Homework 6

Joe Baker, Brett Schreiber, Brian Knotten

January 25, 2018

8

a

Since $L(N) = \overline{L(M)}$, then N can be constructed as a modified M where M's non accepting states are accepting and vice versa.

b

$P = N \cap M$ where the "$\cap$" symbol represents the closure under the DFAs so that $L(P) = \{x | x \in L(N) \wedge x \in L(M)\}$.

c

Every accepting state of P has a transition on the final character of the encoding of w. Since Q does not have an input for w, consider Q as a modification of P which contains no states involved with the transitions for w. The accepting states of Q are all states with any transition over a character in w from the equivalent state in P which leads to an accepting state.

d

Every accepting state of Q has a transition on the final character of the encoding of z. Since R does not have an input for z, consider R as a modification of Q which contains no states involved with the transitions for z. The accepting states of R are all states with transitions over a character in z from the equivalent state in Q that lead to an accepting state.

e

Every accepting state of S has a transition on the final character of the encoding of y. Since R does not have an input for y, consider S as a modification of R which contains no states involved with the transitions for y. The accepting states of S are all states with transitions over a character in y from the equivalent state in R that lead to an accepting state.

f

Treat the DFA S as a directed graph such that each vertex is a state and each directed edge between some vertices q_a and q_b represents the transition $\delta(q_a, c) \rightarrow q_b$ for some character c. Compute a breadth-first search on S starting from the start state q_0 and accept if an accepting state is ever encountered. If the breadth-first search finishes without finding any accept states, then reject.

This algorithm works because if there exists an accepting state in the search's resultant tree, then there exists a path on S from the start state q_0 to an accepting state. Each edge in the path represents a character c, and the path taken to the accepting state is a string $c_1c_2c_3...c_n$. But if there is no path to an accepting state in S, then there is no string of this form, and so $L(S) = \emptyset$

g

First, parse the arithmetic statements into parts based on their order of operations, and then apply these rules to each statement based on the order of operations.

For each arithmetic statement of the form $(a + b = c)$ or $(a + b = c + d)$, create a DFA which only accepts when the equation is true.

For each negation upon any arithmetic statement, perform the complement closure on the DFA D corresponding to the term being negated. That is, switch all the accepting and non-accepting states.

For each and symbol in the statement, perform the intersection closure on the corresponding DFAs to the left and right of the and symbol.

Now, from right to left, conditionally transform the DFA depending on whether the quantifier is existential or universal using the above strategies.

The final result is a DFA S which a TM can then perform a BFS on to determine if $L(S) = \emptyset$. If so, then the first order sentence is never satisfiable. Otherwise, it will return a path which can be interpereted as a valid encoded variable assignment which satisfies the first order sentence.