Theory of Machines and Languages

Fatemeh Deldar

1403-1404

Undecidability

- A language is ***co-Turing-recognizable*** if it is the complement of a Turing-recognizable language
- **Theorem**
  - A language is decidable iff it is Turing-recognizable and co-Turing-recognizable
  - **Proof**
    - let M_1 be the recognizer for A and M_2 be the recognizer for $\overline{A}$
    - The following Turing machine M is a decider for A

      $M =$ "On input w:
      1. Run both M_1 and M_2 on input w in parallel.
      2. If M_1 accepts, *accept*; if M_2 accepts, *reject*."
  - We show that M decides A

Undecidability

- $\overline{A_{TM}}$ is not Turing-recognizable
- **Proof**
  - We know that A_{TM} is Turing-recognizable
  - If $\overline{A_{TM}}$ also were Turing-recognizable, A_{TM} would be decidable ✗

Reducibility

- **A reduction is a way of converting one problem to another problem in such a way that *a solution to the second problem can be used to solve the first problem***
- **if A is reducible to B and B is decidable, A also is decidable**
- **if A is undecidable and reducible to B, B is undecidable**

The Turing Machine Halting Problem

- **The halting problem:**

$$HALT_{\mathsf{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ halts on input } w\}$$

- $\boldsymbol{HALT_{TM}}$ **is undecidable**
  - **Assume that TM $\boldsymbol{R}$ decides $\boldsymbol{HALT_{TM}}$**
  - **We construct TM $\boldsymbol{S}$ to decide $\boldsymbol{A_{TM}}$, with $\boldsymbol{S}$ operating as follows:**

    1. Run TM R on input $\langle M, w\rangle$.
    2. If R rejects, *reject*.
    3. If R accepts, simulate M on w until it halts.
    4. If M has accepted, *accept*; if M has rejected, *reject*.

  - **If $\boldsymbol{R}$ decides $\boldsymbol{HALT_{TM}}$, then $\boldsymbol{S}$ decides $\boldsymbol{A_{TM}}$** ✗

Unrestricted Grammars

- A grammar $G = (V, T, S, P)$ is called **unrestricted** if all the productions are of the form

$$u \to v,$$

where u is in $(V \cup T)^+$ and v is in $(V \cup T)^*$.

There is only one restriction: λ is not allowed as the left side of a production

Unrestricted Grammars

- **Any language generated by an unrestricted grammar is recursively enumerable**
- **For every recursively enumerable language L, there exists an unrestricted grammar G, such that $L = L(G)$**

The family of languages associated with unrestricted grammars is identical with the family of recursively enumerable languages

Unrestricted Grammars

- **Example**

  $S \rightarrow TaU$

  $U \rightarrow \lambda \mid AU$

  $aA \rightarrow Aaa$

  $TA \rightarrow T$

  $T \rightarrow \lambda$

$$L_1 = \{a^{2^n} \mid n \geqslant 0\}$$

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

Context-Sensitive Grammars and Languages

- A grammar $G = (V, T, S, P)$ is said to be **context sensitive** if all productions are of the form

$$x \rightarrow y,$$

where $x,\ y \in (V \cup T)^+$ and

$$|x| \leq |y| \, .$$

The length of successive sentential forms can never decrease

Context-Sensitive Grammars and Languages

- A language L is said to be context sensitive if there exists a context-sensitive grammar G, such that $L = L(G)$ or $L = L(G) \cup \{\lambda\}$.

- **Example**

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

$$L = \{a^n b^n c^n : n \geq 1\}$$

$$
\begin{aligned}
S &\Rightarrow aAbc \Rightarrow abAc \Rightarrow abBbcc \\
&\Rightarrow aBbbcc \Rightarrow aaAbbcc \Rightarrow aabAbcc \\
&\Rightarrow aabbAcc \Rightarrow aabbBbccc \\
&\Rightarrow aabBbbccc \Rightarrow aaBbbbccc \\
&\Rightarrow aaabbbccc.
\end{aligned}
$$

Context-Sensitive Grammars and Languages

Context-Sensitive Languages and Linear Bounded Automata

For every context-sensitive language L not including λ, there exists some linear bounded automaton M such that $L = L(M)$.

If a language L is accepted by some linear bounded automaton M, then there exists a context-sensitive grammar that generates L.

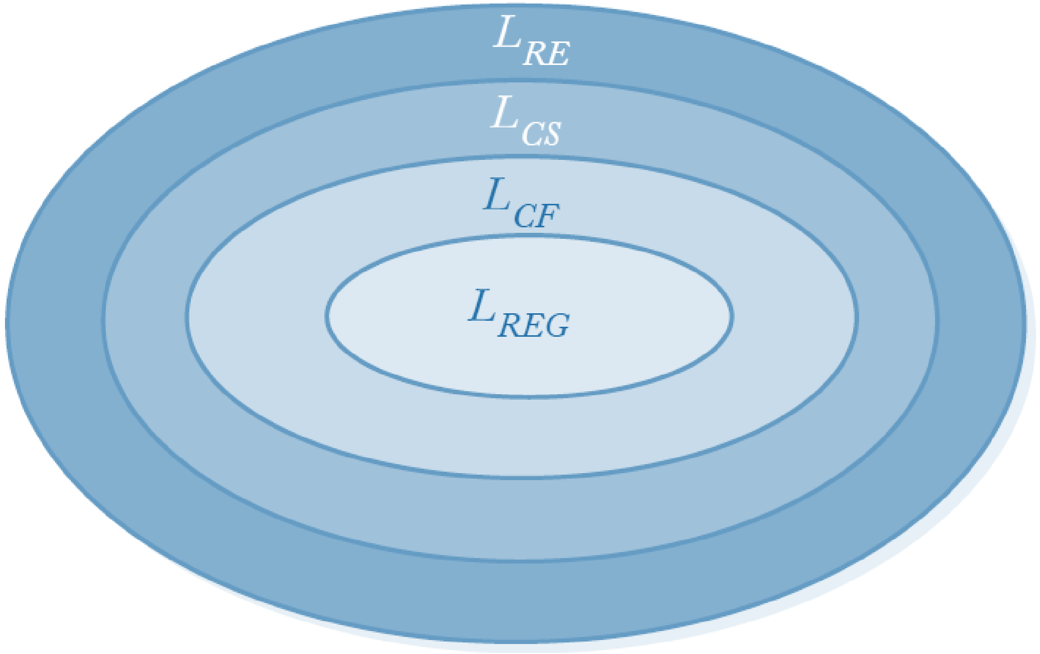
Context-Sensitive Languages and Recursive Languages

Every context-sensitive language L is recursive.

There exists a recursive language that is not context sensitive.

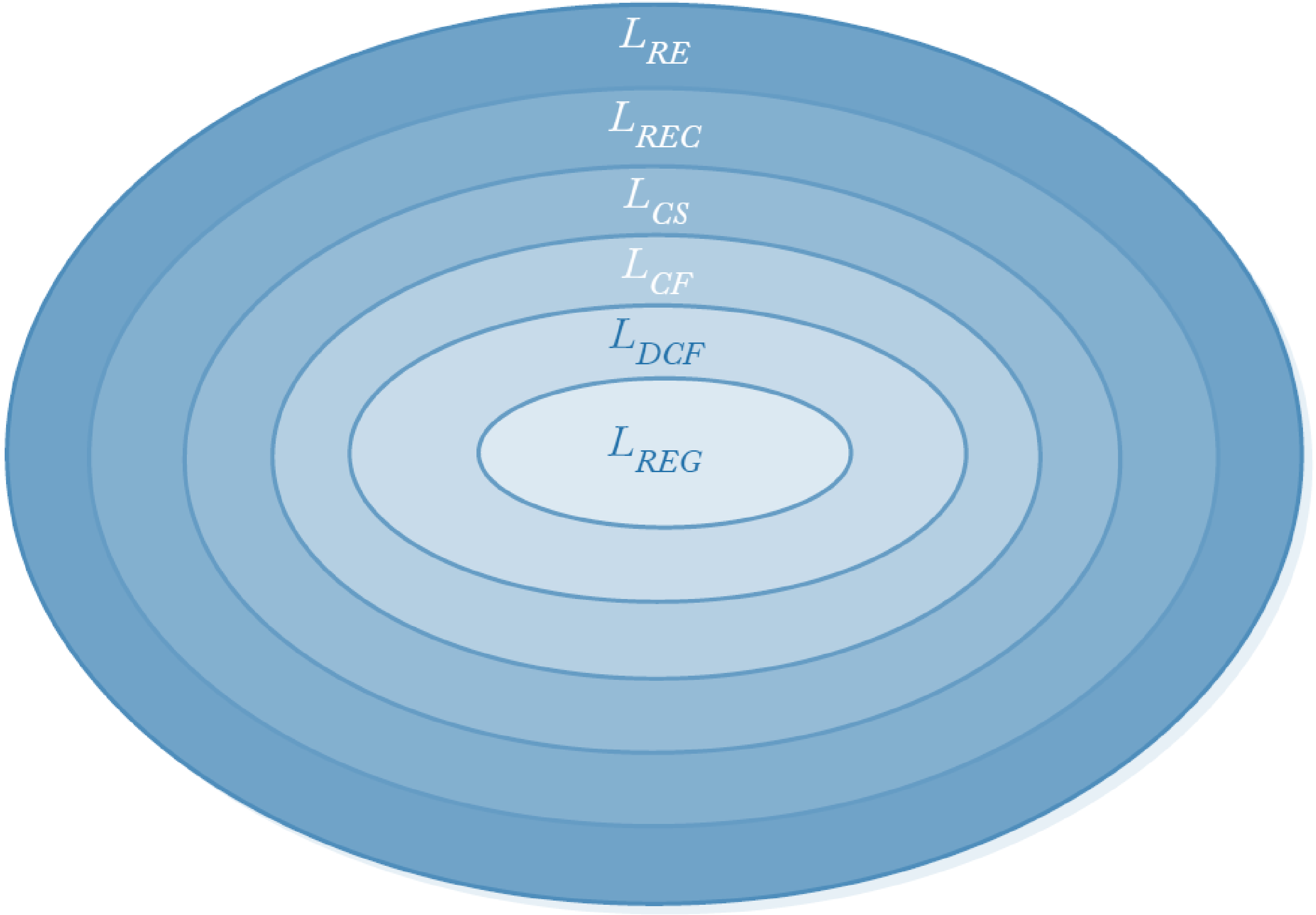
The Chomsky Hierarchy

- Noam Chomsky provided an initial classification into four language types
  - **Type 0**
    - Languages generated by unrestricted grammars, that is, the recursively enumerable languages (L_{RE})
  - **Type 1**
    - Context-sensitive languages (L_{CS})
  - **Type 2**
    - Context-free languages (L_{CF})
  - **Type 3**
    - Regular languages (L_{REG})



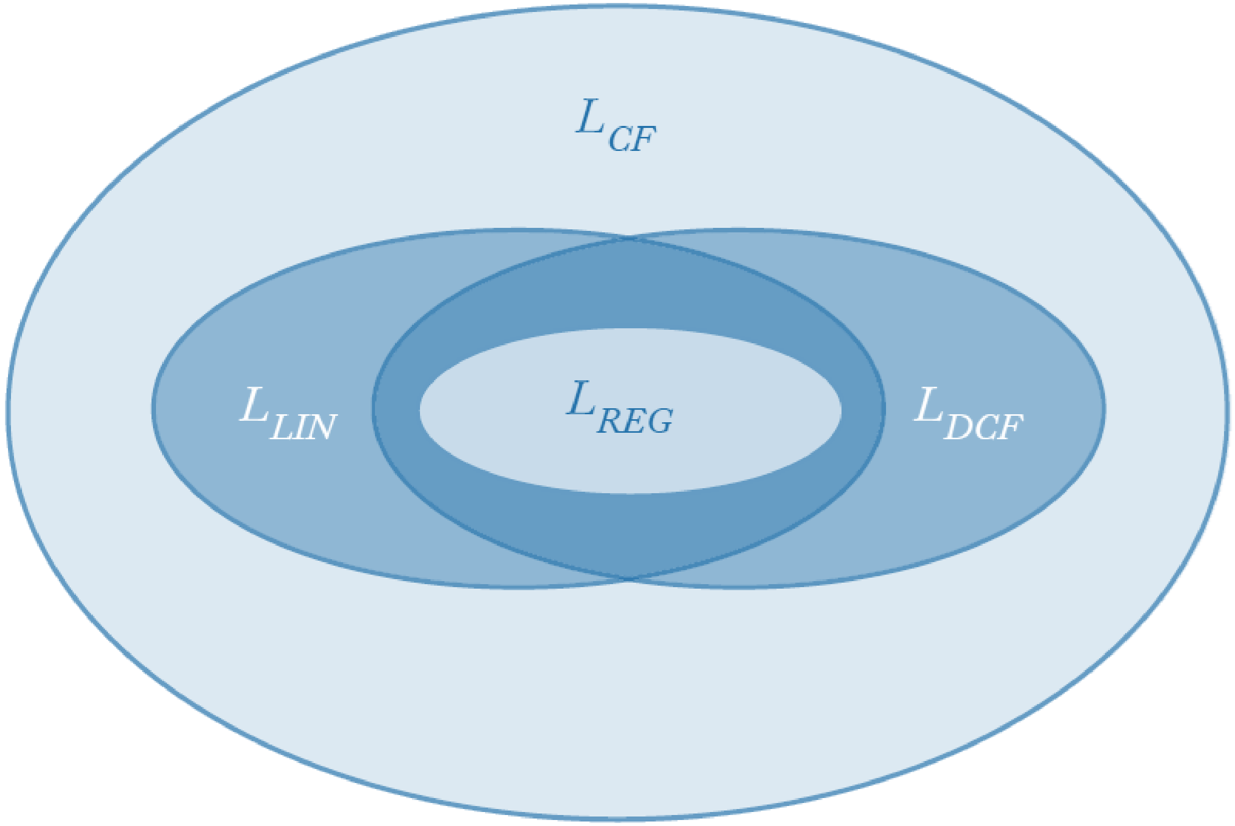
A More Complete Hierarchy

- **Deterministic context-free languages (L_{DCF})**
- **Recursive languages (L_{REC})**



A More Complete Hierarchy

- **Linear languages (L_{LIN})**

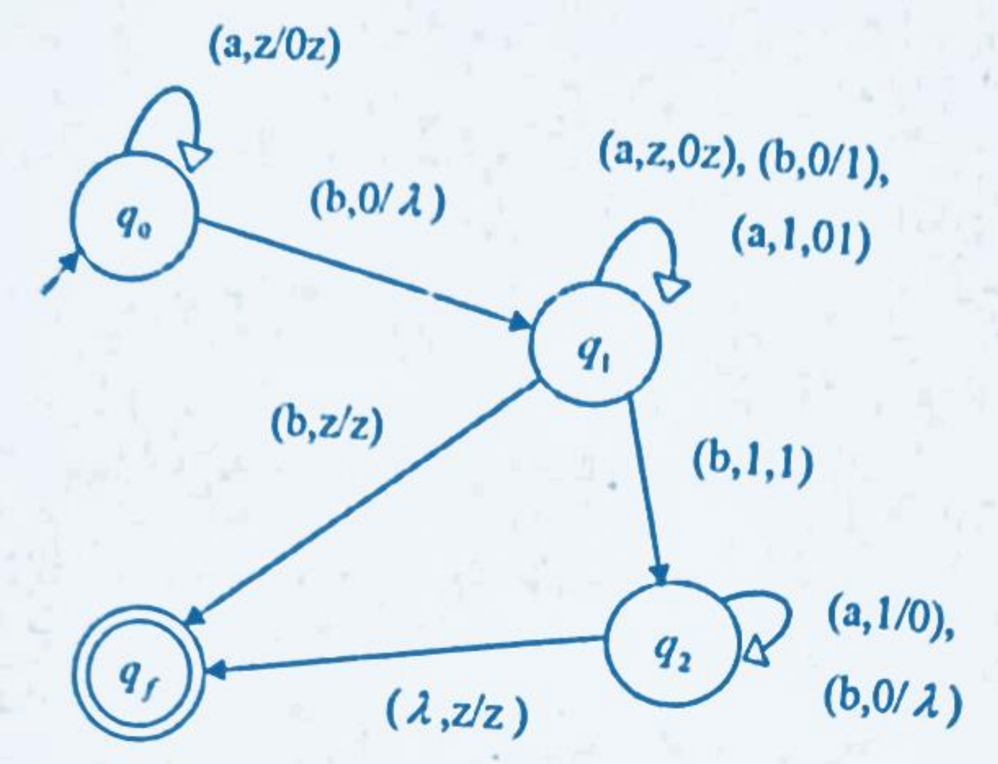


More Examples

More Examples

- **Construct an npda that accept the following language:**

  $L = \{ab(ab)^n b(ab)^n : n \geq 0\}$



More Examples

- با استفاده از لم تزریق نشان دهید که زبان زیر مستقل از متن نیست.

$$\boldsymbol{L = \{a^{n!} : n \geq 0\}}$$

- $\exists m \in N: \quad w = a^{m!} \in L$
- $u = a^{t1}, v = a^{t2}, x = a^{t3}, y = a^{t4}, z = a^{m!-(t1+t2+t3+t4)}$
- که در آن: $|vxy| = t2 + t3 + t4 \leq m$ و $|vy| = t2 + t4 \geq 1$
- $w_i = uv^ixy^iz \qquad , \qquad i = 0 \;\; \Rightarrow \;\; w_0 = a^{m!-(t2+t4)}$
- $t2 + t4 \leq m \;\; \Rightarrow \;\; m! - (t2 + t4) \geq m! - m$
- $\Rightarrow \;\; (m-1)! \leq m\big((m-1)! - 1\big) \leq m! - (t2 + t4)$
- $\Rightarrow \;\; (m-1)! \leq m! - (t2 + t4) \leq m! \quad \Rightarrow$ w_0 عضو زبان نیست ✗

More Examples

- **با استفاده از لم تزریق نشان دهید که زبان زیر مستقل از متن نیست.**

$$L = \{a^n b^j c^k : k > n, k > j\}$$

- رشته $w = a^m b^m c^{m+1} \in L$ را در نظر می‌گیریم.
- در تفکیک رشته به ۵ بخش u، v، x، y و z حالت‌های زیر قابل بررسی هستند:
  - اگر v و y فقط شامل a باشند با انتخاب i=2 تعداد aها بزرگتر یا مساوی تعداد cها شده و شرط k>n نقض می‌شود.
  - اگر v و y فقط شامل b باشند با انتخاب i=2 تعداد bها بزرگتر یا مساوی تعداد cها شده و شرط k>j نقض می‌شود.
  - اگر v و y شامل a و b باشند با انتخاب i=2 تعداد aها و تعداد bها بزرگتر یا مساوی تعداد cها شده و شرط‌های k>n و k>j نقض می‌شود.
  - اگر v و y فقط شامل c باشند با انتخاب i=0 تعداد cها کمتر یا مساوی تعداد aها و bها شده و هر دو شرط k>n و k>j نقض می‌شود.
  - اگر v و y شامل b و c باشند با انتخاب i=0 تعداد cها کمتر یا مساوی تعداد aها شده و شرط k>n نقض می‌شود.
- با توجه به اینکه در همه حالت‌های تفکیک رشته w می‌توانیم با تزریق i رشته‌ای را به دست آوریم که عضو زبان نیست پس طبق لم تزریق زبان مستقل از متن نیست.

More Examples

- Design Turing machines to compute the following function

$$f(x) = \begin{cases} \dfrac{x}{2}, & \text{if } x \text{ is even} \\ \dfrac{x+1}{2}, & \text{if } x \text{ is odd} \end{cases}$$