UNRESTRICTED GRAMMARS
AND TURING MACHINES

Abhijit Das
Sudeshna Kolay

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

March 15, 2022

The Chomsky Hierarchy

Grammar	**Languages**	**Automata**	**Rules**
Type 3 / Right-linear	Regular	DFA / NFA	$A \to aB$, $A \to \varepsilon$
Type 2 / CFG	Context-free	PDA	$A \to \alpha$
Type 1 / CSG	Context-sensitive	LBA	$\alpha A \gamma \to \alpha\beta\gamma$, $\lvert\beta\rvert > 0$
Type 0 / Unrestricted	Recursively enumerable	Turing machines	$\alpha A \gamma \to \beta$

Unrestricted Grammars

- $G = (\Sigma, N, S, P)$, where
  - Σ 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 α 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 $A \in N$.
- $\mathscr{L}(G) = \{w \in \Sigma^* \mid S \rightarrow_G^* w\}$.

Example 1

- $L_1 = \{a^{2^n} \mid n \geqslant 0\}$.
- Productions:

$$
\begin{aligned}
&S \to TaU \\
&U \to \varepsilon \mid AU \\
&aA \to Aaa \\
&TA \to T \\
&T \to \varepsilon
\end{aligned}
$$

- Derivation of a^8 using these productions:

$$
\begin{aligned}
&S \to TaU \to TaAU \to TaAAU \to TaAAAU \to TaAAA \\
&\quad \to TAaaAA \to TaaAA \\
&\quad \to TaAaaA \to TAaaaaA \to TaaaaA \\
&\quad \to TaaaAaa \to TaaAaaaa \to TaAaaaaaa \to TAaaaaaaaa \to Taaaaaaaa \\
&\quad \to aaaaaaaa
\end{aligned}
$$

Example 2

- $L_2 = \{a^n b^n c^n \mid n \geqslant 0\}$.
- Productions:

$$\begin{aligned} &S \to UT \\ &U \to \varepsilon \mid aUbC \\ &Cb \to bC \\ &CT \to Tc \\ &T \to \varepsilon \end{aligned}$$

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

$$\begin{aligned} &S \to UT \to aUbCT \to aaUbCbCT \to aaaUbCbCbCT \to aaabCb{\color{red}Cb}CT \\ &\quad \to aaab{\color{red}Cb}bCCT \to aaabb{\color{red}Cb}CCT \to aaabbbCC{\color{red}CT} \\ &\quad \to aaabbbC{\color{red}CT}c \to aaabbb{\color{red}CT}cc \to aaabbb{\color{red}T}ccc \\ &\quad \to aaabbbccc \end{aligned}$$

Unrestricted Grammars and Turing Machines

Theorem

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

Theorem

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

Unrestricted Grammar to Turing Machine

- 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 $\alpha \to \beta$ of G.
  - Copy α to Tape 3 and β to Tape 4.
  - Compare Tape 2 with Tape 3 starting from the position chosen for Tape 2.
  - If the comparison succeeds, replace α by β on Tape 2 after shifting the contents following α 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.

Turing Machine to Unrestricted Grammar

- 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 $\vdash$, jumps to t'.
- M must know how much of the tape is used.
- So M uses a right endmarker $\dashv$.
- 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 $s \vdash w \dashv$.
- The accepting configuration is $\vdash t' \dashv$.
- 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 $S \rightarrow \vdash t' \dashv$.
- Add the rules $s \vdash \rightarrow \varepsilon$ and $\dashv \rightarrow \varepsilon$.

Turing Machine to Unrestricted Grammar

- Simulation of a right movement of M: $\delta(p,a) = (q,b,R)$.
- $\cdots \underline{\boxed{a}}\boxed{c} \cdots \qquad \to \qquad \cdots \boxed{b}\underline{\boxed{c}} \cdots$
- Add the rule $\quad bq \to pa$.

- Simulation of a left movement of M: $\delta(p,a) = (q,b,L)$ (here $a \neq \vdash$).
- $\cdots \boxed{c}\underline{\boxed{a}} \cdots \qquad \to \qquad \cdots \underline{\boxed{c}}\boxed{b} \cdots$
- For all $c \in \Gamma$, add the rule $\quad qcb \to cpa$.

- M accepts as $s \vdash w \dashv \ \to^* \ \vdash t' \dashv$.
- G works as $S \ \to \ \vdash t' \dashv \ \to^* \ \ s \vdash w \dashv \ \to \ \ w \dashv \ \to \ w$.

Tutorial Exercises

1. Design unrestricted grammars for the following languages.

(a) $\{a^{n^2} \mid n \geqslant 0\}$.
(b) $\{a^n b^n c^n d^n \mid n \geqslant 0\}$.
(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.

$$S \to AS \mid aT \qquad Aa \to aaaA \qquad AT \to T \qquad T \to \varepsilon$$

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