CSE211-Formal Languages and Automata Theory

U3L1 – Context Sensitive Language and Linear Bound Automata

Dr. P. Saravanan
School of Computing
SASTRA Deemed University

Agenda

- Unit III syllabus
- Unrestricted Grammar
- Context-sensitive grammars
- Examples for Context-sensitive grammars
- Characteristics of Context-sensitive grammars
- Context-Sensitive Languages
- Derivations using Context-sensitive grammars
- Linear Bounded Automata
- CSG and LBA

Unit 3: Syllabus

- **Context-sensitive languages**: Context-sensitive grammars (CSG) and languages - linear bounded automata and equivalence with CSG (Textbook 2)
- **Introduction to Turing machines**: The Turing Machine (TM) - Church-Turing thesis - Programming Techniques for Turing Machines – extensions to the Basic Turing Machine – Restricted Turing Machine - Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators

Unrestricted Grammars

- An *unrestricted grammar* has essentially no restrictions on the form of its productions:
  - Any variables and terminals on the left side, in any order
  - Any variables and terminals on the right side, in any order
  - The only restriction is that λ or ε is not allowed as the left side of a production

$$u \rightarrow v$$

String of variables and terminals

String of variables and terminals

Unrestricted Grammars

- A sample unrestricted grammar has productions

$$S \rightarrow S_1B$$
$$S_1 \rightarrow aS_1b$$
$$bB \rightarrow bbbB$$
$$aS_1b \rightarrow aa$$
$$B \rightarrow \varepsilon$$

- Example 2:

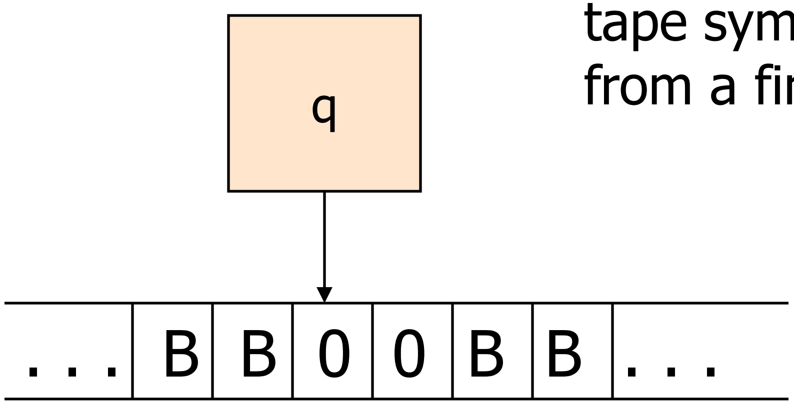
$$S \rightarrow aBc$$
$$aB \rightarrow cA$$
$$Ac \rightarrow d$$

Unrestricted Grammars

- Theorem: A language L is Turing-Acceptable if and only if L is generated by an unrestricted grammar
- Theorem: Any language generated by an unrestricted grammar is recursively enumerable
- Theorem: For every recursively enumerable language L, there exists an unrestricted grammar G that generates L

Turing Machine

- Action: based on the state and the tape symbol under the head: change state, rewrite the symbol and move the head one position either left or right



Infinite tape with
cells containing
tape symbols chosen
from a finite alphabet

$\delta(q, 0) = (q, 0, R)$

$\delta(q, 1) = (f, 0, R)$

$\delta(q, B) = (q, 1, L)$

Context-sensitive grammars

- In a context-sensitive grammar, the only restriction is that,
  - for any production, length of the right side is at least as large as the length of the left side

$$u \rightarrow v$$

String of variables and terminals

String of variables and terminals

and: $$|u| \leq |v|$$

Context Sensitive Grammar: Ex

- Ex: Context Sensitive Grammar

$S \rightarrow abc \mid aAbc$

$Ab \rightarrow bA$

$Ac \rightarrow Bbcc$

$bB \rightarrow Bb$

$aB \rightarrow aa \mid aaA$

Characteristics of CSG

- They are **noncontracting: I**n any derivation, the length of successive sentential forms can never decrease
- The variables may only be replaced in certain contexts (Hence, these grammars are called context-sensitive)
- For instance, in the grammar of Example variable A can only be replaced if it is followed by either b or c

$$
\begin{aligned}
S &\to abc \mid aAbc \\
Ab &\to bA \\
Ac &\to Bbcc \\
bB &\to Bb \\
aB &\to aa \mid aaA
\end{aligned}
$$

Context-Sensitive Languages

- A *language L is context-sensitive* if there is a context-sensitive grammar G, such that either L = L(G) or L = L(G) $\cup \{ \varepsilon \}$
- The family of context-free languages is a subset of the family of context-sensitive languages
- The language $\{ a^n b^n c^n : n \geq 1 \}$ is context-sensitive, since it is generated by the grammar

$$
\begin{aligned}
S &\to abc \mid aAbc \\
Ab &\to bA \\
Ac &\to Bbcc \\
bB &\to Bb \\
aB &\to aa \mid aaA
\end{aligned}
$$

Derivation of Strings Using CSG

- Using the grammar in Example, we derive the string aabbcc

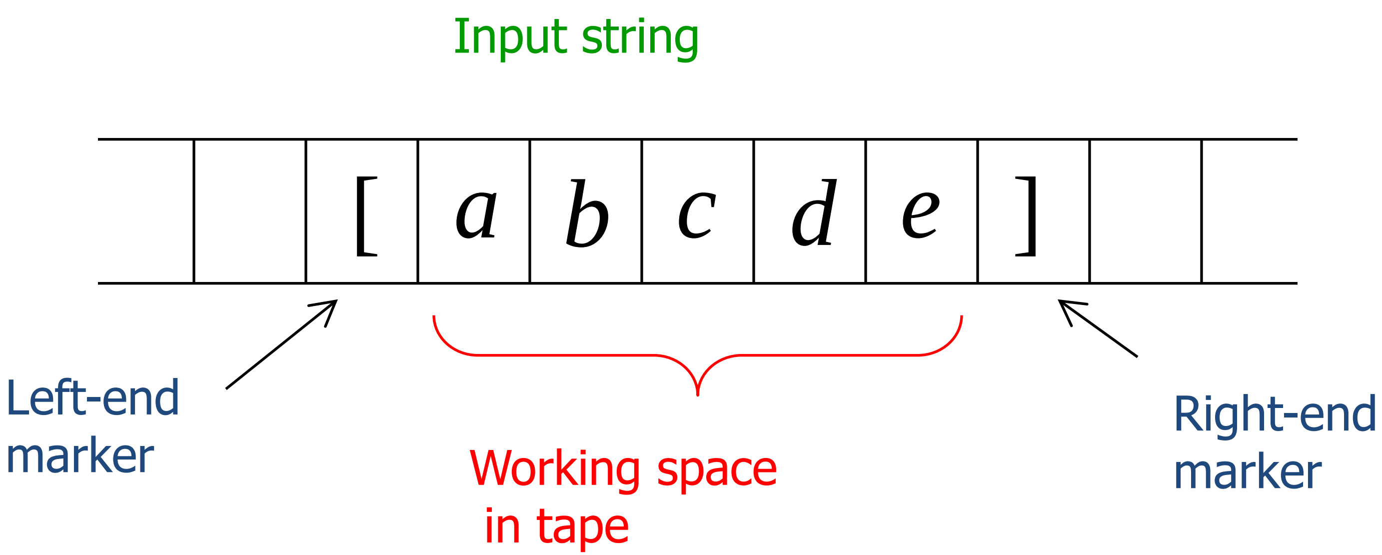
$$
\begin{aligned}
S &\Rightarrow aAbc \\
&\Rightarrow abAc \\
&\Rightarrow abBbcc \\
&\Rightarrow aBbbcc \\
&\Rightarrow aabbcc
\end{aligned}
$$

$$
\begin{aligned}
S &\rightarrow abc \mid aAbc \\
Ab &\rightarrow bA \\
Ac &\rightarrow Bbcc \\
bB &\rightarrow Bb \\
aB &\rightarrow aa \mid aaA
\end{aligned}
$$

- The variables A and B are effectively used as messengers:
    - an A is created on the left, travels to the right of the first c, where it creates another b and c, as well as variable B
    - the newly created B is sent to the left in order to create the corresponding a

Linear-Bounded Automata

- Same as Turing Machines with one difference:
  - The input string tape space is the only tape space allowed to use



All computation is done between end markers

CSL and LBA

- Theorem: For every context-sensitive language L not including λ, there is a linear bounded automaton that recognizes L
- Theorem: If a language L is accepted by a linear bounded automaton M, then there is a context-sensitive grammar that generates L
- Context-sensitive grammars generate exactly the family of languages accepted by linear bounded automata, the context-sensitive languages

Relationship B/w Recursive and Context-Sensitive Languages

- Theorem: Every context-sensitive language is recursive
- Theorem: Some recursive languages are not context-sensitive
- These two theorems help establish a hierarchical relationship among the various classes of automata and languages:
  - Linear bounded automata are less powerful than Turing machines
  - Linear bounded automata are more powerful than pushdown automata

Summary

- Unit III syllabus
- Unrestricted Grammar
- Context-sensitive grammars
- Examples for Context-sensitive grammars
- Characteristics of Context-sensitive grammars
- Context-Sensitive Languages
- Derivations using Context-sensitive grammars
- Linear Bounded Automata
- CSG and LBA

References

- John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, *Introduction to Automata Theory*, Languages, and Computation, Pearson, 3rd Edition, 2011.
- Peter Linz, An Introduction to Formal Languages and Automata, Jones and Bartle Learning International, United Kingdom, 6th Edition, 2016.

Next Class: **Unit III**

Turing Machines

Thank you.