Sources

Diestel's Graph theory: https://diestel-graph-theory.com

As we have seen, the problem 2-COL is in the class P, while 3-COL is NP-complete.

Exercise 1. *Give a reduction of 3-COL to 4-COL (or any k-COL) to show that k-COL is NP complete for $k \geq 3$.*

Last week, we showed that 3-COL also reduces to SAT. In general, we won't need to do this. SAT was the original NP-complete problem by the Cook-Levin Theorem, which states that every problem in NP reduces to SAT. This theorem is due to combined effort of Stephen Cook, Leonid Levin, and Richard Karp, the latter two of whom received Turing Awards for this work.

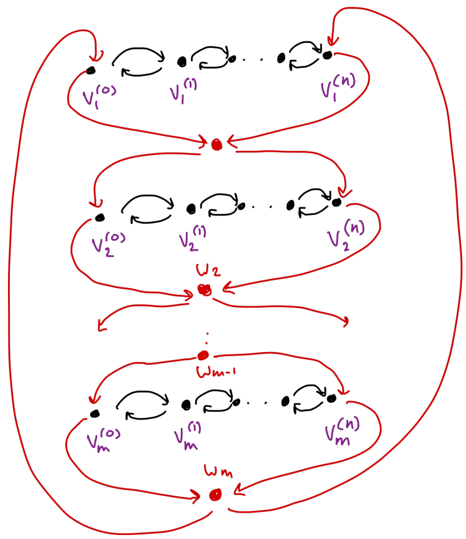
Eulerian circuits and Hamilton cycles

This week we will discuss a few more famous graph problems. We start with the problem of Eulerian circuits and Hamilton cycles, which on the surface look like quite similar problems. A *walk* of length k in a graph is a sequence of vertices $v_0, \ldots, v_k \in V(G)$ such that $v_{i-1}v_i \in E(G)$ for all $1 \leq i \leq k$. We allow repetition of vertices (and edges). An *Eulerian circuit* of a graph is a walk that visits each edge of G exactly once, and whose start- and endpoint are equal ($v_0 = v_k$). In some sense this graph problem is the first graph problem, as it was proposed by Euler in the Seven Bridges of Konigsberg problem in 1736, laying the foundation for the study of graph theory (and topology). In class, we show that the decision problem of deciding whether a graph has an Eulerian circuit is in P.

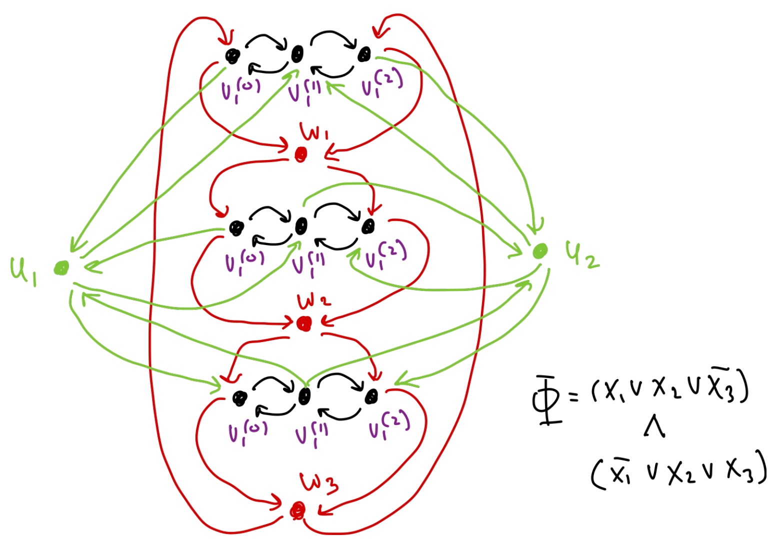
A *Hamiltonian cycle* in a graph is a spanning cycle (it sees all the vertices). This problem is named after William Hamilton, who in 1858 proposed the problem of finding a Hamilton cycle in the graph of a dodecahedron (also known as Hamilton's Puzzle). At the time, he solved this problem algebraically, with a method that does not generalize to other graphs. It is not difficult to see that the decision problem of whether a graph has a Hamilton cycle, or "is Hamiltonian", is in NP. There are various known sufficient conditions for hamiltonicity which are efficient to check, but we will now show that the problem is NP-complete. Again, we will do this by reducing SAT to hamiltonicity. We will take two steps, first reduce 3-SAT to directed hamiltonicity, and then directed to undirected.

Theorem 1. *Directed hamiltonicity is NP-complete.*

Proof. As mentioned, this problem is clearly in NP, and we complete the proof by showing a reduction from 3-SAT to directed hamiltonicity. Once again, this construction is well-known and I am not sure who to credit it to. For a given 3-SAT formula Φ in CNF, we construct a directed graph G so that Φ is satisfiable if and only if G has a directed Hamilton cycle. It is not difficult to see that this construction takes polynomial time. Let $C_1, \ldots, C_n$ be the clauses of Φ and $x_1, \ldots, x_m$ the literals. For each literal x_i, we build a bidirected path $v_i^{(0)}, \ldots, v_i^{(n)}$. We add connector vertices $w_1, \ldots, w_m$ and edges $v_i^{(0)} \to w_i$, $v_i^{(n)} \to w_i$, $w_i \to v_{i+1 \pmod n}^{(0)}$ and $w_i \to v_{i+1 \pmod n}^{(n)}$, for $1 \leq i \leq m$. Now our graph looks like this:



First, verify that this graph has exactly one Hamilton cycle for each True/False assignment of the literals: we'll say that if a Hamilton cycle contains the directed path $v_i^{(0)} \to v_i^{(1)} \to \cdots \to v_i^{(n)}$ the literal is set to True, and if it contains the directed path $v_i^{(n)} \to v_i^{(n-1)} \to \cdots \to v_i^{(0)}$ the literal is set to False. Now, we need to add the constraints given by the clauses of Phi. For each clause C_i, we add a vertex u_i. If the clause contains the (non-negated) literal x_j, we add the edges $x_j^{(i-1)} \to u_i$ and $u_i \to x_j^{(i)}$. If the clause contains the literal $\overline{x_j}$, we add the edges $x_j^{(i)} \to u_i$ and $u_i \to x_j^{(i-1)}$. For example, here is the full graph for the formula $\Phi = (x_1 \wedge x_2 \wedge \overline{x_3}) \wedge (\overline{x_1} \wedge x_2 \wedge x_3)$ looks like this:



To show that this construction works, we must show that if Φ is satisfiable, then this graph G has a Hamilton cycle. Given True/False assignment we construct the cycle as follows: start with the cycle of the form

$$\left\{ \begin{array}{c} v_1^{(0)} \to \cdots \to v_1^{(n)} \\ v_1^{(n)} \to \cdots \to v_1^{(0)} \end{array} \right\} \to w_1 \to \left\{ \begin{array}{c} v_2^{(0)} \to \cdots \to v_2^{(n)} \\ v_2^{(n)} \to \cdots \to v_2^{(0)} \end{array} \right\} \to w_2 \to \cdots \to \left\{ \begin{array}{c} v_m^{(0)} \to \cdots \to v_m^{(n)} \\ v_m^{(n)} \to \cdots \to v_m^{(0)} \end{array} \right\}$$

where each choice in the curly brackets depends on the True/False assignment of x_i as described before. Since this assignment is satisfying, each clause C_i contains a literal x_j or its negation that is True. If this is x_j, we replace the edge $v_j^{(i-1)} \to v_j^{(i)}$ by the path $v_j^{(i-1)} \to u_i \to v_j^{(i)}$, and in the opposite direction if this is $\overline{x_j}$.
Now, we must show that if Φ is not satisfied, then G does not contain a Hamilton cycle. $\square$

Now, we reduce the directed Hamilton cycle problem to the undirected Hamilton cycle problem. We will start this proof in class, and you should finish it on your own.

Exercise 2. *Complete the reduction from the directed Hamilton cycle problem to the undirected Hamilton cycle problem.*

Exercise 3. *A* Hamilton path *is a spanning path. Give a (polynomial time) reduction from the Hamilton cycle to the Hamilton path problem. Then, give an explicit reduction from the Hamilton cycle to the Hamilton path problem. (Note that the latter is not necessary for classification, we have already shown that both problems are NP-complete.)*

1 Independent sets and matching

An *independent set* in a graph G is a set of vertices $I \subseteq V(G)$ such that $vw \notin E(G)$ for all $v, w \in I$. A *matching* in G is a set of edges $M \subseteq E(G)$ such that $e \cap f = \emptyset$ for all $e, f \in M$ (no two edges in M share an endpoint). We will start with the latter. Consider the decision problem: does a given graph G contain a matching of cardinality k? Clearly, this problem is in NP.

Theorem 2. *The matching problem is in P.*

We will focus on the problem of finding maximum matchings. If we can find a matching of maximum size in polynomial time, that gives the answer to the decision problem. Given a graph G and a matching M in G, let an *alternating path (with respect to M)* be a path whose edges alternate being in M and not being in M. We say that a vertex is *unmatched* if none of its incident edges are in M. We say that an alternating path is an *augmenting path* if both of its endpoints are unmatched vertices (in G, not just within the path). We prove the following claim in class:

Claim 3. *If G contains an augmenting path with respect to a matching M, then there exists a matching M' in G such that $|M'| > |M|$.*

Now, we show that the converse is also true, i.e. M is a maximum matching if and only if there are no augmenting paths with respect to M. This boils the problem of finding a maximum matching down to repeatedly checking for augmenting paths, augmenting the matching, and repeat until there are no augmenting paths left.

Claim 4. *If a matching M is not maximum, then there exists an augmenting path with respect to M.*

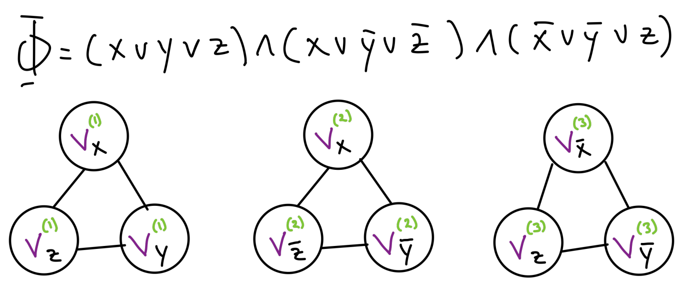
Proof. Suppose that M is not a maximum matching, and let M' be a matching such that $|M'| > |M|$. Consider the graph G' induced by $M \cup M'$. In G', vertices have degree either 1 or 2, and therefore G' consists of a disjoint union of paths and cycles. First note that the cycles in G' cannot be of odd length, as they must alternate edges in M and M', and therefore they each must contain the same number of edges in each of the two matchings. Since $|M'| > |M|$, G' must therefore also contain paths. In particular, it must contain a path of odd length, which starts and ends with an edge in M'. Such a path is an augmenting path with respect to M. □

Exercise 4. *Write an explicit, polynomial time, algorithm to find augmenting paths, given a graph G and matching M.*

Now, we turn out attention to the independent set problem. Consider the decision problem: does a given graph G contain an independent set of cardinality k? This time, we prove that there is a polynomial time reduction from 3-SAT to INDEPENDENT-SET. Again, given a boolean formula Φ, we construct a graph G such that G has an independent set of cardinality k if and only if Φ is satisfiable.

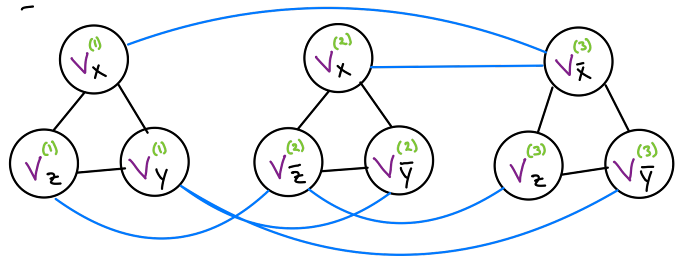
Theorem 5. *The problem INDEPENDENT-SET is NP-complete.*

Proof. Let Φ be a boolean 3-SAT formula, with literals $x_1, \dots, x_m$ and clauses $C_1, \dots, C_n$. This time, we will let vertices correspond to instances of each literal (or its negation) in each clause, and a literal is considered True if the vertex is in an independent set. Then, an independent set corresponding to a satisfying assignment should have cardinality at least n. Vice versa, we need an independent set of cardinality n to hit every clause.
We start with the following base for our graph. For each clause C_i, $(x \wedge y \wedge z)$, add a triangle on vertices $v_x^(i), v_y^(i), v_z^(i)$. Now, the independence number of our graph cannot exceed n, and any independent set of cardinality n has exactly one vertex in each of the triangles.



The above graph has 3^n independent sets, as one can include any vertex from each triangle. You might raise two issues: there may be contradictions, as a literal x (or its negation) can be included in the independent set in one triangle and not in another. Additionally, a literal and its negation could both be included if they appear in different triangles. We will see that the first of these is not a problem. If a literal is included in the independent set at least once we will consider it True. To avoid inclusion of both a literal and its negation, for each literal x we add all edges of the form $v_x^{(i)} v_{\overline{x}}^(j)$, ensuring that either the literal or its negation can ever be in an independent set, never both.

$$\Phi = (x \vee y \vee z) \wedge (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y} \vee z)$$



What if some independent set of cardinality n contains $v_x^{(i)}$, but not $v_x^{(j)}$ for $i \neq j$? In that case, there must be some other vertex $v_y^{(j)}$ in the set, which is interchangeable with $v_x^{(j)}$. This just means that clause C_j contains multiple literals that are True. Vice versa, a satisfying assignment is translated to an independent set by choosing exactly one True literal from each clause arbitrarily. Noting that the graph G is constructed in $O(n^2)$ time, this completes the proof.

□

Exercise 5. *A* vertex cover *in a graph G is a set of vertices S such that every edge has an endpoint in S. In bipartite graphs, the problem of finding a minimum vertex cover is equivalent to finding a maximum independent set. See König's Theorem in Section 2.1 in Diestel. Therefore, for bipartite graphs the vertex cover decision problem is in P. Show that on general graphs, the vertex cover problem is in NP, by reducing from INDEPENDENT-SET.*