Lecture 12: CFL Pumping Lemma and Turing Machines

CSC 320: Foundations of Computer Science

Quinton Yong

quintonyong@uvic.ca

University of Victoria

Pumping Lemma for Context-Free Languages

If $\boldsymbol{L}$ is a context-free language, then there is a number $\boldsymbol{p}$ (pumping length of $\boldsymbol{L}$) such that for **every string $\boldsymbol{s} \in \boldsymbol{L}$** of **length at least $\boldsymbol{p}$**, $\boldsymbol{s}$ can be **divided into five parts $\boldsymbol{s} = \boldsymbol{uvxyz}$** satisfying the following:

1. $|\boldsymbol{vy}| > \mathbf{0}$ (i.e. $\boldsymbol{v}$ and $\boldsymbol{y}$ cannot both be empty)
2. $|\boldsymbol{vxy}| \leq \boldsymbol{p}$
3. $\boldsymbol{uv^ixy^iz} \in \boldsymbol{L}$ for each $\boldsymbol{i} \geq \mathbf{0}$

Note: Since there is no restriction on $\boldsymbol{u}$, we need to consider **all cases** of what the substring $\boldsymbol{vxy}$ (with length $\leq \boldsymbol{p}$) could be

CFL Pumping Lemma Example 1

Prove that $\boldsymbol{L} = \{\ \mathbf{0}^{\boldsymbol{n}}\mathbf{1}^{\boldsymbol{n}}\mathbf{2}^{\boldsymbol{n}} \mid \boldsymbol{n} \geq \mathbf{0}\}$ is not context-free.

Proof:

- Assume for a contradiction that $\boldsymbol{L}$ is context-free.
- Let $\boldsymbol{p}$ be the pumping length given by the pumping lemma.
- We choose $\boldsymbol{s} = \mathbf{0}^{\boldsymbol{p}}\mathbf{1}^{\boldsymbol{p}}\mathbf{2}^{\boldsymbol{p}}$.
- Since $\boldsymbol{s} \in \boldsymbol{L}$ and $|\boldsymbol{s}| \geq \boldsymbol{p}$, according to the PL, we can rewrite $\boldsymbol{s} = \boldsymbol{uvxyz}$ satisfying
    1. $|\boldsymbol{vy}| > \mathbf{0}$ (i.e. $\boldsymbol{v}$ and $\boldsymbol{y}$ cannot both be empty)
    2. $|\boldsymbol{vxy}| \leq \boldsymbol{p}$
    3. $\boldsymbol{uv}^{\boldsymbol{i}}\boldsymbol{xy}^{\boldsymbol{i}}\boldsymbol{z} \in \boldsymbol{L}$ for each $\boldsymbol{i} \geq \mathbf{0}$

CFL Pumping Lemma Example 1

$L = \{\ 0^n 1^n 2^n \mid n \geq 0\ \}$

- Consider how $s = 0^p 1^p 2^p$ can be divided into five strings u, v, x, y, z

$$\underbrace{\overbrace{0 \ldots \ldots \ldots 0}^{p}}_{vxy}\ \underbrace{\overbrace{1 \ldots \ldots \ldots 1}^{p}\ \overbrace{2 \ldots \ldots \ldots 2}^{p}}_{z}$$

somewhere in here ($\leq p$)

- By property 2, $|vxy| \leq p$, we have the following cases:

1. $vxy = 0 \ldots 0$

CFL Pumping Lemma Example 1

$L = \{\ 0^n 1^n 2^n \mid n \geq 0\ \}$

- Consider how $s = 0^p 1^p 2^p$ can be divided into five strings u, v, x, y, z

$$\underbrace{\overbrace{0 \ldots\ldots\ldots 0}^{p}}_{u}\ \underbrace{\overbrace{1 \ldots\ldots\ldots 1}^{p}}_{vxy}\ \underbrace{\overbrace{2 \ldots\ldots\ldots 2}^{p}}_{z}$$

somewhere in here ($\leq p$)

- By property 2, $|vxy| \leq p$, we have the following cases:

1. $vxy = 0 \ldots 0$
2. $vxy = 1 \ldots 1$

CFL Pumping Lemma Example 1

$L = \{\ 0^n 1^n 2^n \mid n \geq 0\ \}$

- Consider how $s = 0^p 1^p 2^p$ can be divided into five strings u, v, x, y, z

$$\underbrace{\overbrace{0 \ldots \ldots \ldots 0}^{p}\ \overbrace{1 \ldots \ldots \ldots 1}^{p}}_{u}\ \underbrace{\overbrace{2 \ldots \ldots \ldots 2}^{p}}_{vxy}$$

somewhere in here ($\leq p$)

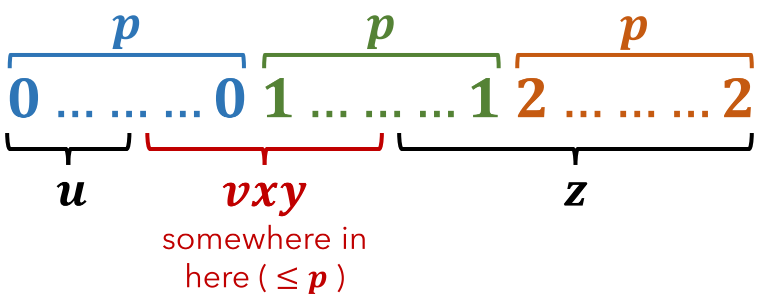
- By property 2, $|vxy| \leq p$, we have the following cases:

1. $vxy = 0 \ldots 0$
2. $vxy = 1 \ldots 1$
3. $vxy = 2 \ldots 2$

CFL Pumping Lemma Example 1

$L = \{\ 0^n 1^n 2^n \mid n \geq 0\ \}$

- Consider how $s = 0^p 1^p 2^p$ can be divided into five strings u, v, x, y, z



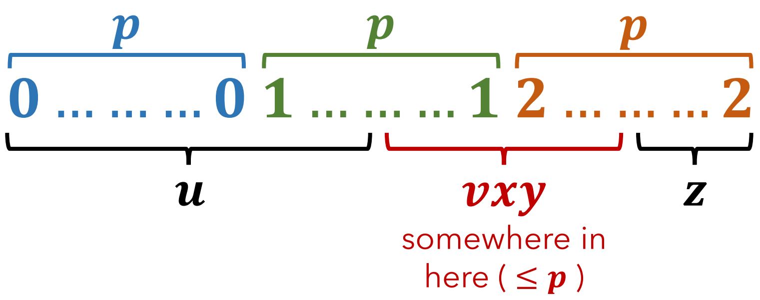
- By property 2, $|vxy| \leq p$, we have the following cases:

1. $vxy = 0 \ldots 0$
2. $vxy = 1 \ldots 1$
3. $vxy = 2 \ldots 2$
4. $vxy = 0 \ldots 0\ 1 \ldots 1$

CFL Pumping Lemma Example 1

$L = \{\ 0^n 1^n 2^n \mid n \geq 0\ \}$

- Consider how $s = 0^p 1^p 2^p$ can be divided into five strings u, v, x, y, z



- By property 2, $|vxy| \leq p$, we have the following cases:

1. $vxy = 0 \ldots 0$
2. $vxy = 1 \ldots 1$
3. $vxy = 2 \ldots 2$
4. $vxy = 0 \ldots 0\ 1 \ldots 1$
5. $vxy = 1 \ldots 1\ 2 \ldots 2$

CFL Pumping Lemma Example 1

$\boxed{L = \{\ 0^n 1^n 2^n \mid n \geq 0\,\}}$

Consider **case 1** where $vxy = 0 \ldots 0$

$$\underbrace{0 \ldots\ldots\ldots 0}_{vxy}\ \underbrace{1 \ldots\ldots\ldots 1\ 2 \ldots\ldots\ldots 2}_{z}$$

somewhere in here ($\leq p$)

- By property 1, $|vy| > 0$ (i.e. v and y cannot both be empty).
- So, v or y (or both) is a non-empty substring of 0's.

- The string $uv^2xy^2z \notin L$ since it increases the number of 0's without increasing the number of 1's or 2's.
- This violates property 3 of the pumping lemma.

CFL Pumping Lemma Example 1

$\boxed{L = \{\ 0^n 1^n 2^n \mid n \geq 0\ \}}$

Consider **case 2** where $vxy = 1 \ldots 1$

$$\underbrace{0 \ldots \ldots \ldots 0}_{u}\ \underbrace{1 \ldots \ldots \ldots 1}_{vxy}\ \underbrace{2 \ldots \ldots \ldots 2}_{z}$$

somewhere in here ($\leq p$)

- By property 1, $|vy| > 0$ (i.e. v and y cannot both be empty).
- So, v or y (or both) is a non-empty substring of 1's.

- The string $uv^2xy^2z \notin L$ since it increases the number of 1's without increasing the number of 0's or 2's.
- This violates property 3 of the pumping lemma.

CFL Pumping Lemma Example 1

$\boxed{L = \{\ 0^n 1^n 2^n \mid n \geq 0\ \}}$

Consider **case 3** where $vxy = 2 \dots 2$

$$\underbrace{0 \dots \dots \dots 0\ 1 \dots \dots \dots 1}_{u}\ \underbrace{2 \dots \dots \dots 2}_{vxy}$$

somewhere in here ($\leq p$)

- By property 1, $|vy| > 0$ (i.e. v and y cannot both be empty).
- So, v or y (or both) is a non-empty substring of 2's.

- The string $uv^2xy^2z \notin L$ since it increases the number of 2's without increasing the number of 0's or 1's.
- This violates property 3 of the pumping lemma.

CFL Pumping Lemma Example 1

$\boxed{L = \{\ 0^n 1^n 2^n \mid n \geq 0\ \}}$

Consider **case 4** where $vxy = 0 \ldots 0\ 1 \ldots 1$

$$\underbrace{0 \ldots \ldots}_{u}\ \underbrace{\ldots 0\ 1 \ldots \ldots}_{vxy}\ \underbrace{\ldots 1\ 2 \ldots \ldots \ldots 2}_{z}$$

somewhere in here ($\leq p$)

- By property 1, $|vy| > 0$ (i.e. v and y cannot both be empty), so we could have:
  - v is non-empty 0's, y is non-empty 1's, or both v is non-empty 0's and y is non-empty 1's
  - Or, either v or y is non-empty substring of $0 \ldots 01 \ldots 1$
- In all cases, $uv^2xy^2z \notin L$ since we increase 0's and/or increase 1's without increasing 2's, or we get 0's and 1's out of order.
- This violates property 3 of the pumping lemma.

CFL Pumping Lemma Example 1

$\boxed{L = \{\ 0^n 1^n 2^n \mid n \geq 0\ \}}$

Consider **case 5** where $vxy = 1 \ldots 1\ 2 \ldots 2$

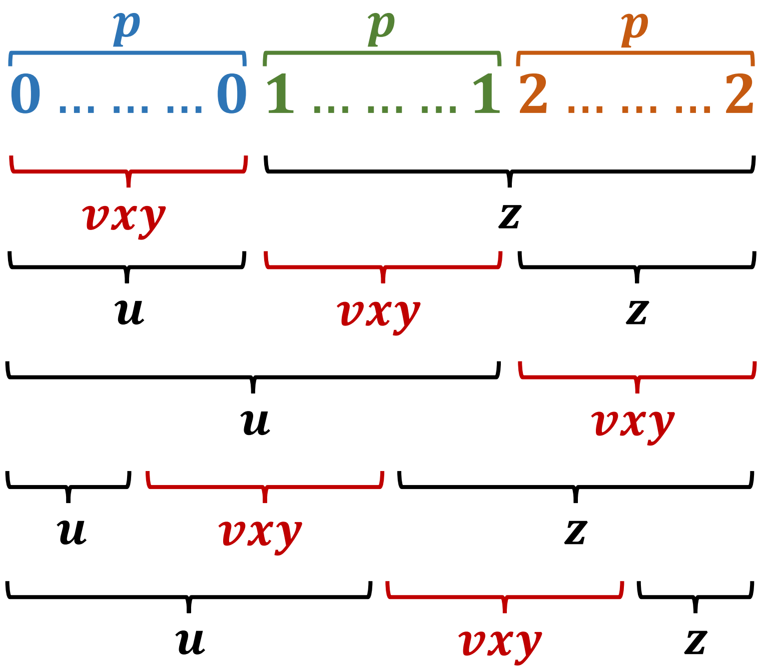
$$\underbrace{0 \ldots \ldots \ldots 0\ 1 \ldots}_{u}\ \underbrace{\ldots \ldots 1\ 2 \ldots}_{vxy}\ \underbrace{\ldots \ldots 2}_{z}$$

somewhere in here ($\leq p$)

- By property 1, $|vy| > 0$ (i.e. v and y cannot both be empty), so we could have:
  - v is non-empty 1's, y is non-empty 2's, or both v is non-empty 1's and y is non-empty 2's
  - Or, either v or y is non-empty substring of $1 \ldots 12 \ldots 2$
- In all cases, $uv^2xy^2z \notin L$ since we increase 1's and/or increase 2's without increasing 0's, or we get 1's and 2's out of order.
- This violates property 3 of the pumping lemma.

CFL Pumping Lemma Example 1

$\boxed{L = \{\ 0^n 1^n 2^n \mid n \geq 0\ \}}$



- We have shown that there is **no way** to rewrite $s = uvxyz$ which satisfies all three conditions of the pumping lemma.
- Therefore, L is not context-free.

CFL Pumping Lemma Example 2

Prove that $\boldsymbol{L} = \{\, \boldsymbol{ww} \mid \boldsymbol{w} \in \{\mathbf{0}, \mathbf{1}\}^* \,\}$ is not context-free.

Proof:

- Assume for a contradiction that $\boldsymbol{L}$ is context-free.
- Let $\boldsymbol{p}$ be the pumping length given by the pumping lemma.
- We choose $\boldsymbol{s} = \mathbf{0}^{\boldsymbol{p}}\mathbf{1}^{\boldsymbol{p}}\mathbf{0}^{\boldsymbol{p}}\mathbf{1}^{\boldsymbol{p}}$.
- Since $\boldsymbol{s} \in \boldsymbol{L}$ and $|\boldsymbol{s}| \geq \boldsymbol{p}$, according to the PL, we can rewrite $\boldsymbol{s} = \boldsymbol{uvxyz}$ satisfying
  1. $|\boldsymbol{vy}| > \mathbf{0}$ (i.e. $\boldsymbol{v}$ and $\boldsymbol{y}$ cannot both be empty)
  2. $|\boldsymbol{vxy}| \leq \boldsymbol{p}$
  3. $\boldsymbol{uv^ixy^iz} \in \boldsymbol{L}$ for each $\boldsymbol{i} \geq \mathbf{0}$

CFL Pumping Lemma Example 2

$L = \{\, ww \mid w \in \{0,1\}^* \,\}$

- Consider how $s = 0^p 1^p 0^p 1^p$ can be divided into five strings u, v, x, y, z

$$\underbrace{\overbrace{0 \ldots \ldots \ldots 0}^{p}}_{vxy}\ \underbrace{\overbrace{1 \ldots \ldots \ldots 1}^{p}\ \overbrace{0 \ldots \ldots \ldots 0}^{p}\ \overbrace{1 \ldots \ldots \ldots 1}^{p}}_{z}$$

somewhere in here ($\leq p$)

$$\underbrace{0 \ldots \ldots \ldots 0\ 1 \ldots \ldots \ldots 1}_{u}\ \underbrace{0 \ldots \ldots \ldots 0}_{vxy}\ \underbrace{1 \ldots \ldots \ldots 1}_{z}$$

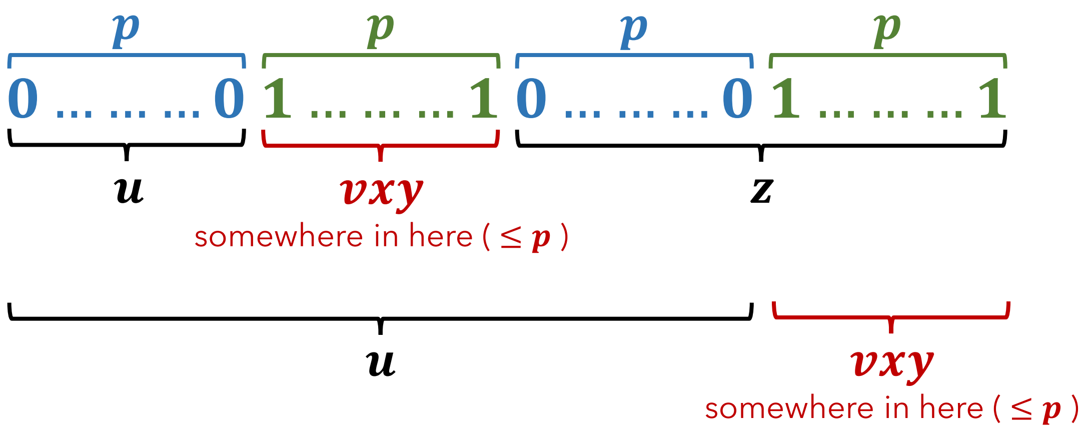
somewhere in here ($\leq p$)

- By property 2, $|vxy| \leq p$, we have the following cases:
    1. $vxy = 0 \ldots 0$

CFL Pumping Lemma Example 2

$L = \{\, ww \mid w \in \{0,1\}^* \,\}$

- Consider how $s = 0^p1^p0^p1^p$ can be divided into five strings u, v, x, y, z

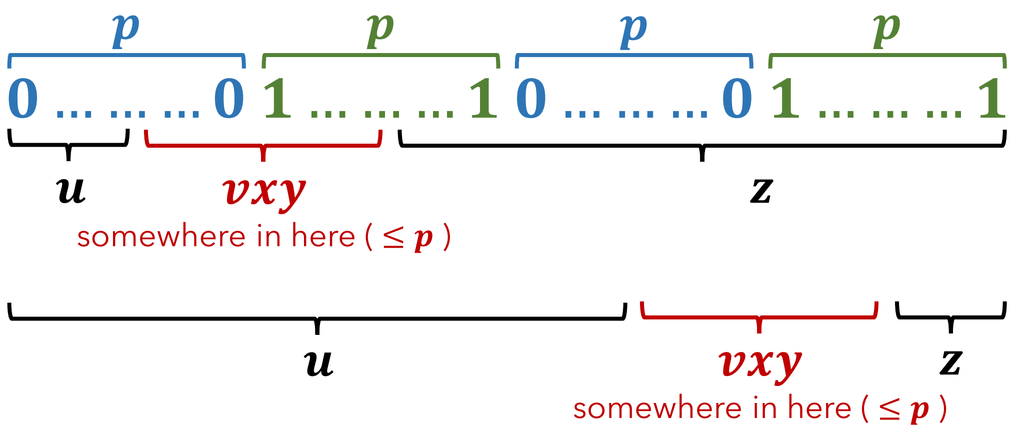


- By property 2, $|vxy| \le p$, we have the following cases:
  1. $vxy = 0 \ldots 0$
  2. $vxy = 1 \ldots 1$

CFL Pumping Lemma Example 2

$L = \{ ww \mid w \in \{0,1\}^* \}$

- Consider how $s = 0^p 1^p 0^p 1^p$ can be divided into five strings u, v, x, y, z



- By property 2, $|vxy| \le p$, we have the following cases:
  1. $vxy = 0 \ldots 0$
  2. $vxy = 1 \ldots 1$
  3. $vxy = 0 \ldots 0\, 1 \ldots 1$

CFL Pumping Lemma Example 2

$\boxed{L = \{ ww \mid w \in \{0,1\}^* \}}$

- Consider how $s = 0^p 1^p 0^p 1^p$ can be divided into five strings u, v, x, y, z

$$\overbrace{0 \ldots \ldots \ldots 0}^{p}\ \overbrace{1 \ldots \ldots \ldots 1}^{p}\ \overbrace{0 \ldots \ldots \ldots 0}^{p}\ \overbrace{1 \ldots \ldots \ldots 1}^{p}$$

u $\qquad$ vxy $\qquad$ z

somewhere in here ($\leq p$)

- By property 2, $|vxy| \leq p$, we have the following cases:
  1. $vxy = 0 \ldots 0$
  2. $vxy = 1 \ldots 1$
  3. $vxy = 0 \ldots 0\ 1 \ldots 1$
  4. $vxy = 1 \ldots 1\ 0 \ldots 0$

CFL Pumping Lemma Example 2

$\boxed{L = \{\, ww \mid w \in \{0,1\}^* \,\}}$

Consider **case 1** where $vxy = 0 \ldots 0$

$$\underbrace{\overbrace{0 \ldots \ldots \ldots 0}^{p}}_{\substack{vxy \\ \text{somewhere in here } (\le p)}}\ \underbrace{\overbrace{1 \ldots \ldots \ldots 1}^{p}\ \overbrace{0 \ldots \ldots \ldots 0}^{p}\ \overbrace{1 \ldots \ldots \ldots 1}^{p}}_{z}$$

$$\underbrace{0 \ldots \ldots \ldots 0\ 1 \ldots \ldots \ldots 1}_{u}\ \underbrace{0 \ldots \ldots \ldots 0}_{\substack{vxy \\ \text{somewhere in here } (\le p)}}\ \underbrace{1 \ldots \ldots \ldots 1}_{z}$$

- By property 1, $|vy| > 0$ (i.e. v and y cannot both be empty). So, v or y (or both) is a non-empty substring of 0's.

- The string $uv^2xy^2z \notin L$ since it increases the number of 0's (somewhere) without increasing the number of 1's.
- This violates property 3 of the pumping lemma.

CFL Pumping Lemma Example 2

$$L = \{\, ww \mid w \in \{0,1\}^* \,\}$$

Consider **case 2** where $vxy = 1 \ldots 1$

$$\underbrace{\overbrace{0 \ldots \ldots \ldots 0}^{p}}_{u}\ \underbrace{\overbrace{1 \ldots \ldots \ldots 1}^{p}}_{\substack{vxy \\ \text{somewhere in here } (\le p)}}\ \underbrace{\overbrace{0 \ldots \ldots \ldots 0}^{p}\ \overbrace{1 \ldots \ldots \ldots 1}^{p}}_{z}$$

$$\underbrace{}_{u}\ \underbrace{}_{\substack{vxy \\ \text{somewhere in here } (\le p)}}$$

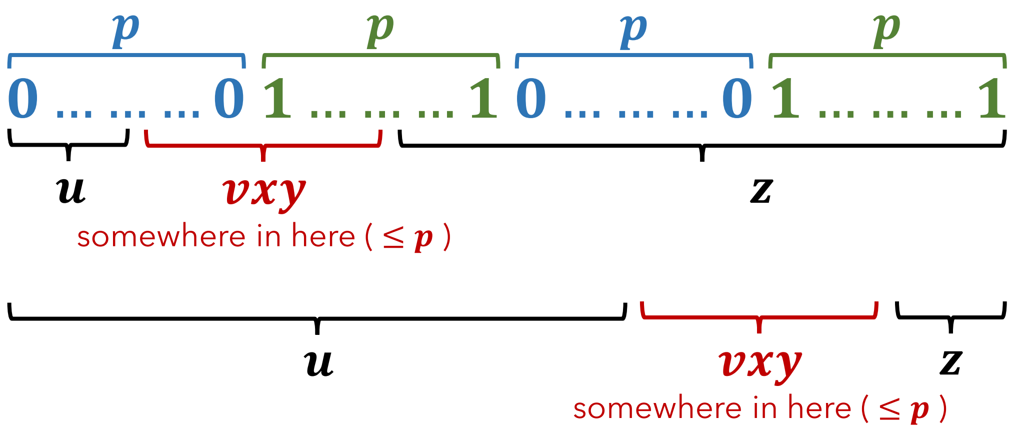
- By property 1, $|vy| > 0$ (i.e. v and y cannot both be empty). So, v or y (or both) is a non-empty substring of 1's.

- The string $uv^2xy^2z \notin L$ since it increases the number of 1's (somewhere) without increasing the number of 0's.
- This violates property 3 of the pumping lemma.

CFL Pumping Lemma Example 2

$L = \{\, ww \mid w \in \{0,1\}^* \,\}$

Consider **case 3** where $vxy = 0 \ldots 0\ 1 \ldots 1$

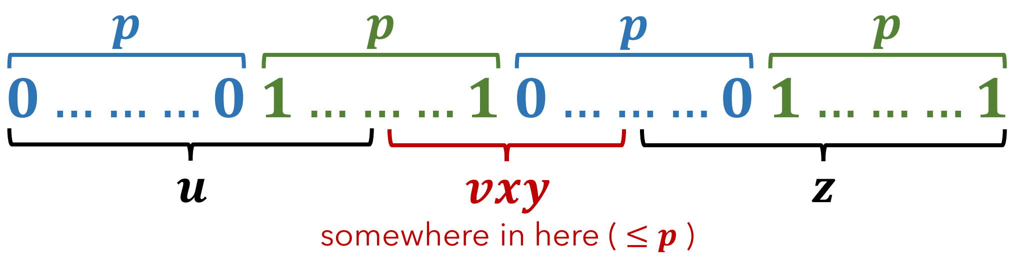


- By property 1, $|vy| > 0$ so we could have:
  - v is non-empty 0's, y is non-empty 1's, or both v is non-empty 0's and y is non-empty 1's, or either v or y is non-empty substring of $0 \ldots 01 \ldots 1$
- In all cases, $uv^2xy^2z \notin L$ since the resulting string will be either $0^k1^l0^p1^p$ or $0^p1^p0^k1^l$ with $k > p$ or $l > p$, or out of order imbalanced 0's and 1's.
- This violates property 3 of the pumping lemma.

CFL Pumping Lemma Example 2

$\boxed{L = \{ww \mid w \in \{0,1\}^*\}}$

Consider **case 4** where $vxy = 1 \ldots 1\ 0 \ldots 0$



- By property 1, $|vy| > 0$, so we could have:
  - v is non-empty 1's, y is non-empty 0's, or both v is non-empty 1's and y is non-empty 0's, or either v or y is non-empty substring of $1 \ldots 10 \ldots 0$
- In all cases, $uv^2xy^2z \notin L$ since the resulting string is of form $0^p1^k0^l1^p$ with $k > p$ or $l > p$, or out of order imbalanced 0's and 1's.
- This violates property 3 of the pumping lemma.

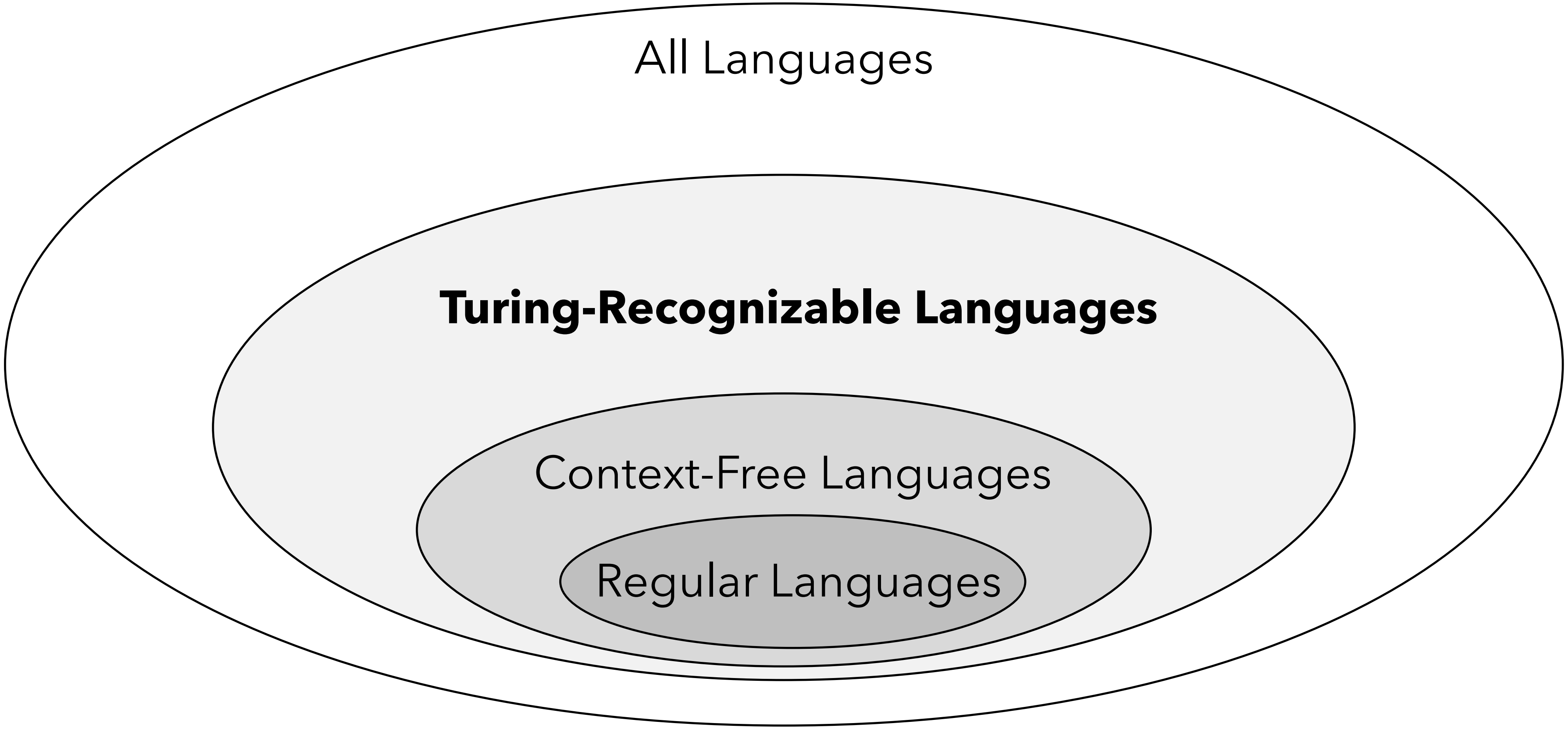
CFL Pumping Lemma Example 2

$\boldsymbol{L = \{\, ww \mid w \in \{0,1\}^* \,\}}$

$$\overbrace{0\,\dots\dots\dots\,0}^{p}\;\overbrace{1\,\dots\dots\dots\,1}^{p}\;\overbrace{0\,\dots\dots\dots\,0}^{p}\;\overbrace{1\,\dots\dots\dots\,1}^{p}$$

- We have shown that there is **no way** to rewrite $\boldsymbol{s = uvxyz}$ which satisfies all three conditions of the pumping lemma .
- Therefore, $\boldsymbol{L}$ is not context-free.

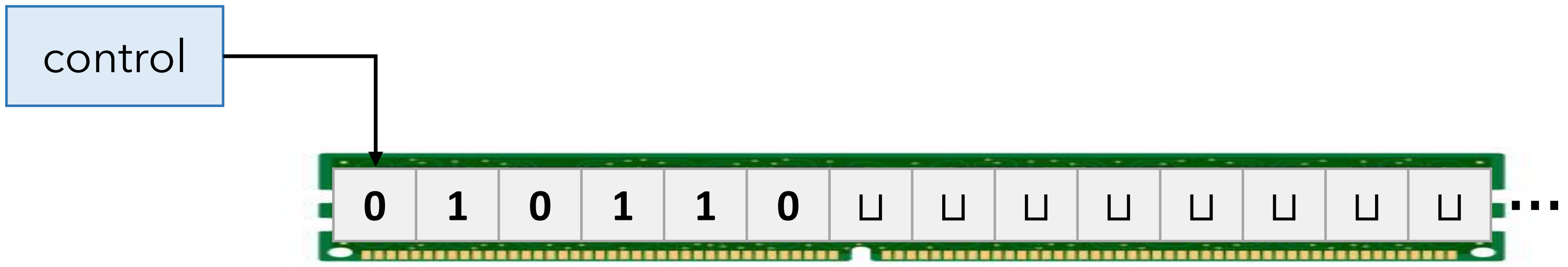
Turing Machines



Turing Machine (TM)

- A much more **powerful computational model** than a FA / PDA

- Similar to a finite automaton, but with **unlimited read / write access** to an **infinite amount of memory**

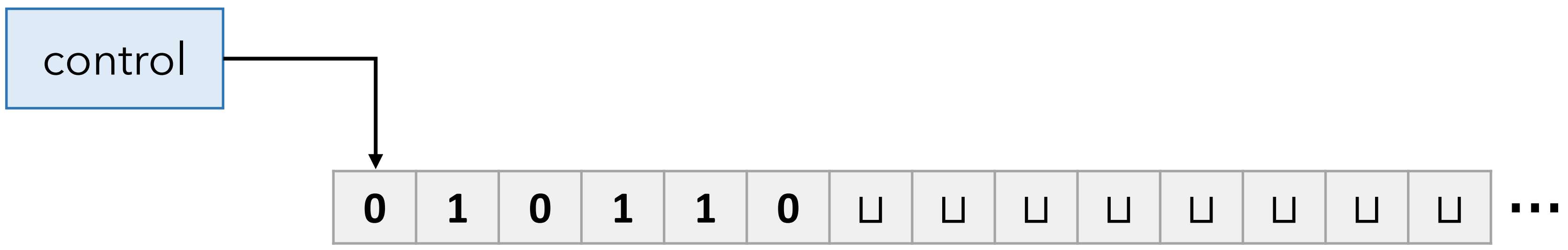


- Model of a **classical computer** (a Turing Machine can do everything a classical computer can do)

Turing Machine (TM)

A Turing Machine consists of:

- **State machine** which defines computation instructions
- **Infinite tape** representing its unlimited memory
- **Tape head** which can
  - read and write symbols
  - move left and right along the tape



Turing Machine Computation (TM)

- Similar to previous computational models, a TM **takes an input string** and **accepts** or does **not accept** the string

Computation:

- Initially, **tape contains** only the **input string**, blank everywhere else
- If TM needs to store information, it can **write information** on tape
- To read information that it has written, TM can **move tape head back** over
- TM continues computing until it decides to produce an output **'accept'** or **'reject'**
  - Outputs obtained by entering designated **accept** or **reject** states
  - If TM doesn't enter an output state, it **continues computation** (infinite loop)

Differences between TMs and FAs

Turing Machines	Finite Automata
1. TM can both read and write from tape	1. FA can only read from input string
2. Read-write head can move both left and right	2. Read head can only move right
3. Tape is infinite	3. No tape
4. Special states for accept and reject (can take place immediately without reading entire input string, or can loop infinitely)	4. Special state for accept, accept or not accept only after reading entire input string

Turing Machine (High Level Example)

Design a TM $\boldsymbol{M}$ which accept strings in the language

$$\boldsymbol{L} = \{\, \boldsymbol{w\#w} \mid \boldsymbol{w} \in \{\mathbf{0}, \mathbf{1}\}^* \,\}$$

High Level Instructions:

- If input string on tape does not contain # in the middle, then output **reject**
- Move tape head **between corresponding places** on two sides of #
  - Mark pairs of matching symbols (replace with an **x**)
  - If all symbols on both sides are crossed off, then output **accept**
  - Otherwise, output **reject**

Turing Machine (High Level Example)

$\boxed{L = \{\, w\#w \mid w \in \{0,1\}^* \,\}}$

High Level Instructions:

- If input string on tape does not contain # in the middle, then output **reject**
- Move tape head **between corresponding places** on two sides of #
  - Mark pairs of matching symbols (replace with an **x**)
  - If all symbols on both sides are crossed off, then output **accept**
  - Otherwise, output **reject**

Input string: $\mathbf{011000\#011000}$

0	1	1	0	0	0	#	0	1	1	0	0	0	⊔	⊔	⊔	…

Turing Machine (High Level Example)

$L = \{\, w\#w \mid w \in \{0,1\}^* \,\}$

High Level Instructions:

- If input string on tape does not contain # in the middle, then output **reject**
- Move tape head **between corresponding places** on two sides of #
  - Mark pairs of matching symbols (replace with an **x**)
  - If all symbols on both sides are crossed off, then output **accept**
  - Otherwise, output **reject**

Input string: **011000#011000**

0	1	1	0	0	0	#	0	1	1	0	0	0	⊔	⊔	⊔	…

Turing Machine (High Level Example)

$L = \{ w\#w \mid w \in \{0,1\}^* \}$

High Level Instructions:

- If input string on tape does not contain # in the middle, then output **reject**
- Move tape head **between corresponding places** on two sides of #
  - Mark pairs of matching symbols (replace with an **x**)
  - If all symbols on both sides are crossed off, then output **accept**
  - Otherwise, output **reject**

Input string: **011000#011000**

x	1	1	0	0	0	#	0	1	1	0	0	0	⊔	⊔	⊔	···

Turing Machine (High Level Example)

$$L = \{\, w\#w \mid w \in \{0,1\}^* \,\}$$

High Level Instructions:

- If input string on tape does not contain # in the middle, then output **reject**
- Move tape head **between corresponding places** on two sides of #
  - Mark pairs of matching symbols (replace with an **x**)
  - If all symbols on both sides are crossed off, then output **accept**
  - Otherwise, output **reject**

Input string: $\mathbf{011000\#011000}$

x	1	1	0	0	0	#	x	1	1	0	0	0	⊔	⊔	⊔	⋯

Turing Machine (High Level Example)

$L = \{\, w\#w \mid w \in \{0,1\}^* \,\}$

High Level Instructions:

- If input string on tape does not contain # in the middle, then output **reject**
- Move tape head **between corresponding places** on two sides of #
  - Mark pairs of matching symbols (replace with an **x**)
  - If all symbols on both sides are crossed off, then output **accept**
  - Otherwise, output **reject**

Input string: **011000#011000**

accept

x	x	x	x	x	x	#	x	x	x	x	x	x	⊔	⊔	⊔

⋯