CSE 311 Section 10

Final Review

Administrivia

Announcements & Reminders

- HW7 Regrade Requests
  - Grades out soon
  - Submit a regrade request if something was graded incorrectly

- HW8
  - Due Tomorrow, Friday 12/8 @ 10pm
  - Late due date 12/11 @ 10pm

- Final Exam
  - Monday 12/11 @ 4:30pm-6:20 @ KNE 130
  - Fill out Form for Conflict Exam

Irregularity

A note for your final...

You **WILL** have a question on the final exam where you will have a choice between either **proving a language is irregular** OR **proving a set is uncountable**.

For section today, we will go over how to prove a language is irregular. There is also a problem in the handout on proving a set is uncountable you can review if you prefer to prepare for that question. You should pick whichever you think is easier for you, and make sure you are prepared to do it on the final exam!

Irregularity Template

Claim: L is an irregular language.

Proof: Suppose, for the sake of contradiction, that L is regular. Then there is a DFA M such that M accepts exactly L.

Let S = [TODO] (S is an infinite set of strings)
Because the DFA is finite, there are two (different) strings x, y in S such that x and y go to the same state when read by M. [TODO] (We don't get to choose x, y, but we can describe them based on that set S we just defined)

Consider the string z = [TODO] (We do get to choose z depending on x, y)

Since x, y led to the same state and M is deterministic, xz and yz will also lead to the same state q in M. Observe that xz = [TODO], so $xz \in L$ but yz = [TODO], so $yz \notin L$. Since q is can be only one of an accept or reject state, M does not actually recognize L. That's a contradiction!

Therefore, L is an irregular language.

Irregularity Example from Lecture

Claim: $\{0^k1^k : k \geq 0\}$ is an irregular language.

Proof: Suppose, for the sake of contradiction, that $L = \{0^k1^k : k \geq 0\}$ is regular. Then there is a DFA M such that M accepts exactly L.

Let $S = \{0^k : k \geq 0\}$
Because the DFA is finite, there are two (different) strings x, y in S such that x and y go to the same state when read by M. Since both are in S, $x = 0^a$ for some integer $a \geq 0$, and $y = 0^b$ for some integer $b \geq 0$, with $a \neq b$.

Consider the string $z = 1^a$.

Since x, y led to the same state and M is deterministic, xz and yz will also lead to the same state q in M. Observe that $xz = 0^a1^a$, so $xz \in L$ but $yz = 0^b1^a$, so $yz \notin L$. Since q is can be only one of an accept or reject state, M does not actually recognize L. That's a contradiction!

Therefore, L is an irregular language.

Problem 1 – Irregularity

a) Let $\Sigma = \{0, 1\}$. Prove that $\{0^n1^n0^n : n \geq 0\}$ is not regular.

b) Let $\Sigma = \{0, 1, 2\}$. Prove that $\{0^n(12)^m : n \geq m \geq 0\}$ is not regular.

Work on this problem with the people around you.

Problem 1 – Irregularity

(a) Let $\Sigma = \{0, 1\}$. Prove that $\{0^n1^n0^n : n \geq 0\}$ is not regular.

Claim: $\{0^n1^n0^n : n \geq 0\}$ is an irregular language.

Proof: Suppose, for the sake of contradiction, that $L = \{0^n1^n0^n : n \geq 0\}$ is regular. Then there is a DFA M such that M accepts exactly L.

Let S = [TODO]
Because the DFA is finite, there are two (different) strings x, y in S such that x and y go to the same state when read by M. [TODO] .

Consider the string z = [TODO] .

Since x, y led to the same state and M is deterministic, xz and yz will also lead to the same state q in M. Observe that xz = [TODO] , so $xz \in L$ but yz = [TODO] , so $yz \notin L$. Since q is can be only one of an accept or reject state, M does not actually recognize L. That's a contradiction!

Therefore, L is an irregular language.

Problem 1 – Irregularity

(b) Let $\Sigma = \{0, 1, 2\}$. Prove that $\{0^n(12)^m : n \geq m \geq 0\}$ is not regular.

Claim: $\{0^n(12)^m : n \geq m \geq 0\}$ is an irregular language.

Proof: Suppose, for the sake of contradiction, that $L = \{0^n(12)^m : n \geq m \geq 0\}$ is regular. Then there is a DFA M such that M accepts exactly L.

Let S = [TODO]
Because the DFA is finite, there are two (different) strings x, y in S such that x and y go to the same state when read by M. [TODO] .

Consider the string z = [TODO] .

Since x, y led to the same state and M is deterministic, xz and yz will also lead to the same state q in M. Observe that xz = [TODO] , so $xz \in L$ but yz = [TODO] , so $yz \notin L$. Since q is can be only one of an accept or reject state, M does not actually recognize L. That's a contradiction!

Therefore, L is an irregular language.

Final Review

Problem 5 – Review: Translations

Translate the following sentences into logical notation if the English statement is given or to an English statement if the logical statement is given, taking into account the domain restriction. Let the domain of discourse be students and courses. Use predicates Student, Course, CseCourse to do the domain restriction. You can use Taking(x, y) which is true if and only if x is taking y. You can also use RobbieTeaches(x) if and only if Robbie teaches x and ContainsTheory(x) if and only if x contains theory.

(a) Every student is taking some course.

(b) There is a student that is not taking every cse course.

(c) Some student has taken only one cse course.

(d) $\forall x[(\text{Course}(x) \land \text{RobbieTeaches}(x)) \rightarrow \text{ContainsTheory}(x)]$

(e) $\exists x\ \text{CseCourse}(x) \land \text{RobbieTeaches}(x) \land \text{ContainsTheory}(x) \land \forall y((\text{CseCourse}(y) \land \text{RobbieTeaches}(y)) \rightarrow x = y)$

Work on this problem with the people around you.

Problem 5 – Review: Translations

a) Every student is taking some course.

b) There is a student that is not taking every cse course.

c) Some student has taken only one cse course.

d) $\forall x[(Course(x) \land RobbieTeaches(x)) \rightarrow ContainsTheory(x)]$

e) $\exists x\ CseCourse(x) \land RobbieTeaches(x) \land ContainsTheory(x) \land \forall y((CseCourse(y) \land RobbieTeaches(y)) \rightarrow x = y)$

Problem 6 – Review: Functions

Let $f : X \to Y$ be a function. For a subset C of X, define $f(C)$ to be the set of elements that f sends C to. In other words, $f(C) = \{f(c) : c \in C\}$.

Let A, B be subsets of X. Prove that $f(A \cap B) \subseteq f(A) \cap f(B)$.

Work on this problem with the people around you.

Problem 6 – Review: Functions

Let $f : X \to Y$ be a function. For a subset C of X, define $f(C)$ to be the set of elements that f sends C to. In other words, $f(C) = \{f(c) : c \in C\}$.

Let A, B be subsets of X. Prove that $f(A \cap B) \subseteq f(A) \cap f(B)$.

Problem 7 – Review: Induction

a) A Husky Tree is a tree built by the following definition:

Basis: A single gold node is a Husky Tree.
Recursive Rules:
1. Let T1, T2 be two Husky Trees, both with root nodes colored gold. Make a new purple root node and attach the roots of T1, T2 to the new node to make a new Husky Tree.
2. Let T1, T2 be two Husky Trees, both with root nodes colored purple. Make a new purple root node and attach the roots of T1, T2 to the new node to make a new Husky Tree.
3. Let T1, T2 be two Husky Trees, one with a purple root, the other with a gold root. Make a new gold root node, and attach the roots of T1, T2 to the new node to make a new Husky Tree.

Use structural induction to show that for every Husky Tree: if it has a purple root, then it has an even number of leaves and if it has a gold root, then it has an odd number of leaves.

Work on this problem with the people around you.

Work on this problem with the people around you.

Problem 7 – Review: Induction

Let $P(x)$ be. We show $P(x)$ holds for all $x \in S$ by structural induction.

Base Case: Show $P(x)$
[Do that for every base cases x in S.]
Let y be an arbitrary element of S not covered by the base cases. By the exclusion rule, y = <recursive rules>

Inductive Hypothesis: Suppose $P(x)$
[Do that for every x listed as in S in the recursive rules.]

Inductive Step: Show $P()$ holds for y.
[You will need a separate case/step for every recursive rule.]

Therefore $P(x)$ holds for all $x \in S$ by the principle of induction.

Problem 7 – Review: Induction

(b) Use induction to prove that for every positive integer n, $1 + 5 + 9 + \cdots + (4n - 3) = n(2n - 1)$

Work on this problem with the people around you.

Problem 7 – Review: Induction

(b) Use induction to prove that for every positive integer n, $1 + 5 + 9 + \cdots + (4n - 3) = n(2n - 1)$

Let $P(n)$ be "". We show $P(n)$ holds for (some) n by induction on n.

Base Case: $P(b)$:

Inductive Hypothesis: Suppose $P(k)$ holds for an arbitrary $k \geq b$.

Inductive Step: Goal: Show $P(k+1)$:

Conclusion: Therefore, $P(n)$ holds for (some) n by the principle of induction.

Problem 8 – Review: Languages

(a) Construct a regular expression that represents binary strings where no occurrence of 11 is followed by a 0.

(b) Construct a CFG that represents the following language: $\{1^x 2^y 3^y 4^x : x, y \geq 0\}$

(c) Construct a DFA that recognizes the language of all binary strings which, when interpreted as a binary number, are divisible by 3. e.g. 11 is 3 in base-10, so should be accepted while 111 is 7 in base-10, so should be rejected. The first bit processed will be the most-significant bit.
Hint: you need to keep track of the remainder %3. What happens to a binary number when you add a 0 at the end? A 1? It's a lot like a shift operation...

(d) Construct a DFA that recognizes the language of all binary strings with an even number of 0's and each 0 is (immediately) followed by at least one 1.

Work on this problem with the people around you.

Problem 8 – Review: Languages

(a) Construct a regular expression that represents binary strings where no occurrence of 11 is followed by a 0.

(b) Construct a CFG that represents the following language: $\{1^x 2^y 3^y 4^x : x, y \geq 0\}$

Problem 8 – Review: Languages

(c) Construct a DFA that recognizes the language of all binary strings which, when interpreted as a binary number, are divisible by 3. e.g. 11 is 3 in base-10, so should be accepted while 111 is 7 in base-10, so should be rejected. The first bit processed will be the most-significant bit.

Hint: you need to keep track of the remainder %3. What happens to a binary number when you add a 0 at the end? A 1? It's a lot like a shift operation...

Problem 8 – Review: Languages

(d) Construct a DFA that recognizes the language of all binary strings with an even number of 0’s and each 0 is (immediately) followed by at least one 1.

That’s All, Folks!

Thanks for coming to section this week!
Any questions?