COMPSCI 250 Final Exam, Fall 2019 - Solutions

Question 1 (30):

Cardie and Duncan, along with several other dogs, recently auditioned for the part of Toto in a community theatre production of *The Wizard of Oz.* The casting committee ranked each dog for three attributes: Appearance, Experience, and Temperament. The predicate $RA(x, y, z)$ means "dog x ranked above dog y for attribute z".

For each attribute z, the binary relation RA_z on the dogs, defined by $RA_z(x, y) \leftrightarrow RA(x, y, z)$, is a strict total order, meaning that it is antireflexive, antisymmetric, transitive, and has no incomparable elements. The unary predicate $GP(x)$ means "dog x got the part".

(a) Translate the following statements to predicate logic or English, as appropriate:

Statement I: Duncan ranked above all other dogs for Appearance, Maeve ranked above all other dogs for Experience, and Cardie ranked above all other dogs for Temperament.

$(\forall x : (x \neq d) \rightarrow RA(d, x, a)) \wedge (\forall x : (x \neq m) \rightarrow RA(m, x, e)) \wedge \forall x : (x \neq c) \rightarrow RA(c, x, t)$.
Using a single quantifier is equivalent, but this is the way the sentence is worded.

Statement II: $\forall x : GP(x) \leftrightarrow \forall y : y \neq x \rightarrow \exists a : \exists b : (a \neq b) \wedge RA(x, y, a) \wedge RA(x, y, b)$.

A dog got the part if and only for every other dog there are two different attributes on which the former dog ranks higher.

It is important to state unambiguously that the two attributes depend on the dog compared against. The wording "for every other dog there are two attributes" mirrors the quantifier order. "Ranks above any other dog on two attributes" is less clear. Many confused this in proofs with "there are two attributes on which it ranks higher than any other dog", which is false (and contradicts S I).

Statement III: Maeve is the one and only dog who both ranked ahead of Duncan for Temperament and ranked ahead of Cardie for Appearance.

$\forall x : (x = m) \leftrightarrow (RA(x, d, t) \wedge RA(x, c, a))$

In the following proofs, make clear what proof rules you use in your reasoning.

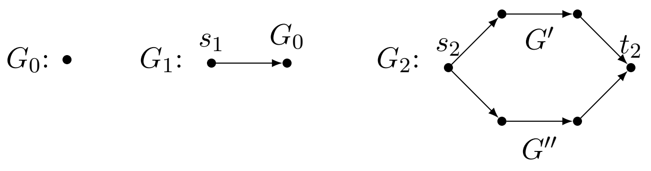
(b) Assuming Statements I, II, and III, prove it can't be that two different dogs both got the part.

Answer: The simplest answer only uses S II. If $GP(x)$ and $GP(y)$ are both true, by Specification and Instantiation on Statement II we get $RA(x, y, r)$ and $RA(x, y, s)$ for two distinct attributes r and s, and also $RA(y, x, u)$ and $RA(y, x, v)$ for two distinct attributes u and v. But there are only three attributes, so the sets $\{r, s\}$ and $\{u, v\}$ must have a common element w. We can't have both $RA(x, y, w)$ and $R(y, x, w)$ since RA_w is antisymmetric.

(c) Prove that Maeve got the part.

Answer: Suppose that Maeve did not get the part. By S II, there exists a dog y who ranked above Maeve on two of the three attributes. Neither of these can be e by S I, so they must be a and t. But if y ranks above m with respect to both a and t, then by transitivity $RA(y, d, t)$ and $RA(y, c, a)$ are both true, contradicting S III.

Question 2 (30): Define the directed graph G_n ($n \geq 0$) as follows. G_0 has one node and no edges. If n is odd, G_n is obtained from G_{n-1} by adding an extra node s_n, with an edge to each node of G_{n-1} that has no incoming edges. If $n > 0$ is even, G_n is obtained by taking two copies of $G_{n/2}$, called G' and G'', and two extra nodes s_n and t_n, with edges from s_n to all nodes of G' and G'' with no incoming edges, and edges to t_n from all nodes of G' and G'' with no outgoing edges.



a (5) Prove by induction a theorem about the number of nodes with no incoming edges and the number of nodes with no outgoing edges in G_n.

We prove by strong induction for any n that G_n has exactly one node with no incoming edges and one node with no outgoing edges. The statement is true for $n = 0$. Let $n > 0$ be arbitrary, and assume the statement is true for all $k < n$. We prove it for G_n, by cases. If n is odd, G_{n-1} has a single node u with indegree 0 and a single node v with outdegree zero, by the IH. The new node s_n has an edge to u, therefore u no longer has indegree 0, but s_n does. Node v is unchanged, so the goal holds. If n is even, the new node s_n has edges to all nodes that had indegree 0, so it becomes the only node with this property. Likewise, t_n has edges from all nodes with outdegree 0, and becomes the only node with this property, so the goal again holds.

b (10) Write and justify a recurrence for the number V_n of nodes in G_n.
Find and prove by induction a formula for V_n when $n = 2^k$, $k \geq 0$.

We have $V_0 = 1$. For odd n, we add a single new node, so $V_n = V_{n-1} + 1$. For even $n > 0$, we have two copies of $G_{n/2}$, and two additional nodes, so $V_n = 2V_{n/2} + 2$. We get $V_2 = 6$, $V_4 = 14$, and conjecture $V_{2^k} = 2^{k+2} - 2$ for $k > 0$. We can prove this by induction on $k > 0$.

The base case holds, for $k = 1$ we have $V_2 = 6 = 2^{1+2} - 2$. Assume the statement holds for arbitrary $k > 0$. Then, $V_{2^{k+1}} = 2V_{2^k} + 2 = 2(2^{k+2}) - 2) + 2 = 2^{k+1+2} - 2$, so the relation holds.

c (5) Prove by induction that $V_n < 4n$ for $n > 0$.

We have seen that $V_n = 4n - 2$ for $n = 2^k$ ($k > 0$). We will prove by strong indu ction $V_n \leq 4n - 2$, which implies $V_n < 4n$. For the base case $n = 1$, the statement holds: $V_1 = 2 \leq 4 \cdot 1 - 2$. We take an arbitrary $n > 1$, assume $V_k \leq 4k - 2$ for all $0 < k < n$, and prove $V_n \leq 4n - 2$. If n is even, $n \geq 2$, so the IH holds for $n/2 > 0$, and $V_{n/2} \leq 4(n/2) - 2 = 2n - 2$. We have $V_n = 2V_{n/2} + 2 \leq 2(2n - 1) + 2 = 4n - 2$. If $n > 1$ is odd, then $n \geq 3$, thus the IH holds for $n - 1 > 0$. We have $V_n = V_{n-1} + 1 \leq 4(n-1) - 2 + 1 = 4n - 5 \leq 4n - 2$, which completes the proof.

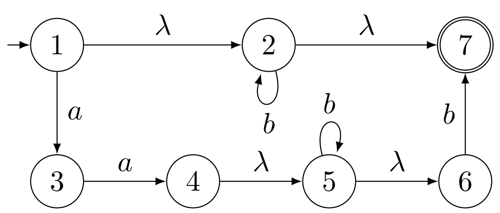
d (10) Write and prove by induction a formula for the number V_n of nodes when $n = 2^k - 1$, $k > 0$. (Hint: look for a formula of the form $V_{2^k - 1} = a \cdot 2^k + b$).

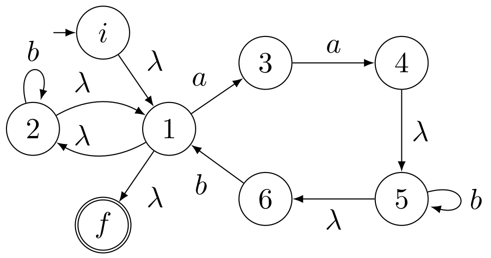
Following the hint, we take $V_{2^k-1} = a \cdot 2^k + b$. For $k = 1$, we want $V_1 = 2 = a \cdot 2^1 + b$, so $b = 2 - 2a$. Take as IH $V_{2^k-1} = a \cdot 2^k + 2 - 2a$ for $k > 0$, and use the odd and then the even rule to compute $V_{2^{k+1}-1} = V_{2^{k+1}-2} + 1 = 2V_{2^k-1} + 2 + 1 = 2(a \cdot 2^k + 2 - 2a) + 3 = a \cdot 2^{k+1} + 2 - 2a + (5 - 2a)$. Our inductive goal holds if $5 - 2a = 0$, so $a = 5/2$. Thus, we have proved $V_{2^k-1} = 5 \cdot 2^{k-1} - 3$ for $k > 0$.

Question 3 (30): Do the following constructions for the language $L = (b^* + aab^*b)^*$.

a (5) Find a λ-NFA N whose language is L.
For full credit, use only constructions shown in the lecture.

For $b^* + aab^*b$ we get the construction:

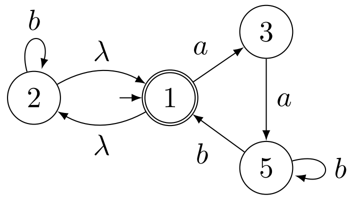


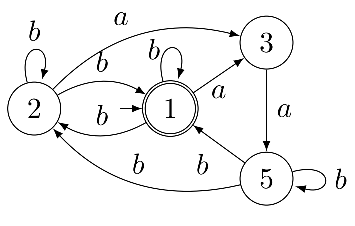


For the outer Kleene star, we merge state 7 with 1, and add the extra initial and final state.

b (10) Build an ordinary NFA N' such that $L(N') = L(N)$, using the construction from the book. You may make simplifications of N before doing this, as long as they do not change the language of the machine and you clearly argue that what you are doing is correct.

We can simplify the automaton N by removing states i and f (which only append λ to any string, thus don't change the language) and making 1 both initial and final state. We can also eliminate state 4, getting an a-transition from 3 to 5, and state 6, getting a b-transition from 5 to 1.





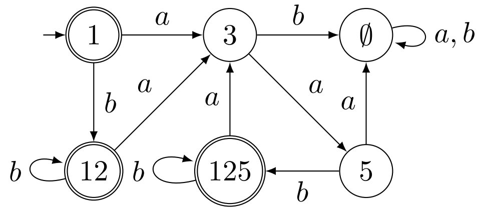
In the transitive closure, we get λ-moves from any of the states 1, 2 to any of these states. Then: $2 \xrightarrow{b} 2$ gives us four transitions, $1, 2 \xrightarrow{b} 1, 2$; $1 \xrightarrow{a} 3$ gives us two transitions $1, 2 \xrightarrow{a} 3$; $5 \xrightarrow{b} 1$ gives us two transitions $5 \xrightarrow{b} 1, 2$. The resulting automaton is shown on the right.

In our initial simplification process, we could also apply the state elimination construction for 2, getting a self-loop $1 \xrightarrow{b} 1$, which would directly give us a three-state NFA with no λ-transitions. This was acceptable as long as you argued it's correct (e.g., by the state elimination rules).

c (5) Using the Subset Construction on N', find a DFA D such that $L(D) = L(N')$.

We construct the transition table:

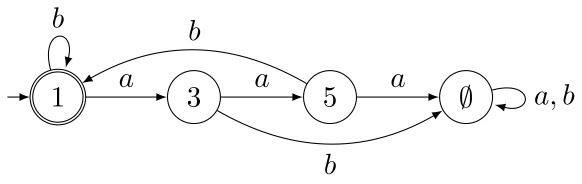
	a	b
1	3	12
3	5	$\emptyset$
12	3	12
5	$\emptyset$	125
125	3	125



States $\{1\}$, $\{1, 2\}$, and $\{1, 2, 5\}$ are final, they contain the final state 1 of N'; state $\emptyset$ is a dead state.

d (5) Find a minimal DFA D' with $L(D') = L(D)$. You may use the minimization construction, or prove directly that your D is already minimal.

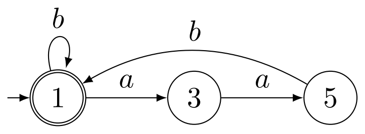
We group states 1, 12, and 125 since they are final, transition on a to 3 and on b to the same group (1, 12, 125), so they are equivalent. Merging them gives us the following automaton:



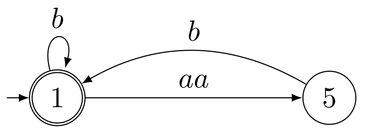
This automaton is minimal. State 1 is final, thus not equivalent to any non-final state. States 3 (reached by a) and 5 (reached by aa) are distinguished by ab (accepted in 1, rejected in $\emptyset$). State $\emptyset$ is not equivalent to either 3 or 5, since it has no paths to a final state, while 3 and 5 have.

e (5) Using the state elimination construction on D or D', find a regular expression for $L(N)$. (The construction may produce a simpler or more complicated expression than the initial one).

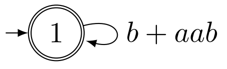
Since the automaton has a unique final state, we could directly do the construction without introducing a special initial and a special final state; we need all paths from 1 to 1. We first eliminate the death state, without adding anything, as there are no edges out of it. We next eliminate 3, as there is only one path through it. Finally, we eliminate 5, which creates a transition labeled $b+aab$ from 1 to 1. The resulting regular expression is $(b+aab)^*$, which we could prove to be equivalent with the original one.



.



.



Question 4 (30p)

The following are fifteen true/false questions, with no explanation needed or wanted, no partial credit for wrong answers, and no penalty for guessing.

(a) The statement $(\forall x : P(x)) \vee \exists y : P(y) \to Q(y)$ is a tautology.
TRUE, if P is false for an element, then the second part holds

(b) The number of antireflexive relations on a finite set with $n > 2$ elements is larger than the number of reflexive relations.
FALSE, they are equal

(c) If the system $x \equiv a \pmod{m}$ and $x \equiv b \pmod{n}$ has a unique natural solution $< mn$, then m and n are relatively prime.
TRUE, otherwise there are either zero or multiple solutions

(d) The prefix and postfix form of an arithmetic expression cannot both have consecutive operators.
FALSE, consider $a * b + c * d$. prefix: $+ * ab * cd$, postfix: $ab * cd * +$

(e) Iterative deepening search will always find the shortest path to the goal (if it exists), even when not using a closed set.
TRUE, it tries all paths in length order.

(f) The regular expression $(ab + ba)^*$ describes the even-length strings in which a and b alternate.
FALSE, it contains abba

(g) Let $\Sigma = \{a\}$, and $L = \{a^n : n \text{ is even or prime}\}$. Then L is not regular.
TRUE, if it were, subtracting the regular language of evens would make primes regular (it isn't).

(h) If $p, q > 0$, the language $a^* \setminus (a^p + a^q)^*$ is finite if and only if p and q are relatively prime.
TRUE, $a^p + a^q$ has all lengths $> pq - p - q$, or only those divisible by $gcd(p, q)$

(i) Let $k > 1$ and L_k be the language of strings over $\{a, b\}$ with no k consecutive equal letters. Then the L_k-equivalence relation has $2k - 1$ equivalence classes.
FALSE, one for λ, $2(k - 1)$ for 1 to $k - 1$ consecutive a's and b's, and one for rejected strings.

(j) Let p and q be two states in a DFA such that $\delta(p, x) = \delta(q, x)$ for any letter $x \in \Sigma$.
Then it is possible that p and q are not equivalent.
TRUE, if one state is accepting and the other one not

(k) A minimal DFAs with $n > 1$ states may have any number $0 \le k \le n$ of final states.
FALSE, if all $n > 1$ states are accepting, it's no longer minimal.

(l) If every left move of a 2WDFA goes to a state which has only right moves, then it cannot hang.
FALSE, it can hang if it moves left from the initial state

(m) If three of the languages $X \cap Y$, $\bar{X} \cap Y$, $X \cap \bar{Y}$, $\bar{X} \cap \bar{Y}$ are decidable, then X and Y are decidable.
TRUE, we use union to get X or $\bar{X}$ and Y or $\bar{Y}$.

(n) The set of Turing machines that write to at most the first 100 tape cells is Turing-decidable.
TRUE, they have a bounded number of configurations, so we can simulate them and see if they halt or revisit a configuration

(o) The set of Turing machines that use an unbounded amount of tape is Turing-recognizable.
FALSE. The set of Turing machines that use only a bounded amount of tape is recognizable (such a TM will halt or revisit a configuration and loop). We can recognize invalid TM encodings. Their union is the complement of the examined language. If both were recognizable, we could decide if a Turing machine uses unbounded tape, and then we could decide halting.