Ex C3 Part 2

- **Task 1**: Liisa realized that for any day, she receives m important (i.e., non-junk) e-mail messages followed by n junk messages, where $m + n = 2k$ for some integer $k \geq 1$.
- **Task 2**: Mikko realized that whenever he wins the card game m consecutive times, he loses n consecutive times after that, where $n = 2m + 1$.

Unit C4: Overview

- Distinguish regular and non-regular languages
  - Pumping Lemma (for regular languages)
- Understand and use the properties of regular languages
  - Closure properties
  - Equivalence of regular languages
- Preview Exercise C4 “Regular Practice”

Regular?

Preview

- Top-down approach to paper writing
- Finite language
- 0^n1^n
- $\{0^m1^n \mid m, n \geq 0\}$
- $\{0^m1^n \mid m \neq n\}$
- $\{w\$w^R \mid w^R$ is the reverse of $w \in \{0, 1\}^*\}$
- $\{ww^R \mid w^R$ is the reverse of $w \in \{0, 1\}^*\}$
- $\{ww \mid w \in \{0, 1\}^*\}$
- $\{ww^Rw \mid w^R$ is the reverse of $w \in \{0, 1\}^*\}$

Analysis of “Regular”ness

- To Show that L is “regular”
  - Proof by existence: Give a regular grammar, RegExp, or FSA
- To Show that L is *not* “regular”
  - Need to prove that no regular grammar (or RegExp or FA) can generate the language
  - Demonstrate that some property of regular languages cannot hold

Main Property of Regular Languages

- *Uncoordinated* repetition
- Inputs longer than the number of FA states $\Rightarrow$ Some state(s) must be repeated
- Any number of that repetition must result in an acceptable input.

cf. pumping gas for free

Pumping Lemma (for regular languages)

To show that a language is *not* regular

- For any <u>infinite</u> regular language L,
- there exists a positive integer n_0 such that
- for any $w \in L$ such that $|w| \geq n_0$,
- there exists a decomposition $w = xyz$ where $|xy| \leq n_0$ and $|y| \geq 1$ such that
- for any $i \geq 0$,
- $xy^iz \in L$

Game-Theoretic Interpretation of FOL

cf. logic-structure connection

- $\forall x \exists y$ (x kicks y)
  - Falsifier (tries to crack $\forall$): Choose x
  - Verifier (tries to support $\exists$): Choose y, based on x
  - Check whether "x kicks y" is true
- $\exists x \forall y$ (x kicks y)
  - Verifier: Choose x
  - Falsifier: Choose y, based on x
  - Check whether "x kicks y" is true

To satisfy: Play a verifier
To reject: Play a falsifier

Group Exercise 1

Satisfy/reject the following

A. Everyone in this class is taking at least one upper-level CS course.

B. Some cat loves every dog at some point.

C. Some cat loves every dog at any point.

Procedure:
- Take the role of either the verifier or falsifier
- To satisfy, the verifier must win.
- To reject, the falsifier must win.

Revisited

Main Property of Regular Languages

- *Uncoordinated* repetition
- Inputs longer than the number of FA states $\Rightarrow$ Some state(s) must be repeated
- Any number of that repetition must result in an acceptable input.

Logic Game for Pumping Lemma

Between a verifier (for $\exists$) and a falsifier (for $\forall$)

- Falsifier: Choose an infinite language L
- Verifier: Choose a positive integer n_0 such that
- Falsifier: Choose $w \in L$ such that $|w| \geq n_0$,
- Verifier: Choose a decomposition $w = xyz$ where $|xy| \leq n_0$ and $|y| \geq 1$ such that
- Falsifier: Choose $i \geq 0$,
- Check whether $xy^iz \in L$

Meaning of $|xy| \leq n_0$

To satisfy: Play a verifier (wrong)
To reject: Play a falsifier (correct)

Misusing Pumping Lemma

Play a verifier (Note: *not* proving that L is regular; *not useful*)

- To show that 0^m1^n has "regular"ness
- Choose $n_0 = 1$
- Anticipate any $0^m1^n \in L$ such that $m + n \geq 1$
- Choose a decomposition $w = xyz$:
  - Case 1 ($m = 0$): 1^m 1^c 1^d ($c \geq 1$)
  - Case 2 ($m \neq 0$): 0^a 0^b 1^n ($b \geq 1$)
- Anticipate any $i \geq 0$
- $0^a0^{b\times i}1^n \in L$, $0^m1^{c\times i}1^d \in L$

Using Pumping Lemma

Play a falsifier: This is how this lemma is used.

- To show that 0^n1^n is *not* regular
- Anticipate any $n_0 \geq 1$
- Choose $0^n1^n \in L$ such that $2n \geq n_0$
- Anticipate any decomposition $w = xyz$:
  - Case 1: 0^a 0^b 1^n ($b \geq 1$)
  - Case 2: 0^n 1^c 1^d ($c \geq 1$)
  - Case 3: 0^a (0^b1^c) 1^d ($b + c \geq 1$)
- Choose $i = 2$
- $0^a0^{2b}1^n \notin L$, $0^n1^{2c}1^d \notin L$, and $0^a(0^b1^c)^21^d \notin L$

Regular?

- Top-down approach to paper writing
- Finite language
- 0^n1^n
- $\{0^m1^n \mid m, n \geq 0\}$
- $\{0^m1^n \mid m \neq n\}$
- $\{w\$w^R \mid w^R$ is the reverse of $w \in \{0, 1\}^*\}$
- $\{ww^R \mid w^R$ is the reverse of $w \in \{0, 1\}^*\}$
- $\{ww \mid w \in \{0, 1\}^*\}$
- $\{ww^Rw \mid w^R$ is the reverse of $w \in \{0, 1\}^*\}$

Group Exercise 2

Satisfy/reject the following

A. $\{w\$w^R \mid w^R$ is the reverse of $w \in \{0, 1\}^*\}$

B. $\{ww^R \mid w^R$ is the reverse of $w \in \{0, 1\}^*\}$

Procedure:
- Take the role of either the verifier or falsifier
- To satisfy, the verifier must win.
- To reject, the falsifier must win.

Closure

Closed under
- Concatenation
- Union
- Kleene closure
- Complement
- Intersection

Equivalence of Regular Languages

- Equivalence of States
  - Equivalent behavior for all strings with respect to acceptance
  - To find out: compare every pair of states against all strings (systematically)
- Minimization of DFAs
  - Merge equivalent states ⇒ partition
- Equivalence of regular languages
  - Merge two DFAs and test equivalence of the two start states

Unit Summary

- Regular vs. non-regular
  - To show regular: Construct a regular grammar, RegExp, or a FSA
  - To show not regular: Use the Pumping Lemma
- Properties
  - Closed under most common operations
  - Unique minimization possible
  - Easy to manipulate regular languages and their specification/processing mechanisms

Summary Question

- The Pumping Lemma is tricky. You must have questions. What are they?
- List other questions as well, if you have.