CS 321 Fall 2014 HW2 - DFAs

1. How do you know if a DFA M:

   (a) accepts the empty string

   Solution: $q_0 \in F$.

   (b) recognizes the empty language

   Solution: $\forall w \in \Sigma^*$, $\delta^*(q_0, w) \notin F$.

   (c) accepts some (i.e., at least one) string

   Solution: $\exists w \in \Sigma^*$ s.t. $\delta^*(q_0, w) \in F$.

2. We defined in class $\delta^*(q, w)$ by decomposing w into $w = xa$ where $a \in \Sigma$. However, as shown in class, we can also define $\delta_L^*(q, w)$ by $w = ax$ instead.

   (a) write that definition.

   Solution:

   $\delta_L^*(q, w) = q$ if $w = \epsilon$

   $\delta_L^*(q, w) = \delta_L^*(\delta(q, a), x)$ if $w = ax$.

   (b) prove (by induction) that these two definitions are equivalent, i.e.:

   $\forall q \in Q$, $\forall w \in \Sigma^*$, prove that $\delta^*(q, w) = \delta_L^*(q, w)$.

   Solution:

   Base case: If $w = \epsilon$, $\delta^*(q, w) = q = \delta_L^*(q, w)$.

   If $w = a$ where $a \in \Sigma$, $\delta^*(q, w) = \delta(q, a) = \delta_L^*(q, w)$.

   Inductive case: Assume the $\delta^*(q, w) = \delta_L^*(q, w)$ for $|w| \leq n$. (IH)

   When $|w| = n + 1$, let $w = axb$,

   $\delta^*(q, w) = \delta(\delta^*(q, ax), b)$ (by def. δ^*)

   $= \delta(\delta_L^*(q, ax), b)$ (by IH)

   $= \delta(\delta_L^*(\delta(q, a), x), b)$ (by def. δ_L^*)

   $= \delta(\delta^*(\delta(q, a), x), b)$ (by IH)

   $= \delta^*(\delta(q, a), xb)$ (by def. δ^*)

   $= \delta_L^*(\delta(q, a), xb)$ (by IH)

   $= \delta_L^*(q, w)$ (by def. δ_L^*)

(c) is there another way of decomposing w? how about $w = xy$ where $x, y \in \Sigma^*$. write your full definition, and make sure it is correct! (test boundary cases)

Solution:

$\delta^*_{LR}(q, w) = \delta^*_{LR}(\delta^*_{LR}(q, x), y)$ if $|w| \geq 2$
$\delta^*_{LR}(q, w) = \delta(q, w)$ if $w \in \Sigma$
$\delta^*_{LR}(q, w) = q$ if $w = \epsilon$

(d) prove (by induction) that your new definition is equivalent to $\delta^*(q, w)$.

Solution:

Base case: If $w = \epsilon$, $\delta^*_{LR}(q, w) = q = \delta^*(q, w)$.
If $w = a$, $a \in \Sigma$, $\delta^*_{LR}(q, a) = \delta(q, a) = \delta^*(q, w)$.
Inductive case: Assume $\delta^*_{LR}(q, w) = \delta^*(q, w)$ for $|w| \leq n$. (IH1)
For $|w| = n + 1$, let $w = xy$, further induction on the length of x to prove $\delta^*_{LR}(q, w) = \delta^*(q, w)$:
Base case: If $|x| = 1$, $\delta^*_{LR}(q, xy) = \delta^*_L(q, xy) = \delta^*(q, xy)$
Inductive case: Assume $\delta^*_{LR}(q, w) = \delta^*(q, w)$ for $|w| = n + 1$ and $|x| = m$. (IH2)
When $|x| = m + 1$, let $x = za$,
$\delta^*_{LR}(q, zay) = \delta^*_{LR}(\delta^*_{LR}(q, za), y)$ (by def. δ^*_{LR})
$= \delta^*(\delta^*(q, za), y)$ (by IH1)
$= \delta^*(\delta(\delta^*(q, z), a), y)$ (by def. δ^*)
$= \delta^*_L(\delta^*(q, z), ay)$ (by def. δ^*_L)
$= \delta^*_{LR}(\delta^*_{LR}(q, z), ay)$ (by (b) and IH1)
$= \delta^*(q, zay)$ (by IH2)

3. Read **only the first 7 pages** of DFA Equivalence and Minimization: https://www.cse.iitb.ac.in/~trivedi/courses/cs208-spring14/lec05.pdf

   Now read our slides 40–41 which showed two examples of testing whether two DFAs are equivalent
   using a variant of the above algorithm (table-filling of state-pairs (p,q) where p is from DFA1
   and q from DFA2).

   (a) Now prove that the 4-state and 3-state solutions of "binary number divisible by 4" are equivalent.

   Solution:

   Both q_0s in the two DFAs are start state.

pair	final	on 0	on 1
(q_0, q_0)	(y, y)	(q_0, q_0)	(q_1, q_1)
(q_1, q_1)	(n, n)	(q_2, q_2)	(q_3, q_1)
(q_3, q_1)	(n, n)	(q_2, q_2)	(q_3, q_1)
(q_2, q_2)	(n, n)	(q_0, q_0)	(q_1, q_1)

(b) Show a simple example where you draw two DFAs which are not equivalent according to the algorithm,
and show a string that is accepted by one DFA and rejected by the other.

Solution:

DFA1: start=q_0, only one transition q_0-0->q_0, $q_0 \in F$

DFA2: start=q_0, $q_0 \in F$

pair	final	on 0
(q_0, q_0)	(y, y)	(q_0, trap)

string: 0

4. What if δ is a partial function (omitting trap state)?
   How would you extend a partial δ to δ^*?
   Write the full definition, and make sure it's correct.
   Solution:
   definition of partial δ_p^*:
   $\delta_p^*(q, w) = q$ if $w = \epsilon$
   $\delta_p^*(q, xa) = \delta_p(q^{'}, a)$ if $q^{'}$ is defined and $\delta_p(q^{'}, a)$ is defined
   $\delta_p^*(q, xa)$ is undefined if $q^{'}$ is undefined or $\delta_p(q^{'}, a)$ is undefined
   where $q^{'} = \delta_p^*(q, x)$ and $q = xa$ in which $a \in \Sigma$.

   definition of correctness:
   δ_p^* is correct if.f. $\forall w$
   $\delta^*(q, w) \neq t \iff \delta_p^*(q, w) = \delta^*(q, w)$, and
   $\delta^*(q, w) = t \iff \delta_p^*(q, w)$ is undefined.
   where t stands for trap state.

proof of correctness ($\forall w$, $\delta_p^*(q, w)$ is correct) :

Base Case: if $w = \epsilon$, $\delta_p^*(q, w) = q = \delta^*(q, w)$.

Inductive Case: assume δ_p^* is correct for $|w| \leq n$.

let $|w| = n + 1$, denote $w = xa$ in which $a \in \Sigma$, $q' = \delta_p^*(q, x)$, and t for trap state.

(1) if q' is undefined,

$\delta^*(q, xa) = \delta(\delta^*(q, x), a) = \delta(t, a) = t$. (by IH on the second to last $=$)

$\delta_p^*(q, xa)$ is undefined (by definition)

correctness holds.

(2) if q' is defined but $\delta_p(q', a)$ is undefined.

$\delta^*(q, xa) = \delta(\delta^*(q, x), a) = \delta(q', a) = t$. (by IH on the second to last $=$)

$\delta_p^*(q, xa)$ is undefined (by definiton)

correctness holds.

(3) Both q' and $\delta_p(q', a)$ are defined.

$\delta^*(q, xa) = \delta(\delta^*(q, x), a) = \delta(q', a)$ (by IH on last $=$)

$\delta_p^*(q, xa) = \delta_p(q', a) = \delta(q', a)$ (by definition of δ_p on last $=$)

correctness holds.

5. Prove $|uv| = |u| + |v|$ by induction on $|u|$.

Solution:

Base case: If $u = \epsilon$, $|uv| = |v| = |u| + |v|$.

Inductive case: Assume the statement holds for $|u| = n$.

For $|u| = n + 1$, $u = wa$,

$|uv| = |wav| = |w| + |av|$ (by IH)

$= |w| + 1 + |v|$ (by IH)

$= |u| + |v|$.

6. Prove $(uv)^R = v^R u^R$ by induction (on what?).

   x^R is the reverse language of x. First define reverse inductively.

   Solution:

   Definition of x^R:

   $x^R = \epsilon$ if $x = \epsilon$

   $x^R = y^R a$ if $x = ay$ where $a \in \Sigma$

   Proof:

   Base case: If $u \in \Sigma$, apparently the statement holds.

   Inductive case: Assume the statement holds for $\forall u, |u| = n$. For $|u| = n + 1$, let $u = aw$.

   $(uv)^R = (awv)^R$

   $= (wv)^R a$ (by def. x^R)

   $= v^R w^R a$ (by IH)

   $= v^R u^R$ (by def. x^R).

7. What's wrong with this proof?

   Theorem(?!): All horses are the same color.

   Proof: Let $P(n)$ be the predicate "in all non-empty collections of n horses, all the horses are the same color."

   We show that $P(n)$ holds for all n by induction on n (using 1 as the base case).

   Base case: Clearly, $P(1)$ holds.

   Induction case: Given $P(n)$, we must show $P(n + 1)$.

   Consider an arbitrary collection of $n + 1$ horses. Remove one horse temporarily. Now we have n horses and hence, by the induction hypothesis, these n horses are all the same color. Now call the exiled horse back and send a different horse away. Again, we have a collection of n horses, which, by the induction hypothesis, are all the same color. Moreover, these n horses are the same color as the first collection. Thus, the horse we brought back was the same color as the second horse we sent away, and all the $n + 1$ horses are the same color.

   Solution: For $n + 1 = 2$, i.e., $n = 1$, the claim that second collection of 1 horse is of the same color as the first collection of 1 horse does not hold.

8. How about this one?

Theorem(?!): $n^2 + n$ is odd for every $n \geq 1$.

Proof: By induction on n (again starting from 1). For the base case, observe that 1 is odd by definition. For the induction step, assume that $n^2 + n$ is odd; we then show that $(n+1)^2 + (n+1)$ is odd as follows.
$(n+1)^2 + (n+1) = n^2 + 2n + 1 + n + 1 = (n^2 + n) + (2n + 2)$. But $n^2 + n$ is odd by the induction hypothesis, and $2n + 2$ is clearly even. Thus, $(n^2 + n) + (2n + 2)$ is the sum of an odd number and an even number, hence odd.
Solution: Base case where $n = 1$ does not hold.