Building Java Programs

Chapter 11
Sets and Maps

reading: 11.2 - 11.3
### Road Map - Quarter

#### CS Concepts
- Client/Implementer
- Efficiency
- Recursion
- Regular Expressions
- Grammars
- Sorting
- Backtracking
- Hashing
- Huffman Compression

#### Java Language
- Exceptions
- Interfaces
- References
- Comparable
- Generics
- Inheritance/Polymorphism
- Abstract Classes

#### Data Structures
- Lists
- Stacks
- Queues
- Sets
- Maps
- Priority Queues

#### Java Collections
- Arrays
- ArrayList
- LinkedList
- Stack
- TreeSet / TreeMap
- HashSet / HashMap
- PriorityQueue
Note Card Drawing(s)

i like your hat
S
  /\  \\
 NP  VP
 /    /
 John V  NP
    /    /
   lost  Det N
          /     /
         his    pants
Languages and grammars

• (formal) **language**: A set of words or symbols.

• **grammar**: A description of a language that describes which sequences of symbols are allowed in that language.
  • describes language *syntax* (rules) but not *semantics* (meaning)
  • can be used to generate strings from a language, or to determine whether a given string belongs to a given language
Backus-Naur (BNF)

• **Backus-Naur Form (BNF):** A syntax for describing language grammars in terms of transformation *rules*, of the form:

  \[ \langle \text{symbol} \rangle ::= \langle \text{expression} \rangle \mid \langle \text{expression} \rangle \ldots \mid \langle \text{expression} \rangle \]

• **terminal**: A fundamental symbol of the language.

• **non-terminal**: A high-level symbol describing language syntax, which can be transformed into other non-terminal or terminal symbol(s) based on the rules of the grammar.

• developed by two Turing-award-winning computer scientists in 1960 to describe their new ALGOL programming language.
An example BNF grammar

\[ \text{<s>}::=\text{<n>} \ \text{<v>}} \]
\[ \text{<n>}::=\text{Marty} \mid \text{Victoria} \mid \text{Stuart} \mid \text{Jessica} \]
\[ \text{<v>}::=\text{cried} \mid \text{slept} \mid \text{belched} \]

- Some sentences that could be generated from this grammar:
  
  Marty slept
  Jessica belched
  Stuart cried
BNF grammar version 2

<s>::=<np> <v>  
<np>::=<pn> | <dp> <n>  
<pn>::=Marty | Victoria | Stuart | Jessica  
<dp>::=a | the  
<n>::=ball | hamster | carrot | computer  
<v>::=cried | slept | belched

- Some sentences that could be generated from this grammar:
  - the carrot cried
  - Jessica belched
  - a computer slept
BNF grammar version 3

<s>::=<np> <v>
[np>::=<pn> | <dp> <adj> <n>
<pn>::=Marty | Victoria | Stuart | Jessica
<dp>::=a | the
<adj>::=silly | invisible | loud | romantic
<n>::=ball | hamster | carrot | computer
<v>::=cried | slept | belched

• Some sentences that could be generated from this grammar:

  the invisible carrot cried
  Jessica belched
  a computer slept
  a romantic ball belched
Grammars and recursion

\[
\begin{align*}
\text{s} & ::= \text{np} \quad \text{v} \\
\text{np} & ::= \text{pn} \quad | \quad \text{dp} \quad \text{adjp} \quad \text{n} \\
\text{pn} & ::= \text{Marty} \quad | \quad \text{Victoria} \quad | \quad \text{Stuart} \quad | \quad \text{Jessica} \\
\text{dp} & ::= \text{a} \quad | \quad \text{the} \\
\text{adjp} & ::= \text{adj} \quad \text{adjp} \quad | \quad \text{adj} \\
\text{adj} & ::= \text{silly} \quad | \quad \text{invisible} \quad | \quad \text{loud} \quad | \quad \text{romantic} \\
\text{n} & ::= \text{ball} \quad | \quad \text{hamster} \quad | \quad \text{carrot} \quad | \quad \text{computer} \\
\text{v} & ::= \text{cried} \quad | \quad \text{slept} \quad | \quad \text{belched}
\end{align*}
\]

- Grammar rules can be defined *recursively*, so that the expansion of a symbol can contain that same symbol.
  - There must also be expressions that expand the symbol into something non-recursive, so that the recursion eventually ends.
### Grammar, final version

\[
<s>::=<np> <vp> \\
<np>::=<dp> <adjp> <n>|<pn> \\
<dp>::=the|a \\
<adjp>::=<adj>|<adj> <adjp> \\
<adj>::=big|fat|green|wonderful|faulty|subliminal \\
<n>::=dog|cat|man|university|father|mother|child \\
<pn>::=John|Jane|Sally|Spot|Fred|Elmo \\
<vp>::=<tv> <np>|<iv> \\
<tv>::=hit|honored|kissed|helped \\
<iv>::=died|collapsed|laughed|wept
\]

- **Could this grammar generate the following sentences?**
  Fred honored the green wonderful child
  big Jane wept the fat man fat
- **Generate a random sentence using this grammar.**
Sentence generation

<s>
  <np>
    <pn> Fred 
  </np>
  <vp>
    <tv>
    </tv>
    <np>
      <dp>
      </dp>
      <adjp>
        <adj> the 
      </adjp>
      <n>
        <adj> green 
        <adjp>
          <adj> wonderful 
          <adj> child 
        </adjp>
      </n>
  </vp>
</s>