UNRESTRICTED GRAMMARS AND TURING MACHINES

Abhijit Das

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

May 3, 2020

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	$\alpha A \gamma \to \alpha\beta\gamma$, $\|\beta\| > 0$
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
$$\alpha \to \beta$$
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
$$\gamma A \delta \to \beta$$
for any $\beta, \gamma, \delta \in (N \cup \Sigma)^*$, and for any $A \in N$.
- $\mathscr{L}(G) = \{w \in \Sigma^* \mid S \to_G^* w\}$.

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:

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

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:

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

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 $\triangleright$, 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 $s \rhd w \lhd$.
- The accepting configuration is $\rhd t' \lhd$.
- 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 $S \to \rhd t' \lhd$.
- Add the rules $s \rhd \to \varepsilon$ and $\lhd \to \varepsilon$.

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 $a \neq \rhd$).
- $\cdots \boxed{c}\underline{\boxed{a}} \cdots \qquad \to \qquad \cdots \underline{\boxed{c}}\boxed{b} \cdots$
- For all $c \in \Gamma$, add the rule $\quad qcb \to cpa$.

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

Unrestricted Grammars: Special Forms of Productions

Lemma: Any grammar can be converted s.t productions are of the form
$\alpha A \gamma \to \alpha \beta \gamma$, $A \in N$, $\alpha, \beta, \gamma \in (\Sigma \cup N)^*$.

Proof:
- Introduce non-terminal T_a for each $a \in \Sigma$, add $T_a \to a$.
- Replace occurrences of a with T_a. So, $\alpha \to \beta \Rightarrow U_1 \ldots U_m \to V_1 \ldots V_n$.
- For each $\alpha \to \beta$ $(U_1 \ldots U_m \to V_1 \ldots V_n)$ introduce new $W_1, \ldots, W_m \in N$.
- New rules:

$m \leq n$:

$$U_1 U_2 \ldots U_m \to W_1 U_2 \ldots U_m$$
$$W_1 U_2 \ldots U_m \to W_1 W_2 \ldots U_m$$
$$\vdots$$
$$W_1 \ldots W_{m-1} U_m \to W_1 \ldots W_{m-1} W_m V_{m+1} V_{m+2} \ldots V_n$$
$$W_1 W_2 \ldots W_m V_{m+1} \ldots V_n \to V_1 W_2 \ldots W_m V_{m+1} \ldots V_n$$
$$\vdots$$
$$V_1 V_2 \ldots W_m V_{m+1} \ldots V_n \to V_1 V_2 \ldots V_m V_{m+1} \ldots V_n$$

$m \geq n$:

$$U_1 U_2 \ldots U_m \to W_1 U_2 \ldots U_m$$
$$\vdots$$
$$W_1 W_2 \ldots W_{m-1} U_m \to W_1 W_2 \ldots W_{m-1} W_m$$
$$W_1 W_2 \ldots W_n W_{n+1} W_{n+2} \ldots W_m \to W_1 W_2 \ldots W_n W_{n+2} \ldots W_m$$
$$\vdots$$
$$W_1 W_2 \ldots W_n \to V_1 W_2 \ldots W_n$$
$$\vdots$$
$$V_1 \ldots W_n \to V_1 \ldots V_{n-1} V_n$$