Grammars: Defining Languages

Walls & Mirrors Ch. 6.2

### Definitions

- **Language** is a set of strings of symbols from a finite alphabet.
  
  \[ \text{JavaPrograms} = \{ \text{string } w : w \text{ is a syntactically correct Java program} \} \]

- **Grammar** is a set of rules that the strings must follow.

- **Recognition Algorithm** determines whether a string is a member of the language.

### Basics of Grammars

Example: a grammar for Java identifiers

\[
\begin{align*}
<\text{identifier}> & : <\text{letter}> | <\text{identifier}> <\text{digit}> | \\
& | <\text{identifier}> <\text{digit}> | \\
& | <\text{digit}> \\
<\text{letter}> & : a | b | \ldots | x | A | B | \ldots | Z \\
<\text{digit}> & : 0 | 1 | \ldots | 9
\end{align*}
\]

- `|` means "or"
- `.` means "is followed by"
- `<word>` is called a non-terminal, which can be replaced by other symbols depending on the rules.
- Terminals are symbols (e.g., letters, words) from which legal strings are constructed.
- Rules have the form `<word>` = ...

### Syntax Diagrams

Java Identifier

```
letter


digit, letter
```

from the Scanner Java documentation page:
Basics of Grammars

Example: a grammar for Java identifiers

<identifier> = <letter> | <identifier> <letter> | <identifier> <digit> | $ | _ <identifier>
<letter> = a | b | ... | z | A | B | ... | Z
<digit> = 0 | 1 | ... | 9

How do we determine if a string \( w \) is a valid Java identifier, i.e. belongs to the language of Java identifiers?

\[
isId(in\ w: string) : boolean
\begin{array}{l}
\text{if (} w \text{ is of length 1)} \\
\quad \text{if (} w \text{ is a letter)} \\
\quad \quad \text{return true}
\text{else} \\
\quad \quad \text{return false}
\text{else if (the last character of} \ w \text{ is a letter or a digit)} \\
\quad \quad \text{return isId(} w \text{ minus its last character)}
\text{else} \\
\quad \quad \text{return false}
\end{array}
\]

Java Class

\[
<class declaration > =
\text{<modifier>} "class" <identifier>
\text{[ extends <class_name> ]}
\text{[ implements <interface_name> ]}
\text{[ , <interface_name> ]}
\text{[ } <\text{field declaration}> ]
\text{}]}
\]

Prefix Expressions

Grammar for prefix expression (e.g. \(* - a b c\)):

<pre>prefix \* <identifier> | <operator> prefix prefix
<operator> = + | - | * | /
<identifier> = a | b | ... | z
</pre>

Palindromes

Palindromes = \( \{ w : w \text{ reads the same left to right as right to left} \}\)

Recursive definition:

\( w \) is a palindrome if and only if

- the first and last characters of \( w \) are the same
- \( w \) minus its first and last characters is a palindrome

Base case?
Grammar for Palindromes

\[
\begin{align*}
\text{<pal>} & = \text{empty string} \mid \text{<ch>} \mid \text{a} \text{<pal>} \text{a} \mid \ldots \mid \text{Z} \\
\langle \text{pal} \rangle & \text{Z} \\
\text{<ch>} & = \text{a} \mid \text{b} \mid \ldots \mid \text{z} \mid \text{A} \mid \text{B} \mid \ldots \mid \text{Z}
\end{align*}
\]

Let's write a recursive method that decides if a string \( w \) is a palindrome.

The Java Scanner Class

- \text{Scanner} divides an input stream (e.g., from a file) into words separated by delimiters.
- \text{Scanner} defines a grammar for the syntax of numbers.

Recognizing Languages

- \textbf{Parsing} determines whether a sequence of symbols obeys rules of a grammar and assigns the symbols to non-terminals from the grammar.
- More on languages in CS301!