
On the cover page of your assignment, you must list everyone with whom you discussed this assignment, and which problems you discussed with each person. You must also write **and sign** the following statement: “I have read and understood the policy on collaboration on homework assignments stated in the Course Information handout.” Without these, your homework will not be marked.

1. Consider the sentence F , which is $\forall x(\exists yB(x, y) \rightarrow A(x))$. [10]

One of three sentences below is logically equivalent to F , and two of them are not:

- $\forall x\exists y(B(x, y) \rightarrow A(x))$,
- $\forall x\forall y(\neg B(y, x) \vee A(y))$,
- $\forall x(\neg A(x) \rightarrow \exists y\neg B(x, y))$.

For each of the two sentences that are *not* equivalent to F , prove that it is not equivalent to F by showing a structure where that sentence and F have different truth values. (You will receive no credit if you choose the wrong sentences.)

2. Let \mathcal{L} be a first-order logical language with an infinite supply of constant symbols, a_1, a_2, \dots , and two predicate symbols, P (of arity 1) and R (of arity 2). For any $n \in \mathbb{N}$, $n \geq 2$, let T_n be the set of sentences of \mathcal{L} defined by [10]

$$T_n = \{ P(a_1), \neg P(a_n), R(a_1, a_2), R(a_2, a_3), \dots, R(a_{n-1}, a_n) \}.$$

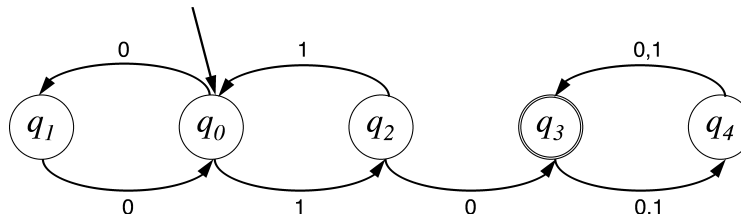
Prove by induction that for all $n \in \mathbb{N}$, $n \geq 2$, T_n logically implies

$$\exists x\exists y(P(x) \wedge (R(x, y) \wedge \neg P(y))).$$

Hint: Use the argument about hockey fans in Assignment 1 for inspiration.

3. In this question, we consider to what extent a DFSA is able to compare two binary numbers to see if one is larger. Of course, a DFSA cannot *return* a truth value, but we can construct one that *recognizes* pairs of binary numbers a and b (presented in some way) such that $a > b$.

Let M be the DFSA over the alphabet $\Sigma = \{0, 1\}$ described by the diagram below:



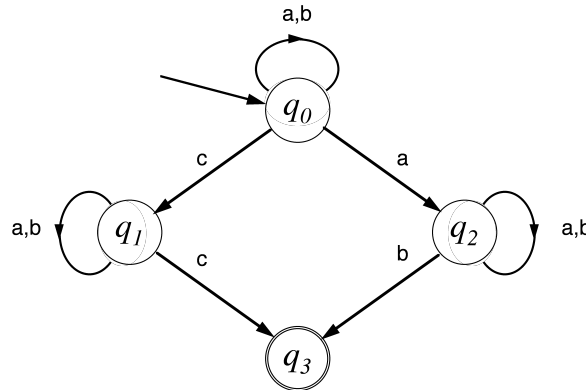
This automaton compares two binary numbers of the same length presented interleaved, most significant bit first. For example, because $110 > 100$, it accepts the string 111000. Because $100 \not> 110$, it rejects the string 110100.

- (a) Prove by induction that [10]

$$\mathcal{L}(M) = \{a_n b_n \cdots a_1 b_1 : n \in \mathbb{N}, n \geq 1, a_i \in \Sigma, b_i \in \Sigma, \text{ and } a > b \text{ where } a = a_n \cdots a_1 \text{ and } b = b_n \cdots b_1\}.$$

You may use without proof the fact that $x \in \mathcal{L}(M)$ iff $x = u \cdot 10 \cdot v$ where $|v|$ is even and where u is a sequence of matching pairs, 00 or 11. In other words, $a > b$ iff a and b agree on bits until some point i where $a_i = 1$ and $b_i = 0$. *Hint:* In thinking about invariants for the 6 states of M (including the dead state), use this formulation in terms of u and v .

- (b) Write a regular expression R such that $\mathcal{L}(R) = \mathcal{L}(M)$. [5]
- (c) Show the diagram of a DFSA that compares two binary numbers of the same non-zero length presented interleaved, but this time *least* significant bit first. [10]
 For example, because $110 > 100$, it should accept the string 001011. Because $100 \not> 110$, it should reject the string 000111. It should of course also reject any string that is of odd length. It is again possible to do this with 6 states.
- (d) Explain informally why there can be no DFSA that compares two binary numbers presented non-interleaved, one after the other. [5]
4. Use the subset construction to convert the NFSA over the alphabet $\{a, b, c\}$ drawn below to a DFSA that recognizes the same language. [10]



5. Let R be the following regular expression: $(0 + [1(01^*0)^*1])^*$. [10]
 (We have used square brackets for readability.)
- (a) Prove that $1101011101 \in \mathcal{L}(R)$.
- (b) Prove that $1011011 \notin \mathcal{L}(R)$.