Today’s topics

Language Translation
   Revising the Syntax Rules and Notation

Upcoming
   Language Translation
      Generating Code

Reading
   *Great Ideas*, Chapters 9
Importance of Language

- Vehicle for programming
- Use for human and machine communications
- Syntax Rules
  - First pass already covered
  - Need to refine the notation
    - Must be suitable for machine to use
- In addition, need to deal with the meaning
- Also, should see Levels or Layers in dealing with computer
  1. Hardware
  2. Machine language
  3. Assembler
  4. Java (or other high-level language)
  5. Application (e.g. Word, Excel, Filemaker, …)
Language Translation

❖ **Goal is to automatically**
  - Translate from **Java**:
    
    \[
    z = x + y;
    \]
  - to **Assembler**:
    
    \[
    \text{copy } \text{ax, } x \\
    \text{add } \text{ax, } y \\
    \text{copy } z, \text{ ax}
    \]

❖ **What is the meaning we are looking for?**

Machine gives assembler statements meaning because the machine knows what to do with them (after trivial translation to binary). E.g., the machine knows what **add** means.
Revise Syntactic Rules

- Need to revise Syntactic Production Rules
  - New rule:
    R1: \(<n>j\) -> a sequence of letters and/or digits that begin with a letter
  - Replaces (have seen this before):
    R1: \(<\text{name}>\) -> a sequence of letters and/or digits that begin with a letter
  - The new R1 says “change \(<n>j\) into a sequence of letters and/or digits that begin with a letter”

- Use rules to modify strings
  - For syntactic productions, must end up with valid Java programs
Using Syntax Rules

- **Examples using R1:**
  
  \(<n>_3 \rightarrow x\)
  
  or
  
  \(<n>_6 \rightarrow data\)
  
  where "n" stand for “name”

- **Further use of R1:**
  
  \((<n>_3 + <n>_6)\)
  
  Replace \(<n>_3\) above to get
  
  \((x + <n>_6)\)
  
  and \(<n>_6\) to get
  
  \((x + data)\)

- **More Rules:**
  
  - R2 : \(<e>_i \rightarrow <n>_j\)
    
    Where “e” stands for “expression”
  
  - Example:
    
    \(<e>_1 \rightarrow <n>_3\)
Using Syntax Rules

- and
  - R3: \(<s>_k \rightarrow <n>_j = <e>_i ;\>

Where “s” stands for “statement”
It says “<s>_k” can be replaced by “<n>_j = <e>_i ;”

- Can now do: \(\text{ans} = \text{data};\)

**derivation**

**rule**

\(<s>_1\) \hspace{1cm} \text{R3}: \(<s>_1 \rightarrow <n>_2 = <e>_3;\>
\(<n>_2 = <e>_3;\) \hspace{1cm} \text{R1}: \(<n>_2 \rightarrow \text{ans}\)
\(\text{ans} = <e>_3;\) \hspace{1cm} \text{R2}: \(<e>_3 \rightarrow <n>_4\)
\(\text{ans} = <n>_4;\) \hspace{1cm} \text{R1}: \(<n>_4 \rightarrow \text{data}\)
\(\text{ans} = \text{data};\)
More Rules

- Need two more rules to make it worthwhile
  - R4: $<e>_i \rightarrow ( <e>_j + <e>_k )$
  - R5: $<e>_i \rightarrow ( <e>_j * <e>_k )$

  These are additional rules for *expressions*

- Can now handle $\text{ANS} = (X + (Y * Z))$;
  (notice shorthand/simplifications used)
Longer Example

\[ \text{ANS} = (X + (Y \times Z)) \];

derivation

\[
\begin{align*}
\text{s1} & \\
\text{n2} &= e3; \\
\text{ANS} &= e3; \\
\text{ANS} &= (e4 + e5); \\
\text{ANS} &= (n6 + e5); \\
\text{ANS} &= (X + e5); \\
\text{ANS} &= (X + (e7 \times e8)); \\
\text{ANS} &= (X + (n9 \times e8)); \\
\text{ANS} &= (X + (Y \times e8)); \\
\text{ANS} &= (X + (Y \times n10)); \\
\text{ANS} &= (X + (Y \times Z));
\end{align*}
\]

rule

\[
\begin{align*}
\text{R3}: \text{s1} & \rightarrow \text{n2} = e3; \\
\text{R1}: \text{n2} & \rightarrow \text{ANS} \\
\text{R4}: \text{e3} & \rightarrow (e4 + e5) \\
\text{R2}: \text{e4} & \rightarrow n6 \\
\text{R1}: \text{n6} & \rightarrow X \\
\text{R5}: \text{e5} & \rightarrow (e7 \times e8) \\
\text{R2}: \text{e7} & \rightarrow n9 \\
\text{R1}: \text{n9} & \rightarrow Y \\
\text{R2}: \text{e8} & \rightarrow n10 \\
\text{R1}: \text{n10} & \rightarrow Z
\end{align*}
\]
Notes

- Abbreviations
  - Just omitted the angle brackets. Could do this because the notation remained unambiguous.

- Role of the subscripts
  - The subscripts are required to make sure each term is unique.
  - Simplest technique is to start subscripts at one and increment every time a different subscript is needed.

- Simple substitution is all that is required!!!
  - If you are doing something more than that, it is probably wrong!

- The notation and form are important
  - You will be expected to match them on tests.