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 to notation
    - Must be suitable for machine to do
- 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:
    \[
    \text{copy } ax, x \\
    \text{add } ax, y \\
    \text{copy } z, ax
    \]

- **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: \(<n>j \rightarrow \text{a sequence of letters and/or digits that begin with a letter}\)
  - Replaces (have seen this before):
    - R1: \(<\text{name}> \rightarrow \text{a sequence of letters and/or digits that begin with a letter}\)
  - 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:**
  \[
  \langle n \rangle_3 \rightarrow x
  \]
  or
  \[
  \langle n \rangle_6 \rightarrow \text{data}
  \]
  where “n” stand for “name”

- **Further use of R1:**
  \[
  (\langle n \rangle_3 + \langle n \rangle_6)
  \]
  Use \( \langle n \rangle_3 \) and \( \langle n \rangle_6 \) above to get
  \[
  (x + \text{data})
  \]

- **More Rules:**
  - **R2:** \( \langle e \rangle_i \rightarrow \langle n \rangle_j \)
    Where “e” stands for “expression”
  - Example:
    \[
    \langle e \rangle_1 \rightarrow \langle n \rangle_3
    \]
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: \(\text{ans} = \text{data};\)

<table>
<thead>
<tr>
<th>derivation</th>
<th>rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;s&gt;_1)</td>
<td>R3: (&lt;s&gt;_1 \rightarrow &lt;n&gt;_2 = &lt;e&gt;_3;)</td>
</tr>
<tr>
<td>(&lt;n&gt;_2 = &lt;e&gt;_3;)</td>
<td>R1: (&lt;n&gt;_2 \rightarrow \text{ans})</td>
</tr>
<tr>
<td>(\text{ans} = &lt;e&gt;_3;)</td>
<td>R2: (&lt;e&gt;_3 \rightarrow &lt;n&gt;_4)</td>
</tr>
<tr>
<td>(\text{ans} = &lt;n&gt;_4;)</td>
<td>R1: (&lt;n&gt;_4 \rightarrow \text{data})</td>
</tr>
<tr>
<td>(\text{ans} = \text{data};)</td>
<td></td>
</tr>
</tbody>
</table>
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**

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

**rule**

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