Lecture 6:

- Formal Grammars

Announcements:

- HW-1 out (due Mon, 2/3)

More on Formal Grammars

There are many types (classes) of formal grammars ... expressive power

- The **regular** grammars specify regular languages (think regular expressions)
- The **context free** grammars specify context-free languages (most PLs)
- ... and so on

Can use grammar rules in "two directions": ... parsing is a mix of both

- Given a string, check if it follows the rules
- Apply the rules to generate (*derive*) strings

Aside: Grammars closely tied to computation ... models of computation

- Applying rules to derive strings is "computing" the strings of language
- Viewing strings as algorithm outputs, the grammar "computes" the outputs

Context-Free Grammars

A context-free grammar consists of:

- an *alphabet* of **terminal** symbols ... e.g., token types or characters
- a set of **non-terminal** symbols ... each denotes a set of strings
- **production rules** of the form $N \to X_1 \ldots X_n$ ... the grammar rules

  ... N a *non-terminal*, $X_1 \ldots X_n$ terminals or non-terminals ($n \geq 0$)
- a **start symbol** (non-terminal) ... explicit or implicit

1-Rule Example:

$S \to$ a ... S is a non-terminal, a is a terminal

- Read as: S *produces the string* "a"
- Thus, S is the one-element set {"a"}
- And in this case, the language of the grammar is just {"a"}

Context-Free Grammars

Concatenation:

$S \to$ a b

- S *produces the string* "a" *followed by* "b" (i.e., "ab")
- The language is {"ab"}

Note: Many ways to define the same language using grammars:

$S \to UV$

$U \to$ a

$V \to$ b

- S assumed to be the start symbol as the LHS of the first rule

Context-Free Grammars

Alternation:

$S \rightarrow$ a

$S \rightarrow$ b

- S *can produce the string* "a" *or the string* "b"
- The language is {"a", "b"}

Often use the special symbol | as shorthand:

$S \rightarrow$ a | b

()'s can be used for grouping ... part of grammar rule, not language

$S \rightarrow$ a | (b | c) ... {"ab", "ac"}

- equivalent to: $S \rightarrow$ a | T and $T \rightarrow$ b | c

Context-Free Grammars

The empty string:

$S \rightarrow$

- S *produces the empty string* ""
- The language is {""}

Often use ε as shorthand for empty string:

$S \rightarrow$ a | ε

- S produces the language {"", "a"}

Context-Free Grammars

The star operator: ... aka Kleene star

$S \rightarrow \texttt{a}^*$

- S *produces strings with zero or more* a*'s* ... {"", "a", "aa", ...}

Q: What is the language of this grammar?

$S \rightarrow \texttt{a}^* \, \texttt{b}^*$

- strings with any number of a's followed by any number of b's
- sometimes written as: $\{\texttt{a}^i\texttt{b}^j \mid i, j \geq 0\}$ ... set-builder notation

Q: What about this one?

$S \rightarrow (\, \texttt{a} \mid \texttt{b} \,)^*$ ... ()'s part of grammar rule, not the language

- strings with any combination of a's and b's in any order

Context-Free Grammars

Note: alternation has lower precedence than other "operators"

- The rule: $S \rightarrow \texttt{a}\,\texttt{b}^*\texttt{c} \mid \texttt{d}^*\texttt{e}$
- Is the same as: $S \rightarrow (\texttt{a}\,\texttt{b}^*\texttt{c}) \mid (\texttt{d}^*\texttt{e})$

Note: star has higher precedence:

- The rule: $S \rightarrow \texttt{a}\,\texttt{b}^*\texttt{c}$
- Is the same as: $S \rightarrow \texttt{a}\,(\,\texttt{b}\,)^*\texttt{c}$

Q: What is the language of this grammar rule?

$S \rightarrow (\, \texttt{a} \mid \texttt{b} \,)^* \mid (\, \texttt{d} \mid \texttt{e} \,)^*$

- empty string, all combos of a and b, and all combos of d and e

Context-Free Grammars

Recursion: non-terminal either *directly* used in same rule, or *indirectly* ...

Direct Example:

$S \rightarrow \mathtt{a}\, S\, \mathtt{b} \mid \varepsilon$ ... S occurs (directly) in S rule

- the language of S is $\{\mathtt{a}^i\, \mathtt{b}^i \mid i \geq 0\}$
- note this is not possible to express using just a star (*)
- but star (*) can be implemented using recursion ... $S \rightarrow a\, S \mid \varepsilon$

Indirect Example:

$S \rightarrow T \mid \varepsilon$

$T \rightarrow \mathtt{a}\, S\, \mathtt{b}$