# Timed Lab 5 – XML Probability Processing

This is a Timed Lab; this Timed Lab is worth 21 **Exam** points.

| For this Timed Lab, you *may* use | However, you *may not* |
|---|---|
| • Course notes<br>• Homeworks<br>• Recitation assignments<br>• Other course material<br>• Any material you may find on the Internet that don't involve communicating "live" with other people. | • Communicate with other people/students in real-time via any means. This means no Facebook, email, Piazza, IM, IRC, cell phones, Google Talk, smoke signals, etc.<br>• Share code with other students.<br>• Look at other students work. |

The TAs will be available to answer clarifying questions about the problem, but they are not permitted to give assistance with debugging code, program logic, etc. You will have an entire recitation period to work on this assignment; this time begins *exactly* when your recitation begins and ends *exactly* when your recitation ends: No extra time will be given if you arrive late, except in highly extenuating circumstances that must be approved by Dr. Summet.

T-Square will not permit any late submissions; ensure that you submit your code to T-Square several times to prevent earning a zero due to you being unable to submit. Your TAs will give a verbal warning 10 and 5 minutes before the end of the recitation period; you should submit at these times. If you are taking this timed lab out of section (with approval from Dr. Summet or the Head TA), please e-mail your code to your grading TA by the end of the recitation you are in. Modifying your code after you leave the recitation room will result in a grade of zero.

In your collaboration statement, if you use code from somewhere that is *not* a class resource (i.e. not listed on the course calendar), please list where this code came from. Ensure that you fill out the header at the top of the file.

## Problem Description:

In this timed lab, your task is to read in and parse an XML file of the percent chance of a given team winning a particular sports game and compute the likelihood that a particular team will win a game against another particular team, for that same game. This task will be divided up into at least three separate functions, although you may include additional functions if you wish.

URL for sample data:
**http://www.cc.gatech.edu/classes/AY2012/cs2316_fall/codesamples/gamesdata.xml**

## Quick Statistics Overview:

The XML file that you are given provides you with a percentage that a team will win a particular game (not given any specific opponent). To compute the probability that one team will win against another team, we must remember that the other team must lose. To get the probability that the second team will lose, subtract the second team's win probability from 1. Therefore, we multiply the probability that the first team will win and the second team will lose to get the probability that the first team will win a particular game against team two. For example, if UGA has a win percentage at football of 0.055, and GT has a win percentage of 0.99 for football, the percentage chance that UGA will win against GT is: 0.055 x (1 - 0.99) = 0.00055. *NOTE: The statistics behind this formula are not good, just implement the formula given, and learn the correct way to do this in ISYE 2027.*

## Function 1 - parseXMLData:

This function accepts a single string parameter that is the URL of the XML file. It should download the XML file from the web and retrieve the information in the XML file to build a dictionary of the data. The dictionary will be organized such that a key in the dictionary will be a tuple consisting of the name of the game followed by the name of the team, and the value will be the win percentage of the team for that game, as a float. Your function must return this dictionary when you are finished parsing all the data in the XML file.

Using the gamesdata.xml file as an example, one entry in the dictionary would be dictname[("football", "GT")] → 0.99, while another would be dictname[("basketball", "UGA")] → 0.055

## Function 2 - getOtherTeamsForGame:

This function will take in three parameters: the dictionary that contains the data you retrieved from the XML file, the game that you wish to find the teams for, and the team you wish to exclude from your results. This function will return a list of all keys (tuples) that have the given game in the tuple and are not the team you wish to exclude. For example, if we called getOtherTeamsForGame(myDict, "football", "GT"), we would get back all of the tuples with "football" as the game except the one where "GT" is the team.

## Function 3 - getTeamWinProbabilities:

This function will take in three parameters: the dictionary that contains the data you retrieved from the XML file, the game that you wish to find the win probabilities for (as a string), and a team name (a string). This function should compute the probability that the team named in your third parameter will win each game against every other team in your dictionary for that given game (HINT: getOtherTeamsForGame may be useful here). You should return a list of tuples, where each tuple is formatted as (winning team, losing team, win probability against losing team). Refer to the statistics overview above on how to compute the win probability against the losing team.

One way these functions could be used together would be code like the following:

```
url = "http://www.cc.gatech.edu/classes/AY2012/cs2316_fall/codesamples/gamesdata.xml"
teamData = parseXMLData(url)
winProbs = getTeamWinProbabilities(teamData, "football", "GT")
print(winprobs)
[('GT', 'UGA', 0.891), ('GT', 'UT', 0.342342), ('GT', 'AUB', 0.43045199999999995),
('GT', 'LSU', 0.040590000000000036)]
```

## Grading:

**parseXMLData:**

+1: Correct function header
+3: Successfully downloads XML file from given webpage.
+1: Parses XML into ElementTree
+3: Creates dictionary with all data from the XML file included.
+1: Dictionary is in correct format.
+1: Dictionary is returned.

**getOtherTeamsForGame:**

+1: Correct function header
+2: Returned list only includes teams for the given game
+1: Returned list excludes the given team from the results
+1: Returned list is in correct format.

**getOtherTeamsForGame:**

+1: Correct function header
+2: Correctly calculates win probability against other team
+2: Returned correct list with all matchups between the given team and the remaining teams included.
+1: Returned a list of tuples.