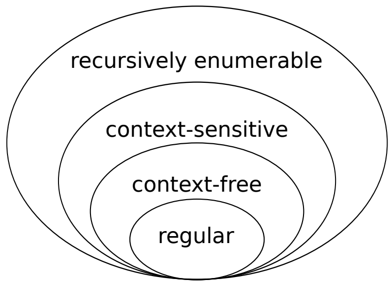
# CS3100 Final Note Sheet

## Notation

| | |
|---|---|
| $L^R$ | The reversal of language $L$. |
| $L^*$ | The Kleene-Star of $L$. |
| $AB$ | The concatenation of language $A$ and $B$. |
| $h(L)$ | A homomorphism (a function that maps every input to a unique output) of $L$. |
| $A \setminus B$ | Set difference of $A$ and $B$. $A-B$ is the same thing. |

## Chomsky Hierarchy




## Regular Languages

### Closures

Where $R$ is a regular language, $L$ is 'not regular', and ? is Unknown.

**Closed:**

$\overline{R} \rightarrow R$ $\quad h(R) \rightarrow R$
$R^* \rightarrow R$ $\quad R \cup R \rightarrow R$
$R^R \rightarrow R$ $\quad R \cap R \rightarrow R$
$RR \rightarrow R$ $\quad R \setminus R \rightarrow R$

**Unclosed:**

$R \cap L \rightarrow ?$
$R \cup L \rightarrow ?$
$L \cup L \rightarrow ?$

### Pumping Lemma

$$\begin{aligned}
&\exists N \in \mathbb{N} : \\
&\forall w \in L : |w| \geq N \Rightarrow \\
\exists xyz \in \Sigma^* : \quad & w = xyz \\
&\wedge |xy| \leq N \\
&\wedge y \neq \varepsilon \\
&\wedge \forall i \geq 0 : xy^iz \in L
\end{aligned}$$

## Context Free Languages

### Closures

Where $C$ is a context-free language, $R$ is a regular language, and ? is an Unknown language.

**Closed:**

$C^R \rightarrow C$ $\quad C \cup C \rightarrow C$
$C^* \rightarrow C$ $\quad C \cap R \rightarrow C$
$CC \rightarrow C$ $\quad C \cup R \rightarrow R$
$h(C) \rightarrow C$

**Unclosed:**

$\overline{C} \rightarrow ?$
$C \cap C \rightarrow ?$
$C \setminus C \rightarrow ?$

### Pumping Lemma

$$\begin{aligned}
&\exists N \in \mathbb{N} : \\
&\forall w \in L : |w| \geq N \Rightarrow \\
\exists uvxyz \in \Sigma^* : \quad & w = uvxyz \\
&\wedge |vy| > 0 \\
&\wedge |vxy| \leq N \\
&\wedge \forall i \geq 0 : uv^ixy^iz \in L
\end{aligned}$$

### Ambiguous Context Free Languages

These are languages that have two separate parse-trees. To prove that a language is ambiguous, show that it actually has two separate parse-trees.

Example ambiguous grammar:

$$E \rightarrow E + E \mid E * E \mid \texttt{NUMBER}$$

### Consistency and Completeness

**Consistency:** All strings generated by a grammar are in the language.

**Completeness:** The grammar generates all strings in the language.

You cannot know that a grammar defines a language until you show both. For example, if we want to define the language $\{a^nb^n | n \in \mathbb{N}\}$, the grammar:

$$S \rightarrow aabb$$

Is *consistent* because it only generates strings in the language, but not complete because it doesn't generate all strings in the language. Likewise, the grammar:

$$S \rightarrow aS \mid bS \mid \varepsilon \qquad (\textit{grammar for } \{a, b\}^*)$$

Is complete, it generates all possible strings in the language, but not consistent because it generates many strings that are, in-fact, outside of the language.

### RL and LL Grammar Conversion