Tandem Pumping

- Analogous to regular languages there is a pumping lemma for context free languages. The idea is that you can pump a context free language at **two** places (but not more).
- Theorem: Given a context free language L, there is a number p (**tandem-pumping number**) such that any string in L of length $\geq p$ is tandem-pumpable within a substring of length p. In particular, for all $w \in L$ with $|w| \geq p$ we we can write:
  - $w = uvxyz$
  - $|vy| \geq 1$ (pumpable areas are non-empty)
  - $|vxy| \leq p$ (pumping inside length-p portion)
  - $uv^ixy^iz \in L$ for all $i \geq 0$ (tandem-pump v and y)
- If there is no such p the language is not context-free.

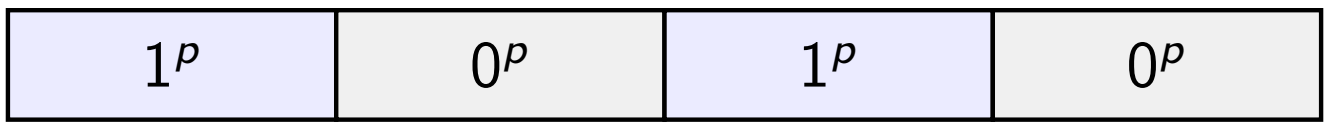
The Two Pumping Lemmas Side by Side

	Regular languages	**Context-free languages**
what we look at	the run of the FA on w	the parse tree of w
finitely many ...	states	variables
...so one repeats	a state	a variable
the repetition is	a *loop*: $q \xrightarrow{y} q$	a *self-embedding*: $A \Rightarrow^* vAy$
decomposition	$w = xyz$, $\lvert y\rvert \geq 1$	$w = uvxyz$, $\lvert vy\rvert \geq 1$
what gets pumped	y alone: $xy^iz \in L$	v *and* y: $uv^ixy^iz \in L$
window	$\lvert xy\rvert \leq p$	$\lvert vxy\rvert \leq p$

- The argument is the same in both cases: a long string forces a repetition, and a repetition can be repeated. Only the *kind* of repetition differs. Both are one-way tests.

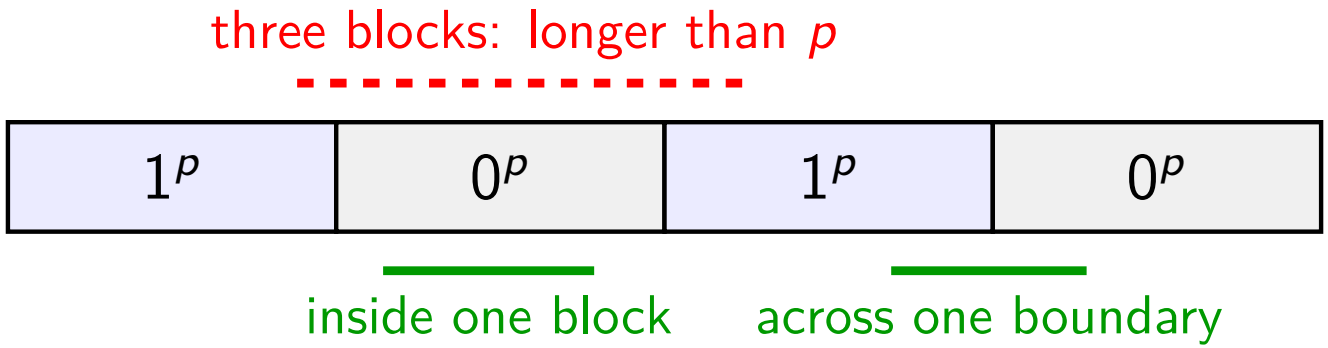
Proving Non-Context Freeness: Example

- $L = \{\, 1^n 0^n 1^n 0^n \mid n \geq 0 \,\}$; take $w = 1^p 0^p 1^p 0^p \in L$.



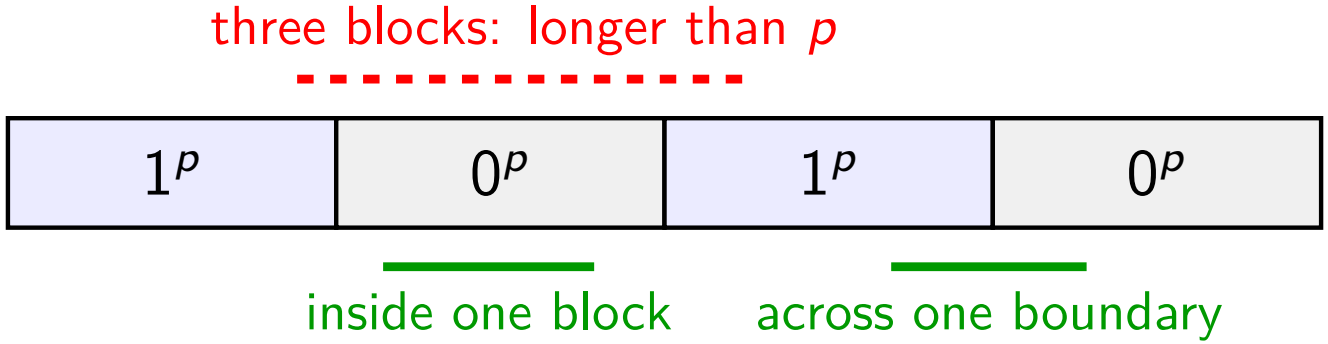
Proving Non-Context Freeness: Example

- $L = \{ 1^n 0^n 1^n 0^n \mid n \geq 0 \}$; take $w = 1^p 0^p 1^p 0^p \in L$.
- Each block has length p and $|vxy| \leq p$: the window vxy touches *at most two neighbouring blocks*.



Proving Non-Context Freeness: Example

- $L = \{ 1^n 0^n 1^n 0^n \mid n \geq 0 \}$; take $w = 1^p 0^p 1^p 0^p \in L$.
- Each block has length p and $|vxy| \leq p$: the window vxy touches *at most two neighbouring blocks*.



- Pumping v and y changes only the block(s) they touch, so *at least two blocks stay unchanged* – but L needs all four equal (and if v or y itself contains both 0s and 1s, pumping it up mixes the blocks).
- No split $w = uvxyz$ works $\Rightarrow$ L is **not context-free**.

Proving Non-Context Freeness: You try!

- $L = \{\, x{=}y{+}z \mid x,\ y,$ and z are binary bit-strings satisfying the equation $\}$
- Show that L is not context-free.
- As usual now, the hard part is to come up with a word in L which cannot be pumped.

Context Sensitive Grammars

- An even more general form of grammars exists.
  In general, a non-context free grammar is one in which whole mixed variable/terminal substrings are replaced at a time.
  For example with $\Sigma = \{a, b, c\}$ consider:

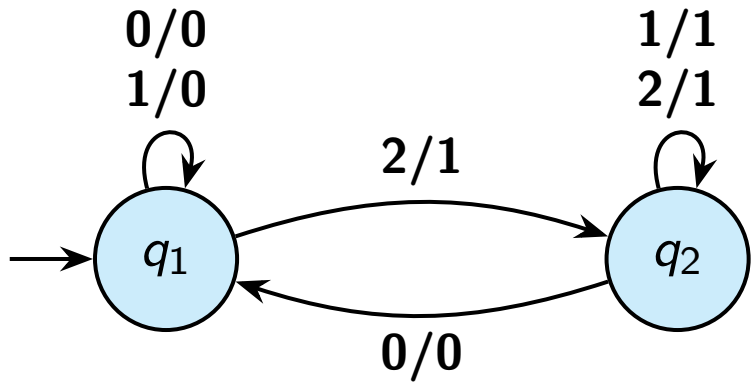
$$\begin{array}{lll} S \rightarrow \varepsilon \mid ASBC & & aB \rightarrow ab \\ A \rightarrow a & & bB \rightarrow bb \\ CB \rightarrow BC & & bC \rightarrow bc \\ & & cC \rightarrow cc \end{array}$$

  Question: What language is generated by this non-context-free grammar?

- When length of LHS always $\leq$ length of RHS (plus some other minor restrictions), these general grammars are called **context sensitive**.

Transducers

- Definition: A finite state **transducer** (FST) is a type of finite automaton whose **output is a string** and not just accept or reject.
- Each transition is labeled with *two* symbols: the input symbol (as for automata) and the output symbol.
  - We allow ε as output (no symbol is added to the string).
- The figure on the right shows an FST with input alphabet $\{0, 1, 2\}$ and output alphabet $\{0, 1\}$.



- Exercise: Can you design a transducer that produces the inverted bit-string of the input string (e.g. 01001 $\rightarrow$ 10110)?