Topics

- Formal definitions
- Numeric Computation
- Text Manipulation

- Upcoming
  ↗ Catch up on reading
Grammar

- English and other natural languages have structure

\[
\begin{align*}
\text{S} & \rightarrow \text{NP} \text{ VP} \\
& \rightarrow \text{N} \mid \text{A} \text{N} \mid \text{PP} \\
\text{VP} & \rightarrow \text{V} \mid \text{V} \text{N} \\
\text{N} & \rightarrow \text{DOG} \mid \text{FLEAS} \mid \text{PERSON} \mid \ldots \\
\text{V} & \rightarrow \text{RAN} \mid \text{BIT} \mid \ldots
\end{align*}
\]

- Process of taking sentence and fitting it to grammar is called *parsing*

DOG BIT PERSON

\[
\begin{align*}
\text{NP} \text{ VP} \\
\rightarrow \text{N} \text{ VP} \\
\rightarrow \text{VP} \text{ N} \\
\rightarrow \text{S}
\end{align*}
\]

- Parsing English is complex because of *context dependence*
Formal specifications

- Need a precise notation of syntax of a language
- Grammars can be used for generation and also can be used
- Context-free grammars

\[ <\text{name}> \rightarrow \text{sequence of letters and/or digits that begins with a letter} \]
\[ <\text{name}> \rightarrow \text{guessB} \]
\[ <\text{name}> \rightarrow \text{msg42} \]
- Substitute as many times as necessary. All legal statements can be generated this way
  - Want \( \text{person} = \text{firstn} + " " + \text{lastn}; \)
  - How do we get this from our grammar?
Random Sentence Generator

- Constructs sentences, paragraphs, and even papers that fit a prescribed format.
- The format is specified by a set of rules called a grammar.
- A grammar consists of a set of definitions.
- Each definition is a set of productions.
- Examples of grammars:
  - Extension request
  - College rejection
  - Poem
  - [http://www.cs.duke.edu/courses/fall01(cps001/code/grammars/](http://www.cs.duke.edu/courses/fall01(cps001/code/grammars/)]
- Natural languages have grammars.

\[
\text{\texttt{S}} \Rightarrow \text{\texttt{NP}} \text{\texttt{VP}}
\]
Poem Grammar

- All grammars begin with start rule

```
{  
  <start>
  The <object> <verb> tonight. ;
}
```

- Nonterminals are indicated by angle brackets

```
{  
  <object>
  waves ;
  big yellow flowers ;
  slugs ;
}
```
More on the poem grammar

- **Nonterminals can refer to other nonterminals**

  ```
  { 
  <verb>
  sigh <adverb> ;
  portend like <object> ;
  }

  { 
  <adverb>
  warily ;
  grumpily and <adverb> ;
  }
  ```
Generating a poem

- all sentences start with `<start>`

<start>

- There is only one production in the definition of `<start>`

  The `<object>` `<verb>` tonight.

- Expand each grammar element from left to right
- "The" is a terminal, so it is simply printed –
- `<object>` is a non-terminal, so it must be expanded
- Choose one:
  - waves
  - big yellow flowers
  - slugs
- Suppose that 'slugs' is chosen
Generating a poem

The slugs <verb> tonight.
\[\Rightarrow\] <-<verb> is a non-terminal, so it must be expanded –
\[\Rightarrow\] Choose one:
   1. sigh <adverb>
   2. portend like <object>

The slugs sigh <adverb> tonight.
\[\Rightarrow\] <adverb> is a non-terminal, so it must be expanded
   1. warily
   2. grumpily

The slugs sigh grumpily tonight.
\[\Rightarrow\] "tonight." is a non-terminal so it is simply printed
\[\Rightarrow\] There are no more non-terminals to expand!
\[\Rightarrow\] The grammar has generated a complete poem
Dealing with numbers

- **Primitive data type: int**
  - Does not require a new statement to create
  - Primitive types not classes
  - Must *declare*
  - Should *initialize* (Java sets to 0)
  - Other primitive types include: boolean, char, double

- **Operations using integers**
  - +, -, *, /, %
  - Operator Precedence
Dealing with text

- **Strings are a class and not a primitive datatype**
- **Declaration:**
  ```java
  String message;
  ```
- **String Constants**
  ```
  “Good Morning World!”
  ```
- **String Assignment**
  ```java
  message = "It's Friday";
  ```
Manipulating Strings

- **Methods for manipulation**
  - int length()
  - int indexOf(String st)
  - String substring(int start, int end)

- **Getting String Data from user**
  - The TextField class has getText() method
  - Use:
    ```
    message = mg.getText();
    ```
    - where mg is a TextField and message is a String
Analog vs. Digital

- **Digital**
  - On or off
  - Computer is made up of millions of "switches" or transistors

- **Analog**
  - Continuously varying
  - Analog vs. Digital clocks

- **What does it mean to have digital music/video?**
  - CDs/DVDs vs. records/VHS tapes
Advantages of digital media

- “Lossless” copying
- **Current event:**
  - Apple has introduced new computer with DVD writing ability
  - No DVD copying ability however
  - Limit technology so that people cannot even attempt to copy data that may or may not be legal
- **Copying copyrighted material for sale vs. “time-shifting”**
- **Napster**
- **Audio Home Recording Act of 1992**
  - Serial Copy Management Systems
  - Opinions?