`

The Chomsky Hierarchy

Shakir Ullah Shah

Chomsky Hierarchy

- Type-0 grammars (unrestricted grammars) include all formal grammars.
- Type-1 grammars (context-sensitive grammars) generate the context-sensitive languages.

The chomsky hierarchy

- Type-2 grammars (context-free grammars) generate the context-free languages.
- Context free languages are the theoretical basis for the syntax of most programming languages.

- Type-3 grammars (regular grammars) generate the regular languages.
- These languages are exactly all languages that can be decided by a finite state automaton. Additionally, this family of formal languages can be obtained by regular expressions.

Linear-Bounded Automata:

Same as Turing Machines with one difference:

the input string tape space is the only tape space allowed to use

Linear-Bounded Automata:

Turing machine.

- **No limit on length of tape.**

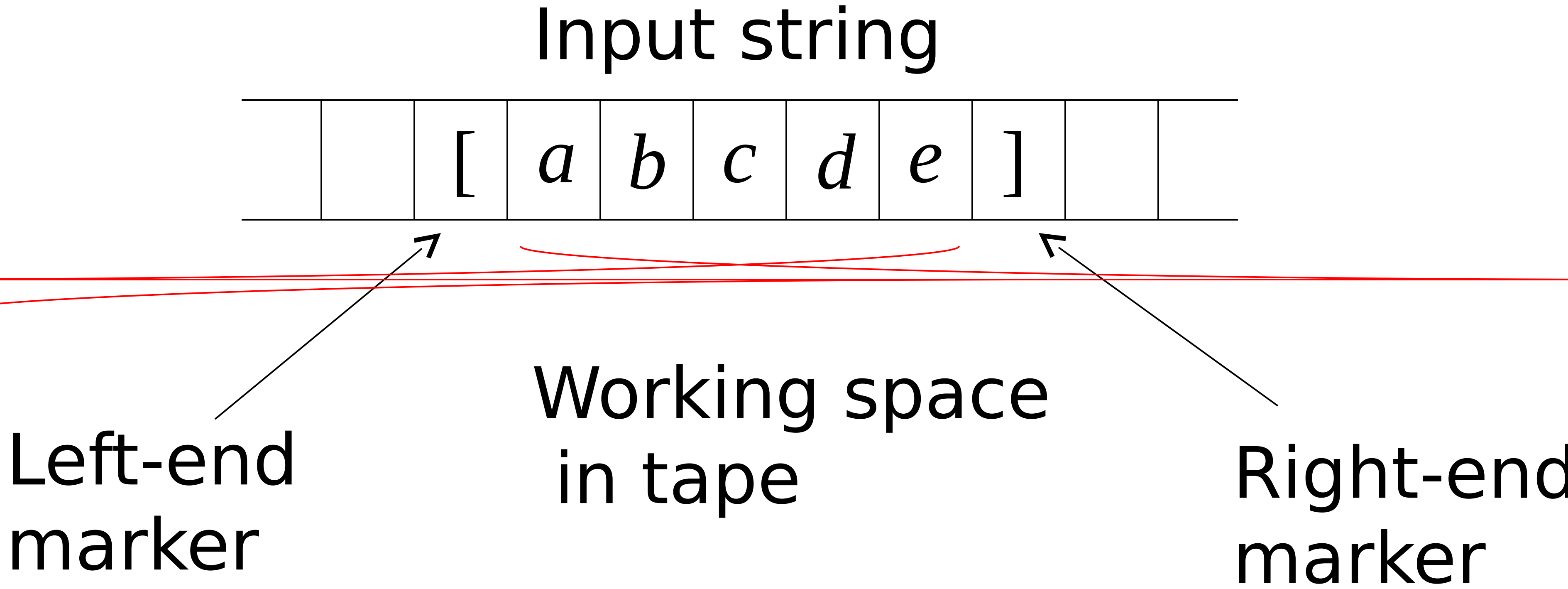
Linear bounded automata (LBA).

- **A single tape TM that can only write on the portion of the tape containing the input.**
- **Note: allowed to increase alphabet size if desired.**

LBA is strictly less powerful than TM.

- **There are languages that can be recognized by TM but not a LBA.**
- **We won’t dwell on LBA in this course.**

Linear Bounded Automaton (LBA)



ll computation is done between end markers

Example languages accepted by LBAs:

$$L = \{a^n b^n c^n\} \qquad L = \{a^{n!}\}$$

LBA's have more power than PDA's
(pushdown automata)

LBA's have less power than Turing Machines

Type-0 grammar (Unrestricted Grammar)

Productions

$$u \rightarrow v$$

String of variables and terminals

String of variables and terminals

Type-0 grammar (Unrestricted Grammar)

- They generate exactly all languages that can be recognized by a Turing machine.
- These languages are also known as the recursively enumerable languages.
- This is different from the recursive languages which can be *decided* by an always halting Turing machine.

Example unrestricted grammar:

$$S \rightarrow aBc$$

$$aB \rightarrow cA$$

$$Ac \rightarrow d$$

Theorem:

A language L is Turing-Acceptable if and only if L is generated by an unrestricted grammar

Context-Sensitive Grammars:

Type-1 grammar

Productions

$$u \rightarrow v$$

String of variables and terminals

String of variables and terminals

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

Context-Sensitive Grammars: Type-1 grammar

- The rule is allowed if *S* does not appear on the right side of any rule.

- The languages described by these grammars are exactly all languages that can be recognized by a non-deterministic Turing machine whose tape is bounded by a constant times the length of the input.

The language $\{a^n b^n c^n\}$

is context-sensitive:

$$S \to abc \mid aAbc$$

$$Ab \to bA$$

$$Ac \to Bbcc$$

$$bB \to Bb$$

$$aB \to aa \mid aaA$$

Theorem:

language L is context sensistive

if and only if

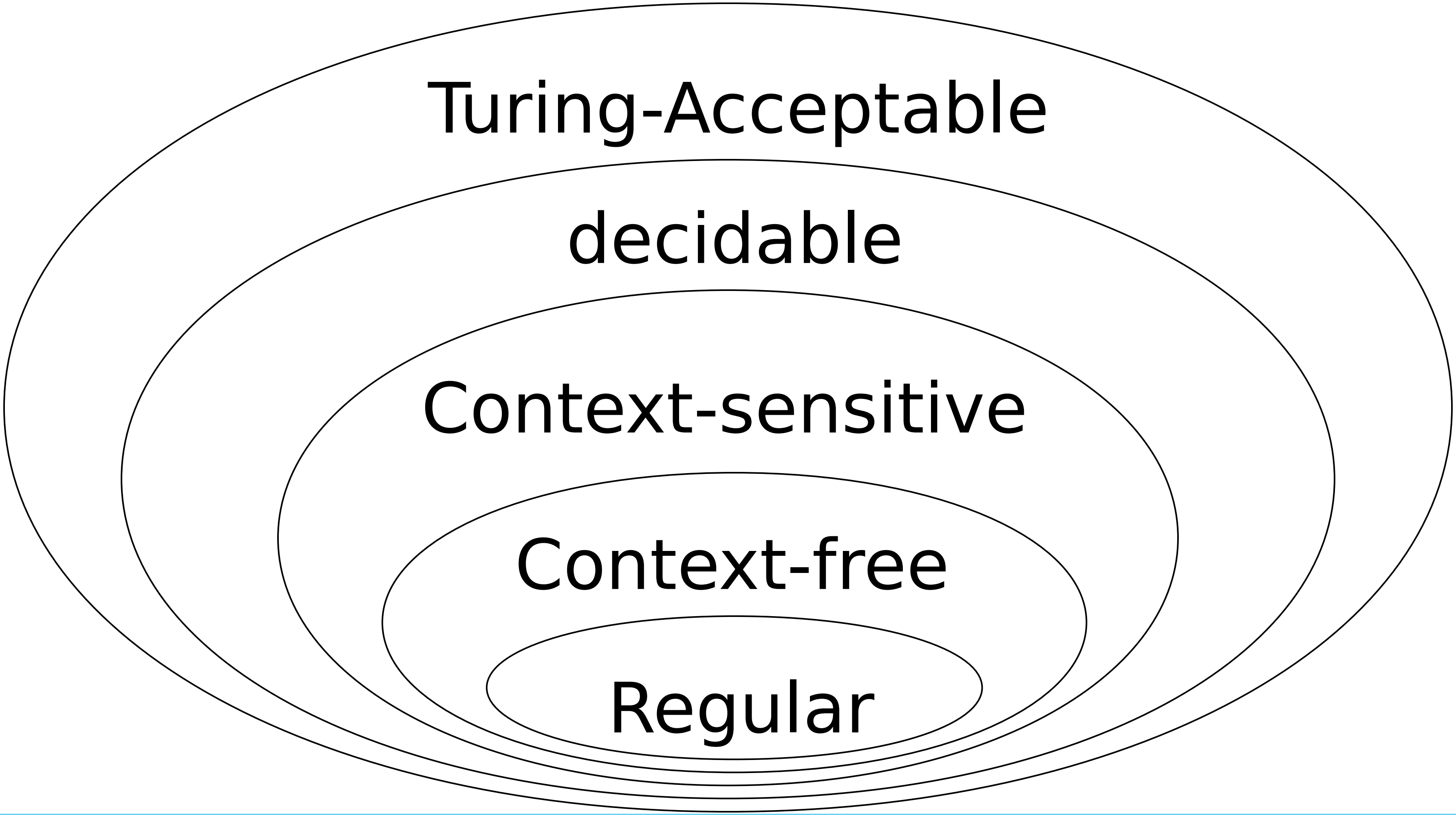
is accepted by a Linear-Bounded automator

Observation:

here is a language which is context-sensitive

ut not decidable

The Chomsky Hierarchy



Summary

Automata theory: formal languages and formal grammars			
Chomsky hierarchy	**Grammars**	**Languages**	**Minimal automaton**
Type-0	(unrestricted)	Recursively enumerable	Turing machine
	(unrestricted)	Recursive	Decider
Type-1	Context-sensitive	Context-sensitive	Linear-bounded
Type-2	Context-free	Context-free	Pushdown
Type-3	Regular	Regular	Finite
Each category of languages or grammars is a proper superset of the category directly beneath it.			