## Today's lecture: Non-Regular Languages and the Pumping Lemma

## Reading:

**Today:** Nonregular Languages
Read: *Kozen* lecture 12 or *Sipser* 1.4.

**September 28:** More examples of the pumping Lemma
Read: *Kozen* lecture 13 or *Sipser* 1.4.

**September 30:** Survey of other topics related to finite automata

**October 3:** Context Free Languages and Grammars
Read: *Kozen* lecture 19 or *Sipser* 2.1.

**October 5:** Balanced Parentheses
Read: *Kozen* lecture 20 (or *Sipser* 2.1).

**October 7:** Chomsky and Greibach normal forms
Read: *Kozen* lecture 21 (or *Sipser* 2.1).

**October 10:** Thanksgiving, no lecture

**October 12:** Non-Context-Free Languages
Read: *Kozen* lecture 22 or *Sipser* 2.3.

**October 14:** Non-deterministic, Pushdown Automata
Read: *Kozen* lecture 23 or *Sipser* 2.2.

**October 17:** From CFLs to PDAs
Read: *Kozen* lecture 24 (or *Sipser* 2.2).

**October 19:** From PDAs to CFLs
Read: *Kozen* lecture 25 (or *Sipser* 2.2).

**October 21:** Deterministic PDAs
Read: *Kozen* lectures E and F.

**October 24:** Parsing
Read: *Kozen* lecture 26 (not in Sipser)

**October 26:** Midterm: In class.

**October 28:** A Parsing Algorithm
Read: *Kozen* lecture 27 (not in Sipser)

**October 31:** LALR Parsing
Read: TBD

**I.** A non-regular language: $B = \{w \in \Sigma^* | \exists n \in \mathbb{Z}^+ . w = a^n b^n\}$

**A.** We'll prove this by contradiction – assume that $B$ is regular.

**B.** Then, there exists a DFA $M$, that accepts language $B$.

**1.** Let $M = (Q, \Sigma, \delta, q_0, F)$, and let $k = |Q|$.

**2.** Let

$$\begin{aligned} w &= a^k b^k \\ p(i) &= \hat{\delta}(q_0, a^i) \end{aligned}$$

Intuitively, $w$ is a string in $B$ that is long enough to ensure that $M$ will run out of states when processing it. The function $p(i)$ gives the state that $M$ reaches after processing the first $i$ $a$'s of $w$:

| $i$: | 0 | 1 | 2 | 3 | ... | $k$ |
|---|---|---|---|---|---|---|
| input: | $a$ | $a$ | $a$ | $a$ | ... | $a$ |
| $M$'s state: | $p(0)$ | $p(1)$ | $p(2)$ | $p(3)$ | ... | $p(k+1)$ |

**3.** **Pigeon hole** $p(i)$.
The range of $p$ is $Q$ which has only $k$ distinct values. There are $k+1$ distinct values for $i$ in $0 \ldots k$. Therefore, we can find distinct $i$ and $j$ in $0 \ldots k$ such that $p(i) = p(j)$. For those who like to see this written as a mathematical formula:

$$\exists i, j \in [0 \ldots k]. \; (i \neq j) \wedge (p(i) = p(j))$$

Thus, we can choose $i$ and $j$ such that $0 \leq i < j \leq k$ and $p(i) = p(j)$.

**4.** Let $d = j - i$.
Because $i < j$, $d > 0$. Because $p(i) = p(j)$, we conclude $\hat{\delta}(p(i), a^d) = p(i)$. More generally, this periodicity holds for any state $p(g)$ with $g \geq i$:

$$\forall g \geq i. \; \forall m \in \mathbb{Z}^+. \; p(g + md) = p(g)$$

**5.** In particular, $p(k+d) = p(k)$. Let $w' = a^{k+d} b^k$.

$$\begin{array}{lll} \hat{\delta}(q_0, w') & & \\ = & \hat{\delta}(q_0, a^{k+d} b^k) & \text{def. } w' \\ = & \hat{\delta}(\hat{\delta}(q_0, a^{k+d}), b^k), & \text{prop. of } \hat{\delta} \\ = & \hat{\delta}(p(k+d), b^k), & \text{def. } p(i) \\ = & \hat{\delta}(p(k), b^k), & \text{shown above} \\ = & \hat{\delta}(\hat{\delta}(q_0, a^k), b^k), & \text{def. } p(i) \\ = & \hat{\delta}(q_0, a^k b^k), & \text{prop. of } \hat{\delta} \\ \in & F & \text{assumption that } M \text{ accepts } B \end{array}$$

But this means that $M$ accepts $w'$ and $w' \notin B$. A contradiction.

**C.** $B$ is not regular, because assuming that it is leads to a contradiction.

**II.** The pumping lemma.

**A.** A generalization of the example given above.
Let $B$ be a language. If $B$ is regular, then it is recognized by some DFA $M = (Q, \Sigma, \delta, q_0, F)$. Let $k = |Q|$. Let $w \in B$ be a string with $|B| > k$. Then for any substring of $w$ with length greater than $k$, there must be some state that is visited two or more times. We can use this to create other strings that must be in $B$.

**B.** The **Pumping Lemma**:
Let $B$ be a regular language. There exists an integer $k$, such that for any strings $x$, $y$, and $z$ with $xyz \in B$ and $|y| \geq k$, there exists strings $u$, $v$, and $w$ with $uvw = y$ such that for every $m \geq 0$, $xuv^m wz \in B$.

Our proof is by contradiction and is a generalization of our proof that $a^n b^n$ is not regular.

**1.** Proof:

**a.** If $B$ is regular, then there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ that accepts a string iff it is in $B$. Let $k = |Q|$. Let $x$, $y$, and $z$ be strings such that $xyz \in B$ and $|y| \geq k$.

**b.** For $0 \leq i \leq |y|$, let $s(i)$ be the first $i$ symbols of $y$. Let

$$p(i) \quad = \quad \hat{\delta}(q_0, x \cdot s(i))$$

**c.** Because $|y| \geq k$, there are at least $k+1$ possible values for $i$, but only $k$ possible values for $p(i)$. Choose $i, j$ with $0 \leq i < j \leq |y|$ such that $p(i) = p(j)$.

**d.** Now, we break $y$ into three pieces, $u$, $v$, and $w$. Let $u = s(i)$. Choose $v$ such that $s(j) = s(i) \cdot v$, and choose $w$ such that $y = uvw$. By these choices, we have

| | | | | |
|---|---|---|---|---|
| 1. | $\hat{\delta}(q_0, xu)$ | $=$ | $p(i)$, | def. $u$ and $p(i)$ |
| 2. | $\hat{\delta}(p(i), v)$ | $=$ | $p(i)$, | def. $v$ and $p(j) = p(i)$ |
| 3. | $\hat{\delta}(p(i), v^m)$ | $=$ | $\hat{\delta}(p(i), v)$ | above, and induction on $m$ |
| 4. | $\hat{\delta}(q_0, xuv^m)$ | $=$ | $p(i)$, | (1) and (3) |
| 5. | $\hat{\delta}(q_0, xyz)$ | $=$ | $\hat{\delta}(q_0, xuvwz)$ | $y = uvw$ |
| 6. | | $=$ | $\hat{\delta}(\hat{\delta}(\hat{\delta}(q_0, xu), v), wz)$, | prop. of $\hat{\delta}$ |
| 7. | | $=$ | $\hat{\delta}(\hat{\delta}(p(i), v), wz)$, | (1) |
| 8. | | $=$ | $\hat{\delta}(\hat{\delta}(p(i), v^m), wz)$, | (3) |
| 9. | | $=$ | $\hat{\delta}(\hat{\delta}(\hat{\delta}(q_0, xu), v^m), wz)$, | (1) |
| 10. | | $=$ | $\hat{\delta}(xuv^m wz)$, | prop. of $\hat{\delta}$ |
| 11. | $\hat{\delta}(xyz)$ | $\in$ | $F$, | $xyz \in B, L(M) = B$ |
| 12. | $\hat{\delta}(xuv^m wz)$ | $\in$ | $F$, | $\hat{\delta}(xuv^m wz) = \hat{\delta}(xyz)$, (5)-(10) |
| 13. | $xuv^m wz$ | $\in$ | $B$, | $L(M) = B$, note that $m$ is arbitrary |

**2.** Interpretation: The pumping lemma says that every regular language $B$ has an associated integer, $k$, such that for any string $s \in B$, any substring of $s$ of length $k$ or greater has a substring that can be repeated as many times as you like (including 0) and the resulting string is still in the language.

**a.** Adding extra copies of this string to the original $s$ is the "pumping" part of the pumping lemma.

**b.** The bit about dividing $s$ into $x$, $y$, and $z$ says that this pumping property can be applied to any substring that you like, as long as it's long enough.

**C.** The contrapositive of the pumping lemma:

Let $B$ be a language. *If* for any integer $k$, you can find a string $xyz \in B$ with $|y| \geq k$ such that there is no way to choose $u$, $v$, and $w$ with $y = uvw$ so that $xuv^m wz \in B$ for any $m > 0$, *then* $B$ is not regular.

This contrapositive formulation provides a very useful way to prove that a language is not regular. Kozen views it as a game. You want to prove that a language is not regular, the "demon" (i.e. a hypothetical adversary) plays the other side of the game. If you can win no matter what moves the demon makes, then the language is not regular. Here are the "rules" of the game:

**1.** The demon choses $k$.

**2.** You pick $x$, $y$, and $z$ such that $xyz \in B$ and $y \geq k$. Your goal is to choose $y$ so that there is no way to make it any shorter and still get a string in $B$ or there is no way to replicate some piece of it and still get a string in $B$. You choose $x$ and $z$ in such a way as to force the demon to work on a part of the string that you know can't be altered.

**3.** The demon picks $u$, $v$, and $w$ such that $uvw = y$.

**4.** You pick $m \geq 0$ such that $xuv^m wz \notin B$.

This kind of game perspective is common in applications of formal automata to verification problems. In this case, the automaton is the model of all possible system behaviours. The demon is trying to make the system fail. For example, the demon can decide the outcomes of non-deterministic choices to try to make something bad happen. For example, the demon could choose the inputs that will be applied to the system. In a concurrent system where there are several

computers working at the same time, the demon may choose an ordering of events that would cause the system to fail. The verification task is to show that the system works no matter what the demon chooses.

**III.** One more example: let $\Sigma = \{a\}$; let $B = \{w \in \Sigma^* | \mathsf{prime}(|a|)\}$, where $\mathsf{prime}(n)$ is true iff $n$ is a prime number. We'll prove that $B$ is not regular using the contrapositive (game with a demon) form of the pumping lemma.

**A.** Demon's choice: $k$
Let $k$ be the pumping lemma constant for $B$.

**B.** Our choice: $x$, $y$, and $x$

**1.** Let $p$ be the smallest prime that is greater than $(k+1)! + 1$. Such a $p$ exists because there are an infinite number of primes.

**2.** Note that for $n \in [(k+1)! + 2 \ldots (k+1)! + k + 1]$, Let $r = n - (k+1)!$. Because $2 \leq r \leq k+1$, $(k+1)!/r$ is an integer. Accordingly, $r$, and $1 + (k+1)!/r$ are integer factors of $n$. Thus, $n$ is composite,

**3.** It follows from III.B.2 that $p > (k+1)! + (k+1)$.

**4.** We choose $x = a^{(k+1)!+2}$, $y = a^{p-|x|}$, and $z = \epsilon$.
Because $p > (k+1)! + (k+1)$, $|y| \geq k$. Note that $xyz = a^p \in B$.

**C.** Demon's choice: $u$, $v$, and $w$
By the assumption that $B$ is regular, there exist strings $u$, $v$, and $w$ such that $y = uvw$, $|v| > 0$, and for all $m \geq 0$, $xuv^m wz \in B$ (the pumping lemma).

**D.** Our choice: $m = 0$. Because $|x| = (k+1)! + 2$, $|xuwz| \geq (k+1)! + 2$. Likewise, because $|v| > 0$, $|xuwz| < p$. But, we've $p$ is the smallest prime greater than $(k+1)! + 1$. Thus $|xuwz|$ is composite, and $xuwz \notin B$.

Because we have a winning strategy no matter what the demon chooses, we have shown that $B$ is not regular.