Recall we were considering the following problem: A league of teams is playing a series of matches, and we assume that the “better” team ALWAYS beats the “worse” team (we do not know the ranking of teams in advance). Given the schedule, we wanted to know how many different match results were possible.

Examples:

- If the only two matches are \((a, b)\) and \((b, c)\), any pair of results are possible, so there are 4 total results.
- If the three matches are \((a, b)\), \((b, c)\), and \((c, a)\), then at first glance you might assume there to be 8 possible match results (2 choices of winner for each game, so \(2^3\) total). However, the results “\(a\) beats \(b\) beats \(c\) beats \(a\)” and “\(b\) beats \(a\) beats \(c\) beats \(b\)” are impossible, so there’s actually only 6.
- More generally, if every team plays every other team, then what will happen will be that one “best” team wins all their games, one “second best” team beats every team but the best team, and so on. There are \(n!\) ways of ordering the \(n\) teams from best to worst, so \(n!\) possible league results.

We can turn this into a graph problem by connecting two teams if they play each other at some point in the season. What Richard Stanley proved in his 1973 paper “acyclic orientations of graphs”[1] is the following:

**Theorem 0.1.** The total number of match results is \(P(G, -1)(-1)^{|V(G)|} = |P(G, -1)|\).

You can prove this result by induction. A rough sketch as follows: Consider the last game played in the season (say it’s between \(a\) and \(b\)). Once all the remaining games have been played, there are two possibilities:

One is that you already know which team of \(a\) and \(b\) is better, as you’ve already seen a series of other games where \(a\) beat \(c\) who beat \(d\) who ... who beat \(b\). In this case there’s only one possible result for the last game.

The other is that no such path exists, and the last game can go either way. The key observation is this: If the results so far aren’t enough to tell whether team \(a\) or team \(b\) are better, than they are for all intents and purposes the same team. We could combine the results and say that there’s a new team \(a/b\). Any team which beat either \(a\) or \(b\) beat \(a/b\), and any team which lost to either \(a\) or \(b\) lost to \(a/b\). Doing this can’t lead to an inconsistent series of results: If it did, we could use them to tell whether \(a\) or \(b\) was a better team.

Now let’s put this in Graph theoretic terms. Let \(\omega(G)\) be the number of match results consistent with a graph \(G\). A consistent match result corresponds to an assignment of a direction to every edge in the graph so that no cycles are formed. Ignoring the last game corresponds to considering \(G_e\) instead of \(G\). Every assignment of directions for \(G_e\) corresponds to at least one successful assignment of directions for \(G\). However, there are some graphs where the last edge can go either way. Those correspond exactly to the cases we can combine the two teams, that is
to say the ways of assigning directions for \( G'_e \). In other words

\[
\omega(G) = \omega(G_e) + \omega(G'_e)
\]

On the other hand, we know that

\[
P(G, \lambda) = P(G_e, \lambda) - P(G'_e, \lambda)
\]

Multiplying by \((-1)^{|V(G)|} = (-1)^{|V(G_e)|} = (-1)^{|V(G'_e)|} + 1\), we see that

\[
(-1)^{|V(G)|} P(G, \lambda) = (-1)^{|V(G_e)|} P(G_e, \lambda) + (-1)^{|V(G'_e)|} P(G'_e, \lambda)
\]

In other words, \( \omega(G) \) and \((-1)^{|V(G)|} P(G, \lambda)\) satisfy the same recursion, EXCEPT for initial conditions. If we can match the initial conditions up, we’ve got that they’re actually the same function.

The initial conditions for our chromatic polynomial recursion were the graphs without any edges. If a graph has \( n \) vertices and no edges, there are \( \lambda^n \) ways to color the graph, so

\[
(-1)^{|V(G)|} P(G, \lambda) = (-1)^n \lambda^n
\]

In terms of our tournament, an empty graph corresponds to a tournament with no games, for which there is exactly one result. It follows that \( \omega(G) = 1 \). Comparing the two, we see that if \( \lambda = -1 \) the initial conditions match, so by induction the polynomials must match on every graph.

References