Rhea Mae Edwards

CS 321: Homework #5

1. Pumping Lemma Technique

> 1. Adversary picks a positive integer p
> 2. You pick $w \in A$ with $|w| \geq p$
> 3. Adversary splits up w into $w = xyz$ with $|xy| \leq p$ and $|y| > 0$
> 4. You pick a nonnegative integer i
>
> You win the game if $xy^iz \notin A$. If you can describe a strategy in which you *always* win, then A is not regular.

Prove languages are not regular…

a) $\{w \in \{a, b\}^* \mid \text{num}(aa, w) = \text{num}(bbb, w)\}$

Note: Be careful in your counting: *bbbb* contains the substring *bbb* two times.

1. Adversary picks a positive integer p.

2. I pick $w = ((aa)a^{p-1})^p\,((bbb)b^{p-1})^p$

Satisfying $w \in A$ with $|w| = 2(2p – 1) = 4p – 1 \geq p$.

3. Adversary splits up $w = xyz$ with $|xy| \leq p$ and $|y| > 0$

4. I pick i = 2

$xy^iz = xyyz$ has $p + |y|$ number of *aa*’s

$p – 1$ number of *bbb*’s

Since $p + |y| \neq p – 1$, xy^iz is not $\in$ A, proving the language is not regular.

b) $\{w \in \{a, b\}^* \mid |w| \text{ is a square number}\}$

Hint: Choose i so you can argue that the length of the resulting string is strictly between adjacent squares, i.e. $n^2 < |xy^iz| < (n + 1)^2$ for some n.

1. Adversary picks a positive integer p.

2. I pick $w = a^{2p}$

Satisfying $w \in A$ with $|w| = 2p \geq p$.

3. Adversary splits up $w = xyz$ with $|xy| \leq p$ and $|y| > 0$

4. I pick i = 1/3

$xy^i z = xy^{1/3} z$

Since (1/3) is not a multiple of 2 with creating a square number for a *w*,
xy^iz is not $\in$ A, proving the language is not regular.

2. Describe a CFL

a) $\{w \in \{0, 1\}^* \mid \overline{w} = \text{rev}(w)\}$

Note: $\overline{w}$ denotes flipping every bit in the string w, for example: $\overline{00101} = 11010$.

S → A | B | C | D | ε

A → A01 | ε

B → B10 | ε

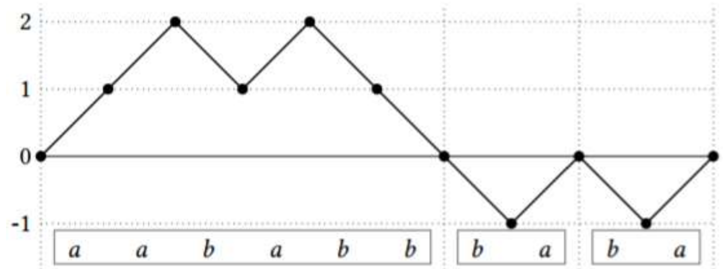
C → 0C1 | ε

D → 1D0 | ε

- Holds true for all the strings that are considered in the CFL.

b) $\{w \in \{a, b\}^* \mid \text{num}(a, w) = \text{num}(b, w)\}$

Hint: Consider the following example where we graph num(a, w) – num(b, w) for al prefixes of w:



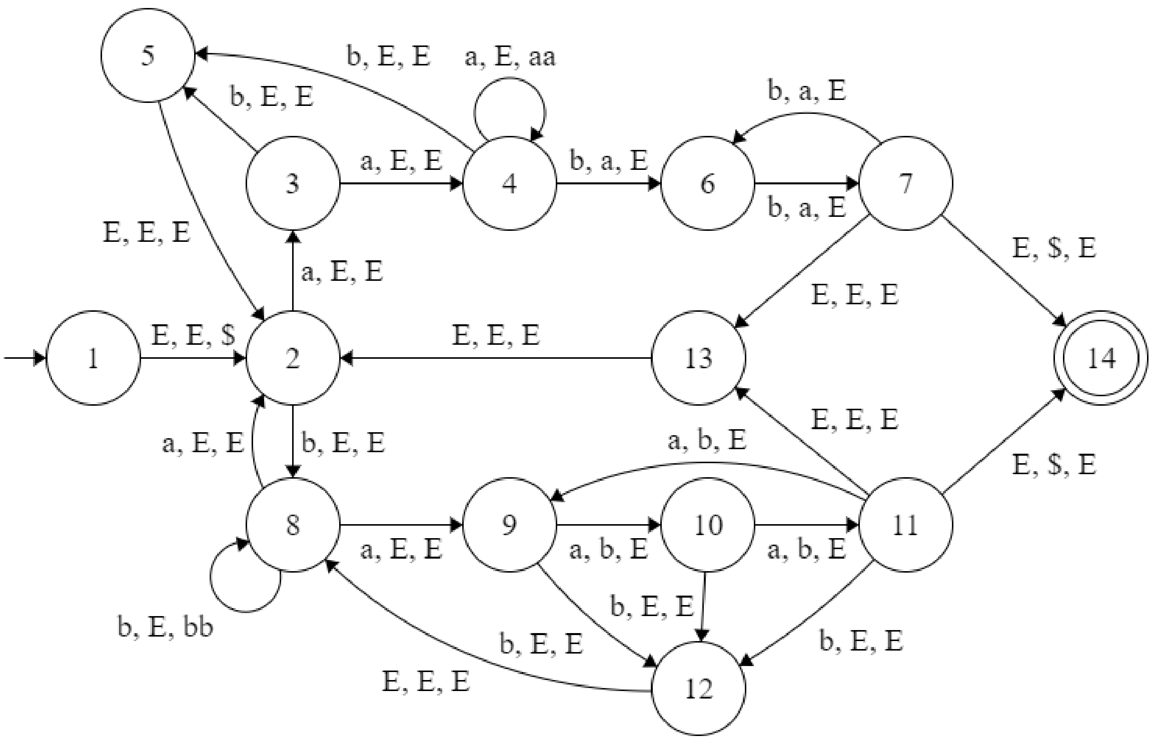
Imagine breaking up the string into pieces every time the graph crosses the x-axis.
Look at the first and last character of each such piece and think recursively.

S → Sab | Sba | aSb | bSa | abS | baS | ε

- Makes sure that for every a, there's a b.

3. Describe a PDA

{$w \in \{a, b\}^*$ | num(aaa, w) = num(bb, w)}



PDA does not consider…

- Non-*aaa* strings that are not substrings a *aaa* of the top-half of the graph
- An additional single *b* character to a *bb* string then accepting of the top-half of the graph
- An additional single *a* character or double *a* string to a *aaa* string then accepting of the bottom-half of the graph

where these cases should have been valid representations within this PDA.