In this assignment I tried to investigate the network of the ATP’s (Association of Tennis Professionals) top 100 ranked players. In all of the graphs below, I used the vertices to represent the players (one hundred vertices). The binary relation I used in order to create different types of edges was the players’ career head to head statistics, meaning how many matches each player has won/lost against each of the other players in the list. I acquired the data from [1] and processed it using python, and sqlite3, to get it in the proper format for use in NodeXL.

1 Is the greatest player of all times (Federer) equivalent to Jan-Lennard Struff (who?)

In this example I drew an edge between every two players for which their results are almost tied. By almost tied I mean that $\frac{|\text{Won} - \text{Lost}|}{\text{Games Played}} \leq 0.1$. Then I grouped the nodes according to edge clusters, and removed edges between clusters. Then I labeled the nodes by their ATP ranking. The title is somewhat misleading since the head to head statistics "equality" relation is not transitive, therefore not an equivalence relation, but the idea is clear.

The results seem quite astounding. A priori, I would have guessed that players ranked close together, and therefore probably of the same quality, will be almost tied, and therefore in same clusters, while players ranked far apart from each other would be in different clusters, since probably when meeting a lower ranked player, the better player beat his opponent most times. To my surprise, I found two interesting results. The first was that the graph had just one connectivity component, which means that every player has at least one player which he is quite equal to in head to head, and in turn he is "equal" to another and so on. The second, some of the best players of all time were ranked with mediocre players. For example players ranked number 1 (Rafael Nadal), and number 3 (David Ferrer) were clustered in the same group as players ranked 97, 87, 74, etc. Same goes for number 2 and 7 (Novak Djokovic, and Roger Federer respectively), who are clustered with world numbers 100, 99 (Jan-Lennard Struff), 95, and similar.
2 Players Dominance is an "almost transitive" relation

In this directed graph I drew an edge from player A to player B, if and only if player A has an outstanding record against player B. Namely, if he won at least 90% of their matches. I denoted this relation as player A dominates player B. Obviously, by construction there cannot be any cycles of exactly two nodes, but there is no guarantee that there are no cycles at all. I laid the graph out in a topological ordering. The size of the nodes is decided according to their out-degree, meaning the number of players they are dominant against.
The results are rather as expected. It is clear that the graph ended up acyclic. Also I managed to put it in topologically order. One can clearly see a list of players (on the left side of the image) that are not dominated by any other player, meaning no player has a significant head to head advantage against them. Not surprisingly we can see most of the top ranked players there, such as Rafael Nadal, Novak Djokovic, Andy Murray, and Roger Federer. I was surprised on the other hand to see some lower ranked players in that list, such as 62 ranked Lleyton Hewitt, which might make sense being a former top ranked player, and 68 ranked Kukushkin Michael, which I find quite remarkable. On the right hand side of the image the two rightmost columns are players that are dominated by at least one player, but do not dominate any other player. Not surprisingly these are mostly lower ranked players, but again I was surprised to see a relatively high ranked player like number 18, and former number 5, Tommy Robredo.

In the headline above I used the term "almost transitive" because A dominating B, and B dominating C, does not yield A dominating C, only that C is not dominating A, for every A,B,C in our players list.

3 Biggest rivalries are among old top players

In this graph, I explored the biggest rivalries among the current top 100 players. I drew edges between every pair of players that played at least twelve matches against each other. Obviously this is an undirected graph, with relatively few vertices and edges. To emphasize players with many such rivals, I set the size of the vertex to be proportional to it's degree.
As we can see, most players to have such rivalries are top ranked players, which makes sense, because usually rivalries are created among players that consistently reach the final rounds (it is hard to meet the same players over and over, in stages where there are 64 players, while it gets easier in stages where there are 8 players or less, and both players consistently reach these stages). Another factor that can affect this parameter is the players' age. This can be seen in players that appear on that list even though they are not top ranked. Players such as the 53rd ranked and 32 years old Nikolay Davydenko, or the 62 seed and 32 years old Lleyton Hewitt. Taking both factors into account (top ranked, and age), leads to the immediate result that the player with the most such rivalries is 7 ranked Roger Federer, which is both 32 years old, and top ranked through almost his entire career.

4 NodeXL

I used NodeXL[2] to analyze the network. Generally speaking I didn’t like using NodeXL. Though it is quite powerful, I think it is missing some key features I would expect in a tool like that, and I believe it was generally not easy and intuitive to learn and use.

4.1 Pros

The fact that NodeXL is built on Excel framework makes it accessible to almost any Windows PC user, and easy to install. Another major upside for using NodeXL is it’s very easy built-in import tools from several types of networks such as Twitter, emails, youtube, etc. (which I ended up not
using, but tried it at first before finally ending up with the ATP network). Even when I imported
the data from a manually created CSV file, it very quickly figured the vertices names out of the
edges table, which was very handy. Another thing I liked was the way it allows the user to use
certain calculated graph metrics (such as node’s degree) as visual properties (such as node’s size or
color).

4.2 Cons

First, I believe the biggest disadvantage of this tool is that it wasn’t ported to different platforms,
and most significantly to OS-X which already has running Excel versions. Another big problem
I had working with NodeXL was that it didn’t allow any action to be undone. Specifically when
dragging several vertices to wrong places, caused some cluttering that was hard to unclutter, while
redoing the layout undid anything I already arranged on screen. Also there were several important
features I was hoping to see, and couldn’t find in NodeXL, such as: finding cycles, topological
ordering (sorting), tree layout (if no cycles). Also I thought it was annoying that changes made
were not reflected immediately on the graph, but required clicking "Refresh Graph" or re-calculating
the graph metrics, and so on. I believe this was done intentionally, since redrawing of very large
graph can take awhile, and might not be wanted after every minor change, but it should at least
offer this option for small-medium graphs.

4.3 Suggestions

As stated in the Cons section, I believe that the most important suggestion would be to port NodeXL
to other platform. Also implementing a few well known and solved algorithms like topological order,
and tree layout, would be very important as well. Other than these two, I believe that the help
utility, which wasn’t very helpful for me, would be the next most important thing.

References
