Final Review Session

CSE 311 - Sp 2022

Anjali Agarwal
Alex Fang
Allie Pfleger

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

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

Warm-Up - Translations

(a) Every student is taking some course.

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

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

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

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

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

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

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

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

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

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

Every course taught by Robbie contains theory.

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

Warm-Up - 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.

Warm-up: Predicate Logic

Express each of these system specifications using predicates, quantifiers, and logical connectives. For some of these problems, more than one translation will be reasonable depending on your choice of predicates.

(a) Every user has access to an electronic mailbox

(b) The system mailbox can be accessed by everyone in the group if the file system is locked.

(c) The firewall is in a diagnostic state only if the proxy server is in a diagnostic state.

(d) At least one router is functioning normally if the throughput is between 100kbps and 500 kbps and the proxy server is not in diagnostic mode.

Warm-up: Predicate Logic Solutions

(a) Every user has access to an electronic mailbox.

Let the domain be users and mailboxes. Let User(x) be "x is a user", let Mailbox(y) be "y is a mailbox", and let Access(x, y) be "x has access to y".

$$\forall x\,(\mathrm{User}(x) \rightarrow (\exists y\,(\mathrm{Mailbox}(y) \land \mathrm{Access}(x, y))))$$

(b) The system mailbox can be accessed by everyone in the group if the file system is locked.

Warm-up: Predicate Logic Solutions

(a) Every user has access to an electronic mailbox.

Let the domain be users and mailboxes. Let User(x) be "x is a user", let Mailbox(y) be "y is a mailbox", and let Access(x, y) be "x has access to y".

$$\forall x\,(\mathrm{User}(x) \rightarrow (\exists y\,(\mathrm{Mailbox}(y) \land \mathrm{Access}(x, y))))$$

(b) The system mailbox can be accessed by everyone in the group if the file system is locked.

Solution 1: Let the domain be people in the group. Let CanAccessSM(x) be "x has access to the system mailbox". Let p be the proposition "the file system is locked."

$$p \rightarrow \forall x\,\mathrm{CanAccessSM}(x).$$

Solution2: Let the domain be people and mailboxes and use Access(x, y) as defined in the solution to part (a), and then also add InGroup(x) for "x is in the group", and let SystemMailBox be the name for the system mailbox.

$$\mathrm{FileSystemLocked} \rightarrow \forall x\,(\mathrm{InGroup}(x) \rightarrow \mathrm{Access}(x, \mathrm{SystemMailBox})).$$

Warm-up: Predicate Logic Solutions

(c) The firewall is in a diagnostic state only if the proxy server is in a diagnostic state.

Let the domain be all applications. Let Firewall(x) be "x is the firewall", and let ProxyServer(x) be "x is the proxy server." Let Diagnostic(x) be "x is in a diagnostic state".

$$\forall x\ \forall y\ ((\text{Firewall}(x)\ \land\ \text{Diagnostic}(x)) \rightarrow (\text{ProxyServer}(y) \rightarrow \text{Diagnostic}(y))$$

(d) At least one router is functioning normally if the throughput is between 100kbps and 500 kbps and the proxy server is not in diagnostic mode.

Warm-up: Predicate Logic Solutions

(c) The firewall is in a diagnostic state only if the proxy server is in a diagnostic state.

Let the domain be all applications. Let Firewall(x) be "x is the firewall", and let ProxyServer(x) be "x is the proxy server." Let Diagnostic(x) be "x is in a diagnostic state".

$$\forall x\ \forall y\ ((\text{Firewall}(x) \land \text{Diagnostic}(x)) \rightarrow (\text{ProxyServer}(y) \rightarrow \text{Diagnostic}(y))$$

(d) At least one router is functioning normally if the throughput is between 100kbps and 500 kbps and the proxy server is not in diagnostic mode.

Let the domain be all applications and routers. Let Router(x) be "x is a router", and let ProxyServer(x) be "x is the proxy server." Let Diagnostic(x) be "x is in a diagnostic state". Let p be "the throughput is between 100kbps and 500 kbps". Let Functioning(y) be "y is functioning normally".

$$p \land \forall x\ (\neg\, \text{ProxyServer}(x) \lor \neg\, \text{Diagnostic}(x))) \rightarrow \exists y\ (\text{Router}(y) \land \text{Functioning}(y))$$

Practice Final: 1. Regularly Irregular

Let $\Sigma = \{0, 1\}$. Prove that the language $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ is irregular.

Practice Final: 1. Regularly Irregular Solution

Let $\Sigma = \{0, 1\}$. Prove that the language $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ is irregular.

Suppose, for the sake of contradiction, that $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ 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.

Practice Final: 1. Regularly Irregular Solution

Let $\Sigma = \{0, 1\}$. Prove that the language $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ is irregular.

Suppose, for the sake of contradiction, that $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ 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.

Practice Final: 1. Regularly Irregular Solution

Let $\Sigma = \{0, 1\}$. Prove that the language $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ is irregular.

Suppose, for the sake of contradiction, that $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ 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.

Practice Final: 1. Regularly Irregular Solution

Let $\Sigma = \{0, 1\}$. Prove that the language $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ is irregular.

Suppose, for the sake of contradiction, that $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ 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 $z = 1^b$.

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.

Practice Final: 1. Regularly Irregular Solution

Let $\Sigma = \{0, 1\}$. Prove that the language $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ is irregular.

Suppose, for the sake of contradiction, that $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ 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 $z = 1^b$.

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 $xz = 0^a 1^b$, 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.

Practice Final: 1. Regularly Irregular Solution

Let $\Sigma = \{0, 1\}$. Prove that the language $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ is irregular.

Suppose, for the sake of contradiction, that $L = \{x \in \Sigma^* : \#_0(x) < \#_1(x)\}$ 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 $z = 1^b$.

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

Practice Final: 2. Recurrences, Recurrences

Define

$$T(n) = \begin{cases} n & \text{if } n = 0, 1 \\ 4T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n & \text{otherwise} \end{cases}$$

Prove that $T(n) \leq n^3$ for $n \geq 3$

Practice Final: 2. Recurrences, Recurrences Solution

Let P(n) be "$T(n) \le n^3$" for $n \ge 3$. We prove P(n) by strong induction on n.

Base Cases.

Induction Hypothesis.

Induction Step.

Conclusion.

Practice Final: 2. Recurrences, Recurrences Solution

Let P(n) be "$T(n) \leq n^3$" for $n \geq 3$. We prove P(n) by strong induction on n.

Base Cases. When n = 3:

When n = 4:

When n = 5:

Induction Hypothesis.

Induction Step.

Conclusion.

Practice Final: 2. Recurrences, Recurrences Solution

Let P(n) be "T(n) ≤ n^3" for n ≥ 3. We prove P(n) by strong induction on n.

Base Cases. When n = 3: $T(3) = 4T\left(\left\lfloor \frac{3}{2} \right\rfloor\right) + 3 = 4T\left(1\right) + 3 = 7 \leq 27 = 3^3$.

When n = 4: $T(4) = 4T\left(\left\lfloor \frac{4}{2} \right\rfloor\right) + 4 = 4T\left(2\right) + 4 = 28 \leq 64 = 4^3$.

When n = 5: $T(5) = 4T\left(\left\lfloor \frac{5}{2} \right\rfloor\right) + 5 = 4T\left(2\right) + 5 = 29 \leq 4^4$.

Induction Hypothesis.

Induction Step.

Conclusion.

Practice Final: 2. Recurrences, Recurrences Solution

Let P(n) be "T(n) ≤ n^3" for n ≥ 3. We prove P(n) by strong induction on n.

Base Cases. When n = 3: $T(3) = 4T\left(\left\lfloor \frac{3}{2} \right\rfloor\right) + 3 = 4T(1) + 3 = 7 \leq 27 = 3^3$.

When n = 4: $T(4) = 4T\left(\left\lfloor \frac{4}{2} \right\rfloor\right) + 4 = 4T(2) + 4 = 28 \leq 64 = 4^3$.

When n = 5: $T(5) = 4T\left(\left\lfloor \frac{5}{2} \right\rfloor\right) + 5 = 4T(2) + 5 = 29 \leq 4^4$.

Induction Hypothesis. Suppose P(3) ∧ P(4) ∧ · · · ∧ P(k) for some k ≥ 5.

Induction Step.

Conclusion.

Practice Final: 2. Recurrences, Recurrences Solution

Let P(n) be "T(n) ≤ n^3" for n ≥ 3. We prove P(n) by strong induction on n.

Base Cases. When n = 3: $T(3) = 4T\left(\left\lfloor \frac{3}{2} \right\rfloor\right) + 3 = 4T(1) + 3 = 7 \leq 27 = 3^3.$

When n = 4: $T(4) = 4T\left(\left\lfloor \frac{4}{2} \right\rfloor\right) + 4 = 4T(2) + 4 = 28 \leq 64 = 4^3.$

When n = 5: $T(5) = 4T\left(\left\lfloor \frac{5}{2} \right\rfloor\right) + 5 = 4T(2) + 5 = 29 \leq 4^4.$

Induction Hypothesis. Suppose P(3) ∧ P(4) ∧ · · · ∧ P(k) for some k ≥ 5.

Induction Step.

$$
\begin{aligned}
T(k+1) &= 4T\left(\left\lfloor \frac{k+1}{2} \right\rfloor\right) + k + 1, && \text{because } k+1 \geq 2.\\
&\leq 4\left(\left\lfloor \frac{k+1}{2} \right\rfloor\right)^3 + k + 1, && \text{by IH.}\\
&\leq 4\left(\frac{k+1}{2}\right)^3 + k + 1, && \text{by def of floor.}\\
&= 4\left(\frac{(k+1)^3}{2^3}\right) + k + 1, && \text{by algebra.}\\
&= \frac{(k+1)^3}{2} + k + 1, && \text{by algebra.}\\
&= \frac{(k+1)((k+1)^2 + 2)}{2}, && \text{by algebra.}\\
&\leq \frac{(k+1)((k+1)^2 + (k+1)^2)}{2}, && \text{because } (k+1)^2 \geq 2.\\
&= (k+1)^3, && \text{by algebra}
\end{aligned}
$$

Conclusion.

Practice Final: 2. Recurrences, Recurrences Solution

Let P(n) be "$T(n) \le n^3$" for $n \ge 3$. We prove P(n) by strong induction on n.

Base Cases. When n = 3: $T(3) = 4T\left(\left\lfloor \frac{3}{2} \right\rfloor\right) + 3 = 4T(1) + 3 = 7 \le 27 = 3^3$.

When n = 4: $T(4) = 4T\left(\left\lfloor \frac{4}{2} \right\rfloor\right) + 4 = 4T(2) + 4 = 28 \le 64 = 4^3$.

When n = 5: $T(5) = 4T\left(\left\lfloor \frac{5}{2} \right\rfloor\right) + 5 = 4T(2) + 5 = 29 \le 4^4$.

Induction Hypothesis. Suppose P(3) $\wedge$ P(4) $\wedge \cdots \wedge$ P(k) for some $k \ge 5$.

Induction Step.

$$\begin{aligned}
T(k+1) &= 4T\left(\left\lfloor \frac{k+1}{2} \right\rfloor\right) + k + 1, && \text{because } k+1 \ge 2. \\
&\le 4\left(\left\lfloor \frac{k+1}{2} \right\rfloor\right)^3 + k + 1, && \text{by IH.} \\
&\le 4\left(\frac{k+1}{2}\right)^3 + k + 1, && \text{by def of floor.} \\
&= 4\left(\frac{(k+1)^3}{2^3}\right) + k + 1, && \text{by algebra.} \\
&= \frac{(k+1)^3}{2} + k + 1, && \text{by algebra.} \\
&= \frac{(k+1)((k+1)^2 + 2)}{2}, && \text{by algebra.} \\
&\le \frac{(k+1)((k+1)^2 + (k+1)^2)}{2}, && \text{because } (k+1)^2 \ge 2. \\
&= (k+1)^3, && \text{by algebra}
\end{aligned}$$

Conclusion. Therefore, P(n) holds for all $n \ge 3$ by the principle of induction.

Practice Final: 3. All The Machines!

Let $\Sigma = \{0, 1, 2\}$. Consider $L = \{w \in \Sigma^*$: Every 1 in the string has at least one 0 before and after it, and the 0s need not be directly adjacent to the 1$\}$.

(a) Give a regular expression that represents L.

(b) Give a DFA that recognizes L.

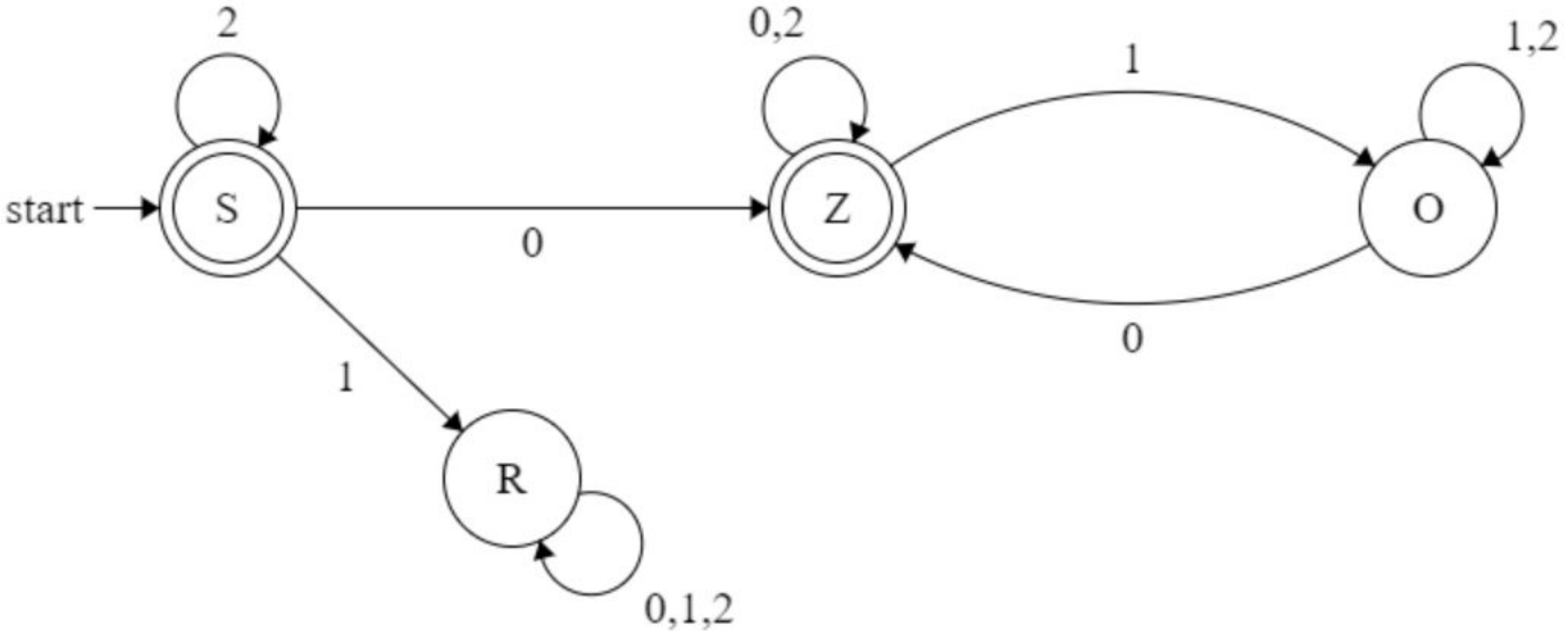
(c) Give a CFG that generates L.

Practice Final: 3. All The Machines! Solution

(b) Give a DFA that recognizes L.

Practice Final: 3. All The Machines! Solution

(b) Give a DFA that recognizes L.



Practice Final: 3. All The Machines! Solution

(a) Give a regular expression that represents L.

$(0 \cup 2)^* (0(0 \cup 1 \cup 2)^*0)^* (0 \cup 2)^*$

(c) Give a CFG that generates L.

Practice Final: 3. All The Machines! Solution

(a) Give a regular expression that represents L.

$(0 \cup 2)^* (0(0 \cup 1 \cup 2)^*0)^* (0 \cup 2)^*$

(c) Give a CFG that generates L.

$\boldsymbol{S} \rightarrow 0\boldsymbol{S} \mid 2\boldsymbol{S} \mid \boldsymbol{S}2 \mid 0\boldsymbol{T}0 \mid \varepsilon$
$\boldsymbol{T} \rightarrow 0\boldsymbol{T} \mid 1\boldsymbol{T} \mid 2\boldsymbol{T} \mid \varepsilon$

Practice Final: 4. Structural CFGs

Consider the following CFG: $\boldsymbol{S} \rightarrow \varepsilon \mid \boldsymbol{SS} \mid \boldsymbol{S}1 \mid \boldsymbol{S}01$. Another way of writing the recursive definition of this set, Q, is as follows:

- $\varepsilon \in Q$
- If $S \in Q$, then $S1 \in Q$ and $S01 \in Q$
- If $S, T \in Q$, then $ST \in Q$.

Prove, by structural induction that if $w \in Q$, then w has at least as many 1's as 0's

Practice Final: 4. Structural CFGs

We go by structural induction on w. Let $P(w)$ be "$\#0(w) \leq \#1(w)$" for $w \in \Sigma^*$.

Base Case. When $w = \varepsilon$, note that $\#0(w) = 0 = \#1(w)$. So, the claim is true.

Practice Final: 4. Structural CFGs

We go by structural induction on w. Let $P(w)$ be "$\#0(w) \le \#1(w)$" for $w \in \Sigma*$.

Base Case. When $w = \varepsilon$, note that $\#0(w) = 0 = \#1(w)$. So, the claim is true.

Induction Hypothesis. Suppose $P(w)$, $P(v)$ are true for some w, v generated by the grammar.

Induction Step 1. Note that $\#0(w1) = \#0(w) \le \#1(w) + 1 = \#1(w1)$ by IH, and $\#0(w01) = \#0(w) + 1 \le \#1(w) + 1 = \#1(w01)$ by IH.

Induction Step 2. Note that $\#0(wv) = \#0(w) + \#0(v) \le \#1(w) + \#1(v)$ by IH.

Since the claim is true for all recursive rules, the claim is true for all strings generated by the grammar.

Practice Final: 5. Tralse!

For each of the following answer True or False and give a short explanation of your answer.

(a) Any subset of a regular language is also regular.

(b) The set of programs that loop forever on at least one input is decidable.

(c) If $\mathbb{R} \subseteq A$ for some set A, then A is uncountable.

(d) If the domain of discourse is people, the logical statement
$\exists x\, (\mathrm{P}(x) \rightarrow \forall y\, (x \neq y \rightarrow \neg P(y))$
can be correctly translated as "There exists a unique person who has property P".

(e) $\exists x\, (\forall y\, \mathrm{P}(x, y)) \rightarrow \forall y\, (\exists x\, \mathrm{P}(x, y))$ is true regardless of what predicate P is.

Practice Final: 5. Tralse! Solution

(a) Any subset of a regular language is also regular.

Take an irregular language, say, $\{0^n1^n: n \geq 0\}$.

Is it a part of some regular language? $\{0^*1^*\}$

Practice Final: 5. Tralse! Solution

(b) The set of programs that loop forever on at least one input is decidable.

If this problem is decidable, then I can use its decider to decide the halting problem!

halts(P, I):

Let Q(input) = "ignore input and run *P* on *I*".

Return whether Q halts on at least one input.

Practice Final: 5. Tralse! Solution

(c) If $\mathbb{R} \subseteq A$ for some set A, then A is uncountable.

If A were countable, then we would have the following line of reasoning:

1. There's a surjection (onto relation) from N to A.
2. There's a surjection (onto relation) from A to R.
3. Combining the two, there's a surjection (onto relation) from N to R.

🤯

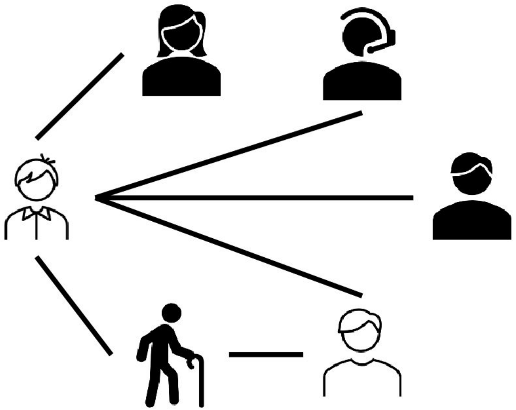
Practice Final: 5. Tralse! Solution

(d) If the domain of discourse is people, the logical statement
$\exists x\,(\mathrm{P}(x) \rightarrow \forall y\,(x \neq y \rightarrow \neg P(y))$
can be correctly translated as "There exists a unique person who has property P".

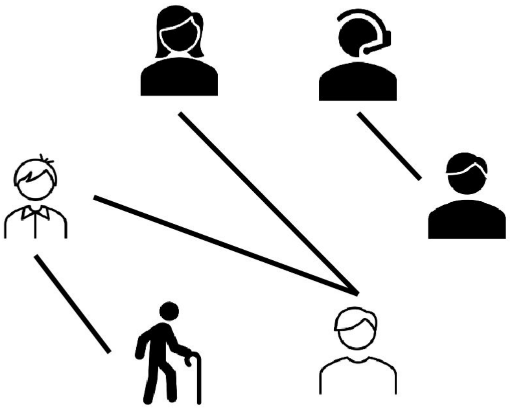
Practice Final: 5. Tralse! Solution

(e) $\exists x\,(\forall y\, P(x, y)) \rightarrow \forall y\,(\exists x\, P(x, y))$ is true regardless of what predicate P is.

Someone is friends with everyone.



Everyone is friends with someone.



Practice Final: BONUS Set Proof

$A = \{x : x \equiv k \pmod 4\}$, $B = \{x : x = 4r+k$ for some integer $r\}$. Prove $A = B$ for all integer k

Practice Final: BONUS Set Proof

A = {x : x ≡ k (mod 4)}, B = {x : x = 4r+k for some integer r}. Prove A = B for all integer k

Let k be an arbitrary integer.

First we show that $A \subseteq B$. Let $x \in A$ be arbitrary. Then by definition of A, $x \equiv k \pmod{4}$. Then by definition of mod, $4 \mid (x - k)$. Then by definition of divides, there exists an integer r such that $x - k = 4r$. Then $x = 4r + k$. So $x \in B$. Since x was arbitrary, $A \subseteq B$.

Now we show that $B \subseteq A$. Let $x \in B$ be arbitrary. Then by definition of B, $x = 4r + k$ for some integer r. So $x - k = 4r$. Then by definition of mod, $4 \mid (x - k)$. Then by definition of mod, $x \equiv k \pmod{4}$. So $x \in A$. Since x was arbitrary, $B \subseteq A$.

Then $A \subseteq B$ and $B \subseteq A$, so $A = B$. Since k was arbitrary, $A = B$ for all integers k.

Practice Final: 6. Relationships!

(b) Let S = $\{(X, Y) : X, Y \in \mathscr{P}(\mathbb{N}) \land X \subseteq Y\}$.
Recall that R is antisymmetric iff $((a, b) \in R \land a \neq b) \rightarrow (b, a) \notin$ R.
Prove that S is antisymmetric.

Practice Final: 6. Relationships! Solution

(b) Let $S = \{(X, Y) : X, Y \in \mathscr{P}(\mathbb{N}) \wedge X \subseteq Y\}$.
Recall that R is antisymmetric iff $((a, b) \in R \wedge a \neq b) \rightarrow (b, a) \notin$ R.
Prove that S is antisymmetric.

Let $(A, B) \in S$ be arbitrary such that $A \neq B$. Then by definition of S, $A, B \in P(\mathbb{N})$ and $A \subseteq B$. Since $A \neq B$, there is an element $x \in B$ such that $x \notin A$. Then by definition of subset, B is not a subset of A. Therefore $(B, A) \notin S$. So S is antisymmetric.

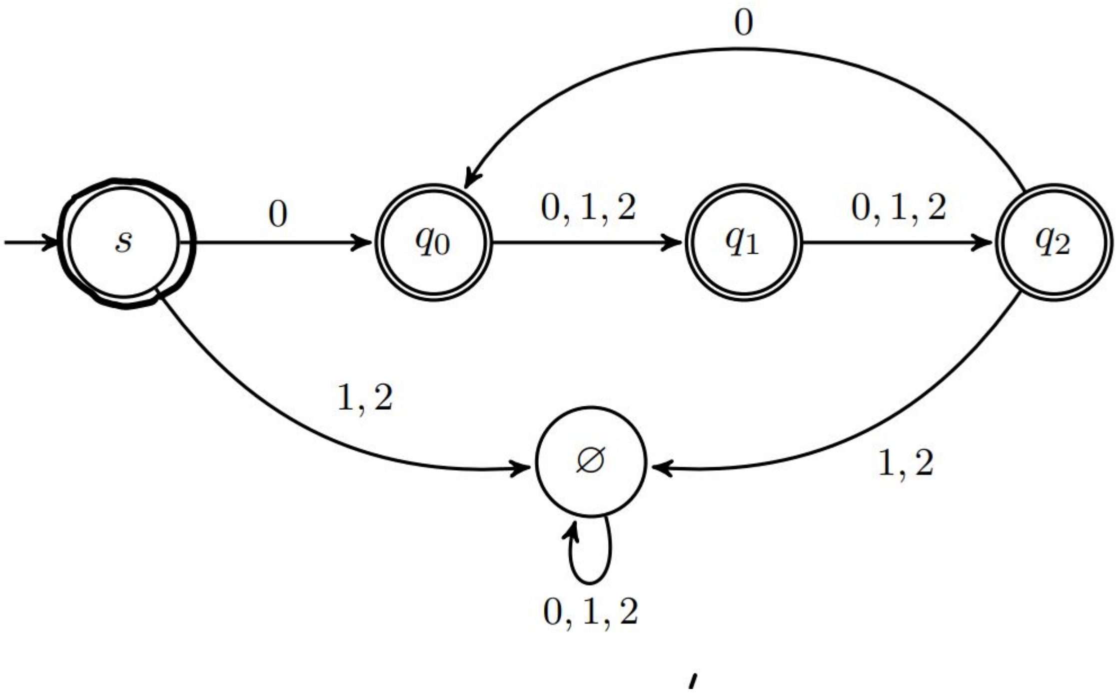
Practice Final: 8. Modern DFAs

Let $\Sigma = \{0, 1, 2\}$.

Construct a DFA that recognizes exactly strings with a 0 in all positions i where $i \% 3 = 0$.

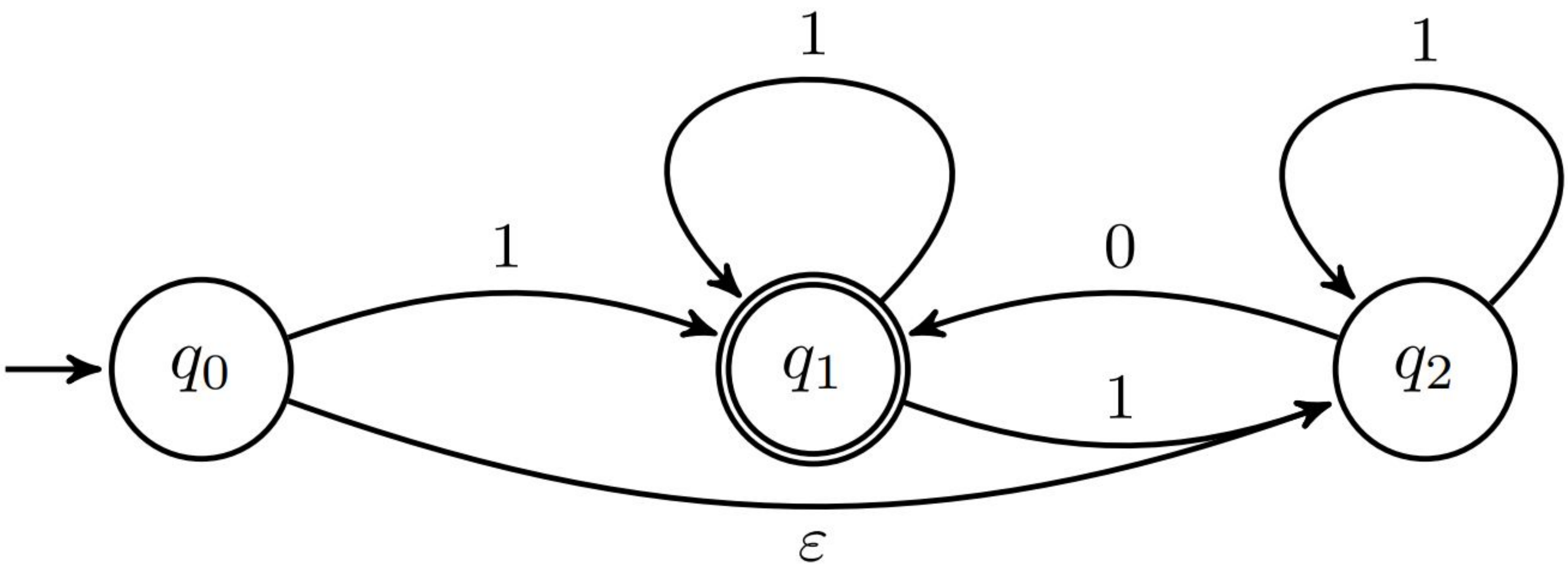
Practice Final: 8. Modern DFAs Solution

Let Σ = {0, 1, 2}. Construct a DFA that recognizes exactly strings with a 0 in all positions *i* where *i*%3 = 0.



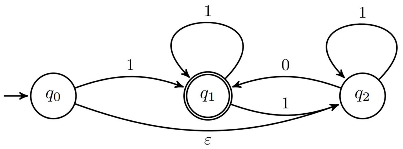
Practice Final: 7. Construction Paper!

Convert the following NFA into a DFA using the algorithm from lecture.



Practice Final: 7. Construction Paper! Solution

Convert the following NFA into a DFA using the algorithm from lecture.



That’s All, Folks!

Any questions?