CprE 310 / Spring 2019

Lecture 7: Intro to Relational Thinking

We have discussed a few different proof techniques for proving implications of the form $p \implies q$, namely:

- Direct proofs (assume p is true, show that q is true)
- Proof by contraposition (assume q is false, show that p is false)
- Trivial proofs
- Vacuous proofs
- Proofs by counterexample
- Proof by cases

We will be using the above proof techniques over and over again throughout 310, so it may be useful to understand the different techniques as well as you can! However, let us depart from the standard type of *mathematical* proofs, and take a step towards the type of mathematical proofs that are more common in computer science and engineering.

Suppose we were to draw parallels between mathematical concepts and software concepts. As discussed before, propositions and predicates are analogous to *variables* and *functions*. But how about more complex, higher level concepts such as connectivity, modularity, and dependencies? To understand these formally, we need a new language – that of *graph theory*.

Graph theory

Today's lecture will involve a somewhat informal introduction, emphasizing applications of graphs. The next few lectures will revolve around building the mathematics of graph analysis.

Graphs, of course, are one of the most important objects of study in discrete mathematics. Graphs can be used to study problems ranging from computer networks, to social networks, to the Web, to transportation networks, to biological networks etc etc etc. Suffices to say that applications of graph-based algorithms impact all of engineering.

Definitions

A graph G is a collections of *nodes*, together with a set of *edges* between nodes *E*.

Edges can be either equipped with a direction (arrow) or be without any directionality. Depending on this, the graph G is called *directed* or *undirected*.

The *degree* of a node in a simple graph is the number of edges touching that node.

In a directed graph, there are two notions of degree for each node: *in-degrees* (number of incoming arrows) and the *out-degree* (number of outgoing arrows). In an undirected graph there is only one notion of degree since edges do not have arrows.

A *path* in an undirected graph is a sequence of edges that connect a sequence of vertices that are distinct from one another.

Let us illustrate these ideas in a concrete example. Consider a simple social network with 9 persons, some of which are friends with each other. We represent the people as nodes, and friendship between people as edges. An example such network is given by Figure 1.

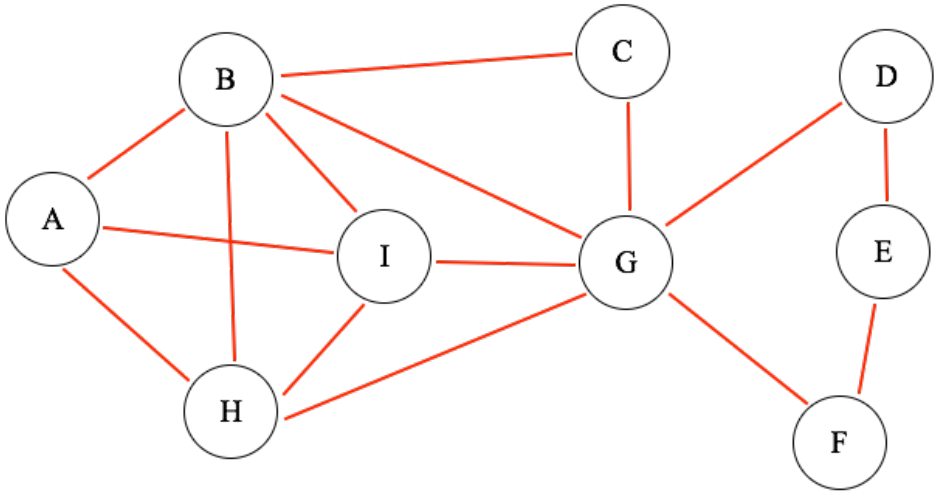

Figure 1: Social network

Given this network, let's try to answer some natural questions using the tools of graph theory. Here is the first question:

> A *clique* is a group of friends who all mutually know each other. What is the largest clique in the graph?

We quickly see that there are several triangles (i.e., cliques of size 3) and one group of 4 - $\{A, B, H, I\}$ - which is a clique of size 4. In fact, this is the largest clique in the graph. But how do we *prove* this fact?

We prove that there is no clique of size 5, via contradiction. Suppose, to the contrary, that there exists a clique of size 5. That means that there is a group of 5 people, each of which knows the four other people in the group. Therefore, the degree of each node corresponding to these 5 people is *at least 4*. However, if one enumerates all the degrees in the graph, we get:

- One node of degree 6 (G)
- One node of degree 5 (B)
- Two nodes of degree 4 (H,I)
- One node of degree 3 (A)
- Four nodes of degree 2 (C,D,E,F)

Therefore, it is not true that there are 5 nodes of degree 4. By contradiction, there is is no clique of size 5.

Next question:

> Who is the most "centrally connected" person in the social network?

This is a more qualitative question; however, a natural notion of "central connectivity", again, is the degree of each node. By the above enumeration, G has the highest degree, and therefore, is the most "centrally connected" person.

Last question:

> Suppose the people in the network continue to interact in parties and social events, i.e., they get acquainted with friends of friends, etc. Which two currently unacquainted people are (a) most likely to become acquainted? (b) least likely to become acquainted?

This is a more challenging question. But again, one can solve it using graph theory tools. The trick is to realize that "likeliness of X and Y becoming acquainted" is related to "are there many short paths between X and Y".

Again, by inspection, we see that there are 3 distinct paths of length 2 from A to G (and that this is the highest, i.e., no other pair of people who are currently unacquainted have a higher number of paths of size 2 between them.) Therefore A and G are most likely to be acquainted. Similarly, A and E are least likely to be acquainted since there is only one path of size 5.

One last point: this example involves a fairly simple graph where all the calculations can be done by hand. But imagine posing the same questions, but on a Facebook-scale graph involving (potentially) billions of nodes and edges. How do we reason (mathematically) about such graphs?

This is a vast area of study – but one of immense practical importance that everyone should be aware of. Interesting anecdote: in the late nineties, two graduate students in Stanford (re)discovered a new, ultra-efficient algorithm for estimating "important" nodes of a massive graph (such as the World Wide Web) purely from degree information of the graph. They (Larry Page and Sergei Brin) called the algorithm PageRank, and this algorithm worked very well; so much so that the two founded a small company in the early 00s called *Google*. The rest, as they say, is history.