5

8.4. Context-Sensitive Languages and the Chomsky Hierarchy

Definition 8.16. Context-Sensitive Grammars

A context-sensitive grammar (CSG) is an unrestricted grammar in which no production is length-decreasing. In other words, every production is of the form $\alpha \rightarrow \beta$, where $|\beta| \geq |\alpha|$.

A language is a context-sensitive language (CSL) if it can be generated by a context-sensitive grammar.

Example 8.12. A Grammar Generating $\{a^n b^n c^n | n \geq 1\}$

$$S \rightarrow SABC \mid LABC$$

$$BA \rightarrow AB \ CB \rightarrow BC \ CA \rightarrow AC$$

$$LA \rightarrow a aA \rightarrow aa \ aB \rightarrow ab \ bB \rightarrow bb \ bC \rightarrow bc \ cC \rightarrow cc$$

Not context-sensitive.
Example 8.17. A CSG Generating
$$L = \{ anbn|n \geq 1 \}$$

S \rightarrow SABC | A

B \rightarrow AB CB
C \rightarrow BC CA
A \rightarrow AC

9

Definition 8.10. Linear-Bounded Automata
A linear-bounded automaton (LBA) is a 5-tuple $M = (Q, \Sigma, \Gamma, q_0, \delta)$ that is identical to a nondeterministic Turing machine, with the following exception.

There are two extra tape symbols [ and ] assumed not to be elements of the tape alphabet $\Gamma$.

The initial configuration of $M$ corresponding to input $x$ is $q_0[|x|, [\text{first square to the right of } x].$

During its computation, $M$ is not permitted to replace either of these brackets or to move its tape head to the left of the [ or to the right of the ].

Theorem 8.19. If $L \subseteq \Sigma^*$ is a context-sensitive language, then there is a linear-bounded automaton that accepts $L$.

Proof. Much like the proof of Theorem 8.13, except

• two tape tracks instead of move past input
• reject also if we want to write on ]

Theorem 8.13. For every unrestricted grammar $G$, there is a Turing machine $T$ with $L(T) = L(G)$.

Proof. 1. Generate (every possible) input string for $T$(two copies),
2. Simulate computation of $T$ for this input string as derivation in $G$ on the tape of a Turing machine
3. If $T$ reaches accept state, reconstruct original input string.

Ad 2. Move $\delta(p, a) = (q, b, R)$ of $T$ yields production $p(\sigma_1a) \rightarrow (\sigma_1b)q$

Ad 3. Propagate $\sigma_1$ all over the string $p(\sigma_1\sigma_2) \rightarrow \sigma_1$, for $\sigma_1 \in \Sigma$

Theorem 8.20. If $L \subseteq \Sigma^*$ is accepted by a linear-bounded automaton $M = (Q, \Sigma, \Gamma, q_0, \delta)$, then there is a context-sensitive grammar $G$ generating $L - \{\Lambda\}$.

Proof. Much like the proof of Theorem 8.14, except

• consider $\sigma_1\sigma_2$ as a single symbol
• no additional (∆∆)'s needed
• incorporate [ and ] in leftmost/rightmost symbols of string.
Chomsky hierarchy

3

regular languages
regular grammar
FA

2

context-free languages
context-free grammar
PDA

1

context-sensitive languages
context-sensitive grammar
LBA

0

recursive languages
unrestricted grammar
TM

Theorem 8.22.
Every context-sensitive language is recursive.

Proof...

Definition 8.23.
A set \( A \) of the same size as \( B \) or larger than \( B \). Two sets \( A \) and \( B \), either finite or infinite, are the same size if there is a bijection \( f: A \to B \). A set \( A \) is larger than \( B \) if some subset of \( A \) is as large as \( B \) but \( A \) itself is not.

Definition 8.24.
Countably infinite and countable sets. A set \( A \) is countably infinite (the same size as \( \mathbb{N} \)) if there is a bijection \( f: \mathbb{N} \to A \), or a list \( a_0, a_1, \ldots \) of elements of \( A \) such that every element of \( A \) appears exactly once in the list. A set \( A \) is countably infinite (the same size as \( \mathbb{N} \)) if there is a bijection \( f: \mathbb{N} \to A \).

Theorem 8.25.
Every infinite set has a countably infinite subset, and every subset of a countably infinite set is countable.

Example 8.26.
The set \( \mathbb{N} \times \mathbb{N} \) is countable.

Remark. (modulo \( \Lambda \))

### Table

<table>
<thead>
<tr>
<th>Class</th>
<th>Language Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>unrestricted</td>
</tr>
<tr>
<td>1</td>
<td>context-sensitive</td>
</tr>
<tr>
<td>2</td>
<td>context-free</td>
</tr>
<tr>
<td>3</td>
<td>regular</td>
</tr>
<tr>
<td>4</td>
<td>regular expression</td>
</tr>
</tbody>
</table>

Chomsky hierarchy

What about recursive languages?

Proof...
Example 8.28. A Countable Union of Countable Sets Is Countable

\[ S = \bigcup_{i=0}^{\infty} S_i \]

Same construction as in Example 8.26, but...

Example 8.29. Languages Are Countable Sets

\[ L \subseteq \Sigma^* = \bigcup_{i=0}^{\infty} \Sigma_i \]

Two ways to list \( \Sigma^* \)

Example 8.30. The Set of Turing Machines Is Countable

Let \( T \) be the set of Turing machines. There is an injective function \( e : T \rightarrow \{0, 1\}^* \) (the encoding function). Hence, the set of recursively enumerable languages is countable.

Example 8.31. The Set of Languages Is Uncountable (continued)

There are uncountably many languages over \( \{0, 1\} \). Hence, because \( N \) and \( \{0, 1\} \) are the same size, the set of languages is uncountable.

Example 8.31. The Set of Languages Is Uncountable (continued)

\( A = \{ i \in N \mid i \notin A_i \} \)

For every set \( A_0, A_1, A_2, \ldots \) of subsets of \( N \) that is complete, there is at least one subset of \( N \) that is not in the list.

Example 8.31. The Set of Languages Is Uncountable (continued)

\( A = \{ i \in N \mid i \notin A_i \} \).

There are uncountably many languages over \( \{0, 1\} \).

Example 8.31. The Set of Languages Is Uncountable (continued)

There are uncountably many languages over \( \{0, 1\} \).

Example 8.32. Not all languages are recursively enumerable.

\[ \text{The set of languages over } \{0, 1\} \text{ that are not recursively enumerable is uncountable.} \]

Proof...

Because there are uncountably many languages over \( \{0, 1\} \), there are uncountably many Turing machines. Let \( T \) be the set of Turing machines.

Example 8.32. Not all languages are recursively enumerable. (including Exercise 8.38)