# Stuttgart Fall School in CL
# Class 2: Rewrite Grammars, CFGs

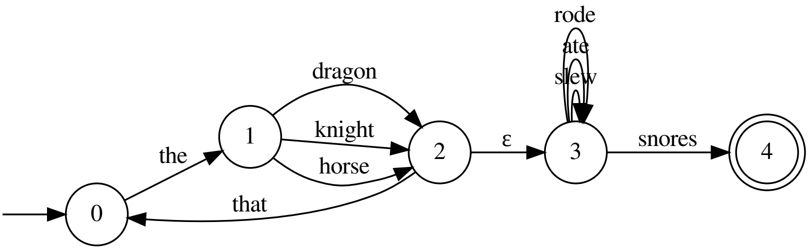
Dr. Meaghan Fowlie

September 16, 2019

## Review: Quiz

Which of these sentences is accepted by the following FSA?

1. the horse the knight rode snores
2. the dragon that the knight slew snores
3. that the dragon rode snores
4. the dragon ate
5. the horse that the knight that the dragon ate rode snores
6. the dragon rode ate slew ate snores



# Rewrite Grammars

Let N be a set of *non-terminal symbols*, $\Sigma$ a set of terminal symbols, and $N \cap \epsilon = \emptyset$. Choose a distinguished set of *start symbols* $S \subseteq N$.
Let $\alpha, \beta, \gamma \in (N \cup \Sigma)^*$ and $X \in N$. A *rewrite grammar* has a set of rules R of the form

$$\alpha X \beta \quad \rightarrow \quad \gamma$$

Example

$$\begin{array}{lcl} 0 & \rightarrow & \text{the } 1 \\ 1 & \rightarrow & \text{dragon } 2 \\ 1 & \rightarrow & \text{knight } 2 \\ 1 & \rightarrow & \text{horse } 2 \\ 2 & \rightarrow & \text{that } 0 \\ 2 & \rightarrow & \epsilon\ 3 \\ 3 & \rightarrow & \text{rode } 3 \\ 3 & \rightarrow & \text{ate } 3 \\ 3 & \rightarrow & \text{slew } 3 \\ 3 & \rightarrow & \text{snores} \end{array}$$

## Rewrite Grammars

Let N be a set of *non-terminal symbols*, $\Sigma$ a set of terminal symbols, and $N \cap \epsilon = \emptyset$. Choose a distinguished set of *start symbols* $S \subseteq N$.
Let $\alpha, \beta, \gamma \in (N \cup \Sigma)^*$ and $X \in N$. A *rewrite grammar* has a set of rules R of the form

$$\alpha X \beta \quad \rightarrow \quad \gamma$$

Example

$$
\begin{array}{lcl}
0 & \rightarrow & \text{the } 1 \\
1 & \rightarrow & \text{dragon } 2 \\
1 & \rightarrow & \text{knight } 2 \\
1 & \rightarrow & \text{horse } 2 \\
2 & \rightarrow & \text{that } 0 \\
2 & \rightarrow & \epsilon\ 3 \\
3 & \rightarrow & \text{rode } 3 \\
3 & \rightarrow & \text{ate } 3 \\
3 & \rightarrow & \text{slew } 3 \\
3 & \rightarrow & \text{snores}
\end{array}
$$

# Rewrite Grammars: Example

A *rewrite grammar* is a 4-tuple $\langle \Sigma, \text{N}, \text{S}, \text{R} \rangle$
(i.e. $\langle$terminals, nonterminals, start symbols, rules $\rangle$)

Example

$G_{\text{knight}} = \langle$\{the, that, dragon, knight, horse, that, rode, ate, slew, snores\},

$$\{0,1,2,3\},\ \{0\},\ \begin{array}{lcl} 0 & \rightarrow & \text{the } 1 \\ 1 & \rightarrow & \text{dragon } 2 \\ 1 & \rightarrow & \text{knight } 2 \\ 1 & \rightarrow & \text{horse } 2 \\ 2 & \rightarrow & \text{that } 0 \\ 2 & \rightarrow & \epsilon\ 3 \\ 3 & \rightarrow & \text{rode } 3 \\ 3 & \rightarrow & \text{ate } 3 \\ 3 & \rightarrow & \text{slew } 3 \\ 3 & \rightarrow & \text{snores} \end{array}$$

$\rangle$

# The language of the grammar

A rewrite grammar $G$ defines a language $L(G) \subseteq \Sigma^*$ as follows:

1. Take a start symbol s from $S$. This is your string w.
2. if $w \in \Sigma^*$ then $w \in L$.
3. If there is a substring x of w which matches the LHS of a rule, replace x with the RHS of the rule. Go back to step 2.

We say G *generates* L(G) (hence "generative grammar")

# Language

$$
\begin{array}{lll}
0 & \to & \text{the } 1 \\
1 & \to & \text{dragon } 2 \\
1 & \to & \text{knight } 2 \\
1 & \to & \text{horse } 2 \\
2 & \to & \text{that } 0 \\
2 & \to & \epsilon\ 3 \\
3 & \to & \text{rode } 3 \\
3 & \to & \text{ate } 3 \\
3 & \to & \text{slew } 3 \\
3 & \to & \text{snores}
\end{array}
$$

## Example

0 $\Rightarrow$ the 1 $\Rightarrow$ the knight 2 $\Rightarrow$ the knight that 0 $\Rightarrow$the knight that the 1 $\Rightarrow$ the knight that the horse 2 $\Rightarrow$ the knight that the horse 3 $\Rightarrow$ the knight that the horse snores $\in \Sigma^*$

Therefore, *the knight that the horse snores* $\in L(G_{\text{knight}})$

# Language

$$
\begin{array}{lcl}
0 & \rightarrow & \text{the } 1 \\
1 & \rightarrow & \text{dragon } 2 \\
1 & \rightarrow & \text{knight } 2 \\
1 & \rightarrow & \text{horse } 2 \\
2 & \rightarrow & \text{that } 0 \\
2 & \rightarrow & \epsilon\ 3 \\
3 & \rightarrow & \text{rode } 3 \\
3 & \rightarrow & \text{ate } 3 \\
3 & \rightarrow & \text{slew } 3 \\
3 & \rightarrow & \text{snores}
\end{array}
$$

Example

0 $\Rightarrow$ the 1 $\Rightarrow$ the knight 2 $\Rightarrow$ the knight that 0 $\Rightarrow$the knight that the 1 $\Rightarrow$ the knight that the horse 2 $\Rightarrow$ the knight that the horse 3 $\Rightarrow$ the knight that the horse snores $\in \Sigma^*$

Therefore, *the knight that the horse snores* $\in L(G_{\text{knight}})$

# Language

$$
\begin{array}{lcl}
0 & \rightarrow & \text{the } 1 \\
1 & \rightarrow & \text{dragon } 2 \\
1 & \rightarrow & \text{knight } 2 \\
1 & \rightarrow & \text{horse } 2 \\
2 & \rightarrow & \text{that } 0 \\
2 & \rightarrow & \epsilon\ 3 \\
3 & \rightarrow & \text{rode } 3 \\
3 & \rightarrow & \text{ate } 3 \\
3 & \rightarrow & \text{slew } 3 \\
3 & \rightarrow & \text{snores}
\end{array}
$$

**Example**

0 $\Rightarrow$ the 1 $\Rightarrow$ the knight 2 $\Rightarrow$ the knight that 0 $\Rightarrow$the knight that the 1 $\Rightarrow$ the knight that the horse 2 $\Rightarrow$ the knight that the horse 3 $\Rightarrow$ the knight that the horse snores $\in \Sigma^*$

Therefore, *the knight that the horse snores* $\in L(G_{\text{knight}})$

# Language

$$
\begin{array}{lcl}
0 & \rightarrow & \text{the } 1 \\
1 & \rightarrow & \text{dragon } 2 \\
1 & \rightarrow & \text{knight } 2 \\
1 & \rightarrow & \text{horse } 2 \\
2 & \rightarrow & \text{that } 0 \\
2 & \rightarrow & \epsilon\ 3 \\
3 & \rightarrow & \text{rode } 3 \\
3 & \rightarrow & \text{ate } 3 \\
3 & \rightarrow & \text{slew } 3 \\
3 & \rightarrow & \text{snores}
\end{array}
$$

**Example**

0 $\Rightarrow$ the 1 $\Rightarrow$ the knight 2 $\Rightarrow$ the knight that 0 $\Rightarrow$the knight that the 1 $\Rightarrow$ the knight that the horse 2 $\Rightarrow$ the knight that the horse 3 $\Rightarrow$ the knight that the horse snores $\in \Sigma^*$

Therefore, *the knight that the horse snores* $\in L(G_{\text{knight}})$

# Language

$$
\begin{array}{lcl}
0 & \rightarrow & \text{the } 1 \\
1 & \rightarrow & \text{dragon } 2 \\
1 & \rightarrow & \text{knight } 2 \\
1 & \rightarrow & \text{horse } 2 \\
2 & \rightarrow & \text{that } 0 \\
2 & \rightarrow & \epsilon\ 3 \\
3 & \rightarrow & \text{rode } 3 \\
3 & \rightarrow & \text{ate } 3 \\
3 & \rightarrow & \text{slew } 3 \\
3 & \rightarrow & \text{snores}
\end{array}
$$

**Example**

0 $\Rightarrow$ the 1 $\Rightarrow$ the knight 2 $\Rightarrow$ the knight that 0 $\Rightarrow$the knight that the 1 $\Rightarrow$ the knight that the horse 2 $\Rightarrow$ the knight that the horse 3 $\Rightarrow$ the knight that the horse snores $\in \Sigma^*$

Therefore, *the knight that the horse snores* $\in L(G_{\text{knight}})$

# Language

$$
\begin{array}{lcl}
0 & \to & \text{the } 1 \\
1 & \to & \text{dragon } 2 \\
1 & \to & \text{knight } 2 \\
1 & \to & \text{horse } 2 \\
2 & \to & \text{that } 0 \\
2 & \to & \epsilon\ 3 \\
3 & \to & \text{rode } 3 \\
3 & \to & \text{ate } 3 \\
3 & \to & \text{slew } 3 \\
3 & \to & \text{snores}
\end{array}
$$

**Example**

0 $\Rightarrow$ the 1 $\Rightarrow$ the knight 2 $\Rightarrow$ the knight that 0 $\Rightarrow$the knight that the 1 $\Rightarrow$ the knight that the horse 2 $\Rightarrow$ the knight that the horse 3 $\Rightarrow$ the knight that the horse snores $\in \Sigma^*$

Therefore, *the knight that the horse snores* $\in L(G_{\text{knight}})$

# Language

$$
\begin{array}{lcl}
0 & \rightarrow & \text{the } 1 \\
1 & \rightarrow & \text{dragon } 2 \\
1 & \rightarrow & \text{knight } 2 \\
1 & \rightarrow & \text{horse } 2 \\
2 & \rightarrow & \text{that } 0 \\
2 & \rightarrow & \epsilon \; 3 \\
3 & \rightarrow & \text{rode } 3 \\
3 & \rightarrow & \text{ate } 3 \\
3 & \rightarrow & \text{slew } 3 \\
3 & \rightarrow & \text{snores}
\end{array}
$$

Example

0 $\Rightarrow$ the 1 $\Rightarrow$ the knight 2 $\Rightarrow$ the knight that 0 $\Rightarrow$the knight that the 1 $\Rightarrow$ the knight that the horse 2 $\Rightarrow$ the knight that the horse 3 $\Rightarrow$ the knight that the horse snores $\in \Sigma^*$

Therefore, *the knight that the horse snores* $\in L(G_{\text{knight}})$

# Language

$$
\begin{array}{lcl}
0 & \rightarrow & \text{the } 1 \\
1 & \rightarrow & \text{dragon } 2 \\
1 & \rightarrow & \text{knight } 2 \\
1 & \rightarrow & \text{horse } 2 \\
2 & \rightarrow & \text{that } 0 \\
2 & \rightarrow & \epsilon\ 3 \\
3 & \rightarrow & \text{rode } 3 \\
3 & \rightarrow & \text{ate } 3 \\
3 & \rightarrow & \text{slew } 3 \\
3 & \rightarrow & \text{snores}
\end{array}
$$

**Example**

0 $\Rightarrow$ the 1 $\Rightarrow$ the knight 2 $\Rightarrow$ the knight that 0 $\Rightarrow$the knight that the 1 $\Rightarrow$ the knight that the horse 2 $\Rightarrow$ the knight that the horse 3 $\Rightarrow$ the knight that the horse snores $\in \Sigma^*$

Therefore, *the knight that the horse snores* $\in L(G_{\text{knight}})$

# Language

$$
\begin{array}{lll}
0 & \rightarrow & \text{the } 1 \\
1 & \rightarrow & \text{dragon } 2 \\
1 & \rightarrow & \text{knight } 2 \\
1 & \rightarrow & \text{horse } 2 \\
2 & \rightarrow & \text{that } 0 \\
2 & \rightarrow & \epsilon \ 3 \\
3 & \rightarrow & \text{rode } 3 \\
3 & \rightarrow & \text{ate } 3 \\
3 & \rightarrow & \text{slew } 3 \\
3 & \rightarrow & \text{snores}
\end{array}
$$

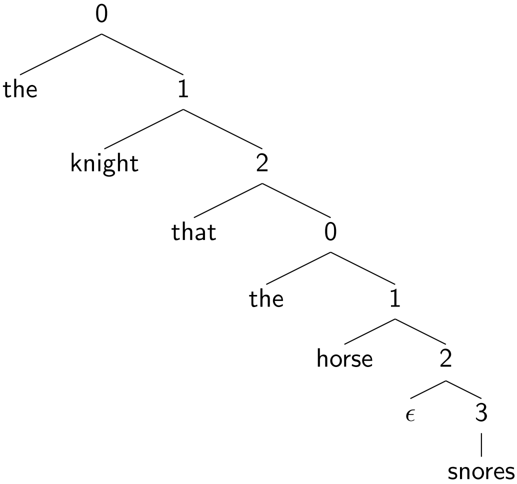
## Example

0 $\Rightarrow$ the 1 $\Rightarrow$ the knight 2 $\Rightarrow$ the knight that 0 $\Rightarrow$the knight that the 1 $\Rightarrow$ the knight that the horse 2 $\Rightarrow$ the knight that the horse 3 $\Rightarrow$ the knight that the horse snores $\in \Sigma^*$

Therefore, *the knight that the horse snores* $\in L(G_{\text{knight}})$

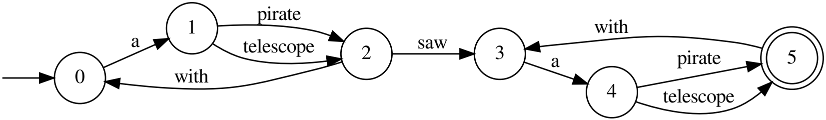
# Parse Trees

$0 \Rightarrow$ the $1 \Rightarrow$ the knight $2 \Rightarrow$ the knight that $0 \Rightarrow$the knight that the $1 \Rightarrow$ the knight that the horse $2 \Rightarrow$ the knight that the horse $3 \Rightarrow$ the knight that the horse snores



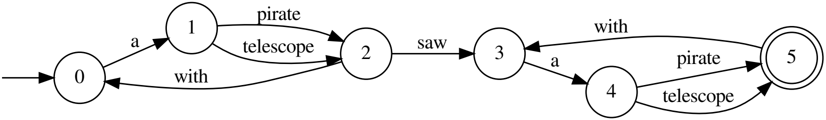
# Exercise: parse trees

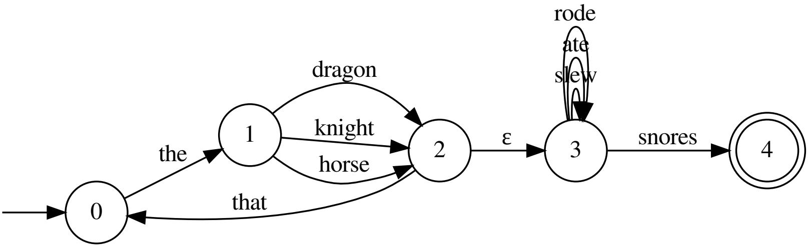
1. Rewrite $G_{pirate}$ as a rewrite grammar
2. Draw parse trees for a few sentences in $L(G_{pirate})$.



# Discussion

What are some conceptual and linguistic shortcomings of $G_{\text{pirate}}$ and $G_{\text{knight}}$?



## Rewrite Grammars

Recall rewrite grammar rule definition: Let $\alpha, \beta, \gamma \in (N \cup \Sigma)^*$ and $X \in N$.
A *rewrite grammar* has a set of rules R of the form

$\alpha X \beta \quad \rightarrow \quad \gamma$

But our rules all look like this:

$$\begin{array}{ccl} 1 & \rightarrow & \text{pirate } 2 \\ 5 & \rightarrow & \text{telescope} \\ \hline \text{NT}_1 & \rightarrow & \text{t } (\text{NT}_2) \end{array}$$

Even though possible rewrite rules include:

$$\begin{array}{rcl} 2\ 0\ \text{a with } 3 \text{ pirate} & \rightarrow & 5 \text{ saw with } 1 \\ 3 & \rightarrow & 1\ 2 \\ 2 \text{ telescope} & \rightarrow & \text{pirate} \end{array}$$

## Rewrite Grammars

Recall rewrite grammar rule definition: Let $\alpha, \beta, \gamma \in (N \cup \Sigma)^*$ and $X \in N$.
A *rewrite grammar* has a set of rules R of the form

$$\alpha X \beta \quad \rightarrow \quad \gamma$$

But our rules all look like this:

$$\begin{array}{rcl} 1 & \rightarrow & \text{pirate } 2 \\ 5 & \rightarrow & \text{telescope} \\ \hline \text{NT}_1 & \rightarrow & \text{t } (\text{NT}_2) \end{array}$$

Even though possible rewrite rules include:

$$\begin{array}{rcl} \text{2 0 a with 3 pirate} & \rightarrow & \text{5 saw with 1} \\ 3 & \rightarrow & 1\ 2 \\ \text{2 telescope} & \rightarrow & \text{pirate} \end{array}$$

# Rewrite Grammars

Recall rewrite grammar rule definition: Let $\alpha, \beta, \gamma \in (N \cup \Sigma)^*$ and $X \in N$.
A *rewrite grammar* has a set of rules R of the form

$$\alpha X \beta \quad \to \quad \gamma$$

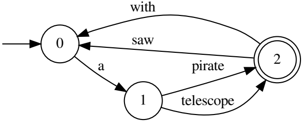
But our rules all look like this:

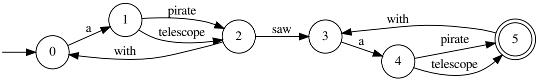
$$\begin{array}{rcl} 1 & \to & \text{pirate 2} \\ 5 & \to & \text{telescope} \\ \hline \text{NT}_1 & \to & \text{t (NT}_2\text{)} \end{array}$$

Even though possible rewrite rules include:

$$\begin{array}{rcl} \text{2 0 a with 3 pirate} & \to & \text{5 saw with 1} \\ 3 & \to & \text{1 2} \\ \text{2 telescope} & \to & \text{pirate} \end{array}$$

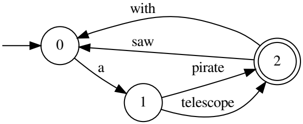
# Context Free Grammars

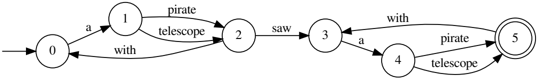



| | | | | | |
|---|---|---|---|---|---|
| 0 | $\rightarrow$ | a 1 | DP | $\rightarrow$ | D NP |
| 3 | $\rightarrow$ | a 4 | | | |
| | | | D | $\rightarrow$ | a |
| 1 | $\rightarrow$ | pirate 2 | NP | $\rightarrow$ | N (PP) |
| 4 | $\rightarrow$ | pirate 5 | | | |
| 1 | $\rightarrow$ | telescope 2 | | | |
| 4 | $\rightarrow$ | telescope 5 | | | |
| | | | N | $\rightarrow$ | pirate \| telescope |
| 2 | $\rightarrow$ | with 0 | PP | $\rightarrow$ | P DP |
| 5 | $\rightarrow$ | with 3 | | | |
| | | | P | $\rightarrow$ | with |

# Context Free Grammars



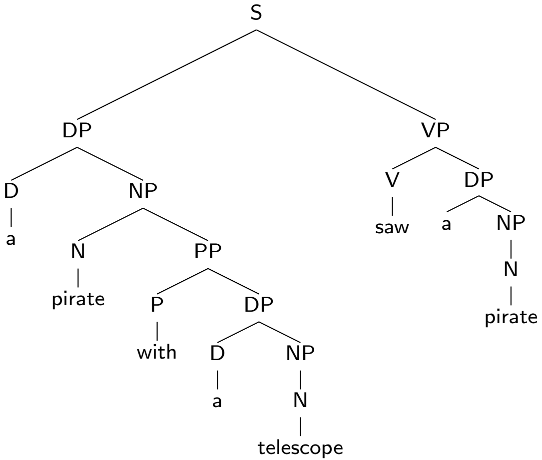




$$
\begin{array}{rcl|rcl}
2 & \rightarrow & \text{saw } 3 & \text{VP} & \rightarrow & \text{V DP} \\
 & & & \text{S} & \rightarrow & \text{DP VP} \\
 & & & \text{V} & \rightarrow & \text{saw}
\end{array}
$$

# Context Free Grammars

**$\mathbf{G_{pirate\text{-}CFG}}$**

| | | |
|---|---|---|
| DP | $\rightarrow$ | D NP |
| D | $\rightarrow$ | a |
| NP | $\rightarrow$ | N (PP) |
| N | $\rightarrow$ | pirate \| telescope |
| PP | $\rightarrow$ | P DP |
| P | $\rightarrow$ | with |
| VP | $\rightarrow$ | V DP |
| S | $\rightarrow$ | DP VP |
| V | $\rightarrow$ | saw |




**Discussion:** How is this parse tree different from the one for the same sentence using $\mathrm{G_{pirate\text{-}FSA}}$?

# Chomksy Hierarchy

Recall rewrite grammar rule definition: Let $\alpha, \beta \in (N \cup \Sigma)^*$. A *rewrite grammar* has a set of rules R of the form

$\alpha \quad \rightarrow \quad \beta$

So far we've seen:

| | | | | | | |
|---|---|---|---|---|---|---|
| NT | $\rightarrow$ | t (NT) | all rules | NT | $\rightarrow$ | t NT |
| 0 | $\rightarrow$ | a 1 | OR all rules | NT | $\rightarrow$ | NT t |
| 4 | $\rightarrow$ | pirate | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| NT | $\rightarrow$ | NT NT | all rules | NT | $\rightarrow$ | $\beta$ |
| S | $\rightarrow$ | DP VP | | | | |
| NT | $\rightarrow$ | t | | | | |
| N | $\rightarrow$ | pirate | | | | |

# Chomksy Hierarchy

Recall rewrite grammar rule definition: Let $\alpha, \beta \in (N \cup \Sigma)^*$. A *rewrite grammar* has a set of rules R of the form

$\alpha \quad \rightarrow \quad \beta$

So far we've seen:

**Regular Grammar**

| | | | | | | |
|---|---|---|---|---|---|---|
| NT | $\rightarrow$ | t (NT) | all rules | NT | $\rightarrow$ | t NT |
| 0 | $\rightarrow$ | a 1 | OR all rules | NT | $\rightarrow$ | NT t |
| 4 | $\rightarrow$ | pirate | | | | |

**Context Free Grammar**

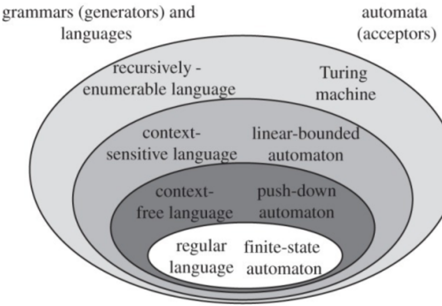
| | | | | | | |
|---|---|---|---|---|---|---|
| NT | $\rightarrow$ | NT NT | all rules | NT | $\rightarrow$ | $\beta$ |
| S | $\rightarrow$ | DP VP | | | | |
| NT | $\rightarrow$ | t | | | | |
| N | $\rightarrow$ | pirate | | | | |

# Chomsky Hierarchy

| Grammar | Language | Rewrite rules | Automaton |
|---|---|---|---|
| Type 0 | Recursively Enumerable | $\alpha X \beta \rightarrow \gamma$ | Turing machine |
| Type 1 | Context Sensitive | $\alpha X \beta \rightarrow \alpha \gamma \beta$ | Linear bounded non-deterministic Turing machine |
| Type 2 | Context Free | $X \rightarrow \gamma$ | Push-down Automaton |
| Type 3 | Regular | $X \rightarrow t\ (X)$<br>or $X \rightarrow (X)\ t$ | Finite State Automaton |

$\alpha, \beta, \gamma \in (N \cup \Sigma)^*$, $X \in N$, $t \in \Sigma$

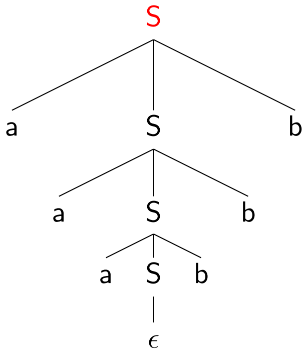
# Chomsky Hierarchy




the traditional Chomsky hierarchy

# Context-free grammars

aaabbb




Example: $a^n b^n$

1. S→a S b
2. S→$\epsilon$

**Question:** *Can you generate $a^n b^n$ with a regular grammar?*

# Weak Generative Capacity

**Definition**

The *weak generative capacity* of a (class of) grammar(s) is the (class of) string language(s) it can generate

**Example: The language $a^n b^n$**

- $a^n b^n = \{a^n b^n \mid n \in \mathbb{N}\} = \{\epsilon, ab, aabb, aaabbb, aaaabbbb, \ldots\}$
- $a^n b^n$ is strictly context-free language (i.e. it's not regular).
- The weak generative capacity of context free grammars, but not of regular grammars, includes $a^n b^n$.

# Weak Generative Capacity

### Definition

The *weak generative capacity* of a (class of) grammar(s) is the (class of) string language(s) it can generate

### Example: The language $a^n b^n$

- $a^n b^n = \{a^n b^n \mid n \in \mathbb{N}\} = \{\epsilon, ab, aabb, aaabbb, aaaabbbb, \ldots\}$
- $a^n b^n$ is strictly context-free language (i.e. it's not regular).
- The weak generative capacity of context free grammars, but not of regular grammars, includes $a^n b^n$.

# Weak Generative Capacity

**Definition**

The *weak generative capacity* of a (class of) grammar(s) is the (class of) string language(s) it can generate

**Example: The language $a^n b^n$**

- $a^n b^n = \{a^n b^n \mid n \in \mathbb{N}\} = \{\epsilon, ab, aabb, aaabbb, aaaabbbb, \ldots\}$
- $a^n b^n$ is strictly context-free language (i.e. it's not regular).
- The weak generative capacity of context free grammars, but not of regular grammars, includes $a^n b^n$.

# Strong Generative Capacity

The *strong generative capacity* of a (class of) grammar(s) is the (class of) derivations/parses/structural descriptions of its language(s).

Exercise: Do $G_1$ and $G_2$ have the same weak and/or strong generative capacity?

| $G_1$ | | |
|---|---|---|
| S | $\rightarrow$ | A B |
| A | $\rightarrow$ | a A |
| A | $\rightarrow$ | $\epsilon$ |
| B | $\rightarrow$ | b B |
| B | $\rightarrow$ | $\epsilon$ |

| $G_2$ | | |
|---|---|---|
| S | $\rightarrow$ | a S |
| S | $\rightarrow$ | b B |
| B | $\rightarrow$ | b B |
| S | $\rightarrow$ | $\epsilon$ |
| B | $\rightarrow$ | $\epsilon$ |