COMS W3261 — Lecture 8, Part 1:

CFPL Examples. (Part 2 — Turing Machines.)

Teaser: Which actor played A. Turing in the 2014 film The Imitation Game?

Who played the cryptologist/codebreaker Joan Clarke?

Answers: Benedict Cumberbatch, Kiera Knightly.

Announcements: HW 4 due on 7/26, at 11:59 PM EST
HW 5 due on 8/2/21, at 11:59 PM EST
Slight change in video format: notes after.
// notes like this added afterwards. (like this)

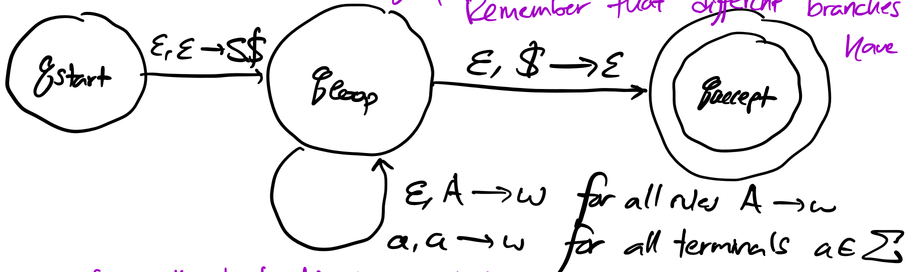
Readings: (2.3 — CFPL)
3.1 — Turing Machines and Recognizability, Decidability.

Today:
1. Review
2. CFPL Examples
3. Turing Machine
4. Recognizability & Decidability.

1. Review. CFG $\Leftrightarrow$ PDA.

$\hookrightarrow$ Converted from a CFG $\longrightarrow$ PDA.

← q_{loop} nondeterministically fires all derivations. Remember that different branches may have different stacks.

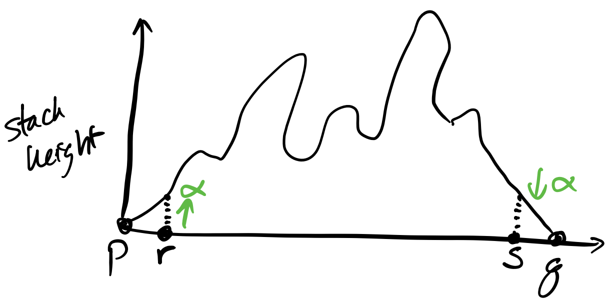


(Recall: We sometimes "pushed strings" — but this was shorthand for using several states to push one symbol at a time.)

↳ Converted from a PDA → CFG.

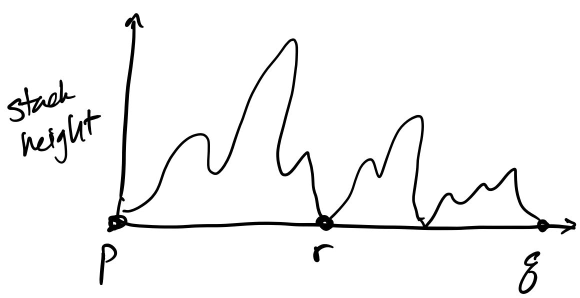
A_{pq} : variable to generate all strings that correspond to a computational path from p to q in our PDA w/empty stacks before and after.

Two cases: (We showed that we could assume all PDAs have one accept state and end with an empty stack without loss of generality.)



$$A_{pq} \to a A_{rs} b$$

a, b are the input symbols in Σ_ε that made as go $p \to r$ and $s \to q$.



$$A_{pq} \to A_{pr} A_{rq}.$$

<u>Punchline:</u> Any language is <u>Context-Free</u> if and only if some PDA recognizes it.

⊂ = described by CFG, by definition.

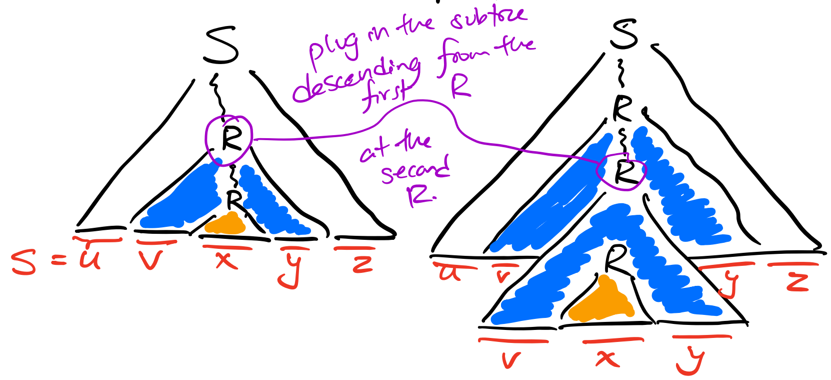
Theorem. (PL for CF languages.) If L is a CF language, there exists a "pumping length" p such that, for all $s \in L$ with $|s| \geq p$, s can be divided into 5 substrings $s = uvxyz$ such that

(1) For all $i \geq 0$, $uv^ixy^iz \in L$. // corresponds to "loop" in parse tree

(2) $|vy| > 0$, // one of our repeated substrings is nontrivial

(3) $|vxy| \leq p$. // we find a loop within p layers from the bottom of our parse tree (see below.)

Proof idea: CFLs have CFGs with a finite number of variables; sufficiently long strings repeat a variable in every derivation, and this "loop" in derivation can be pumped.



Example. Show that $B = \{a^n b^n c^n \mid n \geq 0\}$ is not context-free.

(CFPL tells us that all CF languages have a pumping length such that all sufficiently long strings have some decomposition $s = uvxyz$ that can't be pumped.)

(Thus to show not CF: Show any sufficiently long string s.t. no decomposition can be pumped.)

Proof. (1) Assume B is context-free.

(2) Thus B satisfies the CFPL, so B has a PL p.

For all $s \in B$, $|s| \geq p$, s can be divided into u, v, x, y, z s.t.

$$uv^ixy^iz \in B \text{ for all } i \geq 0,$$
$$|vy| > 0,$$
$$|vxy| \leq p.$$

// want to show: there exists some long string s such that no division works.

(3) Choose $s = a^pb^pc^p$. $s \in B$, $|s| = 3p$. Consider all possible decompositions into substrings (into two cases.)

i.e., all a's, all b's, or all c's.

Case 1. v and y contain at most one type of symbol each. Thus, as $|vy| \geq 0$, at most two and at least one of the three substrings a^p, b^p, c^p increases in length in $uvvxyyz$. Therefore the number of a's, b's and c's is no longer equal $\rightarrow uv^2xy^2z \notin B$.

Case 2. At least one of v and y contains two symbol types in $\{a, b, c\}$. Now, pumping v and y creates a string

$uvvxyyz$ that contains some symbols out of order:

$$\left(s = aaa\underbrace{ab}_{v}bbb\underbrace{cc}_{y}cc. \quad uuvxyyz = aaa\overbrace{ab}^{v}\overbrace{bab}^{v}bbbcccccc. \right)$$

So $uv^2xy^2z \notin B$.

Thus any division of s fails the CFPL conditions, and thus s is not pumpable. B is thus not CF. ∎

Example. Let $D = \{ww \mid w \in \{0,1\}^*\}$. We show D is not context-free.

(1) Assume for contradiction that D is CF.

(2) Therefore D satisfies CFPL and has a pumping length p.

This means that for $s \in D$, $|s| \geq p$, we have $s = uvxyz$

$uv^ixy^iz \in D$ for all $i \geq 0$,

$|vy| > 0$

$|vxy| \leq p$.

(3) Choose $s = 0^p 1 0^p 1$? $s \in D$, $|s| \geq p$.

$$\underbrace{0000 \cdots}_{u} \underbrace{0}_{v}\underbrace{0}_{x}\underbrace{0}_{y}\underbrace{1\ 0000 \cdots 000\ 1}_{z}.$$

$|vy| > 0$ ✓

$|vxy| \leq p$ ✓

$uv^ixy^iz \in D$, $\forall i$ ✓

$$uv^2xy^2z = 0^{p+1} 1 0^{p+1} 1 \in D.$$

No — this string can be pumped.

// Just because one string can be pumped doesn't mean that all strings can be pumped!

Choose $s = 0^p 1^p 0^p 1^p$. $s \in D$, $|s| = 4p \geq p$.

We prove that no division of s satisfies our condition by dividing all decompositions into cases.

<u>Case 1</u>: vxy is a substring of the first $0^p 1^p$ substring of s.

inserted another v inserted another y.

$$000 \cdots \underbrace{0000\ 11\cancel{1} \cdots}_{vxy} 1111\ 0^p 1^p$$

Now, consider pumping v and y to get uv^2xy^2z. Now, as $|vxy| \leq p$, we've added $\leq p$ characters to this first substring. This means that the <u>midpoint</u> of our string is now a 1. Now the first half of our string starts with 0 and the second half starts with 1, so $uv^2xy^2z \neq ww$ for any $w \in \{0,1\}^*$.

<u>Case 2</u>: vxy is a substring of the second $0^p 1^p$ substring of s. (similar to Case 1.) // now, we compare the end of the first half (0)

<u>Case 3:</u> vxy straddles the midpoint of s. to the end of the second half (1).

$$0000\cdots000111\cdots111\underbrace{000\cdots00}_{vxy}11\cdots\cdot11$$

Because $|vxy| \le p$, it doesn't include any of the first p zeroes or any of the last p ones.

Now, consider pumping down to the string $uxz = uv^0xy^0z$, which is $0^p1^i0^j1^p$ for some i and j. i and j can't both be p, as this would imply $|vy| = 0$. However, this implies that $0^p1^i \neq 0^j1^p$, and thus $uxz \notin D$.

Therefore no decomposition of s into uvxyz satisfies the pumping lemma, so D is not context-free. ■

Next up: Turing Machines!