Perl Regular Expressions

10-28-2005

- What did we talk about last class?
- Do you have any questions about the reading?

- Context Free (CF) grammars have productions with a single nonterminal on the left and any string of terminals and nonterminals on the right.
- The languages of CF grammars are those generated by pushdown automata.
- Most programming languages are defined by CF grammars.
- Have limited memory, but access to memory isn't random.

- Context Sensitive (CS) grammars have productions of the following form.
  - $\alpha A \beta \rightarrow \alpha \gamma \beta$
- α, β, and γ are arbitrary strings of terminals and nonterminals.
- These are generated by a linear-bounded non-deterministic Turing machine.
- These are the least used of the grammars. Even their theory isn't all that well understood.
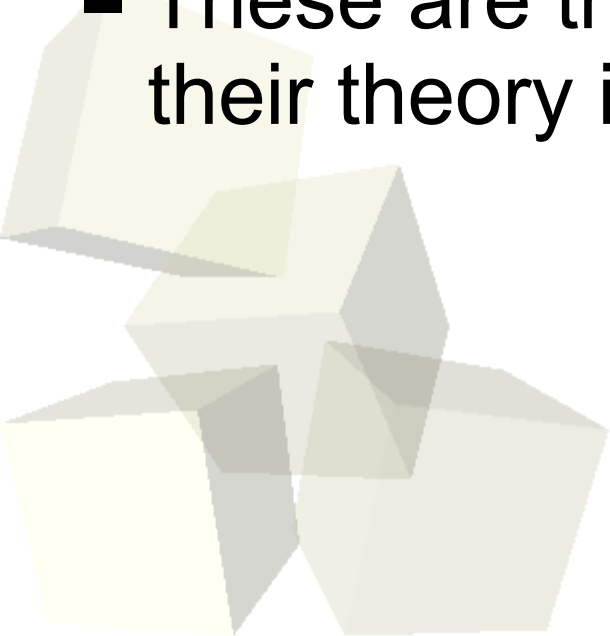
- Recursively enumerable grammars allow any type of production.
- These grammars are computationally equivalent to a full Turing machine so they can generate anything that you want within the bounds of what can be computed.
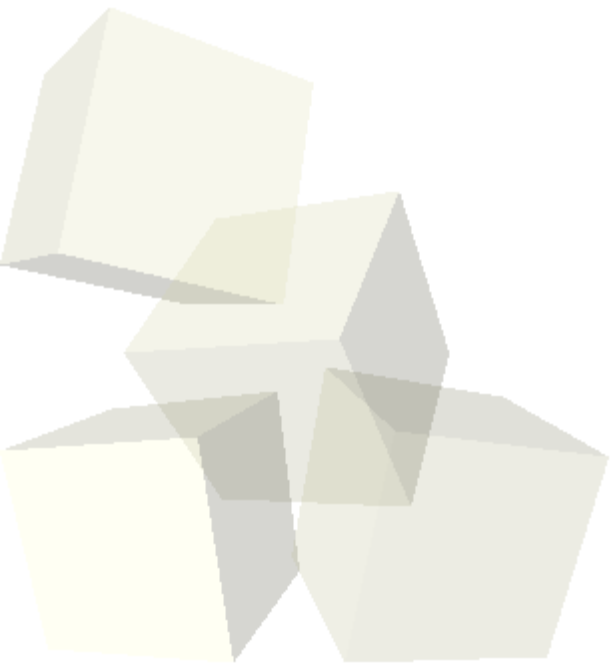
- There are other types of grammars that aren't part of the Chomsky hierarchy.
- We will talk about L-systems a bit later. They are an example of a non-Chomsky set of grammars and they also have different levels of complexity.
- The primary difference between L-systems and Chomsky grammars is that Chomsky grammars replace one randomly selected nonterminal at a time while L-systems replace everything at each step. They also don't technically have terminal symbols.

- We want to write some code that employs Perl regular expressions.
- EPIC (Perl in Eclipse) also has a tool for testing regular expressions. Let's go look at that a bit.
- Now we want to do some simple examples of regular expressions in Perl. Let's start by redoing our telephone book code from last time, but now I want the phone numbers on the same line as the names.
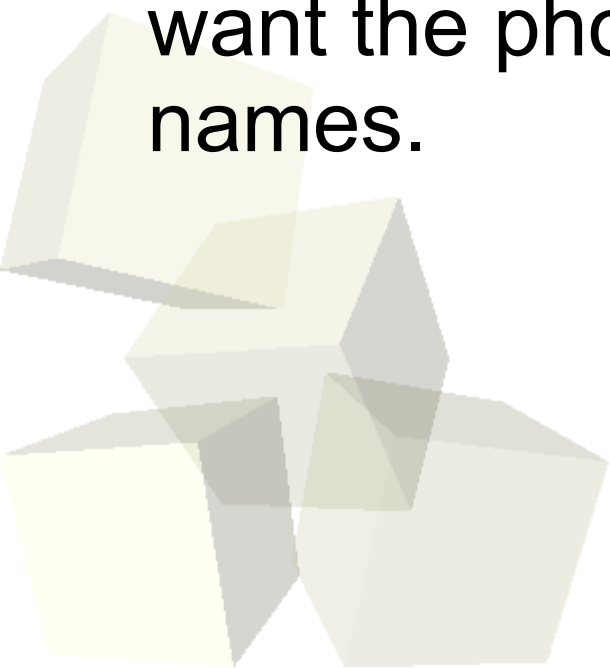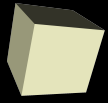
- I will post assignment #7 and kept it pretty simple so it is still due on Monday.