CSE 311 - HW 8
Eric Boris
Tim Coulter, Alyssa Cote
December 2019

1 State of the Art (Submitted Online)

2 Regular as Clockwork (Submitted Online)

3 State's Evidence (Submitted Online)

4 Enemy of the State (Submitted Online)

5 Not Again

5(a)

1. Let P(A) be the claim that T(neg_p(A)) $\equiv$ T(A) or T(neg_p(A)) $\equiv \neg$ T(A). Prove P(A) for all A $\in$ Prop by structural induction.
2. Base Case: Let p, q be arbitrary members of $\mathcal{A}$. Then P(Atomic(q)) says when p = q that T(neg_p(Atomic(q))) = $\neg$ T((Atomic(q))) because

$$\begin{aligned} \text{T}(\text{neg}_p(\text{Atomic}(q))) &= \text{T}(\text{NOT}(\text{Atomic}(q))) && \text{Definition of neg}_p \\ &= \neg\text{T}(\text{Atomic}(q)) && \text{Definition of T} \end{aligned}$$

   or when p $\neq$ q that T(neg_p(Atomic(q))) = T((Atomic(q))) by the definition of neg_p.
3. Inductive Hypothesis: Assume that P(A) and P(B) hold for some arbitrary A,B $\in$ Prop.
4. Show P(NOT(A)) and P(XOR(A)) as follows:
   Case T(neg_p(NOT(A)))

$$\begin{aligned} T(\text{neg}_p(\text{NOT}(A))) &= T(\text{NOT}(\text{neg}_p(A))) && \text{Definition of ngz} \\ &= \neg T(\text{neg}_p(A)) && \text{Definition of T} \\ &= \neg T(A) && \text{Inductive Hypothesis} \\ &= T(\text{NOT}(A)) && \text{Definition of T} \end{aligned}$$

$$\begin{aligned} T(\text{neg}_p(\text{NOT}(A))) &= T(\text{NOT}(\text{neg}_p(A))) && \text{Definition of ngz} \\ &= \neg T(\text{neg}_p(A)) && \text{Definition of T} \\ &= \neg\neg T(A) && \text{Inductive Hypothesis} \\ &= \neg T(\text{NOT}(A)) && \text{Definition of T} \end{aligned}$$

   Case T(neg_p(XOR(A, B)))

$$\begin{aligned} T(\text{neg}_p(\text{XOR}(A, B))) &= T(\text{XOR}(\text{neg}_p(A), \text{neg}_p(B))) && \text{Definition of ngz} \\ &= (T(\text{neg}_p(A))) \oplus (T(\text{neg}_p(B))) && \text{Definition of T} \\ &= T(A) \oplus T(B) && \text{Inductive Hypothesis} \\ &= T(\text{XOR}(A, B)) && \text{Definition of T} \end{aligned}$$

$$
\begin{aligned}
T(\text{neg}_p(\text{XOR}(A,B))) &= T(\text{XOR}(\text{neg}_p(A), \text{neg}_p(B))) && \text{Definition of ngz} \\
&= (T(\text{neg}_p(A))) \oplus (T(\text{neg}_p(B))) && \text{Definition of T} \\
&= \neg T(A) \oplus T(B) && \text{Inductive Hypothesis} \\
&= \neg T(\text{XOR}(A,B)) && \text{Definition of T}
\end{aligned}
$$

$$
\begin{aligned}
T(\text{neg}_p(\text{XOR}(A,B))) &= T(\text{XOR}(\text{neg}_p(A), \text{neg}_p(B))) && \text{Definition of ngz} \\
&= (T(\text{neg}_p(A))) \oplus (T(\text{neg}_p(B))) && \text{Definition of T} \\
&= T(A) \oplus \neg T(B) && \text{Inductive Hypothesis} \\
&= \neg T(\text{XOR}(A,B)) && \text{Definition of T}
\end{aligned}
$$

$$
\begin{aligned}
T(\text{neg}_p(\text{XOR}(A,B))) &= T(\text{XOR}(\text{neg}_p(A), \text{neg}_p(B))) && \text{Definition of ngz} \\
&= (T(\text{neg}_p(A))) \oplus (T(\text{neg}_p(B))) && \text{Definition of T} \\
&= \neg T(A) \oplus \neg T(B) && \text{Inductive Hypothesis} \\
&= T(A) \oplus T(B) && \text{Definition of } \oplus \\
&= T(\text{XOR}(A,B)) && \text{Definition of T}
\end{aligned}
$$

5. Thus, P(A) holds for all parse trees A $\in$ Prop by structural induction. $\square$

5(b)

We've shown that with combinations of NOT and XOR T(neg$_p$(A)) can only generate T(A) or $\neg$T(A). So, we know that when T(neg$_p$(A)) $\equiv$ T(A) their truth table columns will be the same or when T(neg$_p$(A)) $\equiv \neg$T(A) their truth table columns will be the same.

This is in contrast to combinations of NOT and OR which can generate a truth table with any combination of truth values, NOT and XOR are much more limited in only being able to create the same truth values or the negation thereof.

5(c)

p	q	¬p	¬q	p∧q	¬p⊕q	p⊕¬q	p⊕q	¬p⊕¬q
T	T	F	F	F	F	F	T	T
T	F	F	T	T	T	T	F	F
F	T	T	F	T	T	T	F	F
F	F	T	T	T	F	F	T	T

As the truth table above shows, there is no pairing of p and q with $\oplus$ that matches the column with p, q, and $\wedge$. Therefore, not all propositional statements can be represented with $\neg$ and $\oplus$.

6 Just Irregular Guy

6(a)

1. Suppose for contradiction that some DFA, M, recognizes L = $\{0^x 1^m 0^y | x, m, y > 1 \text{ and } x \equiv y (\text{mod } m)\}$.

2. Let $S = \{0^x : x \geq 0\}$.

3. Since S is infinite and M has finitely many states, there must be two strings, 0^a and 0^{a+1} in S for $a \equiv a(\text{mod } m)$ and $m > 1$ that end up at the same state of M.

4. Consider appending $1^m 0^a$ to each of the two strings.

5. Note that $0^a 1^m 0^a \in L$ since $a \equiv a(\text{mod } m)$ but $0^{a+1} 1^m 0^a \notin L$ since $a + 1 \not\equiv a(\text{mod } m)$ when $m > 1$. Since 0^a and 0^{a+1} both end up at the same state of M, and we appended the same string $1^m 0^a$, both 0^a and 0^{a+1} end at the same state q of M. Since $0^a 1^m 0^a \in L$ and $0^{a+1} 1^m 0^a \notin L$, M does not recognize L.

6. Thus, no DFA recognizes L.

6(b)

1. Suppose for contradiction that some DFA, M, recognizes L = {Unicode strings that are syntatically valid JSON}.

2. Let $S = \{\{^x : x \geq 0\}$.

3. Since S is infinite and M has finitely many states, there must be two strings, $\{^a$ and $\{^b$ in S for $a \neq b$ that end up at the same state of M.

4. Consider appending $\}^a$ to each of the two strings.

5. Note that $\{^a\}^a \in L$ since $a = a$ but $\{^b\}^a \notin L$ since $a \neq b$. Since $\{^a$ and $\{^b$ both end up at the same state of M, and we appended the same string $\}^a$, both $\{^a$ and $\{^b$ end at the same state q of M. Since $\{^a\}^a \in L$ and $\{^b\}^a \notin L$, M does not recognize L.

6. Thus, no DFA recognizes L.