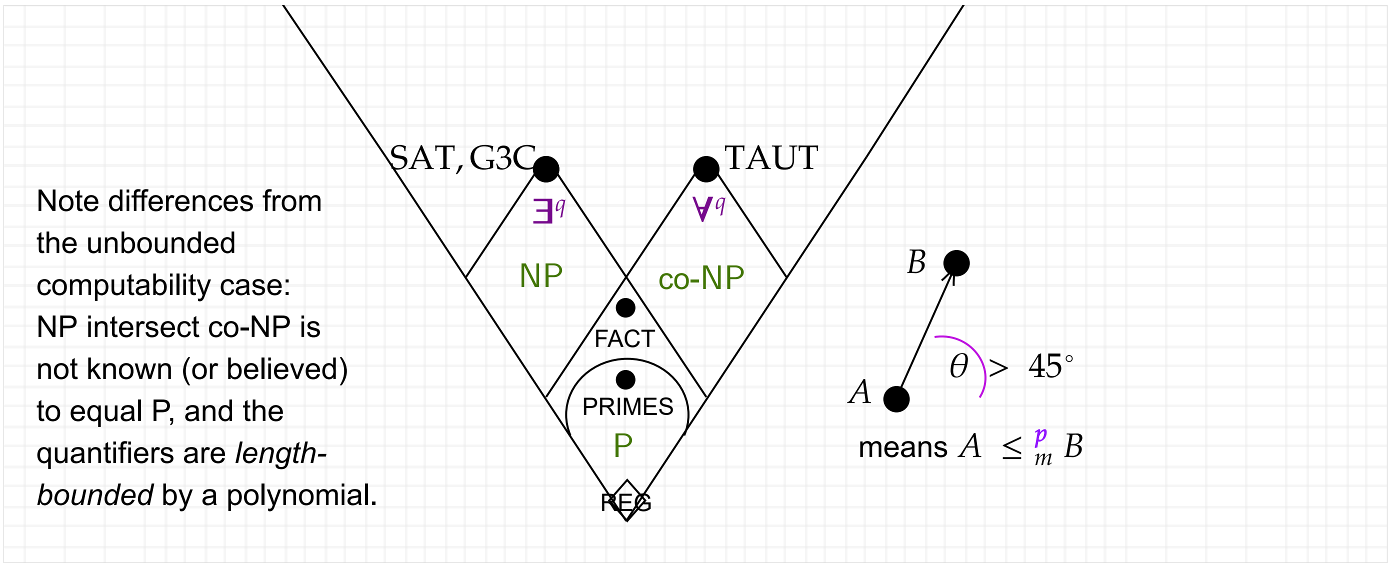
CSE491/596 Lecture Friday 10/23: Problems in NP and Poly-Time Reductions

To repeat the picture at the end of Wednesday's lecture, adding one more language:

TAUT:
Instance: A Boolean formula ϕ', same as for SAT.
Question: Is ϕ' a **tautology**, that is, true for all assignments?

Note that ϕ is unsatisfiable $\equiv$ every assignment a makes $\phi(a)$ false $\iff$ every assignment a makes $\phi'(a)$ true, where $\phi' = \neg\phi$. Thus TAUT is essentially the complement of SAT.



Note differences from the unbounded computability case: NP intersect co-NP is not known (or believed) to equal P, and the quantifiers are *length-bounded* by a polynomial.

PRIMES $= \{2, 3, 5, 7, 11, 13, 17, 19, 23, \dots\}$ (encoded as, say, $10, 11, 101, 111, 1011, \dots$)

This language was formally shown to belong to P only in 2004, but had long been known to be "almost there" in numerous senses.

FACT:
Instance: An integer N and an integer k.
Question: Does N have a prime factor p such that $p \leq k$?

If you can always answer yes/no in polynomial time $r(n)$, where $n \approx \log_2 N$ is the number of bits in N, then you can do *binary search* to *find* a factor p of N in time $O(nr(n))$. By doing $N' = n/p$ and repeating you can get the complete factorization of N in polynomial time. This is something that the human race currently does **not** want us to be able to do, as it would (more than Covid?) "destroy the world economy" by shredding the basket in which most of our security eggs are still placed. But to indicate proximity to this peril, we note:

FACT: **FACT** is in NP $\cap$ co-NP.

Proof: Suppose the answer to an instance $\langle N, k\rangle$ is *yes*. We can verify it by guessing the **unique prime factorization** (u.p.f.) of N as $N = p_1^{a_1} p_2^{a_2} \cdots p_\ell^{a_\ell}$. Although the right-hand side may seem long, ℓ cannot be bigger than the number of bits of N in binary because each p_i is at least 2, and bigger powers only make ℓ have to be smaller. The length of the u.p.f. is $O(n)$. To verify it, one must verify that each p_i is prime---but this is in polynomial time as above---and then simply multiply everything together and check that the result is N. Finally to verify the *yes* answer, check that at least one of the p_i is $\leq k$.

Now suppose the answer to an instance $\langle N, k\rangle$ is *no*. We can verify it by guessing the **unique prime factorization** (u.p.f.) of N as $N = p_1^{a_1} p_2^{a_2} \cdots p_\ell^{a_\ell}$. Although the right-hand side may seem long, ℓ cannot be bigger than the number of bits of N in binary because each p_i is at least 2, and bigger powers only make ℓ have to be smaller. The length of the u.p.f. is $O(n)$. To verify it, one must verify that each p_i is prime---but this is in polynomial time as above---and then simply multiply everything together and check that the result is N. Finally to verify the *no* answer, check that none of the p_i is $\leq k$.

Thus we can verify both the *yes* and *no* cases (with the same witness!), so both the language and its complement belong to NP. ☒

This makes the contrast to RE $\cap$ co-RE $=$ REC all the more important. Of course, we don't know NP $\neq$ P either, in contrast to RE $\neq$ REC. What restores much of the analogy is the similarity under reductions and having complete problems. We've seen what comes next already:

Definition: $A \leq_m^p B$ if there is a function $f : \Sigma^* \to \Sigma^*$ that is computable in polynomial time such that for all $x \in \Sigma^*$, $x \in A \iff f(x) \in B$.

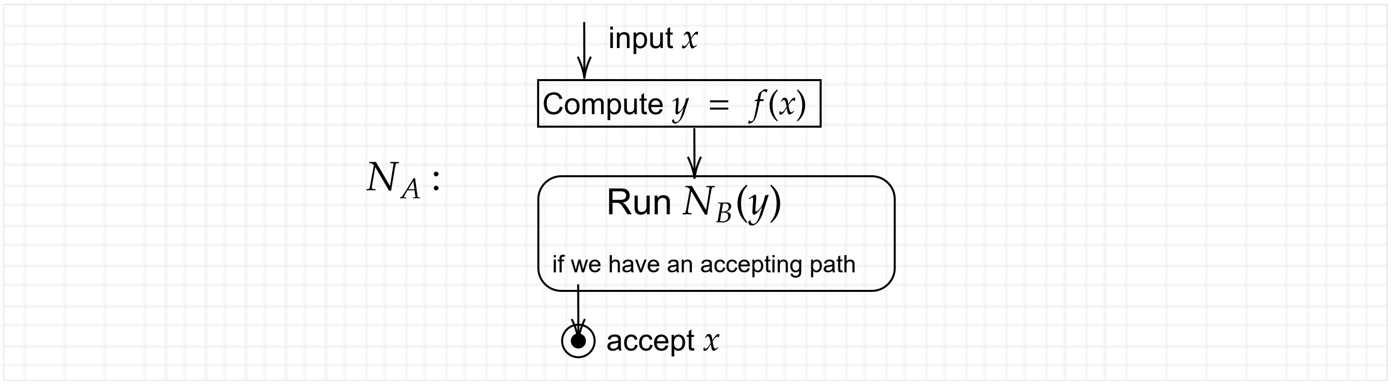
This is sometimes called a "Karp reduction" after Richard M. Karp but saying polynomial-time mapping reduction (or many-one reduction) is clear. (There is a corresponding notion called "Cook reduction" after Stephen Cook that uses oracles, but let's *ignore* it for now.)

Theorem: Suppose $A \leq_m^p B$. Then:

(a) $B \in$ P $\implies A \in$ P. So $A \notin$ P $\implies B \notin$ P.
(b) $B \in$ NP $\implies A \in$ NP. So $A \notin$ NP $\implies B \notin$ NP.
(c) $B \in$ co-NP $\implies A \in$ co-NP. So $A \notin$ co-NP $\implies B \notin$ co-NP.

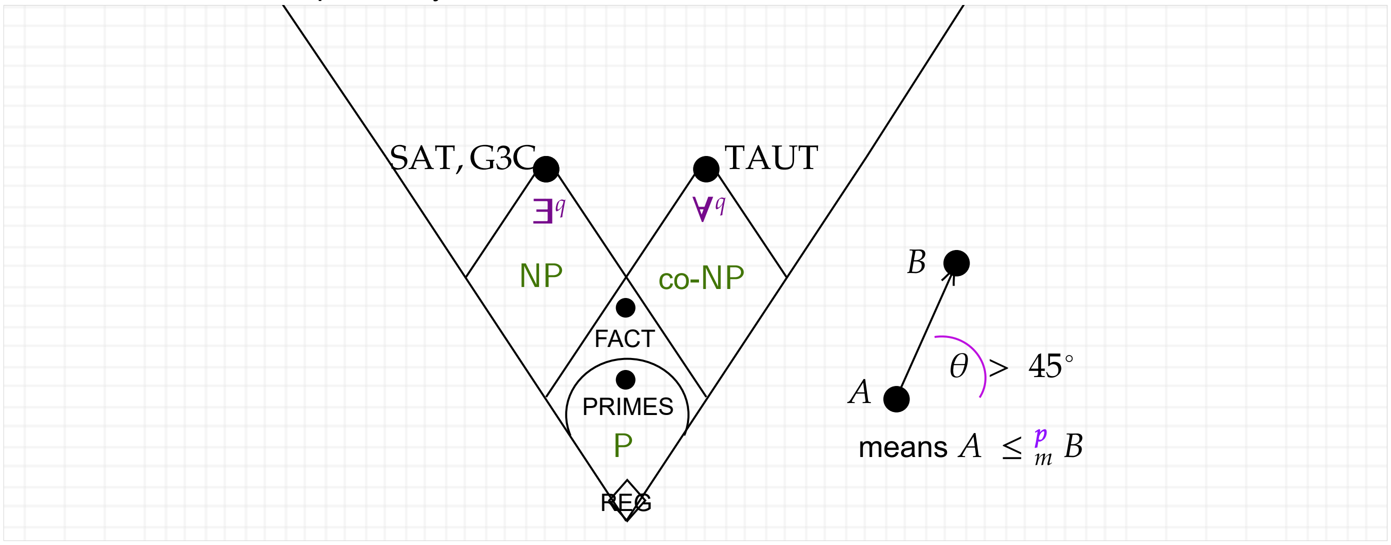
The proof is similar to the one with REC and RE and co-RE : We take a machine M_B whose language is B and the reduction function f and create the machine M_A that on any input x computes $y = f(x)$ and runs $M_B(y)$, accepting x if and when M_B accepts y. There are two particular details:

- The composition of two polynomials p and q is a polynomial. Thus if f is computable in $p(n)$ tile, then it follows in particular that $|y| \le p(|x|)$. So if M_B runs in $q(m)$ time, then $M_A(x)$ takes at most $q(p(|x|))$ time, which is a polynomial in $n = |x|$. This shows (a).
- The mapping and timing works in (b) with a polynomial-time NTM N_B in place of a DTM M_B. In that case we get a polynomial-time NTM N_A, which is what we need for $A \in$ NP.



Part (c) again follows simply because $x \in A \iff f(x) \in B$ is the same as $x \notin A \iff f(x) \notin B$. This also means that NP $\cap$ co-NP is likewise **closed downward under** $\le_m^p$.

This is all summed up visually in the "cone diagram"---except that we don't know if the lines are definite because NP $=$ P is a possibility.



There is one other "grain of salt" that must be taken with all these diagrams: If A and B are two languages in P (technically, other than the languages $\emptyset$ or Σ^* but we sometimes ignore this distinction), then automatically $A \equiv_m^p B$ (this is a good self-study exercise, including why we have the technicality). Thus to keep up the geometrical intuition of a steep angle meaning $A \le_m^p B$, we would have to warp the diagram so that P is a single point---squshed even more than how the above shows REG as a tiny subclass of P.

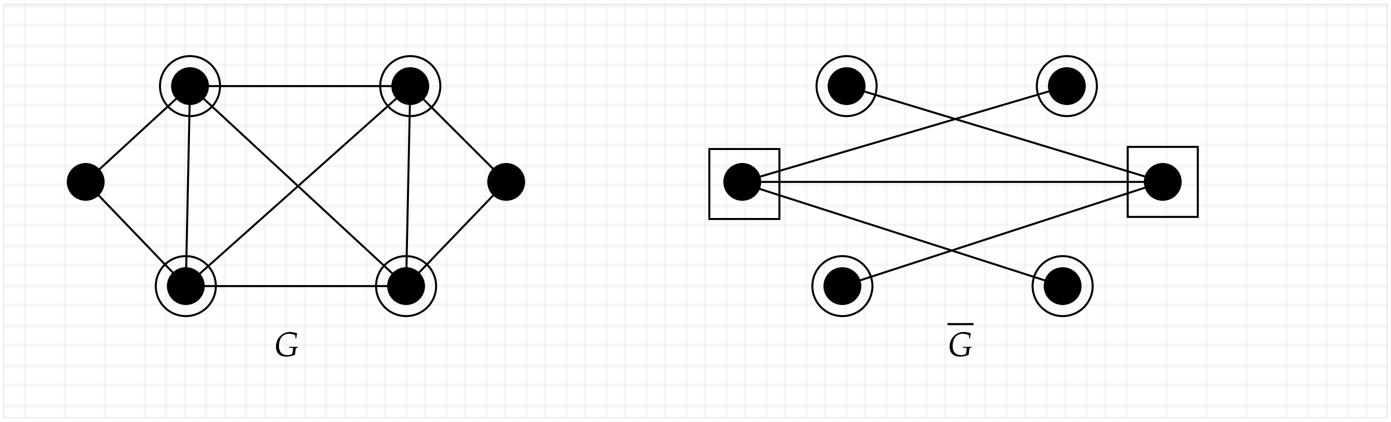
Up at the top of NP (and hence also the top of co-NP) we will get a lot of more meaningful reduction equivalence thanks to completeness. Before tackling Cook's Theorem on the NP-completeness of SAT, let's see some simpler examples. Consider these decision problems:

CLIQUE
Instance: An undirected graph $G = (V, E)$ and a number $k \geq 1$.
Question: Does there exist a set $S \subseteq V$ of k (or more) nodes such that for each pair $u, v \in S$, (u, v) is an edge in E?

INDEPENDENT SET
Instance: An undirected graph $G = (V, E)$ and a number $k \geq 1$.
Question: Does there exist a set $S \subseteq V$ of k (or more) nodes such that for each pair $u, v \in S$, (u, v) is **not** an edge in E?

Important to keep straight: The languages of these problems are *not* complements of each other, despite their differing by just the word "not" at the end. Both languages are in NP with S as the witness. An important point is that with $n = |V|$, there are 2^n subsets S that might have to be considered. A polynomial-time algorithm cannot try each one. Within S, however, there are at most n^2 pairs (u, v) that have to be considered. Those can all be iterated through to check the body of the condition in quadratic time, so it becomes a polynomial-time decidable predicate $R(G, S)$. It is not even true that this predicate gets negated between the two languages, because it includes the "for each" part. It is because this runs over only polynomially-many pairs that I suggest the convention of saying "for each" rather than "for all" there. What actually gets complemented *is the graph* G, as expressed by this fact:

G has a clique of size k $\iff$ the complementary graph $\overline{G}$ has an independent set of size k.



Therefore, the simple reduction function $f(G, k) = (\overline{G}, k)$ reduces CLIQUE to IND SET and also vice-versa, so the problems are $\equiv_m^p$ equivalent. [Note that this skips writing the angle brackets around

$\langle G, k \rangle$; by now that's AOK.] A second fact yields a second equivalence:

The complement of an independent set S in G is a set S' of nodes such that every edge involves a node in S'. Such an S' is called (somewhat midleadingly, IMHO) a **vertex cover**. Therefore:

$$G \text{ has an independent set of size (at least) } k \iff G \text{ has a vertex cover of size (at most) } n-k.$$

Note that the graph G stays the same; instead we flip around the target number from k nodes to $|V|-k$ nodes. In practice, when we're trying to optimize, we want to *maximize* cliques and independent sets and *minimize* vertex covers. The latter gives rise to this decision problem:

VERTEX COVER (VC)
Instance: A graph G and a number $\ell \geq 1$.
Question: Does G have a vertex cover of size (at most) ℓ?

Then IND SET and VC reduce to each other via the reduction $g(G,k) = (G, n-k)$ (where it is understood that $G = (V,E)$ and $n = |V|$.)

[Next: The NP-completeness of SAT, followed by reductions from SAT to all these problems (by transitivity, reducing to IND SET will be enough). Now is the time to read the Allender-Loui-Regan notes: first Chapter 27 but skim/skipping all the proofs, ignoring "alternation", and stopping after the first two pages on circuits. Then read the first four sections of Chapter 28.]