Experience Report – Session 3

Lecture 3 focused on the measurement problem and the problem of (in)complete testing. I chose to work on path counting and coverage. This has been a problem area for students doing midterms.

Exercise Process

I divided students into 4 groups, who worked in parallel. Each had the same assignment: draw 3 flowcharts for 3 different subprograms. Make them interesting. You will pass them to the table (group) across from you and they will answer three questions about it:

1. How many tests will it take to achieve complete statement coverage? (And what are they?)
2. Describe, using this flowchart, a bug that would not be detected by the tests you use to achieve statement coverage, and show the test you would use to expose it.
3. How many tests would it take to achieve complete path coverage?

Don't be shy about giving them an interesting chart—give them one you'd like to see analyzed this way. They won't hold back when they design the charts they want *you* to analyze.

Students needed about 15 minutes to draw their flowcharts—we stopped at 2 rather than 3. They passed the charts to the opposite table. With my encouragement, most groups worked at whiteboards.

After about 20 minutes of discussion, we reconvened and students made presentations. Our rules of presentation:

- The presentation starts as "clarifying questions only" session. Students were allowed to ask only those questions that were intended to help the presenter get her message across more clearly.
- When the presenter said she was finished, course discussion shifted to "open season"—any question was permissible, as was debate with the approach of the presenters.

What Happened

We had time for 2 student presentations. The charts were similar across groups, so these two presentations covered the issues that came up across the room.
Students realized that it was easy to find bugs that statement coverage wouldn't lead them to. Several of them played with this and also found bugs that they could miss even with branch coverage.

Path coverage was an infinite problem whenever there was a loop with no forced termination rule. (Forced termination? The program exits the loop if it goes through the loop N times.)

Different groups individually discovered bits of a general point: flow-based coverage doesn't deal with the semantics of the program. It doesn't consider special cases or any other aspect of the meaning of the actions you do or the data you enter. I summarized this point using examples from the different tables.

One of the tables was particularly interesting for the semantics issue. They started out by analyzing a telephone system and were resistant to working with the chart as a formal representation, independent of the semantics of the phone system. They discovered that they could design tests much more quickly by thinking only of the chart and not of what the items on the chart represented, but they were concerned that their tests were not as insightful.