Section 10

CSE 311 - Sp 2022

Announcements and Reminders

- HW8
  - Last homework!
  - Due yesterday, late due date Saturday 6/4 @ 10pm
- Final Review Session:
  - Saturday 6/4 @ 1-3 pm
  - In-person in CSE2 G01
- Final Exam Info:
  - In-person on Monday 6/6 @ 12:30 pm
  - Majority of students in Kane 120, some students in smaller extra location for increased distancing
  - If you will be in the other location, you should have received an email. Please let us know if you have any questions or concerns!

IMPORTANT!

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

For section today, we will go over how to prove a language is irregular. There is also a problem 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

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: $\{0^k 1^k : k \geq 0\}$ is an irregular language.

Proof: Suppose, for the sake of contradiction, that $L = \{0^k 1^k : k \geq 0\}$ 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 $xz = 0^a 1^a$, so $xz \in L$ but $yz = 0^b 1^a$, 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 $\{0^n 1^n 0^n : n \geq 0\}$ 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, and then we'll go over it together!

Problem 1 - Irregularity

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

Claim: $\{0^n 1^n 0^n : n \geq 0\}$ is an irregular language.

Proof: Suppose, for the sake of contradiction, that $L = \{0^n 1^n 0^n : n \geq 0\}$ 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

(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 $S = \{0^n1^n : n \geq 0\}$
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. [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

(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 $S = \{0^n1^n : n \geq 0\}$
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, $x = 0^a1^a$ for some integer $a \geq 0$, and $y = 0^b1^b$ for some integer $b \geq 0$, with $a \neq b$.

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

(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 $S = \{0^n1^n : n \geq 0\}$
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, $x = 0^a1^a$ for some integer $a \geq 0$, and $y = 0^b1^b$ for some integer $b \geq 0$, with $a \neq b$.

Consider the string $z = 0^a$.

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 = [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

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

Claim: $\{0^n 1^n 0^n : n \geq 0\}$ is an irregular language.

Proof: Suppose, for the sake of contradiction, that $L = \{0^n 1^n 0^n : n \geq 0\}$ is regular. Then there is a DFA M such that M accepts exactly L.

Let $S = \{0^n 1^n : n \geq 0\}$
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, $x = 0^a 1^a$ for some integer $a \geq 0$, and $y = 0^b 1^b$ for some integer $b \geq 0$, with $a \neq b$.

Consider the string $z = 0^a$.

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 $xz = 0^a 1^a 0^a$, so $xz \in L$ but $yz = 0^b 1^b 0^a$, 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

(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.

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 $S = \{0^n : n \geq 0\}$
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. [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 $S = \{0^n : n \geq 0\}$
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 $a > b$.

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 $S = \{0^n : n \geq 0\}$
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 $a > b$.

Consider the string $z = (12)^a$.

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 = [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 $S = \{0^n : n \geq 0\}$
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 $a > b$.

Consider the string $z = (12)^a$.

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 $xz = 0^a(12)^a$, so $xz \in L$ but $yz = 0^b(12)^a$, 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.

Cardinality (Uncountability)

Some Definitions

A function $\boldsymbol{f}: \boldsymbol{A} \to \boldsymbol{B}$ maps every element of $\boldsymbol{A}$ to one element of $\boldsymbol{B}$
$\boldsymbol{A}$ is the "domain", $\boldsymbol{B}$ is the "co-domain"

- One-to-one (aka injection)
  - A function $\boldsymbol{f}:\boldsymbol{A} \to \boldsymbol{B}$ is one-to-one iff $\forall \boldsymbol{a} \forall \boldsymbol{b}(\boldsymbol{f}(\boldsymbol{a}) = \boldsymbol{f}(\boldsymbol{b}) \to \boldsymbol{a} = \boldsymbol{b})$
  - Every output has at most one possible input
- Onto (aka surjection)
  - A function $\boldsymbol{f}:\boldsymbol{A} \to \boldsymbol{B}$ is onto iff $\forall \boldsymbol{b} \in \boldsymbol{B} \exists \boldsymbol{a} \in \boldsymbol{A}(\boldsymbol{b} = \boldsymbol{f}(\boldsymbol{a}))$
  - Every output has at least one input that maps to it.
- Bijection
  - A function $\boldsymbol{f}:\boldsymbol{A} \to \boldsymbol{B}$ is a bijection iff $\boldsymbol{f}$ is one-to-one and onto
  - A bijection maps every element of the domain to exactly one element of the co-domain, and every element of the co-domain to exactly one element of the domain.

Two sets $\boldsymbol{A}, \boldsymbol{B}$ have the same size (same cardinality) if and only if there is a bijection $\boldsymbol{f}: \boldsymbol{A} \to \boldsymbol{B}$

Problem 2 - Cardinality

(a) You are a pirate. You begin in a square on a 2D grid which is infinite in all directions. In other words, wherever you are, you may move up, down, left, or right. Some single square on the infinite grid has treasure on it. Find a way to ensure you find the treasure in finitely many moves.

(b) Prove that $\{3x : x \in \mathbb{N}\}$ is countable

(c) Prove that the set of irrational numbers is uncountable. Hint: Use the fact that the rationals are countable and that the reals are uncountable.

(d) Prove that P($\mathbb{N}$) is uncountable.

Work on parts (a) and (b) of this problem with the people around you, and then we'll go over it together!

Problem 2 - Cardinality

(a) You are a pirate. You begin in a square on a 2D grid which is infinite in all directions. In other words, wherever you are, you may move up, down, left, or right. Some single square on the infinite grid has treasure on it. Find a way to ensure you find the treasure in finitely many moves.

(b) Prove that $\{3x : x \in \mathbb{N}\}$ is countable

Problem 2 - Cardinality

(a) You are a pirate. You begin in a square on a 2D grid which is infinite in all directions. In other words, wherever you are, you may move up, down, left, or right. Some single square on the infinite grid has treasure on it. Find a way to ensure you find the treasure in finitely many moves.

Explore the square you are currently on. Explore the unexplored perimeter of the explored region until you find the treasure (your path will look a bit like a spiral).

(b) Prove that $\{3x : x \in \mathbb{N}\}$ is countable

Problem 2 - Cardinality

(a) You are a pirate. You begin in a square on a 2D grid which is infinite in all directions. In other words, wherever you are, you may move up, down, left, or right. Some single square on the infinite grid has treasure on it. Find a way to ensure you find the treasure in finitely many moves.

Explore the square you are currently on. Explore the unexplored perimeter of the explored region until you find the treasure (your path will look a bit like a spiral).

(b) Prove that $\{3x : x \in \mathbb{N}\}$ is countable

We can enumerate the set as follows:

f(0) = 0
f(1) = 3
f(2) = 6
f(i) = 3i

Since every natural number appears on the left, and every number in S appears on the right, this enumeration spans both sets, so S is countable

Problem 2 - Cardinality

(c) Prove that the set of irrational numbers is uncountable. Hint: Use the fact that the rationals are countable
and that the reals are uncountable.

(d) Prove that $P(\mathbb{N})$ is uncountable.

Problem 2 - Cardinality

(c) Prove that the set of irrational numbers is uncountable. Hint: Use the fact that the rationals are countable
and that the reals are uncountable.

We first prove that the union of two countable sets is countable. Consider two arbitrary countable sets C_1 and C_2. We can enumerate $C_1 \cup C_2$ by mapping even natural numbers to C_1 and odd natural numbers to C_2.
Now, assume that the set of irrationals is countable. Then the reals would be countable, since the reals are the union of the irrationals (countable by assumption) and the rationals (countable). However, we have already shown that the reals are uncountable, which is a contradiction. Therefore, our assumption that the set of irrationals is countable is false, and the irrationals must be uncountable.

(d) Prove that $P(\mathbb{N})$ is uncountable.

Problem 2 - Cardinality

(c) Prove that the set of irrational numbers is uncountable. Hint: Use the fact that the rationals are countable

and that the reals are uncountable.

We first prove that the union of two countable sets is countable. Consider two arbitrary countable sets C_1 and C_2. We can enumerate $C_1 \cup C_2$ by mapping even natural numbers to C_1 and odd natural numbers to C_2.
Now, assume that the set of irrationals is countable. Then the reals would be countable, since the reals are the union of the irrationals (countable by assumption) and the rationals (countable). However, we have already shown that the reals are uncountable, which is a contradiction. Therefore, our assumption that the set of irrationals is countable is false, and the irrationals must be uncountable.

(d) Prove that $P(\mathbb{N})$ is uncountable.

Assume for the sake of contradiction that $P(\mathbb{N})$ is countable. This means we can define an enumeration of elements S_i in P. Let s_i be the binary set representation of S_i in N. For example, for the set 0, 1, 2, the binary set representation would be 111000 . . .
We then construct a new subset $X \subset \mathbb{N}$ such that $x[i] = s_i[i]$ (that is, $x[i]$ is 1 if $s_i[i]$ is 0, and $x[i]$ is 0 otherwise). Note that X is not any of S_i, since it differs from S_i on the ith natural number. However, X still represents a valid subset of the natural numbers, which means our enumeration is incomplete, which is a contradiction. Since the above proof works for any listing of $P(\mathbb{N})$, no listing can be created for $P(\mathbb{N})$, and therefore $P(\mathbb{N})$ is uncountable.

Final Review

Problem 5 - 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) $\forall x[(\texttt{Course}(x) \wedge \texttt{RobbieTeaches}(x)) \rightarrow \texttt{ContainsTheory}(x)]$

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

Problem 5 - 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) $\forall x[(\texttt{Course}(x) \land \texttt{RobbieTeaches}(x)) \rightarrow \texttt{ContainsTheory}(x)]$

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

Problem 5 - Translations

(a) Every student is taking some course.

$\forall x \exists y(\text{Student}(x) \rightarrow [\text{Course}(y) \wedge \text{Taking}(x, y)])$

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

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

(d) $\forall x[(\texttt{Course}(x) \wedge \texttt{RobbieTeaches}(x)) \rightarrow \texttt{ContainsTheory}(x)]$

(e) $\exists x\, \texttt{CseCourse}(x) \wedge \texttt{RobbieTeaches}(x) \wedge \texttt{ContainsTheory}(x) \wedge \forall y((\texttt{CseCourse}(y) \wedge \texttt{RobbieTeaches}(y)) \rightarrow x = y)$

Problem 5 - Translations

(a) Every student is taking some course.

$\forall x \exists y(\text{Student}(x) \rightarrow [\text{Course}(y) \land \text{Taking}(x, y)])$

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

$\exists x \forall y[\text{Student}(x) \land (\text{CseCourse}(y) \rightarrow \neg \text{Taking}(x, y))]$

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

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

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

Problem 5 - Translations

(a) Every student is taking some course.

$\forall x \exists y(\text{Student}(x) \rightarrow [\text{Course}(y) \wedge \text{Taking}(x, y)])$

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

$\exists x \forall y[\text{Student}(x) \wedge (\text{CseCourse}(y) \rightarrow \neg \text{Taking}(x, y))]$

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

$\exists x \exists y[\text{Student}(x) \wedge \text{CseCourse}(y) \wedge \text{Taking}(x, y) \wedge \forall z((\text{CseCourse}(z) \wedge \text{Taking}(x, z)) \rightarrow y = z))]$

(d) $\forall x[(\texttt{Course}(x) \wedge \texttt{RobbieTeaches}(x)) \rightarrow \texttt{ContainsTheory}(x)]$

Every course taught by Robbie contains theory.

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

Problem 5 - Translations

(a) Every student is taking some course.

$\forall x \exists y(\text{Student}(x) \rightarrow [\text{Course}(y) \land \text{Taking}(x, y)])$

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

$\exists x \forall y[\text{Student}(x) \land (\text{CseCourse}(y) \rightarrow \neg \text{Taking}(x, y))]$

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

$\exists x \exists y[\text{Student}(x) \land \text{CseCourse}(y) \land \text{Taking}(x, y) \land \forall z((\text{CseCourse}(z) \land \text{Taking}(x, z)) \rightarrow y = z)]$

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

Every course taught by Robbie contains theory.

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

There is only one cse course that Robbie teaches and that course contains theory.

Problem 6 - 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 6 - 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 $f(A \cap B) \subseteq f(A) \cap f(B)$.

Let $y \in f(A \cap B)$ be arbitrary.

Then there exists some element $x \in A \cap B$ such that $f(x) = y$.
Then by the definition of intersection, $x \in A$ and $x \in B$. Then $f(x) \in f(A)$ and $f(x) \in f(B)$. Thus $y \in f(A)$ and $y \in f(B)$.

By definition of intersection, $y \in f(A) \cap f(B)$.

Since y was arbitrary, $f(A \cap B) \subseteq f(A) \cap f(B)$.

Problem 7 - 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.

Problem 7 - Induction (a)

Let P(T) be "if T has a purple root, then it has an even number of leaves and if T has a gold root, then it has an odd number of leaves." We show P(T) holds for all Husky Trees T by structural induction.

Base Case: Let T be a Husky Tree made from the basis step. By the definition of Husky Tree, T must be a single gold node. That node is also a leaf node (since it has no children) so there are an odd number (specifically, 1) of leaves, as required for a gold root node.

Inductive Hypothesis: Let T1, T2 be arbitrary Husky Trees, and suppose P(T1) and P(T2).

Inductive Step: We will have separate cases for each possible rule.
Rule 1: Suppose T1 and T2 both have gold roots. By the recursive rule, T has a purple root. By inductive hypothesis on T1, since T1's root is gold, it has an odd number of leaves. Similarly by IH, T2 has an odd number of leaves. T's leaves are exactly the leaves of T1 and T2, so the total number of leaves in T is the sum of two odd numbers, which is even. Thus T has an even number of leaves, as is required for a purple root. ThusP(T) holds.
Rule 2: Suppose T1 and T2 both have purple roots. By the recursive rule, T has a purple root. By inductive hypothesis on T1, since T1's root is purple, it has an even number of leaves. Similarly by IH, T2 has an even number of leaves. T's leaves are exactly the leaves of T1 and T2, so the total number of leaves in T is the sum of two even numbers, which is even. Thus T has an even number of leaves, as is required for a purple root. Thus P(T) holds.
Rule 3: Suppose T1 and T2 have opposite colored roots. Let T1 be the one with a gold root, and T2 the one with the purple root. By the recursive rule, T has a gold root. By inductive hypothesis on T1, since T1's root is gold, it has an odd number of leaves. Similarly, by IH, T2 has an even number of leaves since it has a purple root. T's leaves are exactly the leaves of T1 and T2, so the total number of leaves in T is the sum of an odd number and an even number, which is odd. Thus T has an odd number of leaves, as is required for a gold root. Thus P(T) holds.

Conclusion: By the principle of induction, we have that for every Husky Tree, T: P(T) holds.

Problem 7 - Induction

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

Problem 7 - Induction (b)

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

For $n \in \mathbb{Z}^+$ let $P(n)$ be "$1 + 5 + 9 + \cdots + (4n - 3) = n(2n - 1)$." We show $P(n)$ for all $n \in \mathbb{Z}^+$ by induction on n.

Base Case: We have $1 = 1(1) = 1(2 - 1)$ which is $P(1)$ so the base case holds.

Inductive Hypothesis: Suppose $P(k)$ holds for some arbitrary integer $k \geq 1$.

Inductive Step: Goal: Show $1 + 5 + 9 + \cdots + (4(k + 1) - 3) = (k + 1)(2(k + 1) - 1)$.
We have:

$$
\begin{aligned}
1 + 5 + 9 + \cdots + (4(k + 1) - 3) &= 1 + 5 + 9 + \cdots + (4k - 3) + (4(k + 1) - 3) \\
&= k(2k - 1) + (4(k + 1) - 3) \\
&\qquad \text{[Inductive Hypothesis]} \\
&= k(2k - 1) + (4k + 1) = 2k\ 2 + 3k + 1 = (k + 1)(2k + 1) \qquad \text{[Factor]} \\
&= (k + 1)(2(k + 1) - 1)
\end{aligned}
$$

This proves $P(k + 1)$.

Conclusion: $P(n)$ holds for all $n \in \mathbb{Z}^+$ by the principle of induction.

Problem 8 - 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.

Problem 8 - 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 - Languages

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

$(0^* (10)^*)^* 1^*$

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

Problem 8 - Languages

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

$(0^* (10)^*)^* 1^*$

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

$\mathbf{S} \rightarrow 1\mathbf{S}4 \mid \mathbf{T}$
$\mathbf{T} \rightarrow 2\mathbf{T}3 \mid \varepsilon$

Problem 8 - Languages

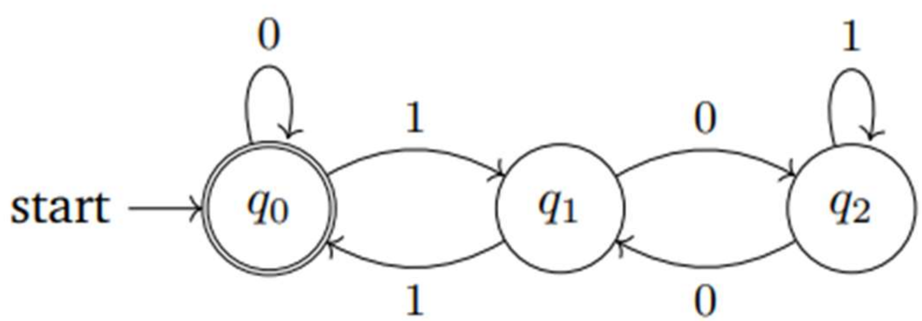
(c) Construct a DFA that recognizes the language of all binary strings which, when interpreted as a binary number, are divisible by 3. e.g. 11 is 3 in base-10, so should be accepted while 111 is 7 in base-10, so should be rejected. The first bit processed will be the most-significant bit.

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 - 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...

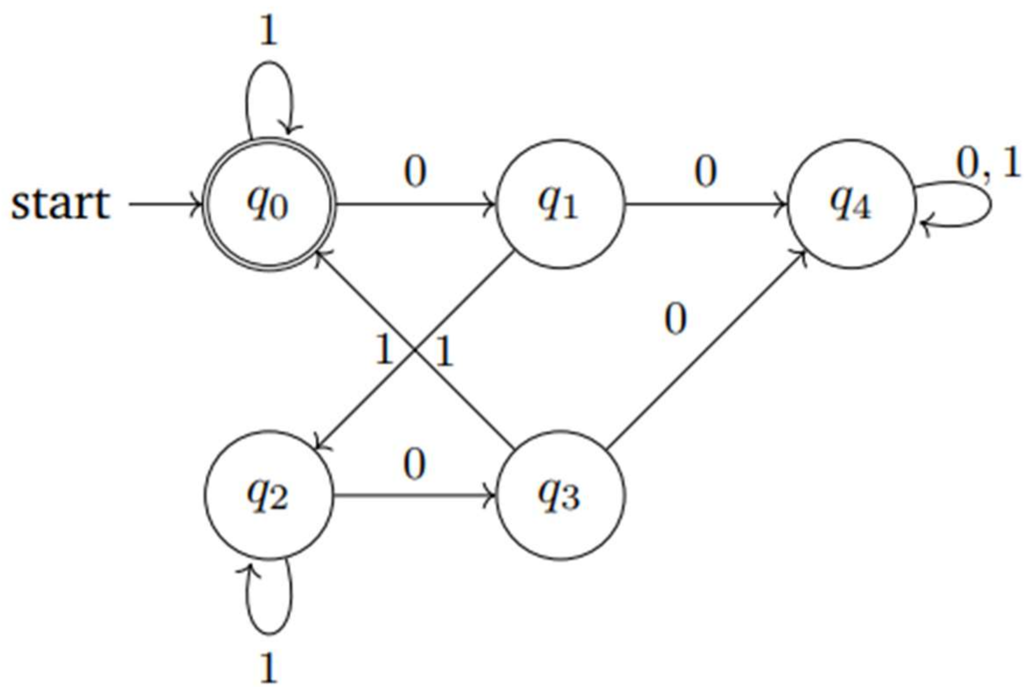


Problem 8 - Languages

(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.

Problem 8 - 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.



q0: even number of 0's, with final 0 followed by at least one 1

q1: odd number of 0's, with final 0 not yet followed by at least one 1

q2: odd number of 0's, with final 0 followed by at least one 1

q3: even number of 0's, with final 0 not yet followed by at least one 1

q4: garbage state where at least one 0 is not followed by at least one 1

That’s All, Folks!

Any questions?