

## CS 334 – Fall 2004: Project

Design your own special class of abstract computing machines. For example, you might design a category of push-down machines with two stacks instead of just one or Turing machines with several tapes instead of just one or some combination of finite state machines. Be creative. You can even design an “oracle” machine.

Whatever machine you design, you should satisfy the following requirements:

1. Give a complete, formal description of your machine. This will include a tuple that includes states, alphabet(s), transitions, etc. Using your description anyone should be able to design a particular machine that accomplishes some task.
2. Compare your class of machines with those you have studied and place the languages it deals with in the Chomsky hierarchy. Your comparison should tell whether your machines are as powerful or more powerful than other machines we have studied and you should explain why.
3. Is it reasonable to think of your machines as algorithms? Why or why not?
4. Give several examples of your machines (at least 3), each doing some task that helps to illustrate your claims in parts 2 and 3.
5. Tell whether or not non-deterministic machines in your class are more powerful than deterministic ones and justify your claim.
6. Include two quiz like exercises for other students to try out your machine. You should have answers ready for the exercises.
7. All of your requirements should be fulfilled with material written for an audience who already know the material in your textbook about FSA's, PDA's and Turing machines, such as the students taking CS 334. Anyone reading what you have written should understand how to use your machines properly.

You will receive a possible 25 points for each of the first 6 requirements as stated above, making a total of 150 points for your project.

Projects are due on Monday, December 6. You should be ready on that day to give a 10 minute presentation to the class about your class of machines.