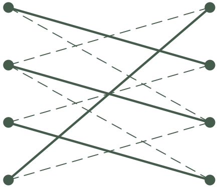
5 Matching, Covers, and Packing

From previous "more applied" areas of Graph theory we now shift closer towards the traditional theory, and survey the classical topics which belong to every GT curriculum.

We start with *matchings* in graphs, and related topics of *covering and packing*. Interestingly, graph matchings play a role, e.g., in statistical physics.



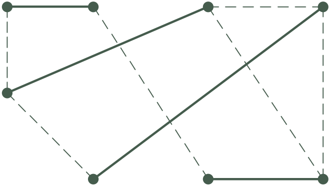
□

Brief outline of this lecture

- Matching in graphs and in bipartite graphs, stable marriage.
- Perfect matchings, Hall's theorem, factors in graphs, factorizations.
- Vertex cover. Packings in graphs, related coverings.
- Counting objects in graphs, e.g. spanning trees.

5.1 Matching in Graphs and Bipartite graphs

Imagine the task to "match suitable couples" such that nobody is in more than one relationship:



Definition 5.1. ***A matching*** in an undirected graph G is a subset of edges $M \subseteq E(G)$ such that no two edges from M share a vertex.
(The vertices incident to the edges in M are called *matched*. $|M|$ is the *size*.) □

Fact: We illustrate the definition with several simple claims:

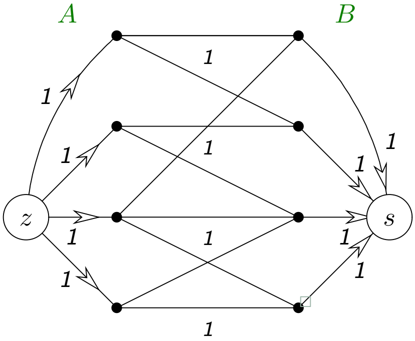
- No matching in G can have more than $\lfloor |V(G)|/2 \rfloor$ edges.
- If G is a star, then no matching in G has more than one edge. □
- In any G, one can find a matching with (at least) $\lceil \delta(G)/2 \rceil$ edges.
  (Simply pick the edges greedily from any un-matched vertex...)

matching = párování, $\delta(G)$ = min. degree = minimální stupeň

Solving the bipartite matching problem

The prime example and application of graph matching is in bipartite graphs. □

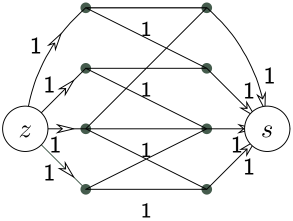
Algorithm 5.2. ***Finding maximum cardinality matching in a bipartite graph.***
Given a bipartite graph G with the vertex parts A, B, $A \dot\cup B = V(G)$, construct the following flow network $\bar{G}^+$:



- Run Algorithm 4.14 on $\bar{G}^+$.
- Since all the capacities are integral (1), the resulting max. flow will be integral as well (0 or 1). □Form $M \subseteq E(G)$ from the edges of flow value 1.

** Nalezení největšího párování v bipartitním grafu **

Flows to bipartite matchings



Proof of Algorithm 5.2:

- A matching $M \subseteq E(G)$ gives a flow sized $|M|$; val. 1 through each edge of M. □
- Conversely, assume a flow f computed by Algorithm 4.14 on $\bar{G}^+$. Proposition 4.16 implies that $f(e) \in \{0,1\}$ (only possible int. vals.) for all $e \in E(G^+)$.
- Then every vertex of G has total flow 0 or 1 (by the capacity from z or to s). □
- Select $M = \{e \in E(G) : f(e) = 1\}$ and so M is a matching and $|M| = \|f\|$. □

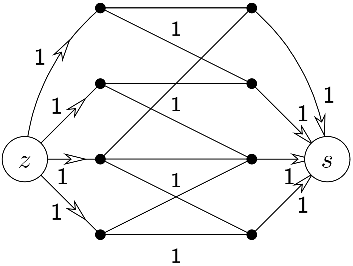
Closer relation to network flows

- Residual (augmenting) paths in $\bar{G}^+$ = *free alternating paths* in G;
  free – both ends of the path in G are currently not matched,
  alternating – every second edge of the path is in the current matching.
- Flow network cuts in $\bar{G}^+$ = so called *vertex covers* in G;
  i.e., the subsets of $V(G)$ hitting all the edges in $E(G)$ – see later.

free alternating path = volná střídavá cesta

Better runtime bound

Theorem 5.3. (Hopcroft–Karp algorithm) *For a bipartite graph $G = (V, E)$, one can find a maximum matching in time $O(|E| \cdot \sqrt{|V|})$.* □



Proof (sketch): Run Dinitz's algorithm (4.5) on the network $\bar{G}^+$ (see above): □

- In one round, all the shortest free alternating paths (edge-disjoint) are found in $O(|E|)$ time, and the same time is taken by the augmenting step. □
- After $\sqrt{|V|}$ rounds, no free alternating path is thus shorter than $\sqrt{|V|}$, and hence there are at most $O(\sqrt{|V|})$ such paths left! □
- All the leftover free alt paths are finished in next $O(\sqrt{|V|})$ rounds.

□

Matching and Optimization

Consider the following problems:

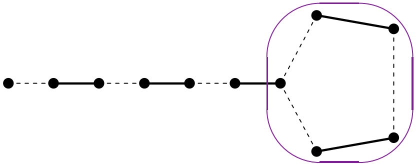
- *Maximum weighted matching* in a weighted bipartite graph (G, w) asks for a matching $M \subseteq E(G)$ maximizing $\sum_{e \in M} w(e)$.

  → A solution can again be obtained similarly to free alternating paths. □

- What about maximum cardinality *matching in general graphs*?

  → The idea of augmenting free alternating paths can be used, too. However;□ how can one find such an augmenting path efficiently in this general case?□

  → **Edmonds' Blossom algorithm**: recursively contract odd cycles forming alternating "*blossoms*" (until eventually reaching the easy bipart. case).



the "blossom"

□

- *Maximum weighted matching in general graphs* can be solved efficiently as well [Edmonds] — this is a crucial uneasy result in combinatorial optimization.

Stable marriage theorem

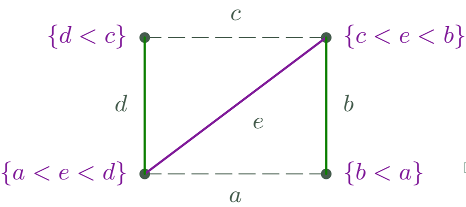
Consider also the following problem related to matchings:

- Given is a graph G together with *matching preferences*: for every vertex $v \in V(G)$, its matching prefence is a linear order $<_v$ on the edges incident with v.

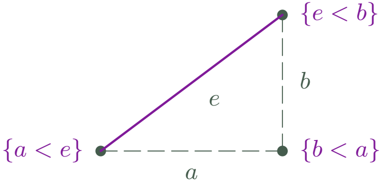
  Informally, for two edges $f = vw$ and $f' = vw'$, the relation $f <_v f'$ means that v would “better like” to be matched with w' than with w. □

- Call a *matching $M \subseteq E(G)$ stable* if for every $e \in E(G) \setminus M$ there exists $f \in M$ such that e, f share a common end v and $e <_v f$.

  Informally, no “local switch” in matching M is profitable for both parties (ends of e).



□



□

Theorem 5.4. (Gale and Shapley) *For any bipartite graph G and any matching preferences for G, there exists a stable matching.*

stable matching = stabilní párování

Proof of stable marriage

Theorem 5.4. For any bipartite graph G and any matching preferences for G, there exists a stable matching.

Proof (sketch): Let $A \dot\cup B = V(G)$ be the vertex parts of bipartite G. Consider a matching $M \subseteq E(G)$ and the following terms: □

- Call a vertex $b \in B$ *available for* a vertex $a \in A$ if $ab \in E(G)$ and the following holds: whenever b is matched with $a'b \in M$, then $a'b <_b ab$.

  Informally, b is happy to couple a (even possibly breaking up with its current match).□

- Call a vertex $a \in A$ *content with* M if a is not matched at all, or a is matched in M to the best available neighbour in B. □

Starting with empty $M = \emptyset$, repeat the following step (until it fails to find suit. a, b):

- Find an unmatched vertex $a \in A$ and its best available neighbour $b \in B$, and add ab to M while possibly discarding another b-edge from M. □

One can observe two invariants of this procedure; (A) all the vertices in A stay content with M, and (B) no v. in B gets worse while one $b \in B$ gets strictly better each step.□

The procedure thus terminates, eventually, and the result is a stable matching. □

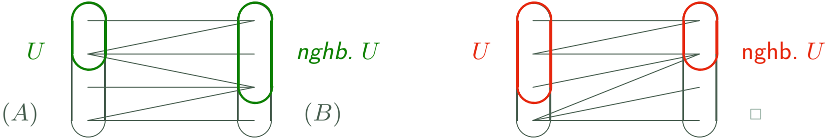
5.2 Perfect matchings

A matching of maximum possible cardinality is characterized as follows:

Definition: A matching $M \subseteq E(G)$ in a graph G is a *perfect matching* if every vertex of G is an end of some edge in M. □

Fact: A perfect matching exists only if the graph has an even number of vertices. □

Theorem 5.5. *Let G be a bipartite graph with the parts A, B, i.e. $A \dot\cup B = V(G)$. Then G contains a matching of size $|A|$ (in other words, with whole A matched) if, and only if, for every $U \subseteq A$ the number of neighbours of U in B is at least $|U|$.*



□

Recall Hall's Theorem 4.19:

There exists a system of its distinct representatives of $\{M_1, M_2, \ldots, M_k\}$ if, and only if,

$$\forall J \subseteq \{1, 2, \ldots, k\} : \left| \bigcup_{j \in J} M_j \right| \geq |J| \,,$$

i.e., the union of any subfamily has at least as many elements as the number of sets in it. □

Where is Thm. 5.5 in this one? $A =$ the sets $M_1, \ldots, M_k$, $B =$ their elements, $E(G) \sim \in$.

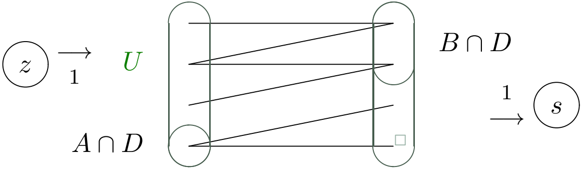
perfect matching = perfektní párování

Hall's theorem revisited

Theorem 5.5. (repeated) *Let G be a bipartite graph with the parts A, B, i.e. $A \dot\cup B = V(G)$. Then G contains a matching of size $|A|$ (in other words, with whole A matched) if, and only if, for every $U \subseteq A$ the number of neighbours of U in B is at least $|U|$.*

Proof (a sketch, see Theorem 5.13 for a related full proof):

- We apply the flow–cut duality (Theorem 4.14 and Algorithm 5.2) onto the associated network $\bar{G}^+$ (with all capacities 1). Schematically:



- So, there is no matching of size $|A|$ iff $\bar{G}^+$ has a cut of size $< |A|$.

  Let $D \subseteq V(G)$ by vertices incid. with this cut, one with each cut edge; $|D| < |A|$.

- Consider now $U = A \setminus D$—all the neighbours of U must belong to $B \cap D$, and

$$|B \cap D| = |D| - |A \cap D| < |A| - |A \cap D| = |A \setminus D| = |U| \,,$$

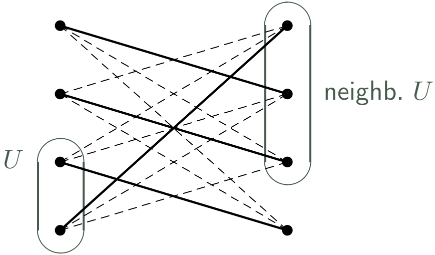
a contradiction to the assumptions. □

Theorem 5.5. (repeated) *Let G be a bipartite graph with the parts A, B, i.e. $A \dot{\cup} B = V(G)$. Then G contains a matching of size $|A|$ (in other words, with whole A matched) if, and only if, for every $U \subseteq A$ the number of neighbours of U in B is at least $|U|$.*

Corollary 5.6. *For every $k > 0$, a k-regular bipartite graph contains a perf. matching.*

Proof: Note that $|A| = |B|$ where A, B are the parts of the bip. graph.

Take any $U \subseteq A$. The number of edges incident with U is $k \cdot |U|$ (by k-regularity), and each neighbour of U in B meets $\leq k$ of these edges.

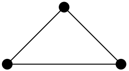


Hence there are at least $(k \cdot |U|)/\,k = |U|$ neighbours of U in B. □

Perfect matchings in regular graphs

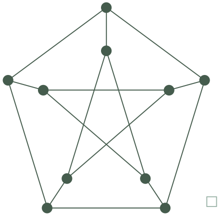
Consider a k-regular graph G.

- Recall (Cor. 5.6); if G is bipartite, then a perfect matching exists. □
- For non-bipartite G, a perfect matching may not exist.
  Take, e.g., any odd cycle ($k = 2$)...

  
  □

  What if the number of vertices is even? No, take two odd cycles... □
- An old result of Petersen claims ($k = 3$):

  If G is 3-regular and 2-connected (enough to say "G has no bridges"), then G has a perfect matching.


□

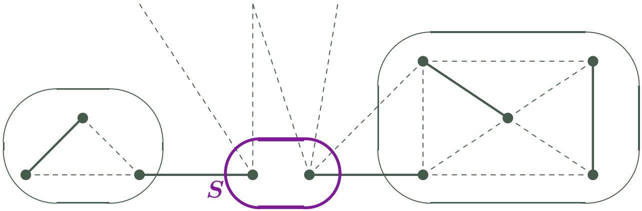
- Is there some nice general characterization of the graphs with a perfect matching?

Tutte's condition

Theorem 5.7. (Tutte)
A graph G has a perfect matching if, and only if, for all $S \subseteq V(G)$;

- *the graph $G \setminus S$ has at most $|S|$ odd components (those of odd number of vert.).*

Note that the stated condition is clearly necessary:



Though, we skip the rather involved proof of this beautiful characterization...

* *Tutteho podmínka* *

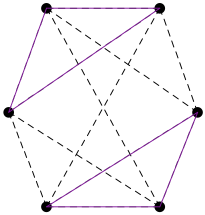
5.3 Factors as a generalization of matching

A perfect matching has the significant property that every graph vertex is incident with exactly one matching edge. What if this number "one" is raised up?

Definition 5.8. ***An r-factor*** in an undirected graph G is a spanning r-regular (i.e., on all vert. of G and with all degrees r) subgraph of G.

Fact: A perfect matching in a graph is the same as a 1-*factor*. □

Example 5.9. *Any 4-regular graph contains a 2-factor.*



A solution is short, but tricky…

Let G be 4-regular, and assume that G is connected (if not, find 2-factors in each of its components). Since 4 is an even degree, Euler's theorem 2.17 applies to G. □

Hence G consists of one *closed trail* $W = (v_0, e_1, v_1, e_2, v_2, \dots, e_m, v_0)$; a walk covering every edge of G exactly once. Note that the length $m = 2|V(G)|$ is even in this case.

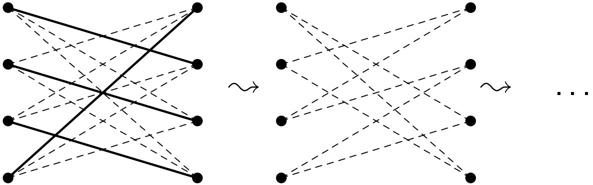
We select every second edge of W, i.e., $e_1, e_3, e_5, \dots, e_{m-1}$. So, for every vertex of G we select precisely two of the four incident edges, giving a 2-factor. □

r-factor = r-faktor

Factorization into perfect matchings

We speak about a *factorization* of a graph (cf. more general packing in Section 5.4) if the edge set of a graph is to be decomposed into edge-disjoint subgraphs of a certain kind... A *r-factorization* is a factorization into r-factors. □

Proposition 5.10. *For every $k > 0$, a k-regular bipartite graph has a 1-factorization.*□



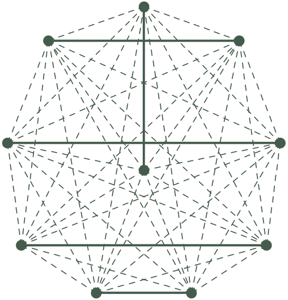
Proof: Find the first 1-factor $M \subseteq E(G)$ by Corollary 5.6, and continue by induction with factorization of the $(k-1)$-regular subgraph $G \setminus M$. □

factorization = rozklad

Proposition 5.11. *For every n, the complete graph K_{2n} has a 1-factorization.* □

Proof, a brief sketch:

Take the perfect matching of n edges from the picture, and rotate it $2n-1$ times around the central vertex.



Why is this a factorization? □

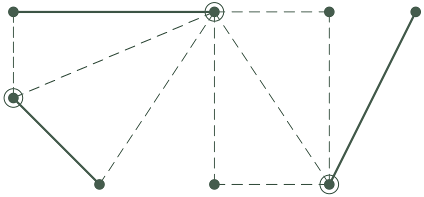
Very briefly, every edge of K_{2n} not incident with the centre in this picture has its "diagonal length", and the chosen perfect matching covers each such "length" once. □

5.4 Packing and Covers

Consider now the following kind of "two-sided" graph problems in which the task is to disjointly pack (a large number of) certain objects into a graph—the *packing* side. □

Seeing the problem from the opposite side, one could ask about finding (a few of) obstacles which prevent packing any more objects in the graph—the *covering* side.□

- For example, the network flow–cut duality is of this kind; we would like to "pack" as much flow as possible, while the obstacles are the edges of a z–s cut in the network. □
- In the setting of matchings, the following related notion pops up:



Definition: A *vertex cover* in a graph G is such a set $C \subseteq V(G)$ that every edge of G is incident with a vertex of C, i.e. $G \setminus C$ has no edges. □

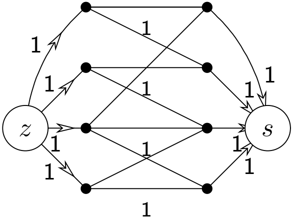
Proposition 5.12. *If C is a vertex cover in a graph G, then no matching in G may contain more than $|C|$ edges.*

packing = pakování, covering = pokrývání, vertex cover = vrcholové pokrytí

Matching–cover duality in bipartite graphs

Theorem 5.13. (König) *For an undirected bipartite graph G, the maximum cardinality of a matching in G equals the minimum cardinality of a vertex cover in G.*

Proof: Recall (Prop. 5.12) that no matching may be larger than the minimum cardinality vertex cover in G. □



Construct the same flow network $\bar{G}^+$ as in Algorithm 5.2. The maximum flow (01-valued) in $\bar{G}^+$, corresponding to the maximum cardinality matching $M \subseteq E(G)$, is by Theorem 4.10 equal to the minimum edge-cut $X \subseteq E(G^+)$ in $\bar{G}^+$. □

Selecting one vertex from each edge of X, and not z, s, one gets $C \subseteq V(G)$ hitting all of X, such that clearly $|C| \leq |X| = |M|$. We claim that C is a vertex cover in G: Suppose not, and let $e = uv$ be a ("leftover") edge in $G \setminus C$. Then z–u–v–s is a residual path in $\bar{G}^+$ avoiding X, a contradiction to X being a cut in $\bar{G}^+$. □

Hence, together with Prop. 5.12, $|M| \leq |C| \leq |X| = |M|$, and the equality holds. □

More on packing–covering dualities

- For general graphs, König's Thm. 5.13 fails; e.g., the complete graph K_n has a minimum vertex cover of $n-1$ vertices while a maximum matching of only $\lfloor n/2 \rfloor$ edges.□ Though:

Proposition 5.14. *In any graph G, the maximum cardinality of a matching and the minimum vertex cover size are within a multiplicative factor of 2 away from each other.*□

Proof: Let $C \subseteq V(G)$ be a smallest vertex cover in G. Then no matching in G is larger than $|C|$ (Proposition 5.12); $|M| \leq |C|$.

Conversely, take a maximum matching $M \subseteq E(G)$, and set $D = V(M)$ the vertices incident with M. Since M is maximum, no edge of G may avoid D, and so D is a vertex cover of size $|D| \leq 2 \cdot |M|$. □ □

- In other words, G cont. a matching of m edges or a v. cover of $\leq 2m$ vertices.

  A prime and famous result (a difficult one) in this direction is the following:

Theorem 5.15. (Erdős–Pósa) *There ex. a function f such that, for any $k \in \mathbb{N}$, every graph contains k pairwise disjoint cycles or $\leq f(k)$ vertices hitting every cycle of G.*

(Vertices hitting every cycle in a graph are called *feedback vertex set FVS*.)

Packing edge-disjoint trees

In this example, note that we deal with edge-disjoint objects (spanning trees), i.e., we allow them to share vertices but not edges of the underlying graph G.

- If $\mathcal{P}$ is a *partition* of the vertex set $V(G)$, then $|\mathcal{P}|$ denotes the number of parts. An edge $e \in E(G)$ is *crossing $\mathcal{P}$* if its two ends are in distinct parts.
- If $T \subseteq G$ is a spanning tree then, clearly, at least $|\mathcal{P}| - 1$ of its edges cross $\mathcal{P}$. □

Theorem 5.16. (Nash–Williams, Tutte)
A (multi)graph contains k pairwise edge-disjoint spanning trees if, and only if, every partition $\mathcal{P}$ of $V(G)$ is crossed by at least $k \cdot (|\mathcal{P}| - 1)$ edges of G. □

The proof of this statement (in the "if" direction) is not easy. . .

Theorem 5.17. (Nash–Williams)
A (multi)graph can be partitioned into $\leq k$ forests if, and only if, every vertex subset $U \subseteq V(G)$ induces a subgraph with $\leq k \cdot (|U| - 1)$ edges of G. □

The proof here is similar to the previous one, and not easy either (see [Diestel]).

* *Pakování hranově disjunktních stromů* *

Path covers

In the last example of covering-packing duality, we aim to cover all the vertices of a digraph by pairwise disjoint directed paths, and relate these paths to pairwise nonadjacent vertices (of the paths). □ The full result reads:

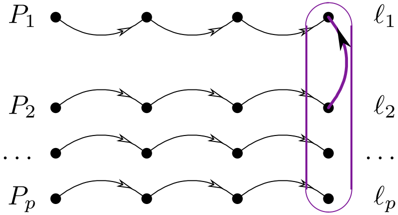
Theorem 5.18. (Gallai–Milgram)
Every digraph G contains a collection of pairw. disjoint paths $P_1, \ldots, P_p$ such that

a) $P_1, \ldots, P_p$ cover all the vertices, i.e. $P_1 \cup \cdots \cup P_p = V(G)$, and □

b) these paths have independent representatives, that is, a collection of pairwise nonadjacent vertices $R = \{r_i \in V(P_i) : i = 1, \ldots, p\}$.

Sketch. Clearly, some paths satisfying a) exist – take singleton vertices as the paths.□

The core idea is to take any such a covering collection of paths that the set of path ends $L = \{\ell_i \in V(P_i) \textit{ the end} : i = 1, \ldots, p\}$ is minimal by inclusion. . .



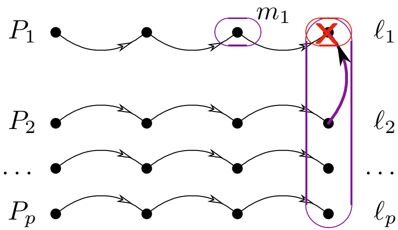
* pokrytí or. grafu cestami *

Claim : In a digraph G, let $P_1, \dots, P_p$ be a collection of pairw. disjoint paths covering all the vertices of G, such that the set of path ends $L = \{\ell_i \in V(P_i)$ *the end* $: i = 1, \dots, p\}$ is minimal by inclusion

- meaning that no other covering coll. of paths has the ends in a proper subset of L.

Then the paths $P_1, \dots, P_p$ have independent representatives. □

Proof: We would like to say that the set of ends L are the desired representatives.



By means of contradiction, assume there is an edge $e = \ell_2\ell_1 \subseteq L$. Then, using induction on $|V(G)|$, consider the subgraph $G \setminus \ell_1$ covered by the paths $P_1 - \ell_1, P_2, \dots, P_p$:□

- there exist such independent representatives by induction in $G \setminus \ell_1$, or □
- the new set of ends $L_1 = \{m_1, \ell_2, \dots, \ell_p\}$ is not minimal—some collection of paths covering $G \setminus \ell_1$ ends in a proper subset of L_1. □

The latter, though, contradicts original minimality of the ends L in G. □

5.5 Appendix: Counting objects in graphs

How many distinct objects (this time not disjoint) one can find in a given graph?

- (Trivial) Count the number of vertices, $|V(G)|$, and of edges, $|E(G)|$. □
- (Rather routine) How many triangles / 4-cycles are there in a graph G? □
- (???) Now, how many perfect matchings are there in a (say bipartite) graph G?

  Is this again a routine question (with a routine algorithmic answer)? □

  Surprisingly not! Even though we can easily decide the existence of a perfect matching, and to find one, we cannot find or count all of them.

  $\rightarrow$ This problem is "#P-hard". □
- Is there something else (nontrivial) which can be counted easily in graphs? Yes.

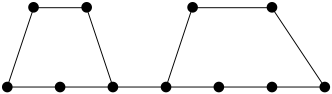
** počítání objektů v grafu **

Counting spanning trees

Recall a definition from Section 2.3:

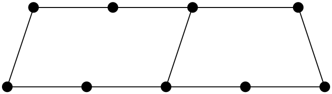
- A *spanning tree* of G is a subgraph T of G such that T is a tree and $V(T) = V(G)$. □

Example 5.19. *How many distinct spanning trees are contained in the following graph?*

 □

We argue as follows; which edges have to be removed from our graph to obtain a tree? Easily, one from each of the two cycles. Hence, by the principle of independent choices, we have got $5 \cdot 6 = 30$ spanning trees. □ □

Example 5.20. *How many distinct spanning trees are contained in the following graph?*

 □

Similarly as in 5.19, we may remove the "diagonal" edge and one more, giving 9 spanning trees. Or, we keep the "diagonal" and remove one edge from the left and one from the right cycles, giving additional $4 \cdot 5 = 20$ choices. Altogether $9 + 20 = 29$ spanning trees. □

Cayley formula and beyond

The following claim is considered a "beautiful gem" of graph theory.

Theorem 5.21. (Cayley)
The complete graph K_n for $n > 0$ contains exactly $\boldsymbol{n^{n-2}}$ distinct spanning trees.

Several nice and short proofs of this gem exist, see the textbook [Matoušek–Nešetřil]. . . □

Even more generally:

Definition. The *Laplace matrix* $\boldsymbol{Q} = (q_{ij})_{i,j=1}^n$ of an n-vertex graph G is defined:

- $q_{ii} = d_G(i)$ (vertex degree),
- $q_{ij} = 0$ for non-adjacent vertices $i \neq j$,
- $q_{ij} = -1$ for adjacent vertices $i \neq j$. □

Theorem 5.22. *Let $\boldsymbol{Q}$ be the Laplace matrix of a graph G, and let a matrix $\boldsymbol{Q}'$ result from $\boldsymbol{Q}$ by erasing the first row and the first column. Then the number of distinct spanning trees in G equals the determinant $|\boldsymbol{Q}'|$.* □

Corollary 5.23. *One can count the spanning trees in any graph in polynomial time.*

A proof of Thm. 5.22 is, however, quite difficult and using advanced linear algebra.