UNRESTRICTED GRAMMARS
AND TURING MACHINES

Abhijit Das

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

May 3, 2020
# The Chomsky Hierarchy

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Languages</th>
<th>Automata</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 3 / Right-linear</td>
<td>Regular</td>
<td>DFA / NFA</td>
<td>$A \rightarrow aB, A \rightarrow \varepsilon$</td>
</tr>
<tr>
<td>Type 2 / CFG</td>
<td>Context-free</td>
<td>PDA</td>
<td>$A \rightarrow \alpha$</td>
</tr>
<tr>
<td>Type 1 / CSG</td>
<td>Context-sensitive</td>
<td>LBA</td>
<td>$\alpha A \gamma \rightarrow \alpha \beta \gamma,</td>
</tr>
<tr>
<td>Type 0 / Unrestricted</td>
<td>Recursively enumerable</td>
<td>Turing machines</td>
<td>$\alpha A \gamma \rightarrow \beta$</td>
</tr>
</tbody>
</table>
Unrestricted Grammars

• $G = (\Sigma, N, S, P)$, where
  • $\Sigma$ is the set of terminal symbols,
  • $N$ is the set of non-terminal symbols ($N \cap \Sigma = \emptyset$),
  • $S \in N$ is the start symbol, and
  • $P$ is a finite set of rules or productions.

• Each production is of the form
  \[ \alpha \rightarrow \beta \]
  for any $\alpha, \beta \in (N \cup \Sigma)^*$ with $\alpha$ containing at least one non-terminal symbol.

• Such a production can also be written as
  \[ \gamma A \delta \rightarrow \beta \]
  for any $\beta, \gamma, \delta \in (N \cup \Sigma)^*$, and for any $A \in N$.

• $L(G) = \{ w \in \Sigma^* \mid S \rightarrow^*_G w \}$. 
Example 1

- \( L_1 = \{a^{2^n} \mid n \geq 0\} \).
- Productions:
  
  \[
  
  \begin{align*}
  S & \rightarrow TaU \\
  U & \rightarrow \varepsilon \mid AU \\
  aA & \rightarrow Aaa \\
  TA & \rightarrow T \\
  T & \rightarrow \varepsilon
  \end{align*}
  \]

- Derivation of \( a^8 \) using these productions:
  
  \[
  
  \begin{align*}
  S & \rightarrow TaU \rightarrow TaAU \rightarrow TaAAU \rightarrow TaAAAAU \rightarrow TaAAA \\
  & \rightarrow TAaaAA \rightarrow TaaAA \\
  & \rightarrow TaAaaA \rightarrow TAaaaaA \rightarrow TaaaaA \\
  & \rightarrow TaAAAa \rightarrow TaAaaaaaA \rightarrow TaAaaaaaaA \rightarrow TAaaaaaaaA \rightarrow TaaaaaaaA \\
  & \rightarrow aaaaaaaa
  \end{align*}
  \]
Example 2

- $L_2 = \{a^n b^n c^n \mid n \geq 0\}$.

- Productions:

  $S \rightarrow UT$
  $U \rightarrow \varepsilon \mid aUbC$
  $Cb \rightarrow bC$
  $CT \rightarrow Tc$
  $T \rightarrow \varepsilon$

- Derivation of $a^3 b^3 c^3$ using these productions:

  $S \rightarrow UT \rightarrow aUUbCT \rightarrow aaaUbCbCbCT \rightarrow aaabCbCbCT$
  $\rightarrow aaabCbCcCCT \rightarrow aaabbCbCCT \rightarrow aaabbbCCCT$
  $\rightarrow aaabbbCCTc \rightarrow aaabbbCTcc \rightarrow aaabbbTccc$
  $\rightarrow aaabbbccc$
Theorem

Given an unrestricted grammar $G$, there exists a Turing machine $M$ such that $L(M) = L(G)$.

Theorem

Given a Turing machine $M$, there exists an unrestricted grammar $G$ such that $L(G) = L(M)$. 
To construct a TM $M$ from an unrestricted grammar $G$.

$M$ is designed as a four-tape nondeterministic machine.

The input is provided to the first tape. It is never changed.

The second tape contains sentential forms in the derivation process. It is initialized by the symbol $S$.

$M$ keeps on repeating:

- Nondeterministically choose a position on the second tape.
- Nondeterministically choose a production $\alpha \rightarrow \beta$ of $G$.
- Copy $\alpha$ to Tape 3 and $\beta$ to Tape 4.
- Compare Tape 2 with Tape 3 starting from the position chosen for Tape 2.
- If the comparison succeeds, replace $\alpha$ by $\beta$ on Tape 2 after shifting the contents following $\alpha$ on Tape 2 if $|\alpha| \neq |\beta|$.
- Compare Tape 1 with Tape 2. If they have identical contents, accept.

$M$ is not necessarily a total TM.
To construct an unrestricted grammar $G$ from a TM $M$.

Assume that $M$ is a one-tape deterministic machine.

First, make some changes to $M$.

We want $M$ to halt with an empty tape after accepting.

Add a new accept state $t'$.

When $M$ reaches the old accept state, it erases the entire tape, and after seeing the left endmarker $\triangleright$, jumps to $t'$.

$M$ must know how much of the tape is used.

So $M$ uses a right endmarker $\triangleleft$.

This marker is shifted right if $M$ wants to extend the used portion of the tape.

During erasing at state $t$, this marker is moved left until it touches the left endmarker.
Turing Machine to Unrestricted Grammar

- $G$ simulates the working of $M$ from end to beginning.
- The configurations of $M$ are the sentential forms.
- On input $w$, the initial configuration of $M$ is $s \triangleright w \triangleleft$.
- The accepting configuration is $\triangleright t' \triangleleft$.
- The non-terminal symbols of $G$ consist of:
  - $\Gamma \setminus \Sigma$,
  - $Q$ (assume that $Q \cap \Gamma = \emptyset$).
  - A new start symbol $S$ not covered by the above two.
- Add the rule $S \rightarrow \triangleright t' \triangleleft$.
- Add the rules $s \triangleright \rightarrow \varepsilon$ and $\triangleleft \rightarrow \varepsilon$. 

FLAT, Spring 2020

Abhijit Das
Turing Machine to Unrestricted Grammar

- Simulation of a right movement of $M$: $\delta(p, a) = (q, b, R)$.

  - $\cdots \underline{a} \underline{c} \cdots \rightarrow \cdots \underline{b} \underline{c} \cdots$

  - Add the rule $bq \rightarrow pa$.

- Simulation of a left movement of $M$: $\delta(p, a) = (q, b, L)$ (here $a \neq \succ$).

  - $\cdots \underline{c} \underline{a} \cdots \rightarrow \cdots \underline{c} \underline{b} \cdots$

  - For all $c \in \Gamma$, add the rule $qcb \rightarrow cpa$.

- $M$ accepts as $s \succ w \prec \rightarrow^* \succ t' \prec$.

- $G$ works as $S \rightarrow \succ t' \prec \rightarrow^* s \succ w \prec \rightarrow w \prec \rightarrow w$. 
1. Design unrestricted grammars for the following languages.

(a) \( \{a^{n^2} \mid n \geq 0\} \).

(b) \( \{a^n b^n c^n d^n \mid n \geq 0\} \).

(c) \( \{w \in \{a, b, c\}^* \mid \#a(w) = \#b(w) = \#c(w)\} \).

(d) \( \{ww \mid w \in \{a, b\}^*\} \).

(e) \( \{a^i b^j c^k d^l \mid i = k \text{ and } j = l\} \).

2. Consider the unrestricted grammar over the singleton alphabet \( \Sigma = \{a\} \), having the start symbol \( S \), and with the following productions.

\[
S \rightarrow AS \mid aT \\
Aa \rightarrow aaaA \\
AT \rightarrow T \\
T \rightarrow \varepsilon
\]

What is the language generated by this unrestricted grammar? Justify.
3. Prove that any grammar can be converted to an equivalent grammar with rules of the form $\alpha A \gamma \to \alpha \beta \gamma$ for $A \in N$ and $\alpha, \beta, \gamma \in (\Sigma \cup N)^*$.  

4. Write context-sensitive grammars for the following languages.  

(a) $\{a^{2^n} \mid n \geq 0\}$.  
(b) $\{a^n b^n c^n \mid n \geq 1\}$.  
