Question 1: Grammars

You will build an interface to a student database. You should be able to enroll yourself in any course, and withdraw from any course that you have enrolled in. You should also be able to have the system list the courses that you are enrolled in, and find out some basic information about each course.

You could build a finite state dialogue system in which it takes you several steps to specify your command. Instead, build a system where you can specify the entire command in a single utterance. For instance, you should be able to say the following types of sentences.

“enroll in CSE550”
“withdraw from CSE503”
“list available courses”
“who is teaching ECE543”
“what is CSE581 about”
“list my courses”

Build a dialogue model with just a start state, end state and two intermediate states. The first intermediate state should issue a prompt such as “how can i help you.” For this state, specify a regular grammar that will allow you to issue the above commands. Make the grammar concise by using a subrule that specifies available courses.

Have the second state just repeat back what was recognized.

Prepare a list of 20 different sentences that are in your grammar and that reasonably covers everything that it allows. Check how well the system did in recognizing each one.

Writeup 1.1 Hand in your grammar. Explain whether you did anything to improve recognition of the course names and to improve the speech synthesis of them.

Question 2: Adding in Semantic Interpretation

For the above system, make up a semantic representation language. Make sure that the representation language is simple enough that the terminal substitution formalism can be used to construct it. Do not make the semantics any more complicated than you need for the above sample of sentences.

Writeup 2.2 Give the semantics for each of the six sentences above.

Writeup 2.3 Augment your regular grammar grammar with the terminal substitution formalism so that the recognizer outputs your semantic representation.

Question 3: A Complete System

Rather than have the second state of your system simple repeat back what was said, turn this into a complete working system. To keep this simple, just implement the ‘enroll’, ‘withdraw’ and ‘list my courses’ commands. You should be able to have the following dialogue.
The system should issue an error message if you try to enroll in a course you are already enrolled in, or try to withdraw from a course you are not in.

This exercise will force you to learn some Tcl programming, which you will need for later assignments.

You might need to be aware of local versus global variables in Tcl. You might also need to be aware of associative arrays in Tcl. There is lots of free documentation about Tcl on the web, including tutorials.

To get you started, you might want the following line of Tcl to declare all available courses in your start state:

```
set courses  {CSE550  ECE543  CSE581  ... }
```

Note that this will be redundant with the list of courses in your speech recognition grammar.

You could use the associative array name `taking` to be true of a course if the user has enrolled in it. In your start state, To begin with, you can set taking to be 0 for each course, with the following.

```
foreach course $courses {
    set taking($course) 0
}
```

To check if a person is taking CSE550, you can just do the following:

```
if { $taking($course) == 1 } {
    ...
}
```

**Writeup 3.4** Hand in the Tcl code for each state.