Constrained Outerstring Graphs

Jan Kratochvíl

Charles University, Prague, Czech Republic

joint work with

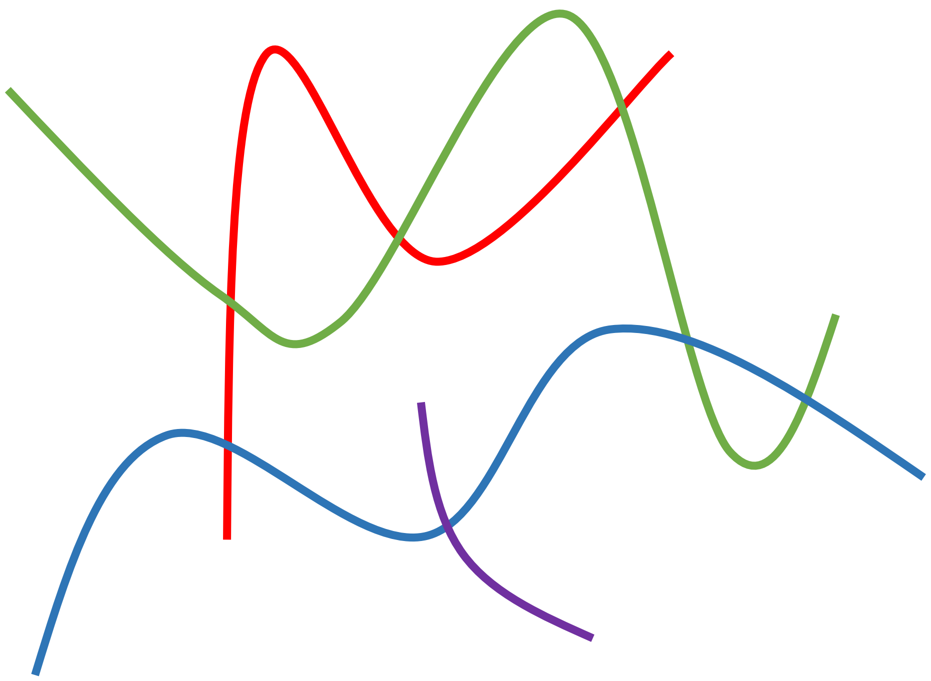
T. Biedl, S. Cornelsen, and I. Rutter

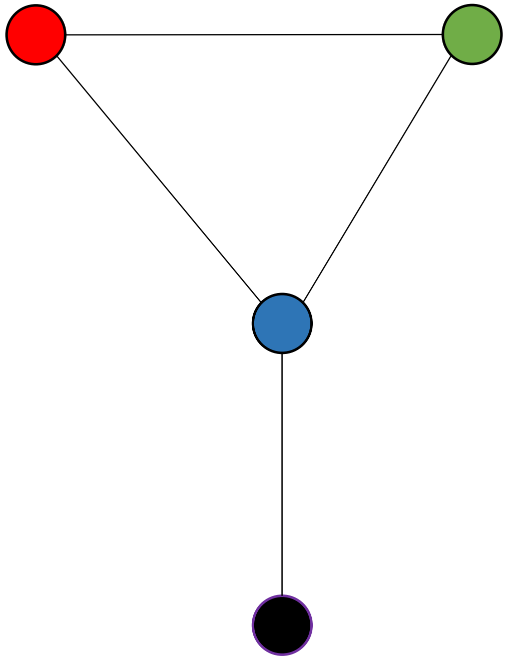
GD 2024

Wien, September 20, 2024

String graphs

- Intersection graphs of curves in the plane





String graphs

- Intersection graphs of curves in the plane
- 1966 Sinden: Topology of thin film RC-circuits, Bell System Journal

Topology of Thin Film RC Circuits

By F. W. SINDEN

(Manuscript received August 31, 1966)

Integrated RC circuits can be made by depositing exceedingly thin metallic and dielectric films in suitable patterns on an insulating substrate. Resistors are strips of conductor; capacitors are patches on which conducting, dielectric, and conducting layers are superimposed. Since conductors can cross at capacitor patches, RC networks need not be strictly planar to be realizable in thin film.

Determining which RC circuits are realizable poses new problems in topology which are remarkably simple to state but are as yet unsolved. The results reported here are fragmentary, but they do cover some cases of small order that may be of practical interest.

String graphs

- Intersection graphs of curves in the plane
- 1966 Sinden: Topology of thin film RC-circuits, Bell System Journal

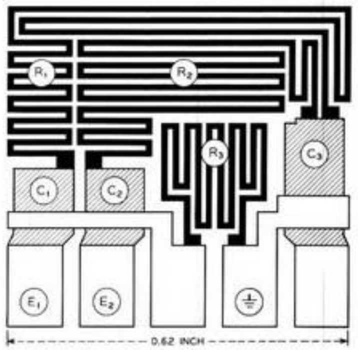


Fig. 1 — Thin film layout for a notch filter (courtesy W. H. Orr). Black region is bottom conductor; shaded region is dielectric; white region is top conductor.

Finding feasible layouts, or even determining when they exist, leads to unsolved problems in topology. The results presented here give answers only in special cases. Moreover, these results concern only the topological side of the problem; *electrical* equivalences are not taken into account. It is assumed that the network is given topologically and that

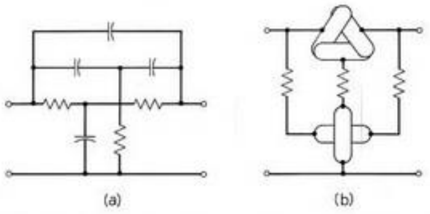


Fig. 2 — (a) Nonplanar circuit ("twin-tee", Ref. 3, p. 309); (b) schematic thin film layout for the circuit in (a).

terminals to the outside are located in given fixed positions on the periphery of the board.

II. SEPARATION OF THE RESISTIVE AND CAPACITIVE PARTS

Given an RC network N, let R_N be the purely resistive network obtained by replacing every capacitor by a direct connection. Clearly N is not realizable in thin film unless R_N is. R_N is realizable only if its graph (a vertex for each conductor, an edge for each resistor) is planar under the restrictions imposed by the locations of the terminals to the outside (see Fig. 3). This observation provides a first check: if R_N is not planar, there is no need to proceed further.

Each vertex in the graph of R_N replaces a purely capacitive network. In Fig. 3, for example, the vertex V in R_N replaces the network shown in Fig. 4.

One way to construct a realization of N is to construct realizations for the individual vertex-networks, and then to fit these into the planar layout of R_N. Since the layout of R_N may not be unique (there may be more than one ordering of edges about a vertex) the conditions on the vertex-networks may not be unique.

Another approach, discussed briefly in the final section, is to modify algorithms for purely capacitive networks to take account of resistors. In either case, one needs to study the purely capacitive networks first.

III. PURE C NETWORKS

A pure C network is a set of zero-resistance conductors $c_1, \cdots, c_r$ some pairs of which are connected by capacitors. The problem of finding a feasible layout for such a network is the following:

For each conductor c_i find a connected region R_i in the plane such that

(i) R_i and R_j have common points if and only if c_i and c_j are connected by a capacitor, and

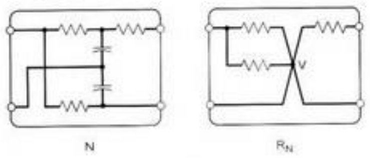
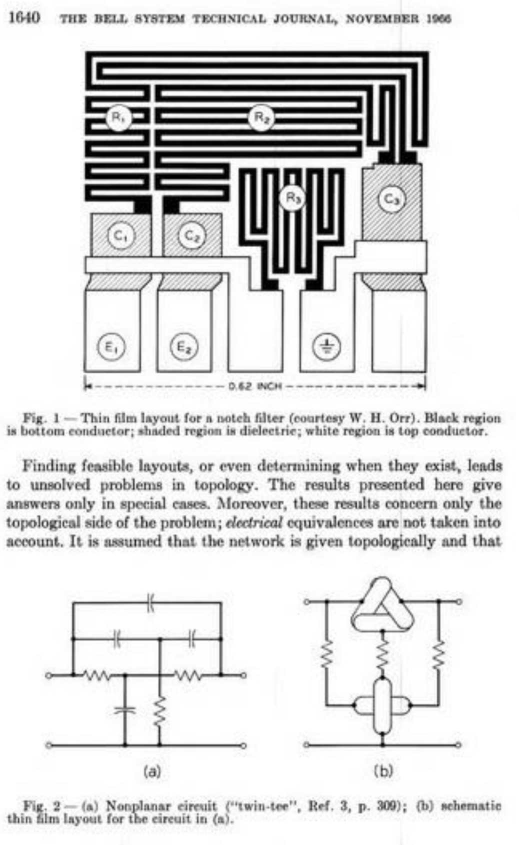


Fig. 3 — Nonplanar RC network N and reduced purely resistive network R_N.

String graphs

- Intersection graphs of curves in the plane
- 1966 Sinden: Topology of thin film RC-circuits, Bell System Journal

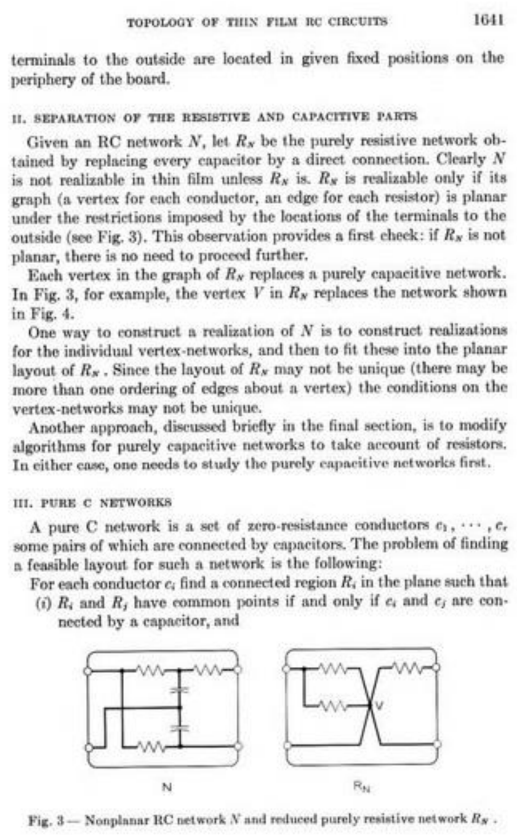


1640 THE BELL SYSTEM TECHNICAL JOURNAL, NOVEMBER 1966

Fig. 1 — Thin film layout for a notch filter (courtesy W. H. Orr). Black region is bottom conductor; shaded region is dielectric; white region is top conductor.

Finding feasible layouts, or even determining when they exist, leads to unsolved problems in topology. The results presented here give answers only in special cases. Moreover, these results concern only the topological side of the problem; *electrical* equivalences are not taken into account. It is assumed that the network is given topologically and that

Fig. 2 — (a) Nonplanar circuit ("twin-tee", Ref. 3, p. 309); (b) schematic thin film layout for the circuit in (a).



TOPOLOGY OF THIN FILM RC CIRCUITS 1641

terminals to the outside are located in given fixed positions on the periphery of the board.

II. SEPARATION OF THE RESISTIVE AND CAPACITIVE PARTS

Given an RC network N, let R_N be the purely resistive network obtained by replacing every capacitor by a direct connection. Clearly N is not realizable in thin film unless R_N is. R_N is realizable only if its graph (a vertex for each conductor, an edge for each resistor) is planar under the restrictions imposed by the locations of the terminals to the outside (see Fig. 3). This observation provides a first check: if R_N is not planar, there is no need to proceed further.

Each vertex in the graph of R_N replaces a purely capacitive network. In Fig. 3, for example, the vertex V in R_N replaces the network shown in Fig. 4.

One way to construct a realization of N is to construct realizations for the individual vertex-networks, and then to fit these into the planar layout of R_N. Since the layout of R_N may not be unique (there may be more than one ordering of edges about a vertex) the conditions on the vertex-networks may not be unique.

Another approach, discussed briefly in the final section, is to modify algorithms for purely capacitive networks to take account of resistors. In either case, one needs to study the purely capacitive networks first.

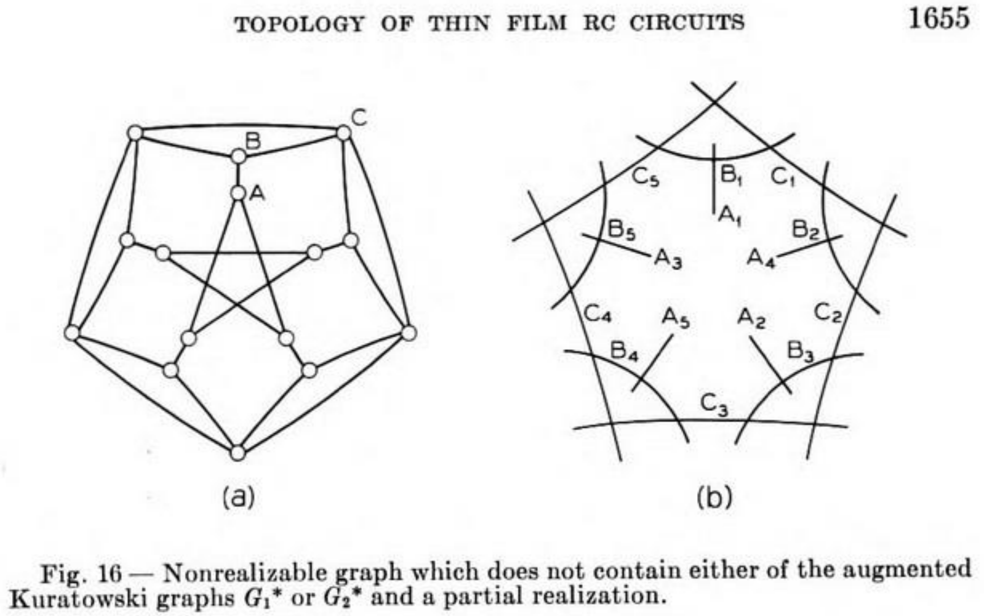
III. PURE C NETWORKS

A pure C network is a set of zero-resistance conductors $c_1, \cdots, c_r$ some pairs of which are connected by capacitors. The problem of finding a feasible layout for such a network is the following:

For each conductor c_i find a connected region R_i in the plane such that

(i) R_i and R_j have common points if and only if c_i and c_j are connected by a capacitor, and

Fig. 3 — Nonplanar RC network N and reduced purely resistive network R_N.



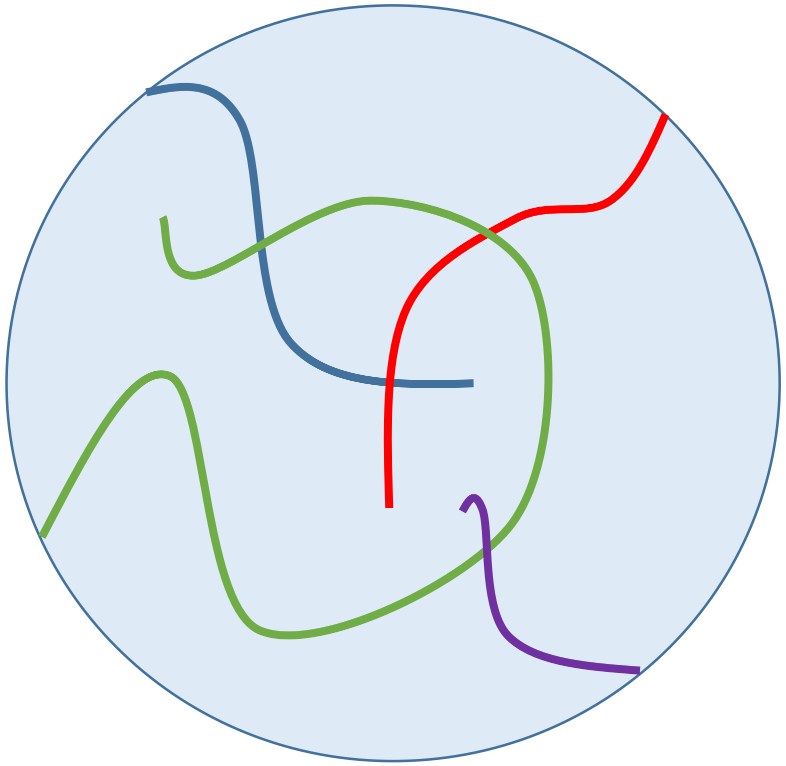
TOPOLOGY OF THIN FILM RC CIRCUITS 1655

Fig. 16 — Nonrealizable graph which does not contain either of the augmented Kuratowski graphs G_1* or G_2* and a partial realization.

String graphs

- Intersection graphs of curves in the plane
- 1966 Sinden: Topology of thin film RC-circuits, Bell System Journal
- 1974 Graham: Open problem in Keztheley
- 1976 Ehrlich, Even, Tarjan: Coloring intersection graphs of segments is NP-hard
- 1991 JK: Recognizing string graphs is NP-hard
- 1991 JK, Matoušek: There are string graphs that require exponential number of crossing points in every representation
- 2001 Schaefer, Stefankovic: Upper bound on the number of crossing points – recognizing string graphs is decidable
- 2002 Schaefer, Sedgwick, Stefankovic: Recognizing string graphs in NP

Outer-string graphs



Outer-string graphs

Implicitly introduced in Sinden (1966)

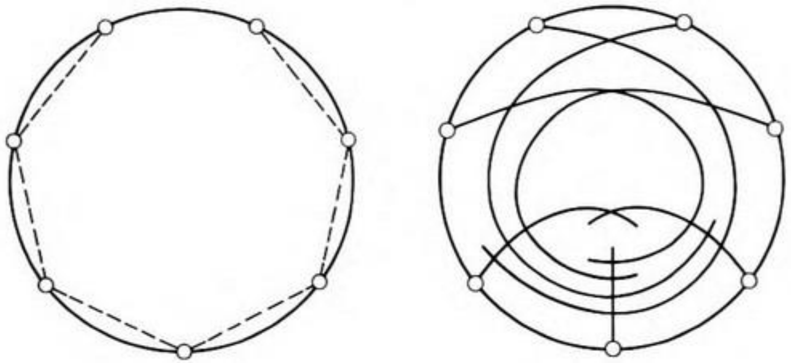
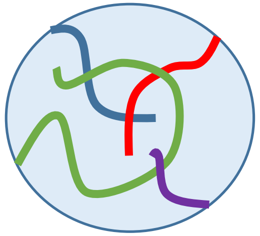


Fig. 13 — Realization of the empty chain of order seven.

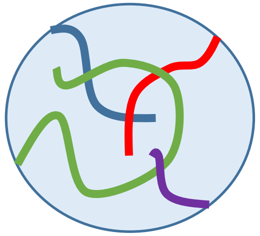
Outer-string graphs



Implicitly introduced in Sinden (1966)

Biedl, Biniaz, Derka (2018): May require exponential number of crossing points in every representation

Outer-string graphs



Implicitly introduced in Sinden (1966)

Biedl, Biniaz, Derka (2018): May require exponential number of crossing points in every representation

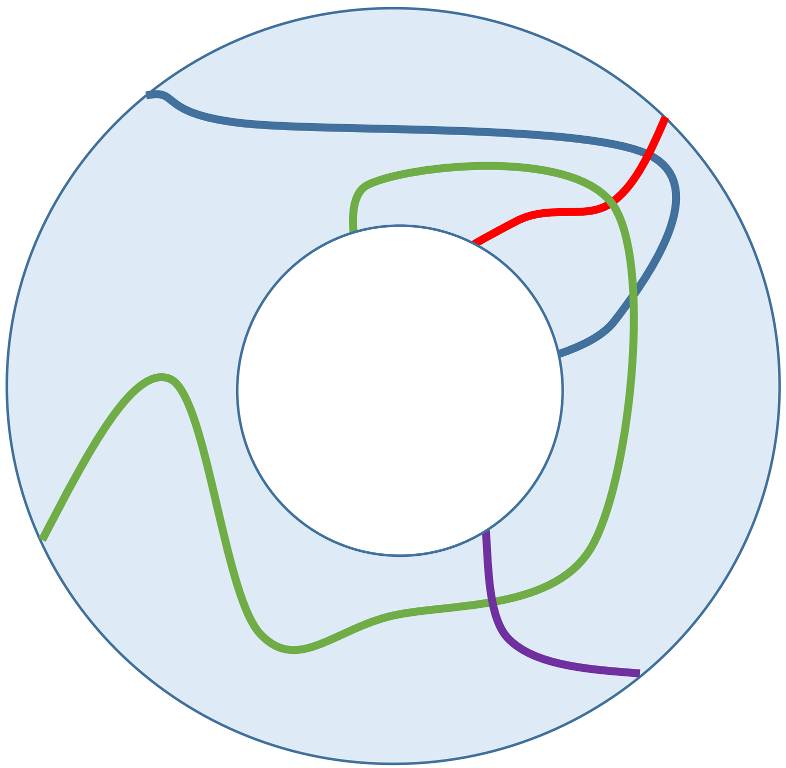
NP-hard to recognize

Recognition of outer-string graphs

1993 Middendorf, Pfeiffer: Recognition of cylinder graphs is NP-hard

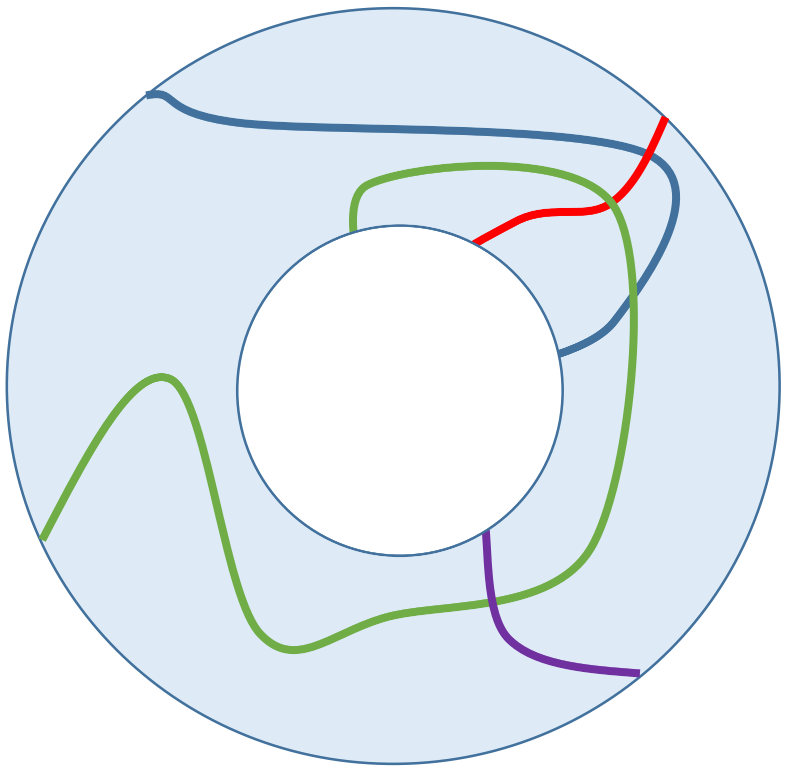
Reduction from Hasse diagram recognition

$H \to G$ so that H is Hasse iff G is cylinder graph



Recognition of outer-string graphs

1993 Middendorf, Pfeiffer: Recognition of cylinder graphs is NP-hard



Reduction from Hasse diagram recognition

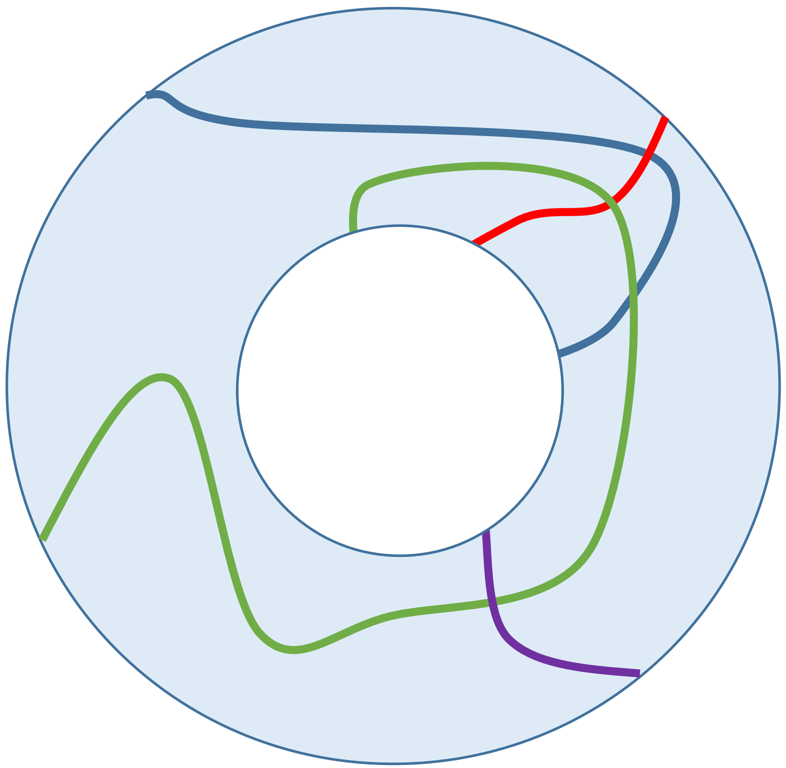
$H \rightarrow G$ so that H is Hasse iff G is cylinder graph

In fact: H Hasse implies G is cylinder, and

H not Hasse implies G not outer-string

Recognition of outer-string graphs

1993 Middendorf, Pfeiffer: Recognition of cylinder graphs is NP-hard



Reduction from Hasse diagram recognition

$H \rightarrow G$ so that H is Hasse iff G is cylinder graph

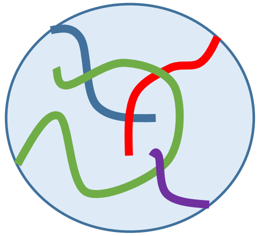
In fact: H Hasse implies G is cylinder, and

H not Hasse implies G not outer-string

Graphclasses.org: Complexity of recognition of outer-string graphs is unknown

Rok, Walczak (2019)

Outer-string graphs



Implicitly introduced in Sinden (1966)

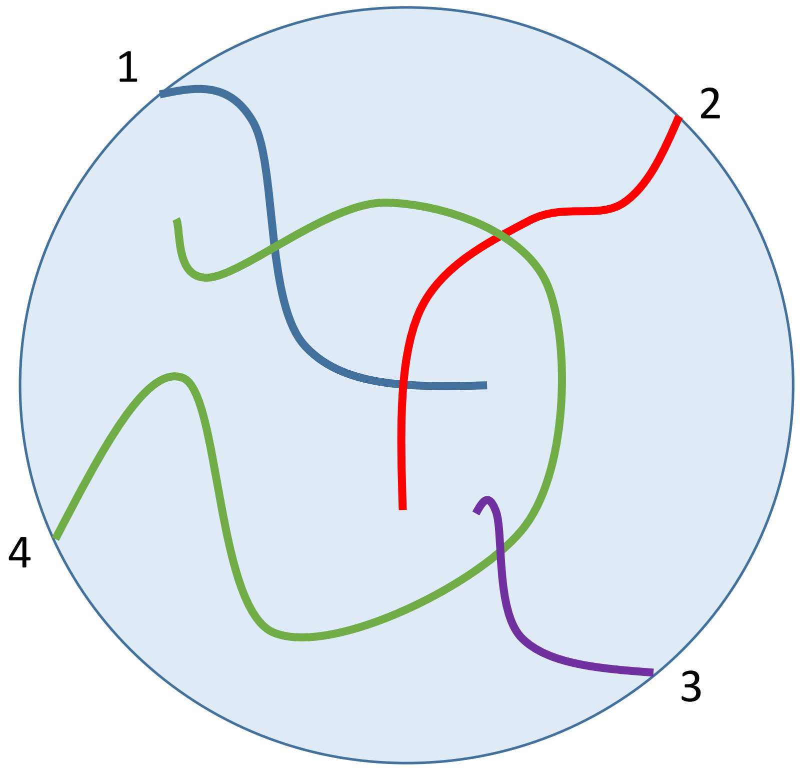
Biedl, Biniaz, Derka (2018): May require exponential number of crossing points in every representation

NP-hard to recognize

Rok, Walczak (2019): Outer-string graphs are χ-bounded

Constrained outer-string graphs

The cyclic order in which the strings are tied to the boundary is given as part of input.



Constrained outer-string graphs

Theorem 1: A necessary condition for a constrained graph G to be realizable is that $\bar{G}$ contain no empty cycles of order four or more.

Proof: (*i*) If G is an empty cycle of order four, then G is not realizable. This is easily verified by inspection. If, therefore, $\bar{G}$ *contains* an empty cycle of order four, then G is not realizable.

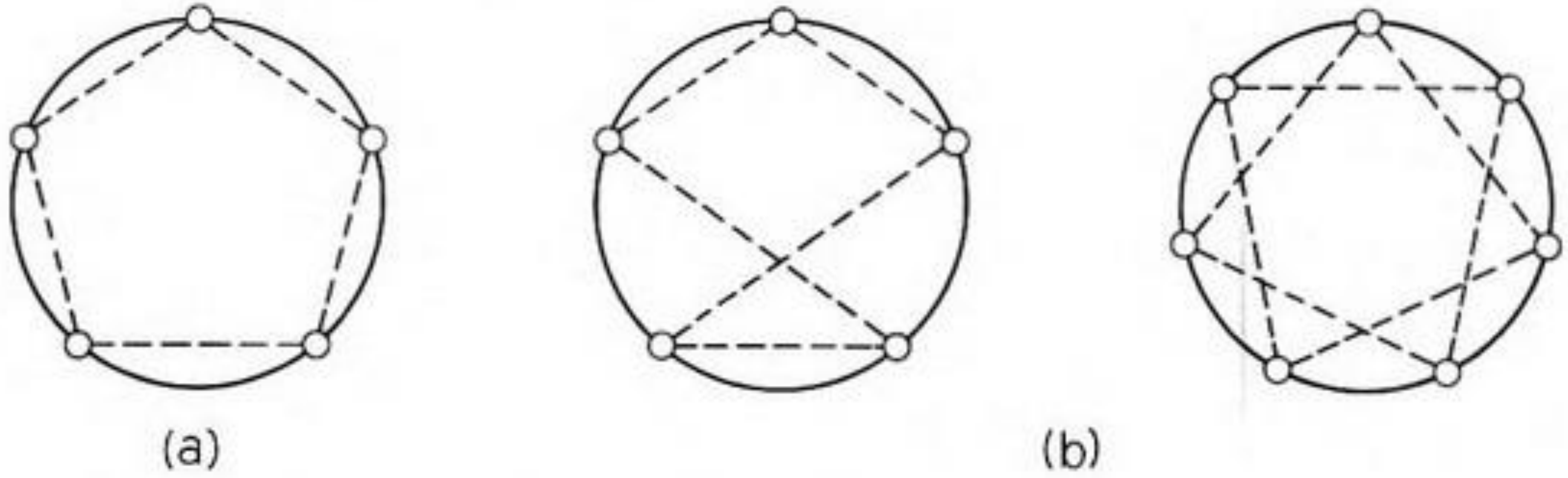


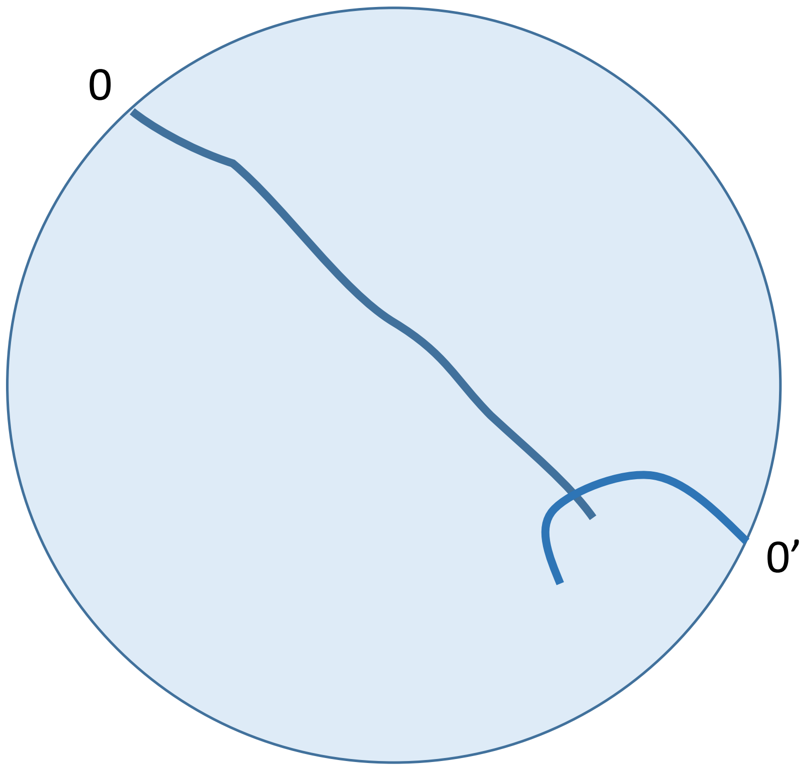
Fig. 9 — (a) Empty cycle in $\bar{G}$, (b) non-cycles. Dashed edges belong to $\bar{G}$; edges not shown belong to G.

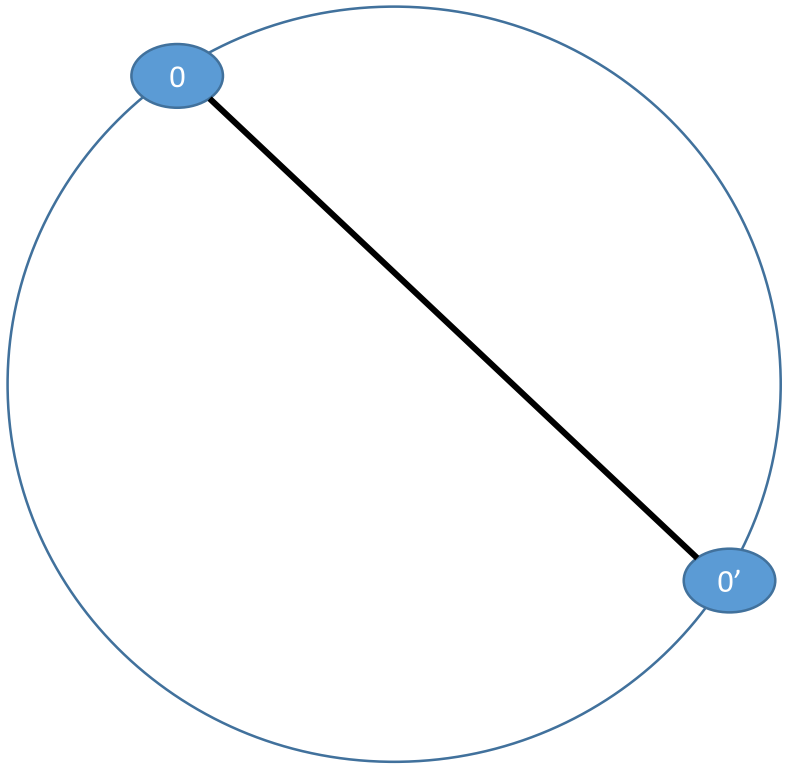
Constrained outer-string graphs

May require exponential number of crossings

Constrained outer-string graphs

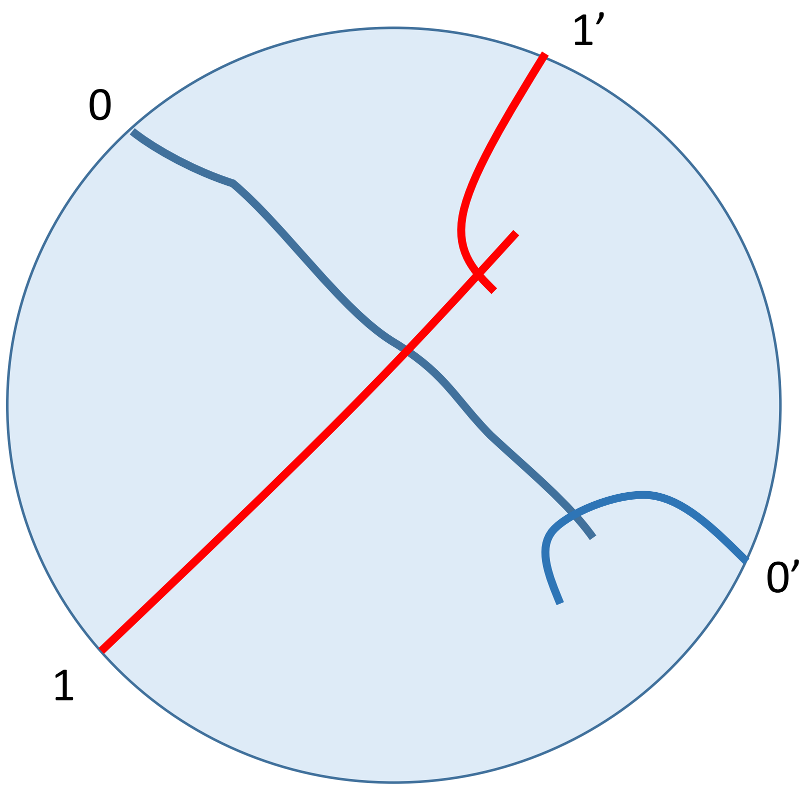
May require exponential number of crossings

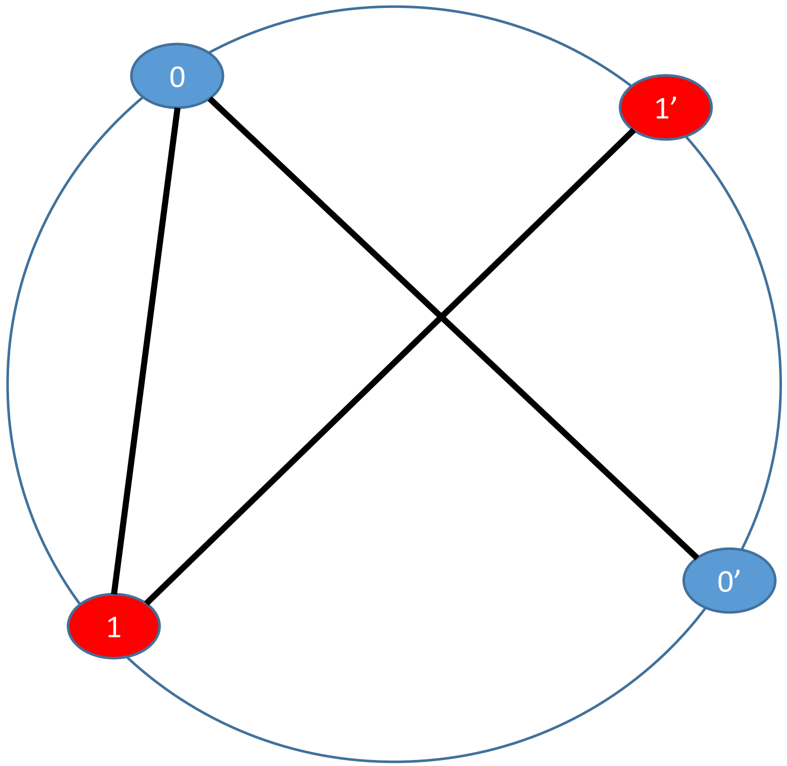




Constrained outer-string graphs

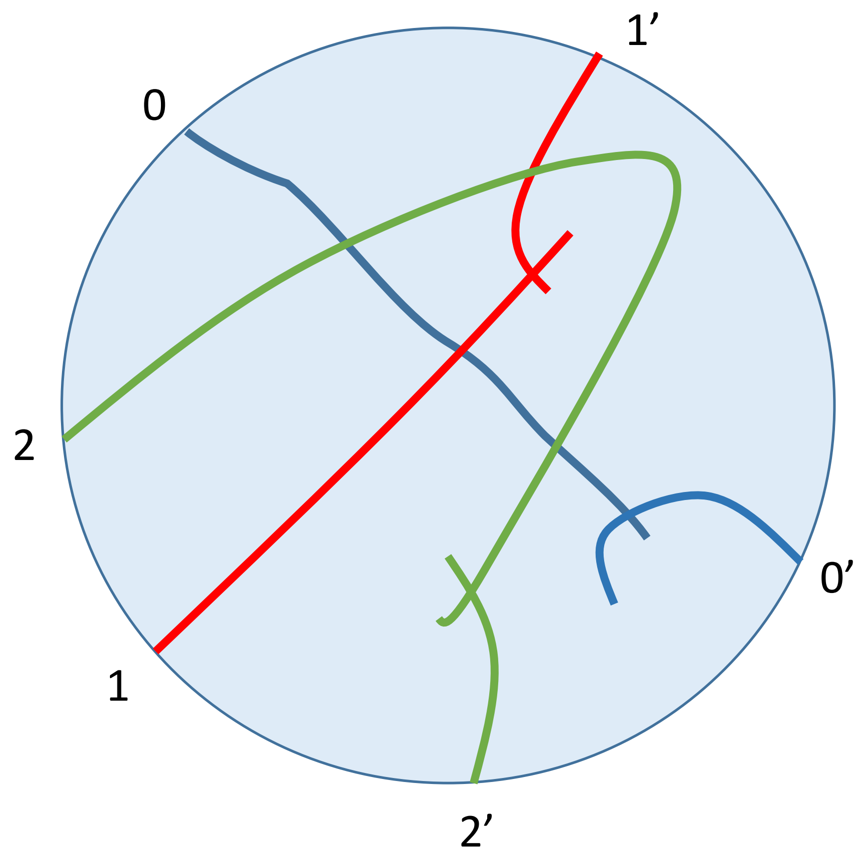
May require exponential number of crossings

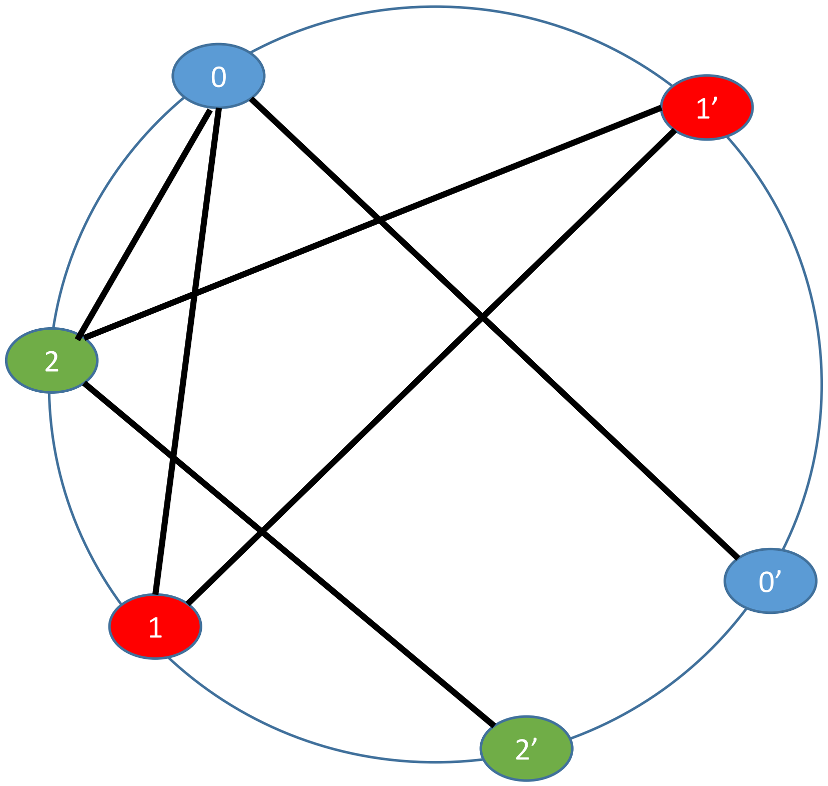




Constrained outer-string graphs

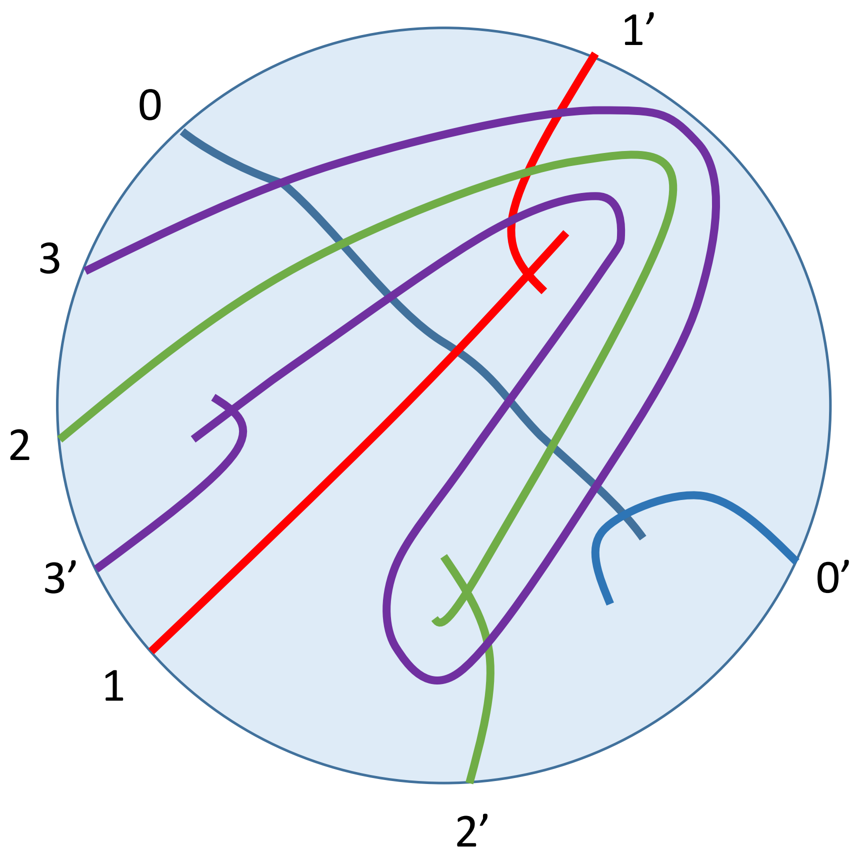
May require exponential number of crossings

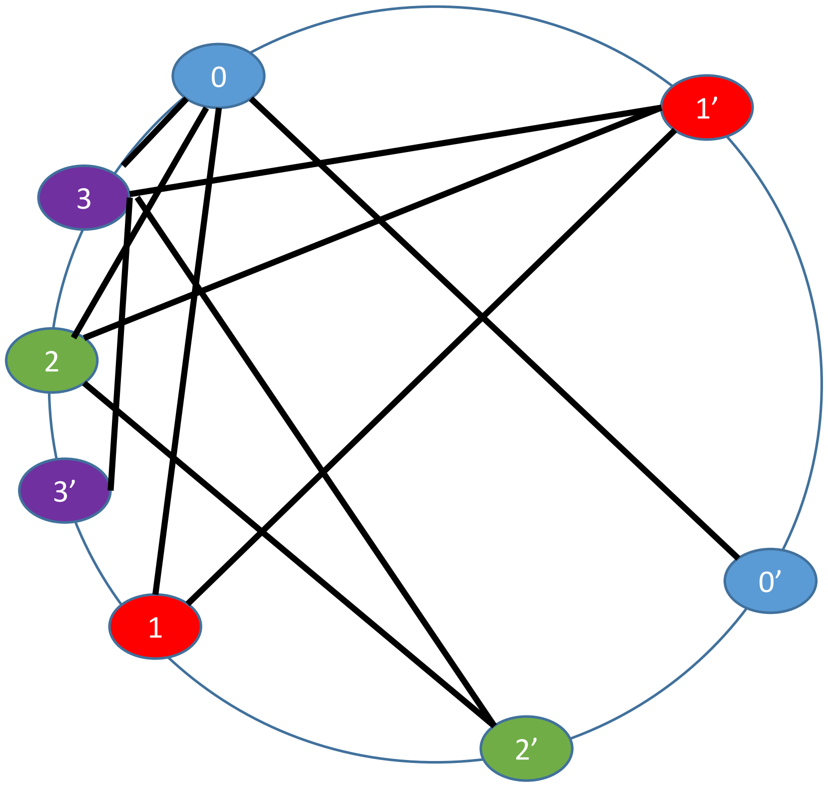




Constrained outer-string graphs

May require exponential number of crossings





Constrained outer-string graphs

May require exponential number of crossings

Recognition is open

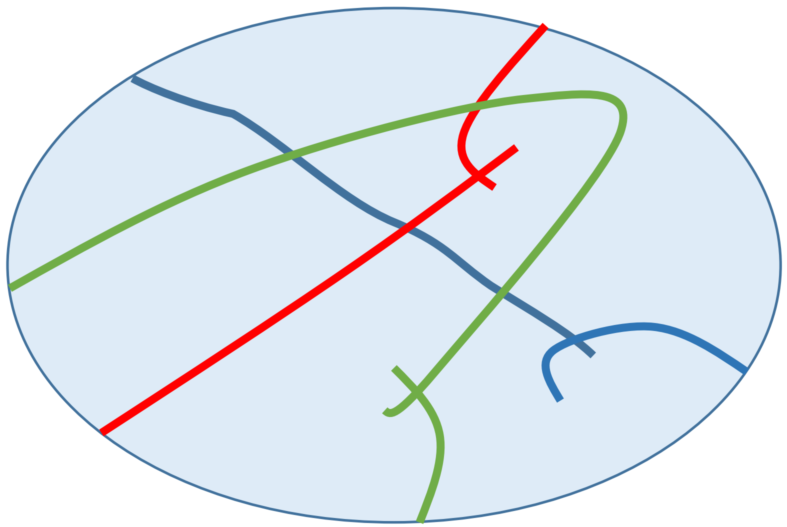
Constrained outer-string graphs

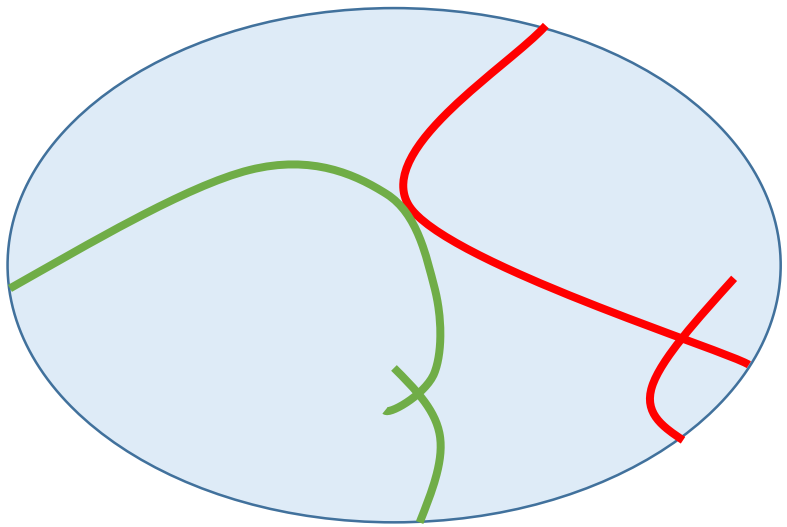
May require exponential number of crossings

Recognition is open

Constrained outer-1-string graphs

Every two curves cross in at most 1 point (touching is not allowed)





Constrained outer-string graphs

May require exponential number of crossings

Recognition is open

Constrained outer-1-string graphs

Every two curves cross in at most 1 point (touching is not allowed)

Recognition is open

Outer-1-string graphs

Recognition is open

Our results

Thm 1: A graph has a constrained outer-string representation for **every** cyclic ordering of its vertices iff it is co-chordal.

Our results

Thm 1: A graph has a constrained outer-string representation for **every** cyclic ordering of its vertices iff it is co-chordal.

Thm 2: Constrained outer-string **chordal graphs** can be recognized in polynomial time, and are described by a single forbidden configuration.

Our results

Thm 1: A graph has a constrained outer-string representation for **every** cyclic ordering of its vertices iff it is co-chordal.

Thm 2: Constrained outer-string **chordal graphs** can be recognized in polynomial time, and are described by a single forbidden configuration.

Thm 3: Constrained outer-1-string **trees** can be recognized in polynomial time, and are described by (infinitely many) forbidden configurations.

Proof of Thm 1

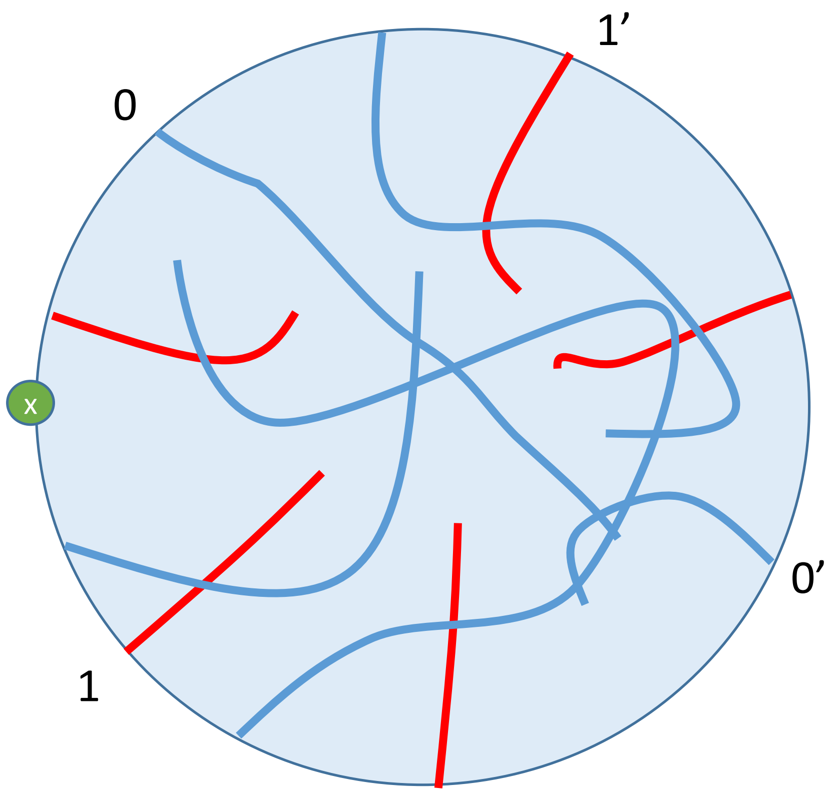
Thm 1: A graph has a constrained outer-string representation for **every** cyclic ordering of its vertices iff it is co-chordal.

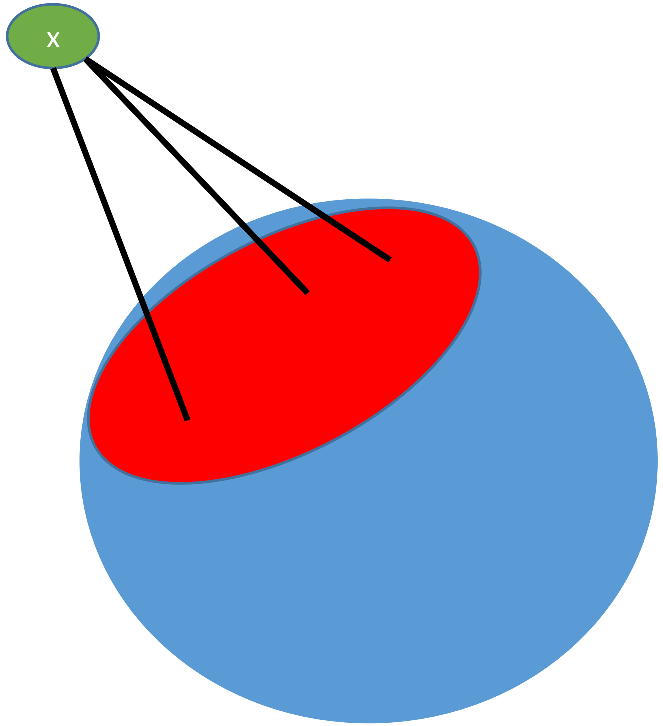
Proof: If G is not co-chordal, then $-G$ contains a chordless cycle C_k with $k \geq 4$. No cyclic ordering of the vertices of G that extends the natural ordering of this C_k allows a constrained outer-string representation (Sinden).

If G is co-chordal, and given a cyclic ordering of its vertices, construct a constrained representation by induction on adding simplicial vertices of $-G$.

Proof of Thm 1

Thm 1: A graph has a constrained outer-string representation for **every** cyclic ordering of its vertices iff it is co-chordal.

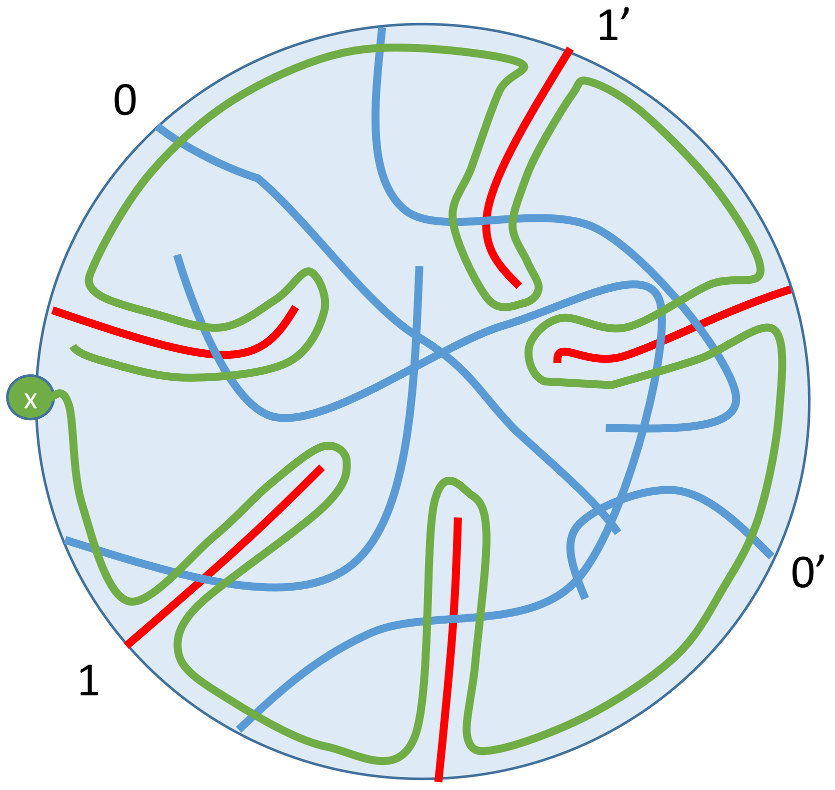


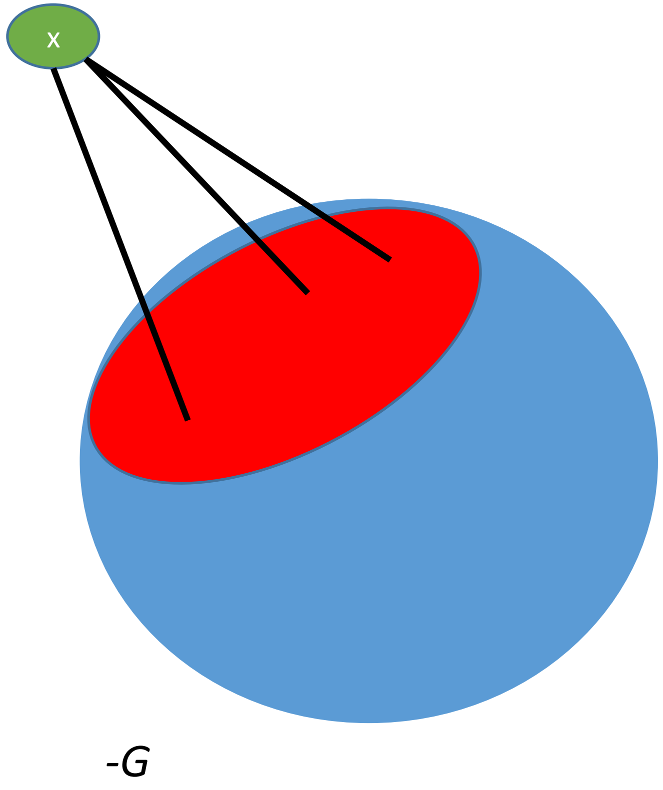


-G

Proof of Thm 1

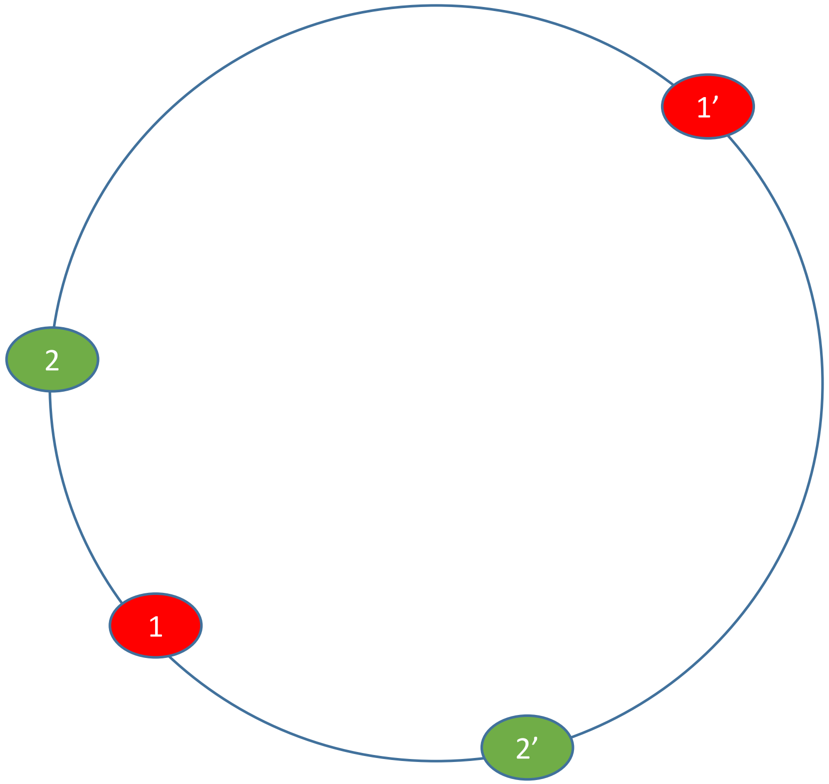
Thm 1: A graph has a constrained outer-string representation for **every** cyclic ordering of its vertices iff it is co-chordal.

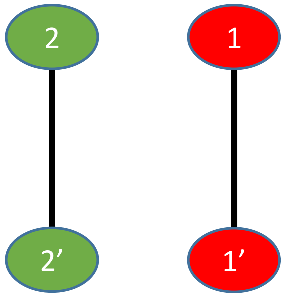




Proof of Thm 2

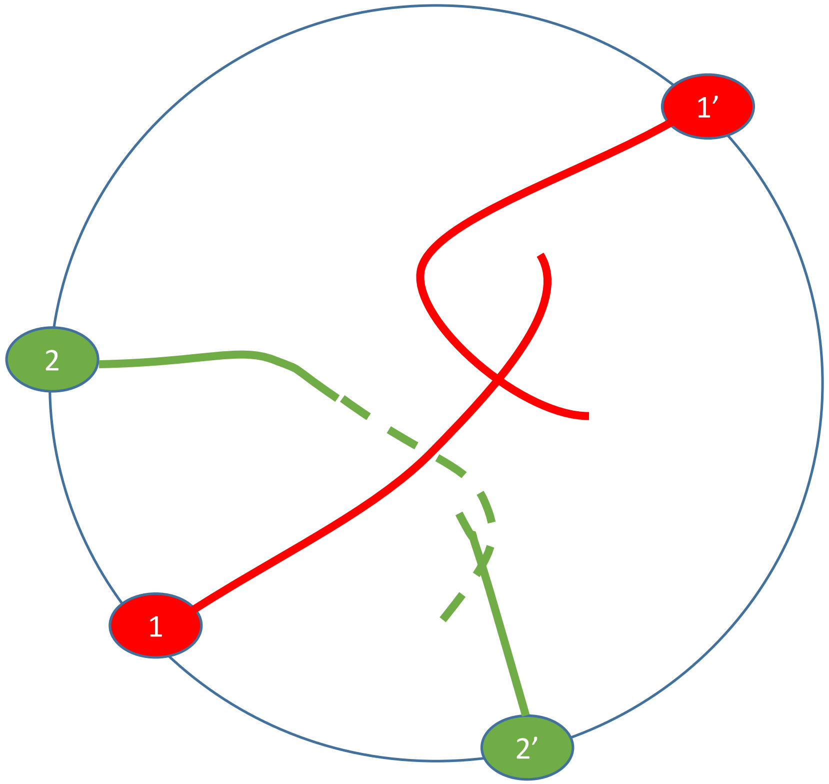
Thm 2: Constrained outer-string **chordal graphs** can be recognized in polynomial time, and are described by a single forbidden configuration – an interleaving pair of independent edges.

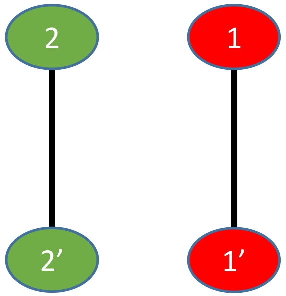




Proof of Thm 2

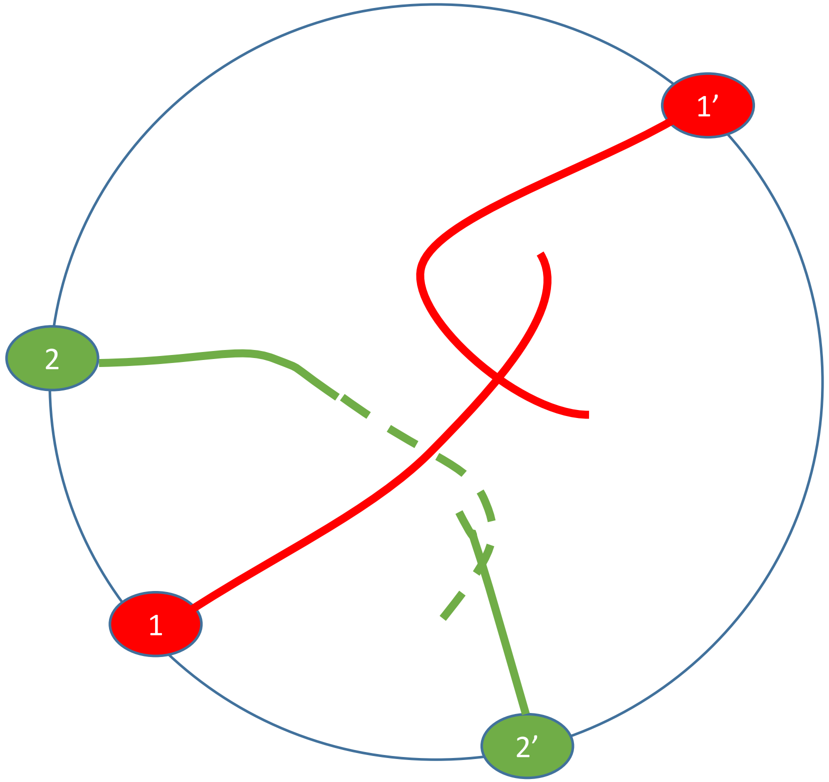
Thm 2: Constrained outer-string **chordal graphs** can be recognized in polynomial time, and are described by a single forbidden configuration – an interleaving pair of independent edges.

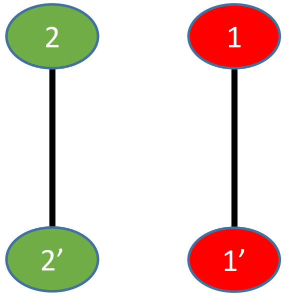


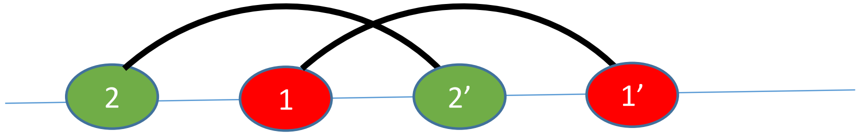


Proof of Thm 2

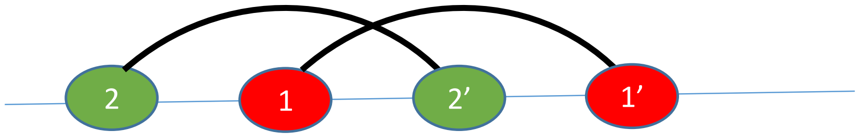
Thm 2: Constrained outer-string **chordal graphs** can be recognized in polynomial time, and are described by a single forbidden configuration – an interleaving pair of independent edges.







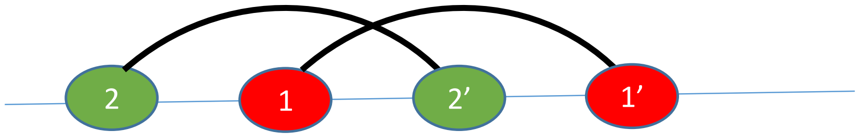
Proof of Thm 2



Thm 2’: A **chordal graph** equipped with an ordering without an interleaving pair of independent edges admits a constrained outer-string representation.

Proof: By induction on the number of pairs of independent edges.

Proof of Thm 2

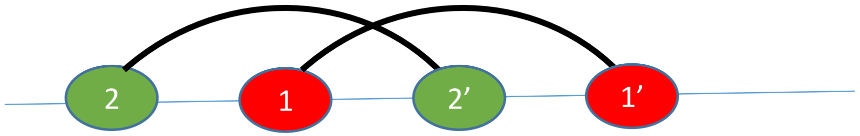


Thm 2’: A **chordal graph** equipped with an ordering without an interleaving pair of independent edges admits a constrained outer-string representation.

Proof: By induction on the number of pairs of independent edges.

1) No two independent edges. A $2K_2$-free chordal graph is a split graph.

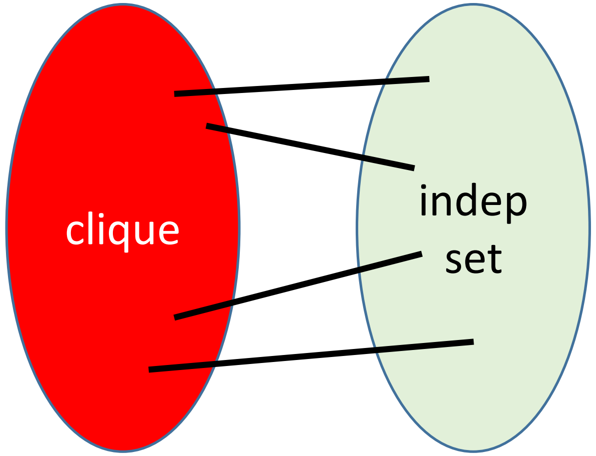
Proof of Thm 2



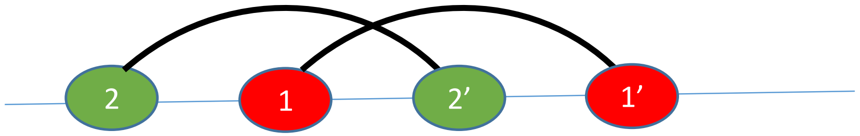
Thm 2’: A **chordal graph** equipped with an ordering without an interleaving pair of independent edges admits a constrained outer-string representation.

Proof: By induction on the number of pairs of independent edges.

1) No two independent edges. A $2K_2$-free chordal graph is a split graph.



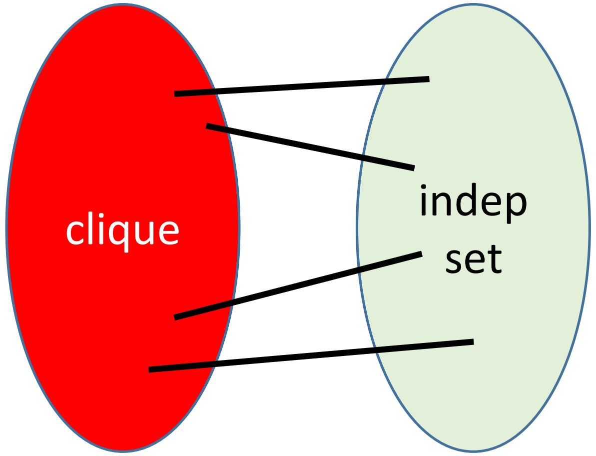
Proof of Thm 2

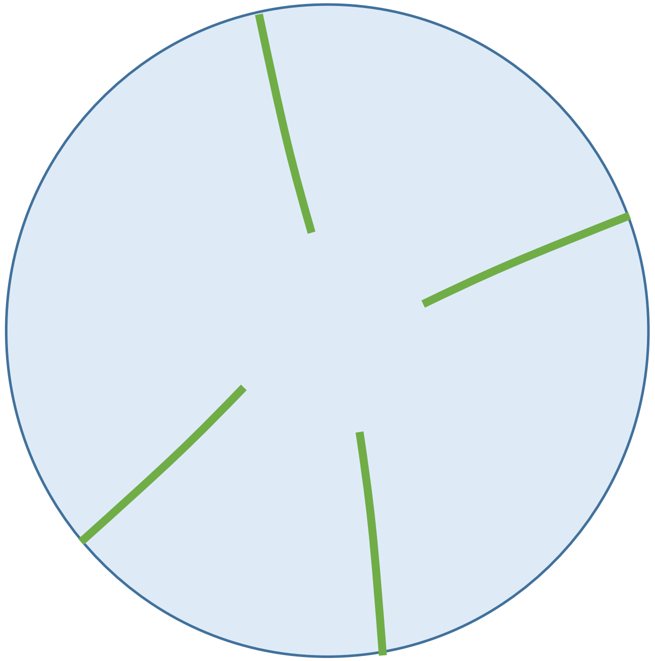


Thm 2': A **chordal graph** equipped with an ordering without an interleaving pair of independent edges admits a constrained outer-string representation.

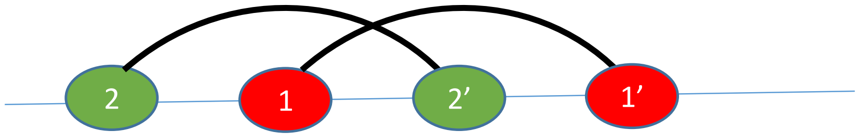
Proof: By induction on the number of pairs of independent edges.

1) No two independent edges. A $2K_2$-free chordal graph is a split graph.





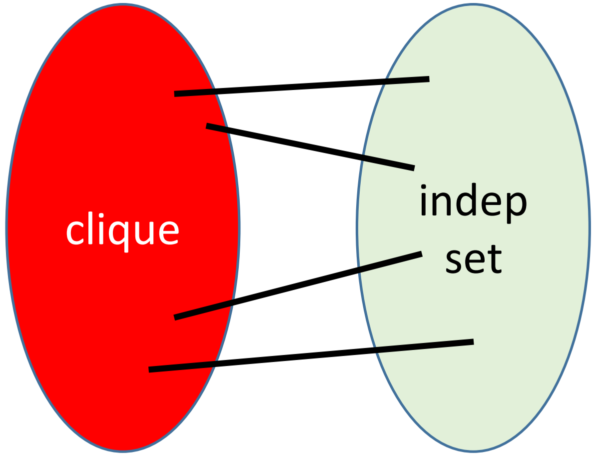
Proof of Thm 2

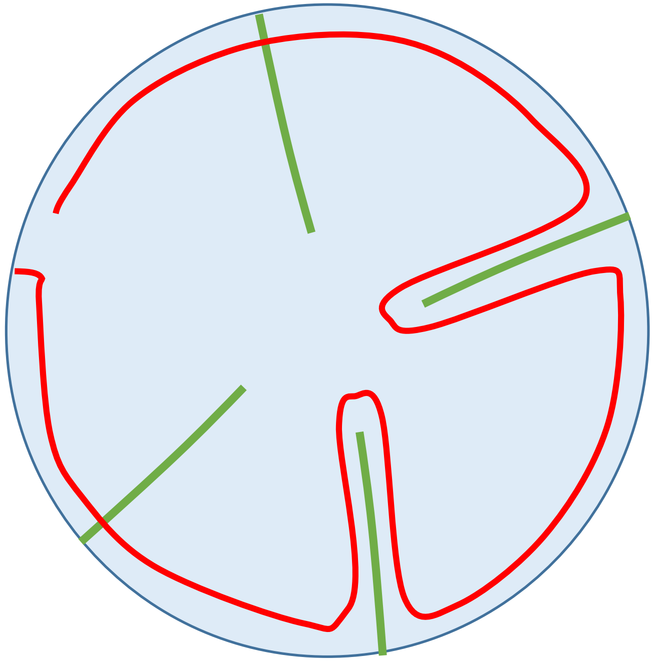


Thm 2': A **chordal graph** equipped with an ordering without an interleaving pair of independent edges admits a constrained outer-string representation.

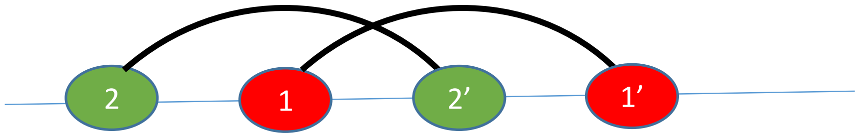
Proof: By induction on the number of pairs of independent edges.

1) No two independent edges. A $2K_2$-free chordal graph is a split graph.





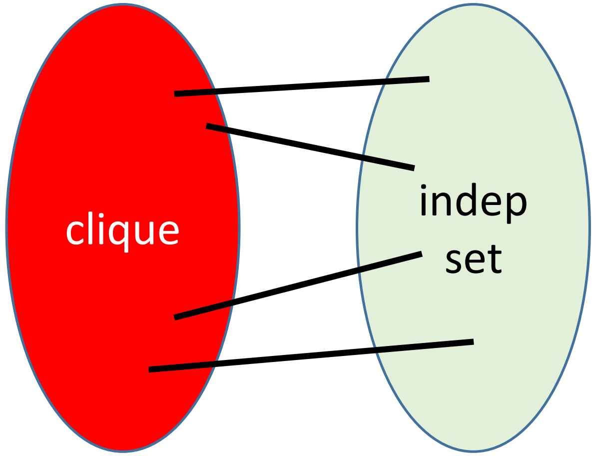
Proof of Thm 2

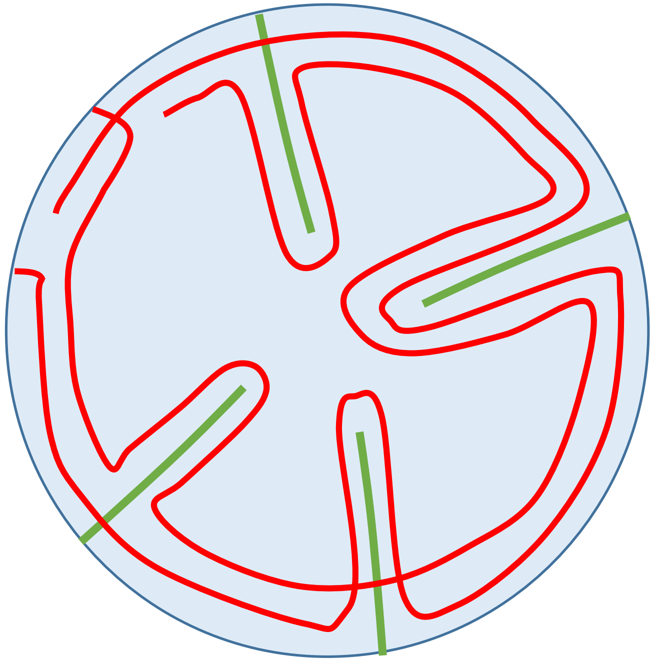


Thm 2’: A **chordal graph** equipped with an ordering without an interleaving pair of independent edges admits a constrained outer-string representation.

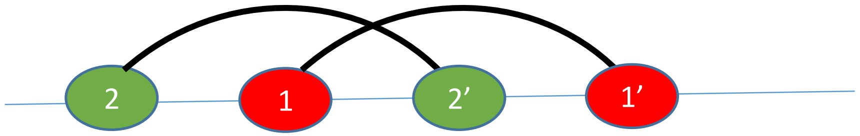
Proof: By induction on the number of pairs of independent edges.

1) No two independent edges. A $2K_2$-free chordal graph is a split graph.

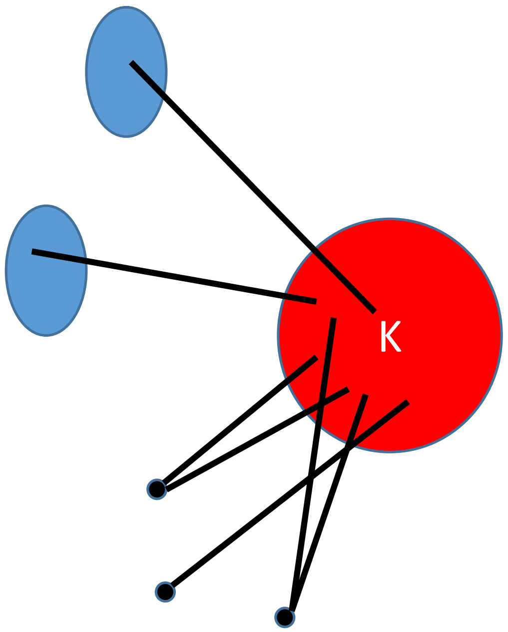




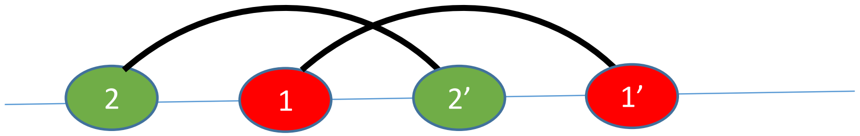
Proof of Thm 2



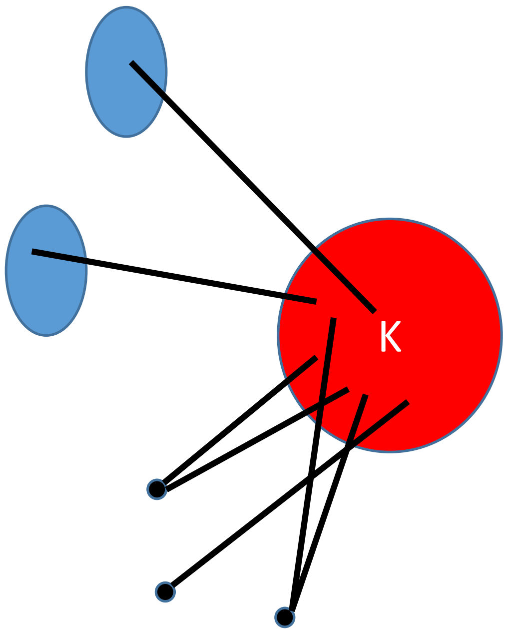
2) The graph *G* has at least one pair of independent edges. Consider a minimal set *K* of vertices such that *G*-*K* has at least 2 nontrivial components (i.e., components with at least 2 vertices).



Proof of Thm 2

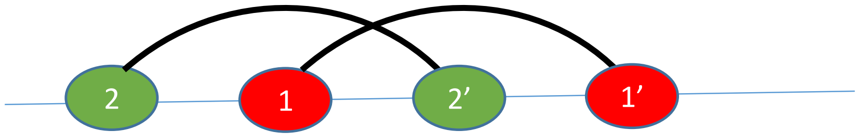


2) The graph G has at least one pair of independent edges. Consider a minimal set K of vertices such that G-K has at least 2 nontrivial components (i.e., components with at least 2 vertices).

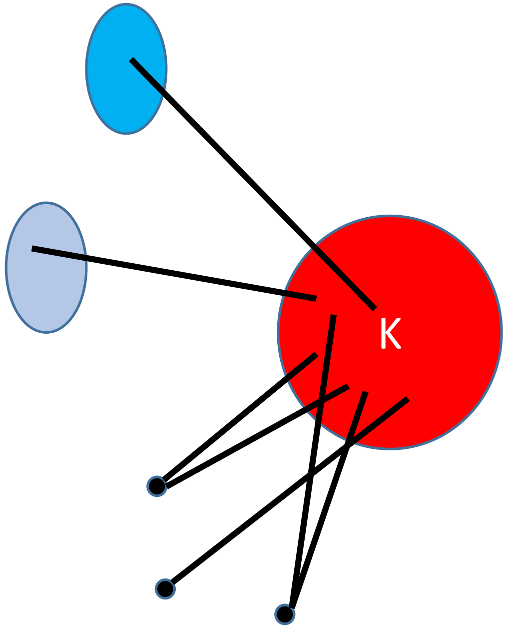


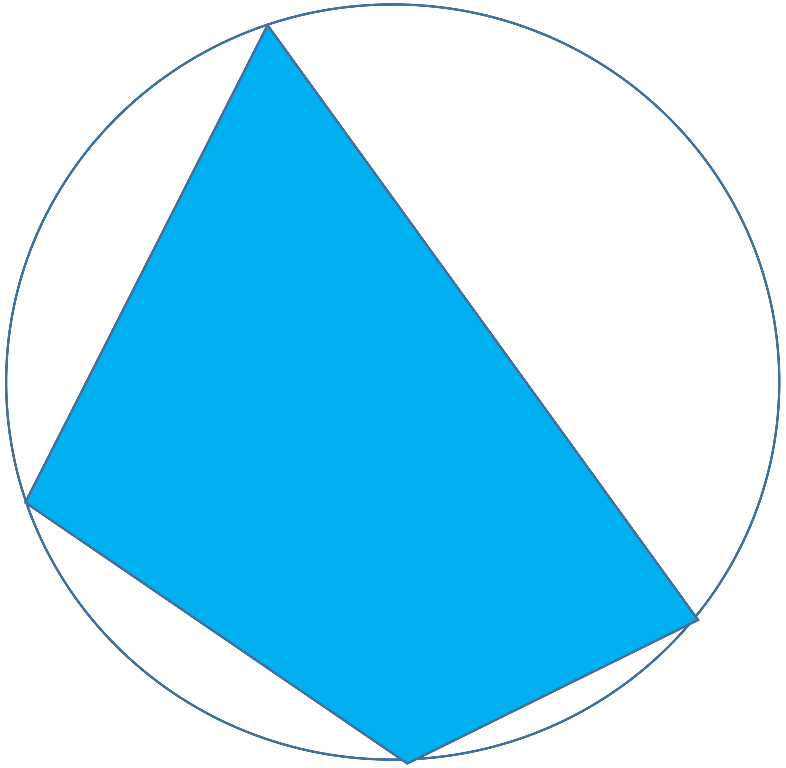
Claim 1. K is a clique, every vertex of K is adjacent to some vertex in every nontrivial component.

Proof of Thm 2

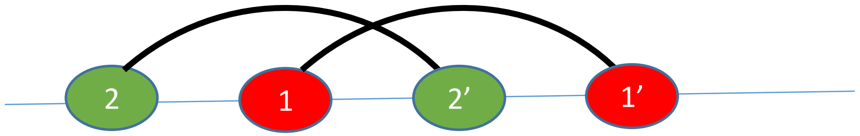


2) Claim 1. *K* is a clique, every vertex of *K* is adjacent to some vertex in every nontrivial component.

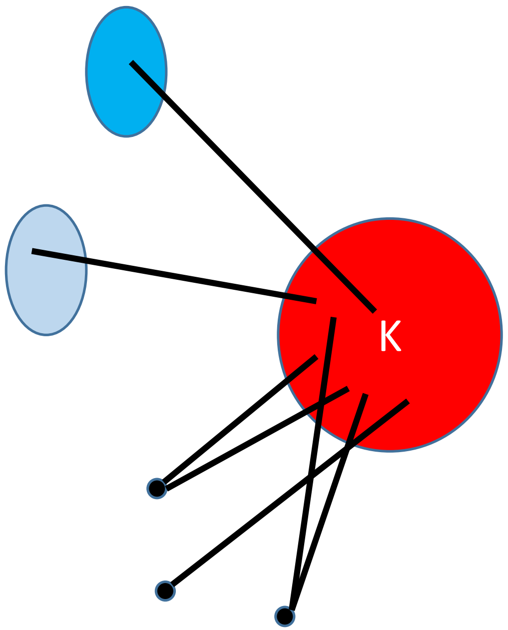


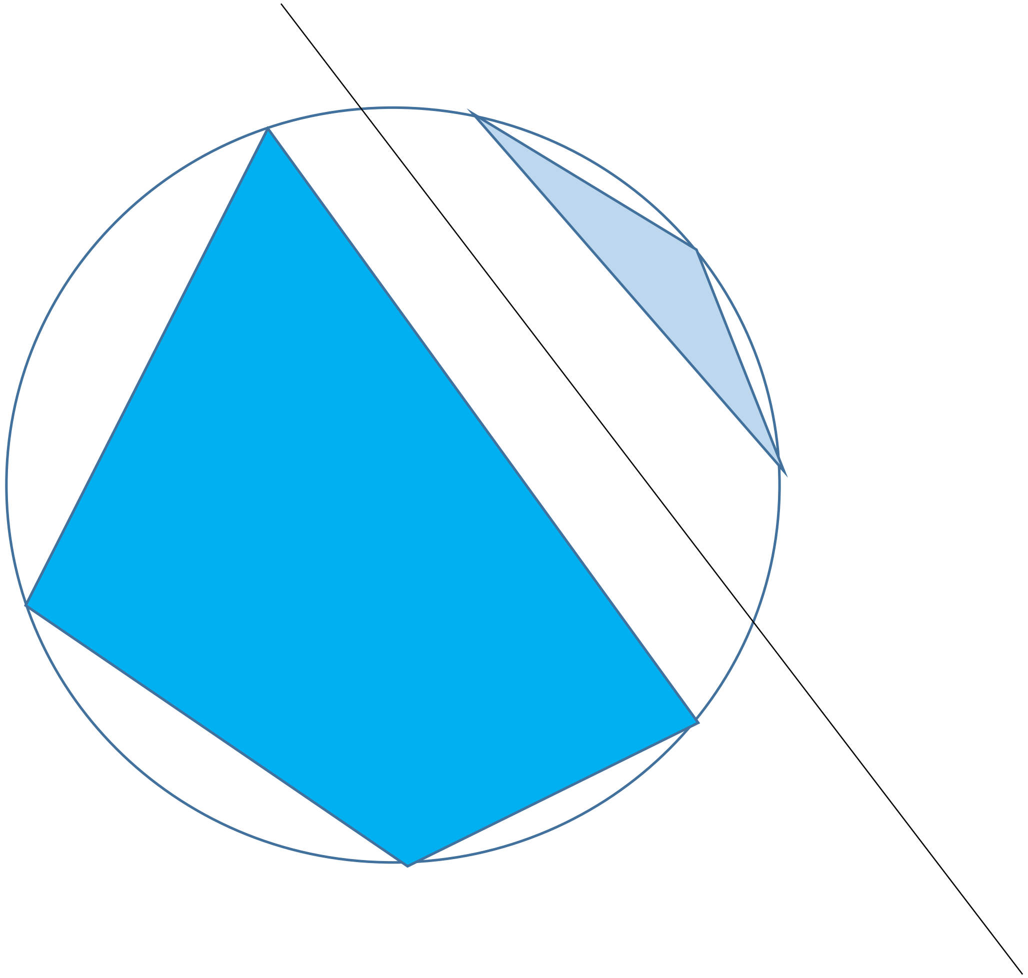


Proof of Thm 2

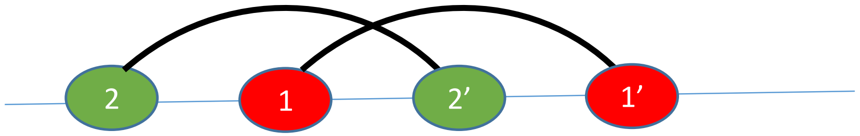


2) Claim 1. *K* is a clique, every vertex of *K* is adjacent to some vertex in every nontrivial component.

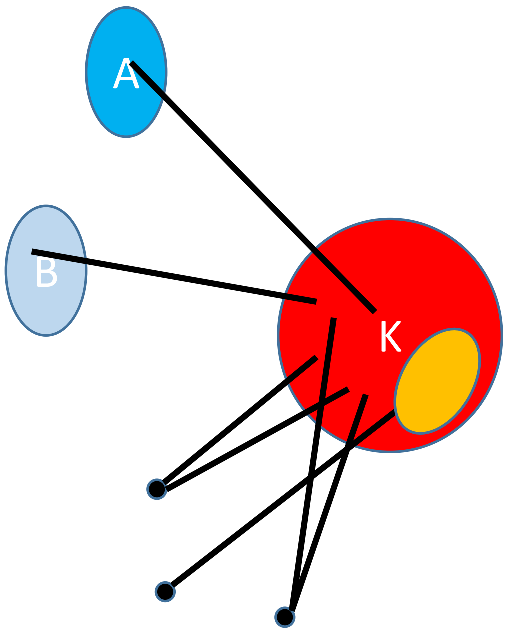


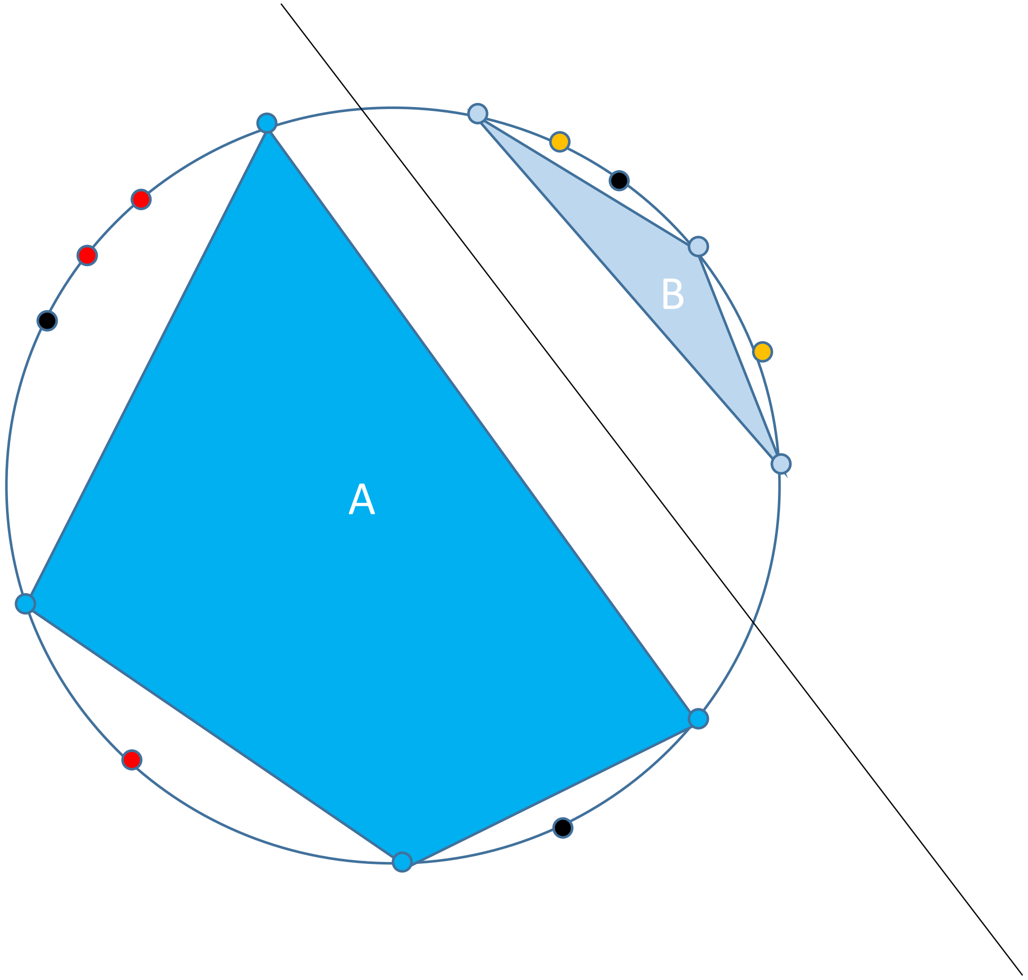


Proof of Thm 2

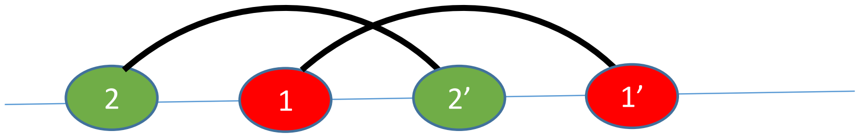


2) Claim 1. *K* is a clique, every vertex of *K* is adjacent to some vertex in every nontrivial component.

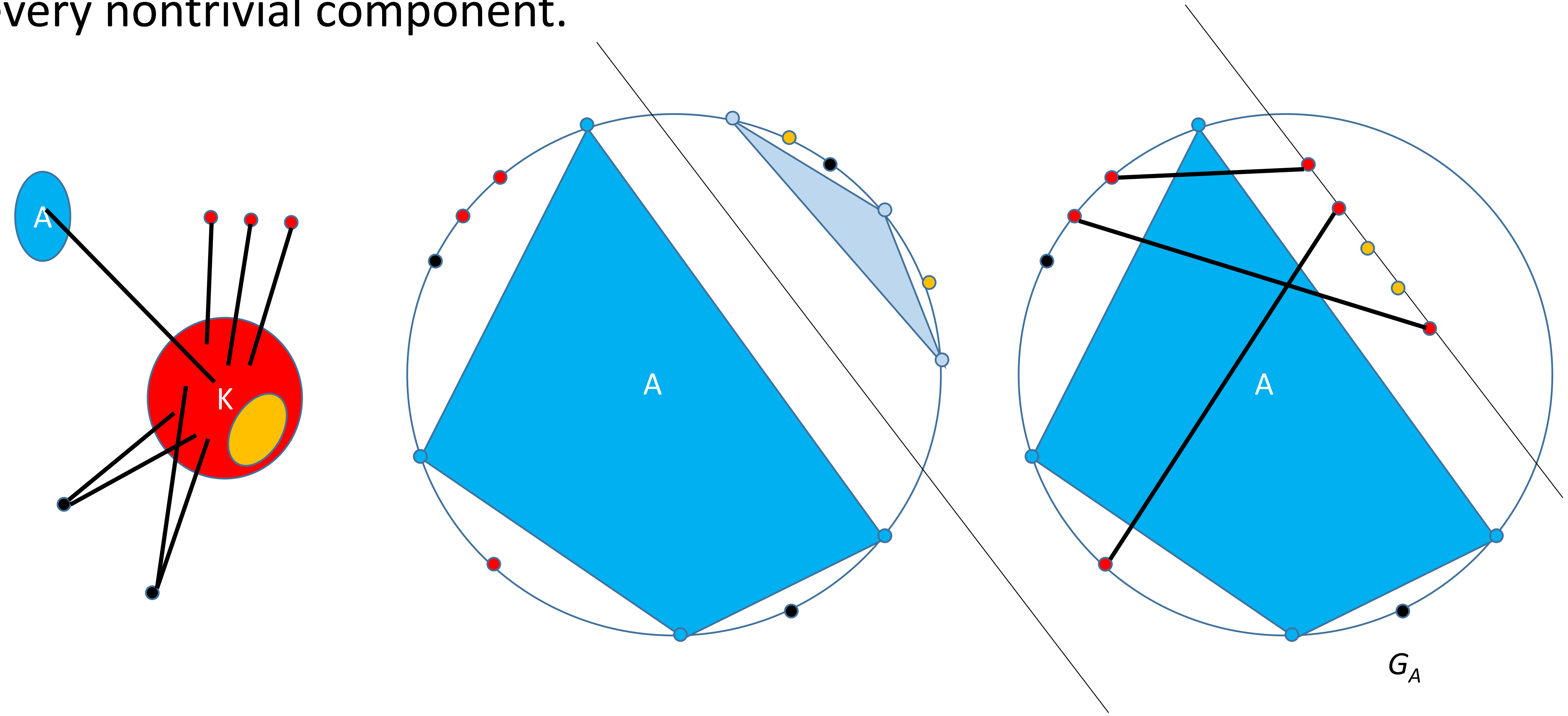




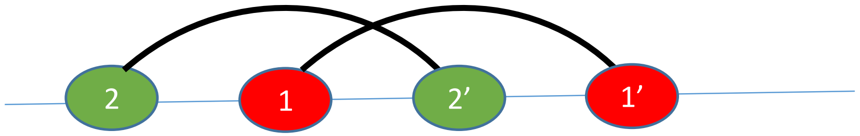
Proof of Thm 2



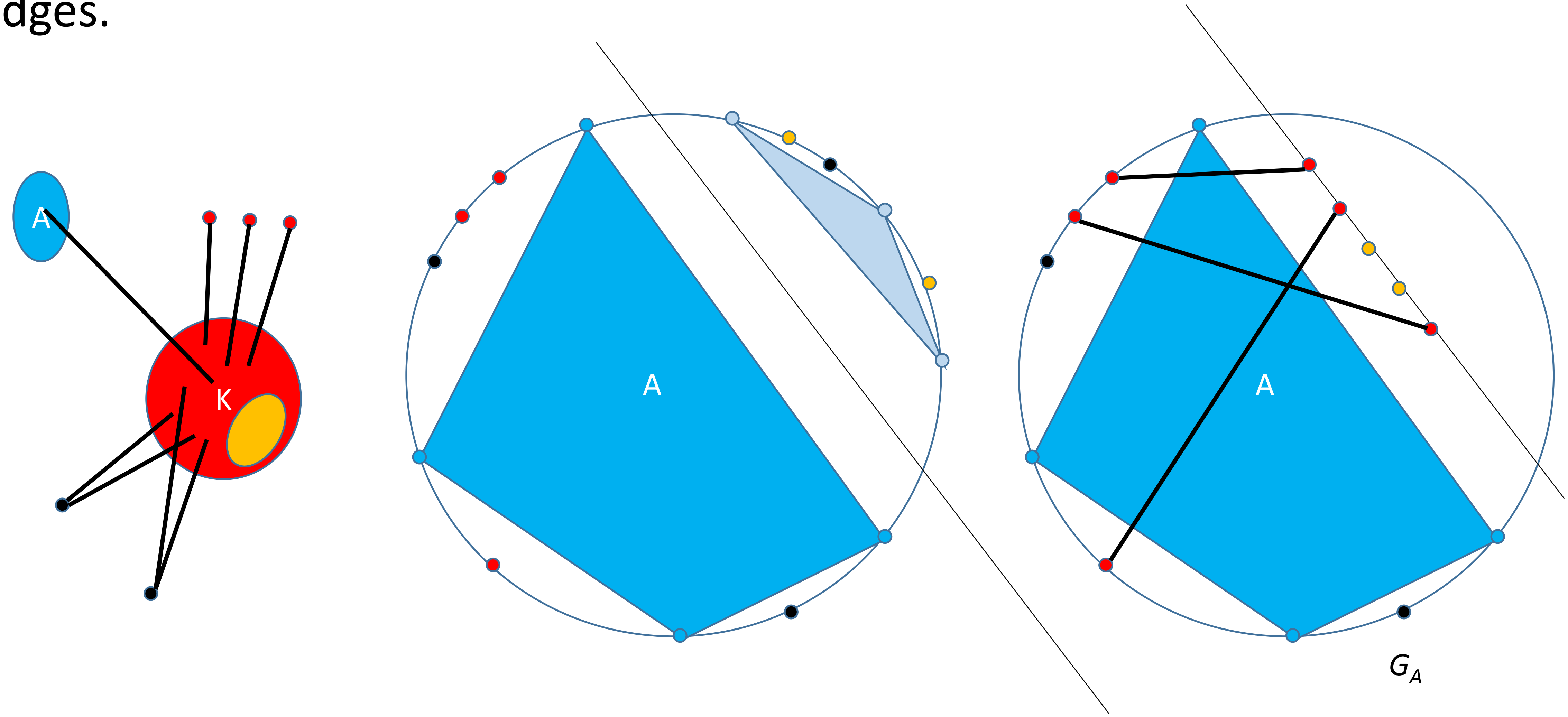
2) Claim 1. *K* is a clique, every vertex of *K* is adjacent to some vertex in every nontrivial component.



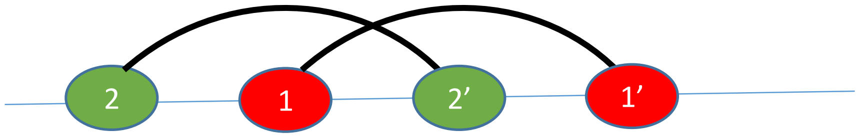
Proof of Thm 2



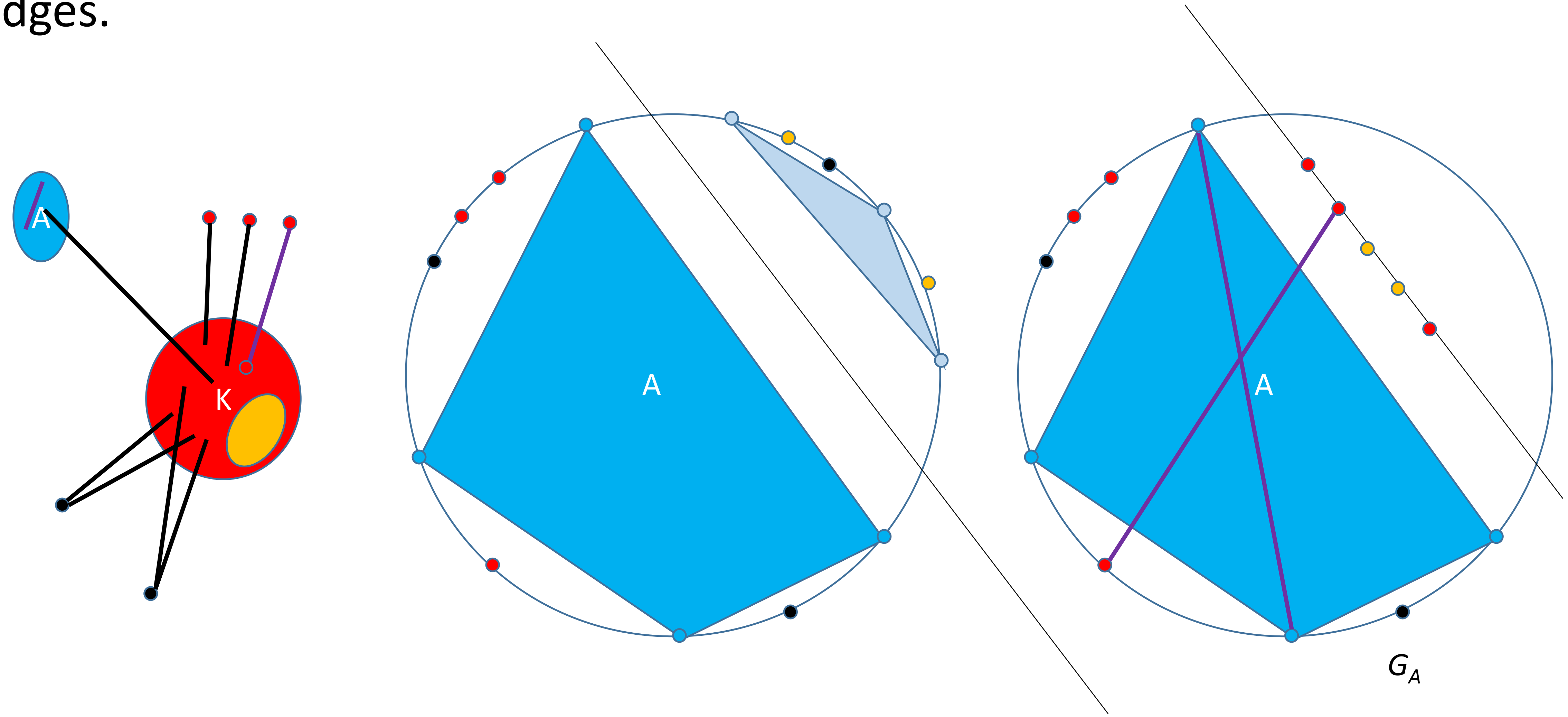
2) Claim 2. Both G_A and G_B contain no pair of interleaving independent edges.



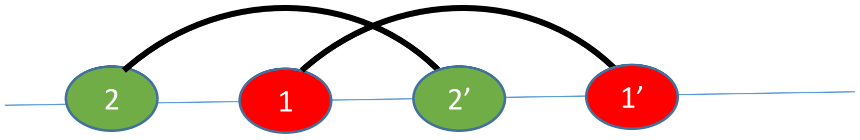
Proof of Thm 2



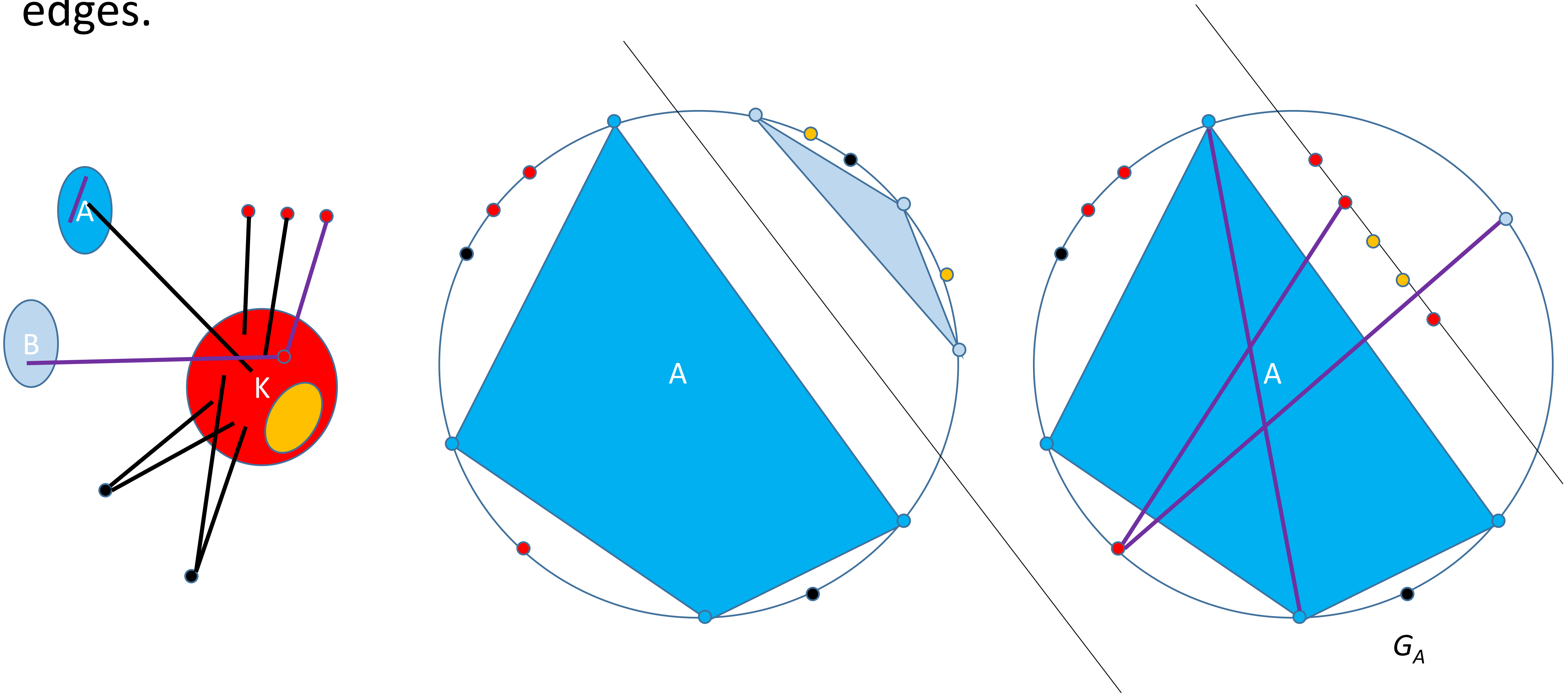
2) Claim 2. Both G_A and G_B contain no pair of interleaving independent edges.



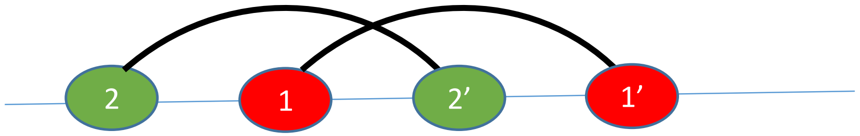
Proof of Thm 2



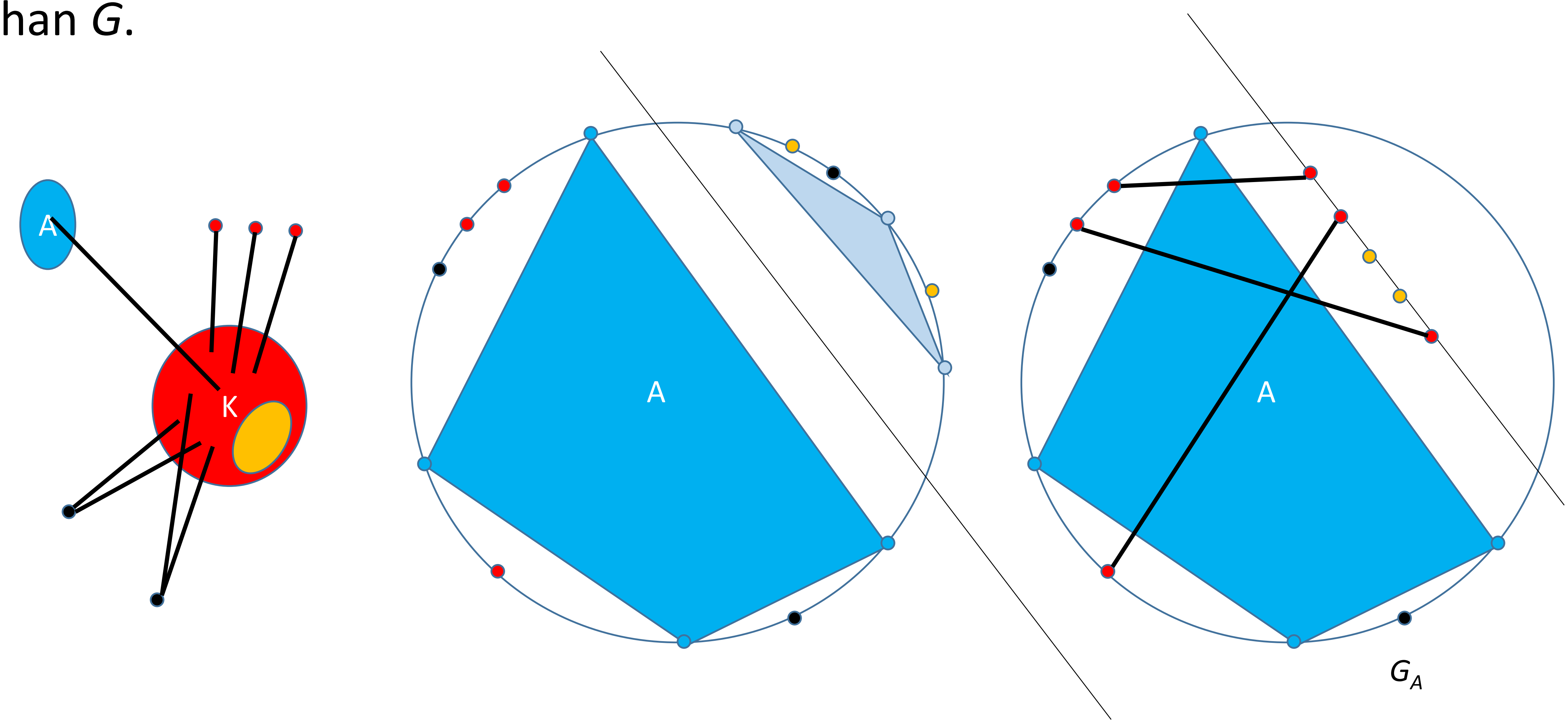
2) Claim 2. Both G_A and G_B contain no pair of interleaving independent edges.



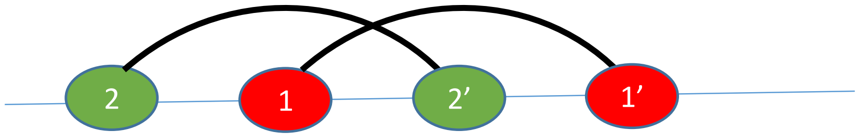
Proof of Thm 2



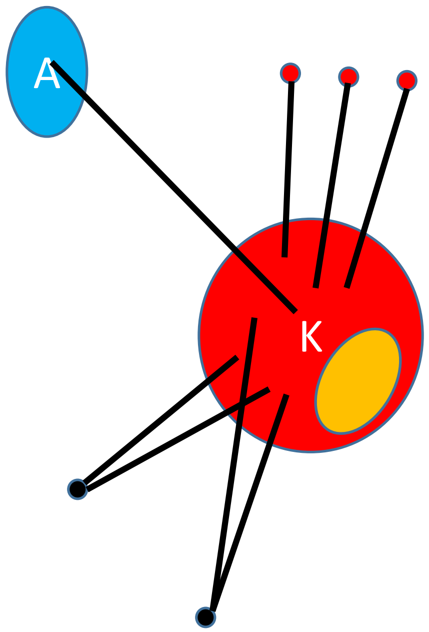
2) Claim 3. Both G_A and G_B contain fewer pairs of independent edges than G.

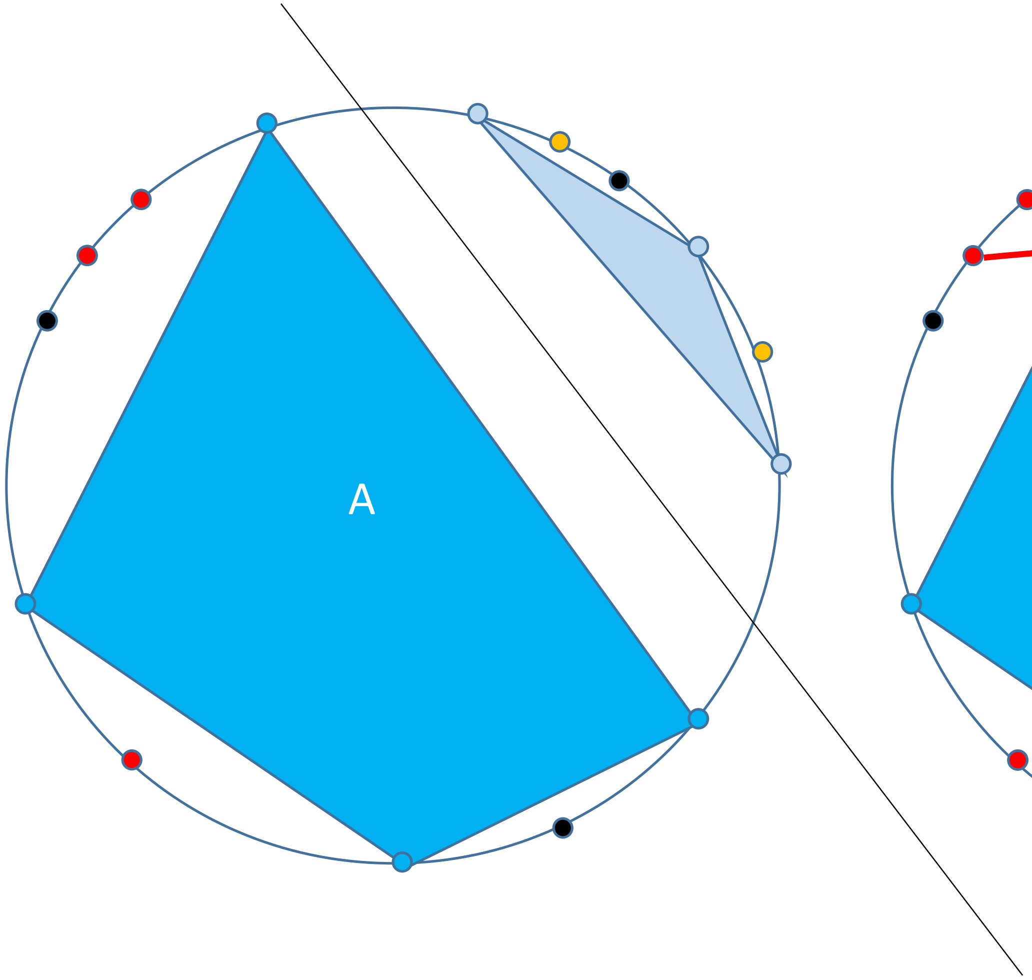


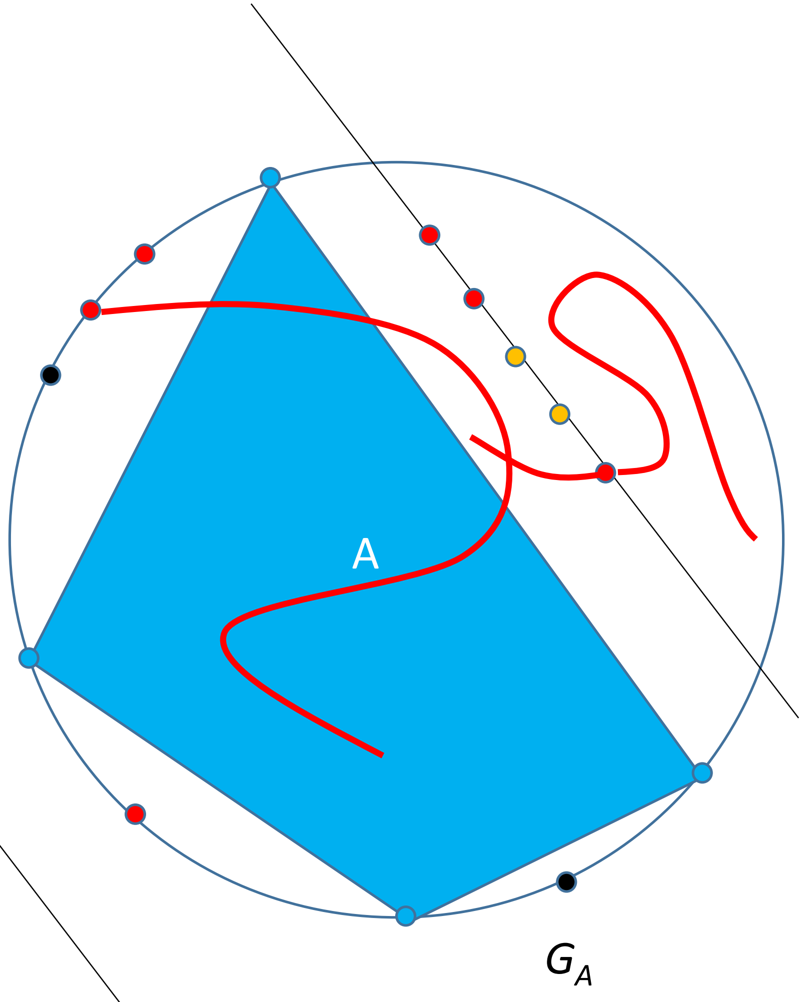
Proof of Thm 2



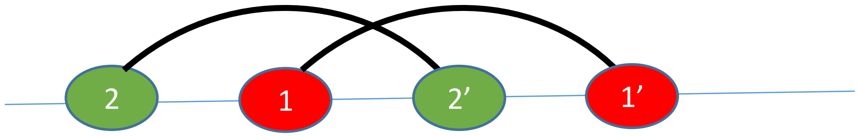
2) Induction. Both G_A and G_B admit constrained outer-string representations. And these can be merged.



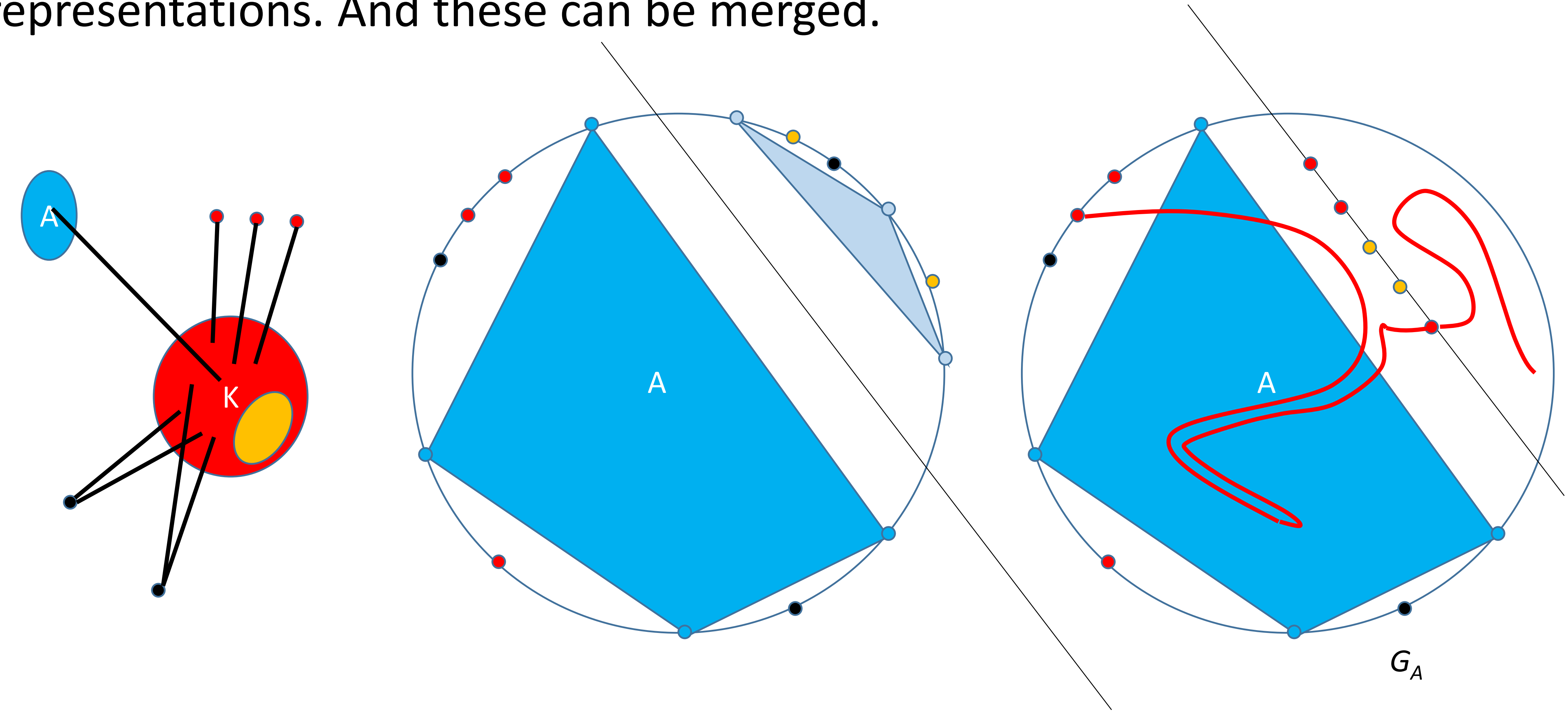




Proof of Thm 2

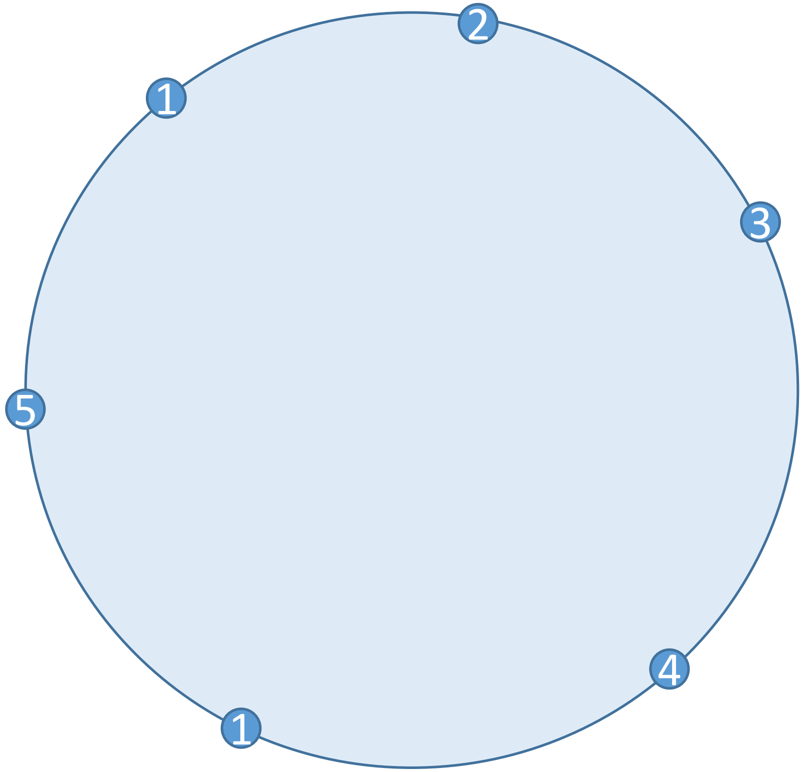


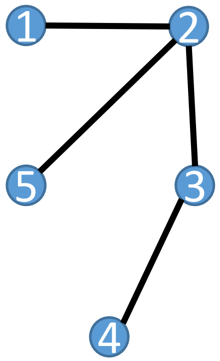
2) Induction. Both G_A and G_B admit constrained outer-string representations. And these can be merged.



Before the proof of Thm 3

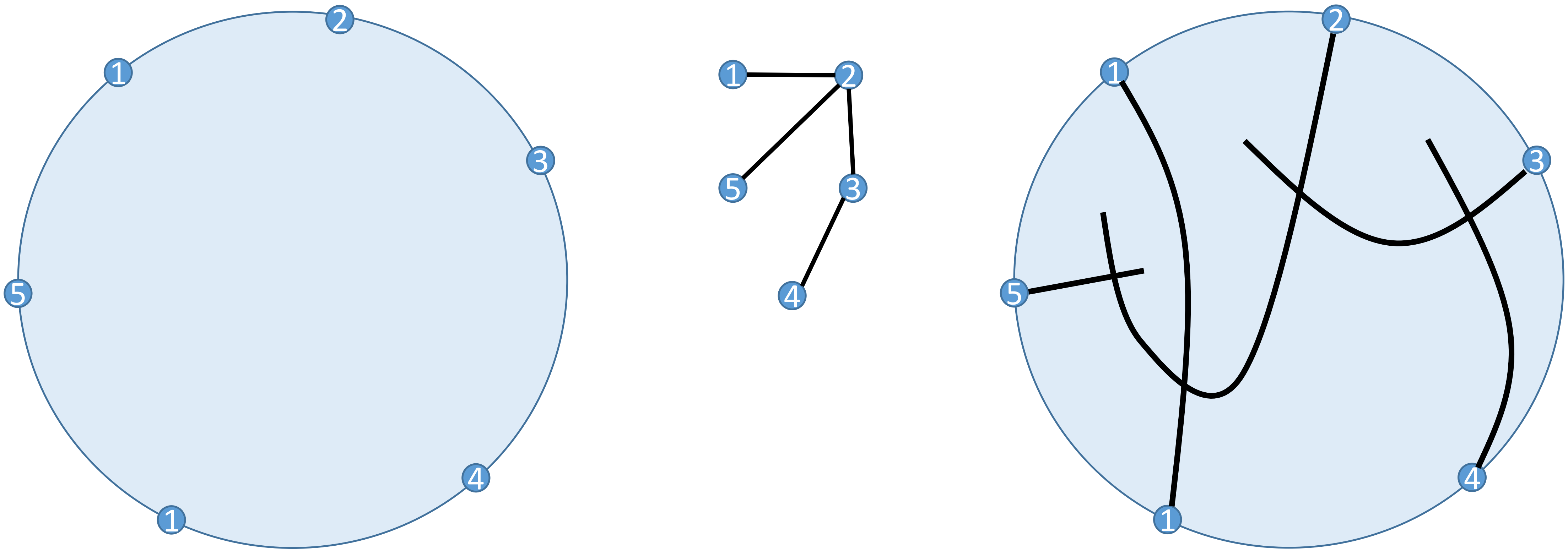
More general setting: In the input cyclic ordering, we allow some vertices appear twice. These occurrences then correspond to the end-points of the corresponding curves.





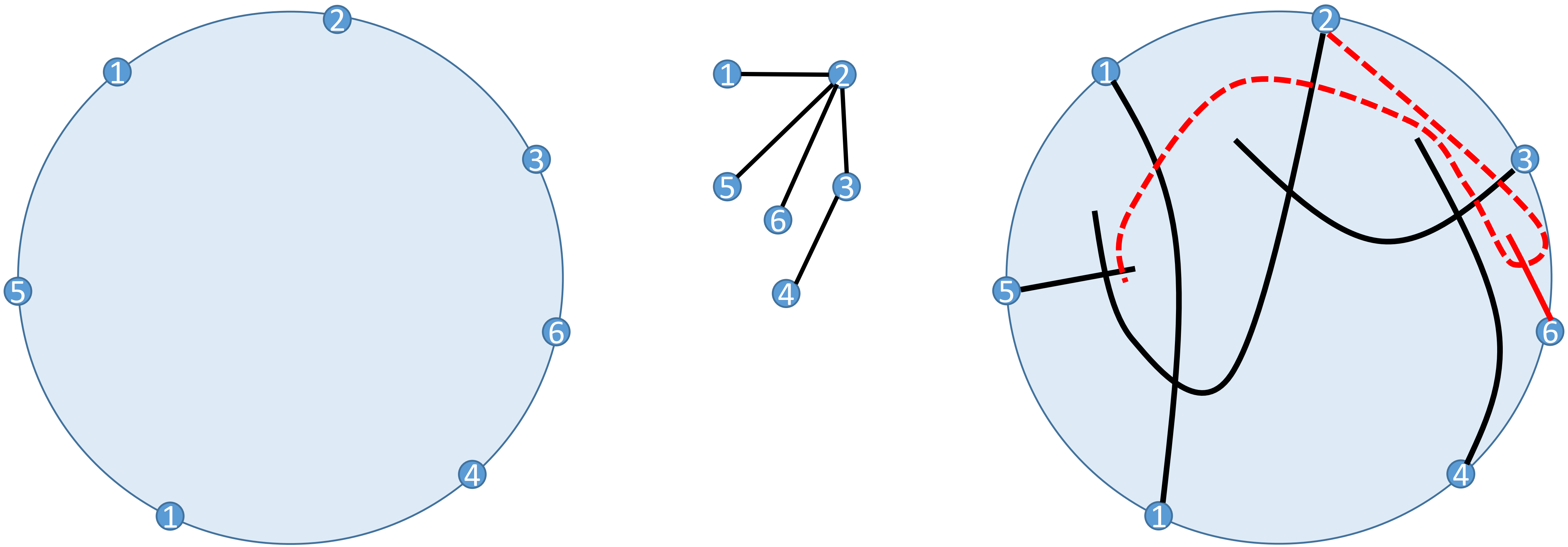
Before the proof of Thm 3

More general setting: In the input cyclic ordering, we allow some vertices appear twice. These occurrences then correspond to the end-points of the corresponding curves.



Before the proof of Thm 3

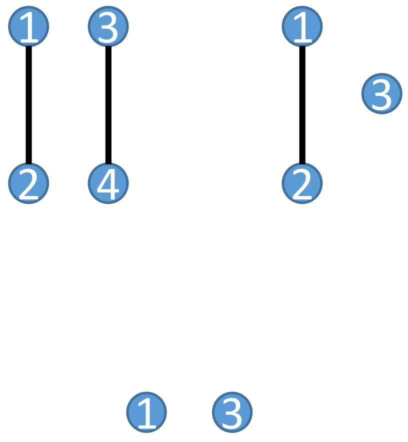
More general setting: In the input cyclic ordering, we allow some vertices appear twice. These occurrences then correspond to the end-points of the corresponding curves.

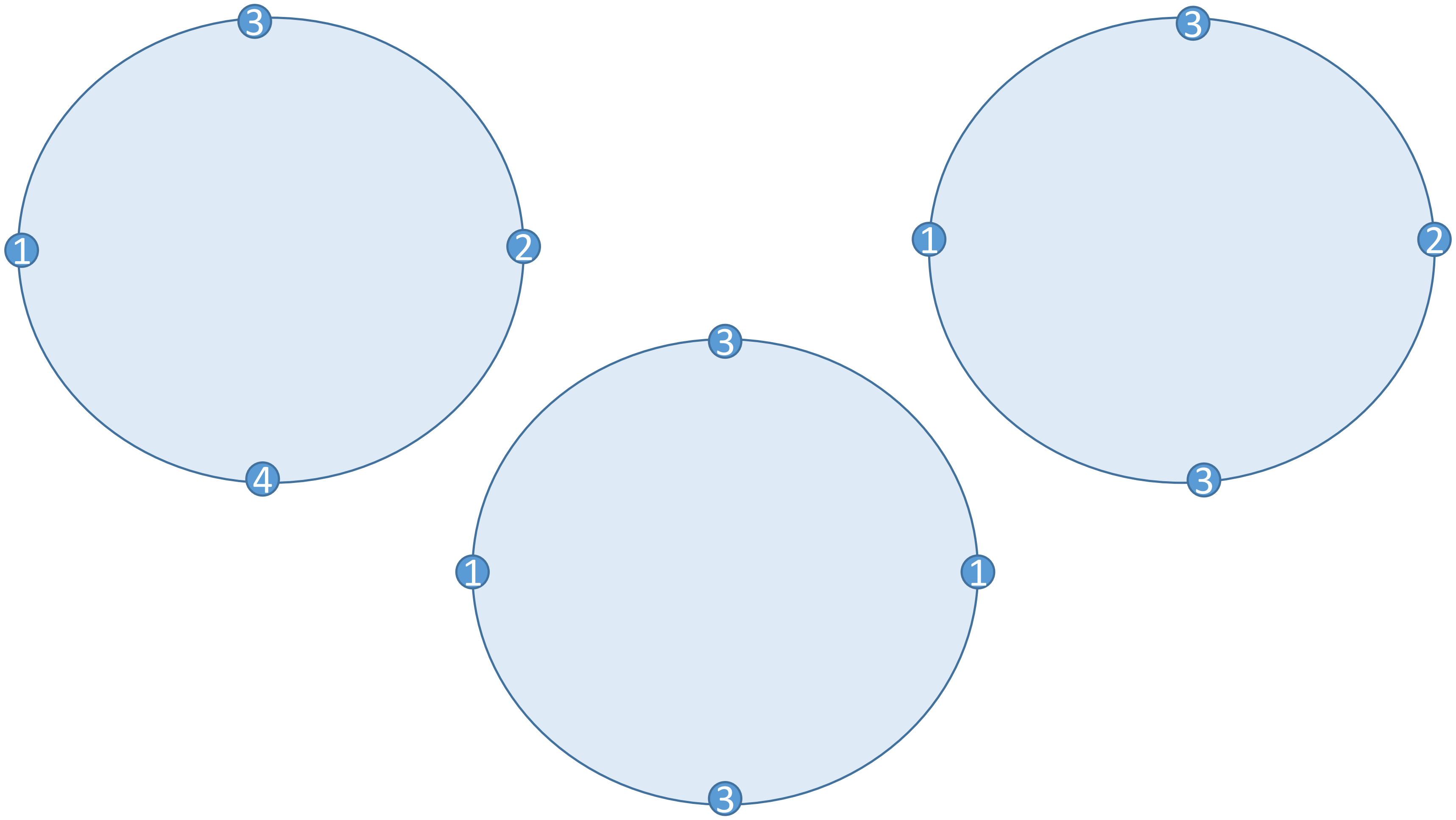


Obstructions for constrained outer-1-string trees

Thm 3: Constrained outer-1-string **trees** can be recognized in polynomial time, and are described by 3 types of forbidden configurations.

Pair-obstruction

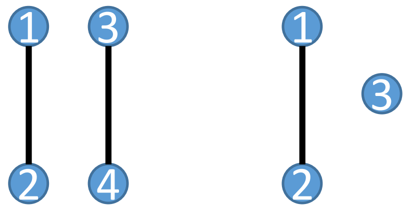


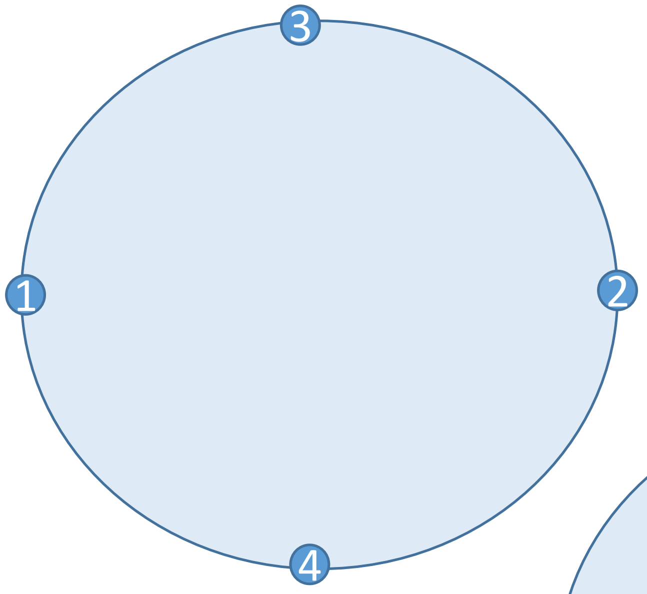


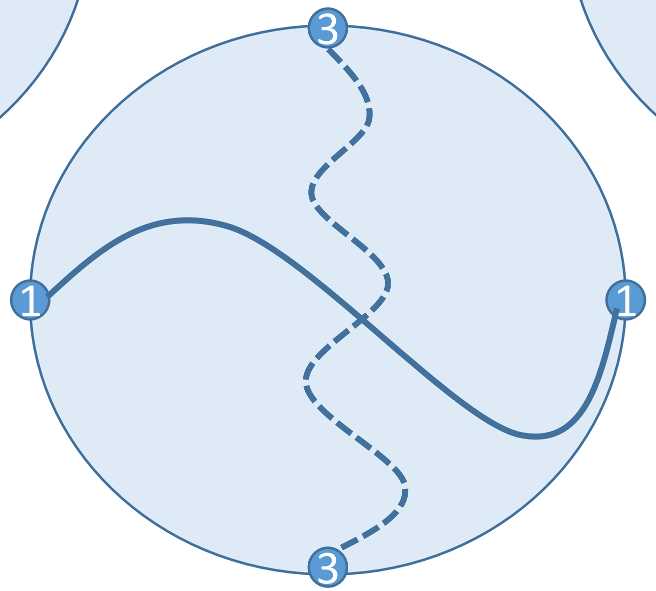
Obstructions for constrained outer-1-string trees

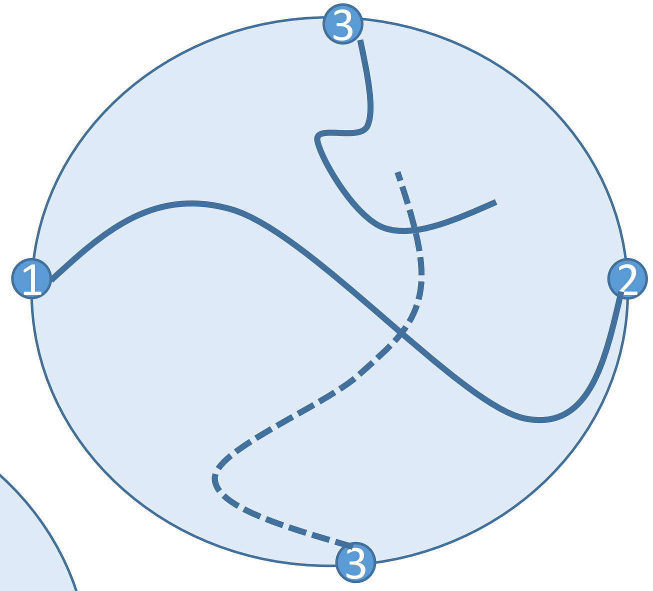
Thm 3: Constrained outer-1-string **trees** can be recognized in polynomial time, and are described by 3 types of forbidden configurations.

Pair-obstruction





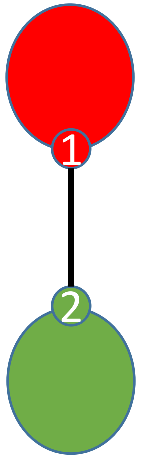


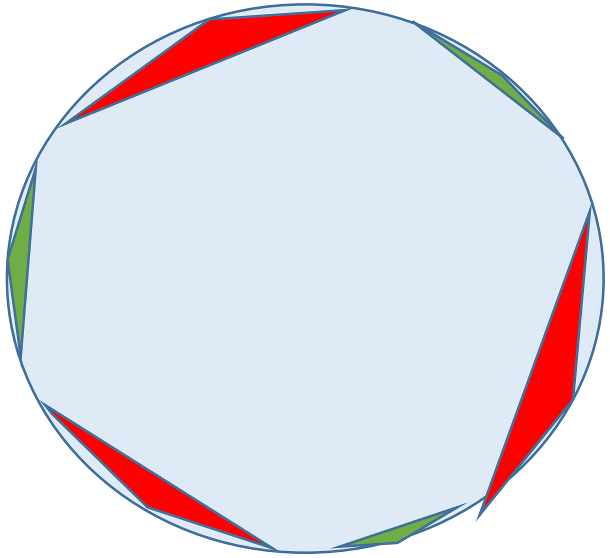


Obstructions for constrained outer-1-string trees

Thm 3: Constrained outer-1-string **trees** can be recognized in polynomial time, and are described by 3 types of forbidden configurations.

Bridge-obstruction

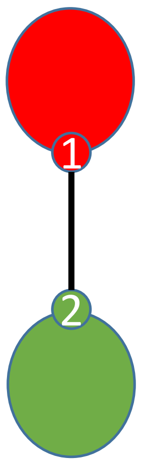


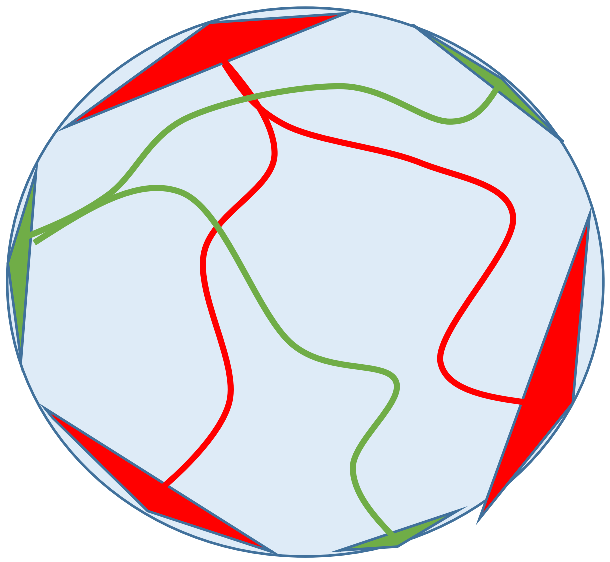


Obstructions for constrained outer-1-string trees

Thm 3: Constrained outer-1-string **trees** can be recognized in polynomial time, and are described by 3 types of forbidden configurations.

Bridge-obstruction

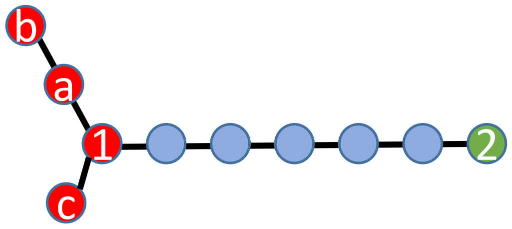


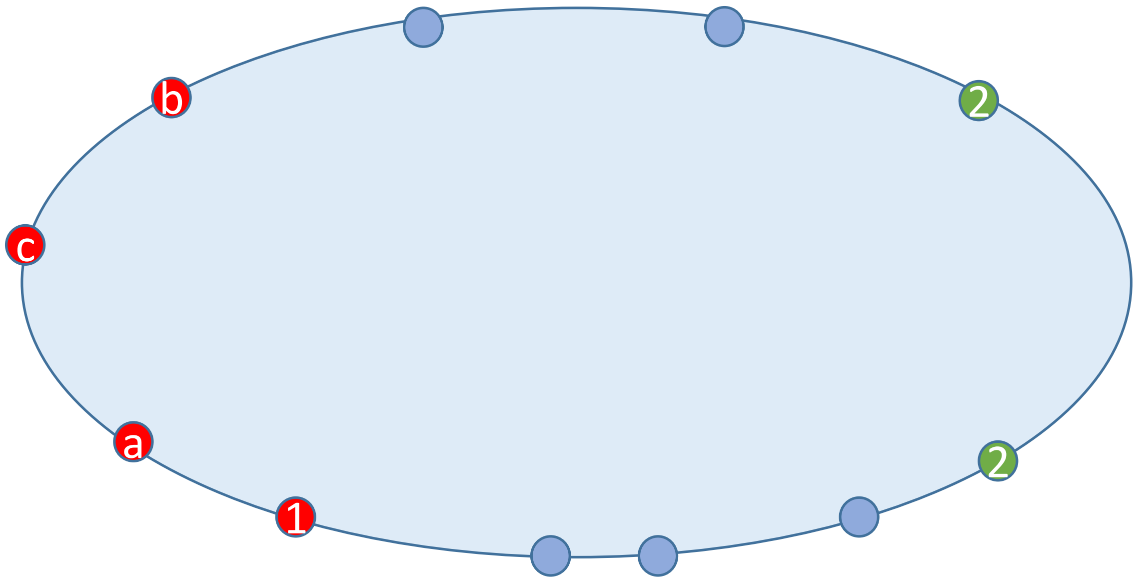


Obstructions for constrained outer-1-string trees

Thm 3: Constrained outer-1-string **trees** can be recognized in polynomial time, and are described by 3 types of forbidden configurations.

Path-obstruction

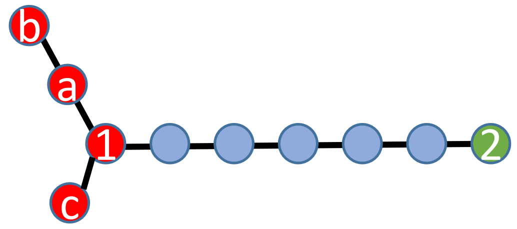


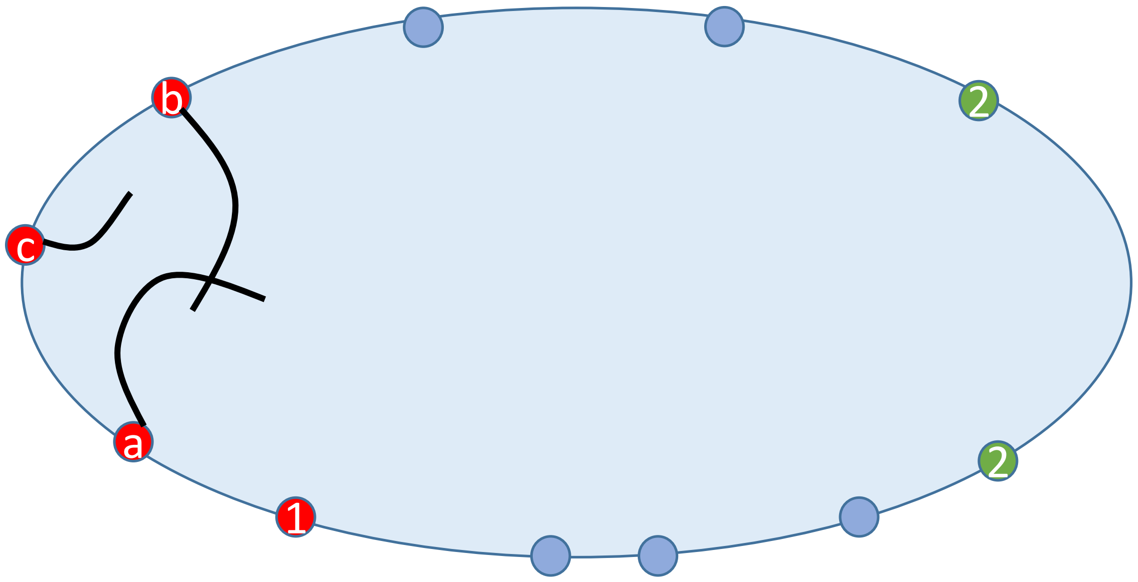


Obstructions for constrained outer-1-string trees

Thm 3: Constrained outer-1-string **trees** can be recognized in polynomial time, and are described by 3 types of forbidden configurations.

Path-obstruction

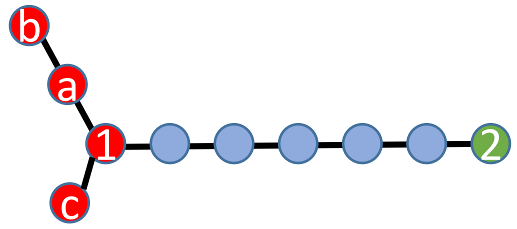


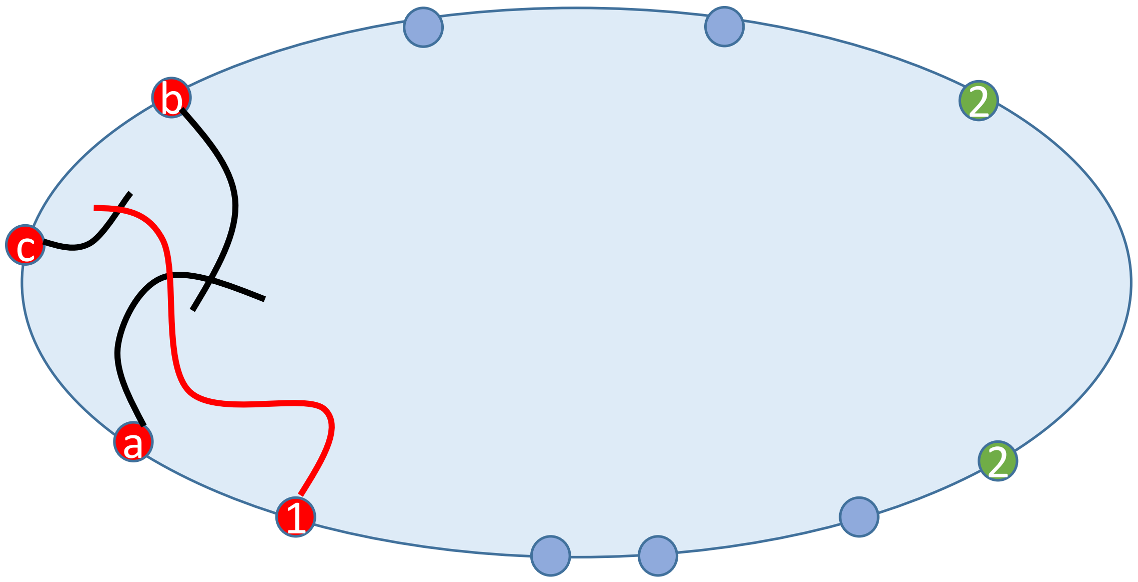


Obstructions for constrained outer-1-string trees

Thm 3: Constrained outer-1-string **trees** can be recognized in polynomial time, and are described by 3 types of forbidden configurations.

Path-obstruction

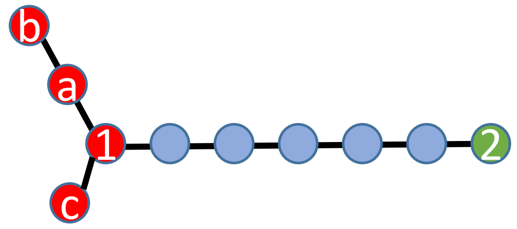


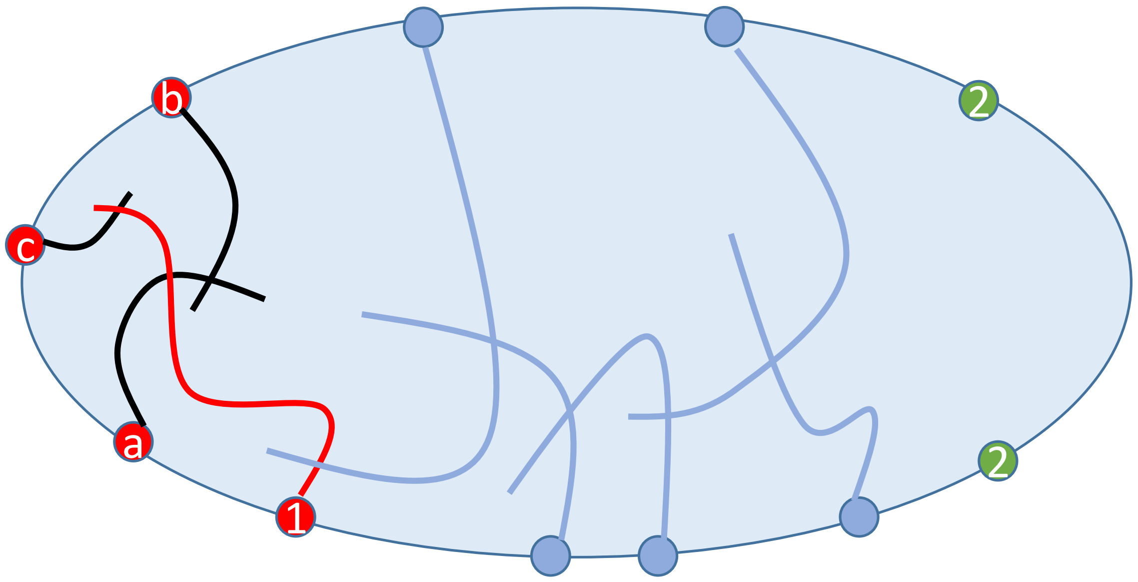


Obstructions for constrained outer-1-string trees

Thm 3: Constrained outer-1-string **trees** can be recognized in polynomial time, and are described by 3 types of forbidden configurations.

Path-obstruction

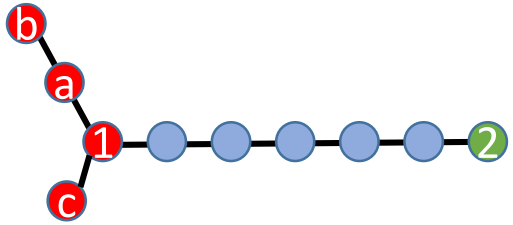


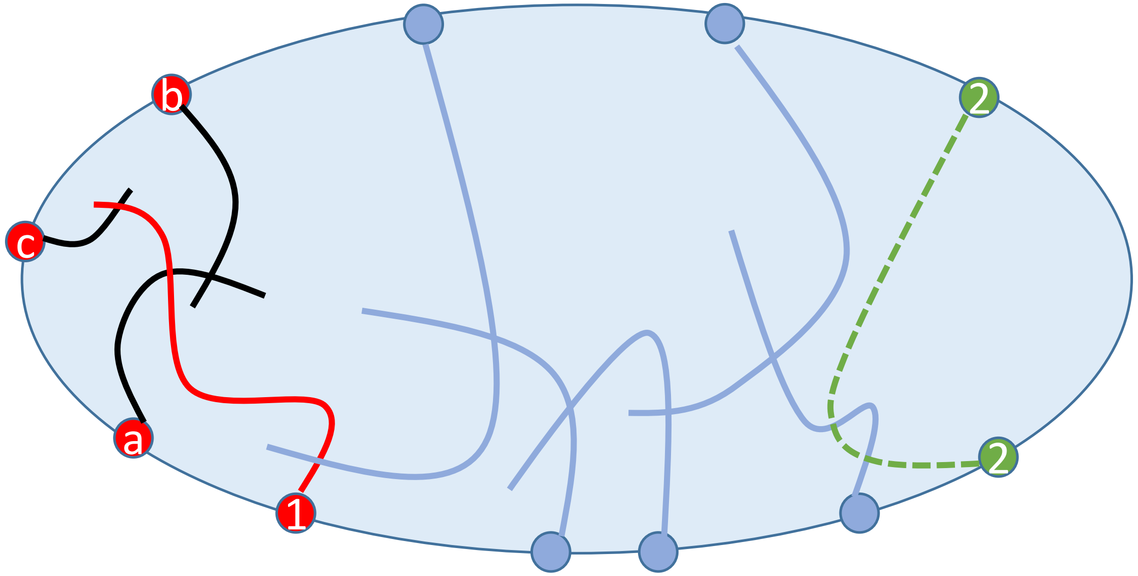


Obstructions for constrained outer-1-string trees

Thm 3: Constrained outer-1-string **trees** can be recognized in polynomial time, and are described by 3 types of forbidden configurations.

Path-obstruction

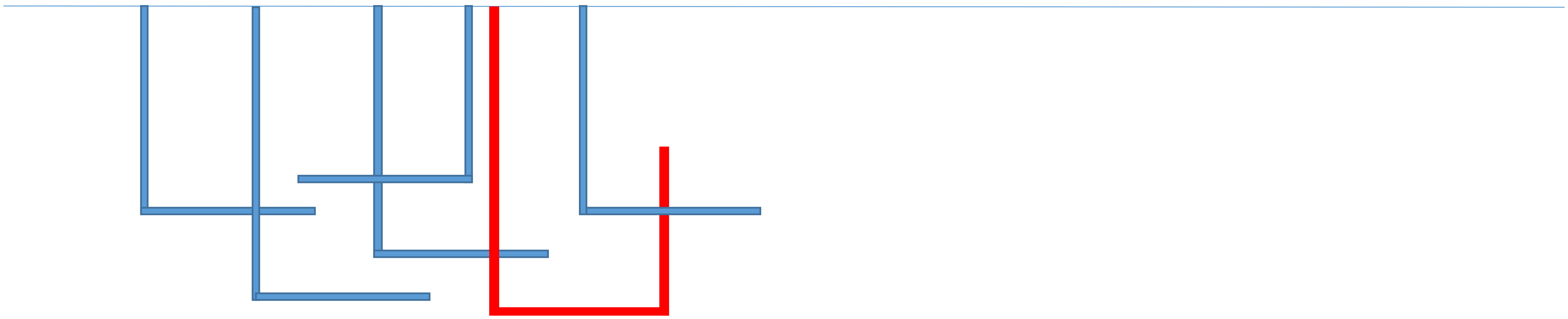




L-shapes and U-shapes

Thm 4: If a cyclically constrained tree admits an outer-1-string representation, then it admits a constrained U-shaped 1-string representation for any linear order induced by the cyclic one.

Linearly constrained trees that admit L-shaped 1-string representations can be recognized in polynomial time.



Thank you