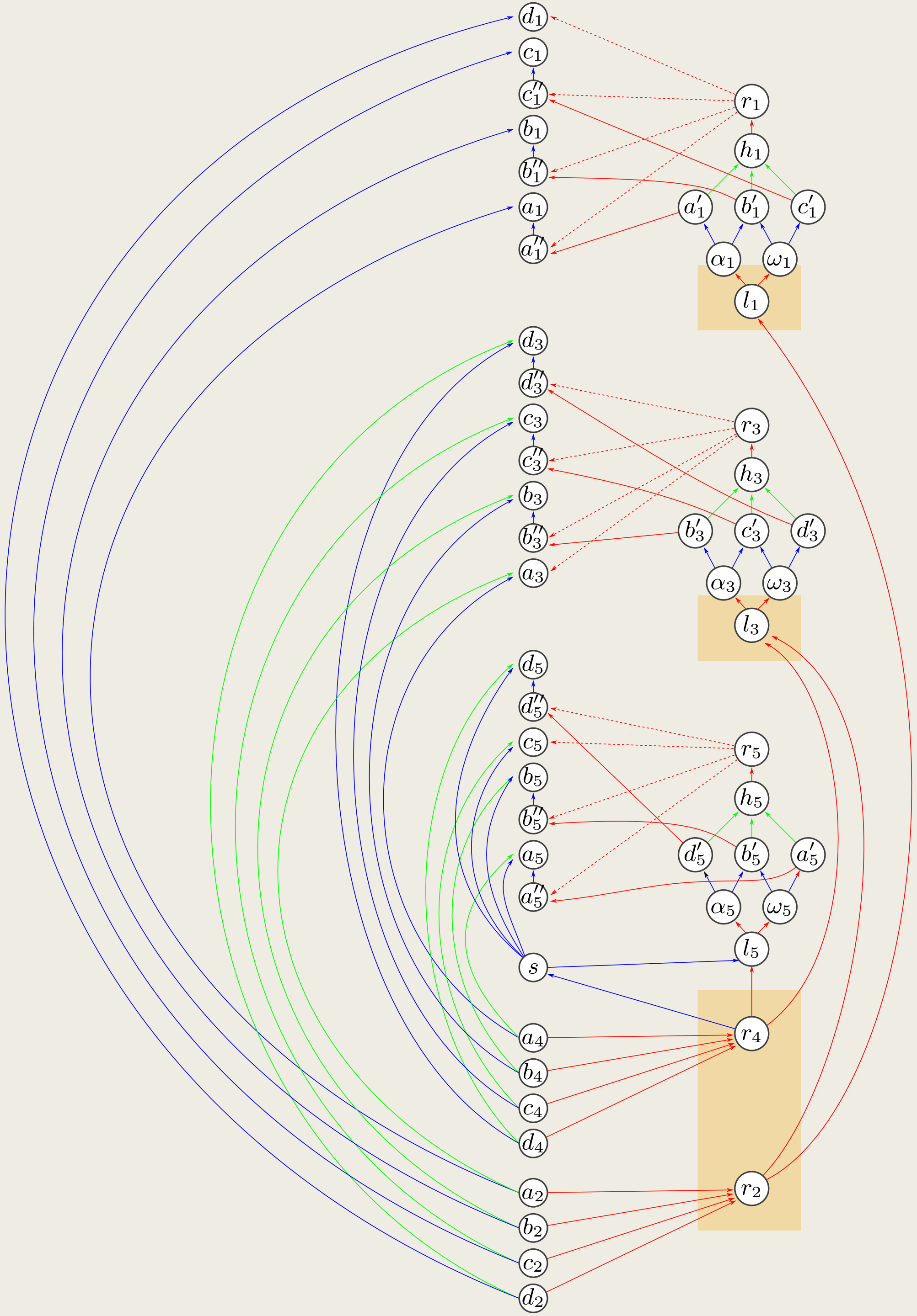
Red edges from ordered triple gadget ensures valid ordering of variables in clauses and corresponding vertices in the order preserving gadget

Order Preserving Gadgets



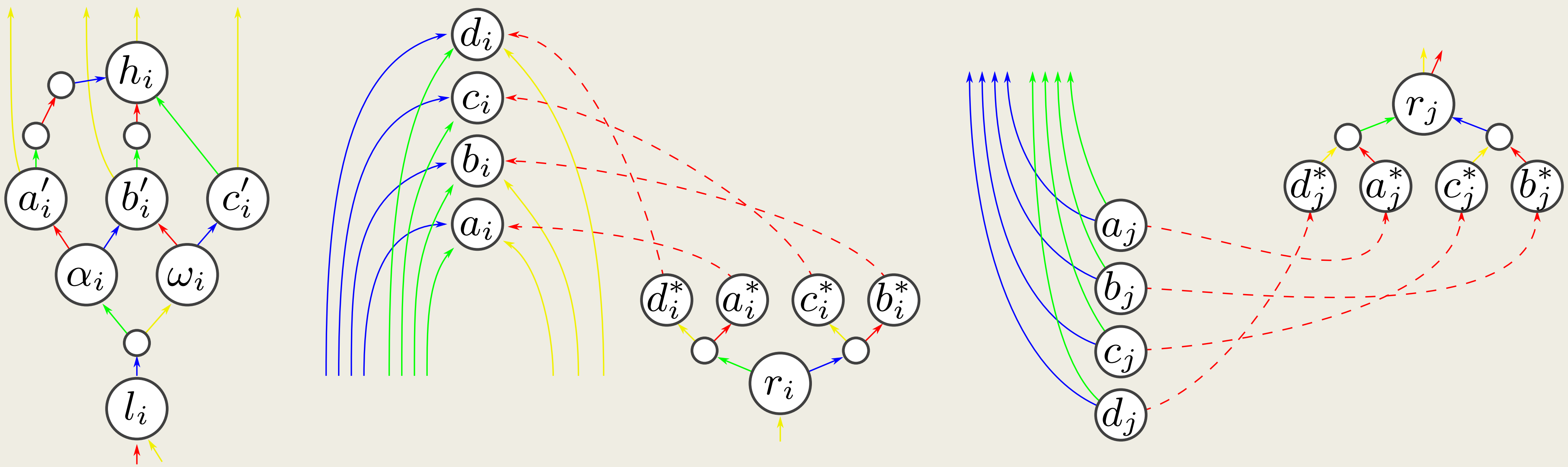
Order of a_i, b_i, c_i, d_i in π represent valid ordering of a, b, c, d in ϕ

Full construction



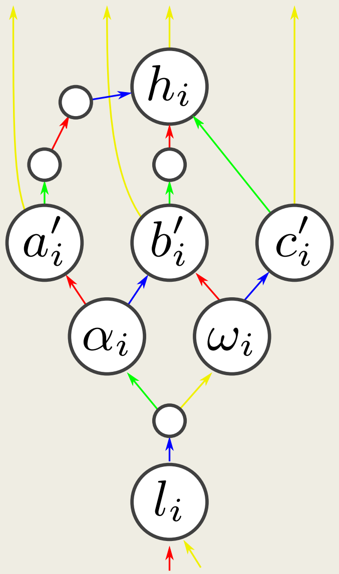
Maintains order of ordered triple gadgets and order preserving gadgets

4 page Matching is NP-complete

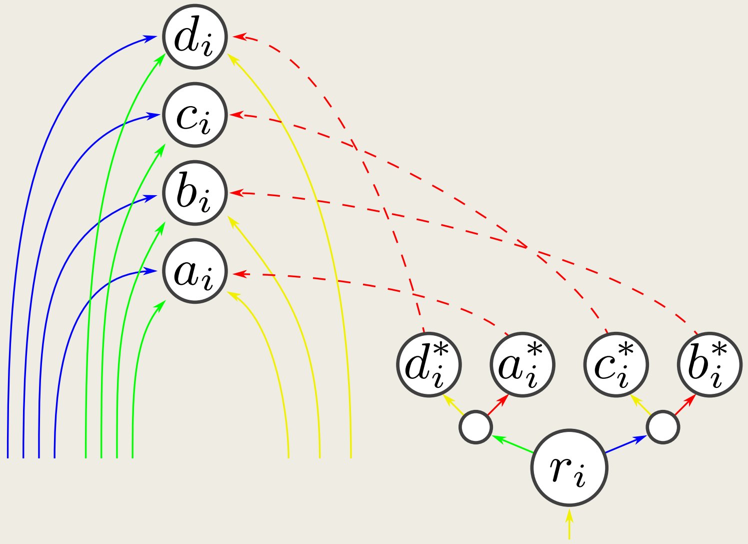


Ordered triple gadget

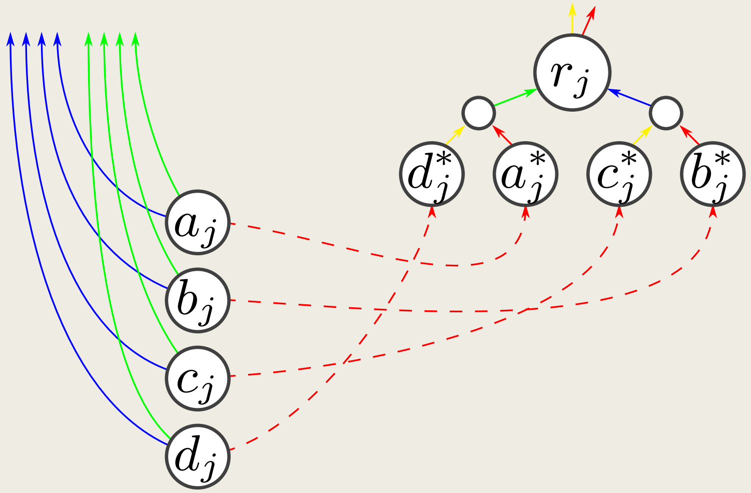
4 page Matching is NP-complete



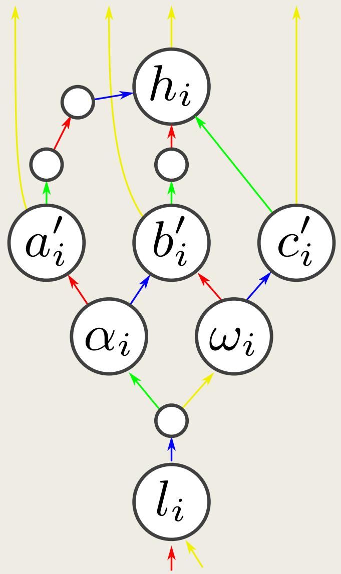
Ordered triple gadget



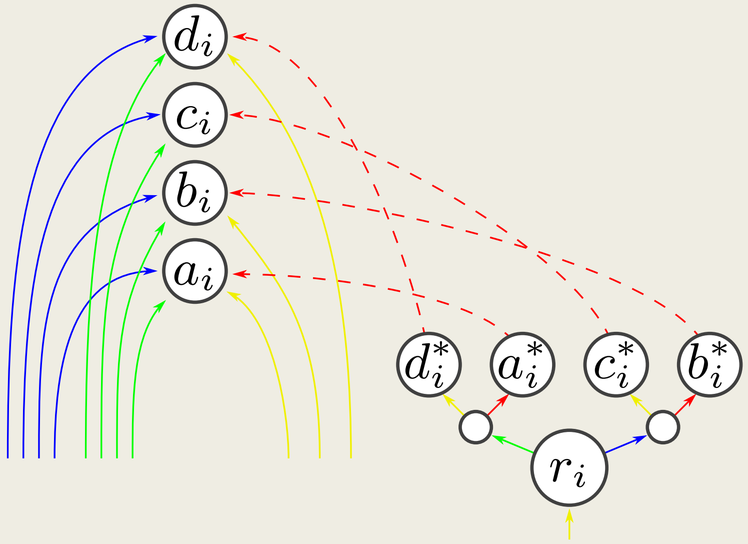
Odd index order preserving gadget



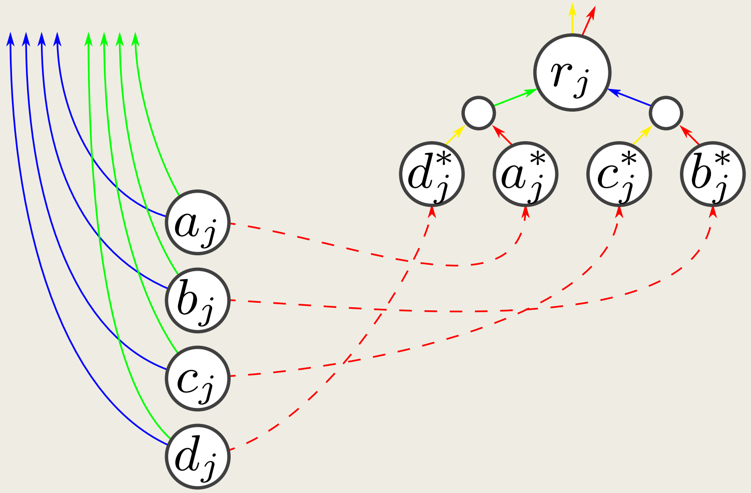
UMPBE-4 is NP-complete



Ordered triple gadget

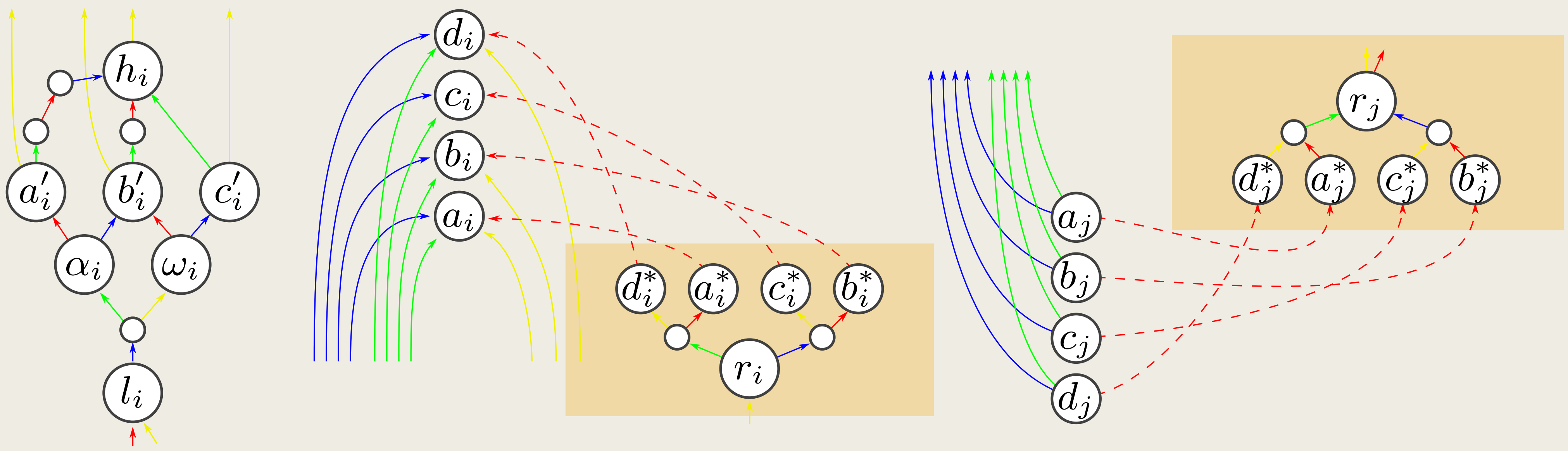


Odd index order preserving gadget



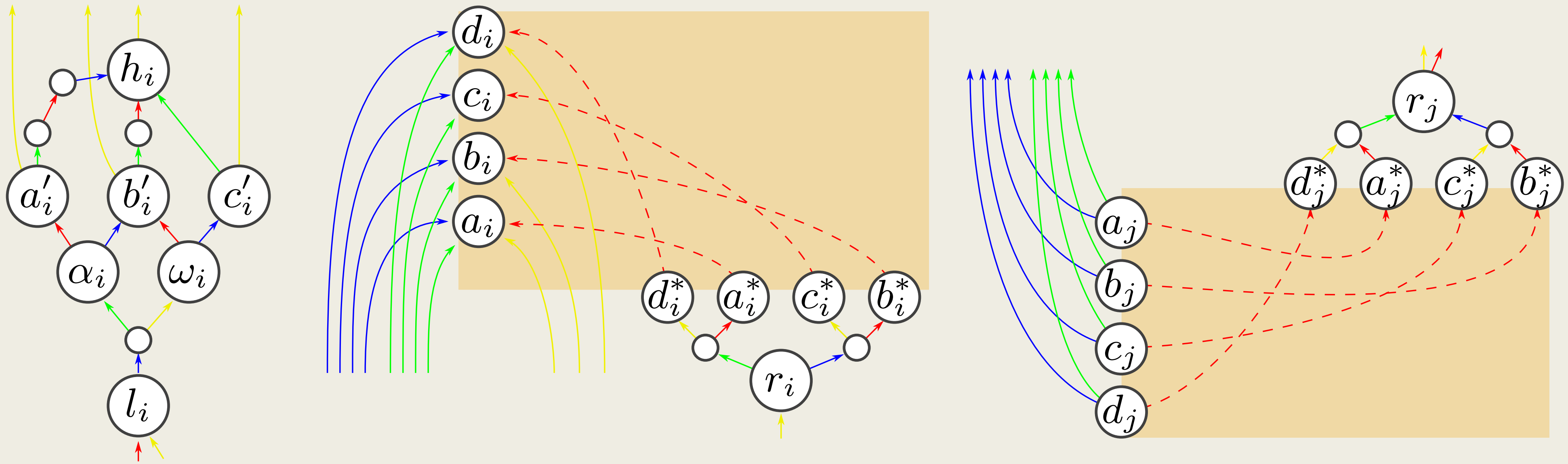
Even index order preserving gadget

4 page Matching is NP-complete



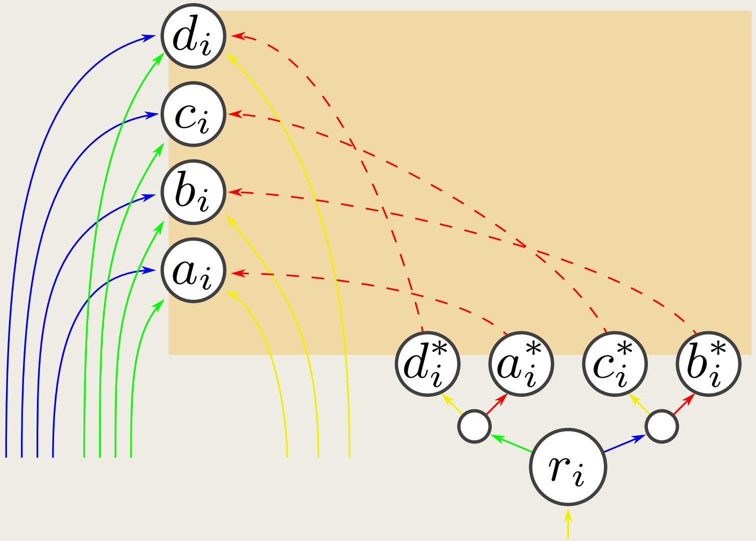
Order preserving trees to preserve order of ordered triple and order preserving gadgets

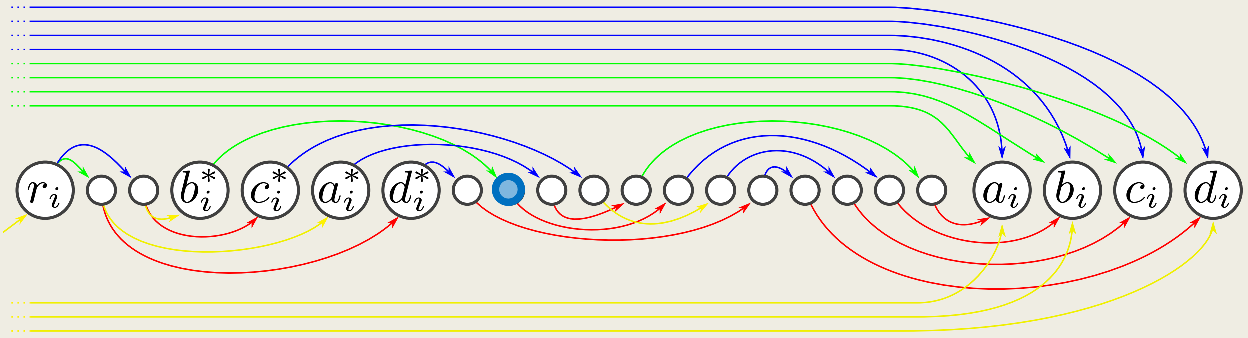
4 page Matching is NP-complete



Dashed red lines represent paths of alternating colors of length n

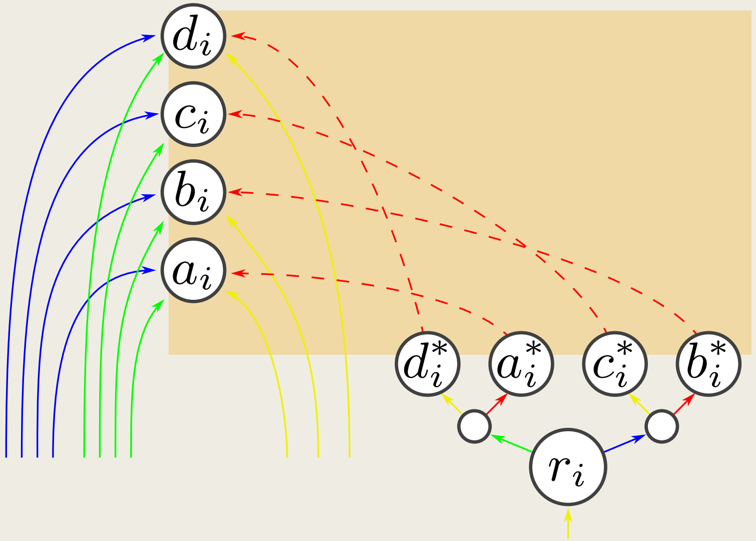
Embedding paths

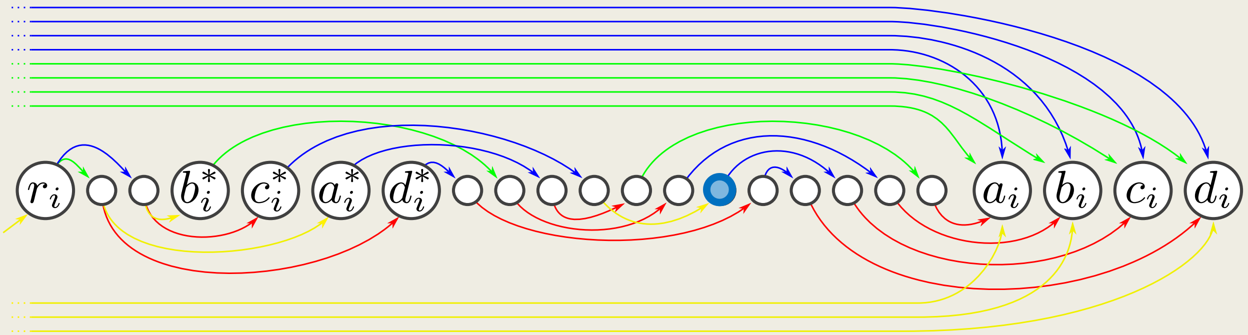




Vertices are inserted in insertion sort manner

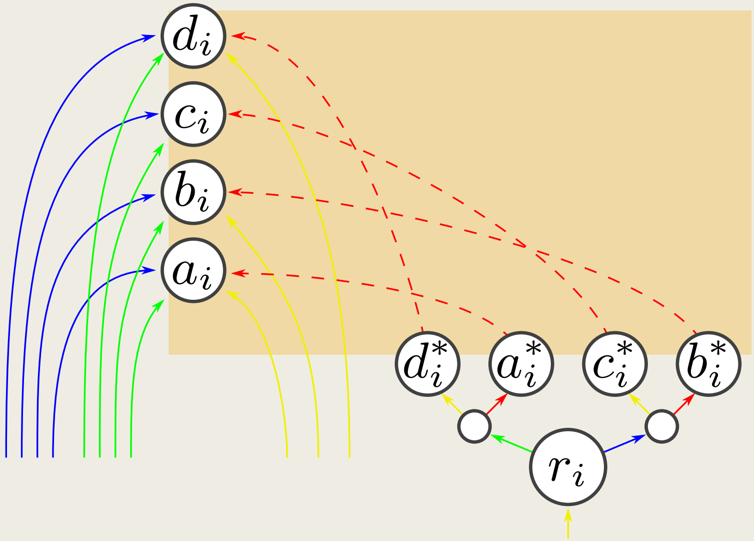
Embedding paths

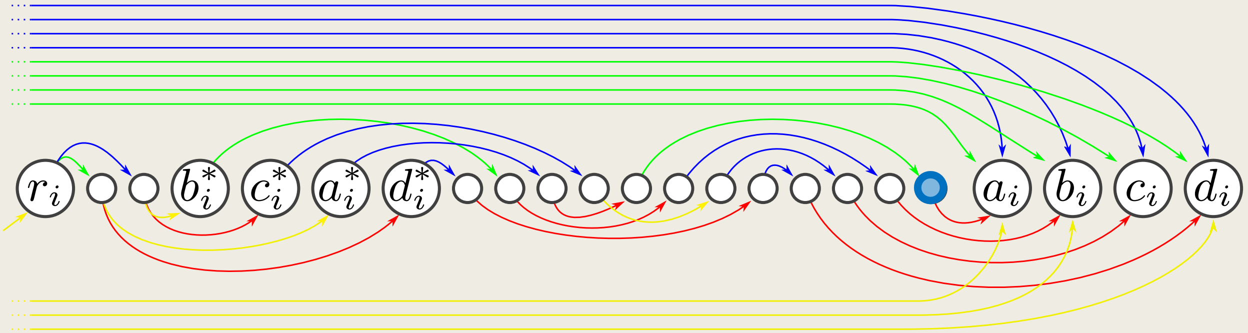




Vertices are inserted in insertion sort manner

Embedding paths

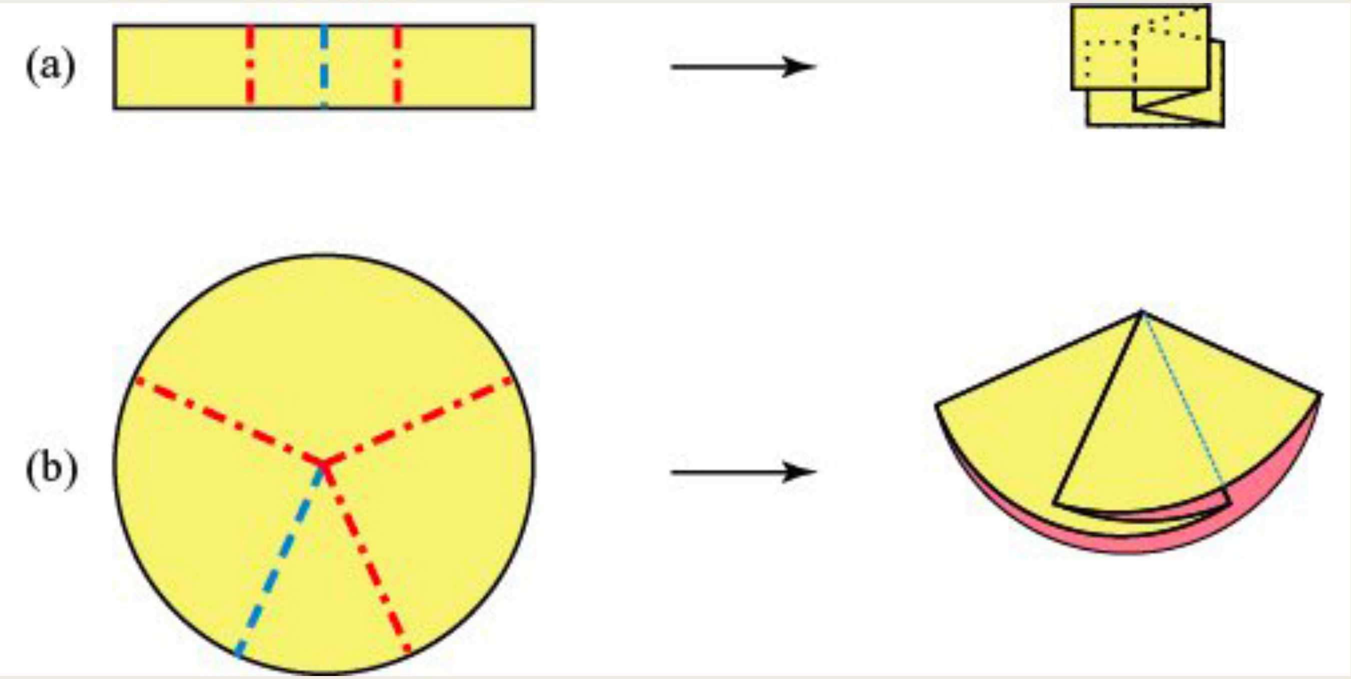




Vertices are inserted in insertion sort manner

Linear Time 2 page Matching

- Linear time algorithm for 2 page Matching by reducing to 1D origami or single vertex flat foldability

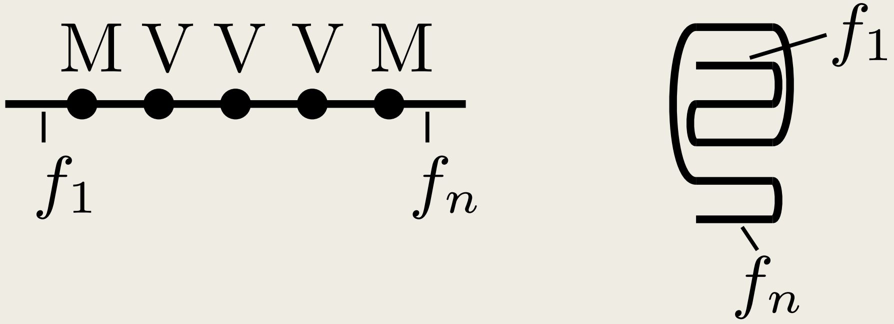


a) 1D Origami

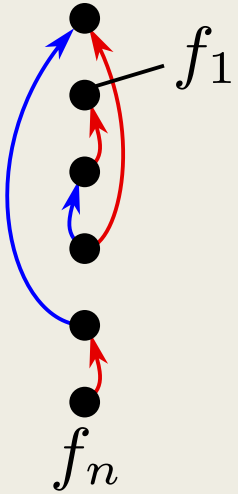
b) Single vertex foldability

Linear Time 2 page Matching

- Linear time algorithm for 2 page Matching by reducing to 1D origami or single vertex flat foldability



1D Origami Crease Pattern and Folding



DAG producing crease pattern